
Association Learning inspired by the Symbol Grounding Problem



Thesis approved by
the Department of Computer Science
Technische Universität Kaiserslautern
for the award of the Doctoral Degree
Doctor of Natural Sciences (Dr. rer. nat)

to

Federico Alberto Raue

Date of Defense: 20th-August-2018
Dean: Prof. Dr. Stefan Deßloch
Reviewer: Prof. Dr. Prof. h.c. Andreas Dengel
Reviewer: Prof. Dr. Marcus Liwicki

D 386

ABSTRACT

The Symbol Grounding Problem (SGP) is one of the first attempts to proposed a hypothesis about mapping abstract concepts and the real world. For example, the concept "ball" can be represented by an object with a round shape (visual modality) and phonemes /b/ /a/ /l/ (audio modality). This thesis is inspired by the association learning presented in infant development. Newborns can associate visual and audio modalities of the same concept that are presented at the same time for vocabulary acquisition task.

The goal of this thesis is to develop a novel framework that combines the constraints of the Symbol Grounding Problem and Neural Networks in a simplified scenario of association learning in infants. The first motivation is that the network output can be considered as numerical symbolic features because the attributes of input samples are already embedded. The second motivation is the association between two samples is predefined before training via the same vectorial representation. This thesis proposes to associate two samples and the vectorial representation during training. Two scenarios are considered: sample pair association and sequence pair association.

Three main contributions are presented in this work. The first contribution is a novel Symbolic Association Model based on two parallel MLPs. The association task is defined by learning that two instances that represent one concept. Moreover, a novel training algorithm is defined by matching the output vectors of the MLPs with a statistical distribution for obtaining the relationship between concepts and vectorial representations. The second contribution is a novel Symbolic Association Model based on two parallel LSTM networks that are trained on weakly labeled sequences. The definition of association task is extended to learn that two sequences represent the same series of concepts. This model uses a training algorithm that is similar to MLP-based approach. The last contribution is a Classless Association. The association task is defined by learning based on the relationship of two samples that represents the same unknown concept.

In summary, the contributions of this thesis are to extend Artificial Intelligence and Cognitive Computation research with a new constraint that is cognitive motivated. Moreover, two training algorithms with a new constraint are proposed for two cases: single and sequence associations. Besides, a new training rule with no-labels with promising results is proposed.

DEDICATION AND ACKNOWLEDGEMENTS

I am really grateful for all the people who have helped and supported me during my Ph.d. student life, which the outcome is this thesis.

Firstly, I would like to express my sincere gratitude to my advisor Prof. Andreas Dengel and Prof. Marcus Liwicki for the continuous support and encouragement. Their guidance helped me in all the time of research and writing of this thesis. Additionally, I would also like to thank Prof. Thomas M. Breuel for his valuable guidance at the beginning of research. Furthermore, I would also like to thank the members of my committee Prof. Katharina Zweig and Prof. Marius Kloft for their comments and suggestions.

Secondly, I would like to thank my colleagues at IURP and MADM for the stimulating discussions about my research work and for all the fun we have during this period. I am really grateful for helping to review my thesis and the oral defense to Wonmin Byeon, Sebastian Palacio, Mohammad Reza Yousefi, Karsten Droste, Cristina Guerrero, Nervo Verdezoto, Alfredo Cevallos, Joern Hees, Joachim Folz, Tushar Karayil, Philipp Blandfort, Heiko Maus, and Ansgar Bernardi. Also, I want to thank my former colleagues Nibal Nayef, Adnan UI Hasan, Ludwig Schmidt-Hackenberg, Ilya Mezhirov, and Saurav Biswas for the great discussions. I would like to express my appreciation for Ingrid Romani and Brigitte Selzer for their administrative supports

Last but not least, I would like to thank my big family. My wife Irina was the real star of this research work and dissertation. She has always been the most crucial support in this adventure in Germany. I am looking forward to seeing what life brings to us. I want to dedicate this achievement to my mother Anita Elena, my mother-in-law Svietita, my sister Maria Elena, and my brother Francisco. I am really grateful for their spiritually support through writing this thesis and in my life. Also, I want to thanks my grandmother, uncles, aunts, and cousins, for keeping me in their thoughts and prayers.

TABLE OF CONTENTS

	Page
List of Tables	ix
List of Figures	xi
Abbreviations	xv
1 Introduction	1
1.1 Symbol Grounding Problem (SGP)	2
1.2 Infant Learning	6
1.2.1 Visual Perception	6
1.2.2 Auditory Perception	7
1.2.3 Link between Visual and Auditory Perceptions	8
1.3 Association Learning	11
1.4 Research Goals and Hypotheses	12
1.5 Contributions	13
1.6 Thesis Structure	14
2 Background	17
2.1 Supervised Classification	17
2.2 Neural Networks (NNs)	20
2.2.1 Multi-layer Perceptrons (MLPs)	20
2.2.2 Recurrent Neural Networks (RNNs)	22
2.2.3 Long Short-Term Memory (LSTM) Networks	25
2.2.4 Connectionist Temporal Classification (CTC)	28
2.3 Summary	30
3 Association Learning Framework	31
3.1 Problem Definition	31

TABLE OF CONTENTS

3.2	General Framework for Symbolic Association Learning	34
3.3	Symbolic MLP-based Approach	35
3.4	Symbolic LSTM-based Approach	40
3.5	Summary	44
4	Association Learning for Input Pairs	47
4.1	Problem Definition	48
4.2	Datasets	49
4.3	Features and Network Setups	51
4.4	Results and Discussion	52
4.4.1	Mono-modal Scenario	53
4.4.2	Multi-modal Scenario	55
4.5	Summary	59
5	Association Learning in Sequences	63
5.1	Problem Definition	64
5.2	Scenario 1: Parallel Mono-modal Sequences	65
5.2.1	Dataset Preparation	66
5.2.2	Input Features and LSTM setup	67
5.2.3	Mono-modal Latent Space produced by one LSTM Network	67
5.2.4	Mono-modal Latent Space produced by two LSTM Networks	70
5.3	Scenario 2: Parallel Multi-modal Sequences	71
5.3.1	Dataset Preparation	71
5.3.2	Input Features and LSTM setup	74
5.3.3	Multi-modal Latent Space produced by two LSTMs	75
5.4	Summary	79
6	Association Learning in Sequences with missing Concepts	81
6.1	Problem Definition	82
6.2	Handling Missing Elements	82
6.3	Experiments	84
6.3.1	Multi-modal Setups	86
6.3.2	Input Features and LSTM setup	87
6.4	Results and Discussion	88
6.5	Summary	92
7	Classless Association	93

7.1	Problem Definition	94
7.2	Model	95
7.3	Datasets and Network Setups	100
7.4	Results and Discussion	101
7.5	Summary	103
8	Conclusion and Future Work	107
8.1	Concluding Remarks	108
8.2	Future Directions	109
8.2.1	LSTM-based Approach	111
8.2.2	Classless Association (MLP-based Approach)	111
	Bibliography	113
	Curriculum Vitae	125

LIST OF TABLES

TABLE	Page
4.1 Summary of the mono- and multi-modal sizes	51
4.2 Association Accuracy (%) of the presented model and the traditional approach in the mono-modal scenario	53
4.3 Accuracy (%) of this work and the standard MLP in the mono-modal scenario	53
4.4 Association Accuracy (%) of the symbolic association model and the traditional setup in the multi-modal scenario	57
4.5 Accuracy (%) of the presented model and the traditional setup in the multi- modal scenario.	57
5.1 Sequence Association Accuracy (%) and Label Error Rate (%) of one LSTM network trained independently to each input set and the symbolic association model (implemented by one LSTM).	67
5.2 Sequence Association Accuracy (%) and Label Error Rate (%) between one LSTM network trained independently to each input set and the symbolic association model (implemented by two LSTM networks)	71
5.3 Sequence Association Accuracy (%) and Label Error Rate (%) of the standard LSTM and the symbolic association model in the multi-modal scenario	76
6.1 Sequence Association Accuracy (%) and Label Error Rate (%) from the multi- modal configuration of missing concepts in both modalities	89
7.1 Association Accuracy (%) and Purity (%) results in the classless association scenario	103

LIST OF FIGURES

FIGURE	Page
1.1 Example of Chinese Room Argument	3
1.2 Example of the three mental representations	4
1.3 Simplified pipeline of the association learning in infants	6
1.4 Example of the semantic network of infants between 6- and 18-months old . .	10
2.1 Example of the supervised tasks	18
2.2 Example of the sequence classification	19
2.3 Example of Multilayer Perceptrons	20
2.4 Example of a Vanilla Recurrent Neural Network	23
2.5 RNN unfolded over time	23
2.6 Bidirectional RNNs	26
2.7 LSTM cell	27
2.8 Example of a weakly labeled sequence	28
2.9 Example of the LSTM classification based on CTC training	30
3.1 Example of semantic concepts and vectorial representations	33
3.2 Overview of the association learning framework	35
3.3 Example of the weighting vector role	37
3.4 Example of the elimination process	38
3.5 General overview of the MLP-based approach	40
3.6 Example of Euclidean and DTW alignments	42
3.7 General overview of the LSTM-based approach	45
4.1 Difference between the traditional setup and this work concerning learning elements	48
4.2 Examples of the mono- and multi-modal datasets	50
4.3 Confusion Matrices of traditional setup and this work (MNIST Dataset) . . .	54

4.4	Confusion Matrices of the traditional setup and the symbolic association framework (COIL-20 Dataset)	55
4.5	Example of the training process at different stages (mono-modal scenario) . .	56
4.6	Confusion Matrices between the traditional setup and the symbolic association model (Wikipedia Dataset)	58
4.7	Confusion Matrices between the traditional setup and the symbolic association model (TVGraz Dataset)	59
4.8	Example of the training algorithm at different stages (TVGraz Dataset). . . .	60
5.1	Differences between components of traditional and symbolic association tasks when the input samples are sequences	64
5.2	Several examples of sequences in the mono-modal dataset	66
5.3	Example of the Symbolic Association Learning implemented with one <i>Long Short-Term Memory</i> (LSTM)	68
5.4	Several examples of the classification with the symbolic association framework, one LSTM version	69
5.5	Association Learning using two LSTMs in the mono-modal scenario	70
5.6	Several examples of the prediction step	72
5.7	Examples of the three multi-modal datasets	73
5.8	Example of the symbolic association in the multi-modal scenario	75
5.9	Example of the learning behavior in the multi-modal latent space	77
5.10	Several examples of the prediction in the multi-modal scenario	78
6.1	Association between sequences with partial alignment	83
6.2	Example of LSTM-based approach that handle sequences with missing elements	85
6.3	Example of the multi-modal configuration with missing elements	88
6.4	Several examples of the output classification and DTW cost matrices with missing elements in the multi-modal sequences	90
6.5	Sequence Association Accuracy (%) of second and third multi-modal configurations with several missing concepts	91
6.6	Label Error Rate (%) of the second and third multi-modal configurations with several missing concepts	91
7.1	Problem definition of the classless association	94
7.2	Loss function based on a statistical distribution	97
7.3	Overview of the classless association model	98

7.4	Example of the training algorithms of the classless association	102
7.5	Examples of the best and the worst results in the classless association	104
7.6	Limitation of the classless model related to the number of iterations	104
8.1	Limitation of the classless association model	110

ABBREVIATIONS

AAcc *Association Accuracy.*

AI *Artificial Intelligence.*

AL *Association Learning.*

BoVW *Bag-of-Visual-Word.*

BPTT *Backpropagation Through Time.*

CC *Cognitive Computation.*

CS *Cognitive Science.*

CTC *Connectionist Temporal Classification.*

DTW *Dynamic Time Warping.*

EM *Expectation-Maximization.*

HMM *Hidden Markov Model.*

IL *Infant Learning.*

LDA *Latent Dirichlet Allocation.*

LER *Label Error Rate.*

LSTM *Long Short-Term Memory.*

ML *Machine Learning.*

MLP *Multilayer Perceptron.*

ABBREVIATIONS

MSE *Mean Square Error.*

NN *Neural Network.*

NS *Neuroscience.*

RNN *Recurrent Neural Network.*

SeqAAcc *Sequence Association Accuracy.*

SGP *Symbol Grounding Problem.*

SOM *Self-Organizing Map.*

INTRODUCTION

The human brain is an essential inspiration for *Artificial Intelligence* (AI) when designing and creating computational models, e.g., HMAX [1] and SpikeNet [2]. Computer Vision is one example that simulates similar behavior to the brain, such as, recognizing objects in images. Not only AI but *Cognitive Science* (CS), *Neuroscience* (NS) and *Cognitive Computation* (CC) also investigate how the brain works. For example, Goodale and Milner [3] formulated a hypothesis that two paths are in the primate cortex. One path is linked to vision-for-perception whereas the other path is connected to vision-for-action. Riesenhuber and Poggio [1] proposed the HMAX model for object classification that was inspired by the vision-for-perception path.

A challenging scenario that computation models are well suited to tackle is *Infant Learning* (IL) [4, 5]. In this case, newborns require learning with minimal or no knowledge of their environment. One of the first learning elements is to map *abstract concepts* to their sensory input (in the *vocabulary acquisition task*) [6]. As a result, infants learn that the meaning of a word is associated to several modality representations. Other forms are also important, such as motor sensory or social cues, which are not considered in this thesis [7, 8].

The goal of this thesis is to present a general framework that links cognitive elements and computational models. In this scenario, three concepts related to human cognitive tasks are important to define for setting the constraints of the framework. First, the

Symbol Grounding Problem (SGP) proposes a cognitive theory about how words have their meaning. Second, IL has useful insight into the visual and auditory perceptions and their cross-modal relationship, which the proposed model relies on exploiting the cross-modal information. Third, *Association Learning* (AL) is the link between two or more sensory signals that represent the same meaning. Vision and auditory samples are only considered in this thesis.

1.1 Symbol Grounding Problem (SGP)

Several fields, such as AI, NS, CS, and CC, have been interested in how the human brain works for a long time. One of the most striking features is how the meanings of the words are born. For example, a roundish shaped object and a spoken word /b/ /a/ /l/ represents the abstract concept *ball*. Also, the meaning of another word *violin* can be expressed not only in the visual perception (shape, color, and texture), but also in the audio perception (sound, pitch, and timbre). The abstract concept remains hidden in our brain as a *prototype*. Similarly, the letters and digits (that represent words and numbers, respectively) are elements that humans know how to manipulate. For example, a sequence of written letters represents a word, and a sequence of digits illustrates a mathematical operation.

Harnad [9] proposed that the *mind* is a symbolic system, and the *cognition* is the manipulation of those symbols. He called it the *Symbol Grounding Problem* (SGP). However, the meaning of a symbol is unclear in this context. For example, Merriam-Webster ¹ defines the term *symbol* as:

1. an authoritative summary of faith or doctrine
2. something that stands for or suggests something else by reason of relationship, association, convention, or accidental resemblance; especially
3. an arbitrary or conventional sign used in writing or printing relating to a particular field to represent operations, quantities, elements, relations, or qualities
4. an object or act representing something in the unconscious mind that has been repressed

¹Symbol. (n.d.). Retrieved July 9, 2017, from <https://www.merriam-webster.com/dictionary/symbol>

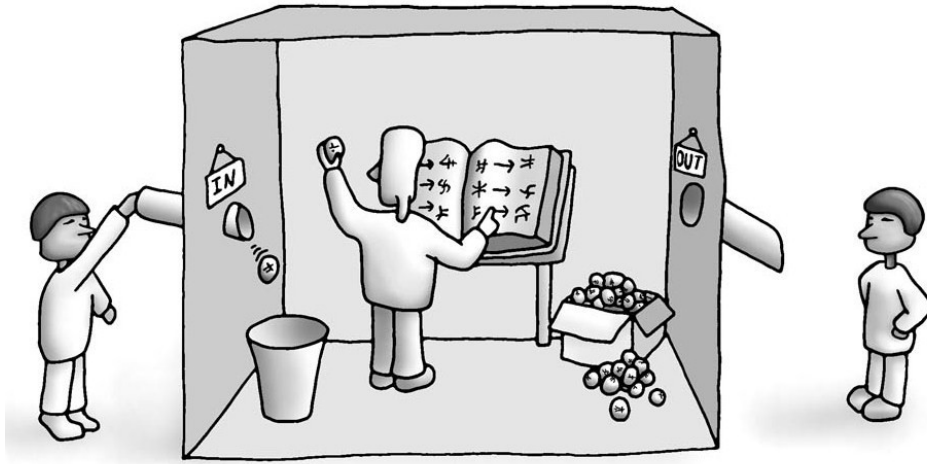


FIGURE 1.1. Example Chinese Room Argument. The person inside of the room who does not know Chinese has a *dictionary* with all Chinese characters. The hypothesis is that the person inside receives a question in Chinese and replies in Chinese using the dictionary. Image retrieved from <https://blog.cloudmiddleman.com/middlemanning-for-fun-and-profit-dd998b32e973>.

5. an act, sound, or object having cultural significance and the capacity to excite or objectify a response

These definitions are too general for the grounding process between symbols and word meanings. Searle [10] explained the term *symbol* with a well-known example: the *Chinese Room Argument*. In this example, one person is inside of the room, and another person is outside of a room. The person inside of the room only speaks English, and the other person only speaks Chinese (see Figure 1.1). In that case, Searle claimed that the person inside could communicate in Chinese if he/she can manipulate (read, understand, and write) the *iconic* characters based on a dictionary (Chinese-English) that contains all possible characters and rules. This example shows that the manipulation is based on the recognition of the shapes but not on the meaning of the words.

Furthermore, Harnad [9] described that the meaning of the words involves three mental representations: iconic, categorical, and symbolic. Iconic representation is the physical projection of the real world. Categorical representation is the learned feature for recognizing objects and events. Symbolic representation is the mapping between the names and the objects. Figure 1.2 shows two examples of the three mental representations.

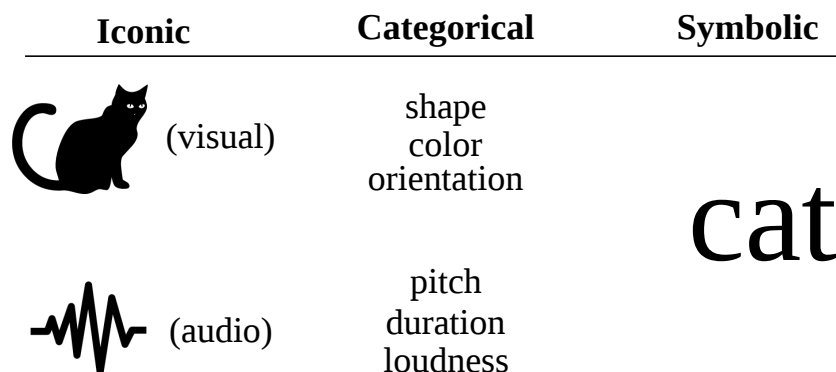


FIGURE 1.2. Example of the three mental representations defined by Harnad [9].

Many challenges remain open in SGP. Mayo [11] presented three limitations regarding AI (robots and artificial life). First, the grounding step between abstract concepts and sensory input is still unknown where it comes and how it works. Infants especially do not know how to ground symbols and acquire information via sensory input (e.g., eyes and ears). Also, an unknown criterion is required for choosing the best category of the iconic information. Second, concepts related to the visual domain are easier to ground because there is a clear representation, such as cat, dog, ball, pizza. However, other abstract concepts are not attached to a specific visible representation, such as politics, victory, or love. Consequently, it is unclear (from the robotics point of view) when the grounded information for defining an intelligent mind. Similar to Mayo, Steels [12] added more observations to SGP. To begin with, the relationship between *meaning*, *conceptualization*, and *symbolization* remains unknown. Moreover, the brain fires neurons depending on the semantic meaning of a phrase. For example, smell words fire neurons in the olfactory processing area, and action words trigger the motor area.

In addition to these open observations, there is not a consensus of the components with their respective tasks in SGP. Taddeo and Floridi [13] summarized the most common approaches into three scenarios: a) representationalism, b) semi-representationalism, and c) non-representationalism. The first approach defines the meaning of the symbol that relies on the conceptual and categorical representations. The second approach has some elements from the first approach but also incorporated ideas from behavior-based robotics. One example of the robotics principles is to define a concept with "a composite description of several components". Another example of behavior-based on robotics is to consider symbols with three elements: a) a *form*, which is the physical representation

(or shape), b) a *meaning*, which is the semantic concept, and c) a *referent*, which is the object that is referred by the input signal. The third approach proposes that a symbolic representation is not required because of intelligent behavior and the interaction in the environment. In other words, intelligent behavior is the only the connection between the sensor information and the actions without any semantic meaning.

SGP has inspired several applications. Some of them are specific, such as Grounding Symbols in the Semantic Web [14], and other cases are too general, e.g. Symbol Emergence in Robotics [15]. Coradeschi et al. [14] have proposed several types of SGP, such as Grounding Words in Actions, Social Symbol Grounding, and even, Grounding Symbols in the Semantic Web. Tellex et al. [16] explained a model that aligns a video of the robot navigation and commands giving in natural language. In this case, the authors described a probabilistic graphical model that can ground objects, places, and paths in the real world (robotics domain) inspired by SGP. Di Nuovo et al. [17] showed a model related to the transition between digit recognition and digit manipulation (math operations). This transition stage is similar to how infants first learn to recognize digits (visual and audio, even to use their limbs to represent them). The authors proposed a neural network that learns the association between visual, audio, and haptic components.

Another task inspired by SGP is giving semantic similarity and relatedness. Similar to the visual grounding, a concept can exploit visual and auditory perceptions. For example, a person can ground the concept *car* not only to shapes and colors but also to sounds (i.e., closing a door, the sound of the engine, the ignition of the car). The authors compared several fusion strategies for combining linguistic and auditory representations [18].

Taniguchi et al. [15] have proposed *Symbol Emergence in Robotics (SER)*, which is inspired by the SGP. In this case, they have summarized several research topics that are important in robotics. Some of them are multi-modal communication, concept formation, language acquisition and mental development, learning interaction strategy, learning motor skills and segmentation of time-series information.

In summary, SGP is a challenging scenario, in which several authors proposed many solutions. Finding the required elements that are involved in the process of semantic acquisition is vital for language understanding. Learning the meaning of words in the early ages is one example of this scenario. Especially, how infants start learning to recognize objects and sounds including the mapping to semantic knowledge.

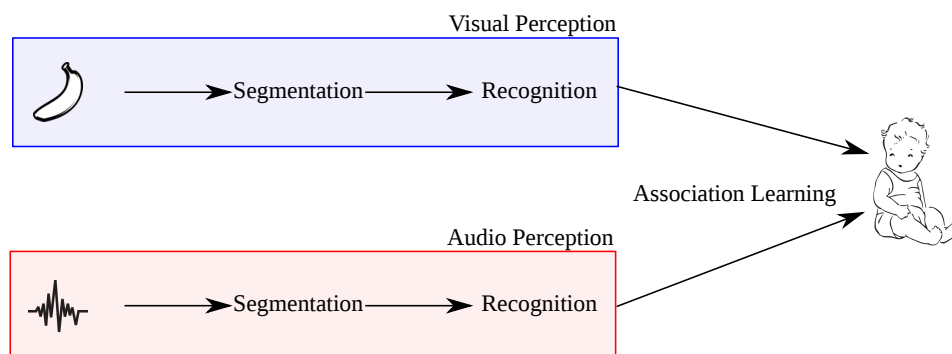


FIGURE 1.3. A simplified pipeline of the association learning in infants. The problem definition in this scenario considered only two tasks from Machine Learning approaches for each modality: segmentation and recognition.

1.2 Infant Learning

Infants can acquire semantic knowledge employing their sensory input. The research community has been interested in finding out how infants learn. Infant Language Center at the University of Pennsylvania², Infant and Child Studies Lab at the University of Toronto Mississauga³, Baby Research Center Nijmegen⁴ are examples of research institutes in Infant Learning. These labs investigate several tasks, e.g. understand language development, measuring attention of newborns.

Additionally, a computation model based on machine learning problem can be inspired by IL. Figure 1.3 shows a minimal scenario where infants learn to segment and recognize two different signals (or iconic representations) including the association between them. This work is constrained to two modalities: visual and audio (audio can also be represented in textual descriptions). However, more factors are also crucial to consider for vocabulary acquisition, such as motor-sensor, attention, and social cues.

1.2.1 Visual Perception

Initially, 3-months old infants show viewpoint invariance, which the ability to recognize objects regardless of the viewpoint. The viewpoint invariance in infants is supported by the following experiments. In that case, Kraebel and Gerhardstein [19] trained several infants with multi-part objects that are horizontally oriented and tested with vertical

²<http://www.psych.upenn.edu/infant/home.html>

³<http://www.utm.utoronto.ca/infant-child-centre/>

⁴<http://www.babyresearchcenter.nl/en/>

oriented objects. Another experiment relies on images that look 2D and 3D objects. Bhatt and Waters [20] found that infants can discriminate images that look like 3D objects (from an adult point of view) whereas failed to detect similar discrepancies in 2D images. The authors inferred that infants could exploit line intersection (similar to adults that exploit constellation of Y, arrow, and T junctions), and shade information for extracting 3D cues.

After 3-months old, infants start developing the features for object recognition. Wilcox [21] proposed a timeline of the developing of common visual attributes: shapes, sizes, patterns, and color. These features are more sensitive depending on age. Initially, 4.5-month-old infants can exploit the shape and size of the objects in events where the objects are partially occluded. The infants do not use the other visual feature (patterns and colors). In contrast, 7.5-month-old infants prefer using patterns, and 11.5-month-old infants prefer using colors for object identification.

Moreover, infants pay attention to the structure in multi-element scenes. Fiser and Aslin [22] found that infants around 9-month-old extract several features from their visual environment based on two information sources: a) co-occurrence frequency of elements and b) the predictability of relations between elements. To recognized *new* environment, infants considered any feature previously learned in *unknown* environments. As a result, the structure is useful for learning higher-level features and constraints. This result also is supported by Spelke [23]. Infants can segment a scene into objects given by three-dimensional surfaces and motions. They convert visual information into units that are connected while those units are moving, keeping the size and shape. Hence, infants might perceive elements that can make them infer about the unity of partially occluded objects. One of the elements are the edges between two adjacent objects.

1.2.2 Auditory Perception

One of the essential tasks that infants require for acquiring vocabulary is the capacity of speech segmentation. In other words, the identification process between word-like units and auditory speech input. Friederici and Wessels [24] have found features that infants use for speech segmentation. First, 6- and 12-month-old infants learn the phonotactic⁵ structure of their mother tongue. Second, 9-month-old infants prefer grammatically correct than and grammatically incorrect word boundaries. Third, the selection between

⁵the area of phonology concerned with the analysis and description of the permitted sound sequences of a language. "Phonotactics." Merriam-Webster.com. Merriam-Webster, n.d. Web. 11 July 2017.

legal over illegal word boundary is based on phonotactic features and not prosodic⁶ features.

Jusczyk [25] have found similar results for word segmentation in infants. The authors claimed that infants used more information sources instead of only using phonotactic features. Some cues related to stress (prosodic) and statistical information appears earlier than phonotactic and allophonic⁷. The authors suggested that this initial strategy for word segmentation helps to develop other features based on phonotactic and allophonic. At the end of the second year, infants have a similar performance to adults for recognizing familiar words.

Jusczyk et al. [26] have been working on an extended-analysis of how significant are the intonation in words. They have analyzed bisyllabic words with two types of intonation: 1) strong/weak and 2) weak/strong. In the first case, 7-month-old infants can correctly recognize all words. However, infants have failed to recognize words in the second case. Newborns have tried to use the strong accent as an anchor for words. For example, they recognize the phrase "guitar is" as "taris". Later on, 10-month-old infants have learned to recognize weak/strong words. Hence, infants between 6- and 10-month-old have learned how to integrate more information from multiple sources into word segmentation.

Pelucchi et al. [27] have demonstrated that infants have a statistical mechanism for learning languages. Eight-month-old infants can learn transitional probabilistic information in unknown speech stimuli. In that particular case, the authors evaluated infants (who their native language is English) with Italian speech stimuli.

1.2.3 Link between Visual and Auditory Perceptions

The vocabulary acquisition gives insight into the relationship between visual and auditory signals. For example, the McGurk effect [28] shows a perceptual phenomenon that links to the visual and auditory signals in the speech perception. Children learn a new word in their vocabulary if they follow these conditions [29]: used with the intention to communicate, has a consistent phonological shape, has a consistent meaning, and has extended to multiple exemplars.

⁶the rhythmic and intonational aspect of language. "Prosody." Merriam-Webster.com. Merriam-Webster, n.d. Web. 11 July 2017.

⁷one of two or more variants of the same phoneme. "Allophone." Merriam-Webster.com. Merriam-Webster, n.d. Web. 11 July 2017.

Words and tones (or fake words) are distinguishable by infants. Balaban and Waxman [30] have proposed a theory that words help to categorize objects. They evaluated two groups of 9-month-old infants in three experiments where one group uses words and the other group uses tones instead of words. Two of the three experiments have suggested that word phrases help to categorize because the audio stimuli enhance the infant visual attention. The authors therefore claimed a connection exists between words and objects because of initial attempts at mapping the words with their meanings. Furthermore, Graf Estes et al. [31] found in 17-month-old infants that they can discriminate between words, syllable sequences that are not words, and familiar sequences with low internal probabilities. This task evaluates if infants can learn the object-word association in these three different auditory stimuli. The results showed that the words are more accessible to being mapped to objects in comparison with the other two options. Asano et al. [32] have examined the brain activity of 11-month-old infant using three EEG-based measurements. The experiment was to evaluate one scenario where an object is semantically and correctly matched with auditory stimuli. Their findings were that brain activity changes in the three measures depending on matching or miss-matching between the object and the auditory stimuli.

An essential step in vocabulary acquisition is to learn nouns that have visual representations (e.g., cat, dog, ball, car). Figure 1.4 shows a semantic network of the most learned English words in 18-month-old infants. Many words are nouns (blue) in comparison to the other lexical classes (adjectives-green and other-yellow). Infants pay more attention to shape attributes for learning nouns. For example, the first nouns that infants learn have similar shapes between them [33]. There is a relationship between the size of the vocabulary acquisition and infants paying more attention to shape, which helps them learn new nouns. Yee et al. [34] found similar results, in which 18- and 24-month-old infants can develop a type of shape representations (between all samples of the same noun category) based on learning the name of objects (mainly nouns). Afterwards, this initial sparse shape representation helps to develop the shape bias and the perceptual similarity mechanism.

Werker et al. [35] analyzed the age of infants that can learn the multi-modal association between words and objects. In this scenario, the experiment evaluates if it is possible to associate without many external factors, such as interaction with the speaker, object manipulation, and no social or contextual cues to direct the attention between the object and the environment. Two groups of different ages have been evaluated under these

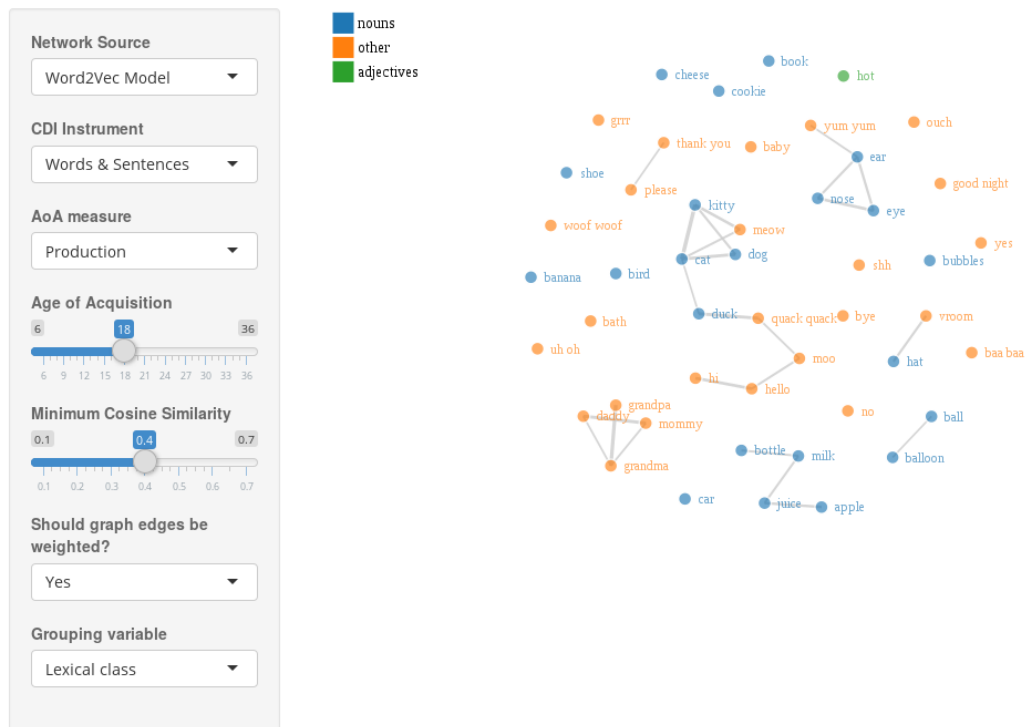


FIGURE 1.4. Example of the semantic network of infants between 6- and 18-months old. Retrieved from <http://wordbank.stanford.edu/analyses?name=networks>.

constraints. The first group (infants between 8- and 12-months old) could not associate the word and the object. The results support that the word and the object were analyzed independently. In contrast, the second group (14-month-old) can learn the matching between the word and the object without a problem.

Another result that supports the relationship between visual perception and vocabulary acquisition is provided by Pruden et al. [36]. In their work, 10-month-old infants tend to learn the mapping between words and object that they are more interested in. This can mislead the association between the word and the interested object. In fact, infants at this age do not consider social cues. In other words, infants are only taking into consideration what the speaker is saying.

Infants with blindness, deafness or hard of hearing tend to delay their vocabulary acquisition [37, 38]. However, Lederberg and Spencer [39] claimed that infants can develop novel strategies (because of brain plasticity) that accelerate the word learning process. For example, infants who are deaf develop novel strategies for visual attention. In such

cases, the parents need to make an extra effort for visual communication [40]. Similar to deaf infants, the parents of blind children try different communication strategies than for sighted children. For example, their language is more directive and structured in the way that parents encourage children to take a more active role regarding conversation and exploring their environment [41].

1.3 Association Learning

As mentioned in Sections 1.1 and 1.2, infants can associate different modalities under the same semantic concept. Aristotle in his book *Laws of Association*⁸ proposed one of the first definitions of *association* in memory, which described three conditions: spatial or temporal contact, similar elements, and different elements. Additionally, many applications are similar to infant learning. Parallel data is more common these days because of several sources of information, such as images with its description, photos that have GPS coordinates, visual and audio elements, and multiple sensor data. With this in mind, several tasks can exploit the relationship between them. For instance, image captioning tasks translate from images to textual descriptions. However, this relationship is in one direction. Another task can be seen as matching if two or more elements represent the same class. Many challenges have the association scenario if the input samples are multi-modal. The first challenge is the feature representation where one modality can exploit information of the other modality. Moreover, fusion techniques can be one way to improve the feature representation to take the best of both representations. The second challenge is the projection from one latent space to another latent space. In this manner, the similarity metric should adapt to the new constraints at the new vectorial space. There are several scenarios for learning associations between modalities. In this section, Partial Labeling, Neural Networks, and Embedded Cognition in Humanoid Robotics are briefly explained.

First, Partial Labeling considers that small portions of the dataset are labeled in the following manner. Some images have a set of candidate labels, but only one label is correct. Several researchers have tried to use images associated with text descriptions or a set of labels in a way that maximizes one label between all possible candidates using SVM [42–44].

⁸http://www.bcp.psych.ualberta.ca/~mike/Pearl_Street/Dictionary/contents/L/lawsofassoc.html

Second, neural networks approach learns to embed the relationship in their structure. In this case, only models that include a cognitive motivation are described. Two approaches are considered for the problem of image-word association: a) Feed-forward networks and b) *Self-Organizing Map* (SOM). One of the first works of auto-associative learning using Neural Networks was proposed by Plunkett et al. [45]. Their model learns to combine the visual perception and the word (meaning) using different architectures where one part of the network learns the visual information. Additionally, another part of the network learns the association. SOMs are neural networks that learns the similarity between elements in an unsupervised environment. In this architecture, the idea is to learn the association between the phonetics of the word and its corresponding visual information [46, 47]. Their network has two components. One component is a receptive field that works as a retinal input, and the other component is orthogonal vectors that represent labels. Another approach for vocabulary acquisition is also based on SOMs [48]. In that case, the model learns the meaning of the words based on the co-occurrence statistics. Li et al. [49] proposed a model that combines two SOMs for learning the association between the word meaning and word form. In this case, they evaluate several scenarios based on the type of semantic information (nouns, adjectives).

Third, the association between images and words can be learned by a humanoid robotic with embedded cognition. The humanoid detects a tutor and the tutor's gaze. Then, the robot learns based on the interaction or mimicking the tutor [50]. Another scenario is to align patches of a scene with their corresponding word. In this case, the same number of patches and words are given to the robot. After training based on graphical models, the robot can align accordingly [51].

In this work, the presented models rely on Neural Networks because the architectures are more biologically plausible when compared to Support Vector Machines (SVM) or Hidden Markov Models (HMM). HMAX model [1, 52], SpikeNet [2, 53] are examples of object recognition that rely on the *biological visual system*. The first approach is inspired by the Hierarchical features in the Visual Cortex, whereas, the second approach is inspired by how the neurons encode information in the brain.

1.4 Research Goals and Hypotheses

The goal of this thesis is to combine SGP and *Neural Network* (NN) into one framework. On the one hand, SGP is an open challenge related to how our brain works. On the other

hand, NNs have been successfully applied to different domains, such as Computer Vision and Speech Recognition. This framework is applied to an association task inspired by infant learning, in which infants learn to match parallel samples that represent the same abstract concepts.

The research hypothesis of this work can be stated as follows:

Neural Networks, mainly Multilayer Perceptrons (MLPs) and Long Short-Term Memory (LSTM), are biologically inspired. Therefore, cognitive constraints should be possible to be included into NNs because of their capacity of embedding discriminant information inside of their architecture. Moreover, the association task can be solved adding not only the recognition of the input samples but also the binding between semantic concepts and the real world. One approach exploits the capacity of MLP for the recognition task. Another approach exploits the capacity of LSTM for the sequence recognition task in weakly labeled data.

In this work, several test cases are used for evaluating this hypothesis. Moreover, the test cases are the association between isolated elements and sequence of elements. Also, the format of the pair samples can be mono- and multi-modal.

1.5 Contributions

The main contributions provided by thesis are to unify the Symbol Grounding Problem and Neural Networks in the association task. A summary of the contributions is described:

1. A new approach for considering the output of the networks, mainly Multi-layer Perceptrons (MLPs) and Long Short-Term Memory (LSTM) as numerical, symbolic features. This assumption is based on the power of neural networks that can embed attributes and features.
2. A theoretical framework for association learning and binding symbolic features and semantic concepts. In other words, this framework does not require the traditional definition of the mapping between semantic concepts and vectorial representations. In contrast, semantic concepts without vectorial representations define the association between pairs of input samples. Furthermore, the framework learns converges to an agreement during training.

3. A novel training algorithm for learning the agreement between two networks. In more detail, the algorithm includes the binding between semantic concepts and vectorial representation given a statistical distribution as part of the loss function. Several test cases have been used for evaluating this algorithm. One of the test cases is to associate pairs of isolated elements that represent the same semantic concept. The other test cases are related to parallel sequences that represent the same ordered sequence of semantic concepts.
4. A new multi-modal dataset is introduced for the association task. In this case, multi-modal sequences have semantic concepts that might present in one or both modalities.
5. A new training rule for learning the association between two isolated elements that represents the same unknown semantic concept. In this case, the classifier learns to match the raw output vector with a statistical distribution.
6. A novel approach for multi-modal learning to use two parallel LSTM networks with weakly labeled sequences. In this case, both LSTMs exploit a generated multi-modal latent space. Moreover, Dynamic Time Warping aligns one latent space produced by one LSTM to the other latent space produced by the other LSTM.

1.6 Thesis Structure

This work is presented into seven chapters. Chapter 2 describes the required background about Neural Networks for understanding the presented framework Chapter 3 presents the general association framework. Chapters 4 and 5 show the results of the association framework for two test cases. Chapter 6 introduces a new constraint for association multi-modal sequences with missing elements. Chapter 7 presents a new constraint for the association problem, which do not use labeled data. Chapter 8 presents the conclusion and future work.

Chapter 2 describes the background information of this thesis. Initially, a formal definition of a supervised task are given. NNs are described, mainly Multilayer Perceptrons (MLPs) and Long Short-Term Memory (LSTM). The presented association framework exploits the capacity and advantages of both architectures.

Chapter 3 explains a general framework for association learning inspired by the Symbol Grounding Problem. First, a problem definition of the association learning and its relation to the Symbol Grounding Problem is given. Second, the association learning framework and each component are explained. In this work, two approaches are used for implementing the association framework. One approach is based on MLP, and the other approach is based on LSTMs.

Chapter 4 presents the results of the framework implemented by MLPs. In this case, the task is to learn the association between two input samples that represent the same semantic concept. Two scenarios are used to evaluate the model: mono- and multi-modal samples. The performance of the presented model reaches similar results to MLP that is trained for the classification task for each input set.

Chapter 5 presents the results of the framework implemented by LSTMs. The association learning is between two parallel sequences. Similar to the MLP framework, two scenarios are also used for evaluating this model: mono- and multi-modal sequences. In both cases, the presented model reaches similar results to LSTM trained for the classification task in one modality.

Chapter 6 explains the model based on LSTM with an extension for handling multi-modal sequences with elements that can be or cannot be in both modalities. In this case, the alignment step is modified for handling only elements that are common in both modalities. The model is tested in two scenarios: random missing elements in both modalities, and a fixed number of missing elements in one modality.

Chapter 7 introduces a new model that learns the association between two elements that represent the same *unknown* semantic concept. The model is evaluated in four datasets. The performance of the classless model is better than two clustering algorithms, and in good range regarding the supervised case.

Chapter 8 summarizes the findings and contributions provided by this thesis. Additionally, the next challenges and future research questions are discussed.

BACKGROUND

The previous chapter described SGP concerning the association between two samples. This chapter explains critical components of the association learning framework in the context of machine learning task. First, *supervised classification* is a task where the goal is to map input samples to categories. In the context of SGP, the categories are abstract concepts that map to the real world. Second, one approach for solving the supervised classification task is *Neural Networks*. Two architectures are essential in this thesis: *Multilayer Perceptrons (MLPs)* and *Recurrent Neural Networks (RNNs)*.

2.1 Supervised Classification

Many examples of *supervised classification* are in the market. For instance, digital cameras can detect if a face is smiling for taking a picture. Supervised classification is a mapping task between elements (e.g., images or texts) and categories. More formally, the mapping function $f : \mathbf{x} \in R^n \rightarrow \{1, \dots, k\}$ where \mathbf{x} is an input sample and the set $\{1, \dots, k\}$ represents possible categories. The following mathematical expression describes a mapping function.

$$y = f(\mathbf{x}), \tag{2.1}$$

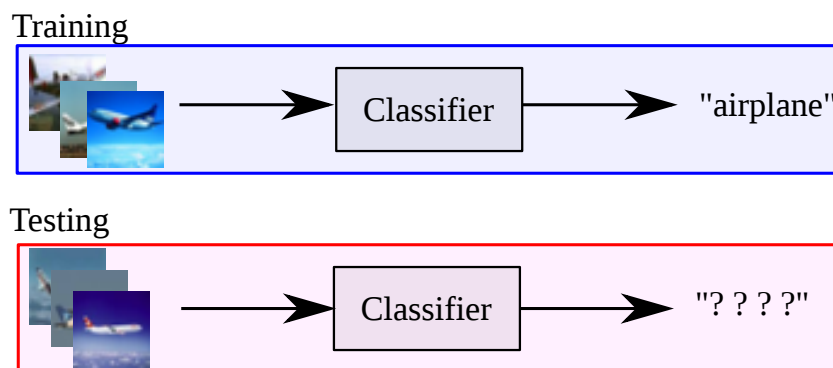


FIGURE 2.1. Example of the object classification task. The training step is based on input samples of the target category. In this case, each image has only one category. Input images are taken from CIFAR-10 dataset [54].

where y is a category represented by a numeric code. As a result, Equation (2.1) is updated to

$$\hat{\mathbf{y}} = f^*(\mathbf{x}; \theta), \quad (2.2)$$

where f^* is a model (called classifier) that approximates f , θ is the parameters of the model, and $\hat{\mathbf{y}} \in R^k$ is the output category given the input \mathbf{x} .

The output vector $\hat{\mathbf{y}}$ is represented commonly by a *one-hot scheme*. This vectorial representation has all set to zero, except a single element, which is set to one. An example of this coding scheme notation is shown as follows

$$\text{category: airplane} \leftrightarrow \mathbf{e}_1^T = [0, 1, 0, \dots, 0, 0], \quad (2.3)$$

where $\mathbf{e}_1 \in R^k$ is a unit vector, and the index 1 represents the position where the vector is not zero. Thus, each category has different representations that are mutually exclusive between them.

So far, supervised classification has been only explained considering "isolated elements" (each image represents only one category). One extension of isolated elements consists of classifying input samples that represent a set of ordered categories (i.e., sequence labels). Similar to the previous scenario, the association learning can be seen

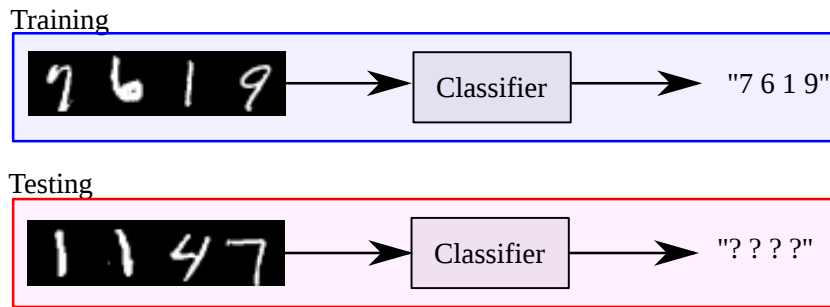


FIGURE 2.2. Example of the sequence classification. The main difference with the object classification task is related to the categories. In this case, each input sample is represented by a series of categories. Note that the training step requires image samples and categories. The testing step classifies input samples into categories. Input images are taken from MNIST dataset [55].

as reading a text line aloud¹. Figures 2.1 and 2.2 show the differences between object and sequence classification.

Machine Learning (ML) is a standard approach to model the mapping function f^* . Mitchell [56] defined machine learning with three terms: a "*Task*" that can be learned based on "*Experience*", and the quality of the learning process can be evaluated with a "*Metric*". In this case, input sample \mathbf{x} represents *Experience*, and it is commonly split into two datasets. Each dataset has a different goal. The model is trained only in one dataset, whereas the other dataset is for evaluation purposes. The evaluation measures the quality of the model for generalization of unseen samples. One dataset is only used for training the model, and the other dataset is used for evaluating how good the model is for generalization of unseen samples.

Furthermore, two pipelines are common for learning the mapping function f . The first approach is *feature engineering*, in which the samples are converted to features, i.e., SIFT [57]. The second approach is to use raw input samples (i.e., pixels of images) for learning the embedding of feature representation and the classification at the same time, such as LeNet-5[55]. The rest of this chapter describes two NN architectures that the proposed association framework relies on.

¹It is a common term in OCR community that is defined by the sentences that are obtained from a scanned page

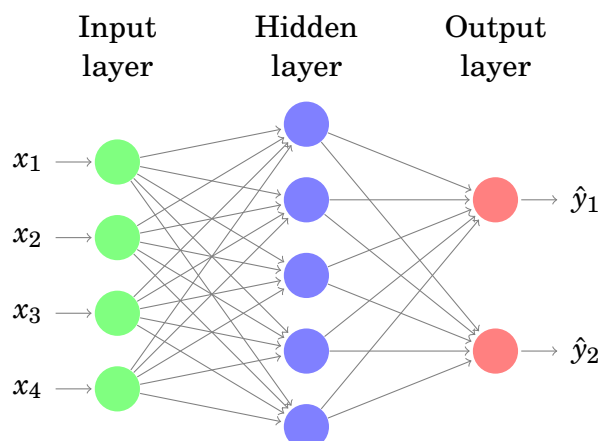


FIGURE 2.3. Example of an MLP. In this case, the network has one hidden layer of five elements. The input and output layers have four and two elements (respectively). Figure generated based on the source code <http://www.texample.net/tikz/examples/neural-network/>.

2.2 Neural Networks (NNs)

The focus of this work lies on NNs because of its biological motivation. Moreover, this approach similar to the human brain is trained based on samples and embeds discriminant information in the connections between neurons. The output layer summarizes information of the categories from input samples, which can be combined with SGP. Additionally, the connections between several modalities (i.e., texts and images) can be also embedded into the connections weights of this approach. Therefore, NNs is the selected approach for combining SGP and IL. This section describes two architectures. *Multilayer Perceptrons* (MLPs) are a common approach for object and text classification. In contrast, *Recurrent Neural Networks* (RNNs) have been applied to sequence learning, e.g., speech recognition.

2.2.1 Multi-layer Perceptrons (MLPs)

Multilayer Perceptrons (MLPs) are composed of layers, which are a group of neurons. Each layer connects to another layer (similar to a composition of functions). MLPs can learn to approximate continuous functions [58, 59]. Thus, an MLP with a hidden layer can be expressed based on Equation (2.2):

$$\hat{\mathbf{y}} = f^1(f^2(\mathbf{x}; \theta^1); \theta^2), \quad (2.4)$$

where function f^2 is the relationship between input and hidden layers and function f^1 is the relationship between hidden and output layers. The function f^2 receives the input sample \mathbf{x} and passes to the hidden layer. Then, the function f^1 receives that information from function f^2 and passes to the output layer. Figure 2.3 shows an example of an MLP that has four input and two output elements. There are mainly three types of layers. First, the input layer (green) receives input samples. Second, the output layer (red) predicts the categories, and there are no connections to other layers. Third, the hidden layers (blue) lay between the input and output layers.

Furthermore, MLPs have two stages: forward and backward steps. The forward step propagates input samples from the input layer to the output layer. The backward step feeds the error measure between the network output and the desired target. In this manner, the parameter networks can be updated based on this error. Formally, an MLP with one hidden layer is defined by the following equations:

$$\mathbf{h} = \sigma(\mathbf{W}_{xh} \cdot \mathbf{x} + \mathbf{b}_{xh}), \quad (2.5)$$

$$\hat{\mathbf{y}} = \sigma(\mathbf{W}_{hy} \cdot \mathbf{h} + \mathbf{b}_{hy}), \quad (2.6)$$

$$\sigma(\mathbf{v}) = \frac{1}{1 + e^{-\mathbf{v}}}, \quad (2.7)$$

where \mathbf{W}_{xh} and \mathbf{W}_{hy} are *weight matrices*, and \mathbf{b}_{xh} and \mathbf{b}_{hy} are *bias vectors*. Note that the matrix \mathbf{W}_{xh} and the bias vector \mathbf{b}_{xh} are the parameter θ^1 in Equation (2.4) (similarly to \mathbf{W}_{hy} , \mathbf{b}_{hy} , and θ^2). In the forward step, the input samples pass through the MLP from the input layer to the output layer (Equations (2.5) and (2.6)). Afterwards, the backward step uses a similar principle to the forward step with the goal of updating the parameters as an optimization problem. Note that the backward step is performed after the forward step. Hence, an error or loss function is required, e.g., *Mean Square Error (MSE)*:

$$J_{MSE} = \frac{1}{N} \sum (\hat{\mathbf{y}} - \mathbf{y})^2, \quad (2.8)$$

where $\mathbf{y} \in R^k$ is the desired target and N is the number of elements in the dataset. Given the loss function, the parameters can be updated based on *gradient descent*. In other words, the parameters changes based on the difference between output vectors (after the *forward step*) and the desired target $\hat{\mathbf{y}}$.

$$\frac{\partial J}{\partial \hat{\mathbf{y}}} = \frac{2}{N} \sum (\hat{\mathbf{y}} - \mathbf{y}), \quad (2.9)$$

Afterwards, the error and the partial derivatives can be propagated for each parameter with the *chain rule*. The derivatives are defined by the following equations

$$\frac{\partial J}{\partial \mathbf{W}_{\mathbf{h}\mathbf{y}}} = \frac{\partial J}{\partial \hat{\mathbf{y}}} \cdot \hat{\mathbf{y}} \cdot (1 - \hat{\mathbf{y}}) \otimes \mathbf{h}, \quad (2.10)$$

$$\frac{\partial J}{\partial \mathbf{b}_{\mathbf{h}\mathbf{y}}} = \frac{\partial J}{\partial \hat{\mathbf{y}}} \cdot \hat{\mathbf{y}} \cdot (1 - \hat{\mathbf{y}}), \quad (2.11)$$

$$\frac{\partial J}{\partial \mathbf{W}_{\mathbf{x}\mathbf{h}}} = \frac{\partial J}{\partial \hat{\mathbf{y}}} \cdot \hat{\mathbf{y}} \cdot (1 - \hat{\mathbf{y}}) \cdot \mathbf{W}_{\mathbf{h}\mathbf{y}} \cdot \mathbf{h} \otimes \mathbf{x}, \quad (2.12)$$

$$\frac{\partial J}{\partial \mathbf{b}_{\mathbf{x}\mathbf{h}}} = \frac{\partial J}{\partial \hat{\mathbf{y}}} \cdot \hat{\mathbf{y}} \cdot (1 - \hat{\mathbf{y}}) \cdot \mathbf{W}_{\mathbf{h}\mathbf{y}} \cdot \mathbf{h}, \quad (2.13)$$

Finally, the method *gradient descent* updates each parameter of the network

$$\mathbf{W}_{\mathbf{h}\mathbf{y}} = \mathbf{W}_{\mathbf{h}\mathbf{y}} - \alpha * \frac{\partial J}{\partial \mathbf{W}_{\mathbf{h}\mathbf{y}}}, \quad (2.14)$$

$$\mathbf{W}_{\mathbf{x}\mathbf{h}} = \mathbf{W}_{\mathbf{x}\mathbf{h}} - \alpha * \frac{\partial J}{\partial \mathbf{W}_{\mathbf{x}\mathbf{h}}}, \quad (2.15)$$

$$\mathbf{b}_{\mathbf{h}\mathbf{y}} = \mathbf{b}_{\mathbf{h}\mathbf{y}} - \alpha * \frac{\partial J}{\partial \mathbf{b}_{\mathbf{h}\mathbf{y}}}, \quad (2.16)$$

$$\mathbf{b}_{\mathbf{x}\mathbf{h}} = \mathbf{b}_{\mathbf{x}\mathbf{h}} - \alpha * \frac{\partial J}{\partial \mathbf{b}_{\mathbf{x}\mathbf{h}}}, \quad (2.17)$$

where α is a learning rate. Note that this example is based on a MLP with one hidden layer. Furthermore, this architecture can have more layers, which new layers are appended to the last layer.

A MLP can predict the category of a sample after training. Therefore, a *winning-take-all rule* is used as a strategy for finding the category given the input sample. Formally, the following equation defines the decision rule, which retrieves the position of the maximum value of the output vector:

$$k^* = \arg \max_k \hat{\mathbf{y}}. \quad (2.18)$$

2.2.2 Recurrent Neural Networks (RNNs)

In the previous section, the description of MLP utilizes isolated elements as input samples. However, sequence learning is also possible. One solution is to use a window with a fixed size that moves over the input sequence. As a result, the network receives the elements inside of the window. This approach requires finding the size of the window that obtains the best performance. The problem of the window size is handled with a different architecture.

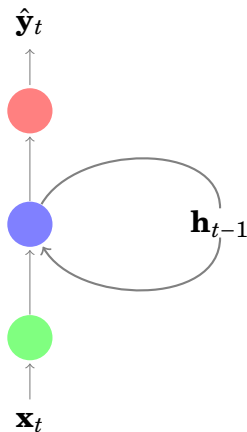


FIGURE 2.4. Example of a Vanilla Recurrent Neural Network. The main feature is a self-loop connection (\mathbf{h}_{t-1}) that encodes or summarizes information of previous states.

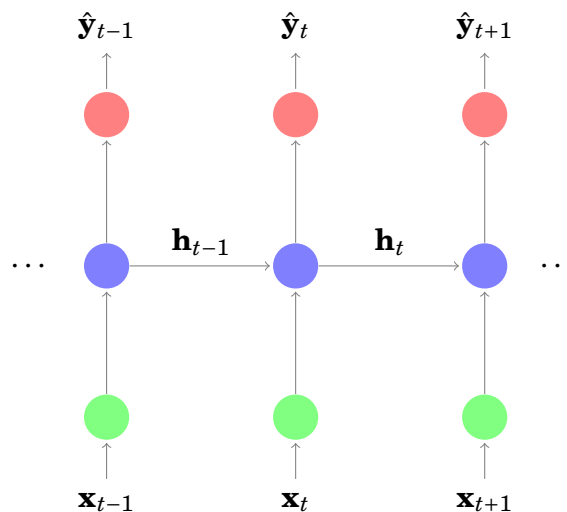


FIGURE 2.5. The flow of information of an RNN is more natural to see if the connections are unfolded over time. For example, the input \mathbf{x}_t exploits the encoded information of \mathbf{h}_{t-1} for predicting $\hat{\mathbf{y}}_t$.

RNNs have been proposed for learning sequences. The main difference between MLPs and RNNs is a self-loop or feedback connection that considers previous states. Figure 2.4 shows an example of an RNN. The recurrent connection has two effects: learning sequences with dynamic sizes and generalization of all input samples.

Similar to MLPs, RNNs have forward and backward steps. The forward step feeds each element of the sequence to the network. Usually, the representation of input vector

is \mathbf{x}_t where $t \in [1, \dots, T]$. The time t represents the length of the sequence (between 1 and T), which is not always related to the time domain. The backward step feeds the error back to the network from the last element T and iterative runs to the first element.

Moreover, a vanilla RNN is defined by the following equations:

$$\mathbf{h}_t = \tanh(\mathbf{W}_{xh} \cdot \mathbf{x}_t + \mathbf{U}_{hh} \cdot \mathbf{h}_{t-1} + \mathbf{b}_h), \quad (2.19)$$

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_{hy} \cdot \mathbf{h}_t + \mathbf{b}_y), \quad (2.20)$$

$$\text{softmax}(\mathbf{v}) = \frac{\mathbf{v}}{\sum_i^N v_i}, \quad (2.21)$$

where the weight matrices $\mathbf{W}_{xh} \in R^{n,h}$ and $\mathbf{W}_{hy} \in R^{h,k}$ are similar to the weight matrices of an MLP. The feedback or recurrent connection is defined by the matrix \mathbf{U}_{hh} . Figure 2.5 shows the information flow between connections in the forward step if RNN is unfolded over time. Note that the initial state of the recurrent connection is $\mathbf{h}_0 = \mathbf{0}$. Also, the output vector in the last position $\hat{\mathbf{y}}_T$ encodes a summary of the sequence. In this case, the training step uses another loss function, called *cross-entropy* because the output layer has a *softmax* function:

$$\mathbf{J}_{cross-entropy}^t = - \sum_{i=1}^N \sum_{t=1}^T \mathbf{y}_t \log \hat{\mathbf{y}}_t, \quad (2.22)$$

Werbos [60] proposed an algorithm called *Backpropagation Through Time (BPTT)* for training RNN. First, the gradient of each parameter is calculated using the chain rule. One difference in comparison to MLP is that the gradients update the RNN parameters for each time step t . Note that the backpropagation is iteratively calculated from the last element to the first element. In summary, the gradients for each parameter are:

$$\frac{\partial \mathbf{J}^t}{\partial \mathbf{W}_{hy}} = \frac{\partial \mathbf{J}^t}{\partial \hat{\mathbf{y}}} \otimes \mathbf{h}_t, \quad (2.23)$$

$$\frac{\partial \mathbf{J}^t}{\partial \mathbf{b}_y} = \frac{\partial \mathbf{J}^t}{\partial \hat{\mathbf{y}}}, \quad (2.24)$$

$$\frac{\partial \mathbf{J}^t}{\partial \mathbf{h}_t} = (1 - \tanh(\mathbf{h}_t)^2) \cdot \left(\mathbf{W}_{hy} \cdot \frac{\partial \mathbf{J}^t}{\partial \hat{\mathbf{y}}} + \frac{\partial \mathbf{J}^{t+1}}{\partial \mathbf{h}_{t+1}} \right), \quad (2.25)$$

$$\frac{\partial \mathbf{J}^t}{\partial \mathbf{b}_h} = \frac{\partial \mathbf{J}^t}{\partial \mathbf{h}_t}, \quad (2.26)$$

$$\frac{\partial \mathbf{J}^t}{\partial \mathbf{W}_{xh}} = \frac{\partial \mathbf{J}^t}{\partial \mathbf{h}_t} \otimes \mathbf{x}_t, \quad (2.27)$$

$$\frac{\partial \mathbf{J}^t}{\partial \mathbf{U}_{hh}} = \frac{\partial \mathbf{J}^t}{\partial \mathbf{h}_t} \otimes \mathbf{h}_{t-1}. \quad (2.28)$$

Second, all gradients are summed up and applied gradient descent method for updating the parameters. For example, the matrix \mathbf{W}_{hy} that represents a parameter of an RNN is updated by the following equations:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}_{hy}} = \sum_{t=1}^T \frac{\partial \mathcal{J}^t}{\partial \mathbf{W}_{hy}}, \quad (2.29)$$

$$\mathbf{W}_{hy} = \mathbf{W}_{hy} - \alpha \frac{\partial \mathcal{J}}{\partial \mathbf{W}_{hy}}. \quad (2.30)$$

So far, vanilla RNNs have used only the information of the previous timestep t . Under some circumstances, the model can also exploit information from future timesteps (both timesteps $t - 1$ and $t + 1$). Consequently, one model can combine two RNNs that employs both contexts. A *forward* RNN runs over the sequence from 1 to T , and a *backward* RNN runs over from T to 1. This architecture is called *Bidirectional RNN* [61], which is formulated as follows:

$$\hat{\mathbf{y}}_t^f = \text{RNN}^f(\mathbf{x}_t, \theta^f), t \in [1, \dots, T], \quad (2.31)$$

$$\hat{\mathbf{y}}_t^b = \text{RNN}^b(\mathbf{x}_t, \theta^b), t \in [T, \dots, 1], \quad (2.32)$$

$$\hat{\mathbf{y}}_t = \text{combination}(\hat{\mathbf{y}}_t^f, \hat{\mathbf{y}}_t^b), \quad (2.33)$$

where the function combination represents the unification between $\hat{\mathbf{y}}_t^f$ and $\hat{\mathbf{y}}_t^b$, such as concatenate both vectors, element-wise addition or multiplication operation. Therefore, the model learns to predict based on all elements in the sequence (e.g., global context). Figure 2.6 shows an example of a bidirectional RNN unfolded over time.

2.2.3 Long Short-Term Memory (LSTM) Networks

Considering an RNN unfolding over time, each timestep can be interpreted as one layer in MLPs. Each layer has a multiplicative effect in the backward step. Therefore, the more number of layers can produce gradient values to be extreme small or big. On the one hand, values closed to zero can produce small changes in the network parameters. On the other hand, large values can present mathematical overflow on the computers. Hence, RNNs have limitations for long sequences because the gradient might be close to zero. This effect is called it the *vanishing gradient problem* [62].

One solution is controlling the information that goes inside and outside of the network. With this in mind, *Long Short-Term Memory (LSTM)* has been proposed [63, 64], where several gates manage the flow of information. In contrast with RNNs, an LSTM has a

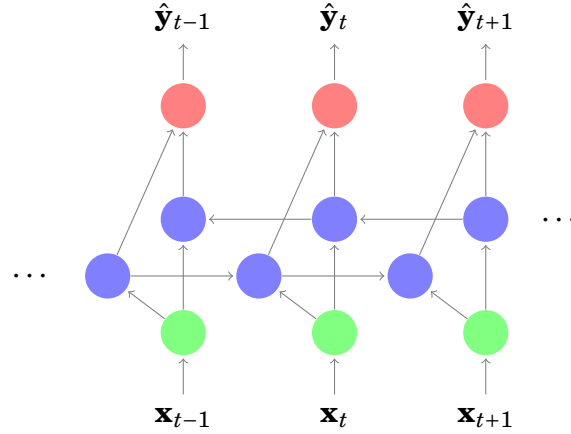


FIGURE 2.6. Bidirectional RNNs merge a *forward* RNN and a *backward* RNN in order to exploit information of all sequence. Thus, the input \mathbf{x}_t exploits both directions $(\mathbf{h}_{t-1}, \mathbf{h}_{t+1})$ for predicting $\hat{\mathbf{y}}_t$.

cell \mathbf{c}_t that stores the state of the network and a self-loop connection that is controlled by the forget gate \mathbf{f}_t . Besides, two more gates (input \mathbf{i}_t and output \mathbf{o}_t gates) are included to control the inflow and outflow information in the LSTM cell. Both gates control how much information enters the cell and passes to the next time step. Figure 2.7 shows an LSTM cell with all its internal connections.

The forward step of LSTM is a set of equations that has three phases. First, the input and the forget gates receive the input sample \mathbf{x}_t and the previous hidden state \mathbf{h}_{t-1} .

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{U}_i \cdot \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (2.34)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{U}_f \cdot \mathbf{h}_{t-1} + \mathbf{b}_f). \quad (2.35)$$

Second, the update of the cell state depends on two factors. The first factor is how much information from the input makes is through the input gate. The second factor is how much knowledge is reset from the previous state of the cell by the forget gate.

$$\tilde{\mathbf{C}} = \tanh(\mathbf{W}_c \cdot \mathbf{x}_t + \mathbf{U}_c \cdot \mathbf{h}_{t-1} + \mathbf{b}_c), \quad (2.36)$$

$$\mathbf{c}_t = \mathbf{f}_t \cdot \mathbf{C}_{t-1} + \mathbf{i}_t \cdot \tilde{\mathbf{C}}. \quad (2.37)$$

Finally, the output gate and the current state of the cell produce the output vector of an LSTM at timestep t :

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{U}_o \cdot \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (2.38)$$

$$\mathbf{h}_t = \mathbf{o}_t \cdot \tanh(\mathbf{c}_t). \quad (2.39)$$

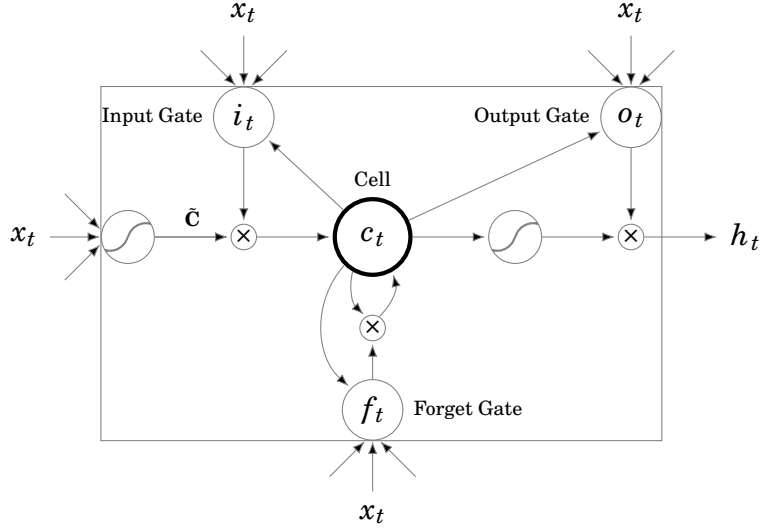


FIGURE 2.7. Example of the connections of LSTM. It can be observed how the gates (input i_t , forget f_t , and output o_t) control the flow of information. Note that each gate is a learnable component that exploits sub-patterns of the sequence. Image generated based on the Tikz source code <https://tex.stackexchange.com/questions/332747/how-to-draw-a-diagram-of-long-short-term-memory>. Additionally, the image is taken from [65].

The backward step is similar to vanilla RNNs. Note that gradients are propagated from the output gate to the cell state and from the cell state to the other gates (input and forget gates). The following equations summarize the backward step of one LSTM cell:

$$\frac{\partial J^t}{\partial \mathbf{o}_t} = \frac{\partial J^t}{\partial \mathbf{h}_t} \cdot \tanh(\mathbf{c}_t) \cdot \mathbf{o}_t \cdot (1 - \mathbf{o}_t), \quad (2.40)$$

$$\frac{\partial J^t}{\partial \mathbf{c}_t} = \frac{\partial J^t}{\partial \mathbf{h}_t} \cdot \mathbf{o}_t \cdot (1 - \tanh(\mathbf{c}_t)^2) + \frac{\partial J^{t+1}}{\partial \mathbf{c}_{t+1}} \cdot \mathbf{f}_{t+1}, \quad (2.41)$$

$$\frac{\partial J^t}{\partial \mathbf{i}_t} = \frac{\partial J^t}{\partial \mathbf{c}_t} \cdot \mathbf{i}_t \cdot (1 - \mathbf{i}_t) \cdot \tilde{\mathbf{C}}, \quad (2.42)$$

$$\frac{\partial J^t}{\partial \mathbf{f}_t} = \frac{\partial J^t}{\partial \mathbf{c}_t} \cdot \mathbf{c}_{t-1} \cdot \mathbf{f}_t \cdot (1 - \mathbf{f}_t), \quad (2.43)$$

$$\frac{\partial J^t}{\partial \tilde{\mathbf{C}}} = \frac{\partial J^t}{\partial \mathbf{c}_t} \cdot \mathbf{i}_t \cdot (1 - \tanh(\tilde{\mathbf{C}})^2). \quad (2.44)$$

In this case, the update step of LSTM parameters is similar to RNN parameters. For

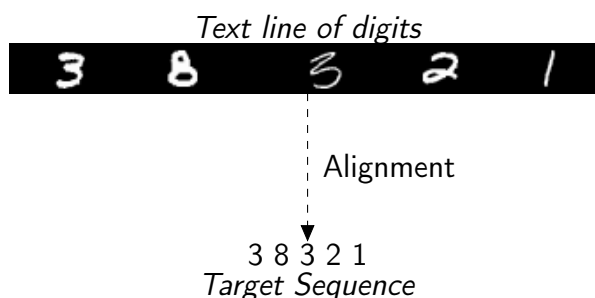


FIGURE 2.8. Example of a weakly labeled sequence. The length of the text line is longer than the length of the target sequence. In this case, the effort annotating this input sequence is less than annotating each column of the sequence. Input images are taken from MNIST dataset [55].

example, the input gate (\mathbf{W}_i and \mathbf{U}_i) are updated with the following equations:

$$\frac{\partial J}{\partial \mathbf{W}_i} = \sum_{t=1}^T \frac{\partial J^t}{\partial \mathbf{i}_t} \otimes \mathbf{x}_t, \quad (2.45)$$

$$\frac{\partial J}{\partial \mathbf{U}_i} = \sum_{t=1}^T \frac{\partial J^t}{\partial \mathbf{i}_t} \otimes \mathbf{h}_{t-1}, \quad (2.46)$$

2.2.4 Connectionist Temporal Classification (CTC)

Annotating each input sample \mathbf{x}_t with its respective target \mathbf{y}_t is a time-consuming task. To simplify the annotation effort, Graves et al. [66] proposed a new output layer called *Connectionist Temporal Classification (CTC)* that aligns LSTM outputs to labels. For explanations purposes, CTC is explained using OCR task as test case. However, CTC can be applied to more scenarios, such as Speech Recognition. OCR task predicts texts (string) given scanned text lines (images). A text line that can be represented by $\mathbf{x}_1, \dots, \mathbf{x}_T$ is only annotated with a set of ordered categories c_1, \dots, c_d where $d < T$. Hence, the targets $\mathbf{y}_1, \dots, \mathbf{y}_T$ are not required as a part of the annotated dataset. This section describes the CTC training. Please refer to the original work for more information [66].

Initially, the sequence labeling task has not only K categories as before but also a new category called *blank* or *non-label* (\mathbf{b}). The motivation for the new category is to constrain the learning algorithm on two transitions: label-to-label and non-label-to-label.

The probability of sequence label is given by

$$p(l|\mathbf{x}) = \sum_{\pi \in B^{-1}(l)} p(\pi|\mathbf{x}), \quad (2.47)$$

$$p(\pi|\mathbf{x}) = \prod_{t=1}^T \hat{y}_t, \quad (2.48)$$

where π is a set of possible paths that can lead to the sequence label l . One significant component is the function $B : K'^T \rightarrow K^{\leq T}$, where K'^T are a sequence with length T , whereas, $K^{\leq T}$ are a sequence shorter or equal than T . In this case, a many-to-one function B eliminates any repeated element or non-label. For example, the following sequences a--bb--cc-a, --a-bc--aa can produce the target sequence label abca. The combination of different paths of the same labeling provides enough information for avoiding the requirement of knowing exactly the position of the error sequence.

Afterwards, the target sequence is prepared for the training algorithm. The blank category is added to the original sequence label: between each pair of labels, at the beginning and at the end of the sequence. For instance, the sequence "1 2 3" is converted to "␣1␣2␣3␣". Equation (2.47) can be solved similarly to *Hidden Markov Model* (HMM) [67], which is trained with a dynamic programming approach for propagating the forward and backward probabilities of the sequence. Finally, a decoding step is applied for labeling the input sequence based on the LSTM output vectors. One standard approach is to retrieve the maximum element per timestep. Later, the repeated categories between the blank class are removed from the target sequence. Also, the blank class is removed. Figure 2.9 shows an example of sequence classification based on decoding step.

In summary, CTC training extends the pipeline of LSTM that was described in this section. The new module is presented between the forward and backward step. The goal is to obtain target vectors for each time step, which are not provided as part of the annotated dataset. The new pipeline with CTC training is described as follows

1. The input sequence \mathbf{x}_t is fed to LSTM
2. The forward-backward algorithm \mathbf{CTC}_t explained in this section is applied given the current output probabilities $\hat{\mathbf{y}}_t$
3. The error is calculated between the output $\hat{\mathbf{y}}_t$ and \mathbf{CTC}_t
4. LSTM parameters are updated given the error

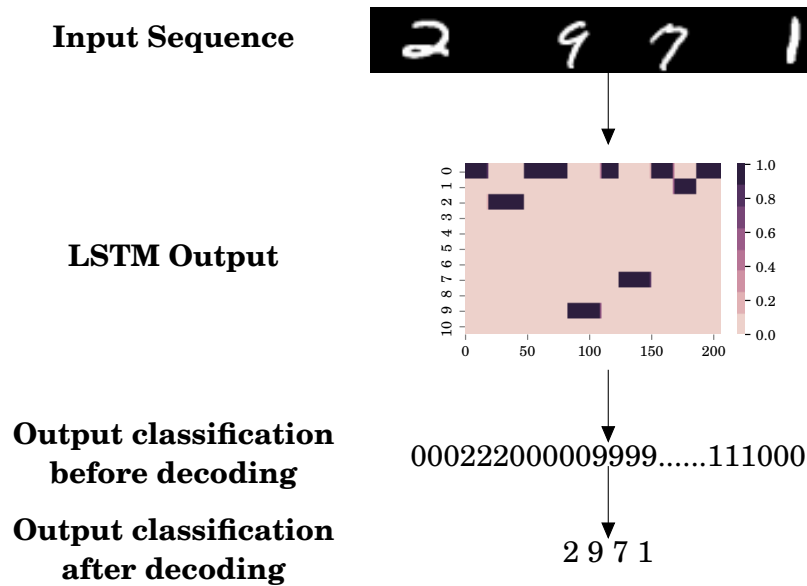


FIGURE 2.9. Example of the LSTM classification based on CTC training. The *blank class* is represented by the index zero. The decoding step converts the output vectors to sequence of categories. Input images are taken from MNIST dataset [55].

2.3 Summary

This chapter describes a definition of ML in terms of three components: Task, Experience, and Metric. Additionally, two different architectures of NNs have been described. The first architecture is an MLP, which can be defined by function compositions. This architecture has been successfully applied to object and text classification. The other architecture is LSTM, which is a RNN with gates. This architecture has been applied to sequence learning, such as speech recognition.

The next chapter presents a new framework for the association task that is inspired by SGP and infant learning. The motivation of using NNs is based on the discriminant information embedded in the weight between neurons. Therefore, the output layer can be interpreted as numerical symbolic feature. Additionally, the combination of several modalities can be also handled based on embedding in NNs.

ASSOCIATION LEARNING FRAMEWORK

This chapter presents a new association model that learns the link between two input samples of the same category. Two scenarios are considered in this work. The first scenario is constrained to pairs of elements that represent the same category. The second scenario is extended to pairs of sequences. Each sequence represents the same series of categories. Note that the input representations can be mono- and multi-modal.

The models explain in this chapter appeared in ICANN2016 [68], ICDAR2015 [69], and CoCo2016 [70]. Section 3.1 describes the association learning in terms of a machine learning task. Furthermore, association learning is defined by learning that two elements represent the same category. Section 3.2 explains a main framework for association learning with two parallel NNs. Section 3.3 presents one version of the association framework based on two parallel MLPs. This model learns the association in the case that two input samples represent the same semantic concept. Section 3.4 explains another version of association framework based on two parallel LSTMs. This model type can learn the association between two weakly labeled sequences.

3.1 Problem Definition

The previous chapter formally described the supervised classification task, in which the objective is to find a mapping function between input samples and their respective

category. The association task of two input samples can be expressed similar to the supervised scenario presented in Chapter 2:

$$\hat{\mathbf{y}} = f^*(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}; \theta), \quad (3.1)$$

where the goal is to classify two input samples $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ with the same category $\hat{\mathbf{y}}$. The input type of both samples is not necessarily the same. For example, the association learning can be between images and their textual descriptions.

This section introduces a novel constraint inspired by SGP, which proposes to learn the association without predefined scheme before training. With this in mind, two *definitions* are explained for understanding the new constraint in the association scenario.

Semantic Concepts (SeC) are the possible categories into which input samples can be classified, e.g., cat, dog, airplane. This is represented by the set of categories $K = \{1, \dots, k\}$.

Vectorial Representations (VR) are the raw numerical output vectors that neural networks internally represent semantic concepts, which is usually based on a one-hot scheme (*c.f.* Section 2.1). This is represented by the set of unit vectors $VR = \{\mathbf{e}_1, \dots, \mathbf{e}_k\}$.

This chapter uses the term *traditional approach* to refer to the relationship between semantic concepts and vectorial representations before training. This thesis proposes a new condition that learns the assignment between semantic concepts and vectorial representations during training. The presented constraint is supported by two factors. The first factor is the output vectors of NNs, which can be considered as *numerical symbolic features* without predefined categorical information, i.e. do not directly refer to categories. The second factor consists on learning an agreement or matching between two different input samples and their respective concept simultaneously. This phenomenon resembles the way infants learn new concepts (*c.f.* Section 1.2). Note that the constraint add a new dimension for learning.

More formally, the relation between the two new definitions can be expressed as a matrix $\mathbf{E} \in R^{k \times k}$ where each row and each column represent a semantic concept and a

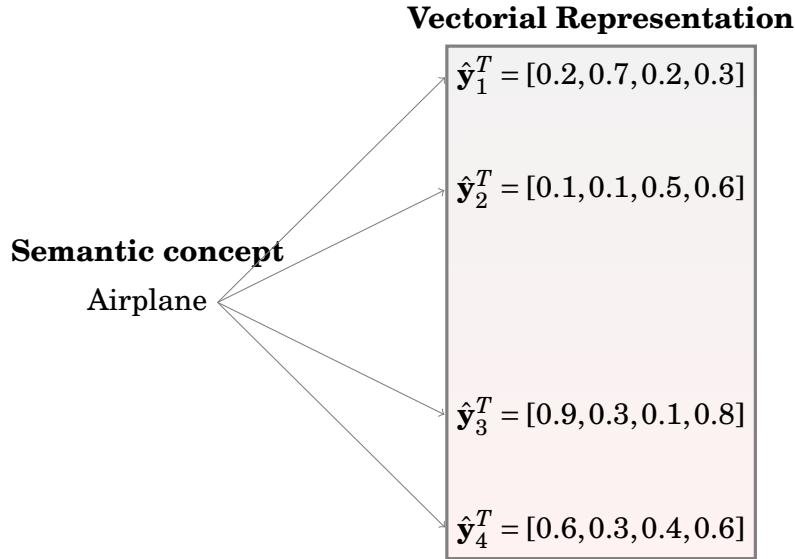


FIGURE 3.1. Example of semantic concepts and vectorial representations. Usually, the *traditional approach* requires to define the representation of the category *airplane* based on the vectorial representations $\hat{\mathbf{y}}_1$, $\hat{\mathbf{y}}_2$, $\hat{\mathbf{y}}_3$, and $\hat{\mathbf{y}}_4$. The selection between one of the four options is made before training model.

vectorial representation, respectively. For example, that relationship in a scenario of four categories ($k = 4$) can be defined by one of the following two matrices¹ $\mathbf{E}^{(1)}$ and $\mathbf{E}^{(2)}$:

$$\mathbf{E}^{(1)} = \begin{bmatrix} | & | & | & | \\ \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_4 \\ | & | & | & | \end{bmatrix} \text{ or } \mathbf{E}^{(2)} = \begin{bmatrix} | & | & | & | \\ \mathbf{e}_3 & \mathbf{e}_4 & \mathbf{e}_1 & \mathbf{e}_2 \\ | & | & | & | \end{bmatrix}, \quad (3.2)$$

The decision of choosing $\mathbf{E}^{(1)}$ over $\mathbf{E}^{(2)}$ or vice versa occurs externally to the model before training. A person decides each representation for each category. This work proposes that the training algorithm includes the relation represented by \mathbf{E} as a learning parameter. Therefore, Equation (2.2) is updated as follows:

$$\hat{\mathbf{y}} = f^*(\mathbf{x}; \theta, g(K, E)), \quad (3.3)$$

where $g(K, E)$ is a learnable parameter of the relation between the set of potential categories k and the set of possible unit vectors \mathbf{e} .

¹Note that there are $k!$ possible combinations of matrix \mathbf{E} for this example

With this in mind, the association learning can be expressed as follows. Given two sets of input elements $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ and a set semantic concepts \mathbf{K} where an element in $\mathbf{X}^{(1)}$ is linked to one element in $\mathbf{X}^{(2)}$ with the same semantic concept. The goal of the association learning is

$$f^1(\mathbf{X}^{(1)}; \theta^{(1)}, g^{(1)}(K, \mathbf{E}^{(1)})) = f^2(\mathbf{X}^{(2)}; \theta^{(2)}, g^{(2)}(K, \mathbf{E}^{(2)})), \quad (3.4)$$

$$\mathbf{E}^{(1)} \equiv \mathbf{E}^{(2)}. \quad (3.5)$$

This constraint requires that the model self-learns the matching to the same k for each $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ pair during training. Similar behavior occurs in infants while they are learning this kind of agreement between categories and sensory information. Infants are learning the agreement between modalities where the same semantic concept may have several formats, e.g. visual and audio. In this case, agreement is referred to that children can classify with the same category information collected by different sensory input signals.

3.2 General Framework for Symbolic Association Learning

Association learning requires two input sets with the same semantic concepts. The proposed model has two parallel NNs, in which each NN learns to classify one input. This architecture is flexible regarding sample formats, which are not required to be the same.

Siamese Networks proposed by Chopra et al. [71] uses two parallel NNs but for applies face verification. The task is defined by two face samples, which may correspond to the same or different person (binary classification). One requirement of their model is to have two labels for each pair of faces: same and difference. Instead, the presented model here can learn multiple categories and match different types of input pairs, such as (image, text) or (image, audio). Hence, the motivation behind it is to be more flexible by learning the function f independently. Both models are associated to update the model parameters $(\theta^{(1)}, \theta^{(2)}, \mathbf{E}^{(1)}, \text{ and } \mathbf{E}^{(2)})$

The training algorithm follows an *Expectation-Maximization* (EM) approach [72],

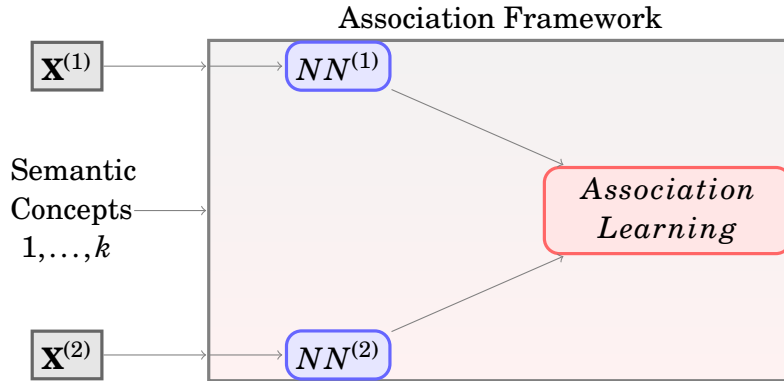


FIGURE 3.2. Overview of the association learning framework. Two parallel NNs learn the agreement between two input samples. In other words, the prediction of both input samples must be the same.

which consists of two steps. The first step (called *Expectation-step*) predicts the output of the model given the current parameters. Then, the second step (called *Maximization-step*) updates the parameters giving the current output of the model. These two steps are iteratively applied between each other.

Concretely, the *E-step* propagates each input sample through the respective NN. The association learning module uses the *raw* output vector to extract matrices $\mathbf{E}^{(1)}$ and $\mathbf{E}^{(2)}$. Afterwards, the *M-step* propagates the error from $NN^{(1)}$ back to $NN^{(2)}$, and vice versa. This error propagation approach leads to both networks agreeing on the same semantic concept. Figure 3.2 shows the general association learning framework. In this work, two scenarios have been considered: two input samples associated with 1) one semantic concept and 2) multiple semantic concepts.

3.3 Symbolic MLP-based Approach

In this section, the general association framework is implemented using two parallel MLPs. The goal is to associate two samples that represent the same semantic concept, which each sample pair corresponds to only one category.

The presented approach describes a solution for association of isolated elements, i.e.

an image represents only a digit. Formally, the input set can be defined as

$$\mathbf{X} = \mathbf{X}^{(1)} \cup \mathbf{X}^{(2)} \cup K', \quad (3.6)$$

$$K' = K - \mathbf{E}, \quad (3.7)$$

where $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are two disjoint sets that represent the same semantic concepts K without their vectorial representation \mathbf{E} . The approach has two parallel MLPs that are defined by

$$\hat{\mathbf{y}}^{(1)} = MLP^{(1)}(\mathbf{x}^{(1)}, \theta^{(1)}), \quad (3.8)$$

$$\hat{\mathbf{y}}^{(2)} = MLP^{(2)}(\mathbf{x}^{(2)}, \theta^{(2)}). \quad (3.9)$$

The training procedure follows the EM-algorithm. The key elements of this procedure is to learn the association between SeC and VR based on two *weighting vectors* $\gamma_j^{(1)}, \gamma_j^{(2)} \in R^k$ where $j = 1, \dots, k$. The intuition behind the *weighting vectors* is to attach semantic concepts to the raw output vector. Also, their role is to modify the output distribution of the output vectors. Figure 3.3 shows an example of the weighting vector in one dimension. Consider an output vector with values similar to the original state (top histogram). In that case, the maximum element is at position three. However, the weighting vectors γ_1 and γ_2 can modify the distribution. For instance, *Option 1* shows that the maximum element is at position one. In contrast, the maximum value is at position four in the *Option 2*. In summary, the weighting vectors have the role of changing the distribution between all semantic concepts with the constraint that each semantic concept has a different representation.

The E-step predicts vectorial representation for each semantic concept k . First, the network receives a mini-batch of m input samples.

$$\mathbf{z}_j^{(1)} = \frac{1}{m} \sum_{i=1}^m \text{power}(\hat{\mathbf{y}}_i^{(1)}, \gamma_j^{(1)}) \quad \text{where } j = 1, \dots, k, \quad (3.10)$$

$$\mathbf{z}_j^{(2)} = \frac{1}{m} \sum_{i=1}^m \text{power}(\hat{\mathbf{y}}_i^{(2)}, \gamma_j^{(2)}) \quad \text{where } j = 1, \dots, k. \quad (3.11)$$

Afterwards, the training algorithm assembles the matrices $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ by concatenating all vectors \mathbf{z}_j where $j=1, \dots, k$. The goal of this process is to match the relationship between semantic concepts and vectorial representations. The matrices ($\mathbf{Z}^{(*)}$) are transformed to an assembly of one-hot vectors ($\mathbf{E}^{(*)}$). One way to find this relationship is to take indices of the maximum value and then setting to zero the row and column of that position

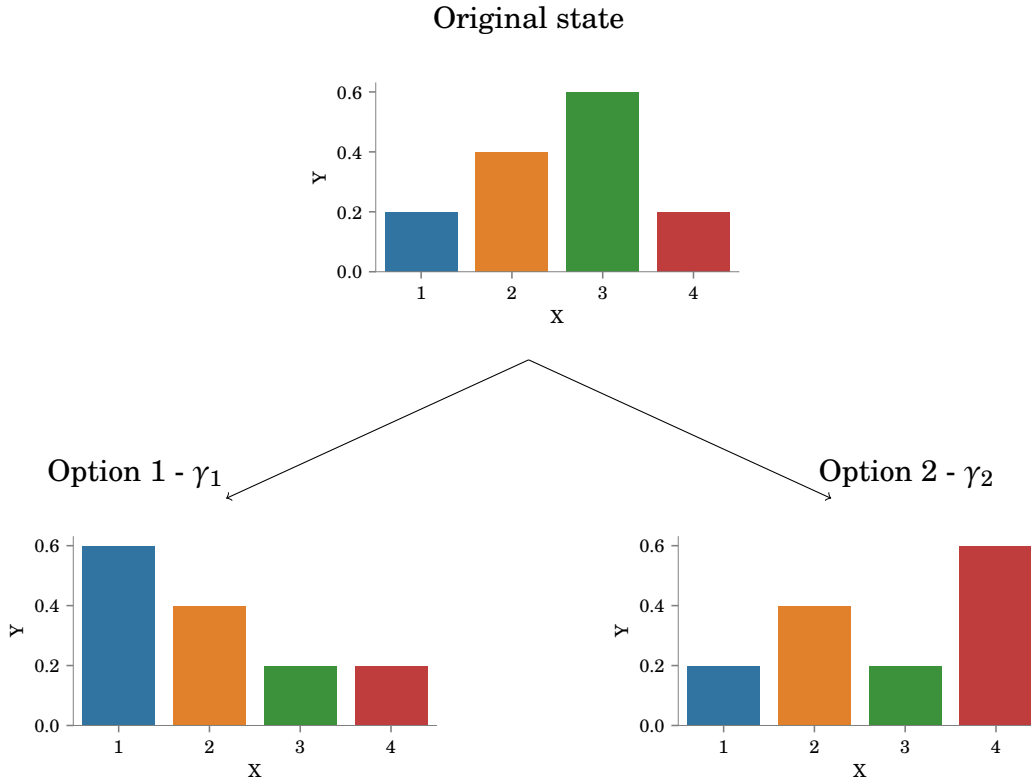


FIGURE 3.3. Example of the weighting vector role. Note that the maximum element of the original state is three. However, the weighting vectors can modify this arrangement. In this example, Option 1 and 2 show different distributions where the maximum values change.

(implemented by `max_operation` function). This process is repeated k times. Figure 3.4 shows an example of this transformation when the number of categories k is equal to four. Note that the matrix \mathbf{E} has $k!$ possible combinations.

$$\mathbf{Z}^{(1)} = \begin{bmatrix} \left| \right. & \left| \right. & \dots & \left| \right. & \left| \right. \\ \mathbf{z}_1^{(1)} & \mathbf{z}_2^{(1)} & \dots & \mathbf{z}_{k-1}^{(1)} & \mathbf{z}_k^{(1)} \\ \left| \right. & \left| \right. & \dots & \left| \right. & \left| \right. \end{bmatrix}, \quad (3.12)$$

$$\mathbf{Z}^{(2)} = \begin{bmatrix} \left| \right. & \left| \right. & \dots & \left| \right. & \left| \right. \\ \mathbf{z}_1^{(2)} & \mathbf{z}_2^{(2)} & \dots & \mathbf{z}_{k-1}^{(2)} & \mathbf{z}_k^{(2)} \\ \left| \right. & \left| \right. & \dots & \left| \right. & \left| \right. \end{bmatrix}, \quad (3.13)$$

$$\mathbf{E}^{(1)} = \text{max_operation}(\mathbf{Z}^{(1)}), \quad (3.14)$$

$$\mathbf{E}^{(2)} = \text{max_operation}(\mathbf{Z}^{(2)}). \quad (3.15)$$

$$\mathbf{Z}^{(1)} = \begin{bmatrix} 0.3 & 0.7 & 0.8 & 0.1 \\ 0.4 & 0.2 & 0.9 & 0.4 \\ 0.2 & 0.9 & 0.3 & 0.3 \\ 0.7 & 0.7 & 0.2 & 0.2 \end{bmatrix} \longrightarrow \mathbf{E}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

FIGURE 3.4. Example of the elimination process. Each column of $\mathbf{Z}^{(1)}$ is a vector that summarizes all input samples of category k . Afterwards, a max operation applied to the matrix four times to obtain a simplified version that is represented by $\mathbf{E}^{(1)}$. The current relation between semantic concepts (columns) and vectorial representation (rows) is shown in the following relation $(\mathbf{K}, \mathbf{E}) = \{(1,4), (2,3), (3,2), (4,1)\}$.

The M-step has two purposes. One is to update the network parameters ($\theta^{(1)}$ and $\theta^{(2)}$). The other purpose is to update the *weighting vectors* ($\gamma^{(1)}$ and $\gamma^{(2)}$) with gradient descent. Besides, the network parameters are updated as the standard backpropagation algorithm seen in Chapter 2. The main difference compared to the standard backpropagation algorithm is the error calculation. The loss function of $MLP^{(1)}$ is MSE between output vector $\hat{\mathbf{y}}^{(1)}$ and the unit vector $\mathbf{e}^{(2)}$, which are obtained after the *E-step*. Note that $\mathbf{e}^{(2)}$ is the vectorial representation of semantic concept k in $MLP^{(2)}$, which might be or might not be the same in $MLP^{(1)}$ in the first iterations of the training step. Furthermore, the training step force both networks to learn the agreement based on this coupling, i.e. $\mathbf{e}^{(1)}$ are $\mathbf{e}^{(2)}$ the same. The loss function for each MLP is defined as follows:

$$\mathcal{J}_{MLP}^{(1)} = \frac{1}{N} \sum_{i=1}^N (\hat{\mathbf{y}}_i^{(1)} - \mathbf{e}_k^{(2)})^2, \quad (3.16)$$

$$\mathcal{J}_{MLP}^{(2)} = \frac{1}{N} \sum_{i=1}^N (\hat{\mathbf{y}}_i^{(2)} - \mathbf{e}_k^{(1)})^2, \quad (3.17)$$

where $\mathbf{e}_k^{(*)}$ is the vectorial representation of the desired semantic concept of the element $\mathbf{x}_i^{(*)}$. In contrast, the loss function of the *weighting concepts* uses a statistical distribution ϕ as a target. For example, the loss function can be computed with respect to a uniform distribution. Therefore, ϕ can be expressed a uniform distribution per semantic concept

based on the vectorial representations. The loss function is defined as follows

$$\mathcal{J}_{\gamma_j}^{(1)} = \left(\mathbf{z}_j^{(1)} - \frac{1}{k} \mathbf{e}_j^{(1)} \right)^2, \quad (3.18)$$

$$\gamma_j^{(1)} = \gamma_j^{(1)} - \alpha * \frac{\partial \mathcal{J}_{\gamma_j}^{(1)}}{\partial \gamma_j^{(1)}}, \quad (3.19)$$

$$\mathcal{J}_{\gamma_j}^{(2)} = \left(\mathbf{z}_j^{(2)} - \frac{1}{k} \mathbf{e}_j^{(2)} \right)^2, \quad (3.20)$$

$$\gamma_j^{(2)} = \gamma_j^{(2)} - \alpha * \frac{\partial \mathcal{J}_{\gamma_j}^{(2)}}{\partial \gamma_j^{(2)}}. \quad (3.21)$$

The prediction step is similar to the one introduced in Chapter 2, but is combined with the *weighing vectors* for mapping from the vectorial representation to the abstract concept. As a reminder, note that the output vectors of this model do not have categorical information. The main steps are as follows. First, the maximum element of the vectorial representation is retrieved (Equations (3.22) and (3.24)). Second, the semantic concept is retrieved based on the vectorial representation and the weighing vectors for each semantic concepts (Equations (3.23) and (3.25)).

$$vr^{(1)} = \arg \max_{vr} \hat{\mathbf{y}}^{(1)}, \quad (3.22)$$

$$k^{(1)*} = \arg \max_k \text{power} \left(\hat{\mathbf{y}}_{vr^{(1)}}^{(1)}, \gamma_{j,k^{(1)}}^{(1)} \right), \quad (3.23)$$

$$vr^{(2)} = \arg \max_{vr} \hat{\mathbf{y}}^{(2)}, \quad (3.24)$$

$$k^{(2)*} = \arg \max_k \text{power} \left(\hat{\mathbf{y}}_{vr^{(2)}}^{(2)}, \gamma_{j,k^{(2)}}^{(2)} \right). \quad (3.25)$$

Figure 3.5 shows a general overview of the association framework with two parallel MLPs. Several components has the MLP-based approach. The first component is the cross-learning between both networks, which force the agreement. This cross-learning is implemented using the vectorial representation of $MLP^{(1)}$ as targets of $MLP^{(2)}$, and vice versa. The second component is to use a statistical distribution for learning the relationship between semantic concepts and vectorial representations. The *weighing vectors* modify the distribution of the output vectors for both learning the relationship and predicting semantic concepts. The next section extends the model to input samples with multiple categories.

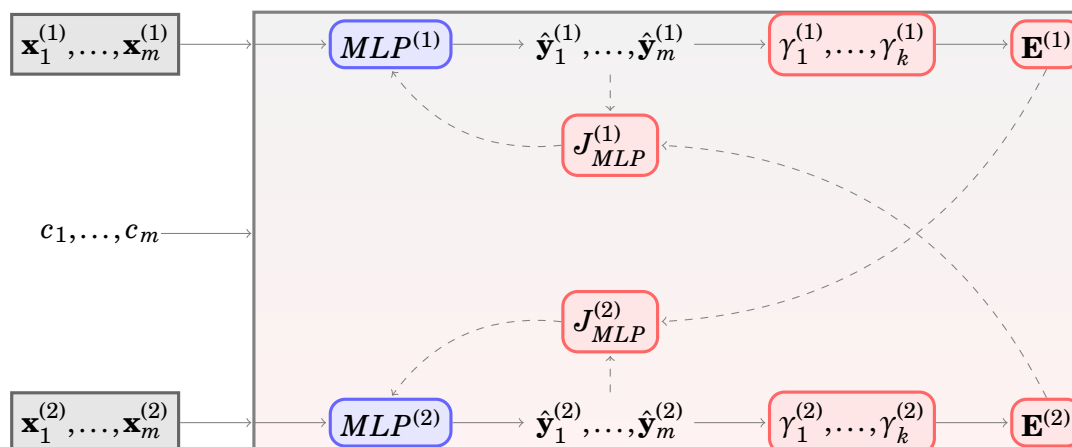


FIGURE 3.5. General overview of the MLP-based approach. The *E-step* (solid line) passes the input samples through each MLP, and obtains each matrix \mathbf{E} . The *M-step* (dashed line) propagates the error back to each network. Note that the matrix $\mathbf{E}^{(1)}$ is used for updating $MLP^{(2)}$, and $\mathbf{E}^{(2)}$ is used for updating $MLP^{(1)}$.

3.4 Symbolic LSTM-based Approach

The previous section has defined a model that can learn the association between two MLPs where the training algorithm includes the relation between semantic concepts and vectorial representations as a new learning component. One step further is proposed in this section, in which the association model learns the relationship in a scenario similar to reading text lines aloud. This test case can be similar to infants gathering samples of maternal speech, when parents want to teach new words via reading aloud story books [24].

One way to represent this scenario might be to convert each word and element into input vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ (respectively). This situation can occur if the training algorithm does not require segmenting input samples beforehand. Moreover, the segmentation process means annotating the position of each word in the input sequence. Sometimes, the words are represented by more than one vector. Another option is to use text lines with their respective audio. This work uses the latter scenario because it has several benefits, which have already mentioned in Section 2.2.4, especially regarding a less annotated dataset.

More formally, sequences $\mathbf{X}^{(1)} = \{\mathbf{x}_1^{(1)}, \dots, \mathbf{x}_{t_1}^{(1)}\}$ and $\mathbf{X}^{(2)} = \{\mathbf{x}_1^{(2)}, \dots, \mathbf{x}_{t_2}^{(2)}\}$ represent the same ordered set of semantic concepts $SeC = \{c_1, \dots, c_s\}$. Like the previous approach with MLPs, the two parallel NNs learn to share their latent space using an *EM-approach*. The main network architecture is LSTMs since this architecture is a standard approach for sequence learning.

Initially, there are two LSTM networks for each sequence that represent the same chain of semantic concepts.

$$\hat{\mathbf{y}}_j^{(1)} = \text{LSTM}^{(1)}(\mathbf{x}_j^{(1)}, \theta^{(1)}) \quad j = 1, \dots, t_1, \quad (3.26)$$

$$\hat{\mathbf{y}}_j^{(2)} = \text{LSTM}^{(2)}(\mathbf{x}_j^{(2)}, \theta^{(2)}) \quad j = 1, \dots, t_2, \quad (3.27)$$

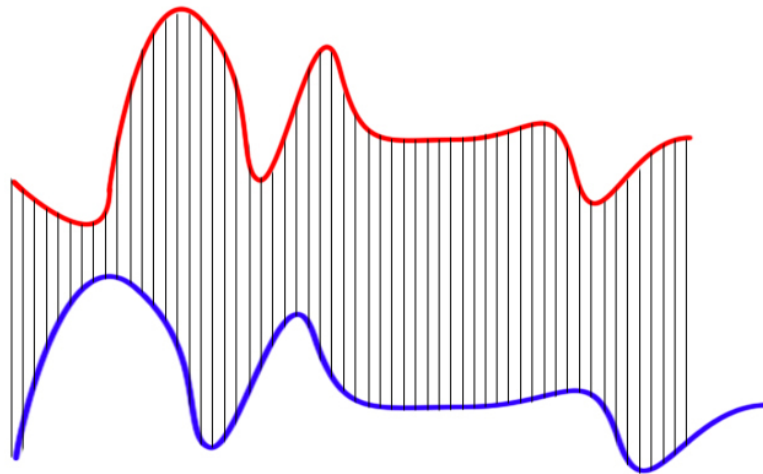
where t_1 and t_2 are the lengths of each sequence. The LSTM-based approach is trained online (or the mini-batch size is one) for each sequence, whereas the MLP approach learns based on mini-batches.

In general, this version of the association framework follows a similar EM-approach for training. The main difference is the application to sequences instead of pairs of samples with one semantic concept. The *E-Step* predicts the relationship between semantic concepts and vectorial representation and the output sequences. Equations (3.10) and (3.11) are updated for manipulating input sequences as follows:

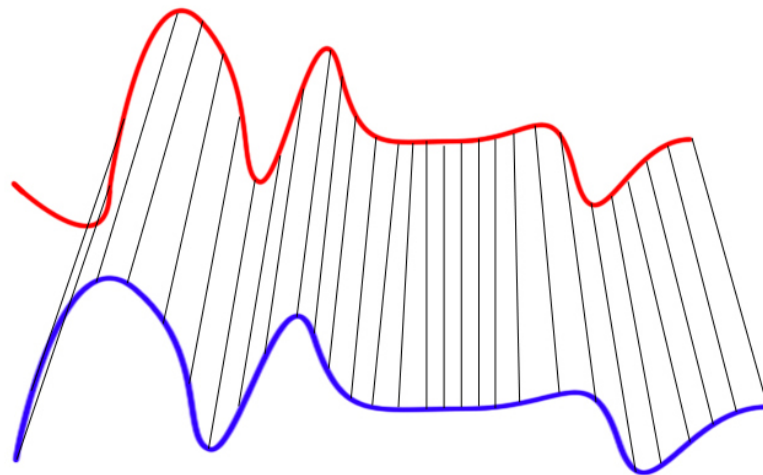
$$\mathbf{z}_j^{(1)} = \frac{1}{t_1} \sum_{t=1}^{t_1} \text{power}(\hat{\mathbf{y}}_t^{(1)}, \gamma_j^{(1)}), \quad (3.28)$$

$$\mathbf{z}_j^{(2)} = \frac{1}{t_2} \sum_{t=1}^{t_2} \text{power}(\hat{\mathbf{y}}_t^{(2)}, \gamma_j^{(2)}) \quad j = 1, \dots, k. \quad (3.29)$$

In this case, the training algorithm summarizes the sequence information to find the mapping $g : \mathbf{Z} \rightarrow \mathbf{E}$. Similar to previous approach with MLP, each vector $\hat{\mathbf{z}}_j^{(1)}$ ($j = 1, \dots, k$) is concatenated to obtain the matrix $\mathbf{Z}^{(1)}$ (the same procedure is applied to $\hat{\mathbf{z}}_j^{(2)}$). After finding the vectorial representation, the CTC step (*c.f.* Section 2.2.4) produces the alignment of each LSTM with the current output sequences $\hat{\mathbf{Y}}^{(1)}$ and $\hat{\mathbf{Y}}^{(2)}$. Note that the approach using MLPs learns the agreement between two input vectors. In contrast, the agreement here is represented by two set of vectors of different lengths. One way to use the information of one sequence as the target of the other is to align them over the time domain. A standard alignment approach is calculated the *Euclidean* distance between each pair of vectors and selecting the pairs with minimum distance. Berndt and Clifford [74] proposed another alignment method that reaches better performance, which is called *Dynamic*



Euclidean Alignment



DTW Alignment

FIGURE 3.6. Example of Euclidean and DTW alignments. Euclidean alignment focuses only on local context because of the minimum distance of each pair. In contrast, DTW alignment exploits the global context of both signals in terms of common sub-paths instead of independent points. Images are retrieved from [73].

Algorithm 1 Pseudocode of the Alignment between two sequences based on DTW.

Require: Two sequences $\mathbf{a}^{(1)} = \{\mathbf{a}_1^{(1)}, \dots, \mathbf{a}_{t_1}^{(1)}\}$, $\mathbf{a}^{(2)} = \{\mathbf{a}_1^{(2)}, \dots, \mathbf{a}_{t_2}^{(2)}\}$

```

{Initialize matrix DTW}
for i=1 TO  $t_1$  do
   $DTW[i,0] \leftarrow \textit{infinity}$ 
end for

for i=1 TO  $t_2$  do
   $DTW[0,i] \leftarrow \textit{infinity}$ 
end for

for i=1 TO  $t_1$  do
  for j=1 TO  $t_2$  do
     $d \leftarrow \textit{Euclidean\_distance}(\mathbf{a}_i^{(1)}, \mathbf{a}_j^{(2)})$ 
     $DTW[i,j] \leftarrow d + \textit{minimum} \begin{cases} DTW[i-1,j-1] \\ DTW[i-1,j] \\ DTW[i,j-1] \end{cases}$ 
  end for
end for

```

Time Warping (DTW). Their method combines two signals that minimizes the cost of all pair elements. Figure 3.6 shows the difference between Euclidean and DTW alignment.

Dynamic programming solves the DTW alignment problem. Algorithm 1 describes the general algorithm for aligning two signals. This approach performs the DTW alignment twice: one from $LSTM^{(1)}$ to $LSTM^{(2)}$ and the other from $LSTM^{(2)}$ to $LSTM^{(1)}$. Therefore, cost matrices for both $DTW^{(1 \rightarrow 2)}$ and $DTW^{(2 \rightarrow 1)}$ are calculated for sequence alignment, i.e., from sequence 1 to sequence 2. These matrices are defined by the following relationship:

$$DTW^{(1 \rightarrow 2)} = dist(\mathbf{CTC}_i^{(1)}, \mathbf{CTC}_j^{(2)}) + \min \begin{cases} DTW^{(1 \rightarrow 2)}[i-1, j-1], \\ DTW^{(1 \rightarrow 2)}[i-1, j], \\ DTW^{(1 \rightarrow 2)}[i, j-1], \end{cases} \quad (3.30)$$

$$DTW^{(2 \rightarrow 1)} = dist(\mathbf{CTC}_i^{(1)}, \mathbf{CTC}_j^{(2)}) + \min \begin{cases} DTW^{(2 \rightarrow 1)}[i-1, j-1], \\ DTW^{(2 \rightarrow 1)}[i-1, j], \\ DTW^{(2 \rightarrow 1)}[i, j-1]. \end{cases} \quad (3.31)$$

where $DTW^{(1 \rightarrow 2)} \in R^{t_1, t_2}$ is the cost matrix obtained after applying Algorithm 1, $dist[i, j]$ is the *Euclidian distance* between $\mathbf{CTC}_i^{(1)}$ and $\mathbf{CTC}_j^{(2)}$ where $i \in t_1$ and $j \in t_2$. Then, a path mapping $p^{(1)}: \hat{\mathbf{y}}_i^{(1)} \rightarrow \hat{\mathbf{y}}_j^{(2)}$ where $i = 1, \dots, t_1$ and $j = 1, \dots, t_2$ links each vector $\hat{\mathbf{y}}^{(1)}$ to one vector $\hat{\mathbf{y}}^{(2)}$. In other words, each vector $\hat{\mathbf{y}}^{(1)}$ is linked to one vector $\hat{\mathbf{y}}^{(2)}$.

The *M-step* updates the LSTM parameters $\theta^{(1)}$ and $\theta^{(2)}$ and the *weighting vectors* $\gamma^{(1)}$ and $\gamma^{(2)}$. The weighting vectors are updated similarly to Equations (3.18) and (3.19). To train both LSTM networks, one LSTM network learns their parameters using the output of the other LSTM network as a target. Each LSTM employs this step. This cross-training is plausible because of the alignment path obtained by DTW. As a result, the loss function of each LSTM is

$$J_{LSTM}^{(1)} = \hat{\mathbf{y}}_{j_1}^{(1)} - \mathbf{CTC}_{j_1}^{(2)} \quad j_1 = 1, \dots, t_1, \quad (3.32)$$

$$J_{LSTM}^{(2)} = \hat{\mathbf{y}}_{j_2}^{(2)} - \mathbf{CTC}_{j_2}^{(1)} \quad j_2 = 1, \dots, t_2. \quad (3.33)$$

Retrieving semantic concepts from output sequences follow the same approach described in the previous section. Note that Equations (3.22) to (3.25) are applied to each timestep.

3.5 Summary

This chapter describes the association problem in the context of SGP and infant learning. The motivation behind this approach is that infants have a limited knowledge about their environment. Therefore, the association between different sensory input signals of the same category is not predefined. Additionally, the agreement of the same abstract concept between several sensory input signals is slowly learned by infants.

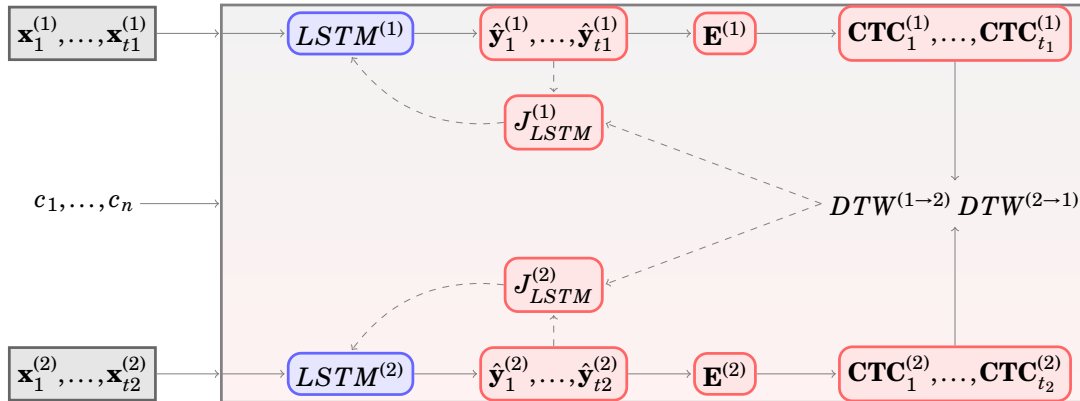


FIGURE 3.7. General overview of the Association Learning based on LSTM.

Similar to the MLP version, the training process follows an EM approach. The *E-step* (solid line) passes each input sequence to their respective LSTM network. Also, the relation between semantic concepts and vectorial representations are obtained. The *M-step* (dashed lines) updates the LSTM parameters based on the alignment latent space. In this case, $LSTM^{(1)}$ is trained based on the latent space generated by $LSTM^{(2)}$.

The presented association framework learns to match two different input samples that represents the same category. Two scenarios are considered: sample pairs of one category and multiple categories. The first scenario case matches two independent elements of the same abstract concept. The proposed solution uses two parallel MLPs. The second scenario matches two sequences that are weakly annotated. The proposed solution is based on two parallel LSTMs. Chapters 4 and 5 show the evaluation of these methods. Each chapter explains the results in mono- and multi-modal scenarios.

ASSOCIATION LEARNING FOR INPUT PAIRS

This chapter describes the results of the following test case: a semantic concept is represented by two samples. The evaluation covers two conditions related to the input format. The first scenario is mono-modal, in which two visual samples are fed to model that learn the symbolic association. The second scenario is multi-modal, in which one input sample comes from the image domain, and the other sample comes from the text domain. The association model is compared to MLPs that are trained on each input element independently. The goal is to measure the capacity of the association model regarding traditional approach where only one input is used for classification. Experiments show that the performance of the association is lower than MLPs. This performance is expected because of a new learning condition where the relationship between semantic concepts and vectorial representation is included as part of the learning process.

The results presented in this chapter appeared in *ICANN 2016* [68]. This chapter is organized as follows. Section 4.1 explains the association between isolated sample pairs in terms of machine learning components. Section 4.2 describes four datasets, where two of them are mono-modal, and the others are multi-modal. Section 4.3 describes the input features and the network parameters that are used in this section. Section 4.4 presents the results of the association learning framework regarding *Association Accuracy* and *Accuracy*.

4.1 Problem Definition

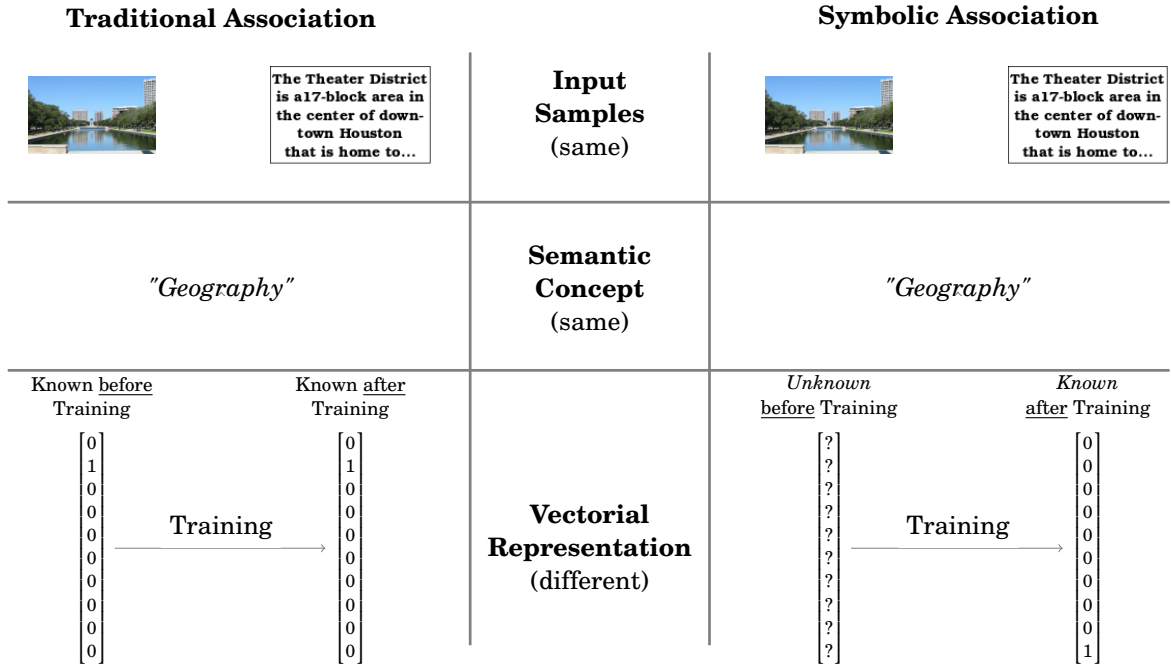


FIGURE 4.1. Difference between the traditional approach and this work concerning learning elements. The traditional approach represents the association by fixing the vectorial representations for each semantic concept in both networks. This setup does not consider that infants learn the association while two elements are presented at the same time. This work adds a new dimension inspired by infant association. In this case, the vectorial representation is self-learned by both networks.

The first challenge of the association model is to match two representations of the same semantic concept. This scenario can occur if two elements appear at the same time in the infant learning scenario. As a reminder, machine learning tasks can be divided into several components: semantic concepts, vectorial representations, classifiers, and sensory input samples. In the traditional approach, each semantic concept is represented by a vectorial representation, and each network uses the same representation. Each network learns to classify input samples given the pre-defined vectorial representation. Thus, the learning parameters are only related to the classifier model. In contrast, the presented model does not require any pre-defined decision about the vectorial representation, and the training algorithm includes the agreement between the vectorial representation as a learning parameter. Hence, the learning parameters are not only related to the

classifiers, but also to the vectorial representation. Figure 4.1 shows an example setup of the difference between the traditional approach and this work.

4.2 Datasets

The evaluation process relies on four datasets. The first two datasets are mono-modal (only images), and the other datasets are multi-modal (images and texts). The descriptions of each dataset are below

MNIST is a dataset for digit recognition [75]. The digits are represented by 28x28 gray-scale images. This dataset is divided into 60,000 and 10,000 samples for training and testing sets (respectively). Besides, MNIST is modified for association learning in the following procedure. First, two disjoint sets are generated from MNIST. Then, both sets are re-arranged in a way that two samples from each set represent the same semantic concept.

COIL-20 is a dataset for object classification [76]. The dataset has 20 objects, and each object is represented by 72 gray-scale images. Each image has been taken at five degrees apart. This dataset does not have a pre-defined training and testing datasets. Thus, all images taken at even angles are considered part of the training set, and the rest of images are considered part of the testing dataset. A similar approach to MNIST is followed for modifying this dataset for the association task.

TVGraz is a multi-modal dataset with ten categories [77]. The dataset was collected by crawling RGB images and their respective web pages. Initially, ten categories from Caltech-256 [78] are used as a query for retrieving the top 1,000 results from Google Image Search. Each image is manually labeled with the following rule: if the image sample has at least a single visible instance of the category, it is labeled as positive sample of the category, otherwise it is labeled as a negative sample of the category. In this work, only positive samples are used. The training and testing set are randomly selected.

Wikipedia featured Articles is another multi-modal dataset collected in October 2009 from the featured Articles offered in Wikipedia [79]. The authors collected the ten most populated categories. Each article is composed of several text sections and RGB images. Thus, a pruning process was required for managing the data in a

more structured manner. At the end of the pruning step, the dataset is based on sections with images and the length of texts is at least 70 words. The size of the dataset is 2,866 samples. The training and testing sets are also randomly selected.

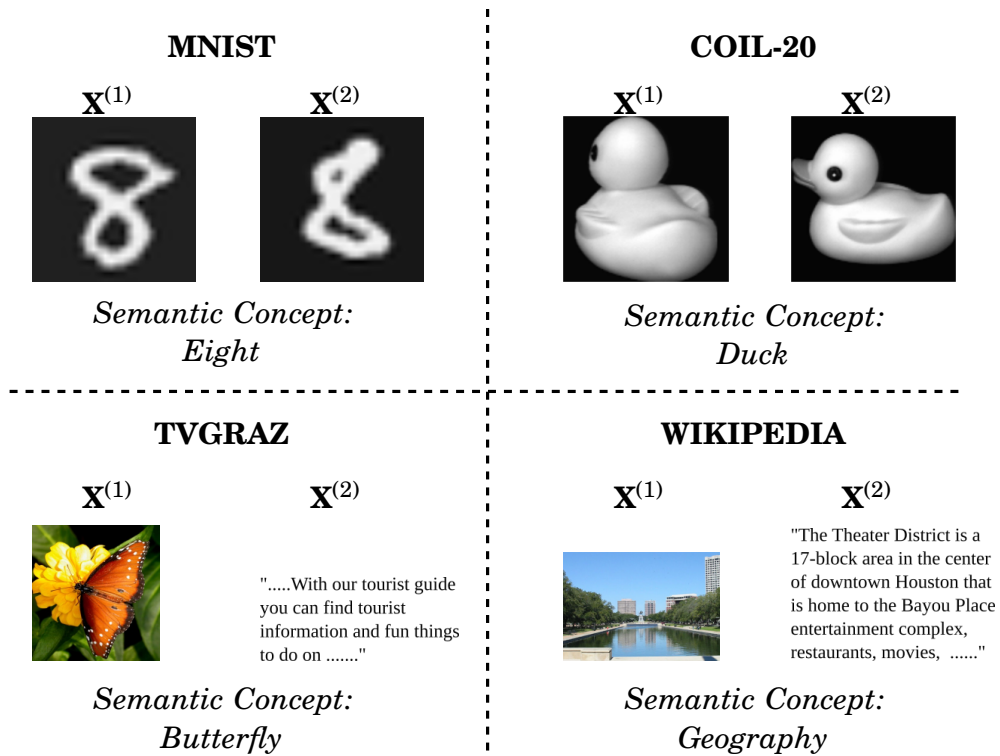


FIGURE 4.2. Examples of four datasets that are used for evaluation. The top datasets are mono-modal, and the bottom datasets are multi-modal.

As mentioned, the mono-modal datasets do not have the structure for the association task. The following procedure modifies both datasets. First, the dataset is divided into two disjoint sets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Second, two elements of the same semantic concept from different sets are linked to each other. The multi-modal datasets are not required to be modified since images and texts have already an established association. Figure 4.2 shows several examples of each dataset. The multi-modal dataset is more challenging because both input domains are different between them.

The evaluation of the model is based on cross-validation. Each dataset is randomly sampled for generating training and testing sets. This process is repeated ten times. Table 4.1 summarizes the sizes of the training and testing set for each dataset.

TABLE 4.1. Description of each dataset. Note that each input sample is actually a pair of elements.

Dataset	Concept	Training Size	Testing Size
<i>MNIST</i>	10	25000	4000
<i>COIL-20</i>	20	360	360
<i>TVGraz</i>	10	1942	652
<i>Wikipedia</i>	10	2146	720

4.3 Features and Network Setups

Each input sample is required to be described using features. In this case, two sets of different features are used. In mono-modal datasets, the raw pixels of the images are used as input features after re-scaling the pixel values between zero and one. Afterwards, the input features are flattened into one dimension. In multi-modal datasets, *Latent Dirichlet Allocation* (LDA) [80] and *Bag-of-Visual-Word* (BoVW) [81] based on SIFT [82] are used for representing texts and images (respectively). LDA is a generative probabilistic model, in which documents are represented as random mixtures over *latent* topics. Each topic is represented by a distribution over the words. Additionally, the topics are not pre-defined, and they are developed based on the likelihood of term co-occurrence. BoVWs is inspired by document classification where each document is represented by a histogram of independent features. Two steps are required for replicating this idea in object classification. First, each image is represented by a set of descriptors, e.g., SIFT. Second, all collected patches are used to generate a visual codebook. The goal is to group similar patches. A standard approach is *K-means clustering*. Each cluster is represented by the centroid of all elements that are in the cluster. Sometimes the cluster center is called *codeword*. Finally, each descriptor of an image is matched to the closest *codeword*, and a histogram of *codewords* represents the image.

In this setup, 100 topics are used for LDA and a codebook of 1024 for BoVW. Moreover, LDA and SIFT features are extracted using NLTK¹ and VLFeat² respectively. Pereira and Vasconcelos [83] also used these features. Finally, the extracted multi-modal features are rescaled to mean zero and standard deviation one.

¹<http://www.nltk.org/>

²<http://www.vlfeat.org/>

The association learning framework based on MLP has the following parameters for each scenario:

Mono-modal scenario The association model has only one hidden layer with 40 neurons for each internal MLP. The learning rate is set to 0.0001 with momentum 0.9. The learning rate for the weighting vectors is set to 0.01. The mini-batch size is set to 1,000 and 360 samples for MNIST and COIL-20, respectively.

Multi-modal scenario The model has one hidden layer with 150 neurons for each internal MLP. The learning rate is 0.00001 and momentum 0.9. The learning rate of the weighting concepts is 0.01. The mini-batch size is 300 samples.

4.4 Results and Discussion

The performance of the model is measured based on two metrics. The first metric evaluates the accuracy if two networks predict the same semantic concept. This metric is called *Association Accuracy* (AAcc) and is formally defined by :

$$AAcc = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(\hat{\mathbf{y}}_i^{(1)}, \hat{\mathbf{y}}_i^{(2)}, \mathbf{y}_i), \quad (4.1)$$

$$\mathbf{1}(\hat{\mathbf{y}}_i^{(1)}, \hat{\mathbf{y}}_i^{(2)}, \mathbf{y}_i) = \begin{cases} 1 & \hat{\mathbf{y}}_i^{(1)} == \hat{\mathbf{y}}_i^{(2)} == \mathbf{y}_i, \\ 0 & \text{otherwise,} \end{cases} \quad (4.2)$$

where $\hat{\mathbf{y}}_i^{(1)}$ and $\hat{\mathbf{y}}_i^{(2)}$ are the output classification from each network, \mathbf{y}_i is the ground-truth label, N is the total number of elements.

The second metric is *Accuracy*, which shows the *ratio* between predictions and correct output predictions. This metric is applied to each network independently and is defined by:

$$Accuracy = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(\hat{\mathbf{y}}_i, \mathbf{y}_i), \quad (4.3)$$

$$\mathbf{1}(\hat{\mathbf{y}}_i, \mathbf{y}_i) = \begin{cases} 1 & \hat{\mathbf{y}}_i == \mathbf{y}_i, \\ 0 & \text{otherwise,} \end{cases} \quad (4.4)$$

where $\hat{\mathbf{y}}_i$ is the prediction of the network (also one network from the association learning framework), and \mathbf{y}_i is the desired semantic label. The rest of this section describes the results of mono- and multi-modal scenarios.

TABLE 4.2. Association Accuracy (%) of the presented model and the traditional approach in the mono-modal scenario. Both cases are using MLP architectures. The performances of both setups are quite similar. It is expected that the new constraint affects this model.

Dataset	This Work	Standard MLP
<i>MNIST</i>	94.61 ± 0.24	95.02 ± 0.32
<i>COIL-20</i>	92.86 ± 1.65	92.94 ± 0.62

4.4.1 Mono-modal Scenario

In this scenario, both networks in the association learning framework are being fed with data that has similar properties. Thus, it is expected to reach similar performance. The association accuracy is compared between the presented model and the standard approach. As a result, the presented model can learn the association without pre-defined the mapping between semantic concepts and vectorial representations. In other words, the model self-learns this relationship. The performance of each model is shown in Table 4.2.

TABLE 4.3. Accuracy (%) of this work and the standard MLP in the mono-modal scenario. The performances of the presented model and the traditional setup are similar. The focus of this comparison is to solve the semantic relationship without losing performance.

Dataset	Format	Method	
		This Work	Standard MLP
<i>MNIST</i>	visual	97.32 ± 0.30	97.50 ± 0.23
	visual	97.18 ± 0.21	97.42 ± 0.22
<i>COIL-20</i>	visual	97.20 ± 1.09	97.39 ± 1.27
	visual	97.17 ± 1.09	96.89 ± 1.12

Comparing both performances, the presented model maintains a similar performance to MLP with the traditional approach (see Table 4.3). Furthermore, the accuracy per

class of both datasets (MNIST and COIL-20) is also similar. The results are observed in Figures 4.3 and 4.4.

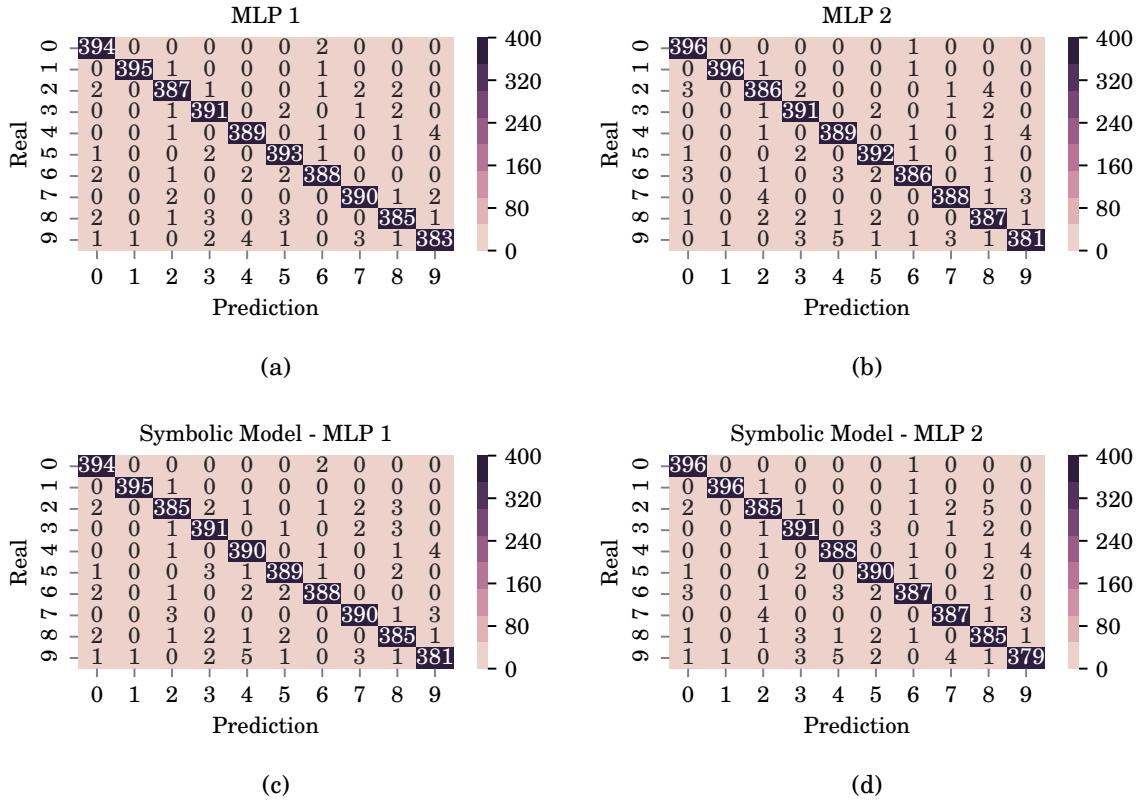


FIGURE 4.3. Confusion Matrices of traditional setup and this work (MNIST Dataset). It is observed that the accuracy per category between the tradition setup (top row) and the presented model (bottom row) has almost identical performance.

Furthermore, the reason why the symbolic association relies on the convergence of the semantic concepts. Figure 4.5 shows an example of several iterations and some principal components of the presented model during training for MNIST. The starting stage shows the initial values for the output vectors $\hat{\mathbf{y}}^{(1)}$, $\hat{\mathbf{y}}^{(2)}$ and weighting vectors $\gamma^{(1)}$, $\gamma^{(2)}$. Note that the mapping between semantic concepts and vectorial representations is not fixed before training, and the association matrix shows only one relation between both MLPs at position (0, 0). In other words, both networks predict all samples as category zero. After the first iteration, the association matrix shows a different relation at position (2, 2) from the starting stage. The output vector has already changed to sparse values. In more detail, the mapping between semantic concepts and the vectorial representation is

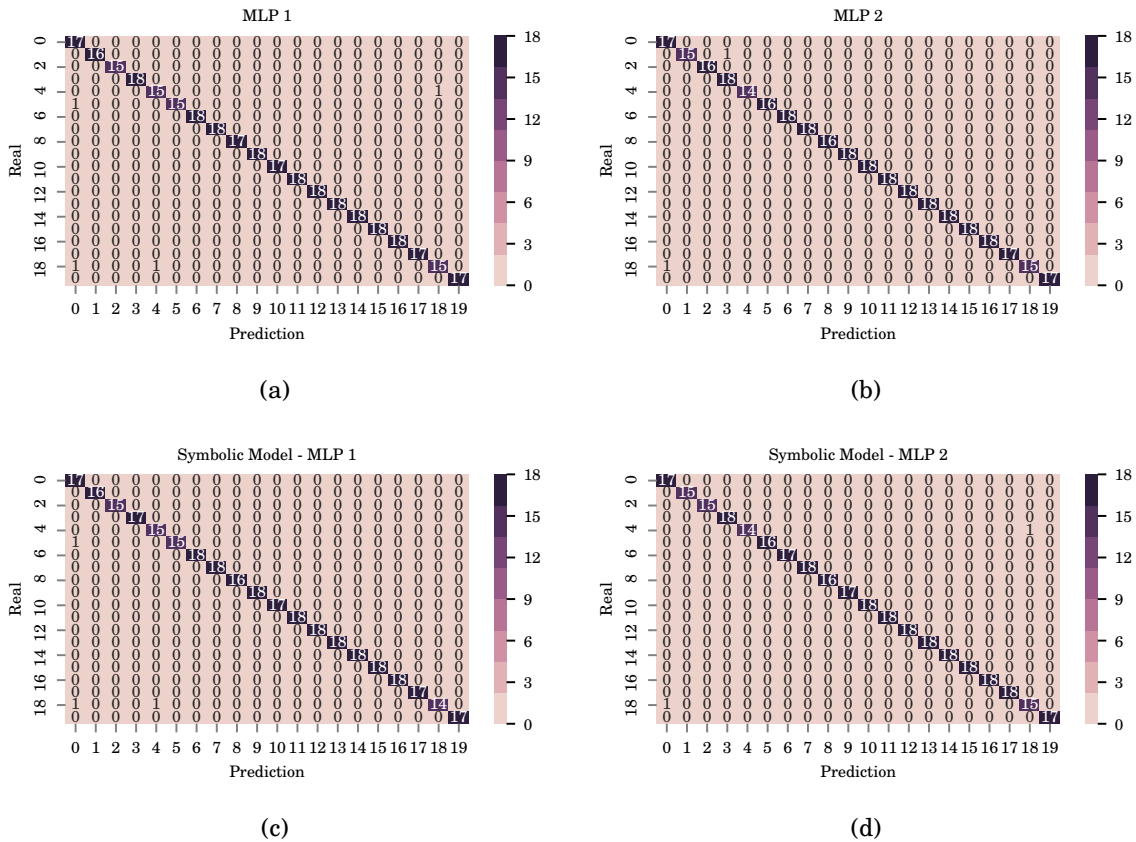


FIGURE 4.4. Confusion Matrices of the traditional setup and the symbolic association framework (COIL-20 Dataset). Similar to Figure 4.3 the performances (top vs bottom rows) are quite similar.

represented by the weighting vectors. For example, the minimum value (light color) for the semantic label at position *four* is mapped to the ground truth at position *three*.

This section has presented the results of the symbolic association framework. The performances in two mono-modal datasets are similar. This behavior is expected because each internal MLP is trained on the same data domain. The next section describes the results of the presented model in a more challenging scenario where the input sets are different, and the performance of each MLP might be different.

4.4.2 Multi-modal Scenario

The association learning framework is evaluated in a multi-modal scenario, where one modality is visual elements, and the other modality is text description elements. As a

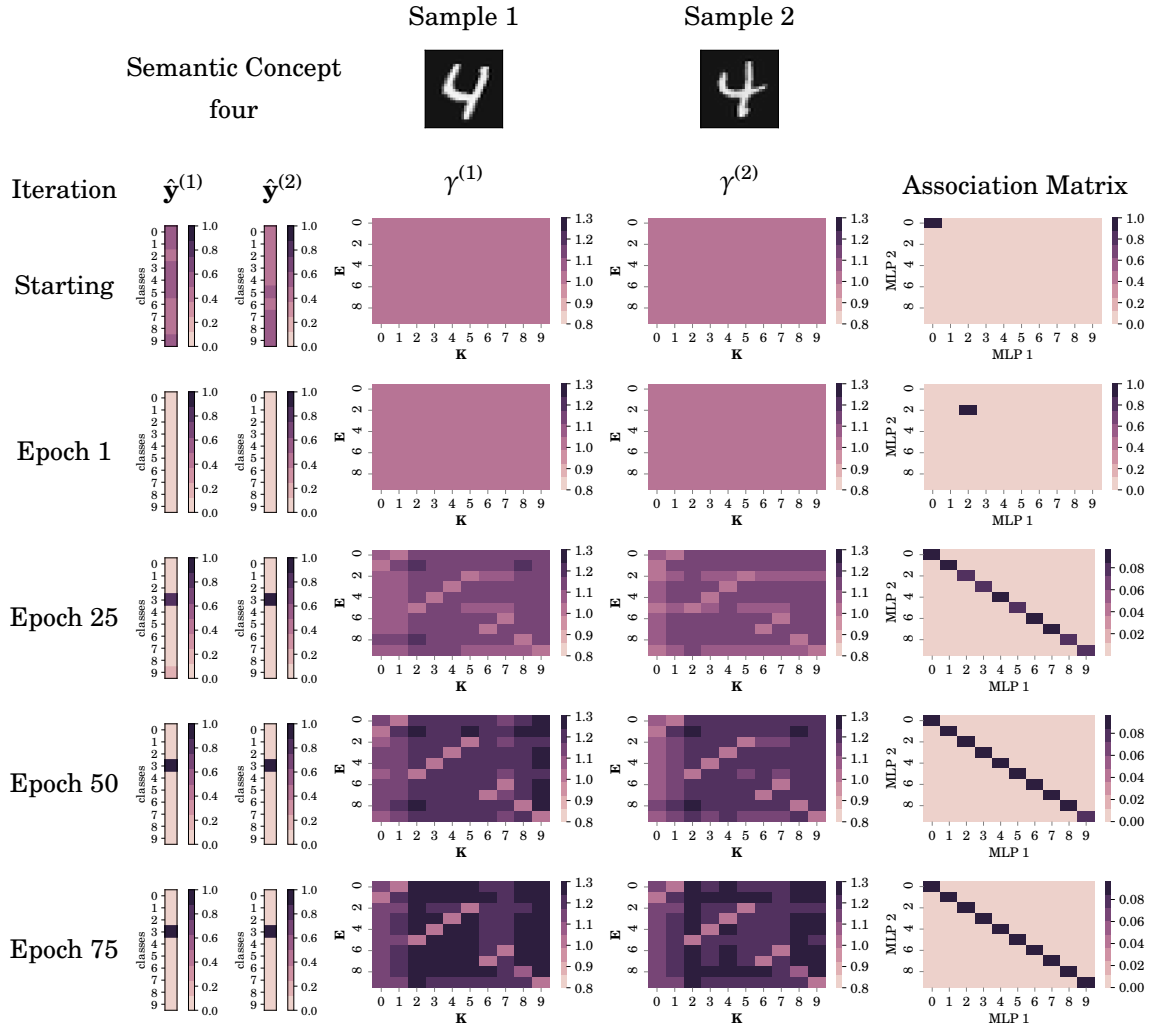


FIGURE 4.5. Example of the training process at different stages (mono-modal scenario). The model can learn because semantic concepts converge between both networks. The convergence is shown in two components. The first component is the weighting vectors $\gamma^{(1)}$ and $\gamma^{(2)}$ that agree on the same relationship (light values show the link between semantic concepts and the vectorial representation). For example, the maximum element of both row output vectors is *three*, which is associated to the semantic label *four* in $\gamma^{(1)}$ and $\gamma^{(2)}$. The second component that shows the convergence is the association matrix. Both networks learn to predict the same semantic concept for each sample pair.

TABLE 4.4. Association Accuracy (%) of the symbolic association model and the traditional setup in the multi-modal scenario. In this case, the performances are not as good as the mono-modal dataset.

Dataset	This Model	Standard MLP
<i>Wikipedia</i>	11.82 ± 2.25	12.97 ± 1.11
<i>TVGraz</i>	28.30 ± 1.45	31.50 ± 1.16

result, the feature space of each modality is different between them and the capacity of both networks might also be different.

TABLE 4.5. Accuracy (%) of the presented model and the traditional setup in the multi-modal scenario. In this case, the model reaches similar performance compared to the standard MLP trained in each modality independently.

Dataset	Format	Method	
		This Model	Standard MLP
<i>Wikipedia</i>	visual	27.44 ± 2.69	28.38 ± 1.60
	text	34.07 ± 2.96	37.25 ± 1.43
<i>TVGraz</i>	visual	53.19 ± 2.74	55.97 ± 1.86
	text	52.74 ± 2.45	53.65 ± 1.38

The association accuracy is shown in Table 4.4. In this case, the performance of the presented model remains similar to the traditional approach that is trained in each modality independently. In more detail, the accuracy of each network also remains similar (see Table 4.5). One of the reasons might be that Wikipedia has more inter-class variability.

Figures 4.6 and 4.7 show a new pattern that has not been observed in the mono-modal scenario. In the Wikipedia dataset, the traditional approach has a problem where many samples are misclassified as the semantic concept *seven* in MLP1. This misclassification does not occur in MLP2. On the other hand, the association framework has the same misclassification in both MLPs. These results suggest that bad performance in one classifier is spread to the other network as well. However, the results of TVGraz dataset show the opposite. Two classifiers have good performance, and it can be beneficial under some conditions. For example, semantic concept *seven* in this work (MLP1 and MLP2) reaches

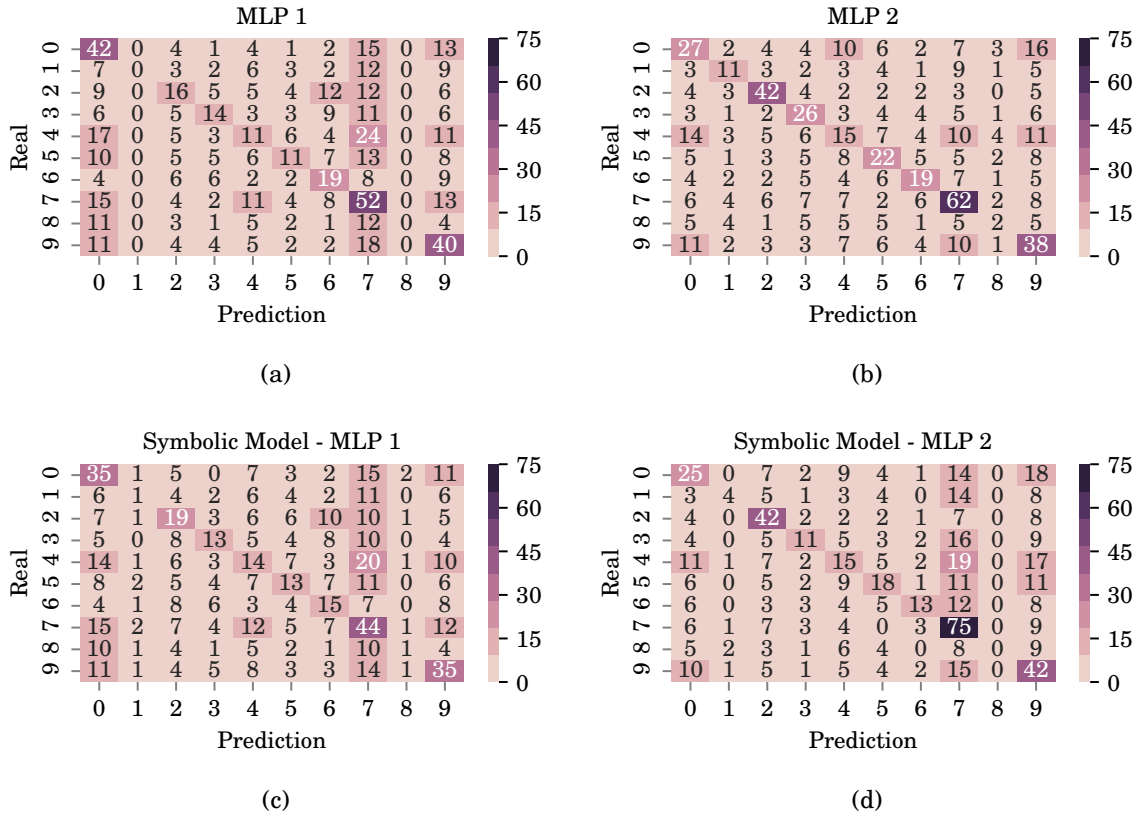


FIGURE 4.6. Confusion Matrices between the traditional setup and the symbolic association model (Wikipedia Dataset). The performances of both models are still similar regardless of modality. However, the symbolic association model reaches a lower accuracy for some semantic concepts. For example, the semantic concept *nine* (c) reaches less accuracy than the independently trained MLP (a). However, the same semantic concept reaches better results in the other internal MLP (d) than the traditional setup (b).

better results compared to MLP trained of each modality independently. However, this claim is not conclusive because most of the cases the performance decreased a little bit.

Similar to the mono-modal scenario, the training algorithm also converges to the same vectorial representation for both networks. Figure 4.8 shows another example of the learning behavior based on one multi-modal sample from TVGraz dataset. It is observed that both networks collapsed into one classification output instead of spreading over all classes. After 100 epochs, there are some initial results that both networks have started learning some of the semantic concepts. After 300 epochs, the weighting vectors of each network have similar relations between semantic concepts and vectorial

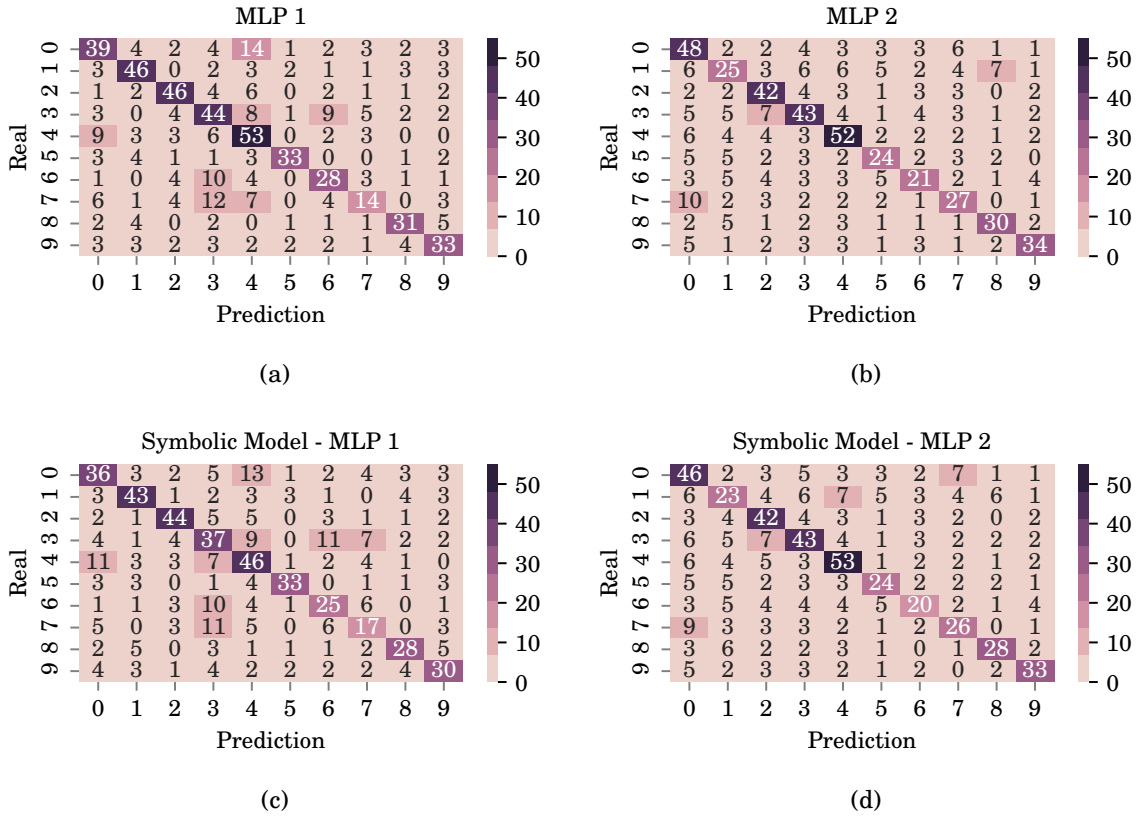


FIGURE 4.7. Confusion Matrices between the traditional setup and the symbolic association model (TVGraz Dataset). As with the previous experiment (Figure 4.6), the performances of both setups are similar. Additionally, the same behaviors are also presented here.

representations. However, the association matrix has still not converged. After training, the association matrix shows that both networks converge to the same semantic concept. Additionally, each weighting vectors shows only one relation (light color at each column) between the semantic concepts and the vectorial representation.

4.5 Summary

This chapter has described the evaluation of the symbolic association. Two cases were used for this purpose. The first case is a mono-modal scenario that uses two visual elements to represent the same semantic concept. The second scenario is a multi-modal scenario, where text and visual elements represent the same semantic concept. Several findings are summarized based on the results.

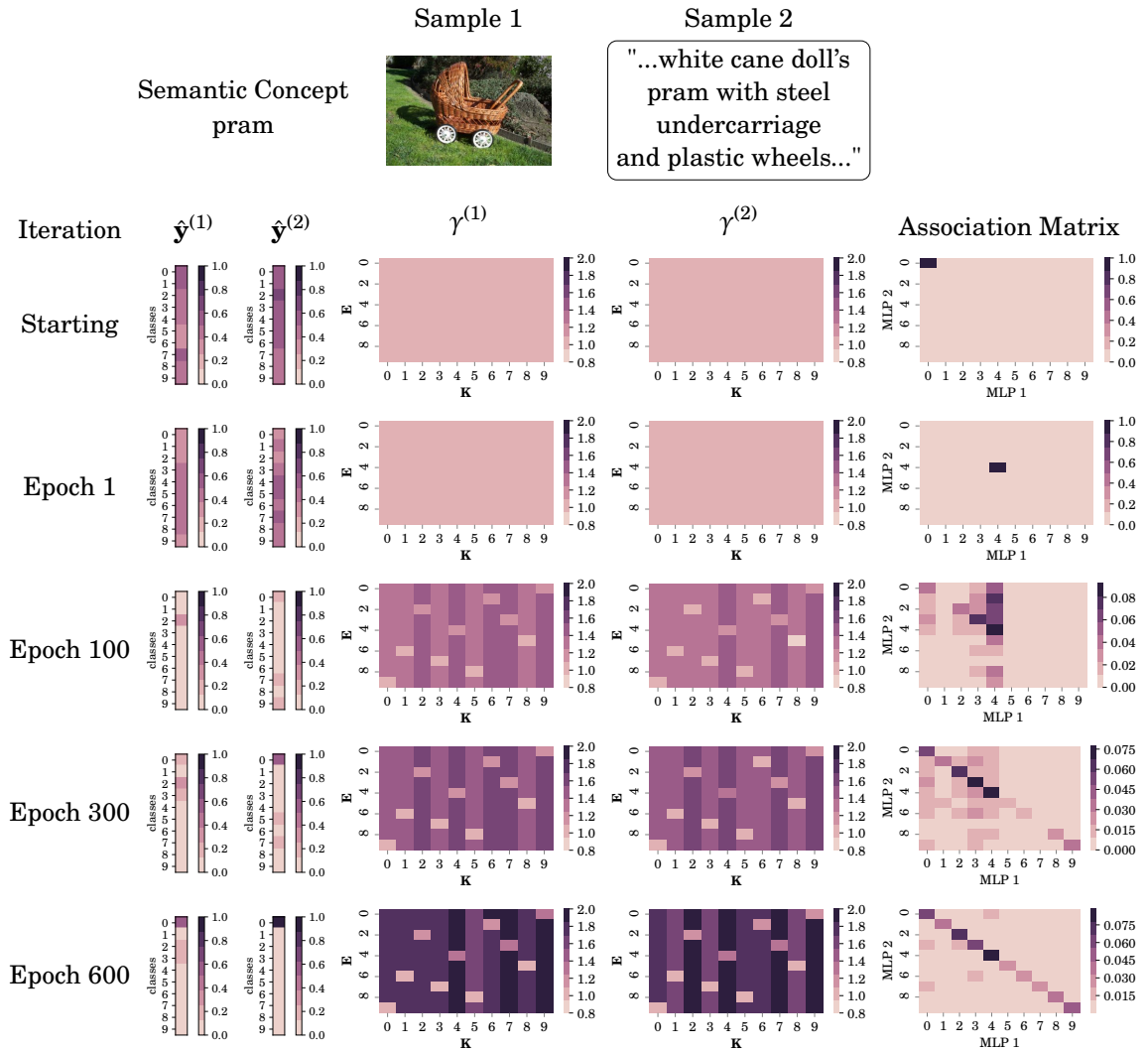


FIGURE 4.8. Example of the training algorithm at different stages (TVGraz Dataset). The convergence pattern that has been explained in Figure 4.5 is also presented in this example. The behavior of weighting vectors $\gamma^{(1)}$ and $\gamma^{(2)}$ and the association matrix remain the same as in the mono-modal case.

- The symbolic association has two internal MLPs, which are compared to MLPs that are trained independently on each input set. The performances between both networks are similar in both test cases: mono- and multi-modal scenarios.
- The symbolic association model works because of convergence of semantic concepts convergence. This behavior is the same in the mono- and multi-modal datasets.
- The internal networks of the association model can weakly transfer their performance between each other. In other words, a network can cause the other network to improve inside of the association model, whereas the MLP trained independently with the traditional approach decreases the performance.

Next chapter presents results of the association framework applied to sequences with weakly labels. In that case, the second version of the association framework exploits LSTM networks that are trained based on a CTC layer.

ASSOCIATION LEARNING IN SEQUENCES

This chapter presents the results of the second version of the symbolic association framework. Similar to the previous chapter, two scenarios are also considered in this chapter. The goal of the first scenario (mono-modal) is to associate two sequences that have the same input type (i.e., text lines) with the same order of semantic concepts. On the other hand, the second scenario (multi-modal) associates two sequences of different input types (e.g., visual and audio). In both cases, the input sequences do not have any pre-segmentation step before training. In other words, the annotation of the scenarios is weakly labeled sequences (*c.f.* Section 2.2.4). Note that both sequences are trained based on the alignment between the output network (i.e., $\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_t$) and the desired target sequence (i.e., c_1, \dots, c_d) where $d \ll t$.

The presented results have been published in ICDAR2015 [69], CoCo2016 [70], and NC²/2016 [84]. The chapter is divided into three sections. Section 5.1 describes the association problem of sequences in terms of three elements: input samples, classifiers, and vectorial representations. Section 5.2 shows the mono-modal association, based on two contexts. In the first context, one input sequence can use the latent space produced by another input sequence. The second context corresponds to one LSTM network, which can learn one input sequence based on the latent space produced by another input sequence and a different LSTM network. Section 5.3 shows a more complex scenario where the input samples are from different formats. As a result, one network learns a latent space that is produced by another network with input samples of different formats.



Traditional Association			Symbolic Association			
		Input Sequence (same)				
<i>"one four two one five"</i>		Semantic Sequence Association (same)	<i>"one four two one five"</i>			
Known before Training $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	Training \rightarrow	Known after Training $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	Vectorial Representation (different)	Unknown before Training $\begin{bmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{bmatrix}$	Training \rightarrow	Known after Training $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$

FIGURE 5.1. Differences between components of traditional and symbolic association tasks when the input samples are sequences. The new setup is more challenging than the association task defined in Chapter 4. In this setup, the association task occurs between two weakly labeled sequences. The main difference between the traditional and symbolic associations is the component of the vectorial representation. In the one hand, the vectorial representation is already defined before training and the association between both sequences are already defined in the traditional association scenario. In the other hand, the symbolic association does not define the association via vectorial representation, which is part of the training step.

The performance of the symbolic association model is similar to the traditional setup.

5.1 Problem Definition

The presented association problem is related to parallel sequences where the format of both samples can be the same or different. For instance, one sample could be a textual description and the other sample an image. Formally, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are represented by a sequence of vectors $\mathbf{x}_1^{(1)}, \dots, \mathbf{x}_{t_1}^{(1)}$ and $\mathbf{x}_1^{(2)}, \dots, \mathbf{x}_{t_2}^{(2)}$, respectively. Both input samples represent the same sequence of semantic concepts c_1, \dots, c_d .

As presented in Figure 5.1, the association scenario here is more challenging due to the following reasons. First, the relationship between both input sequences is *one-to-one*:

the semantic concepts of both input sequences have the same order. However, the relation between feature input vectors is not *one-to-one* because each input sequence is a set of many input vectors. Therefore, the relationship between input vectors of each sequence is not defined because of weakly labeled sequence. As a reminder, note that Section 4.1 described that the association between isolated input pairs. In that case, each input sample is represented by one vector and the link between both input vectors is clearly defined. Second, the learning model is trained on the *Latent Space* of another input sequence, whereas, the association learning in isolated pair samples is based only on the output layer. Hence, one sequence is predicted based on the other sequence.

The evaluation process utilizes two metrics. The first metric is the *Sequence Association Accuracy* (SeqAAcc) for sequences, which is defined by:

$$SeqAAcc = \frac{\sum_{i=1}^N LCS(\hat{\mathbf{y}}_i^{(1)}, \hat{\mathbf{y}}_i^{(2)}, \mathbf{y}_i)}{\sum_{i=1}^N len(\mathbf{y}_i)}, \quad (5.1)$$

where $\hat{\mathbf{y}}_i^{(1)}$ and $\hat{\mathbf{y}}_i^{(2)}$ are the predicted sequence for each input, \mathbf{y}_i is the target sequence. The *function* LCS is the length of the longest common sequence between $\hat{\mathbf{y}}_i^{(1)}$, $\hat{\mathbf{y}}_i^{(2)}$, and \mathbf{y}_i , and $len(\mathbf{y}_i)$ is the number of elements of sequence \mathbf{y}_i .

The second metric evaluates the performance of each network independently. The *Label Error Rate* (LER) is defined by:

$$LER = \frac{1}{N} \sum_{i=1}^N \frac{ED(\hat{\mathbf{y}}_i, \mathbf{y}_i)}{len(\mathbf{y}_i)}, \quad (5.2)$$

where $\hat{\mathbf{y}}_i$ is the predicted sequence of one network, function $ED(\hat{\mathbf{y}}_i, \mathbf{y}_i)$ is the edit distance between the predicted sequence $\hat{\mathbf{y}}_i$ and the desired target \mathbf{y}_i .

5.2 Scenario 1: Parallel Mono-modal Sequences

This section describes the feasibility of the association model based on two conditions. First, it is possible to train a LSTM network with the latent space produced by another input sequence. In this case, the input sequence represents the same array of semantic concepts. Second, the feasibility to train two LSTMs networks that generate two latent

spaces generated from two different input sequences. Again, both input sequences represent the same ordered series of semantic concepts.

5.2.1 Dataset Preparation

For both scenarios, a mono-modal dataset has been produced based on MNIST [75]. The generation procedure is described below:

Generating of semantic concepts: Each semantic concept was randomly generated with a number between four and eight digits. There is not constraint that the same digit can be repeated more than one in the number.

Generating visual sequences: There is an instance per each digit in the sequence. Afterwards, a dynamic black background is located before and after each digit. The length of the background is randomly selected between three and ten columns.

Training and Testing sets: As a reminder, the training and testing are generated using the original split of MNIST dataset. Therefore, this generated dataset also uses the same division. The training set has 50,000 sequences, and the testing set has 15,000 sequences. The experiment design follows cross-validation approach where 10,000 sequences and 3,000 sequences are randomly selected. This random selection is repeated ten times; thus, the dataset presents ten different setups of training and testing sets. Figure 5.2 shows several examples of the generated dataset.

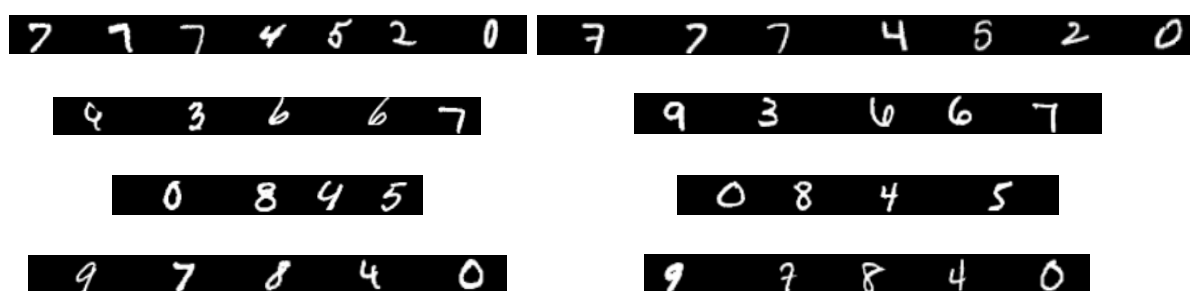


FIGURE 5.2. Several examples of sequences in the mono-modal dataset. It is observed that each row represents the same semantic sequence with different instances.

TABLE 5.1. Sequence Association Accuracy (%) and Label Error Rate (%) of one LSTM network trained independently to each input set and the symbolic association model (implemented by one LSTM). These results show that the training based on input sequence and different output vectors is possible.

Models	SeqAAcc (%)	LER (%)	
		Sequence 1	Sequence 2
LSTM trained for one sequence	93.07 ± 1.47	3.47 ± 0.99	3.52 ± 0.80
association learning (version: one LSTM)	95.87 ± 0.88	2.12 ± 0.46	2.15 ± 0.43

5.2.2 Input Features and LSTM setup

As mentioned, two architectures are evaluated in the mono-modal scenario. In this test case, the visual sequences were normalized between 0.0 and 1.0 for both architectures. Both architectures have similar parameter settings. The hidden size is 20 memory cells, and the weighting vectors are initialized to 1.0. The learning rate is set to 0.0001 with momentum of 0.9 for the first architecture whereas the learning rate is set 0.00001 with the same momentum in the second architecture. The learning rate of the weighting vectors is set to 0.001 in the first architecture, but the second architecture uses a learning rate 0.01.

5.2.3 Mono-modal Latent Space produced by one LSTM Network

This scenario can be understood as a simplified version of the general model described in Section 3.4. Figure 5.3 shows an example of the simplified model. This experimental setup evaluates the LSTM training based on latent spaces generated by a different sequence. The main difference is to use only one LSTM instead of two LSTMs, and the training procedure uses input sequence 1 with the latent space of input sequence 2 and vice versa.

Table 5.1 shows the Sequence Association Accuracy and the Label Error Rate of the association model with one LSTM network and LSTM network trained independently to each input sequence set. One interesting outcome observed is that the effect of using latent space produced by different input samples has a positive effect on the performance. Additionally, the process of learning the mapping between semantic concepts and vectorial representation did not hurt the performance of the model.

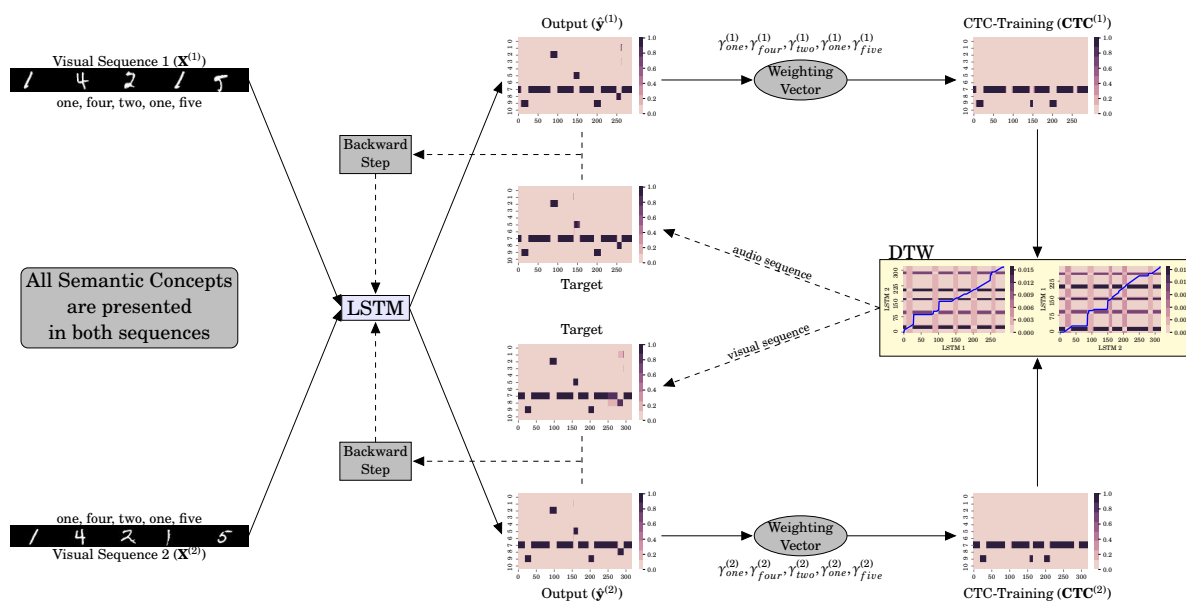


FIGURE 5.3. Example of the Symbolic Association Learning implemented with one LSTM. In this case, the setup evaluates the feasibility of training input sequence ($X^{(1)}$) with a different output sequence ($\hat{y}_{t_2}^{(2)}$), and vice versa. The training follows the same EM-approach. The E -step (solid line) generates the output sequence of each input and CTC-alignment. The M -step (dashed line) updates the parameters of the association model.

Figure 5.4 shows several examples of the predicted sequence and the DTW cost matrices. The output classification is shown in the positions with the maximum activation values (dark color). Note that the output classification are vectors, for instance, $\mathbf{y}_1^{(1)}, \dots, \mathbf{y}_{t_1}^{(1)}$. Each example has a different length (x-axis of output classification columns). The cost matrices show the alignment (blue line from the bottom left corner to the top right corner) between both output vectors. It can be observed that there is a grid pattern in the cost matrix. Those columns and rows are the positions that represent the relationship between each time step. That relationship can be divided into two types: a) semantic concepts between both sequences and b) the segmentation (blank class) between each output classification. The alignment path runs through each of these intersections.

The results indicate that it is viable to train an LSTM network based on DTW alignment and the latent space produce by a different input sample. Next, this model is extended to two LSTM networks. This extension provides more flexibility to the model because each LSTM can have different architectures (i.e. input and memory cells sizes).

5.2. SCENARIO 1: PARALLEL MONO-MODAL SEQUENCES

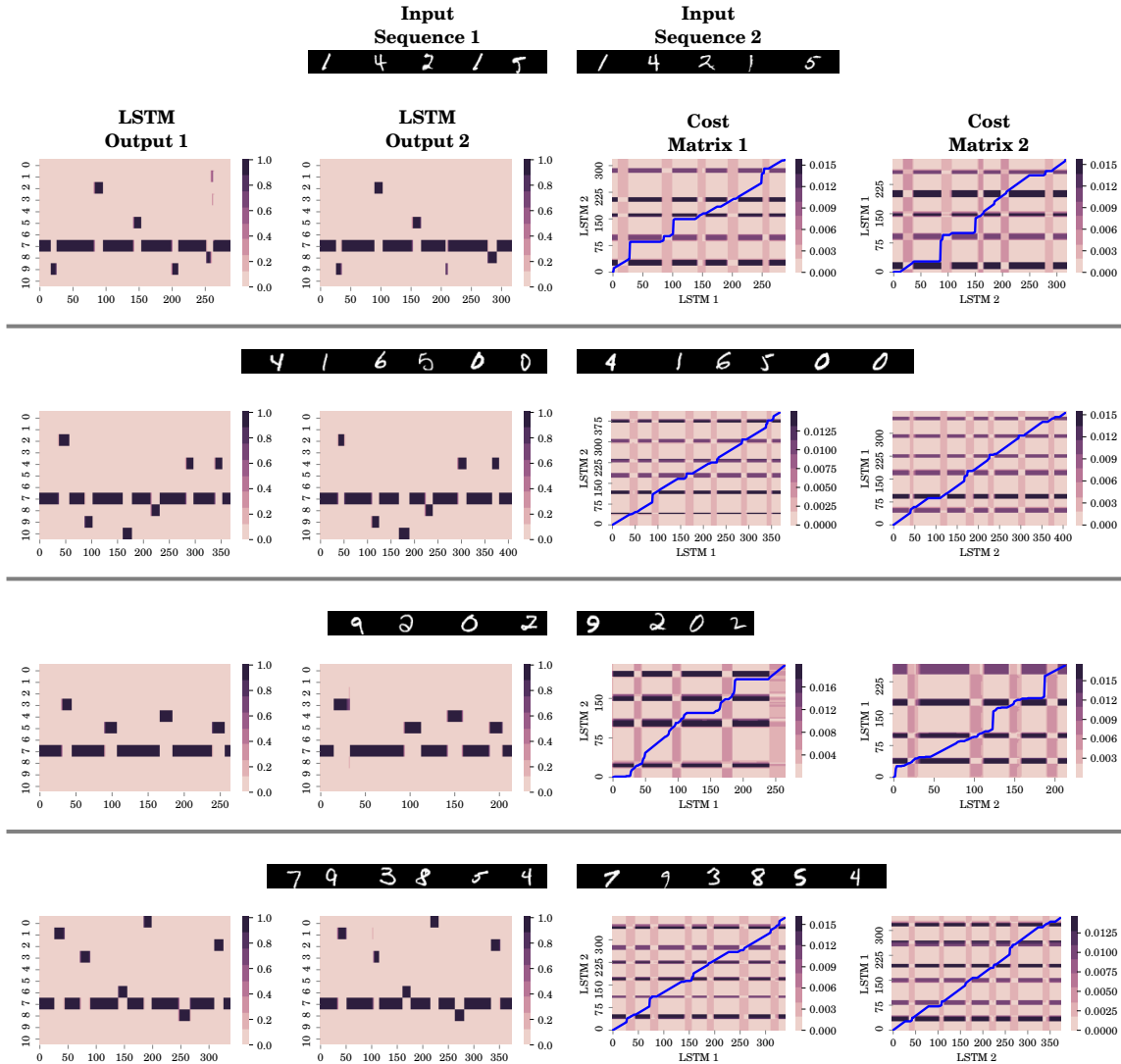


FIGURE 5.4. Several examples of the classification with the symbolic association framework, one LSTM version. First, the DTW alignment can successfully transform information from one sequence to another sequence in a scenario with sequences with different lengths and weakly labeled. Second, the classifications of both output sequences agree on the same vectorial representation. For example, the first digit of the sequence (top row) is one, and both output classifications are represented by the same vectorial representation.

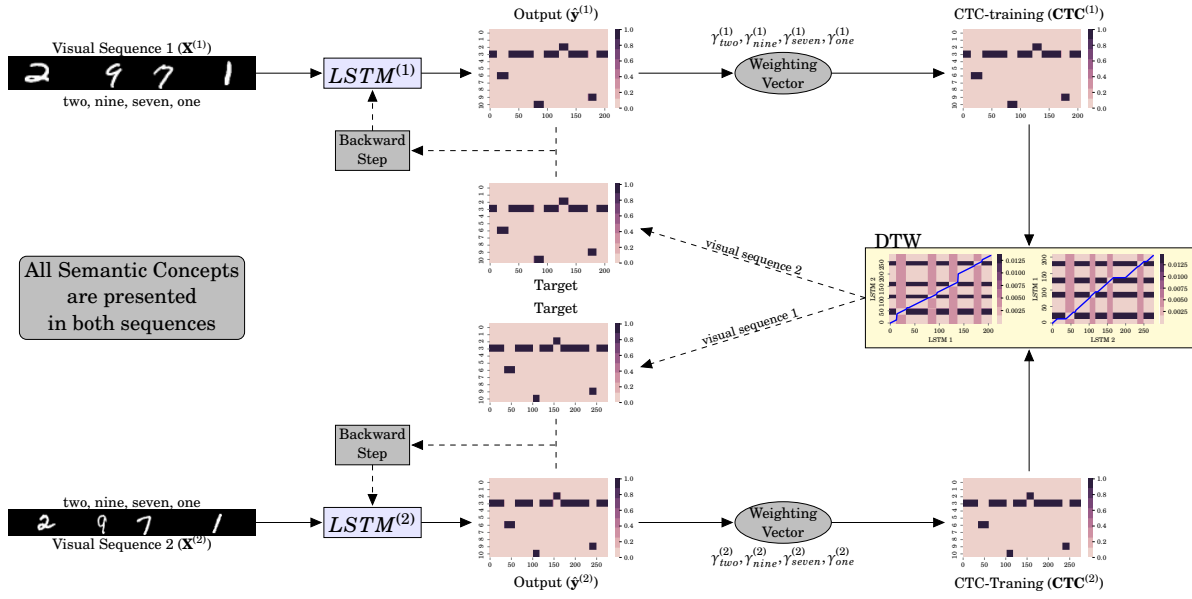


FIGURE 5.5. Association Learning using two LSTMs in the mono-modal scenario. The *E-step* (solid line) can be summarized in two elements: pass each input sequences through each LSTM and CTC alignment based on the output sequences. The *M-step* (dashed line) updates the parameters of both LSTM networks and the weighting vectors.

5.2.4 Mono-modal Latent Space produced by two LSTM Networks

This section presents the results of the symbolic association using two LSTMs. Note that one internal LSTM employs a loss function between an input sequence ($\mathbf{X}^{(1)}$) and a target vector produced by another sequence ($\mathbf{X}^{(2)}$). With this in mind, the goal of this section is to evaluate the training process based on the alignment of two latent spaces that share similar information.

Figure 5.5 shows an example of the symbolic association model implemented by two LSTMs. It can be observed that *LSTM1* learns the feature from input sequence 1 in combination with the latent space generated from input sequence 2. Similar to Section 5.2.3, the model is compared to a single LSTM that is trained to each input set (i.e., input sequence 1). Table 5.2 shows that the presented model reaches similar performance to LSTM trained with the traditional setup. These results show that LSTM can learn in a latent space generated by different input sequence and network.

TABLE 5.2. Sequence Association Accuracy (%) and Label Error Rate (%) between one LSTM network trained independently to each input set and the symbolic association model (implemented by two LSTM networks). These results show the training based on input sequence and different output vectors is possible.

Models	SeqAAcc (%)	LER (%)	
		Sequence 1	Sequence 2
LSTM trained for one sequence	93.07 ± 1.47	3.47 ± 0.99	3.52 ± 0.80
association learning (version: two LSTMs)	95.69 ± 0.27	2.29 ± 0.27	2.21 ± 0.17

Note that the presented model can correctly classify input sequences even if LSTM network trained independently has failed to classify them. Figure 5.6 shows several examples where the presented model and LSTM classify input sample correctly and incorrectly.

In this section, the results have shown that LSTM can learn given a latent space that is not produced by the same input sequence. With this in mind, the question arises if the latent space can learn even in a different input feature set or input format. This question is evaluated in the next section.

5.3 Scenario 2: Parallel Multi-modal Sequences

The symbolic association has been applied to mono-modal sequences. The results in the previous section have shown that latent spaces have useful information for cross-training. This is expected because the input sequences are in the same data domain.

5.3.1 Dataset Preparation

In this section, the evaluation of the presented model occurs in a more complex scenario. Moreover, the formats of each input sequence are different: visual and audio. This scenario is similar to *reading aloud*. Three multi-modal datasets are used for this evaluation purpose. The first dataset is a digit recognition task, the second dataset is a letter recognition task, and the third dataset is a word recognition task. Each of these datasets has visual and audio components.

Digit Recognition This dataset is generated by two components. The first element

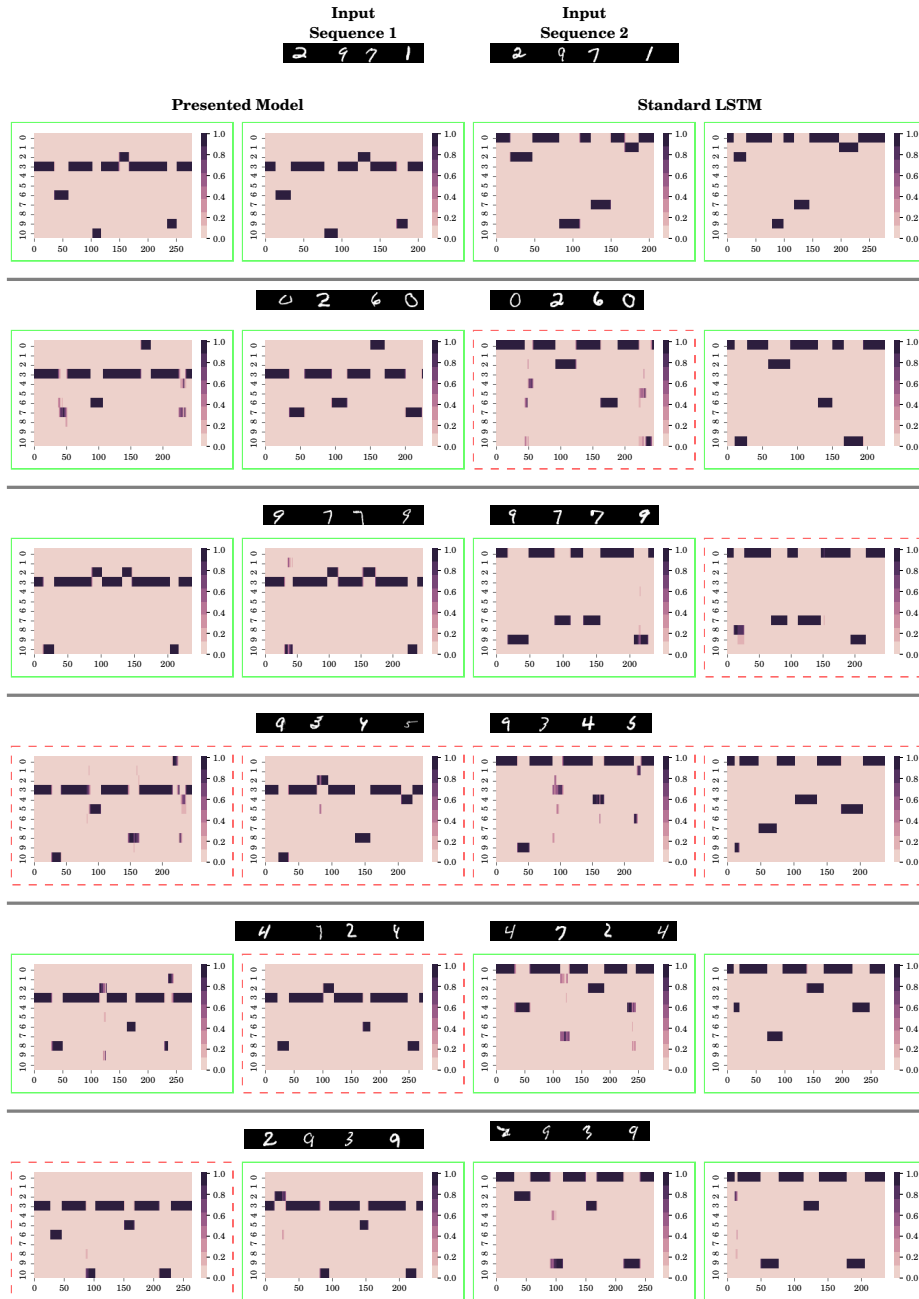


FIGURE 5.6. Several examples of the prediction step (correct-green solid square, incorrect-red dashed square). Two architectures are compared: the symbolic association (first two columns) and a LSTM trained independently on each input set (last two columns). In this manner, it is possible to compare if the presented model adds extra noise during training. There are cases in which that the symbolic association predicts correctly both sequences, whereas one LSTM predicts them incorrectly (second and third rows). On the other hand, there are examples, in which the opposite situation occurs. Symbolic association fails whereas both standard LSTMs are correct.

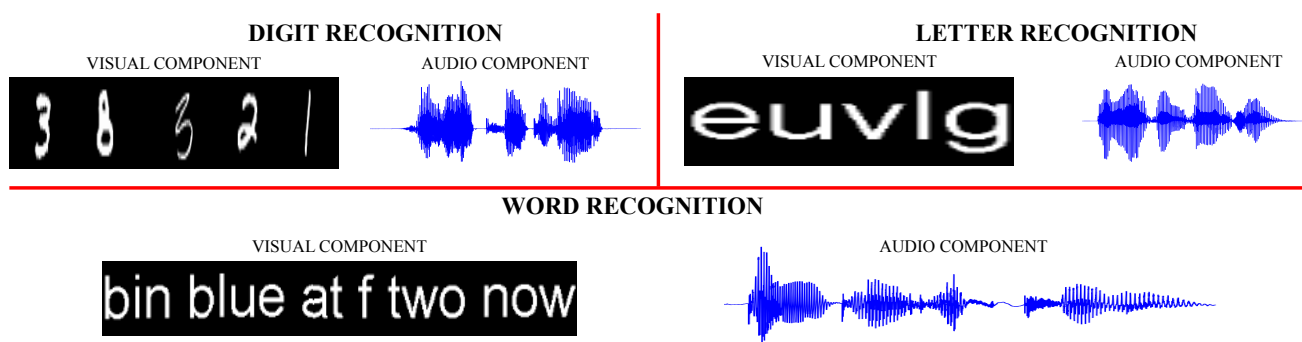


FIGURE 5.7. Examples of the three multi-modal datasets. Note that two datasets (top row) have semantic concepts that are shorter in the visual and audio components than the other dataset (bottom row).

is the visual representation using MNIST [75], and the second element is the audio representation generated by *Festival Toolkit* [85]. The number of semantic concepts is ten. The visual representation follows this procedure. First, the number of digits in the sequence is randomly generated between three and eight elements. A black background is attached after and before each digit. The length of the black background is randomly selected between three and eight columns. Finally, the digits are stacked horizontally. For the audio component, the *Festival Toolkit* generates audio files given the semantic sequence. The artificial voices¹ are randomly selected, from four predefined artificial English speakers in the toolkit, for generating audio files. This procedure is the same as [46] where the authors have also used the *Festival Toolkit* for producing audio components. The training and testing set sizes are 50,000 and 15,000 sequences, respectively. In this case, MNIST has already a pre-defined training and testing sets. Thus, this dataset retains division of the original MNIST. In other words, the digits present in the training set of MNIST, are only used for the training set of this dataset. The same decision is applied to the testing set.

Letter Recognition This dataset is generated following a similar process as the digit recognition dataset. The number of classes is 27 lower case letters. The number of elements in the sequence is randomly selected between three and eight elements. Visual representations are artificially produced by sequences of printed texts. The

¹The toolkit can build synthetic voices based on phone sets, word pronunciation, intonation. For more information, please visit the following link <http://festvox.org/bsv/>.

audio representation is also generated using the *Festival Toolkit*. Since this novel dataset does not have a pre-defined split between training and testing datasets, the size of the dataset is 60,000 sequences.

Word Recognition The last multi-modal dataset is partially generated using *GRID audio visual sentence corpus* [86]. The GRID corpus is employed for learning the alignment between the audio and the movements of lips. The dataset is composed of a small vocabulary of 52 words. In this work, only audio components are used, which is featured by 34 subjects: 18 males and 16 females. Similar to the letter recognition task, the visual representation is created with text lines of printed texts. The size of this dataset is 34,000 sequences.

Each of these multi-modal datasets has a different quality. For example, the visual component of the digit recognition dataset is more complicated regarding the sample variety that is presented in MNIST, whereas printed texts have less variability. On the other hand, the audio component of the word recognition dataset comprises a rich set of speakers because of the GRID corpus. Note that the reported results are the mean of a ten-cross validation, where 10,000 samples² are randomly selected for training. Furthermore, randomly 3,000 sequences³ were selected the testing set. In the last dataset, the selection is made by 50% male voices and 50% female voices for both the training and testing datasets. As a result, the sizes of training and testing sets are 17,000 samples. Figure 5.7 shows several examples of the three multi-modal datasets.

5.3.2 Input Features and LSTM setup

In the multi-modal scenario, the visual and audio elements are in two different input feature spaces. The visual components are represented using the raw pixels normalized between 0.0 and 1.0. The audio components are transformed to Mel-Frequency Cepstral Coefficient (MFCC). In more detail, MFCC is a Fourier-transformation based on filter banks with 40 coefficients (plus energy), which are distributed on a mel-scale, augmented with the first and second derivatives. The size of the audio feature vector is 123. The audio component is normalized to mean zero and variance one. The baseline is similar to the mono-modal scenario, in which the standard LSTM is trained on each input set.

²from 50,000 samples in the first dataset and 60,000 samples in the second dataset

³from 15,000 samples in the first dataset, and 50,000 samples after selecting the training set in the second dataset

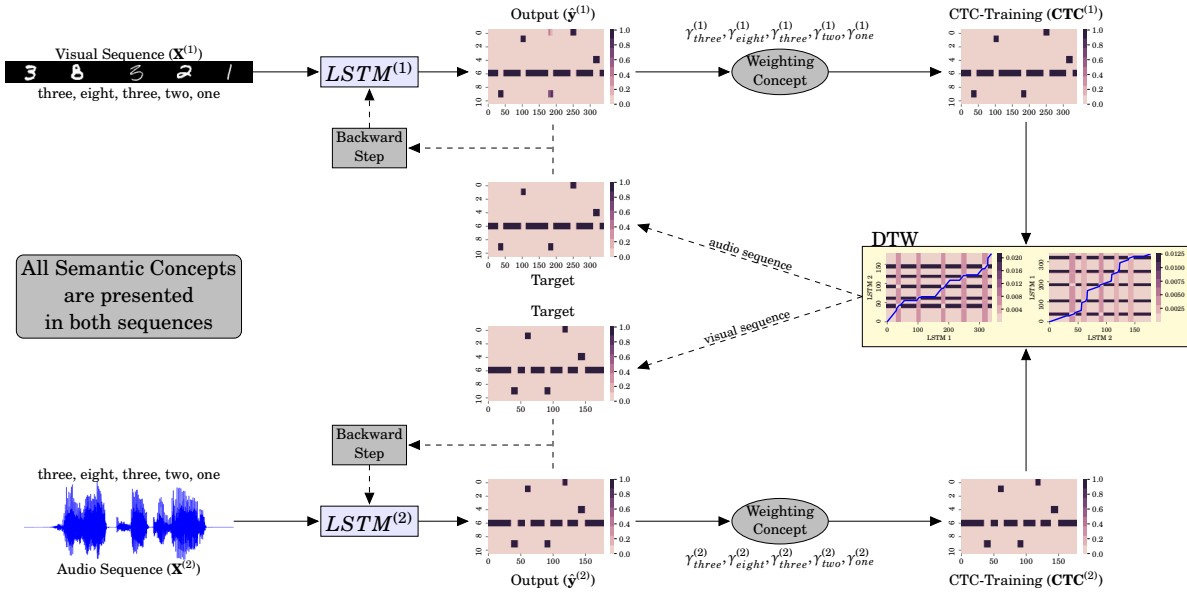


FIGURE 5.8. Example of the symbolic association in the multi-modal scenario. Note that the top LSTM receives a visual sequence and is trained based on the audio sequence, and vice versa. Therefore, a transfer domain occurs via alignment latent spaces.

For example, one LSTM network is trained only on visual elements. The parameters of LSTM (visual components) are: the size of the memory cells is 20 for the first two datasets and 40 for the last dataset, the learning rate of the network is 0.00001, and the momentum is 0.9. The parameters of the other LSTM (audio component) are: memory cell size is 100 for all three datasets, and the learning rate and momentum are the same as the visual elements. The weighting vectors are initialized to 1.0, and the learning rate is set to 0.01 for both networks.

5.3.3 Multi-modal Latent Space produced by two LSTMs

The last scenario is two LSTM networks, which are trained with combined latent spaces that are generated from two different input types. For example, one LSTM receives the visual element as input and is aligned to another audio latent space generated by the other LSTM. Figure 5.8 shows an example of the symbolic association in the multi-modal scenario. Table 5.3 shows that the proposed model reaches a similar performance to that of an LSTM network, which is trained with the traditional approach. The errors of two datasets (Letter and Words recognition) have been slightly increased. This performance

TABLE 5.3. Sequence Association Accuracy (%) and Label Error Rate (%) of the standard LSTM and the symbolic association model in the multi-modal scenario. Similar to the mono-modal scenario, the performances are also similar.

Dataset	Model	SeqAAcc (%)	LER (%)	
			Sequence 1	Sequence 2
Digits	Standard LSTM	96.84 ± 0.69	3.42 ± 0.84	0.08 ± 0.06
	Association Model	97.28 ± 0.57	2.69 ± 0.55	0.15 ± 0.08
Letters	Standard LSTM	99.34 ± 0.12	0.09 ± 0.05	1.06 ± 0.14
	Association Model	98.65 ± 0.65	0.35 ± 0.33	1.24 ± 0.50
Words	Standard LSTM	97.30 ± 0.48	0.45 ± 0.68	3.68 ± 0.27
	Association Model	96.02 ± 0.91	0.51 ± 0.84	3.77 ± 0.40

is expected taking into considerations that a different latent space is used. However, it is observed in the first dataset that the error in visual components is decreased.

Figure 5.9 shows an example of several stages of the training process. The visual element has approximately 300 vectors, whereas the audio component has 150 vectors. Note that the first row at each step represents the LSTM trained in the visual elements and the other row is the audio element. The first element shows that both networks try to agree to the blank class since this element helps to align the non-class elements. The DTW matrices show a weakly alignment between both networks since there are not learned semantic concepts. After 5,000 sequences, one of the networks (audio LSTM) has already learned the input sequence, whereas the other network has started learning some elements (for instance, element around time step 250). From 1,000 to 5,000 sequences, the CTC layer has converged to a multi-modal latent space where the visual and audio samples are combined. The last iteration (20,000 sequences) shows that both networks have already learned the multi-modal sample. The DTW cost matrix show the alignment path between both networks given the current input sample.

Figure 5.10 shows several examples of predicted sequences with their respective DTW cost matrix, which are correct (full line) and incorrect (dashed line) scenarios. In this case, incorrect means that not all elements are correctly classified. Note that in all cases, there are elements that agree in both networks. For example, the last row shows that both LSTM networks agree on the same first element of the predicted sequence

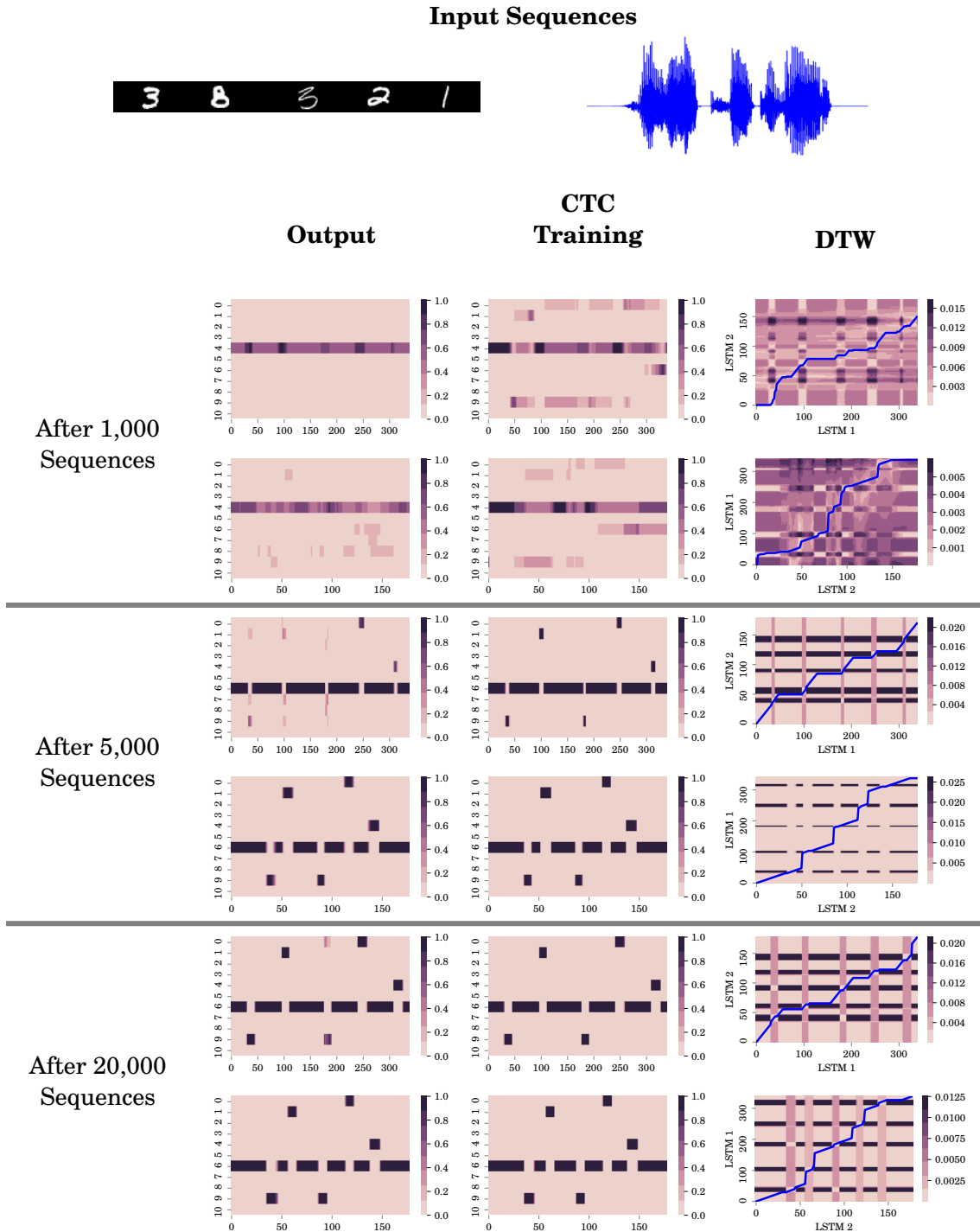


FIGURE 5.9. Example of the learning behavior in the multi-modal latent space. The first element, which both LSTMs agree is the blank class. The last two iterations show how the semantic concepts are slowly learned. Note that audio LSTM learns all elements before the visual LSTM.

even both LSTM predict incorrectly.

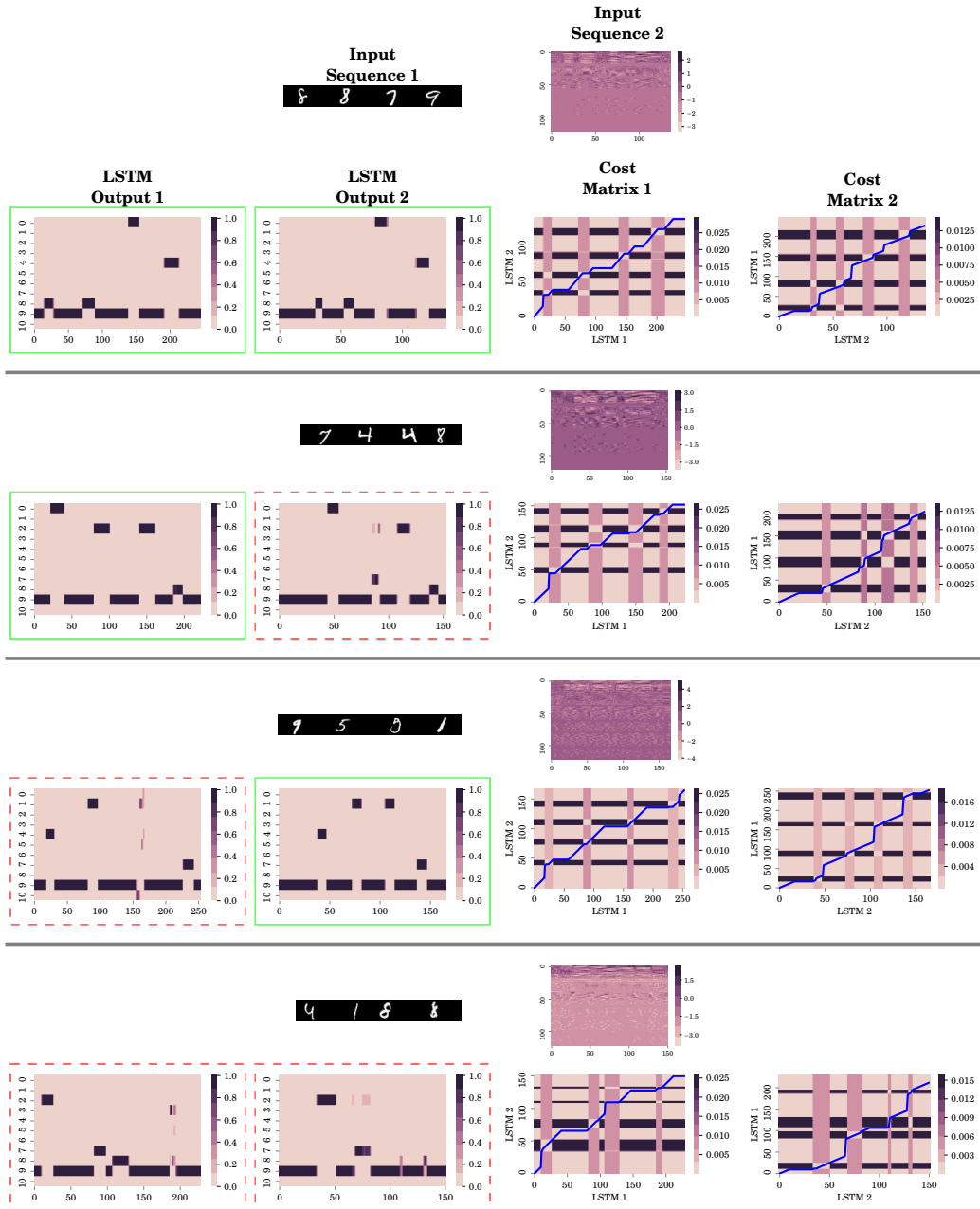


FIGURE 5.10. Several examples of the prediction in the multi-modal scenario. The first row shows two predictions that are correct (green solid square). Also, both results agree on the same vectorial representation. The remaining results show case that the predictions are partially wrong (red dashed square).

5.4 Summary

This chapter has described the evaluation of the second version of the symbolic association framework. Furthermore, two scenarios have been shown: a mono-modal dataset containing pairs of text lines of digits and three multi-modal datasets with text line-audio pairs for digits, letter, and words. Several findings are described below:

- The symbolic association framework based on LSTM networks has performances that are similar to those of LSTM trained on one input set. This outcome is supported in two setups: mono- and multi-modal association. Note that the association is not predefined before training in the symbolic association model, which this condition hurts a little bit the performance of the model.
- The mono- and multi-modal latent spaces allow each LSTM to be trained in different modalities. Hence, there is not a requirement to define metrics between modalities because each LSTM works as a proxy between the data domain to a common latent space.
- Similar to the findings in Chapter 4, the association model works because the convergence of the semantic concepts. In this case, the blank class is the first element to which each internal LSTM network agrees.

One of the limitations occurs if the standard trained LSTM classifies an input sample correctly, whereas the presented model has failed. This result suggests that some information is missing in the latent space generated from the other sequence. Similarly, there are some cases in which the presented model correctly classifies input samples, and the standard trained LSTM predicts incorrectly. This scenario has been shown in the multi-modal digit recognition task. In other words, some information decoded in the audio latent space helps to improve the performance of the visual network. As a future work, the alignment step can be improved for selecting better options. For example, it can be useful that the alignment uses some type of transformations, such as maximum or mean in the time axis.

Another limitation of the model is the current assumption that each element of the sequence is always presented in both sequences. One step further, the model could handle sequences that might be presented in one or both sequences. An extension of the association framework is described in the next chapter.

ASSOCIATION LEARNING IN SEQUENCES WITH MISSING CONCEPTS

This chapter describes an extension of the LSTM-based approach, in which the semantic concepts can appear in one or both modalities. For example, the sentence "one two three" can be associated with only two elements in the sentence "two four three". The experimental design in this chapter evaluates two cases. The first case compares to both the model presented in this chapter that can handle missing elements in multi-modal sequences and two baselines: an LSTM network trained on one modality and the association model presented in Chapter 5. The second case evaluates the effect of the number of missing elements and the importance of the modality. Both cases show that the presented extension reaches better results than model described in Chapter 5 and similar performances to LSTM networks trained only on one modality.

The presented extension and results are based on a published version in the Journal of Artificial Intelligence Research (JAIR) [87]. This chapter is divided into four sections. Section 6.1 explains the new association task where multi-modal sequences (visual, audio) has semantic concepts in one or two channels. Note that this task follows the same idea in Chapter 5 of using *weakly labeled annotation*. Section 6.2 describes the process for handling missing elements, in which relies on using split semantic concepts that are presented in one (missing elements) or both modalities (common elements). Section 6.3 presents three multi-modal setups: missing elements in both channels, missing elements

in the audio channel, and missing elements in the visual channel. Additionally, the network setup of each LSTM network is described. Section 6.4 discusses the results of the proposed extension and compares its performance to the original model and LSTM networks trained on one modality.

6.1 Problem Definition

The new association scenario still applied on multi-modal sequences where each semantic concept might be or might not be in both channels. Therefore, the link between both modalities is partial instead of complete as the previous chapter. Again, two sequences are defined $\mathbf{X}^{(1)} = \{\mathbf{x}_1^{(1)}, \dots, \mathbf{x}_{t_1}^{(1)}\}$ and $\mathbf{X}^{(2)} = \{\mathbf{x}_1^{(2)}, \dots, \mathbf{x}_{t_2}^{(2)}\}$. Additionally, each sequence has a set of semantic concepts $C^{(1)} = \{c_1, \dots, c_{d_1}\}$ and $C^{(2)} = \{c_1, \dots, c_{d_2}\}$. Note that an important condition is that both sequences have at least one shared semantic concept $C^{(1)} \cap C^{(2)} \neq \{\emptyset\}$, and the order between semantic is the same.

As presented in Figure 6.1, this scenario is more challenging because the association is not *one-to-one* like the previous case. However, LSTM-based approach can still align partial sequences with weakly labels and still agree on the same vectorial representation. Similar to Chapter 5, the association model exploits the Latent Space produces by both sequences even with shared semantic concepts. In this chapter, Association Accuracy for sequences and Label Error Rate (Equations (5.1) and (5.2)) are also used.

6.2 Handling Missing Elements

The LSTM-based approach learns to pair in the DTW alignment, where both output sequences are aligned to each other. The alignment is possible because the assumption of both sequences represents the same series of semantic concepts. However, that approach is not feasible if there are missing elements in one of the sequences. Therefore, it is required to define two cases about the semantic concepts: *shared* semantic concepts in both sequences and *missing* semantic concepts in one sequence. Those sets are formulated as follows:

$$S^{share} : C^{(1)} \cap C^{(2)} = \{c_1, \dots, c_{l_1}\} \quad \text{shared semantic concepts,} \quad (6.1)$$

$$S^{(1)} : C^{(1)} \setminus C^{(2)} = \{c_1, \dots, c_{l_2}\} \quad \text{only in modality one,} \quad (6.2)$$

$$S^{(2)} : C^{(2)} \setminus C^{(1)} = \{c_1, \dots, c_{l_3}\} \quad \text{only in modality two.} \quad (6.3)$$

As mentioned in Section 3.4, DTW alignment combines the output sequences obtained by CTC step (Equations (3.30) and (3.31)). One of the outcomes is the alignment path from one sequence to another sequence, which is expressed by $p^{(1)} : \mathbf{CTC}_i^{(1)} \rightarrow \mathbf{CTC}_j^{(2)}$. Therefore, the latent space generated by *sequence 1* is used for training *sequence 2*. There is a limitation with that approach. For example, the visual sequence exploits only information obtained from the audio sequence. However, there are cases in the latent space that it is better to keep the information obtained from the visual sequence. This chapter proposes a solution for exploiting both modalities instead of one. One standard approach for combining two signals is to use a maximum operation between them that

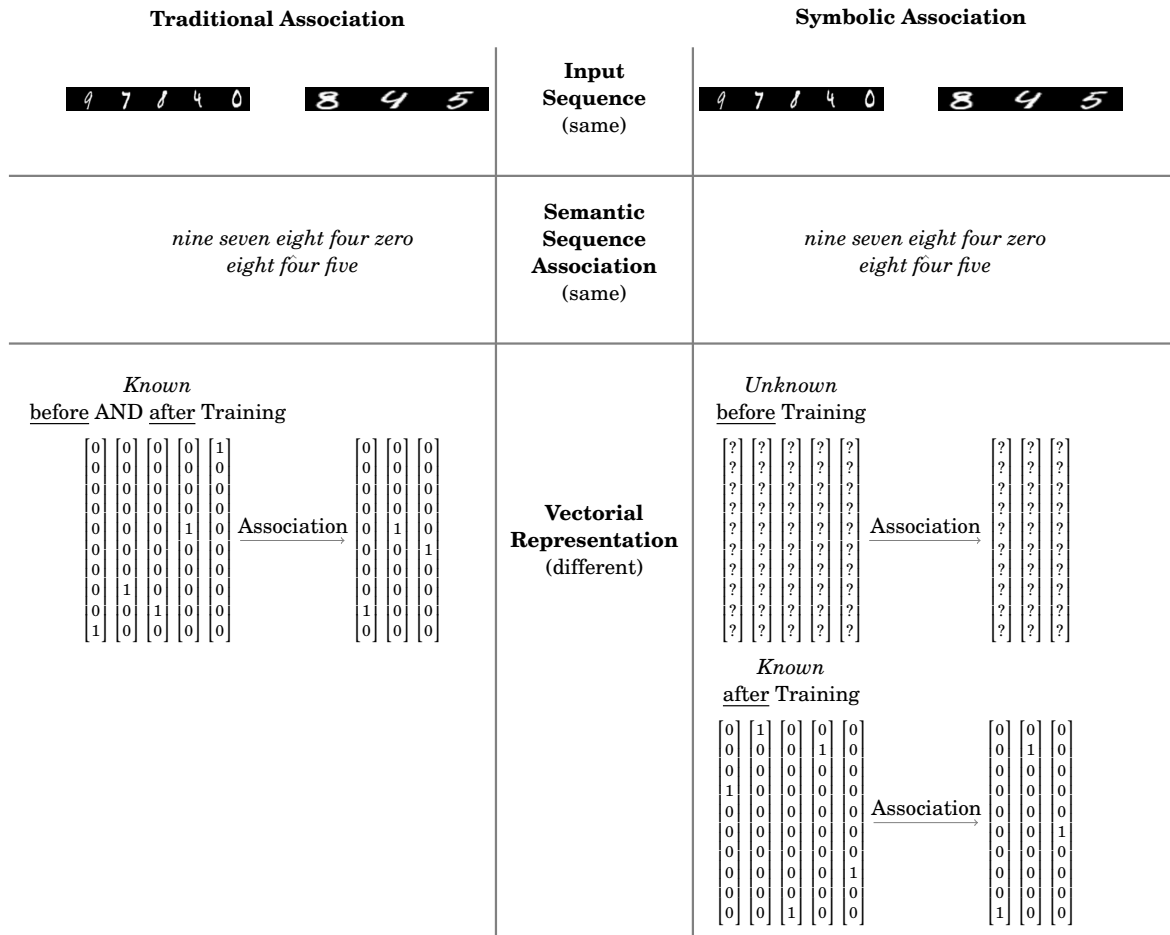


FIGURE 6.1. Association between sequences with partial alignment because some concepts (*eight* and *four*) are presented in both sequences. Note that the problem definition described in Section 5.1 has the same series of concepts in both sequences.

summarizes the best of each modality. This step can be expressed as follows:

$$\mathbf{CTC}_t^{(1,2)} = \begin{cases} \max(\hat{y}_{c,t'}^{(2)}, \hat{y}_{c,t}^{(1)}) & c \in S^{share}, \\ \hat{y}_{c,t}^{(1)} & c \in S^{(1)}, \end{cases} \quad (6.4)$$

$$\mathbf{CTC}_t^{(2,1)} = \begin{cases} \max(\hat{y}_{c,t'}^{(1)}, \hat{y}_{c,t}^{(2)}) & c \in S^{share}, \\ \hat{y}_{c,t}^{(2)} & c \in S^{(2)}, \end{cases} \quad (6.5)$$

where $\hat{y}_{c,t}^{(1)}$ and $\hat{y}_{c,t}^{(2)}$ are scalar values that represent the semantic concept c at timestep t . At this stage, the target vector is based on concatenating all semantic concepts that are in both sequences. Afterwards, both vectors $\mathbf{CTC}_t^{(1,2)}$ and $\mathbf{CTC}_t^{(2,1)}$ are converted to a probability vector¹. Therefore, Equations (3.32) and (3.33) are updated as follows:

$$J_{LSTM}^{(1)} = \hat{\mathbf{y}}_{j_1}^{(1)} - \mathbf{CTC}_{j_1}^{(1,2)} \quad j_1 = 1, \dots, t_1, \quad (6.6)$$

$$J_{LSTM}^{(2)} = \hat{\mathbf{y}}_{j_2}^{(2)} - \mathbf{CTC}_{j_2}^{(2,1)} \quad j_2 = 1, \dots, t_2. \quad (6.7)$$

Figure 6.2 shows an example of the proposed approach for handling multi-modal sequences with missing and shared semantic concepts. Note that this step combines both modalities before the target sequence. Additionally, this model can align semantic concepts with the same vectorial representation. For example, the semantic concept *té* is represented by the same unit vector \mathbf{e}_4 . Also, the non-shared semantic concepts are predicted with different vectorial representations.

6.3 Experiments

The presented approach is evaluated in two multi-modal scenarios. The first scenario is defined with a random number of missing concepts in each modality. The second scenario evaluates the effect of missing concepts in one modality. Additionally, the presented approach is compared to the original model (*c.f.* Chapter 5) and LSTM networks trained on one modality. The results of these experiments are also reported based on the *Sequence Association Accuracy (SeqAAcc)* (Equation (5.1)) and the *Label Error Rate*

¹a probability vector is obtained after applying $\text{norm}(\mathbf{y}) = \mathbf{y} / \sum_i y_i$ where $\mathbf{y} \in R^n$

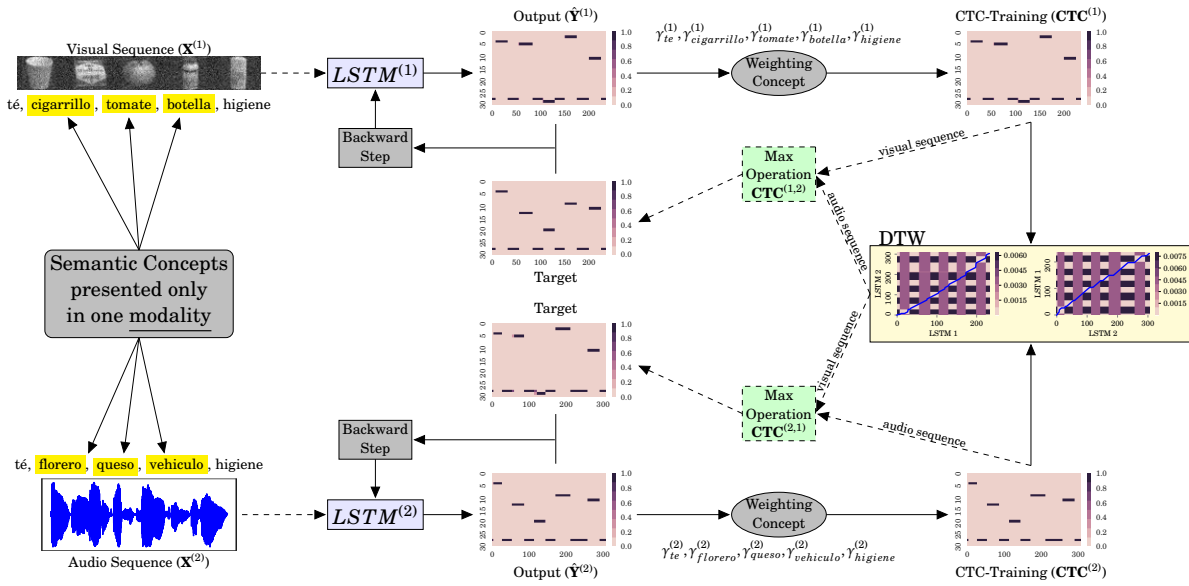


FIGURE 6.2. Example of the proposed approach for handling the new association problem: *semantic concept are presented in one or both modalities*. This approach combines two modalities based on two components (dashed lines and squares): *shared* semantic concepts and a maximum operation. The model can still learn the association with this partial alignment, e.g. concepts *té* and *higiene* agree on the same vectorial representation.

(*LER*) (Equation (5.2)). This section is divided into two parts. The first part explains the process for generating three multi-modal setups. One of the setups generates multi-modal sequences with a random number of missing concepts on each modality. For example, the sequences "*one two three four*" and "*two four six seven*" has the following missing concepts. The missing concepts of the first sequence are "*six*" and "*seven*" based on the second sequence that is used as a reference. The same analysis is applied to the sequence two, in which the missing concepts are "*one*" and "*three*". The other two setups generate multi-modals sequences where the concepts of one modality are a subset of the other modality. For example, the sequences "*one two three four*" and "*two four*" has the following missing concepts. The first sequence does not have any missing concept considering the second sequence as a reference. In contrast, the second sequence has "*one*" and "*three*" as missing concepts considering the first sequence as a reference.

6.3.1 Multi-modal Setups

The procedure of generating each setup follows similar steps described in Section 5.3. As a reminder, note that the multi-modal sequences have two components: visual and audio. The visual component is a horizontal arrange of objects (similar to a *panorama*). The second component is a set of Spanish words whereas the experiments in the previous chapter are based on English words. Three steps have the process of generating the multi-media datasets: generating semantic concepts for each modality, generating the visual component based on the semantic concepts, and generating the audio component based on the semantic concepts. Each of these steps is explained as follows:

Generating Semantic Sequences: As mentioned before three setups are considered, the first setup has missing concepts in both modalities (setup 1: *missing both*), and the other setups have missing concepts only in one modality (setup 2: *missing visual* and setup 3: *missing audio*, respectively). All setups start with a series of ten semantic concepts that are randomly selected without repetition. This initial series is used for representing the concepts for each modality. For the first setup, the next step removes between zero and fives concepts from the initial ten concepts on each modality. Therefore, both sequences have *shared* and *missing* concepts (i.e. $S^{share} \neq \{\emptyset\}$, $S^{(1)} \neq \{\emptyset\}$, and $S^{(2)} \neq \{\emptyset\}$). For the second and third setups, the next step removes a fixed number of concepts from one modality. For instance, the visual channel has ten concepts whereas the audio channel only has eight concepts. Hence, there are *missing* concepts in only one modality (i.e., $S^{share} \neq \{\emptyset\}$, $S^{(1)} \neq \{\emptyset\}$, and $|S^{(2)}| < |S^{(1)}|$). The set of semantic concepts used for all setups are 30 nouns in Spanish: *oso, bote, botella, bol, caja, carro, gato, queso, cigarrillo, gaseosa, bebida, pato, cara, comida, hamburguesa, higiene, líquido, loción, cebolla, pimentón, pera, redondo, sánduche, cuchara, té, teléfono, tomate, florero, vehículo, madera*.

Generating Visual Components: The visual component of the multi-modal sequence is based on the semantic concepts generated in the first step. Furthermore, this channel is constrained to a horizontal arrangement of objects, i.e., *panorama*. Therefore, COIL-100 dataset [88] offers a set of 100 isolated objects with a black background. Moreover, each object has 72 color images that show the object at different angles (five degrees apart). In this work, all images are converted to grayscale and resize to 32x32 pixels. Also, some objects were filtered out because of very similar to another object, same predicted category using a pre-trained network. Thus, a subset of 30 objects is used in this step.

Each concept in the sequence selects randomly one of the images of the selected concepts. Afterwards, all images are horizontally stacked for generating panoramas. A random noise is added to the panoramas as background (i.e. $new_img = panorama + noise$).

Generating Audio Components: Similar to the visual component, the audio sequence is generated based on the concepts that are presented. Therefore, an audio dataset was collected based on the presented vocabulary. Twelve subjects from several countries of Center and South America recorded two times each concept using *Audacity (R) recording and editing software*². Each concept presented in this modality selects one audio file from two recorded ones. Similar to the visual channel, all selected audio are concatenating.

Training and Testing Multi-modal Datasets: The visual and audio components have different splits for training and testing. There are 1,000 multi-modal sequences per subjects. In more detail, the visual and audio components have a different division for training and testing sets. For the visual component, images at odd angles are used for training whereas at even angles are used for testing. For the audio component, eleven subjects are randomly selected for training, and the remaining subjects are used for testing. The experiment follows a 5 cross-validation approach, in which the subjects of the audio component are randomly selected for each fold. Figure 6.3 shows an example of the first multi-modal configuration, in which both sequences have missing concepts.

6.3.2 Input Features and LSTM setup

This chapter follows a similar feature extraction process described in Section 5.3. The visual component is rescaled between 0 and 1. The audio component is converted to MFCC using HTK. Thus, the audio files are represented by a vector with 123 components based on filter banks and expanded with the first and second derivatives. Both components are normalized to mean zero and standard deviation one.

The presented approach is evaluated against two baselines: symbolic multi-modal association (Chapter 5) and LSTM trained on one modality. These three architectures are using the same parameters for evaluating LSTM performances. The visual component is training with a bidirectional LSTM network with 40 memory cells, and the learning rate is 0.0001 with momentum 0.9. Another bidirectional LSTM network is trained on the

²Audacity® software is copyright ©1999-2017 Audacity Team. The name Audacity® is a registered trademark of Dominic Mazzoni.

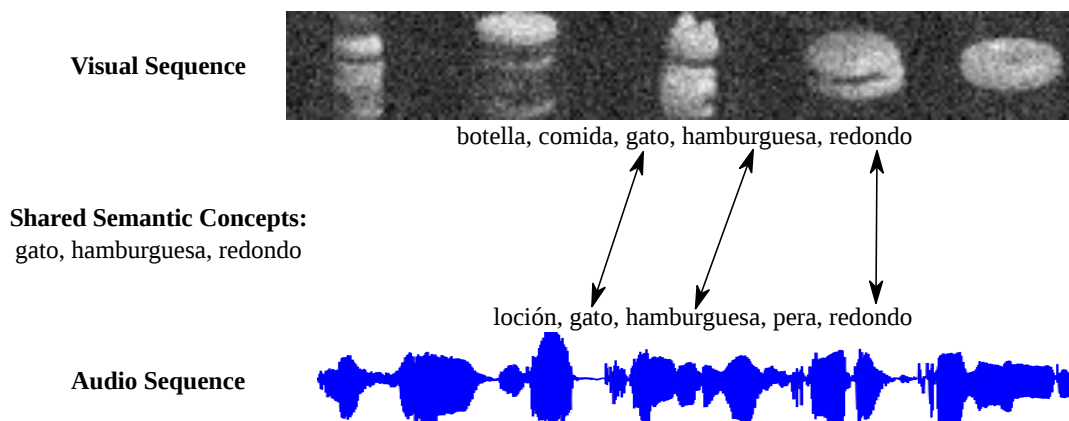


FIGURE 6.3. Example of the multi-modal configuration with missing elements. In this example, the shared concepts are (S^{shared}): *gato*, *hamburguesa*, and *redondo*. This partial alignment is useful for combining the latent spaces of both modalities. Additionally, the non-shared concept in the visual sequence is ($S^{(1)}$) *botella*, whereas the other modality ($S^{(2)}$) presents *loción* and *pera* as non-shared concepts.

audio component with 100 memory cells, and the learning rate and momentum are the same to the other LSTM. The learning rates of the weighting concepts are set 0.001 for both the presented approach and the original model.

6.4 Results and Discussion

This section reports the average of 5-folds for each multi-modal configuration. The first multi-modal setup evaluates the presented model with a random set of missing elements in each sequence. The other setups focus on the relation between the number of missing concepts and the performance. Note that the training set has 11,000 sequences whereas the testing set has 2,000 sequences.

The first results are obtained from the first multi-modal setup: missing concepts in both modalities. The presented approach is compared to two baselines: LSTM trained on each modality independently and the original model described in Chapter 5. As can be seen in Table 6.1, the presented approach reaches better results (*SeqAAcc* and *LER*) than the original model. This outcome is expected because the initial assumption of the original mode is both modalities have the same series of semantic concepts (i.e., $S^{(1)} = S^{(2)}$). Additionally, the presented model reaches also similar results of *AAcc* than

TABLE 6.1. Sequence Association Accuracy (%) and Label Error Rate (%) from the multi-modal configuration of missing concepts in both modalities. The presented approach reaches better results than the original model. The more interesting outcome is that the presented approach reaches a lower error than the baseline in the audio sequences. It can be inferred that the combination of visual and audio sequences helps to reduce the error. However, the presented model reaches higher error than the baseline in the visual sequences.

Model	SeqAAcc (%)	LER (%)	
		visual	audio
LSTM + CTC (baseline)	70.68 ± 6.12	0.14 ± 0.14	35.84 ± 5.35
Original Model ([70], Chapter 5)	19.59 ± 8.90	7.00 ± 2.42	79.01 ± 9.51
Model (missing concepts)	71.52 ± 11.85	0.97 ± 1.58	33.09 ± 10.38

the LSTM networks trained only on one modality. It is interesting to see that the presented approach reaches lower *LER* than LSTM in the audio sequence because of the combination between visual and audio latent spaces. However, the same combination hurts the performance of the presented approach in the visual sequences.

The performance of the presented approach did not decrease with partial alignment between modalities. This model can still learn to agree on the same vectorial representation as mentioned before. Figure 6.4 shows two examples of the vectorial representation agreement between shared concepts. For example, the concepts *madera*, *carro*, and *loción* in the first row agree on the same vectorial representations \mathbf{e}_9 , \mathbf{e}_{24} , and \mathbf{e}_{25} in both modalities³. The second row shows another example where the visual sequence is correctly predicted whereas the audio sequences are not. In that case, the concept *loción* has the same representation \mathbf{e}_{25} in both networks and agree with the prediction in the first row.

The second and third multi-modal setups are based on several datasets, which a fixed number of missing elements are extracted from one component of all multi-modal sequences. Thus, the robustness of the model is analyzed concerning number of missing concepts. This scenario runs between *zero* and *five* missing concepts. Note that *zero* missing concept is the initial assumption of the original model, which shows the effect of the *max operation*. Figure 6.5 shows that the *max operation* improves *SeqAAcc* of the

³Sometimes, one concept is represented by two vectorial representations, but both networks can retrieve the correct concept.

CHAPTER 6. ASSOCIATION LEARNING IN SEQUENCES WITH MISSING CONCEPTS

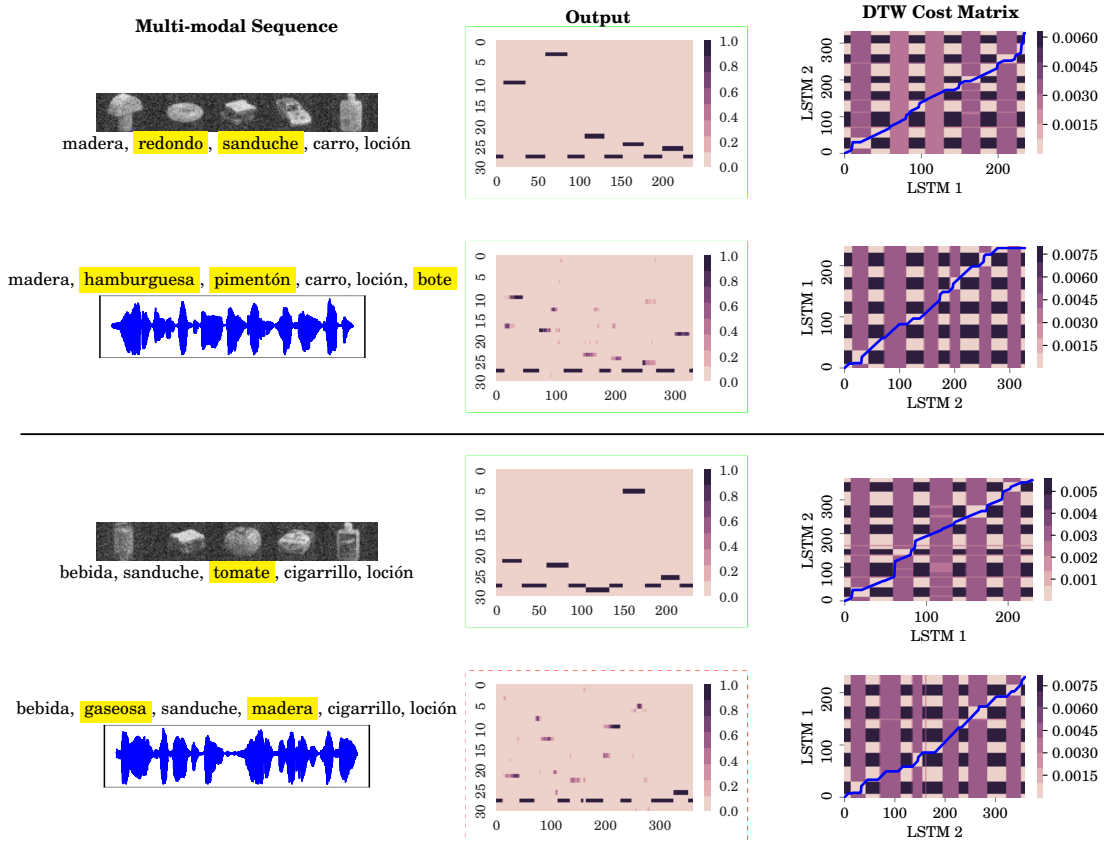


FIGURE 6.4. Several examples of the output classification and DTW cost matrices are shown. The first multi-modal sequence shows an example, in which both output classifications are correct (solid green square). The second multi-modal sequence shows one correct and one incorrect output classification (dashed red square).

presented approach in all missing concept setups and modalities. The original model is negatively affected by increasing the number of missing concepts whereas the presented approach is more robust against that factor. Furthermore, Figure 6.6 presents *LER* of each modality and shows a similar pattern to Figure 6.5. The left figure shows that the original model increases the error of the audio component based on increasing the number of missing concepts in the image components. In contrast, the presented approach keeps the same performance regardless of the missing concepts (up to 50% missing concepts). It can be observed a similar pattern on the right figure.

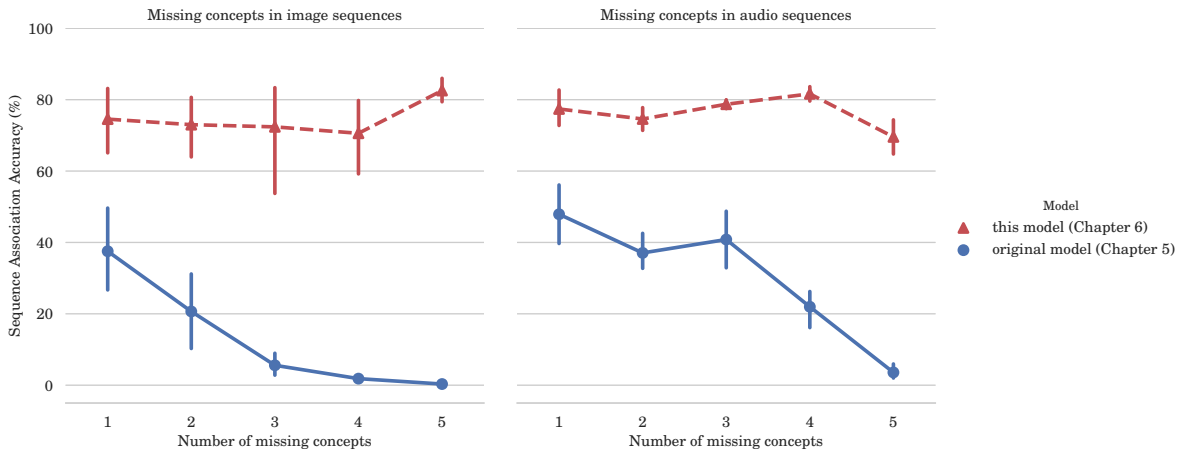


FIGURE 6.5. Sequence Association Accuracy (%) of second and third multi-modal configurations with several missing concepts. The performance of the presented approach (triangle) is better than the original model (circle) in general (i.e., modality, number of missing elements). Note that the *max operation* has a positive effect on the performance.

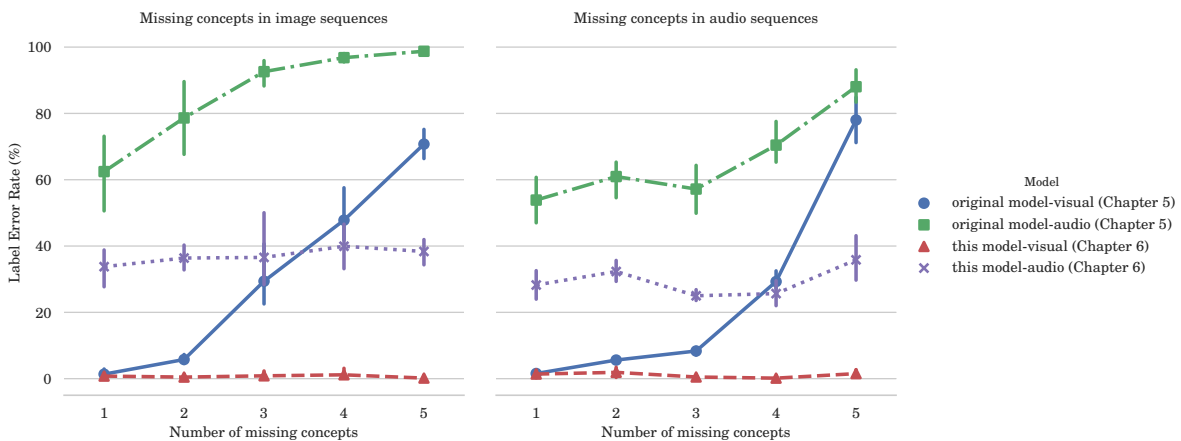


FIGURE 6.6. Label Error Rate (%) of the second and third multi-modal configurations with several missing concepts. The performance of presented approach (visual-triangle and audio-X) is similar in all missing concepts. On the other hand, the performance of the original model (visual-circle and audio-square) is hurt by the number of missing concepts.

6.5 Summary

This chapter describes an extension that handles missing concepts in one or both modalities. The extension relies on partial alignments based on shared concepts in the sequence and a *max operation* for combining. Several findings are described below:

- The presented approach reaches better results than the original model and similar to LSTM. The partial alignment and the *max operation* prove to have a positive effect for handling missing concepts, which is evaluated in three multi-modal configurations.
- Similar to Chapter 5, the presented approach relies on the convergence of the shared concepts, even with a partial alignment instead of a complete alignment.

One of the limitations of the presented approach is the alignment between panoramas and audio, where the relationship is only one dimension. The next step is to align objects that are presented in a two-dimensional image and an audio signal. Moreover, a two-dimensional HMM has already applied to image classification [89]. With this in mind, a two-dimensional HMM can be combined with LSTM networks for sequence classification in images as future work.

CLASSLESS ASSOCIATION

In this chapter, the association of isolated elements relies on a more difficult scenario. The goal is to associate two elements without specifying semantic classes. The proposed model uses a statistical distribution as a target. Similar to the model discussed in Chapter 3, the training algorithm follows an EM-approach for learning the agreement between two NNs. The model is compared to two cases: The first case is a supervised classification, which is implemented by an MLP, and the second case is a traditional unsupervised classification that is evaluated with two clustering algorithms: K-means and Hierarchical Clustering. The performance of the model has reached better purity than both unsupervised algorithms (lower baseline). Moreover, the performance concerning supervised case is comparable regarding the lack of semantic concepts and a weakly loss function.

The proposed architecture and results have been presented in ICANN2017 [90]. This chapter is divided into three sections. Section 7.1 describes the association task, in which the semantic concepts are not available. Section 7.2 presents the novel model called classless association. Section 7.3 describes the experimental setup and the performance of the classless association model.





Task	Input Samples (same)	Abstract Concept Association (different)	Coding Scheme for each input (different)	Classifiers (same)
Supervised Association		"three"	$\begin{matrix} \text{Known before and after} \\ \text{Training} \\ \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \end{matrix}$	<div style="border: 1px solid gray; padding: 2px; width: fit-content; margin-bottom: 10px;">classifier 1</div> <div style="border: 1px solid gray; padding: 2px; width: fit-content;">classifier 2</div>
		"three"		
Classless Association		"unknown"	$\begin{matrix} \text{Unknown before} & \text{Known after} \\ \text{Training} & \text{Training} \\ \begin{matrix} ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{matrix} \end{matrix}$	<div style="border: 1px solid gray; padding: 2px; width: fit-content; margin-bottom: 10px;">classifier 1</div> <div style="border: 1px solid gray; padding: 2px; width: fit-content;">classifier 2</div>
		"unknown"		

FIGURE 7.1. Problem definition of the classless association. It can be observed that two different instances represent the same semantic concept. After the model is trained, both classifiers predict sample pairs by the same index.

7.1 Problem Definition

This section introduces a new association problem. In this scenario, the goal is to associate two instances of the same unknown semantic concept. This task is a more challenging scenario than the association problem described in Chapter 4. Figure 7.1 shows a comparison between two scenarios of the association task: supervised and classless association. Note that the class-less association does require an alternative cost function without labeled data.

In this case, two metrics are used: *Association Accuracy* (AAcc) and *Purity*. The first metric has already been defined in Equations (4.1) and (4.2). This metric only measures how many samples are classified by the same *index*. However, this metric does not show if the networks have learned any semantic concept. The second metric complement the first metric for quantifying the classification. *Purity* is a standard metric for evaluating the quality of two sets: output prediction $\hat{\mathbf{Y}}$ and the ground-truth label \mathbf{Y} as follows:

$$\text{Purity}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^k \max_j |\hat{\mathbf{y}}_i \cap \mathbf{y}_j|, \quad (7.1)$$

where N is the number of elements in the dataset. Formally, two disjoint sets $\mathbf{x}^{(1)} \in R^{n_1}$ and $\mathbf{x}^{(2)} \in R^{n_2}$ represent the association task with the condition that both instances express the same unknown class c . Additionally, there is no sharing information between samples, for instance samples $(\mathbf{x}_1^{(1)}, \mathbf{x}_1^{(2)})$ and $(\mathbf{x}_2^{(1)}, \mathbf{x}_2^{(2)})$ may or may not represent the same category, which is unknown by the model. Some approaches use similarities and dissimilarities for training [91] but this classless association problem is not constrained to those conditions.

7.2 Model

The presented model uses a statistical distribution as an alternative loss function presented in Chapter 2 where there is no requirement of labeled data. Similar to Chapter 4, two parallel MLP networks implement the model. Additionally, the model follows an EM-approach for training. The *E-step* predicts the distribution of the raw output vectors. The *M-step* updates the parameters of the statistical distribution and the network parameters. Furthermore, the model is defined by the following equations:

$$\hat{\mathbf{y}}_i^{(1)} = \text{net}^{(1)}(\mathbf{x}_i^{(1)}, \theta^{(1)}), \quad (7.2)$$

$$\hat{\mathbf{y}}_i^{(2)} = \text{net}^{(2)}(\mathbf{x}_i^{(2)}, \theta^{(2)}), \quad (7.3)$$

where $\text{net}^{(1)}$ and $\text{net}^{(2)}$ are two MLPs with parameters $\theta^{(1)}$ and $\theta^{(2)}$ (respectively). Moreover, another parameter is the desired statistical distribution $\phi \in R^k$, where k is the output size. As a reminder, the goal is to approximate the raw output vectors of a network and a statistical distribution.

The *EM-approach* starts setting each sample $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ with a random index between 1 and k . The indices have the desired target distribution of ϕ . For example, a dataset with ten samples and statistical distribution defined by $\phi = [0.7, 0.3]^T$ can be represented by seven samples with the index 1 and three samples with the index 2.

The *E-step* passed a mini-batch of samples to each network. Then, a post-processing step is applied where an approximation of the current distribution based on raw output vectors is obtained as follows:

$$\hat{\mathbf{z}}^{(1)} = \frac{1}{m} \sum_{i=1}^m \text{power}(\hat{\mathbf{y}}_i^{(1)}, \gamma^{(1)}), \quad (7.4)$$

$$\hat{\mathbf{z}}^{(2)} = \frac{1}{m} \sum_{i=1}^m \text{power}(\hat{\mathbf{y}}_i^{(2)}, \gamma^{(2)}), \quad (7.5)$$

where $\gamma^{(1)}$ and $\gamma^{(2)} \in R^k$ are the parameters that guide the network to learn the statistical distribution, and m is the size of the mini-batch. Note that the term γ described in Chapter 3 is used for learning the relationship between semantic concepts and vectorial representations. Here, the goal is to determine the vectorial representation, and each network learns to group similar input samples by itself.

The *M-step* updates two modules: the terms $\gamma^{(1)}$ and $\gamma^{(2)}$ based on matching output vectors and a statistical distribution and network parameters $\theta^{(1)}$ and $\theta^{(2)}$ based on the current indexes that represent *pseudo-classes*. The first module is updated via variance between the current distribution of the raw vectors and the desired target, which are defined by:

$$J_{\gamma}^{(1)} = \left(\hat{\mathbf{z}}^{(1)} - \phi \right)^2, \quad (7.6)$$

$$J_{\gamma}^{(2)} = \left(\hat{\mathbf{z}}^{(2)} - \phi \right)^2, \quad (7.7)$$

where ϕ is the target distribution. Figure 7.2 shows an example about the novel loss function based on statistical distributions. Gradient descent updates these parameters as shown by the equations:

$$\gamma_{new}^{(1)} = \gamma_{old}^{(1)} - \alpha \frac{\partial J_{\gamma}^{(1)}}{\partial \gamma}, \quad (7.8)$$

$$\gamma_{new}^{(2)} = \gamma_{old}^{(2)} - \alpha \frac{\partial J_{\gamma}^{(2)}}{\partial \gamma}, \quad (7.9)$$

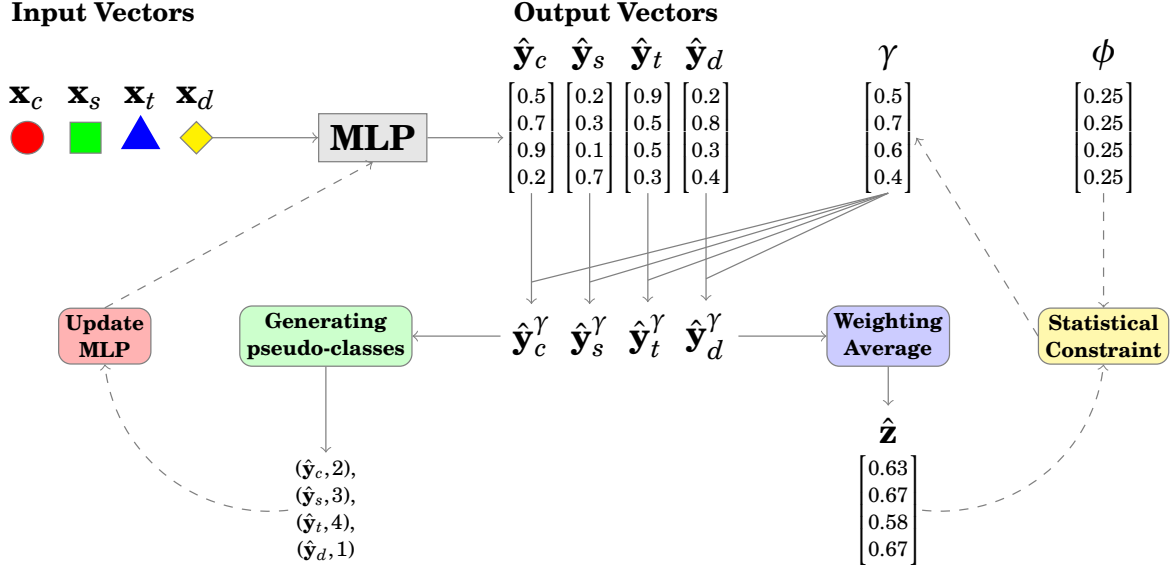


Figure 7.2: Loss function based on a statistical distribution. The *E-step* (solid lines) passes forwarded the input samples. In this manner, an estimation of the current statistical distribution of the output vectors. Afterwards, the *M-step* (dashed lines) updates the weighting vectors based on the new loss function.

where $\partial J_{\gamma}^{(1)}/\partial \gamma$ and $\partial J_{\gamma}^{(2)}/\partial \gamma$ are the derivatives w.r.t γ and α is the learning rate. Second, the network parameters are updated based on the following condition: one network used the indexes generated by the other network and vice versa.

$$J_{MSE}^{(1)} = \frac{1}{N} \sum_{i=1}^N \left(\hat{\mathbf{y}}_i^{(1)} - \mathbf{c}_i^{(2)} \right)^2, \quad (7.10)$$

$$J_{MSE}^{(2)} = \frac{1}{N} \sum_{i=1}^N \left(\hat{\mathbf{y}}_i^{(2)} - \mathbf{c}_i^{(1)} \right)^2, \quad (7.11)$$

where $\mathbf{c}_i^{(1)}$ and $\mathbf{c}_i^{(2)}$ are the vectorial representations of *pseudo-classes*, which are self-defined without categorical information or labels by the network. One crucial step is related to prediction or classification step, which is used for updating the indexes while the network is in training mode. The indexes are used in this model as *pseudo-classes* for the loss functions (Equations (7.10) and (7.11)). In this case, the *pseudo-classes* are updated after a number of iterations, e.g after 1,000 iterations. Therefore, another parameter is chosen when the indexes are updated by retrieving the maximum element of the following equation:

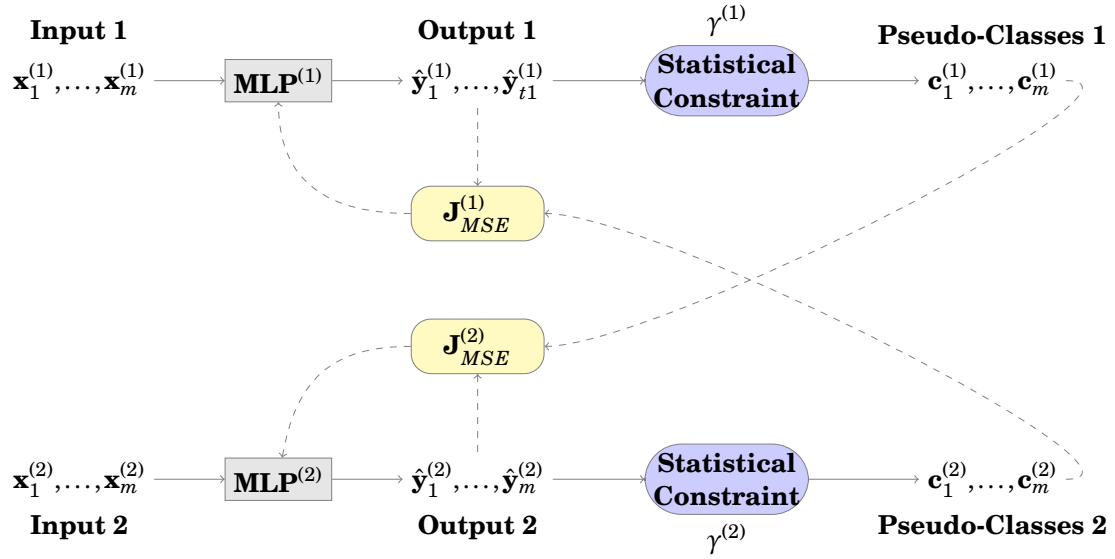


FIGURE 7.3. Overview of the classless association model. This model follows an EM-approach. The *E-step* (solid lines) passes each input sample to each MLP and classifies the samples based on the weighting vector. The *M-step* (dashed lines) updated the parameters.

$$c^{(1)} = \arg \max_k \text{power}(\hat{\mathbf{y}}^{(1)}, \gamma^{(1)}), \quad (7.12)$$

$$c^{(2)} = \arg \max_k \text{power}(\hat{\mathbf{y}}^{(2)}, \gamma^{(2)}), \quad (7.13)$$

The intuition is that similar samples are grouped with the same index or pseudo-class. Figure 7.3 shows the cross-learning between both MLPs. It can be observed that the role of the pseudo-classes is crucial because the model is self-labeling input samples during training. Algorithm 2 shows the pseudo-code of the presented training algorithm in terms of *E*- and *M*-steps.

Algorithm 2 Pseudocode of the Classless Association Training based on matching network output against a statistical distribution.

Require: mini-batch size m , update_classes, learning_rates, $\gamma^{(1)}$, $\gamma^{(2)}$, ϕ
 {Random Initialization of input and pseudo-classes ($\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $c^{(1)}$, $c^{(2)}$)}

for each_epoch=1 TO max_epoch **do**
 {E-STEP}
for i=1 TO M **do**
 {Equations (7.2) and (7.3)}
 $\hat{\mathbf{y}}_i^{(1)} \leftarrow \text{forward_step}(MLP^{(1)}, \mathbf{x}_i^{(1)})$
 $\hat{\mathbf{y}}_i^{(2)} \leftarrow \text{forward_step}(MLP^{(2)}, \mathbf{x}_i^{(2)})$
end for

$\hat{\mathbf{z}}^{(1)} \leftarrow \frac{1}{M} \sum_{i=1}^M \text{power}(\hat{\mathbf{y}}_i^{(1)}, \gamma^{(1)})$
 $\hat{\mathbf{z}}^{(2)} \leftarrow \frac{1}{M} \sum_{i=1}^M \text{power}(\hat{\mathbf{y}}_i^{(2)}, \gamma^{(2)})$

{M-Step}
for i=1 TO M **do**
 { $MLP^{(1)}$ is learning from $MLP^{(2)}$, and vice versa}
 $\text{accumulate_gradient_error}(MLP^{(1)}, \hat{\mathbf{y}}_i^{(1)}, c_i^{(2)})$
 $\text{accumulate_gradient_error}(MLP^{(2)}, \hat{\mathbf{y}}_i^{(2)}, c_i^{(1)})$
end for
 $\text{backward_step}(MLP^{(1)}, C^{(2)})$
 $\text{backward_step}(MLP^{(2)}, C^{(1)})$
 {Equations (7.6) to (7.9)}
 $\text{update_weighting_vector}(\hat{\mathbf{z}}^{(1)}, \gamma^{(1)}, \phi)$
 $\text{update_weighting_vector}(\hat{\mathbf{z}}^{(2)}, \gamma^{(2)}, \phi)$

if each_epoch != 1 and each_epoch mod update_classes == 0 **then**
 {Prediction step: generating new pseudo-classes}
for i=1 TO M **do**
 $c_i^{(1)*} \leftarrow \arg \max_c \text{power}(\hat{\mathbf{y}}_i^{(1)}, \gamma^{(1)})$
 $c_i^{(2)*} \leftarrow \arg \max_c \text{power}(\hat{\mathbf{y}}_i^{(2)}, \gamma^{(2)})$
end for
end if
end for

7.3 Datasets and Network Setups

The model is evaluated in four different setups. Each setup has two disjoint sets that represent the same unknown class. All setups are based on MNIST. The process for generating the training and testing follow these steps. First, the training set of the MNIST is split into two disjoint sets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ where half of the sample of each digit is in one set and the other half is in the other set. In other words, if digit 1 has ten samples, five samples go to set $\mathbf{X}^{(1)}$, and the other five samples go to set $\mathbf{X}^{(2)}$. Second, each sample from $\mathbf{X}^{(1)}$ is linked to another sample from $\mathbf{X}^{(2)}$ with the condition that both samples represent the same category. Note that it is a standard approach to use labeled data to measure the results of unsupervised scenarios [91, 92]. Third, all samples from *input2* have applied the spatial transformations. In this manner, the task has two different sets of feature distributions. Hence, the four variations are: a) Identity, b) Rotation 90 degrees, c) Inverse, and d) Random Rotation.

This process generates 21,000 and 4,000 samples for training and validation sets (respectively). The testing set is generated based on the testing set from MNIST. As a result, the size of the testing set is 4,000 samples. Additionally, each dataset follows a uniform distribution between all digits.

The experimental design follows a ten cross-validation approach. The parameters for the three datasets are the following. Each MLP has two hidden layers with 200 and 100 neurons. The weighting vectors $\gamma^{(1)}$ and $\gamma^{(2)}$ are initialized to 1.0. The learning rate of both networks follows a schedule that the learning rate decreased by half every 1,000 epochs, and the initial value is set to 1.0. The learning rate for weighting vector follows a different approach where the learning rate is based on $1/(100 + epoch)^{0.3}$. The mini-batch size is 25% of the training set (5,250 sample pairs). The motivation behind using a significant mini-batch of samples is to get a closer distribution to the dataset. The selected parameters of the random rotation scenario are different. Similar to the previous setups, the model has two hidden layers for each MLP with 400 and 150 neurons. The learning rate starts at 1.2.

The classless association problem is compared to two baselines. The first baseline is used an upper bound, which the categories are available, and the other baseline is used as lower bound, which categories are not available. Therefore, the presented model is compared to MLP trained with labeled data for each set and two cluster algorithms (K-means and Hierarchical Clustering). The clustering algorithm implementations are

provided by scikit-learn [93].

7.4 Results and Discussion

In this section, the average of a ten cross-validation is reported. Table 7.1 summarizes the results of all datasets. The first three datasets show that classless association model reaches an Association Accuracy (AAcc) below than the supervised case. This metric is not calculated for the clustering approaches because there is no clear link without analyzing each pair of clusters. In more detail, the results of this association only show that a pair of input samples agree on the same index. Nonetheless, it does not show if the model learns to discriminate classes. The purity metric of the first three classes shows a good performance in both cases (supervised and unsupervised scenarios). The performance of the presented model is lower than the supervised case. However, it is still good because of the lack of labeled information. These observations are also supported by comparing the performance to the unsupervised case where the classless model reaches better results. For example, the clustering algorithm reaches approx. 64% whereas the presented model reaches around 89% (MNIST and Inverted MNIST). In a most extreme scenario, this model shows its superiority against the clustering algorithms in the case of random rotation MNIST. The association accuracy is not as good as the first three datasets.

The classless model learns the concept of classes while it is training. Figure 7.4 shows an example of this behavior. Initially, the samples are uniformly distributed between the indexes, therefore the purity is closed to random chance. After 1,000 epochs, the purity increases and the association matrix shows initial results. At this stage, it is unclear what samples are grouped based on the other indexes. However, new groups with a clear pattern are presented after 3,000 epochs. Some examples are digits 1 and 0. After 49,000 epochs, both networks match the classification and that is also supported by the purity of each MLP and the association matrix.

Figure 7.5 compares the best and worst results between all folds. It can be observed that the best result has a perfect matching between both networks (main diagonal in the Association Matrix). The worst result presents a partial matching in the sense that both networks classify the some samples of different classes with the same index, which is supported the low purity of the second MLP. One group with two digits causes this bad performance.

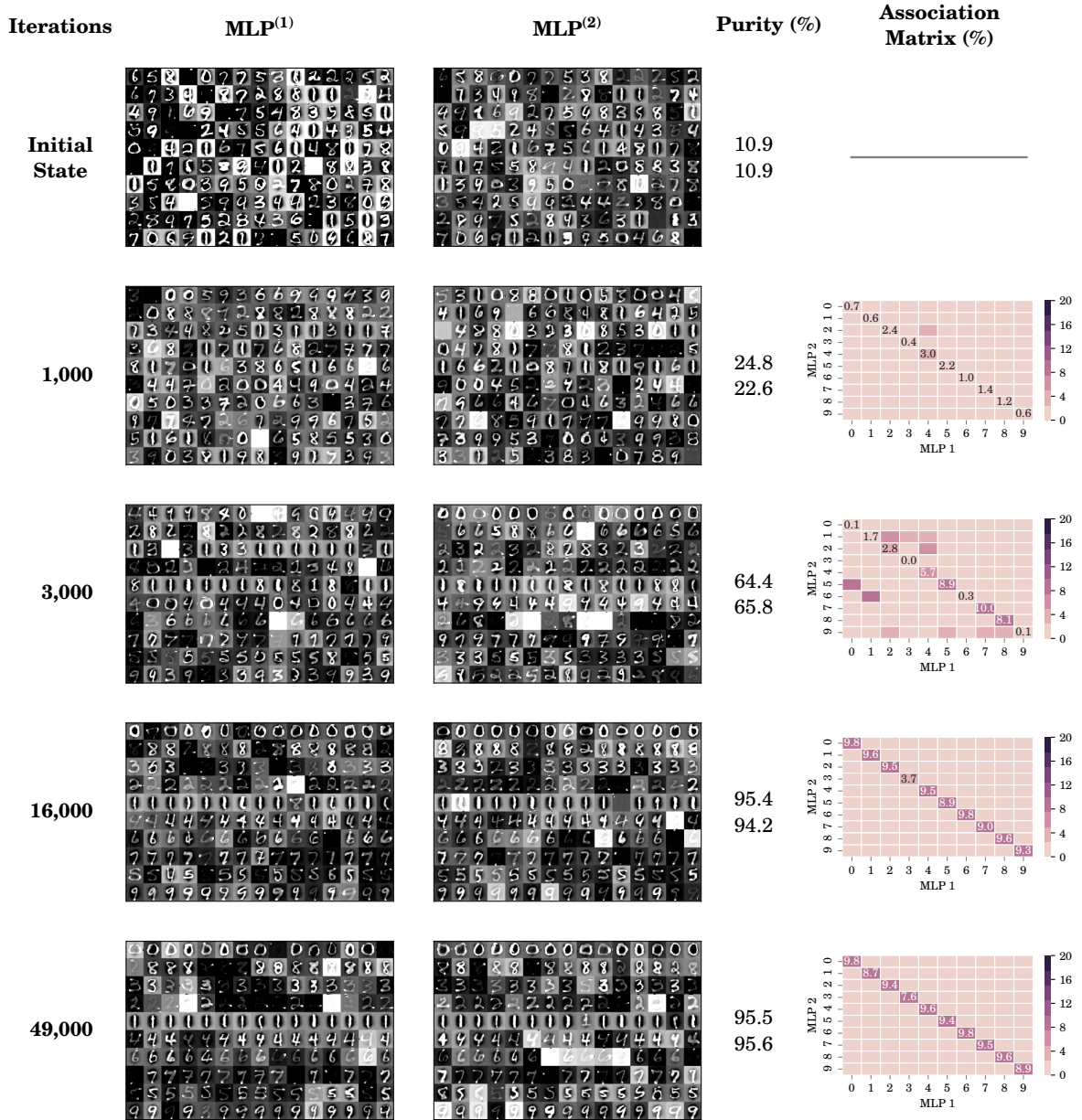


FIGURE 7.4. Example of the training algorithms of the classless association. In this case, the model relies on learning the concept of categories by classifying similar samples with the same index. The brightness and the lightness of digits are modified in the MLP columns for better visualization.

TABLE 7.1. Association Accuracy (%) and Purity (%) results. This model is compared with the supervised scenario (labels are provided) and with K-means and Hierarchical Agglomerative clustering (no label information).







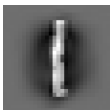
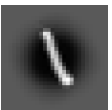
Dataset		Model	Association Accuracy (%)	Purity (%)	
Sample 1	Sample 2			$\mathbf{X}^{(1)}$	$\mathbf{X}^{(2)}$
		supervised association	96.7 ± 0.3	96.7 ± 0.2	96.6 ± 0.3
		classless association	86.1 ± 3.2	89.6 ± 4.5	89.0 ± 4.2
		K-means	-	63.9 ± 2.2	62.5 ± 3.7
		Hierarchical Agglomerative	-	64.9 ± 4.7	64.3 ± 5.5
		supervised association	93.2 ± 0.3	96.4 ± 0.2	96.6 ± 0.2
		classless association	86.5 ± 2.5	82.9 ± 4.5	82.9 ± 4.3
		K-means	-	65.0 ± 2.8	64.0 ± 3.6
		Hierarchical Agglomerative	-	65.4 ± 3.5	64.1 ± 4.1
		supervised association	93.2 ± 0.3	96.5 ± 0.2	96.5 ± 0.2
		classless association	89.2 ± 2.4	89.0 ± 6.8	89.1 ± 6.8
		K-means	-	64.8 ± 2.0	65.0 ± 2.5
		Hierarchical Agglomerative	-	64.8 ± 4.4	64.4 ± 3.8
		supervised association	88.0 ± 0.5	96.5 ± 0.3	90.9 ± 0.5
		classless association	69.3 ± 2.2	75.8 ± 7.3	65.3 ± 5.0
		K-means	-	64.8 ± 2.6	14.8 ± 0.4
		Hierarchical Agglomerative	-	65.9 ± 2.8	15.2 ± 0.5

Figure 7.6 shows a comparison between the learning curve (Purity) of the supervised and class-less cases. Note that the supervised case learns faster with the same network parameters as the class-less association. Also, the classless approach learns slower because the class convergence is learned by similar grouping elements that have similar features.

7.5 Summary

In this chapter, a new association model has been described, in which the two instances of the same *unknown* semantic concepts are the input samples. Additionally, the training algorithm does not rely on labeled datasets. Therefore, a new loss function has been introduced that employs statistical distributions as targets. Several findings are described below

- The performances of the presented model is located between the unsupervised

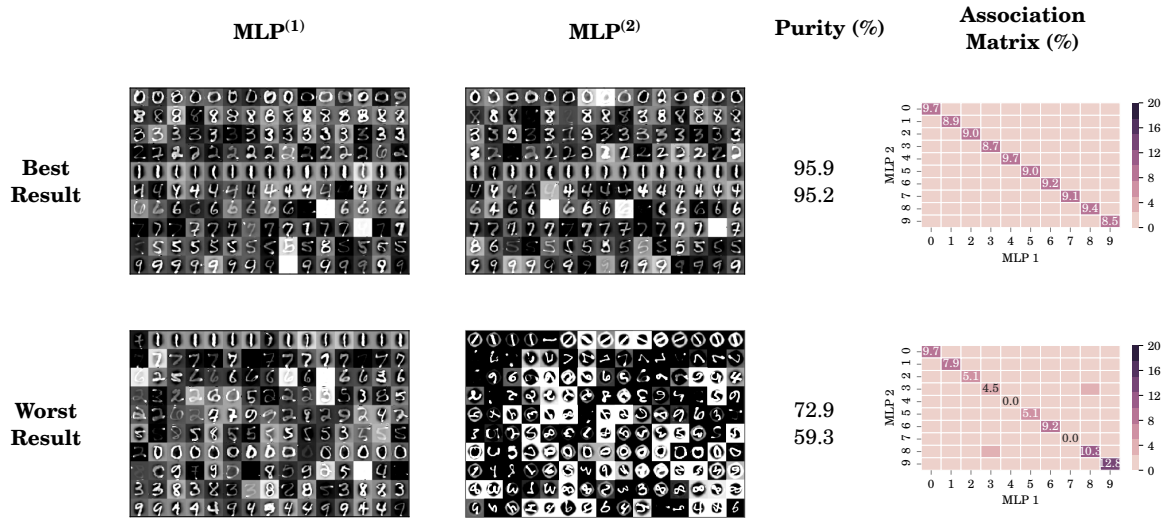


FIGURE 7.5. Examples of the best and the worst results between all folds. The best results relies on similar elements are grouped by the same index. On the other hand, there are some digits that are mixed up, for instances the digits four and nine at index 9. The brightness and the lightness of digits are modified in the MLP columns for better visualization.

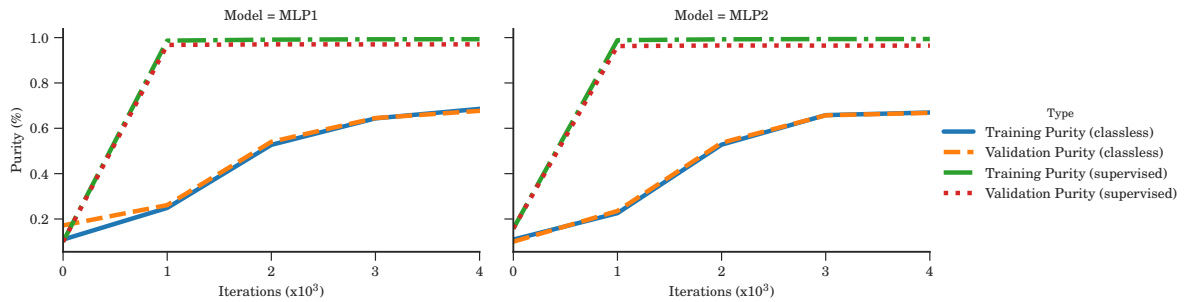


FIGURE 7.6. One limitation of the model is related to the required iterations for converging. It is observed that the supervised scenario requires only 1,000 epochs to reach around 1.0 whereas the classless association reaches around 0.6 after 4,000 epochs. This is caused by the dynamic generation of indexes.

and supervised approaches. In the one hand, the two clustering algorithms reach results between 15% and 65% whereas this model surpasses them with better performances between 65% and 89%. On the other hand, MLP trained with labels (supervised approach) has a better performance than the classless association model with results between 90% and 96%. Hence, it is possible to infer that the classless association has a good trade-off between the lack of labeled data and the performance of the supervised case as baselines.

- The training algorithm can slowly make groups with instances of the same categories. Also, both networks agree on the same group-index. However, one limitation of the model is to group two semantic concepts into one group (random rotation dataset). Thus, the model cannot recover from this condition. In other words, the instances of different classes that are in the same group cannot be split apart by the model.
- The training algorithm is slow comparing to MLP (trained with labeled data) and the clustering algorithms. The main bottleneck is updating the pseudo-classes. As a result, the information that is embedded in the model needs to be updated. One possible solution is that the model decides by itself when the pseudo-classes are updated.

CONCLUSION AND FUTURE WORK

This thesis describes a new symbolic association framework that combines NN and SGP. Additionally, the presented model is also inspired by AL in infants, in which newborns learn that two or more sensory signals represent the same concept. There are many challenges in this scenario. Firstly, SGP is still an open problem because it is unclear how the human brain can map abstract concepts to the real world and can manipulate those concepts. For example, the concepts related to numbers are manipulated for mathematical operations, whereas letters are manipulated for communication. Secondly, infants learn to acquire their vocabulary based on association visual and audio sensory information. For example, a newborn can receive two signals: one signal is the waveform of *play with the ball*, and the other is the visual representation of an environment with the ball. Therefore, the children require several processes to understand these two signals. The waveform and the scene must be segmented into *units* (e.g., words, objects). Afterwards, each unit is classified into concepts where same concepts, such as the *ball*, can be linked to each other modality. Thirdly, the importance of each modality for vocabulary acquisition in children. Consider the concept *triangle*. A visual representation might be a contour with three lines that are connected by three vertexes, and an audio representation might be the sound of *triangle*. Moreover, a third representation might be the feeling of texture that the triangle is made of, e.g., wood, wool, or plastic. Each of those examples are represented in three different formats: visual, audio, and haptic.

The main contributions rely on NN, CC, and CS for learning the association in a

simplified scenario that is inspired by vocabulary acquisition in infants. It is possible to interpret the NN output as numerical *symbolic features* because discriminant features can be embedded in their architectures. The symbolic association has been evaluated in mono- and multi-modal association tasks.

8.1 Concluding Remarks

This thesis has investigated mainly two NN architectures for the association learning: MLP and LSTM. As a general overview, the general association framework exploits of using output or target vectors from one network to train another network. This *cross-learning* approach force that both NN converge to the same category. Algorithm 3 shows the general association algorithm of the presented models.

Algorithm 3 Pseudocode of the Association Learning.

Require: two neural networks: $NN^{(1)}, NN^{(2)}$ and two input sets: $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$
 $\hat{\mathbf{Y}}^{(1)} \leftarrow forward_step(NN^{(1)}, \mathbf{X}^{(1)})$
 $\hat{\mathbf{Y}}^{(2)} \leftarrow forward_step(NN^{(2)}, \mathbf{X}^{(2)})$
 $backward_step(NN^{(1)}, \hat{\mathbf{Y}}^{(2)})$
 $backward_step(NN^{(2)}, \hat{\mathbf{Y}}^{(1)})$

This thesis focuses on the association task in the mono- and multi-modal domains. The mono-modal scenario uses two datasets: digit association and object association. The multi-modal scenario is applied to sample pairs and sequence pairs. The sample pairs are images and texts whereas the sequence pairs are panorama and audios.

Three models have been proposed in this thesis for each association scenario. The first model relies on two NN networks that associate two isolated elements of the same concept. The performance of the model reaches similar results to the traditional approach, which is MLP networks trained independently on each input set. This approach is evaluated in two cases mono- and multi-modal associations based on MNIST and COIL-20 datasets for the first case, and TVGraz and Wikipedia datasets for the second case.

The second model is based on two LSTM networks for associating sequences with weakly labels. In contrast to the MLP-based approach, this model produces two latent spaces that are aligned to each other. This approach has also been evaluated in several scenarios: mono- and multi-modal sequences. The performances are still similar to the traditional case where each LSTM is trained independently in the input set. Moreover,

this model is extended to handling sequences that contain elements that are presented in one or both modalities. The extension of the model has shown better results than the LSTM model that assumes both sequences represent the same series of concepts. This model is robust concerning the number of missing elements and holds a similar performance in each multi-modal setup.

The third model is an extension of MLP-based approach where the association task is defined by two samples that represent the same *unknown* concept. This model relies on a statistical distribution of classes instead of predefined classes. Therefore, the training set does not have any labeled data. This version of the model has been evaluated in a mono-modal dataset where one input set has been applied a spatial transformation, such as identity, rotation 90 degrees, inverse, and random rotation. The performance of the model is compared to two cases. One side is the supervised scenario where the labels are available for each sample, and the other side the labels are not available at all. MLP networks are used as upper bound (supervised), and two cluster algorithms are used as lower bound (unsupervised). The classless model reaches results that are in both cases. Moreover, the performance reaches better than the clustering, especially in the random rotation where clustering algorithms entirely fail to make groups with similar samples. On the other hand, the performance of the class-less model reaches a good performance about the supervised case regarding trade-off between label and accuracy. This limitation is caused by the loss function cannot impose similar samples that represent different categories. Therefore, the model cannot split two or more digits after they are in the same group. For example, the digits three and eight are grouped by the same pseudo-class (see Figure 8.1).

8.2 Future Directions

The presented work has several contributions for symbolic approaches based on NN. This section provides future directions in the association learning framework for two architectures: LSTM- and Classless MLP-based approaches. The goal of proposed directions is to extend the association framework to reach human-level association between different modalities. This section is divided as follows. Section 8.2.1 describes the future work based on the association framework with parallel LSTM. Section 8.2.2 describes the future work based on the classless association

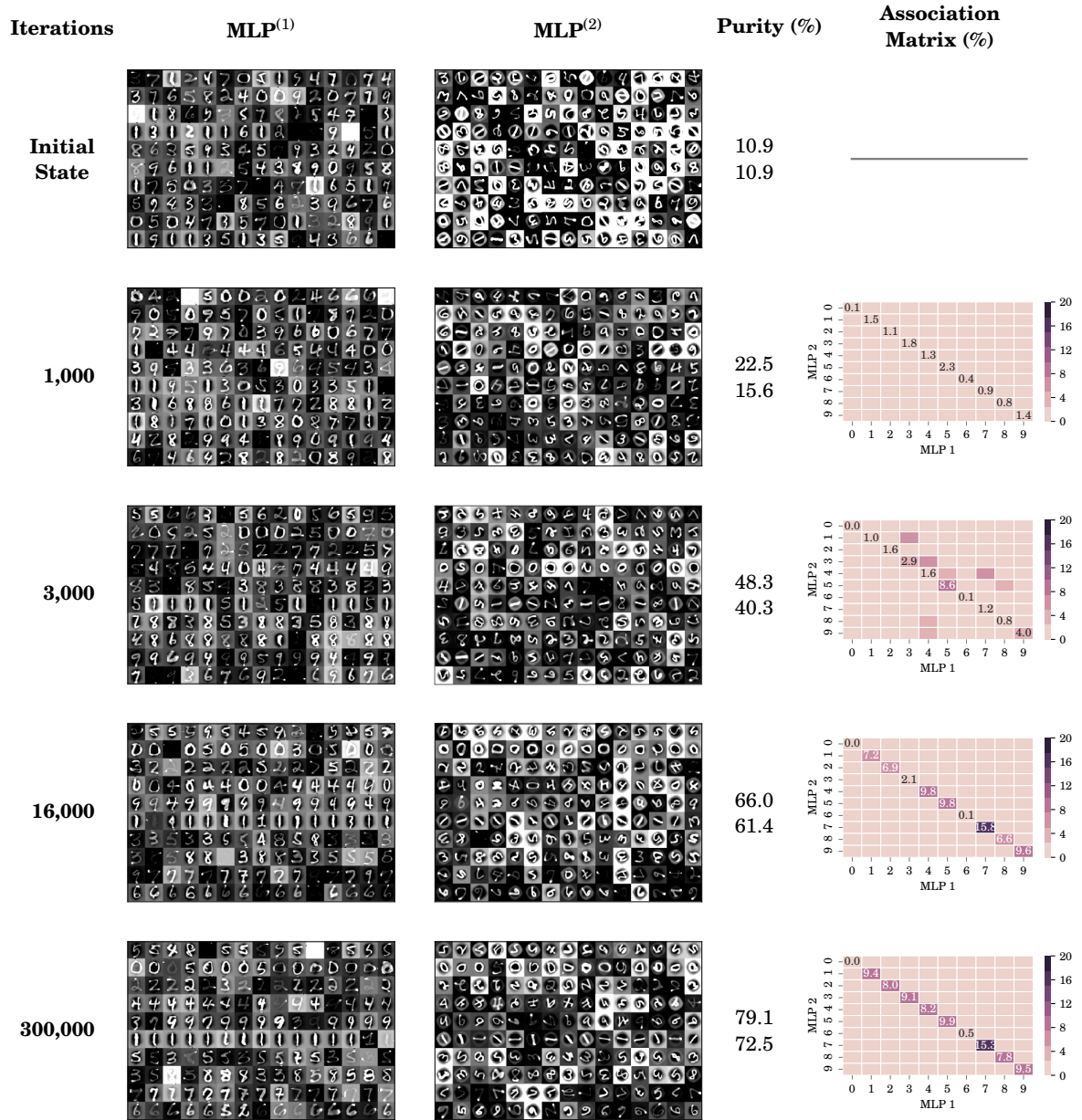


FIGURE 8.1. Limitation of the classless association model is related to the class convergence. Note that the second MLP network starts combining two digits (i.e., three and eight) into one group after 16,000 iterations. Therefore, the loss function cannot impose the constraint of the uniform distribution.

8.2.1 LSTM-based Approach

The model based on LSTM has several directions for future work. In this section, four scenarios are proposed

Association between two-dimensional images and audio: The original assumption is to align panoramas and audio signals. The next step is to align a different setup of two-dimensional images and one-dimensional audio with weakly labels. There are several challenges to be considered in this task. One of the most critical challenges is to segment all objects on the image into one-dimensional sequence with the same order as the audio signal.

More modalities: The current version of the model aligns only two modalities. Furthermore, a third signal can be included based on motor sensors, i.e., recording the process of writing digits. The new information can be combined given two approaches. One approach is to use DTW given the three signals. The other approach is combining the best relation between each pair of signals, for example: selecting two of the following pairs audio-visual, visual-motor, and motor-audio.

More Languages: The different languages can be visually grounded to the same objects. In this scenario, a word in English and a word in French are linked to the same visual representation of the desired concept. This direction can be seen similar to the direction of *more modalities*

Classless LSTM: The last direction is to adapt the same motivation of the Classless MLP-based approach to LSTM. Thus, LSTM with CTC can be trained based on a statistical distribution instead of classes. One of the main challenges is the number of classes that are presented in the sequence whereas one class per input is presented in the MLP-based approach.

8.2.2 Classless Association (MLP-based Approach)

This model has been evaluated in a simplified scenario where the next logical step is to find the limits of the model.

Few- and One-shot Learning: The first direction is to evaluate against other models regarding few- and one-shot learning. This model does not have any information

regarding of samples. However, its performance can be improved with minimal label effort, for instance, one label per class.

More challenging tasks: The second direction is to extend the model to more challenging case, such as object recognition and multi-modal association. Additionally, the training algorithm has been evaluated with fully-connected layers. It is essential to evaluate this model with different layers and architectures, such as Convolutional or Residual connections.

Shared-weights between MLPs: The third direction is to exploit *shared weight* between networks of different modalities. For instance, visual samples can be useful for training a network with text samples. In this case, the networks have mutual information that is embedded.

More distributions: The last direction is to evaluate the robustness of the model with different statistical distributions. At the moment, the model reaches good performances if the training set and the target distribution are both uniform distribution. However, the statistical distributions in real applications are more difficult to obtain. Therefore, it is important to analyze cases that input distribution is different from the target distribution.

BIBLIOGRAPHY

- [1] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–25, November 1999. ISSN 1097-6256. doi: 10.1038/14819. URL <http://www.ncbi.nlm.nih.gov/pubmed/10526343>.
- [2] Arnaud Delorme, Jacques Gautrais, Rufin Van Rullen, and Simon Thorpe. Spikenet: A simulator for modeling large networks of integrate and fire neurons. *Neurocomputing*, 26:989–996, 1999.
- [3] Melvyn A. Goodale and A. David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20 – 25, 1992. ISSN 0166-2236. doi: [https://doi.org/10.1016/0166-2236\(92\)90344-8](https://doi.org/10.1016/0166-2236(92)90344-8). URL <http://www.sciencedirect.com/science/article/pii/0166223692903448>.
- [4] Deb K Roy and Alex P Pentland. Learning words from sights and sounds: A computational model. *Cognitive science*, 26(1):113–146, 2002.
- [5] Lou Boves, Louis ten Bosch, and Roger Moore. Acorns-towards computational modeling of communication and recognition skills. In *Cognitive Informatics, 6th IEEE International Conference on*, pages 349–356. IEEE, 2007.
- [6] Chen Yu. The emergence of links between lexical acquisition and object categorization: A computational study. *Connection science*, 17(3-4):381–397, 2005.
- [7] Nikos K Logothetis and David L Sheinberg. Visual object recognition. *Annual review of neuroscience*, 19(1):577–621, 1996.
- [8] Roberta Michnick Golinkoff, Carolyn B Mervis, and Kathryn Hirsh-Pasek. Early object labels: the case for a developmental lexical principles framework [*]. *Journal of child language*, 21(1):125–155, 1994.

BIBLIOGRAPHY

- [9] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990.
- [10] John R Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3(03): 417–424, 1980.
- [11] Michael J. Mayo. Symbol grounding and its implications for artificial intelligence. In *Proceedings of the 26th Australasian Computer Science Conference - Volume 16*, ACSC '03, pages 55–60, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. ISBN 0-909-92594-1. URL <http://dl.acm.org/citation.cfm?id=783106.783113>.
- [12] Luc Steels. The symbol grounding problem has been solved, so what's next? *Symbols, Embodiment and Meaning*. Oxford University Press, Oxford, UK, (2005):223–244, 2008. ISSN 978-0199217274. doi: 10.1093/acprof:oso/9780199217274.003.0012.
- [13] Mariarosaria Taddeo and Luciano Floridi. Solving the symbol grounding problem: a critical review of fifteen years of research. *Journal of Experimental & Theoretical Artificial Intelligence*, 17(4):419–445, 2005.
- [14] Silvia Coradeschi, Amy Loutfi, and Britta Wrede. A short review of symbol grounding in robotic and intelligent systems. *KI-Künstliche Intelligenz*, 27(2):129–136, 2013.
- [15] Tadahiro Taniguchi, Takayuki Nagai, Tomoaki Nakamura, Naoto Iwahashi, Tetsuya Ogata, and Hideki Asoh. Symbol emergence in robotics: a survey. *Advanced Robotics*, 30(11-12):706–728, 2016.
- [16] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 32(4):64–76, 2011.
- [17] Alessandro Di Nuovo, M Vivian, and Angelo Cangelosi. Grounding fingers, words and numbers in a cognitive developmental robot. In *Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), 2014 IEEE Symposium on*, pages 9–15. IEEE, 2014.
- [18] Douwe Kiela and Stephen Clark. Multi-and cross-modal semantics beyond vision: Grounding in auditory perception. In *EMNLP*, pages 2461–2470, 2015.

- [19] Kimberly S. Kraebel and Peter C. Gerhardstein. Three-month-old infants' object recognition across changes in viewpoint using an operant learning procedure. *Infant behavior & development*, 29(1):11–23, January 2006. ISSN 1934-8800. doi: 10.1016/j.infbeh.2005.10.002. URL <http://www.ncbi.nlm.nih.gov/pubmed/17138257>.
- [20] Ramesh S. Bhatt and Susan E. Waters. Perception of three-dimensional cues in early infancy. *Journal of experimental child psychology*, 70(3):207–24, September 1998. ISSN 0022-0965. doi: 10.1006/jecp.1998.2458. URL <http://www.ncbi.nlm.nih.gov/pubmed/9742180>.
- [21] Teresa Wilcox. Object individuation: infants' use of shape, size, pattern, and color. *Cognition*, 72(2):125–66, September 1999. ISSN 0010-0277. URL <http://www.ncbi.nlm.nih.gov/pubmed/10553669>.
- [22] József Fiser and Richard N Aslin. Statistical learning of new visual feature combinations by infants. *Proceedings of the National Academy of Sciences of the United States of America*, 99(24):15822–6, November 2002. ISSN 0027-8424. doi: 10.1073/pnas.232472899. URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=137800&tool=pmcentrez&rendertype=abstract>.
- [23] Elizabeth S. Spelke. Principles of object perception. *Cognitive Science*, 14(1):29–56, March 1990. ISSN 03640213. doi: 10.1016/0364-0213(90)90025-R. URL [http://doi.wiley.com/10.1016/0364-0213\(90\)90025-R](http://doi.wiley.com/10.1016/0364-0213(90)90025-R).
- [24] Angela D. Friederici and Jeanine M. I. Wessels. Phonotactic knowledge of word boundaries and its use in infant speech perception. *Perception & psychophysics*, 54(3):287–95, September 1993. ISSN 0031-5117. URL <http://www.ncbi.nlm.nih.gov/pubmed/8414887>.
- [25] Peter W. Jusczyk. How infants begin to extract words from speech. *Trends in cognitive sciences*, 3(9):323–328, September 1999. ISSN 1879-307X. URL <http://www.ncbi.nlm.nih.gov/pubmed/10461194>.
- [26] Peter W. Jusczyk, Derek M. Houston, and Mary Newsome. The beginnings of word segmentation in english-learning infants. *Cognitive psychology*, 39(3-4):159–207, 1999. ISSN 0010-0285. doi: 10.1006/cogp.1999.0716. URL <http://www.ncbi.nlm.nih.gov/pubmed/10631011>.
- [27] Bruna Pelucchi, Jessica F Hay, and Jenny R Saffran. Statistical learning in a natural language by 8-month-old infants. *Child development*, 80(3):674–85, 2009.

BIBLIOGRAPHY

- ISSN 1467-8624. doi: 10.1111/j.1467-8624.2009.01290.x. URL <http://www.ncbi.nlm.nih.gov/pubmed/19489896>.
- [28] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264 (5588):746–748, 1976.
- [29] Roberta M Golinkoff and Kathy Hirsh-Pasek. *How babies talk: The magic and mystery of language in the first three years of life*. Penguin, 2000.
- [30] Marie T Balaban and Sandra R Waxman. Do words facilitate object categorization in 9-month-old infants? *Journal of experimental child psychology*, 64(1):3–26, January 1997. ISSN 0022-0965.
- [31] Katharine Graf Estes, Julia L Evans, Martha W Alibali, and Jenny R Saffran. Can infants map meaning to newly segmented words? Statistical segmentation and word learning. *Psychological science*, 18(3):254–60, March 2007. ISSN 0956-7976. doi: 10.1111/j.1467-9280.2007.01885.x. URL <http://www.ncbi.nlm.nih.gov/pubmed/17444923>.
- [32] Michiko Asano, Mutsumi Imai, Sotaro Kita, Keiichi Kitajo, Hiroyuki Okada, and Guillaume Thierry. Sound symbolism scaffolds language development in preverbal infants. *cortex*, 63:196–205, 2015.
- [33] Lisa Gershkoff-Stowe and Linda B Smith. Shape and the first hundred nouns. *Child development*, 75(4):1098–114, 2004. ISSN 0009-3920.
- [34] Meagan Yee, Susan S Jones, and Linda B Smith. Changes in visual object recognition precede the shape bias in early noun learning. *Frontiers in psychology*, 3(December):533, January 2012. ISSN 1664-1078. doi: 10.3389/fpsyg.2012.00533. URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3512352&tool=pmcentrez&rendertype=abstract>.
- [35] Janet F Werker, Leslie B Cohen, Valerie L Lloyd, Marianella Casasola, and Christine L Stager. Acquisition of word-object associations by 14-month-old infants. *Developmental psychology*, 1998. URL http://infantstudies.psych.ubc.ca/uploads/forms/1252960056WerkerEtAl_1998.pdf.
- [36] Shannon M. Pruden, Kathy Hirsh-Pasek, Roberta Michnick Golinkoff, and Elizabeth A. Hennon. The birth of words: ten-month-olds learn words through perceptual salience. *Child development*, 77(2):266–80, 2006. ISSN 0009-3920. doi:

- 10.1111/j.1467-8624.2006.00869.x. URL <http://www.ncbi.nlm.nih.gov/pubmed/16611171>.
- [37] Dorothy Burlingham. Some notes on the development of the blind. *The psychoanalytic study of the child*, 16(1):121–145, 1961.
- [38] John B Brannon. Linguistic word classes in the spoken language of normal, hard-of-hearing, and deaf children. *Journal of speech and hearing research*, 11(2):279–287, 1968.
- [39] Amy R Lederberg and Patricia E Spencer. Vocabulary development of deaf and hard of hearing children. *Context, cognition, and deafness*, pages 88–112, 2001.
- [40] Patricia Elizabeth Spencer. Looking without listening: is audition a prerequisite for normal development of visual attention during infancy? *Journal of deaf studies and deaf education*, 5(4):291–302, January 2000. ISSN 1465-7325.
- [41] Elaine S Andersen, Anne Dunlea, and Linda Kekelis. The impact of input: language acquisition in the visually impaired. *First Language*, 13(37):23–49, January 1993. ISSN 0142-7237.
- [42] Serhat Selcuk Bucak, Rong Jin, and Anil K. Jain. Multi-label learning with incomplete class assignments. In *CVPR 2011*, number iii, pages 2801–2808. IEEE, June 2011. ISBN 978-1-4577-0394-2. doi: 10.1109/CVPR.2011.5995734. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5995734>.
- [43] Timothee Cour, Benjamin Sapp, and Ben Taskar. Learning from partial labels. pages 1225–1261, 2011. URL http://repository.upenn.edu/cis_papers/514/.
- [44] Luo Jie and Francesco Orabona. Learning from candidate labeling sets. *Neural Information Processing Systems*, pages 1–9, 2011. URL <http://eprints.pascal-network.org/archive/00007698/>.
- [45] Kim Plunkett, Chris Sinha, Martin F Møller, and Ole Strandsby. Symbol grounding or the emergence of symbols? vocabulary growth in children and a connectionist net. *Connection Science*, 4(3-4):293–312, 1992.
- [46] Alessio Plebe, V. De la Cruz, and Marco Mazzone. Simulating the acquisition of object names. *ACL 2007*, (June):57, 2007.

BIBLIOGRAPHY

- [47] Julien Mayor and Kim Plunkett. Learning to associate object categories and label categories: A self-organising model. In *Proceedings of the 30th annual conference of the Cognitive Science Society*, pages 697–702, 2008.
- [48] Igor Farkas and Li Ping. A self-organizing neural network model of the acquisition of word meaning. In *Proceedings of the 4th International Conference on Cognitive Modeling*, 2001. doi: 10.1.1.24.4622. URL http://books.google.com/books?hl=en&lr=&id=krqrgY6IZNUC&oi=fnd&pg=PA90&dq=a+self-organizing+neural+network+model+of+the+acquisition+of+word+meaning&ots=nU_SQwUy3l&sig=ALQeXS_q70E-yLnuj0x9joAJT9ohttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.4622.
- [49] Ping Li, Igor Farkas, and Brian MacWhinney. Early lexical development in a self-organizing neural network. *Neural networks : the official journal of the International Neural Network Society*, 17(8-9):1345–62, 2004. ISSN 0893-6080. doi: 10.1016/j.neunet.2004.07.004. URL <http://www.ncbi.nlm.nih.gov/pubmed/15555870>.
- [50] Katrin S Lohan, Karola Pitsch, Katharina J Rohlfing, Kerstin Fischer, Joe Saunders, Hagen Lehmann, Chrystopher Nehaniv, and Britta Wrede. Contingency allows the robot to spot the tutor and to learn from interaction. In *2011 IEEE International Conference on Development and Learning (ICDL)*, pages 1–8. IEEE, August 2011. ISBN 978-1-61284-990-4. doi: 10.1109/DEVLRN.2011.6037341. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6037341>.
- [51] Peter Carbonetto and Nando de Freitas. Why can't José read?: the problem of learning semantic associations in a robot environment. In *Proceedings of the HLT-NAACL 2003 workshop on Learning word meaning from non-linguistic data -*, volume 6, pages 54–61, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119212.1119220. URL <http://dl.acm.org/citation.cfm?id=1119220http://portal.acm.org/citation.cfm?doid=1119212.1119220>.
- [52] Thomas Serre, Lior Wolf, and Tomaso Poggio. Object Recognition with Features Inspired by Visual Cortex. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 994–1000. IEEE, 2005. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.254. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467551http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1467551.

-
- [53] Simon J Thorpe, Rudy Guyonneau, Nicolas Guilbaud, Jong-Mo Allegraud, and Rufin VanRullen. Spikenet: Real-time visual processing with one spike per neuron. *Neurocomputing*, 58:857–864, 2004.
- [54] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.
- [55] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [56] Tom. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN 9780071154673. URL <https://books.google.de/books?id=EoYBngEACAAJ>.
- [57] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [58] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989.
- [59] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL <https://doi.org/10.1007/BF02551274>.
- [60] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [61] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, Nov 1997. ISSN 1053-587X. doi: 10.1109/78.650093.
- [62] Sepp Hochreiter. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, April 1998. ISSN 0218-4885.
- [63] Sepp Hochreiter and Juergen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735—1780, 1997.

- [64] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.
- [65] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.
- [66] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 369–376, New York, New York, USA, 2006. ACM Press. ISBN 1595933832.
- [67] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.
- [68] Federico Raue, Sebastian Palacio, Thomas M. Breuel, Wonmin Byeon, Andreas Dengel, and Marcus Liwicki. *Symbolic Association Using Parallel Multilayer Perceptron*, pages 347–354. Springer International Publishing, Cham, 2016. ISBN 978-3-319-44781-0. doi: 10.1007/978-3-319-44781-0_41. URL https://doi.org/10.1007/978-3-319-44781-0_41.
- [69] Federico Raue, Wonmin Byeon, Thomas. M. Breuel, and Marcus Liwicki. Parallel Sequence Classification using Recurrent Neural Networks and Alignment. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, 2015.
- [70] Federico Raue, Wonmin Byeon, Thomas M. Breuel, and Marcus Liwicki. Symbol grounding in multimodal sequences using recurrent neural network. In *Workshop Cognitive Computation: Integrating Neural and Symbolic Approaches at NIPS*, 2015.
- [71] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.
- [72] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society.*, 39 (1):1–38, 1977.

- [73] Wikimedia Commons. File:euclidean vs dtw.jpg — wikimedia commons, the free media repository, 2016. URL https://commons.wikimedia.org/w/index.php?title=File:Euclidean_vs_DTW.jpg&oldid=191279552. [Online; accessed 20-November-2017].
- [74] Donald J Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. pages 359–370, 1994.
- [75] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [76] Sameer A Nene, Shree K Nayar, and Hiroshi Murase. Columbia object image library (coil-20). Technical report, 1996.
- [77] Inayatullah Khan, Amir Saffari, and Horst Bischof. Tvgraz: Multi-modal learning of object categories by combining textual and visual features. In *AAPR Workshop*, pages 213–224, 2009.
- [78] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.
- [79] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A New Approach to Cross-Modal Multimedia Retrieval. In *ACM International Conference on Multimedia*, pages 251–260, 2010.
- [80] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435.
- [81] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1470–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4.
- [82] David G Lowe. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2, 1999. doi: 10.1109/ICCV.1999.790410. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790410>.

- [83] Jose Costa Pereira and Nuno Vasconcelos. Cross-modal domain adaptation for text-based regularization of image semantics in image retrieval systems. *Computer Vision and Image Understanding*, 124:123 – 135, 2014. ISSN 1077-3142. doi: <http://dx.doi.org/10.1016/j.cviu.2014.03.003>. Large Scale Multimedia Semantic Indexing.
- [84] Federico Raue, Marcus Liwicki, and Andreas Dengel. Symbolic association learning inspired by the symbol grounding problem. In Thomas Villmann and Frank-Michael Schleif, editors, *Workshop New Challenges in Neural Computation (NC²)*, volume 4 of *Machine Learning Reports*, 2016. ISSN:1865-3960 http://www.techfak.uni-bielefeld.de/fschleif/mlr/mlr_04_2016.pdf.
- [85] Paul Taylor, Alan W Black, and Richard Caley. The architecture of the festival speech synthesis system. 1998.
- [86] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006.
- [87] Federico Raue, Andreas Dengel, Thomas M Breuel, and Marcus Liwicki. Symbol grounding association in multimodal sequences with missing elements. *Journal of Artificial Intelligence Research*, 61:787–806, 2018.
- [88] S Nayar, S Nene, and Hiroshi Murase. Columbia object image library (coil-100). Technical report, 1996. URL <http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>.
- [89] Xiang Ma, Dan Schonfeld, and Ashfaq Khokhar. A general two-dimensional hidden markov model and its application in image classification. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI–41. IEEE, 2007.
- [90] Federico Raue, Sebastian Palacio, Andreas Dengel, and Marcus Liwicki. *Classless Association Using Neural Networks*, pages 165–173. Springer International Publishing, Cham, 2017. ISBN 978-3-319-68612-7. doi: 10.1007/978-3-319-68612-7_19. URL https://doi.org/10.1007/978-3-319-68612-7_19.
- [91] Yen-Chang Hsu and Zsolt Kira. Neural network-based clustering using pairwise constraints. *arXiv preprint arXiv:1511.06321*, 2015.

- [92] Ilya Sutskever, Rafal Jozefowicz, Karol Gregor, Danilo Rezende, Tim Lillicrap, and Oriol Vinyals. Towards principled unsupervised learning. *arXiv preprint arXiv:1511.06440*, 2015.
- [93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Federico Raue

Curriculum Vitae

EDUCATION

Ph. D. in Computer Science 11/2012-01/2017

TU Kaiserslautern (Kaiserslautern, Germany)

Thesis: “Association Learning inspired by the Symbol Grounding Problem”. Advisor: Prof. Andreas Dengel.

Qualify Period for being accepted as Ph. D. student 10/2011-11/2012

TU Kaiserslautern (Kaiserslautern, Germany)

The program consisted of 37 credit points and was divided in a set of courses and a research project. I have approved the following courses: High-Performance Computing with GPU, 3D Computer Vision, Multimedia Information Retrieval and Algorithm Engineering. Moreover, the project was about “Benchmarking SpikeNet architecture in Object Recognition task”

Master of Artificial Intelligence 2004-2005

Katholieke Universiteit Leuven (Leuven, Belgium)

Thesis: “Author Recognition of texts posted on the World Wide Web and Fragments of literature books”. Advisor: Prof. M. F. Moens

Computer Engineering 1996-2001

Escuela Superior Politecnica del Litoral (Guayaquil, Ecuador)

Thesis: “Analyze, Design and Implementation of Technical Solution to extend the cover of backbone of he ESPOL by using wireless devices”. Advisor: Enrique Peláez, Ph.D.

WORK EXPERIENCE

Research Assistant 2015-2017

German Research Center of Artificial Intelligence

Member of the Multimedia Opinion Mining (MOM) project.

Project Leader 2006-2011

Information Technology Center (ICT), Full-time

I was in charge of managing a small group of pre-graduate students in order to implement software projects. Also, I was helping to write research proposals in order to obtain funding or grants.

Lecturer 2006-2011

Faculty of Electrical and Computer Engineering

I was a lecture of several courses during 8 semesters. The courses that I taught were Programming Foundations, Artificial Intelligent, Data Structures and Object-Oriented Programming.

Network Administrator 2003-2004
Humboldt German High-School, Full-Time

I was in charge of technical support for students, teachers and administrative staff. Also, I managed and configured all the servers and the network equipments in the high-school

Technical Support 1999-2003
Information Technology Center (ICT), Part-Time

I was in charge of technical and logistic responsible for videoconference sessions. Also, I managed and configured the servers and the network equipments in the high-school

AWARDS

ENNS Travel Grant ICANN2016 09/2016
Scholarship for attending ICANN2016 for presenting my paper <http://e-nms.org/student-grants/winners-2016/>.

PhD scholarship from DAAD 2011-2015
Scholarship for pursuing a PhD in Computer Science at TU Kaiserslautern.

SKILLS

- *Programming Knowledge:* Python(expert), Bash(intermediate)
- *Languages:* Spanish (native speaker), English(highly proficient), German(basic).

PUBLICATION LIST

Mozaffari Chaniyani, S. S., **Raue, F.**, Agne, S., Bukhari, S. S., & Dengel, A. (2018). Reading Type Classification based on Generative Models and Bidirectional Long Short-Term Memory. In User Interfaces for Spatial-Temporal Data Analysis Workshops (UISTDA), Proceedings of the 23rd International Conference on Intelligent User Interfaces.

Palacio, S., Folz, J., Hees, J., **Raue, F.**, & Dengel, A. (2018). What do Deep Networks Like to See. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Raue, F., Breuel, T. M., Dengel, A., & Liwicki, M. (2018). *Symbol Grounding Association in Multimodal Sequences with Missing Elements*. Journal of Artificial Intelligence Research (JAIR). Pre-print version at Arxiv (<http://arxiv.org/abs/1511.04401>)

Raue, F., Palacio, S., Dengel, A., & Liwicki, M. (2017). *Classless Association using Neural Networks*. ICANN 2017: 26th International Conference on Artificial Neural Networks.

Raue, F., Liwicki, M. & Dengel, A. (2016). *Symbolic Association Learning inspired by the Symbol Grounding Problem*. Workshop New Challenges in Neural Computation (NC²) at GCPR

Raue, F., Palacio, S., Breuel, T., Byeon, W., Dengel, A., & Liwicki, M. (2016). *Symbolic Association using Parallel Multilayer Perceptron*. ICANN 2016: 25th International Conference on Artificial Neural Networks.

Raue, F., Byeon, W., Breuel, T. M., & Liwicki, M. (2015). *Symbol Grounding in Multimodal Sequences using Recurrent Neural Networks*. In Workshop Cognitive Computation: Integrating Neural and Symbolic Approaches at NIPS.

Raue, F., Byeon, W., Breuel, T. M., & Liwicki, M. (2015). *Parallel Sequence Classification using Recurrent Neural Networks and Alignment*. In Document Analysis and Recognition (ICDAR), 2015 13th International Conference on (pp. 581-585). IEEE.

TALKS

Raue, F., Palacio, S., Dengel, A., & Liwicki, M. *Unsupervised Association using Multilayer Perceptron*. Online, Poster presented at Deep Learning Summer School (DLSS), 1st-7th August 2016, Montreal, Canada. Retrieved from <https://sites.google.com/site/deeplearningsummerschool2016>

Raue, F., Palacio, S., Breuel, T., Byeon, W., Dengel, A., & Liwicki, M. (2015). *Multimodal Symbolic Association using Parallel Multilayer Perceptron*. Online, Poster presented at Multimodal Machine Learning Workshop (MMML), 11th December, Montreal-Canada. Retrieved from <https://sites.google.com/site/multiml2015/>

