

Machine Learning
Qualifying Exam Report
Georgia Institute of Technology

Ph.D. Student

Nicolas Shu

Committee Members:

David Anderson

Gari Clifford (Advisor)

Evangelos Theodorou

Contents

| | | |
|----------|---|-----------|
| 0.1 | Key Papers | 3 |
| 1 | Speech Recognition in a Medical Environment | 5 |
| 1.1 | Background | 5 |
| 1.2 | Preliminary Data | 5 |
| 1.2.1 | Comparison of Different Commercial Engines | 5 |
| 1.2.2 | Grade Reading Level of Various Texts | 6 |
| 1.3 | Automatic Speech Recognition Schematic for a Medical Setting | 6 |
| 1.3.1 | AI-Complete and ASR-Complete | 6 |
| 1.4 | Speech Enhancement | 6 |
| 1.4.1 | Recurrent Neural Networks for Noise Suppression | 6 |
| 1.4.2 | Non-Negative Matrix Factorization | 8 |
| 1.5 | Blind Source Separation | 8 |
| 1.5.1 | Independent Component Analysis | 8 |
| 1.5.2 | Non-Negative Matrix Factorization (NMF) and Sparse NMF | 8 |
| 1.6 | Speaker Diarisation | 9 |
| 1.7 | Speaker Verification | 9 |
| 1.8 | Speech Recognition | 9 |
| 1.8.1 | Hidden-Markov Models and Gaussian Mixture Models | 9 |
| 1.8.2 | Long Short Term Memory Networks | 9 |
| 1.9 | Features from Speech | 9 |
| 2 | Deep Learning Theory | 11 |
| 2.1 | Deep Relaxation: PDEs for Optimizing Deep Neural Networks | 11 |
| 2.2 | Characterization of Neural Networks as an Encoder-Decoder with Mutual Information | 11 |
| 2.3 | The Stochastic Thermodynamics of Learning | 11 |
| 2.4 | Residual Networks as a Mean-Field Optimal Control Problem | 11 |
| | Appendices | 13 |
| A | Methodologies | 15 |
| A.1 | Long Short-Term Memory and Gated Recurrent Unit | 15 |
| A.2 | Mel-Frequency Cepstrum Coefficients | 15 |
| A.3 | Linear Prediction Coefficients and its Derivatives | 15 |
| | Bibliography | 17 |

0.1 Key Papers

- [E et al., 2018]
- [Shwartz-Ziv and Tishby, 2017]
- [Goldt and Seifert, 2017]
- [Chaudhari et al., 2018]

- [Lee and Seung, 2000]

Chapter 1

Speech Recognition in a Medical Environment

1.1 Background

Automatic speech recognition (ASR) has been a topic that interested many from an early age. Many consider it to have started in the 1950s with Bell Labs' *Audrey* [Moskvitch, 2017], which was able to do a single-speaker digit recognition. Seventy years later, the ASR technologies have grown so that they are present within our personal homes, with Amazon Alexa and Google Home devices [Fowler, 2018], as they have become more affordable. As the speech technologies advance, they approach the concept of ubiquitous computing (often also known as ambient intelligence), which is highly desirable for many industries. Although most of the ASR systems available for commerce are trained in normal language and not medical language, one industry that could greatly benefit from ASR technologies is the health care system, given that the ASR system is able to understand medical language.

Health care is a system which has a very high demand and the services are required to be as detailed as possible due to various reasons, one of which is the concept that it is paramount for a hospital to have a clear and rich track of a patient's medical history. Currently, in order to maintain a patient's medical history, a physician sees multiple people during his/her working hours, and only after their shift is over, does (s)he sit down to write the medical notes. The lag in between seeing a patient and taking notes may sometimes go up to 8-10 hours, and then those notes are often inaccurate. Having an ASR system in a medical environment could greatly help in keeping track of a patient's medical history, where a physician could easily dictate the notes. In a hospital environment, however, such as in an intensive care unit (ICU), a physician who is trying to dictate notes may find him or herself in trouble, as in the ICU, there are multiple background sounds from machines, multiple people speaking, and a great amount of white noise.

Current companies have been creating ASR systems that can understand medical language and allows physicians to dictate their medical notes. One major company that has been "dominating" a lot of the market is Nuance, with their Dragon Medical system. Unfortunately, their dictation system is not yet capable of inferring punctuation marks and markup language onto the text, thus, in order to dictate a segment such as: "37-year-old female presents complainint of urinary frequency, urgency and dysuria along with hematuria and low-grade fever." needs to be dictated as:

"37-year-old female presents complainint of urinary frequency **comma** urgency and dysuria along with hematuria and low-grade fever **period**"

Although the system has very high accuracy results, the system works best when one is in a quiet environment, which is not always a realistic scenario. The goal of this project is to create a transcription engine that can recognize medical language in a noisy environment.

1.2 Preliminary Data

1.2.1 Comparison of Different Commercial Engines

Compare commercial (+ CMU) Text- \rightarrow Voice- \rightarrow Text MIMIC II + 20kLeagues

1.2.2 Grade Reading Level of Various Texts

MIMIC II + 20kLeagues

1.3 Automatic Speech Recognition Schematic for a Medical Setting

In order to have a speech recognition system to be able to understand medical language, it is important to obtain a big picture of the project.

[INSERT FIGURE]

As one can see from the diagram above, the sound signals would come to a microphone or an array of microphones, which would then be passed through a system that performs blind source separation on the sound signal. It is very possible that, in a hospital environment, there may be multiple sources, such as the heart rate monitor, the ventilator, the healthcare providers talking, and others. A blind source separation algorithm would ensure to separate the audio channel to multiple sources. Once the sources have been separated, there would be a classification methodology to identify the physician from the patient, and ...

1.3.1 AI-Complete and ASR-Complete

Although there have been many advancements in the speech technologies, especially through black boxes such as neural networks, and there currently exists very accurate engines to do voice recognition, it still remains to be an unsolved problem. The speech recognition problem is ASR-complete, which falls under the umbrella of AI-complete, which, by analogy of NP-completeness, means that the problem is hypothesized to deal with multiple parts of the AI world, and it cannot be solved by a single algorithm. Current technologies are not able to solve AI-complete problems, and they often require human computation. Since speech recognition is an AI-complete problem, there is still a lot of room for improvement in the fundamental problems that have not yet been solved, and although deep learning has recently shown promising results, deep learning neural networks (DNN) do not necessarily solve fundamental problems, as no one has a truthful understanding of how they work and behave. Therefore, there are a lot of possibilities for the formulation design of the framework (i.e. pipeline).

1.4 Speech Enhancement

Speech enhancement is one of fundamental problems of speech that has not yet been fully solved yet, as noise is subjective, and there isn't necessarily one specific way of performing a fully accurate noise reduction for any type of noise. As previously mentioned before, a ventilator and heart rate monitor are examples of background noise that do not necessarily have a Gaussian distribution (i.e. it is not a white noise), yet, for ASR, they are both irrelevant. There are a number of methodologies that have been created to attempt to solve the problem of speech enhancement, and they may potentially be useful to be implemented onto the pipeline.

1.4.1 Recurrent Neural Networks for Noise Suppression

The work on recurrent neural networks (RNN) was originally worked sequentially by [Jordan, 1986] and [Barak, 1988], and it finally came to be coined by [Cleeremans et al., 1989]. They have shown to work very efficiently with time-series data as their cyclic nature allows them to adapt to incoming inputs in a sequential form, working often times better than convolutional neural networks (CNN), when dealing with time-series data. Other variants of RNNs have been created, which allow them to store "memory" for future cases, with Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997], which were then further expanded with gates for Gated Recurrent Units (GRU) [Gers et al., 2000] specifically to allow such units to forget information, and a visual representation of LSTMs and GRUs are shown on Figure 1.1

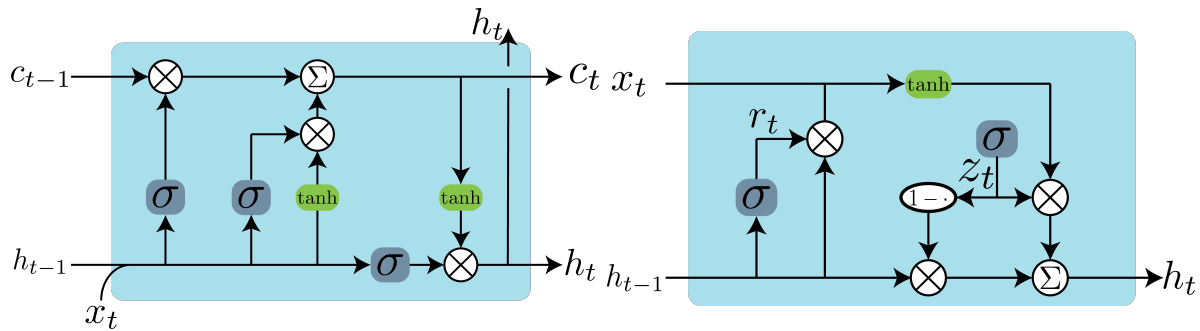


Figure 1.1: A visual representation of the LSTM cell (Left) and the GRU cell (Right). The LSTM contains three representations of its states: the cell state/memory c_t , the hidden states h_t , and the input state x_t , all of which occurs at time t . On the GRU, there are the input state x_t , the hidden state h_t , the update gate z_t and the reset gate r_t . The σ block is used to represent a sigmoid function, and the tanh block is used to represent a hyperbolic tangent for both images. Please see A.1 for the mathematical expressions of the cells above

Because of the nature of the network being able to receive continuous sequential inputs, it becomes a very attractive model to potentially behave as an adaptive filter. Hence, members from the Xiph.Org Foundation, a non-profit organization, along with Mozilla have created a RNN architecture that is based on GRUs called RRNoise [Jean-Marc et al., 2017], which, after trained, is able to reduce a lot of the background noise of audio files, thus enhancing the signal. Below on Figure 1.2 is the topology of the RRNoise architecture, where it outputs voice activity detection (VAD) as well as the gains from the input features.

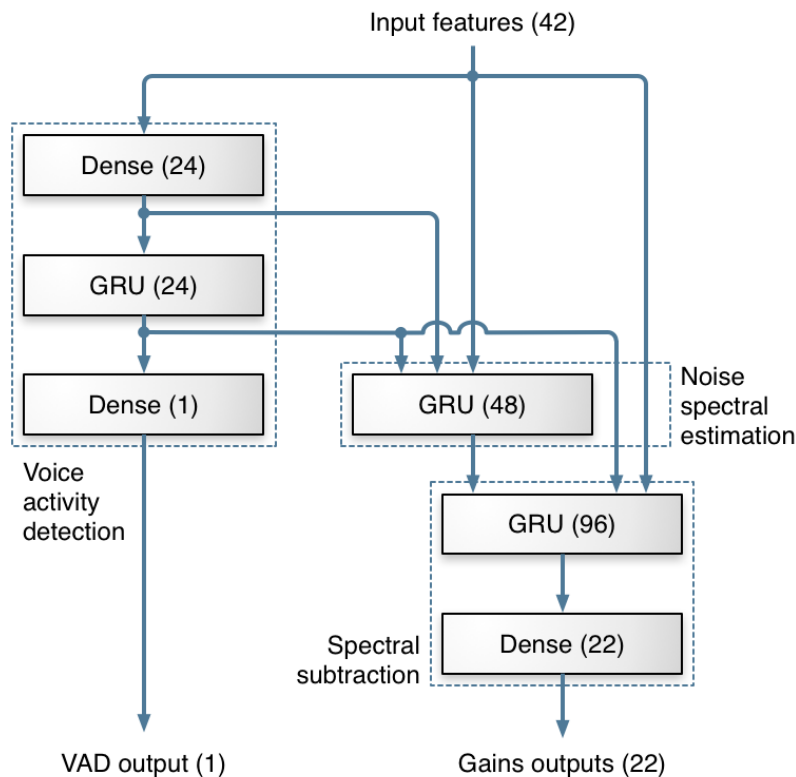


Figure 1.2: The topology of the RRNoise network used to perform noise reduction

1.4.2 Non-Negative Matrix Factorization

Another very neat way of performing speech enhancement is via non-negative matrix factorization. Please skip to the Blind Source Separation's subsection on [Non-Negative Matrix Factorization](#) [Ng et al., 2001]

1.5 Blind Source Separation

Given that a recording seldom contains one and only one speaker speaking at a time, an ASR may find itself troubled in detecting who is the speaker and who should be taken as the background. This is namely known as the Cocktail Party Effect. Its name originates from the idea that, in a cocktail party, there are multiple people speaking to one another. While a human is able to have a conversation with another, focusing on his/her friend while having all of the other speakers (i.e. sources) as background, an ASR system is not able to do so. Hence it is important to be able to separate a mixed signal to then be able to focus on a single individual and this is called blind source separation (BSS). There are two methodologies that are proposed to perform BSS: independent component analysis, and non-negative matrix factorization.

1.5.1 Independent Component Analysis

1.5.2 Non-Negative Matrix Factorization (NMF) and Sparse NMF

One of the major drawbacks from any methodology in ICA is that it requires N or more observations (i.e. microphone recordings) for N sources (e.g. speakers, noise), which, depending on the circumstance, may be limiting. The method of non-negative matrix factorization (NMF), however, is able to perform BSS with very high accuracies, while using a single-channel recording. NMF originated in the field of chemometrics, where chemists were trying to identify contents of solutions via positive matrix factorization [Paatero and Tapper, 1994] [Anttila et al., 1995]. It was later further developed in [Lee and Seung, 1999] and algorithms were then better formulated in [Lee and Seung, 2000].

The concept behind NMF is that a mixed signal should be able to be decomposed into its unmixed signals, under the assumption that all of the superpositioned signals are non-negatively added to one another. In other words, they are positively added, but their value may also be zero. Conceptually, this makes absolute sense. If one see the work in [Lee and Seung, 1999], one will see that many of the components added are interpretations of existing physical features, such as eyebrows and mustache. It does not make sense to add a negative eyebrow to reconstruct one's face. Hence, in retrospect, it makes complete sense why this methodology was being used in chemometrics, as it does not make sense to add a negative sulfate ion to a solution.

One may ask how is this factorization different from other types of factorizations, such as principal component analysis (PCA). First, every factorization has different purposes; for instance, a Cholesky decomposition was specifically made to obtain lower computation complexity in order to invert matrices, where as PCA was made to obtain information about its principal orthogonal components. Second, when looking at the comparison between a signal reconstruction or separation from an NMF and from PCA, PCA poses a lot of problems. PCA only allows for the feature space to represent vectors that are orthogonal, which is not always the case. In the sense of speech, it is almost impossible for one speech signal to be orthogonal to another one. PCA also assumes that the probability distribution of the data follows a multivariate Gaussian, which again is not always the case. A NMF is very promising as it performs a matrix decomposition, which is very desirable for computational reconstruction in the last step of the BSS, and it is able to attain reconstructions just as detailed as some of the other reconstruction techniques, plus one is allowed to determine how many components to use.

Here, the notation follows the notations shown in [Schmidt and Olsson, 2006].

Consider the magnitude of a spectrogram of a mixed speech signal $\mathbf{Y} \in \mathbb{R}^{M \times N}$, which is a summation of R source signals (i.e. $\mathbf{Y} = \sum_{i=1}^R \mathbf{Y}_i$). The spectrogram can be sparsely represented in an overcomplete basis as

$$\mathbf{Y} = \mathbf{D}\mathbf{H}$$

where $\mathbf{D} \in \mathbb{R}^{M \times R}$ represents a compendium of dictionaries with R columns, which can be chosen by the user, and $\mathbf{H} \in \mathbb{R}^{R \times N}$ represents a code matrix, which is sparse and it contains code matrices associated with the

dictionaries in D . The matrix D and H can be represented as

$$D = \begin{bmatrix} | & | & & | \\ D_1 & D_2 & \cdots & D_R \\ | & | & & | \end{bmatrix}$$

$$H = \begin{bmatrix} - & - & - & H_1 & - & - & - \\ - & - & - & H_1 & - & - & - \\ & & & \vdots & & & \\ - & - & - & H_R & - & - & - \end{bmatrix}$$

A simple version of how a NMF would work is shown in Figure 1.3

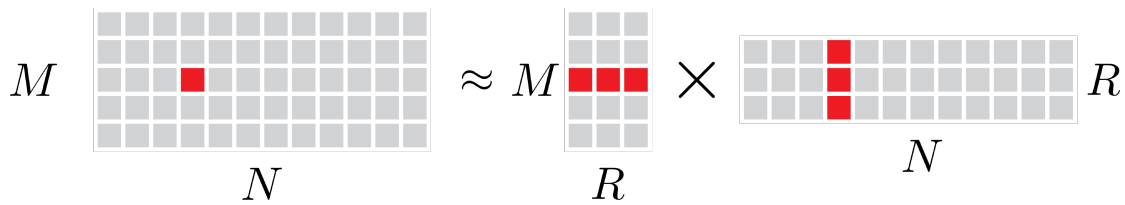


Figure 1.3: A very simple schematic of a non-negative matrix factorization

1.6 Speaker Diarisation

1.7 Speaker Verification

1.8 Speech Recognition

1.8.1 Hidden-Markov Models and Gaussian Mixture Models

1.8.2 Long Short Term Memory Networks

1.9 Features from Speech

Chapter 2

Deep Learning Theory

[Shwartz-Ziv and Tishby, 2017] [Chaudhari et al., 2018] [E et al., 2018]

- 2.1 Deep Relaxation: PDEs for Optimizing Deep Neural Networks
- 2.2 Characterization of Neural Networks as an Encoder-Decoder with Mutual Information
- 2.3 The Stochastic Thermodynamics of Learning
- 2.4 Residual Networks as a Mean-Field Optimal Control Problem

Appendices

Appendix A

Methodologies

A.1 Long Short-Term Memory and Gated Recurrent Unit

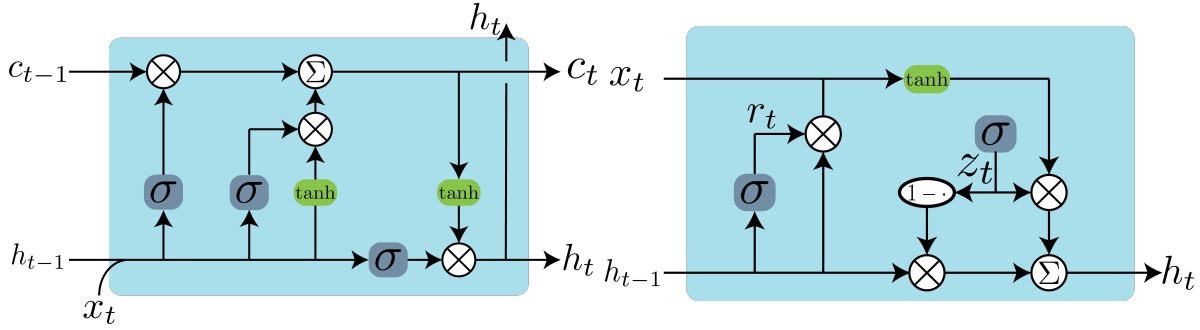


Figure A.1: A visual representation of the LSTM cell and the GRU cell

The LSTM and the GRU cell units are shown in Figure A.1. The equations that govern the LSTM are

$$\begin{cases} \begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \\ c_t f \cdot c_{t-1} + i \cdot g \\ h_t = o \cdot \tanh(c_t) \end{cases}$$

The equations that govern the GRU are

$$\begin{cases} \text{Update Gate:} & z_t = \sigma(W_z[h_{t-1}, x_t]) \\ \text{Reset Gate:} & r_t = \sigma(W_r[h_{t-1}, x_t]) \\ \text{Current Memory Content:} & \tilde{h}_t = \tanh(W \cdot [r_t \cdot h_{t-1}, x_t]) \\ \text{Final Memory:} & h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \end{cases}$$

A.2 Mel-Frequency Cepstrum Coefficients

A.3 Linear Prediction Coefficients and its Derivatives

Bibliography

- [Anttila et al., 1995] Anttila, P., Paatero, P., Tapper, U., and Järvinen, O. (1995). Source identification of bulk wet deposition in finland by positive matrix factorization. *Atmospheric Environment*, 29(14):1705 – 1718.
- [Barak, 1988] Barak, A. P. (1988). Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1:263–269.
- [Chaudhari et al., 2018] Chaudhari, P., Oberman, A., Osher, S., Soatto, S., and Carlier, G. (2018). Deep relaxation: Partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5(30).
- [Cleeremans et al., 1989] Cleeremans, A., Servan-Schreiber, D., and McClelland, J. L. (1989). Finite-state automata and simple recurrent networks. *Neural Computation*, 1(3):372–381.
- [E et al., 2018] E, W., Han, J., and Li, Q. (2018). A mean-field optimal control formulation of deep learning.
- [Fowler, 2018] Fowler, G. (2018). I live with alexa, google assistant and siri. here’s which one you should pick. *The Washington Post: The Switch Review*.
- [Gers et al., 2000] Gers, F. A., Schmidhuber, J., and Cummins, F. A. (2000). Learning to forget: Continual prediction with lstm. *Neural Computation*, 12:2451–2471.
- [Goldt and Seifert, 2017] Goldt, S. and Seifert, U. (2017). Stochastic thermodynamics of learning. *Physical Review Letters*, 118.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- [Jean-Marc et al., 2017] Jean-Marc, Xiph.Org, and Mozilla (2017). Rnoise.
- [Jordan, 1986] Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Cognitive Science Conference*, pages 531–546.
- [Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.
- [Lee and Seung, 2000] Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems, NIPS’00*, pages 535–541, Cambridge, MA, USA. MIT Press.
- [Moskvitch, 2017] Moskvitch, K. (2017). The machines that learned to listen. *BBC Future*.
- [Ng et al., 2001] Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press.
- [Paatero and Tapper, 1994] Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.
- [Schmidt and Olsson, 2006] Schmidt, M. N. and Olsson, R. K. (2006). Single-channel speech separation using sparse non-negative matrix factorization. In *INTERSPEECH*.
- [Shwartz-Ziv and Tishby, 2017] Shwartz-Ziv, R. and Tishby, N. (2017). Opening the black box of deep neural networks via information.