Speaker Recognition using Biology-Inspired Feature Extraction

EDVIN ANDERSSON
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Speaker Recognition using Biology-Inspired Feature Extraction

Edvin Andersson
elt15ean@student.lu.se

Department of Electrical and Information Technology
Lund University
—
IntuiCell

Supervisor: Fredrik Edman

Examiner: Erik Larsson

June 28, 2021

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Distinguishing between people's voices is something the human brain does naturally, using only frequencies picked up by the inner ear. The field of speaker recognition is concerned with making machines do the same thing using digitally sampled speech and data processing. The processing extracts relevant information about the speech from the high dimensional acoustic data which can help the machine understand to which speaker a speech sample belongs. Several methods exist to solve this problem, most of which are based on modelling a sample as a sequence of time frames, each representing the current frequency characteristics of the sound input. A common choice of frequency characteristics are Mel-Frequency Cepstral Coefficients (MFCC), which represent the overall shape of the frequency spectrum representation of the input during each time frame. This thesis presents a different approach, inspired by findings of how the human brain processes tactile sensory input, which lets an unsupervised learning model pick out important combinations of frequencies from the signal. These different combinations of frequencies arise because they have an observed spatiotemporal relationship across multiple data samples and speakers, in which their intensities correlate in time. Extracting spatiotemporal patterns between input frequencies as features instead of the overall spectrum shape can lead to new, more robust ways of encoding auditory data.

Acknowledgements

I would like to thank everyone at IntuiCell for showing such genuine engagement and support during the writing of this thesis. A special thanks to Udaya who showed great devotion helping me wrap my head around the model.

I want to give a big thank you to Fredrik Edman for his guidance and feedback during the writing.

I want to thank my parents Birgitta and Michael for taking care of the farm while I was immersed in this thesis.

Finally, I want to thank my partner Angelika for the loving support and encouragment I have recieved throughout the project.

Popular Science Summary

Recognizing the voices of people you know is probably something you take for granted. However, when considering the task in more detail it is astounding how the brain manages to distingush between the voices of almost everyone you've met solely based on the frequencies picked up by your inner ear. Based on recent insights into how the brain processes the sense of touch, this thesis presents a new way of approaching the problem of making sense of sound.

When you are recognizing someone's voice, you are not distingusishing one certain frequency associated with that person, but rather you recognize a combination of many different frequencies in intricate patterns over time that makes the voice sound familiar. These different frequency combinations are a result of a persons vocal tract which creates resonances at certain frequencies. A persons voice is therefore much like an acoustic fingerprint, as the vocal tract encompasses many different aspects of a persons physiology, from the shape of the tounge to the width of the nostrils. A fingerprint can be photographed and relatively easily reproduced. To reverse-engineer a persons vocal tract from speech is a much more difficult task. For this reason, your voice is a good biometric that can be used for authenticating yourself when accessing private information.

Speaker recognition is the scientific field of trying to determine who is speaking. Today this is normally done with machine learning techniques, in particular artificial neural networks. In order for these networks to easier process all of the intricacies of sound, only the very general shape of the frequency representation is used. This approach is efficient but far from how our own human hearing functions.

At the most basic level, the human brain processes sensory data through the activations of vast numbers of neurons. Recent findings has found that for the sense of touch, the brain might make sense of all this incoming data by learning to distinguish recurring temporal patterns in the activations. Based on these principles this thesis presents a new biologically inspired way of dealing with sound data.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Robust machine learning requires the ability to clearly distinguish between different patterns of incoming data and being able to separate recurring patterns from noise. This is mainly achieved through supervised tuning of parameters during training in order to minimize a certain loss function. Biological systems, in contrast, learn in an unsupervised manner to distinguish between sensory input. In the central nervous system, a huge amount of data is constantly being registered and processed in order to extract meaningful patterns. These sensory input patterns are spatiotemporal, i.e not only do the types of active neural pathways determine how the input is interpreted, but also the order and timing in which they activate. Given that biological systems successfully process large amounts of sensory data, an understanding of this process could lead to new ways of designing machine learning models. It is natural to apply such a biology-inspired processing approach to sensory-like data, since that is the domain it was evolved to function in.

In the case of auditory data, a lot of active research is being carried out in the fields of speech and speaker recognition. Most systems use recurrent or convolutional artificial neural networks (ANNs) for capturing the temporal context of sequential speech segments with good results compared to using conventional ANNs. With some exceptions, most of these ANN models operate on static features extracted from segments of sound containing speech. These features are low dimensional representations of the current auditory character of the sound, which process each feature independently from the others. A frequently used feature extraction technique is Mel-Frequency Cepstral Coefficients (MFCC) extraction which captures the general shape of the frequency spectrum of the sound at each time segment. The main disadvantage of this approach is that relevant time dynamics and/or finer grained acoustic features may be lost because they are invisible to the learning part of the network. Noise is also incorporated into the acoustic features to a relatively high degree. Some type of encoding which reduces the high dimensionality of the acoustic input is often needed to effectively process the data. However, preferably these encodings would be learned by some process which filters out noise and accounts for the temporal context of each feature.

This thesis aims to determine how biology-inspired data processing, based on recent insights into the cuneate nucleus' functioning in processing tactile input, performs compared to MFCCs in the field of speaker recognition. Translating the neurophysiology of the cuneate nucleus into a software model could lead to new ways of encoding high dimensional data with temporal components. The hypoth-

esis is that the same mechanisms through which the cuneate nucleus segregates tactile data can be applied to segregate acoustic data as well. The motivation behind this is that both types of inputs emerge from a large number of receptors which can be interpreted in terms of their spatiotemporal behavior. A model emulating these mechanisms could find encodings which discriminate between speakers when trained with speech data and could hence be used in the field of speaker recognition.

1.1 Background on Speaker Recognition

Speaker recognition (or voice biometrics) is the process of identifying a speaker based on speech. Raw speech audio is high dimensional sequential data from which the goal is to extract *speaker embeddings*. Speaker embeddings are numerical vectors that represent a point in a high dimensional space which can be used to determine how similar two utterances are depending on the distance between their embeddings. This method of extracting embeddings reduces the dimensionality of the input and eliminates the time-dependant component of the data since the whole utterance is condensed into one single vector. This speaker embedding can be used to distinguish between speakers or validate a claimed identity during an authentication session.

1.2 Background on the Cuneate Nucleus

Sensory information is processed very differently in the nervous system of animals compared to the usage of static feature extraction algorithms. The nervous system adapts to input and learns how to distinguish between stimuli in an unsupervised manner. In the case of tactile input, work done by Jörntell H. et al. [9] show that the nervous system is able to segregate between different types of haptic stimuli in the neurons of the cuneate nucleus (CN), a region which processes tactile inputs before they reach the cortex. The neurons manage to dynamically encode unique spatiotemporal patterns from the 10'000s of primary afferents (PAs) emerging from tactile sensors in the skin even though their receptive fields are similar. In essence, the high dimensional input features are encoded before being further processed by the brain.

1.3 Background on Auditory Perception

Similarly to tactile inputs which emerge from mechanoreceptors in the skin, auditory inputs emerge from thousands of hair cells in the cochlea. These hair cells sense vibrations of the fluid inside the cochlea through their stereocilia organelles which convert the vibration's mechanical energy to action potentials which travel to the the brain via the auditory nerve. These hair cells respond in correlation with their position inside the cochlea. The spiral shape of the cochlea acts as a low pass filter, the outer parts responds to high frequencies while the inner parts responds to increasingly lower frequencies. Hence the PAs emerging from the cochlea carry information about specific frequency components of the sound.

Chapter 2
Theory

This chapter will introduce the two feature extraction approaches to be compared which will be called MFCC and Cuneate Nucleus (*CN*) feature extraction respectively. The chapter starts with explaining the general state of the speaker recognition field and the theory behind human speech generation.

2.1 Speaker Recognition

The performance of a speaker recognition system depends mainly on the *speaker transform* and the resulting *speaker embedding*. The speaker transform is a way of mapping the high dimensional acoustic input from an utterance to a low dimensional utterance-level representation of the speaker called the speaker embedding.

The utterance is usually divided into smaller time frames which represents the state of the frequency spectrum at each time instant. The process of applying the speaker transform in order to yield a speaker embedding is shown in figure 2.1. The figure illustrates how the utterance is divided into segments called *frames* which are created by sliding a *window function* across the signal with a certain step length. The signal is multiplied by the window function at each time step to create individual frames where the amplitude of the signal gradually diminshes closer to the edges of the frame. Since a Fast Fourier Transform (FFT) is usually applied to the frames, the window function helps to reduce *spectral leakage* resulting from discontinuities at the frame edges. The features associated with each frame are called *frame level features* to emphasize that they only carry information about that particular time frame and not the whole utterance.

The speaker embedding is an n-dimensional vector of real valued numbers where embeddings of the same speaker are ideally close in the n-dimensional space and distant for different speaker. This means that the quality of a speaker transform depends on the ability to minimize the distance between different utterance's speaker embeddings from the same speaker and maximizing the distance between speakers. Doing so creates data clusters from which a classifier clearly can distinguish between speakers.

There are two subcategories of speaker recognition, *speaker identification* and *speaker verification*. Central to both categories is the concept of a *speaker model*, this model is a general representation of a speaker that is created based on previously recorded utterances from that speaker. This representation can be a numeri-

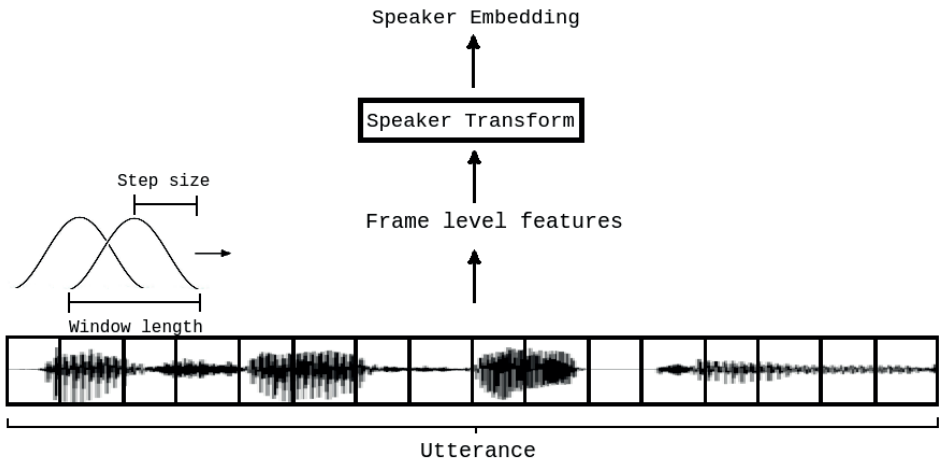


Figure 2.1: Illustration of the process of creating the frame level features used by the speaker transform

cal vector or integrated into the layers of a neural network. Speaker model vectors are usually created during an *enrollment phase* where the average of a number of utterance embeddings are used as a speaker model. For an integrated approach a neural network is simply trained to categorise an utterance into a fixed speaker set.

Speaker identification is the process of identifying an unknown speakers' utterance as coming from some speaker in a fixed set of registered speakers. The speech utterance is compared with speech models from known speakers and the unknown speaker is identified as the speaker whose model had the best match. This problem is suitable for an entirely integrated approach using a neural network because no embeddings needs to be extracted since the classes are limited to the classes used during training of the ANN.

Speaker verification is the process of accepting or rejecting a claimed identity of an unknown speaker based on if an utterance matches a predetermined speaker model well enough. Usually, this speaker model is a vector created by averaging a number of embeddings during the enrollment phase. The performance of a speaker verification system is often measured in its Equal Error Rate (EER) which is the point at which the amount of false accepts and false rejections are equal. This measure is needed since the threshold for accepting a claimed identity can be changed depending on if the desired system should be strict or loose. The EER rate is a way of standardising the performance measure of speaker verification models so that they can be compared more easily. This thesis is exclusively concerned with the speaker verification problem.

2.2 Embedding Model Variations

Until a few years ago, the state-of-the-art embeddings were created using classical statistical methods such as Gaussian Mixture Models (GMM) combined with Universal Background Models (UBM) which were used to model the sequence of

frame level features extracted from audio input. Recently however, speaker recognition systems have undergone a fundamental shift from statistical methods to Deep Neural Network (DNN) embedding extraction which was first proposed by Variani et al. [2]. These embeddings are called *d-vectors* and are based on extracting features from the last hidden layer of a DNN. Features extracted in this way carry condensed information from the previous layers which the DNN uses in order to make a correct classification in the final layer. Typically, the final layer is a softmax layer which turns the output from the previous hidden layer to a probability distribution over the classes in the training set. In speaker recognition each speaker belongs to one class. The underlying hypothesis of this method is that the trained neural network has learned efficient representations of the speakers in the training set which can be generalised to represent also unknown speakers. The cosine similarity metric (which is discussed in more detail in chapter 3, section 3.1) is often used to define how well two DNN embeddings match.

In order to obtain a single utterance-level speaker embedding the frame-level features need to be either concatenated and stacked at the input or fed to the model sequentially to let the model handle the temporal context. The former is the method of choice for Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) architectures while the latter is used for Recurrent Neural Networks (RNN). Generally MLP models have been outperformed by other architectures, such as Long Short-Term Memory (LSTM) [3, 4], Time Delay Neural Network (TDNN) [5] and CNN [6] based models.

2.3 Input Feature Extraction Variations

In most speaker recognition systems the speech input is divided into a number of *frames* using a $25ms$ sliding window from which acoustic features are extracted, usually binned frequencies from the spectrum representation of the input. Further feature extraction from the frequencies is often carried out for dimensionality reduction since the frequency data can be too high-dimensional to be used directly with machine learning methods. However, since the advent of deep learning networks which can handle higher dimensional input, using the binned frequency spectrum directly as features has become increasingly popular.

Despite the shift in architecture the feature extraction step in most models have remained the same, namely extracting frame-level Mel Frequency Cepstral Coefficients (MFCC). MFCCs are *static* acoustic features which capture the general shape of the frequency spectrum (spectral envelope) at each time frame. The MFCC features are static in the sense that the same feature extraction steps are carried out regardless of the temporal context of the feature.

2.4 Speaker Features

To understand the reasoning and efficacy behind MFCCs and other acoustic feature extraction methods it is useful to model human speech generation as a signal convoluted with a filter. This model is called the *source-filter* model. The signal represents the *glottal pulse*, the frequency that originates in the vocal cords. This signal carries information about the speakers pitch. The filter represents a persons vocal tract which transforms the glottal pulse into a more characteristic sound dependant on several factors, including nasal cavity shape, teeth placement, tongue shape etc. From a speaker recognition viewpoint the nature of the vocal tract filter should be accentuated in order to find the most discriminative features. Pitch information can also be important, however, the information that it carries is not as rich as the vocal tract filter. Figure 2.2 illustrates the analogy between human speech generation and convoluting a signal with a filter. The resulting peaks in the frequency domain results from resonance in the human vocal tract and are called *formants*. These high energy frequencies determine the distinct sound of the speech. For example, the approximate frequency positions of the first two formants (F_1 and F_2) are sufficient to distinguish between most spoken vowels. This is a fundamental functional principle in speech recognition since formant placements determine which vowels and by extension which words are spoken. There exists variations of formant positions between speakers for the same vowels. In particular the higher order formants (F_3, F_4, F_5 etc.) are mainly involved in producing the distinct voice for a speaker. Consonants do not produce the same kind of resonance in the vocal tract but instead produces a noise-like peak in the higher end of the frequency spectrum. These consonant frequencies can in principle also carry speaker specific information.

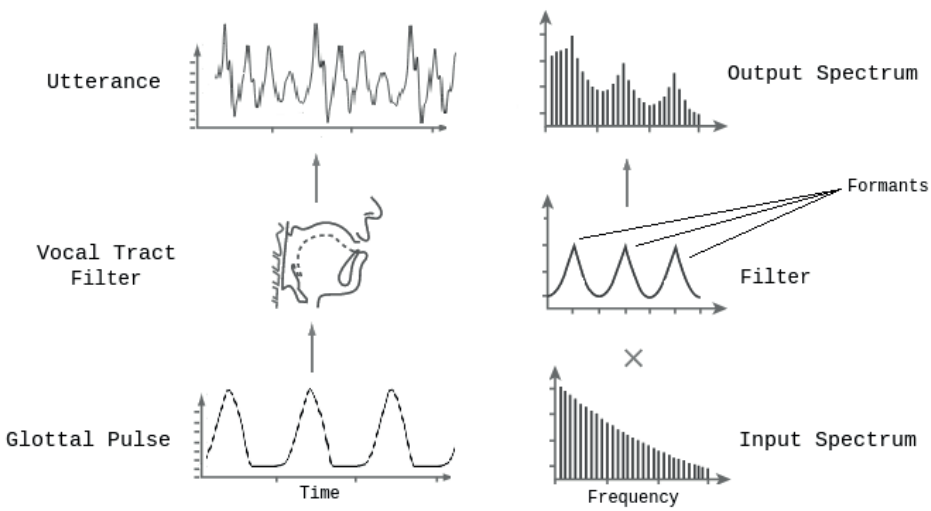


Figure 2.2: Source-filter theory analogy between human vocal tract and frequency filtering

2.5 MFCC Feature Extraction

MFCCs are commonly used in the field of speech recognition and speaker recognition. They capture the identity of sound relevant to human hearing by representing the general shape of the frequency spectrum with frequencies scaled to mirror human auditory perception. Figure 2.3 illustrates the different steps in processing audio into its MFCCs.

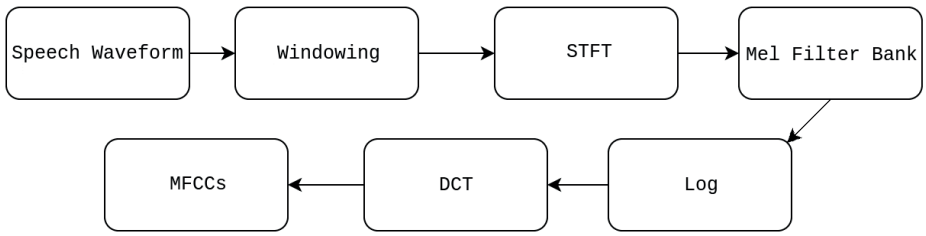


Figure 2.3: Substeps for MFCC extraction

First the audio signal is divided up into frames of typically $25ms$ using a window function. This time span is widely used in the speech analysis field based both on qualitative arguments and more systematic studies [8]. The windows are converted to the frequency spectrum using the Short-Time Fourier Transform (STFT). To better mirror the auditory perception of humans the frequencies are mapped to the *mel* scale through a triangular filterbank. The mel scale is an empirically deduced logarithmic frequency scale which is more dense at lower frequencies, defined as in equation 2.1 below.

$$f_{mel} = 2595 \cdot \log_{10}(1 + \frac{f_{Hz}}{700}) \tag{2.1}$$

Figure 2.4 shows an example of a mel-spaced filterbank which creates a number of weighted averages of frequencies in different frequency bins, these bins and the weight distribution within them are represented by the triangular filters seen in the figure. Each gray-scale coloured triangular filter represents one mel-frequency bin. The overlapping of the filters causes the data to be smoothened and the bins to be correlated. The amount of frequencies per bin is smaller at lower frequencies in order to mimic the observation that human hearing is more sensitive to changes in pitch at low frequencies. The filterbank reduces the dimensionality of the input from the number of frequencies to the number of filters. The logarithm of the spectrum is taken which more closely resembles the way humans perceive sound intensity depending on frequency.

To yield a representation of the general shape of the spectrum (spectral envelope) the discrete cosine transform is applied. This is a way to reduce the dimensionality of the data and to decorrelate the frequencies because of the overlapping triangular filterbanks. The result is a representation of the frequency spectrum in the quefrency domain as a function of the cosine basis functions of different periods. The quefrency domain represents the frequencies present in a spectrum when viewed as a time-domain signal. The lower end of the quefrency axis car-

ries information about the general shape of the spectral envelope while the high end captures the more fast changing spectral details. Usually, the higher cepstral coefficients are discarded as they incorporate too much noise. The benefits of adding higher level MFCCs usually diminishes after approximately the first 10-20 coefficients depending on the application and required level of detail. The MFCC coefficients are usually used as sequential input to machine learning algorithms which classify the audio sequence as coming from a certain speaker (in the case of speaker recognition) or as corresponding to different words (speech recognition).

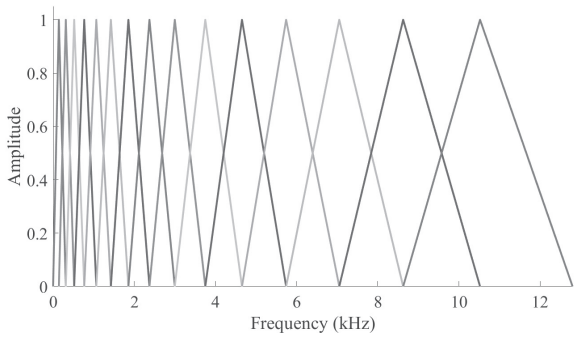


Figure 2.4: Mel spaced triangular filterbank with 15 filters. Each gray-scale coloured triangle represents one frequency bin.

2.5.1 MFCC Deltas

An improvement to static MFCC extraction is to incorporate the first (Δ) and second ($\Delta\Delta$) order derivatives of the coefficients with respect to the discrete time steps. This means that the feature vector carries explicit information about the changes in individual quefrencies of the cepstrum. This addresses some of the temporal context of the input frames. However, the temporal information is limited to the individual changes in each feature and does not capture any temporal relationships between the features. Furthermore, the $\Delta/\Delta\Delta$ features are linear transformations of the input features which in principle can be performed inside the hidden layers of ANNs.

2.6 Cuneate Nucleus

The neurons in the cuneate nucleus (CN) form an interface for tactile sensory input before it reaches the brain cortex. Large numbers of primary afferents (PAs) emerging from receptors in the skin reach the CN which processes the input before relaying it on to further brain networks. Results from in-vivo experiments [9] has shown that the cuneate neurons are able to segregate between different types of tactile stimuli despite sharing a similar receptive field. The neurons seemingly respond to unique combinations of inputs in time, capturing spatiotemporal patterns in the PAs. One of the main characteristics of the cuneate neurons is an almost

binary division of the synaptic weights into high and low weight synapses [10], where low weight synapses are most prevalent. In practice this means that the neurons are mostly responsive to a very sparse set of synapses. This sparse weight distribution indicates learning of the synaptic weights and may have an important role in the cuneate neurons decoding of tactile PAs. The excitatory synapses are observed to have a specific impact for a given neuron, while the inhibitory synapses are relatively uniform and low weighted [11] which means that their effect on the neuron is primarily as a group. Presumably, these synaptic weights are strengthened and weakened gradually through conventional correlation-based synaptic plasticity (*Hebbian learning*).

2.7 Cuneate Nucleus Model

The Cuneate Nucleus model (CN model) is a biology inspired feature extractor emulating the mechanism through which the neurons of the cuneate nucleus encode tactile inputs. The main function of the model is to capture lower dimensional representations of certain spatiotemporal patterns present in input. This model consists of a set of neurons, each connected to a fixed set of artificial PA inputs through a number of synapses. The synapse connections are illustrated in Figure 2.5. A central concept for the model is the *activity* of the neurons, which is summarized as the concentration of calcium ions (Ca^{2+}) in the main compartments of the neurons. This calcium activity is refered to as the Ca^{2+}_{tot} *activity* and is the numerical output reading from the CN model. The Ca^{2+}_{tot} activity is a non-linear function of the local calcium (Ca^{2+}_{loc}) activities in each synaptic space and several intrinsic parameters. The Ca^{2+}_{loc} activity in a synapse is equal to the amplitude of the incoming PA multiplied by the synaptic weight of the synapse. The input A to the main compartment Ca^{2+}_{tot} activity from the n synapses is hence a function of the incoming PA activity to each synapse a_i and its synaptic weight w_i according to 2.2.

$$A = \sum_{i=1}^{n} w_i \cdot a_i \tag{2.2}$$

The input from the synapses A affects Ca^{2+}_{tot} activity through a dynamic relationship called the dynamic model. This model has two main functions, to accentuate sudden increases in activity and to enter into a state of afterhyperpolarization (AHP) after a period of high activity. During the AHP phase it is harder for Ca^{2+}_{loc} activities to raise the Ca^{2+}_{tot} activity level which mimics the way biological neurons function after they have fired. This behaviors results in more distinct learning which is highly dependant on the dynamics of the input PA information. There are two kinds of synapses, *excitatory* and *inhibitory*. Excitatory synapses contribute to a positive change in Ca^{2+}_{tot} activity while inhibitory synapses contribute to a negative change in Ca^{2+}_{tot} activity. To mimic the findings from the in-vivo experiments [11], the excitatory synapses are specific to one PA. In contrast to the excitatory synapses, each CN has one inhibitory synapse which instead is receiving information uniformly across the whole batch of PAs, i.e the inhibitory synapse calcium activity is equal to the sum of all PAs. It is assumed that the high weight excitatory synapses generally have correlated PA activity i.e they respond to specific patterns present in the corresponding PA for all types of stimuli [11].

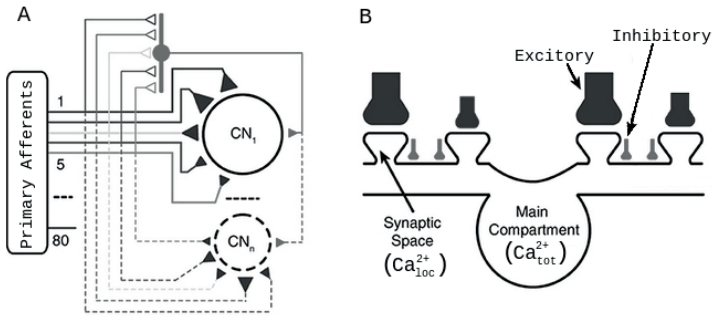


Figure 2.5: Synapse connections for the cuneate neurons. A illustrates the difference between excitatory (dark gray) and inhibitory (light gray) synapse connections. B shows the terminology for the neuron compartments.

2.7.1 Dynamic Feature Extraction

Used as an unsupervised feature extractor, the CN model output is dependant on both the spatial and the temporal components of its input features. This allows for large amounts of information to be extracted from time-changing inputs as a function of both amplitude, rate of change, timing and the internal state of the CN. Each output state will be highly dependant on the previous context, in particular, each PA's effect on the output will be determined not only by its own previous context, but also the previous context of every other PA. This allows the model to gradually become responsive to spatiotemporal relationships between PAs during the correlation based (Hebbian) learning.

2.7.2 Learning of the Synaptic Weights

The synaptic weights of the CN model neurons are gradually learned based on a number of parameters. An overview of the learning model can be seen in Figure 2.6. The figure illustrates the mechanisms which determine the changes in the synaptic weights. The underlying principle is the Hebbian learning rule, meaning that weights of synapses whose activations are correlated with high total Ca^{2+}_{tot} activity are increased (*Long Term Potentiation*, *LTP*) while decorrelated synapses lower their weights (*Long Term Depression*, *LTD*). The inhibitory synapse weights are an exception in that they simply increase/decrease by a fixed magnitude in order to lower the mean Ca^{2+}_{tot} activity to a desirable level called the calcium set point (Ca^{2+}_{set}), which is discussed later in this section. The inhibitory weights are bound in the interval $[-1, 0]$. For excitatory weights, the factors which directly affect the rate of the learning are the degree of correlation, the *excitatory learning rate* (r_{exc}) and the current weight of the synapse. Excitatory synaptic weights exist in the interval $[0, 1]$ and are most susceptible to change at low weights according to a function yielding the *synaptic weight compensation* (K_{comp}) shown in figure

2.7. The change in the weight is proportional to K_{comp} which is a function of the current weight. This model increases the synaptic plasticity of low weight synapses in order for the model to be susceptible to novel input patterns. The *excitatory learning rate* is an intrinsic parameter optimized in order to maximize the amount of samples (*stimulus representations*) processed before the weights converge to the low/high weight binary division. This is in order for the model to generalize better and not be overly biased by the types of stimulus presented in the beginning of learning. Since all neurons in the model share the same behavior, the initial weights of the synapses (seed weights) are different for each neuron. The seed weights are randomly generated based on a log-normal distribution and is the reason why the neurons potentiate different combinations of synapses during training.

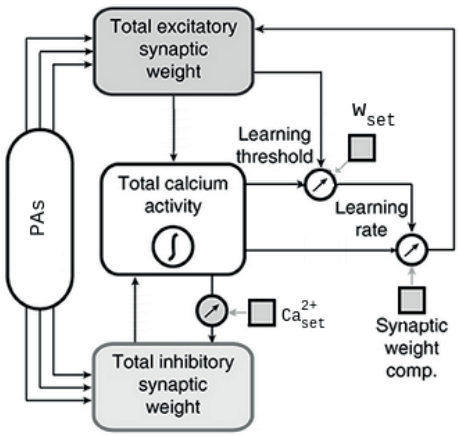


Figure 2.6: The learning model. Small squares indicate static hyperparameters.

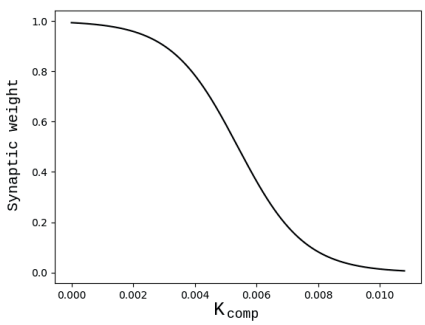


Figure 2.7: Weight compensation

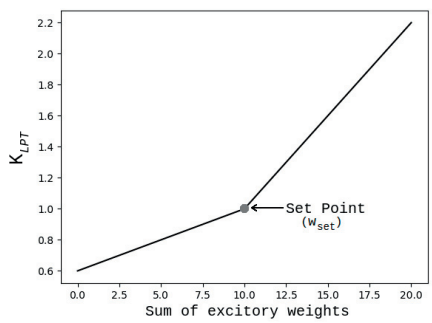


Figure 2.8: Learning threshold gain

In order to define *correlation* more precisely, thus preventing insufficiently correlated synapses from potentiating a *Learning Polarity Threshold* (LPT) is dynamically adapted for the Ca^{2+}_{tot} activity during training. This threshold is set so as to keep the total sum of excitatory synaptic weights around a predefined level called w_{set}. This mechanism keeps the synaptic weights in a desirable operating point by increasing the threshold for correlation if the weights increase beyond the set point and vice versa. The learning polarity threshold is defined by equation 2.3 below.

$$LPT = mean(Ca^{2+}_{tot}) \cdot K_{LPT} \tag{2.3}$$

The gain factor, K_{LPT}, is defined by two linear slopes shown in figure 2.8. The figure shows the excitatory synaptic weight set point w_{set} which is located at the intersection of the two slopes. The mean Ca^{2+}_{tot} activity is defined as the average of the three most recent means in order to smoothen the threshold changes between stimulus presentations. A corresponding correlation threshold, *Local Activity Threshold* (LAT) is also used for all Ca^{2+}_{loc} activities, which is statically set so that only a certain upper range of the local activity is considered active for the learning part of the model. This threshold can be set freely depending on input characteristics in order to determine what input amplitude is significant enough for affecting the learning. In order for correlation to occur both synaptic (LAT) and main compartment (LPT) thresholds must hence be exceeded by the local and total activity respectively. Decorrelation occurs when the local Ca^{2+} threshold (LAT) is exceeded in the synapse but the Ca^{2+}_{tot} activity does not exceed the LPT threshold.

The weight change Δw is defined as in equation 2.4 below. For each synapse, the instantaneous correlations are integrated in order to determine the degree of correlation for each full stimulus presentation ($t \in [0, t_{max}]$). The resulting integral is then multiplied by the *synaptic weight compensation* and the *excitatory learning rate* and the resulting weight change is added/subtracted to the previous synapse weight. An example of such a correlation process is illustrated in figure 2.9.

$$\Delta w = r_{exc} \cdot K_{comp} \cdot \int_0^{t_{max}} (Ca^{2+}_{tot}(t) - LPT) \cdot max\{Ca^{2+}_{loc}(t) - LAT, 0\} dt \tag{2.4}$$

The inhibitory weights are another control mechanism in order to keep the mean Ca^{2+}_{tot} activity around the Ca^{2+}_{set} setpoint. This is required in order to prevent the Ca^{2+}_{tot} activity from being too high which prevents the desired dynamic model behavior. If the mean Ca^{2+}_{tot} level exceeds Ca^{2+}_{set} the inhibitory weight is increased by a fixed small change and vice versa. This change is equal to an intrinsic parameter called the *inhibition learning rate* (r_{inh}).

A summary of all the tunable hyperparameters and their main functions are shown in table 2.1.

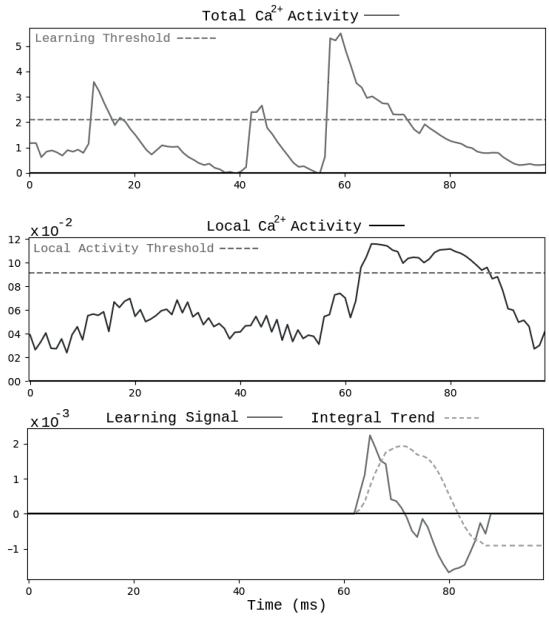


Figure 2.9: Graphs showing a net negative correlation between local Ca^{2+} activity and total Ca^{2+} activity.

Hyperparameter	Function
Excitatory learning rate (r_{exc})	Controls the rate of change for the excitatory weights
Excitatory weights setpoint (w_{set})	Controls the sum of the excitatory weights
Inhibitory learning rate (r_{inh})	Controls the rate of change for the inhibitory weights
Local activity threshold (LAT)	Defines the lowest activity limit for learning
Calcium setpoint (Ca^{2+}_{set})	Controls the Ca^{2+}_{tot} activity

Table 2.1: Overview of the tunable model hyperparameters

2.8 Theoretical Comparison Between MFCC and Cuneate Nucleus Feature Extraction

Whereas MFCC coefficients are a compact way of representing the spectral envelope, the CN model is learning spatiotemporal patterns in the power spectrum. One of the main theoretical differences between the two feature extraction approaches is the degree to which the temporal context affects the output. MFCC coefficients are static in the sense that they capture the cepstral state of each time frame without any notion of its temporal context, instead the context is usually

handled by other models such as RNNs. The output from the CN model neurons given a frequency input is highly dependant on the previous inputs which has changed the internal state of the model.

MFCC coefficients can be complemented with delta coefficients, as discussed in section 2.5.1, which represent the first and second order derivatives for each coefficient which provides temporal context. Because the delta coefficients are a linear transformation of the features they are in theory redundant when combined with any densely connected ANN layer though they can still result in more robust performance. Deltas does not provide any information about the temporal relationships between the different coefficients whereas the CN model output is dependant on the timing of increased activity between the PA frequencies. Certain combinations of activity behaviors in PAs would be required in order for the model neurons to respond with a substantial increase in total activity. In essence this is a way to encode certain spatiotemporal patterns between PAs.

The CN model is unsupervised, just like MFCC extraction, but it adapts the neuron responses to the dataset inputs during training without the need for the dataset's target classes. The patterns that the model becomes sensitive to can be present in multiple speakers however the assumption is that the combinations of outputs from the neurons can as a group distinguish between speakers. For example certain combinations of neurons might be particularily active during an utterance from a specific speaker but not for others, or the timings of the neuron outputs may differ.

MFCC noise can be reduced by discarding the higher order cepstral coefficients as they become increasingly more susceptible to fast changing spectral details. This process functions as a low pass filter, enabling more robust learning but also reducing the amount of finer grained details in the data. The CN model diminishes noise impact during training through lowering the synapse weights of PAs that usually are irrelevant to the main compartment Ca^{2+} activity and requiring the inputs to exceed the synaptic LAT threshold to have an effect on learning. Because the model is sensitive to the timing of increased inputs, i.e temporal patterns, it should in theory not respond to sporadic inputs such as noise.

Chapter 3

Experimental Setup

3.1 Network architecture

In order to compare the effects between using CN model feature extraction and MFCCs two speaker verification models were created which were comparable in every aspect after the feature extraction step. The difference in the feature extraction outlines for the two approaches can be seen in figures 3.1 and 3.2. The main part of the models is the neural network which was based on previous work done with the d-vector approach to speaker recognition, primarily Apple's 2018 paper from the ICASSP conference [3]. This neural network will be called the *embedding network* since it is generating the speaker embeddings.

The embedding network seen in Figure 3.3 and the subsequent classifier were kept the same for both comparisons. The embedding network is an ANN with an LSTM-Linear-Softmax architecture. The size of the linear layer determines the size of the speaker embedding and is fixed at 128 nodes which yields a numerical speaker embedding vector of size 128. The size of the linear layer was chosen based on Apple's model implementation. The size of the softmax layer is equal to the number of speakers in the training set. The LSTM layer size is 512 units and only the last output of the LSTM layer is fed to the densely connected linear layer in order to obtain a single, utterance-level embedding. Like in Apple's model [3] the LSTM activation function is the *tanh* activation function, adam optimization is used for batch gradient descent and categorical crossentropy is used as loss function.

The embedding classifier used is the cosine similarity metric which is defined as the cosine of the angle θ between the speaker model A and the current embedding B which is given according to equation 3.1.

$$Similarity\ score \equiv cos(\theta) = \frac{A \cdot B}{\| A \| \| B \|} \tag{3.1}$$

The similarity score ranges between -1 and 1 where a higher score means that the angle between the two length normalized embeddings is smaller and thus they are considered to be more similar.

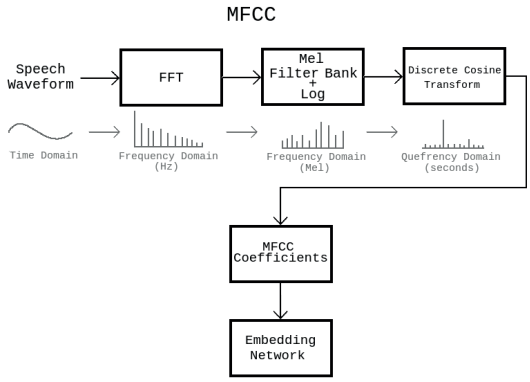


Figure 3.1: MFCC feature extraction outline

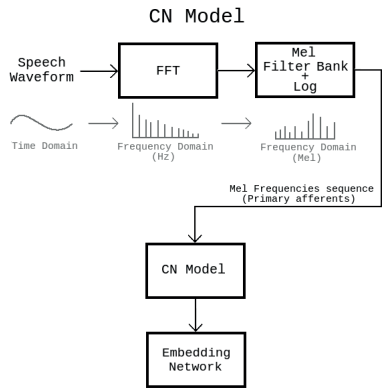


Figure 3.2: CN feature extraction outline

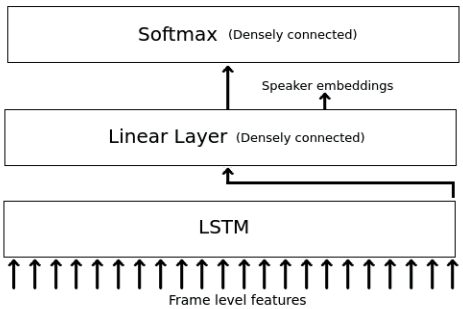


Figure 3.3: Embedding network

3.2 Dataset

The dataset used in the comparison was Google's *Speech Commands* which consists of one word audio samples from a list of 10 short words uttered by different speakers. The dataset is hence semi-text dependent since the same words are uttered by every speaker. There exists repetitions of the same word by the same speaker. For the training and testing of the embedding network 1'500 and 270 speakers were used respectively, totaling in 46'320 and 9'309 number of utterance samples respectively with an average number of samples per speaker of 30. The training of the CN model used 150 speakers with a total of 4'549 utterance samples where each sample is refered to as a *stimulus presentation*. Because speaker verification is an open-set problem where speakers not included in the training set should be able to generate effective speaker models, the training and testing sets consisted of separate speakers. The sample rate of the dataset is $16'000\,Hz$ which meant that the upper frequency limit for the filterbanks was $8'000\,Hz$ according to Nyquist's theorem. The used range of frequencies for all models was hence chosen between 0 and $8'000\,Hz$ in order to maximize the amount of frequencies considered.

3.3 MFCC Model

The MFCC model uses a $25ms$ sliding Hanning window with a step size of $10ms$, which are well-established time parameters for MFCC coefficients. The number of mel-filterbanks used were 26 which is a common MFCC filterbank count for the range 0 to 8'000Hz. As in the Apple implementation, 20 mean-normalized MFCCs were extracted from each frame and these were used as sequential input to the LSTM layer. With the added Δ and $\Delta\Delta$ coefficients, representing the first and second order derivative for each original coefficient, 60 features were extracted in total.

3.4 CN Model

The CN model uses a smaller window and step size ($10ms$ and $4ms$ respectively) than the MFCC model and operates on the logarithmic mel-frequency binned spectrum. The reason for the shorter time frames was to have a higher temporal resolution in order to capture more fine-grained input patterns. A total of 100 mel-filter banks were used so that 100 input sequences representing the changes in the energy of each frequency bin were used as input to the CN model. The reason for using more filterbanks than for the MFCCs was to present the model with more narrow frequency ranges for the PAs to make the learning more specific. Furthermore, the inclusion of a large number of PAs more closely mimicks the functionning of the biological cuneate nucleus. Each neuron in the CN model was connected to all of the 100 inputs but the initial synaptic seed weight for each neuron was randomly assigned using a log-normal distribution (mean = 0, variance = 1). The frequency input signal was preprocessed in order to remove low level amplitude noise and accentuate significant increases in amplitude. Both 10 and 40 neurons were used in different tests in order to evaluate the effect of the number of neurons on the performance. In order to examine the learning behavior of the model and to tune the model hyperparameters the synaptic weights were plotted during training. The seed and end weights were saved in order to determine how the PAs contributed to the weight outcome.

3.5 Evaluations

3.5.1 Dataset Performance

In order to draw conclusions about the performance of the CN model in comparison to MFCC extraction, different Equal Error Rate (EER) performance evaluations were made. One evaluation with a *clean* test set (without any added noise) and several others with varying levels of Signal-to-Noise Ratios (SNR). The noise added to the utterance samples was additive white gaussian noise and the SNR ratio for each sample is defined as in equation 3.2. The performance of a baseline CN model with randomized end weights was examined in order to determine if the learning process was succesfull in choosing the most relevant combinations of end weights. A comparison using the mel filterbanks directly as features to the embedding network was also carried out.

$$SNR = \frac{P_{sample}}{P_{noise}} = \frac{RMS^2_{sample}}{RMS^2_{noise}} = 10 \cdot log_{10} \frac{RMS^2_{sample}}{RMS^2_{noise}} \quad [dB] \tag{3.2}$$

3.5.2 CN Model Behavior

To determine which role the seed weights had on the end weights outcome these were compared after learning in order to find any relationship between them. The end weights of different neurons were also compared to determine if the neurons potentiated unique combinations of synapses.

The difference in the degree of correlation between PA activities of synapses that ended with High End Weights (HEW) and synapses that started with high seed weights but ended with Low End Weights (LEW) was examined to determine if HEW synapses potentiated in relation to the underlying degree of correlation between the HEW synapse's PAs. Correlation here refers to the overall overlap in activity between two PAs. The hypothesis was that HEW synapses would have more correlating PAs on average which would help to explain the end weight distribution since synapses that are active together might have a higher probability of having rises in activity simultaneously. The desired behavior of the model would mean that if two synapses rise in activity together they are more likely to increase the Ca^{2+}_{tot} activity and hence both potentiate together. The level of correlation between two PA sequences was determined based on the *numpy* function *correlate* which computes the cross correlation of two 1-dimensional sequences.

Chapter 4

Results

This chapter presents the results regarding both the general behavior of the CN model in section 4.1 and the performance comparisons between the models in section 4.2.

4.1 CN Model Behavior

4.1.1 End Weight Distributions

The end weight distributions for the CN model provides indications of potential bias towards potentiating certain synapses. Together with the seed weights, the end weights also show the impact of the initial weight configuration on the final synaptic weight distribution. Figures 4.1 and 4.2 are bar graphs showing the synaptic weights of the seed weights (light gray) and end weights (dark gray) for all synapses ranging from synapses with low frequency PAs to high frequency PAs. The lowest weight bars are shown in the foreground to illustrate where bars overlap.

An example of seed and end weights for a neuron can be seen in figure 4.1 where the frequencies range from 0 to 8'000Hz on the mel scale. One of the observations from the results was a tendency for the model to potentiate synapses within particular subregions (bands) of the frequency spectrum. These frequency bands varied depending on the seed weight configuration, where a high inital weight distribution in an area encouraged the model to potentiate the synapses there. When the seed weight distribution was defined as the log-normal distribution across the whole spectrum the model tended to frequently potentiate the highest available frequencies. In order to steer the model into choosing frequencies more evenly across the spectrum the seed weights were adjusted to have a higher mean in the lower half of the spectrum (with a mean of 1 instead of 0). The specific combination of potentiated synapses were different for each synapse due to the different seed weights. This is illustrated in figure 4.2 where four neuron's end weights are presented from the same CN model training.

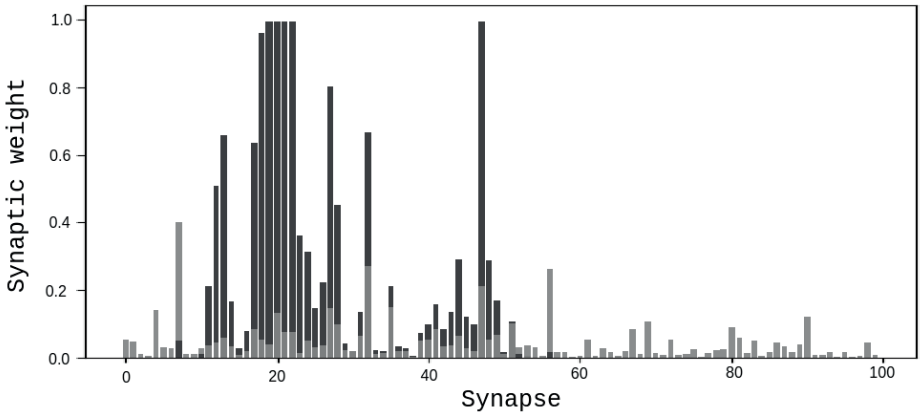
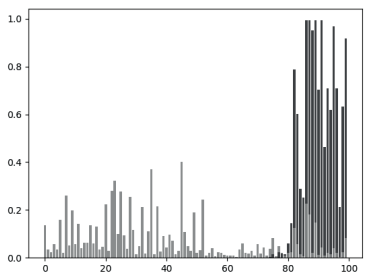
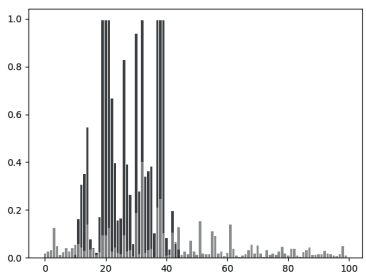


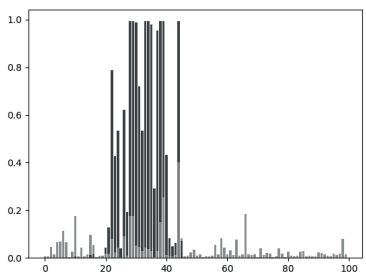
Figure 4.1: Seed weight (light gray) and end weight (dark gray) for each synapse of a neuron with PAs ranging from 0 to 8000 *Hz*. The lowest weight bars are shown in the foreground.



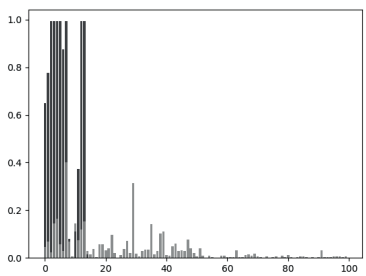
(a) CN1



(b) CN2



(c) CN3



(d) CN4

Figure 4.2: Seed weight (light gray) and end weights (dark gray) for four different neurons during the same CN model training. The lowest weight bars are shown in the foreground.

4.1.2 Seed Weight Bias

The seed weights' role is to steer the CN model neurons into finding different temporal patterns. However, an end weight distribution that is entirely determined by the seed weight magnitude of each synapse is an indication that the learning was too strongly dictated by the seed weights.

There was no bias towards potentiating the highest seed weight synapses across the whole frequency range which can be deduced from the weight distributions in figures 4.1 and 4.2. This behavior is desired but it is possible that one contributing factor is the potentiation of synapses in distinct frequency ranges and the depression of both high and low weight synapses outside this range. Within the HEW synapses there were clear signs that high seed weight synapses were generally more potentiated. Figure 4.3 shows an example of how the synaptic seed weights progressed towards the end weights during training of a neuron. It can be noted that several high seed weight synapses gets depressed during training but they generally belong to synapses outside the neurons particular frequency lump.

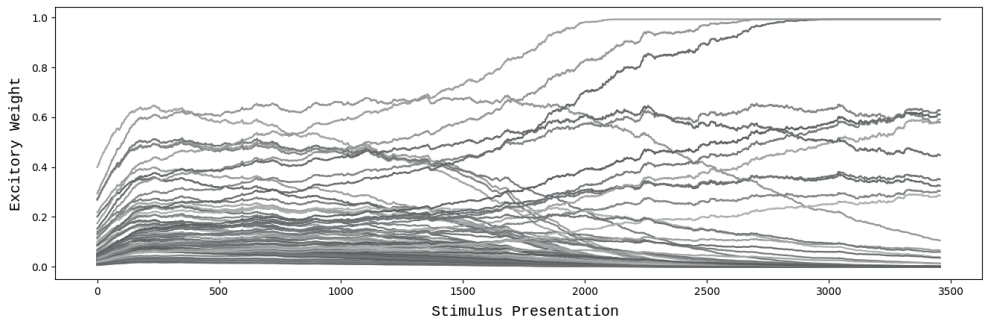


Figure 4.3: Example of synaptic weight progression during learning for a neuron

4.1.3 Correlation Index

The correlation analysis is illustrated by the box plot in figure 4.4 which shows the median degree of correlation for each HEW (black) and LEW (gray) synapse for 30 different seed weight configurations. The degree is represented by the *correlation index* which results from the average correlation of each synapse PA with each of its own HEW/ LEW weight category synapses for every stimulus presentation. The hypothesis that the HEW synapses would have more correlating PAs on average could not be confirmed from the figure but there were clear indications that the HEW synapses had a larger variance in correlation index. This shows that the model potentiated synapses which had a wide variety of correlation amongst themselves and/or depressed synapses which were evenly correlated with each other. This suggests that the model can find spatiotemporal patterns in both generally overlapping and non-overlapping PAs.

The correlation results showed no clear signs of being related to the frequencies of the HEW synapses which was notable since the higher frequency synapses tended to potentiate without any modification of the seed weight configuration. This

could indicate that there was something not captured by the correlation index which caused high frequency synapses to potentiate, which by extension could indicate that the lack of overlap between HEW synapse PAs does not suggest that the learning behavior was undesirable.

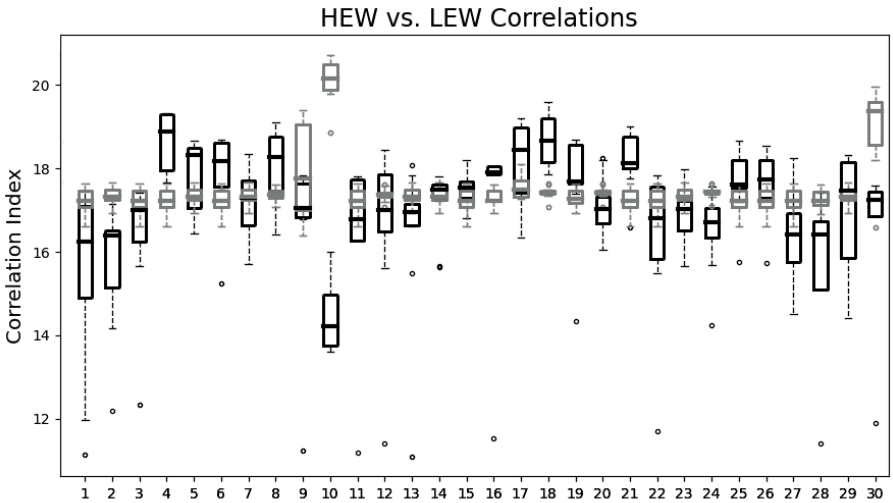
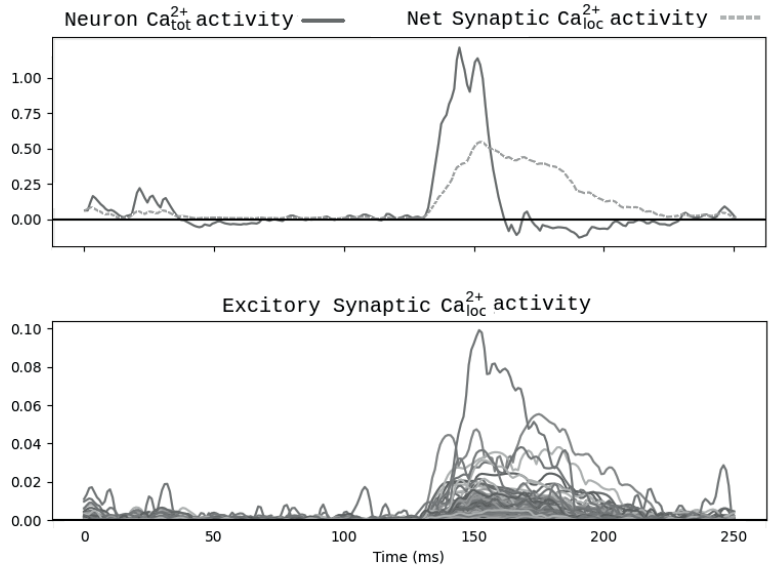


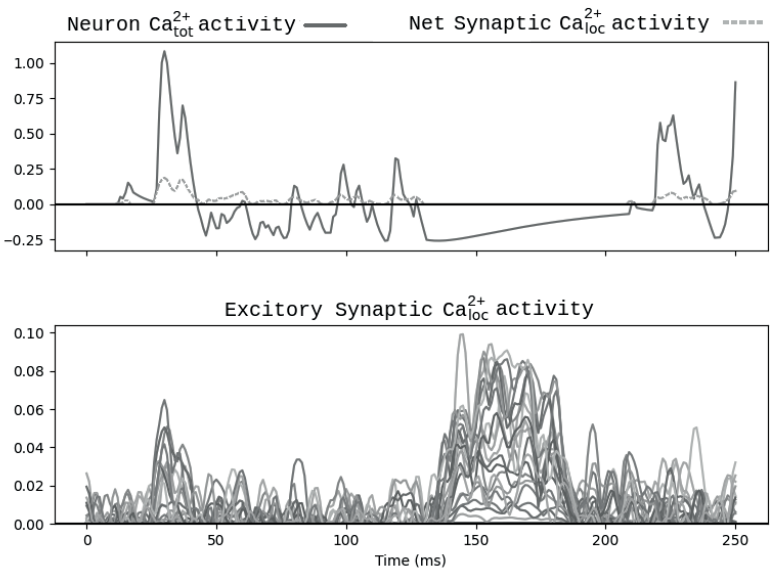
Figure 4.4: Correlation analysis of HEW (black) vs LEW (gray) synapses for different seed weight distributions. Circles indicate data outliers.

4.1.4 Neuron Activation Behavior

The total calcium activity in the neurons behaved in the intended way with sudden activations during short time segments followed by a period of afterhyperpolarization (AHP). During the AHP period the total calcium acitivity was less sensitive to increases in the synaptic calcium activities. Through tuning of the inhibitory synaptic weights overactivation was prevented which otherwise caused the activity to be less dynamic. Figure 4.5 shows the difference in neuron response to the same input for seed weights (a) and end weights (b) for the same neuron presented in figure 4.1. The inhibitory synaptic weight for both configurations was allowed to converge before testing. After the synaptic weights learning phase the neuron responds more dynamically and more localized in time. It can also be noted that the end weight configuration is more responsive to PA's during periods of low general activity.



(a)



(b)

Figure 4.5: Comparison of neuron response to the same PA input. Net synaptic Ca^{2+}_{loc} activity is the sum of both excitatory and inhibitory synaptic activity. The underlying excitatory synaptic acitivty is shown for all excitatory synapses beneath the responses.

4.2 Dataset Performance

The EER performance of the trials with the clean dataset and with different SNR ratios are shown in table 4.1. Figure 4.6 illustrates the gradual decrease in performance for the increasing SNR ratios given in table 4.1.

4.2.1 Clean Dataset

Looking at the results in table 4.1 it is clear that MFCC coefficients outperformed the CN model neuron outputs. For clean data, the temporally dependant features captured by the MFCC+$\Delta/\Delta\Delta$ coefficients greatly improved the performance when added to the MFCC coefficients. Using the mel filterbanks directly lead to better performance than using non-delta MFCCs but did not outperform MFCC+$\Delta/\Delta\Delta$ features for the clean dataset. Increasing the number of neurons improved the performance of the CN model. The CN model with randomized synaptic end weights performed better than with learned weights for the clean dataset which indicate that the learning of the synaptic weights was not optimal, this is further addressed in the conclusion.

Features\SNR (dB)	**Clean**	**20** (13dB)	**1** (0dB)	**0.1** (-10dB)	**0.01** (-20dB)
Mel Filterbanks (40 filt.)	2.5	13.5	25.5	33.4	35.1
MFCC+$\Delta/\Delta\Delta$ (60 Coeff.)	1.4	16.4	30.1	35.0	38.2
MFCC (20 Coeff.)	3.6	14.5	23.1	32.3	34.5
CN (10 Neurons)	16.9	35.0	39.7	43.7	46.2
CN (40 Neurons)	14.5	33.0	34.5	42.1	44.4
CN (10 Rnd. Neurons)	12.5	26.2	38.2	42.5	45.1

Table 4.1: Speaker Verification Equal Error Rate (EER%) for the clean dataset and with different level of SNR.

4.2.2 Noise Dataset

The introduction of noise degraded the performance for all features, particularly for MFCC+$\Delta/\Delta\Delta$ coefficients which reached higher EER rates than non-delta MFCCs for SNR ratios of 20 and above. At SNR 1 the performance difference between the CN models and the MFCC coefficients decreased significantly from the clean dataset. For the CN model with 40 neurons (CN40) and MFCC+$\Delta/\Delta\Delta$ coefficients the percentage point difference changed from 13.1% to 4.4%. Between the same SNR levels (clean to SNR 1) the percentage point increase in EER for CN40 was 20 % while the percentage point increase for the MFCC+$\Delta/\Delta\Delta$ coefficients was 28.7%. Even more significant was the percentage point increase difference between SNR 1 and 20 where CN40 EER only increased by 1.5 % while MFCC+$\Delta/\Delta\Delta$ increased by 13.7%. The CN model with 10 neurons also showed less influence by noise from SNR 1 to 20 with an EER increase of 4.7%.

This distinct characteristic of the performance degradation from noise for CN40 (that it remained relatively unchanged between 20 SNR and 1 SNR) is

likely due to the nonlinear behavior of the CN model. The noise has to surpass a certain level before it has an effect on the total calcium activity of the neuron due to the effects of inhibition and the need for synaptic activities to increase simultaneously. This principle and the results indicate that there are regions of increasing noise levels where the CN model performance remains mainly unchanged before surpassing a level at which it starts deteriating again. The MFCC coefficients intuitively do not show this behavior since the noise is incorporated into the coefficients in a continous manner by gradually scewing the shape of the frequency spectrum.

The MFCC coefficients and the mel filterbanks had a similar performance decrease with added noise which is intuitive since the MFCC coefficients are just a more compact representation of the mel-frequency spectrum.

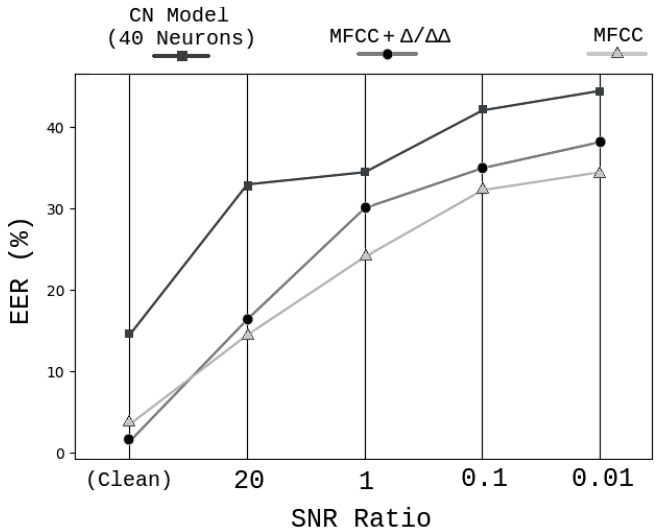


Figure 4.6: Illustration of the effect of different signal-to-noise ratios on EER performance for the CN model, MFCC+$\Delta/\Delta\Delta$ and MFCC features.

Chapter 5

Conclusion

The results showed a clear difference in the performance with clean and noisy data for all model types. However the the CN models, while not performing as well as the other approaches for clean data, shows promise for performing well in noisy conditions. The principles behind the model, based on being susceptible only to specific timings of frequency activities, together with the noise performance results indicate that the model can function relatively undisturbed by increased noise levels below certain thresholds.

The inclusion of the $\Delta/\Delta\Delta$ features for the MFCC coefficients caused the model to perform substantially better for the clean dataset. This indicates that there is a benefit to be gained from presenting the embedding network with features carrying information about their temporal context. However, in the presence of noise the performance of MFCC+$\Delta/\Delta\Delta$ features degraded quickly. This behavior for MFCC delta coefficients to perform well during clean conditions but poorly in noisy conditions is generally well known [12]. This is intuitively reasonable since differentiation tends to amplify noise and in particular for the $\Delta\Delta$ features which are a differentiation of a differentiation. In this domain the CN model has a potential advantage of carrying temporal information while also being less sensitive to noise than MFCC+$\Delta/\Delta\Delta$ features.

The reason for not finding any clear relationship between HEW synapses and PA correlation may be because a high degree of PA overlap may be an unsufficient guarantee for correlation with the total calcium activity for two PAs. For example, two PAs which are active during extended periods may be active together during longer periods of time and hence yield a higher correlation index. However, during this time the neuron may enter the AHP state which causes the corresponding synapses to decorrelate with total calcium activity. The correlation analysis might be best suited for data where the PAs are more sparse and hence cause the correlation of two PAs to be more significant.

The problem with the concentration of HEW synapses in certain bands of the PA frequency spectrum might be a consequence of the fact that the frequency bins of acoustic data generally are correlated with neighbouring frequencies. This could potentially be solved by making the model more susceptible to high temporal resolution patterns between frequency bins since broader patterns otherwise will overshadow finer ones.

5.1 Points of Further Study

The main focus of further studies will be on the learning of the synaptic weights which showed signs of being sub-optimal for the presented model based on the randomized weight results. One of the solutions could be to make the learning process more democratic with respect to potentiating synapses across the whole available spectrum instead of having the clustering behavior observed in the presented model. Such a democratization principle could be hardcoded like a limit to the amount of potentiated synapses in a frequency band or dividing the frequency spectrum up into smaller frequency segments from which a certain amount of synapses may be potentiated. Other less direct solutions might be to find other ways of preprocessing the input or increasing the time resolution further, thus potentially encouraging potentiation of synapses from a wide range of frequencies who share more fine grained temporal patterns.

Another area of interest would be to find other ways of decoding the CN model output instead of using a deep LSTM architecture since the CN model itself already captures some of the temporal context of the input which recurrent neural networks like LSTMs normally are used for.

Finally some things which were outside of the scope of this thesis were to run models with even greater numbers of neurons, test performance on different kinds of noise such as reverberation and real-life background audio, and to run the model on a completely text-independent dataset.

References

[1] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. *Front-end factor analysis for speaker verification.* IEEETransactions on Audio, Speech, and Language Processing. 2010. 19(4):788–798.

[2] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, Javier Gonzalez-Dominguez. *Deep Neural Networks for Small Footprint Text-Dependant Speaker Verification.* 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2014.

[3] Erik Marchi, Stephen Shum, Kyuyeon Hwang, Sachin Kajarekar, Siddharth Sigtia, Hywel Richards, Rob Haynes, Yoon Kim, John Bridle. *Generalised Discriminative Transform via Curriculum Learning for Speaker Recognition.* 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

[4] El-Moneim, Samia & Nassar, M & Dessouky, M.I. & Ismail, Nabil & El-Fishawy, Adel & Abd El-Samie, Fathi. *Text-independent speaker recognition using LSTM-RNN and speech enhancement.* Multimedia Tools and Applications. 2020. 10.1007/s11042-019-08293-7.

[5] David Snyder, Daniel Garcia-Romero, Daniel Povey, Sanjeev Khudanpur. *Deep Neural Network Embeddings for Text-Independent Speaker Verification.* 10.21437/INTERSPEECH.2017-620. 2017.

[6] Mirco Ravanelli, Yoshua Bengio. *Speaker Recognition from Raw Waveform with SincNet.* 2018 IEEE Spoken Language Technology Workshop (SLT), 2018.

[7] G. Bhattacharya, M. J. Alam, P. Kenny. *Deep Speaker Embeddings for Short-Duration Speaker Verification.* Interspeech 2017, 2017.

[8] Paliwal, Kuldip & Lyons, James & Wojcicki, Kamil. *Preference for 20-40 ms window duration in speech analysis.* 4th International Conference on Signal Processing and Communication Systems, ICSPCS'2010 Proceedings. 1 - 4. 2011. 10.1109/ICSPCS.2010.5709770.

[9] Henrik Jörntell, Fredrik Bengtsson, Pontus Geborek, Anton Spanne, Alexander V. Terekhovd, Vincent Haywar. *Segregation of Tactile Input Features in Neurons of the Cuneate Nucleus.* Neuron. 2014. DOI:10.1016/j.neuron.2014.07.038.

[10] Bengtsson, F., Brasselet, R., Johansson, R. S., Arleo, A., & Jörntell, H. Integration of sensory quanta in cuneate nucleus neurons in vivo. 2013. PloS one, 8(2), e56630.

[11] Rongala Udaya B., Spanne Anton, Mazzoni Alberto, Bengtsson Fredrik, Oddo Calogero M., Jörntell Henrik. *Intracellular Dynamics in Cuneate Nucleus Neurons Support Self-Stabilizing Learning of Generalizable Tactile Representations.* Frontiers in Cellular Neuroscience. 2018. 10.3389/fncel.2018.00210. 1662-5102

[12] Kumar, Kshitiz & Kim, Chanwoo & Stern, Richard. *Delta-spectral cepstral coefficients for robust speech recognition.* ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. 2011. 4784-4787. 10.1109/ICASSP.2011.5947425.

Printed by Tryckeriet i E-huset, Lund 2021

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2021-836
http://www.eit.lth.se