VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Richer Restricted Boltzmann Machines

Max Godfrey

Supervisor: Marcus Frean

Submitted in partial fulfilment of the requirements for
Bachelor of Engineering with Honours.

Abstract

This project focuses on verifying a new extension to an existing machine learning artefact, Restricted Boltzmann Machines. This new extension attempts to allow separate models to be trained to represent the different sources in an image. This amounts to blind source separation. The project will verify whether this new approach does work in practice and therefore that the theory is sound. The existing performance tests suggest that it performs very well on small, anecdotal images (2 bits). It appears to perform less outstandingly on larger images, like that found in the MNIST handwritten digit dataset. More tests will be needed to confirm this and find out why discrepancies in the theory and practice are occurring.

Contents

Chapter 1

Introduction

1.1 Problem

This project aims to verify the hypothesis that a new approach to representation learning should be able to generate a better general representation, despite only having access to noisy data. The project will explore this in a new extension to an existing state of the art in machine learning - the Restricted Boltzmann Machine. The new approach has been mathematically verified, but currently lacks reproducible, implemented tests to verify if the application matches the promise of the theory. Because this project sits as an extension to the Restricted Boltzmann Machine, this will be introduced in section 1.2.

1.2 Brief Context

In recent years a new machine learning approach has risen in popularity - the use of stochastic, generative models. These are structures that can be trained to represent input data in a general way. Being generative, these structures try to recreate the input data based on the internal representation they have learnt, creating their own reconstructed version of the input data. These structures, or 'machines' allow highly dimensional data, such as that of an image to be mapped to a much smaller and generalised representation.

One such system is a Restricted Boltzmann Machine, shortened to RBM. RBMs can form models of training data, by training a matrix of 'weights' that allow it to map a visible pattern, like an image, to an internal representation. RBMs perform well in practice, Hinton found that two RBMs layered on top of each other is enough to capture an excellent model for handwritten digits [5]. These layered RBMs or 'Deep Belief Networks' hold the current best performance in several machine learning tasks. Handwritten digit classification is one of such tasks, as well as general image classification [2].

This can reap benefits in machine learning tasks such as classification - being able to take unlabelled data and label it. Another application of this technology is speech recognition, like that found in smartphone voice assistants [8].

1.3 Solution

An appropriate example problem is the Completely Automated Public Turing Test to Tell Humans and Computers Apart. This is more commonly referred to as CAPTCHA and the more recent variant reCAPTCHA [12]. Both are used online for preventing automatic spam on websites and in the case of reCAPTCHA actually helping to build labelled machine learn-

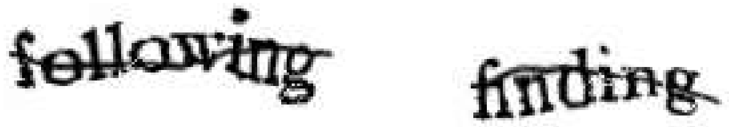

Figure 1.1: An example of a CAPTCHA, in this case a wobbly strikethrough obscures part of the text. [4]

ing datasets. This technology features a series of digits/characters composed with images of noise, as shown in Fig 1.1.

CAPTCHA is analogous to real world image data in a sense it is not perfect, instead being noisy or containing more than one entity. This project aims to verify the idea that if there are good models for two underlying causes, then they can each account for different parts of the image. This essentially means filtering so each model (RBM) can focus on the underlying, non-noisy entity. The second outcome is, given only a noisy dataset, the new approach can train two RBMs, one learning a model for the entity of interest and other modelling the noise. This separates the sources despite only seeing the noisy combination of them.

To achieve the aim of the project, the first goal is to gain enough understanding to extend the existing technology. This will be achieved by implementing the traditional RBM which can then be built on later with the new approach and provide a baseline to test from. The second goal is to implement a series of incremental tests, each increasing complexity of the data. These are to verify performance and therefore the tractability of this new approach and gain understanding of where the approach works. The tests also need to give insight that can help optimise this new approach as efficiency is definitely a risk that needs careful attention when it comes to working with these systems.

Finally, the tests should facilitate the project supervisor's ability to understand the problem and gain insight into the applications of the new approach.

1.4 Scope Changes

The approach to testing (in a performance sense) of the the theory has been refined, with the size of the datasets and images that were being tested with, being reduced to evaluate a more atomic case. This increases understandability and has made it easier to judge what a correct output/performance is. Given the theory is untested in practice this reduces the risk by always ensuring that value is being added for the appropriate effort.

Also, the approach to evaluation has been altered as ongoing evaluation allows the author to incrementally 'tick off' the sizes and entities of datasets where the algorithm works. As a result of these changes the updated Gantt Chart for this project is included in fig 4.1

Chapter 2

Background

This projects work is an extension of the existing state of the art, the Restricted Boltzmann Machine and therefore the origins and work related to Restricted Boltzmann Machines (RBMs) will occur in section 2.2. First, an example to ground this discussion will be given; the Cocktail Party Problem.

2.1 The Cocktail Party Problem

The Cocktail Party Problem as explored by McDermott J. in [9], nicely illustrates the concept of Blind Source Separation. Blind Source Separation is, as the name implies, is being able to take a noisy multi-cause input and separate it out into the underlying causes or 'sources'.

The Cocktail Party Problem centres on the way a partygoer can focus on a single conversation, despite there being multiple conversations going on in the background. The listener has to distinguish, given the mixed audio signal, the underlying conversation of interest.

The Cocktail Party Problem is applicable to the goals of the approach this project explores. The approach aims to leverage an understanding of the different models, or in the context of this problem, separating the sources.

2.2 The Restricted Boltzmann Machine

As touched on in the context section 1.2, extending Restricted Boltzmann Machine forms the basis of the project. In this section the RBM will be explained with enough detail to understand at a high level how the theory being tested works. It is not in scope of this project to provide the proof for the theory.

The RBM was originally conceived by Smolensky P under the name 'Harmonium' [13] in 1986. The example Smolensky presents as a 'cognitive task' that the RBM could solve is similar to that in which this project is focused; identifying an image of text with parts obscured, effectively separating the sources by ignoring the obscuring noise.

Hinton G. brought the Restricted Boltzmann Machine back into popularity recently, via a new tractable training technique [5]. This allows training in a greedy fashion where the individual units of the representation are independent. This means a hidden representation can be generated in a single pass. This is important because given a rich dataset like an audio signal in the Cocktail Party Problem, a single pass limits the complexity of training to be the number of times you run over the dataset.

The new approach sacrifices this single pass with the aim to construct a more representative internal model of multi-cause input data compared to the traditional approach. This requires an understanding of the RBMs structure which is akin too that of a fully connected

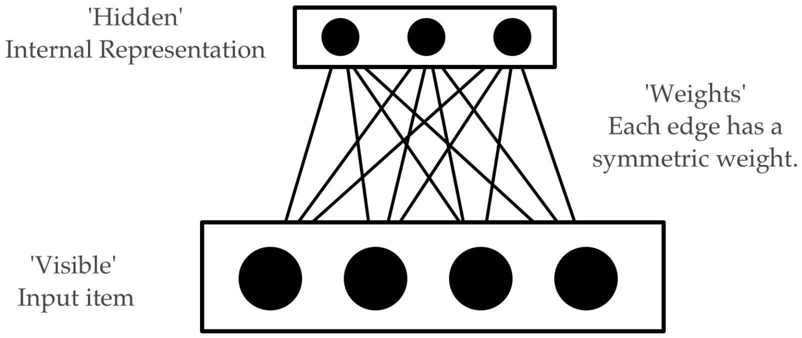

Figure 2.1: Visual representation of an RBM.

bipartite graph. In the RBM, the edges between the groups of nodes each have an associated 'weight'. These weights are tweaked during the training process to maximise the probability of generating the best representation of all the data items in the training set.

Fig 2.1 shows a graphical representation of an RBM, in this case with three hidden units and four visible units. Each edge between the visible and hidden layers has a symmetric weight. This weight is used to transform one layer's (visible or hidden) representation to the other. Both the visible and hidden units are binary, i.e. $\in 0, 1$.

2.2.1 Internal Representation

The new approach takes the form of having two Restricted Boltzmann Machines, one trained on the entity of interest, another on noise. This amounts to slightly changing the way the internal representation is generated for a given input vector. In the context of images, say a single digit CAPTCHA, this approach aims to allow the digit model to take responsibility for the parts of the images that look like a digit. Conversely, the noise model is able to take responsibility for parts of the image that look like noise.

The internal representation of the input for RBMs is described below. Given an RBM with j hidden units (for representing the input), i input units, and weights matrix W_{ji} that represent the symmetric weights between the input and the representation, we want to find the internal representation for all h_j units given an input vector $\tilde{v}$. First the weighted sum into a given hidden unit is calculated via

$$\psi_j = \sum_i (W_{ji} v_i)$$

A Bernoulli trial is performed to see if based on ψ_j, the hidden unit h_j gets set to 1 or 0. Let x be a random floating point such that $x \in [0 - 1]$.

$$P(h_j = 1|\tilde{v}) = \begin{cases} \sigma(\psi_j) > x & 1 \\ \sigma(\psi_j) \leq x & 0 \end{cases}$$

Here we see the logistic or sigmoid function σ. This acts as a smooth threshold for deciding whether to turn on the hidden unit.

2.2.2 New Approach

The new approach introduces a second RBM. so a change in notation is required. A subscript or superscript A or B can be added to all the values defined above to specify they come from $model_A$ or $model_B$ respectively. The update to the hidden units of $model_A$ is as defined above, with a slight change to the calculation of ψ_j:

$$\psi_j = \sum_i (W_{ji} v_i) + \sum_i (C^A_{ji})$$

Where C^A_{ji} is the correction that is applied, this is where the interaction between the two RBM's takes place allowing one to take responsibility for different parts of the input. The proof of this theorem is not in the scope of the project, however the definition for C^A_{ji} can be briefly discussed. It is defined as:

$$C^A_{ji} \;=\; \log \left[\frac{\sigma(\phi_i^{j=0})}{\sigma(\phi_i^{j=1})} . \frac{\sigma(\phi_i^{j=1} + \epsilon_i)}{\sigma(\phi_i^{j=0} + \epsilon_i)} \right]$$

Where ϕ_i is the weighted sum into the *ith* visible node. $\phi_i^{j=0}$ is the weighted sum into the *ith* visible node without any contribution from the weight $W_j i$. Conversely, $\phi_i^{j=1}$ is the weighted sum into the visible node i where the *jth* hidden node is turned on; its weight contributing to the weighted sum. Finally, ϵ is the contribution from the 'other' RBM. It is the weighted sum into the *ith* visible unit from the other RBM.

This correction adds an expensive calculation to generating the hidden representation. Unfortunately, it introduces dependancy between the hidden units, meaning we have to repeat this calculation x number of times so the probabilities of turning off/on the hidden units can reach equilibrium.

Chapter 3

Work Done

A large amount of the accomplishments in the project so far have been in gaining the understanding needed to really explore and verify this new technique - given it is built upon the existing work of Restricted Boltzmann Machines. This understanding means that robust correctness tests can be written. This in turn ensures that results produced have more credibility, so the implementation is known to be sound.

3.1 Design

3.1.1 Language Choice

The project is currently implemented using Python 3.4, however three other languages were considered for this project, Matlab, R and Java. Several factors were considered that lead to this decision.

Efficiency A large amount of data is required to train a Restricted Boltzmann Machine and the new approach has a higher complexity than the traditional RBM implementation. The language choice should not prevent the algorithm from being run in reasonable time, as this would hamper the ability to test it for correctness and performance.

Ease of use The project requires significant up front learning which means some familiarity with the language or a shallow learning curve will help keep velocity/progress high. The focus can then be on understanding the required theory instead of language niches.

Support for testing From a correctness standpoint, being able to evaluate the algorithms will require confidence of a correct implementation and therefore help lend credibility to subsequent findings.

Machine Learning Library support Being able to leverage libraries allows the focus to be on implementing and testing the new algorithm.

Stakeholder collaboration The projects supervisor who conceived the new approach inherently has a good understanding of how it works and therefore having a language they are comfortable working in can help with ensuring implementation correctness. For instance pair programming has been employed which is facilitated by a programming language common to both developers.

All of the languages considered have the efficiency for machine learning tasks, despite Python being on the slower end of the spectrum of languages, versus that of the compiled

JVM languages like Matlab or Java, it is still fast enough. The complexity of the traditional RBM is relative to what is trying to be modelled, less of the language implementing it. Java and Python have robust testing suites given they are used for non-academic goals. Matlab and R do have suites however the author has not had experience with them. Choosing from one of these two languages would increase the amount of upfront learning required and therefore increase this risk. The author had the most experience with Python and Java, however Python was favoured due to it's Matlab-like library NumPy. Expressive matrix syntax combined with the familiarity and brevity of Python made it a compelling option.

3.1.2 Library Choice

Considerations regarding library choice only really apply to the part of the system that needs to be robust, the number crunching.

Given the choice of Python, the decision to use the NumPy library was almost inherent. NumPy is a linear algebra library that offers fast implementations of matrix operations in a concise, expressive syntax. For instance, a whole training set of images, each image of size x^2, represented by a 3 by x by x matrix, can then be operated on all at once. Also NumPy is interoperable with open source Python Machine Learning Libraries. For instance Scikit-Learn [11] and Theano [3] which allows greater reproducibility in my results. Finally, it is a mature, well tested library, being an evolution of the Numeric python library which started developing in 1995 [1].

3.1.3 Dataset Choice

A crucial part of the project is ensuring that the tests can be reproducible, allowing any results to be used as evidence in a paper. Hence the importance of choose a dataset that has some credibility or can allow this approach to be compared to existing work. The MNIST digit dataset was originally chosen as a good dataset for this projects tests. There has been work using RBMs [5] and other machine learning approaches [6], to represent these handwritten digits previously. This gives a point of reference to compare performance.

It is worth noting: In section 3.2.1 it is highlighted that the results from testing with this dataset are not outstanding, hence a more minimal toy-dataset is being used until more understanding can be gained about what is and is not working.

3.2 Implementation

The traditional Restricted Boltzmann Machine has been implemented.

The new theory applies a correction when constructing the hidden representation of the noisy input. This correction allows each model (RBM) to take responsibility for different parts of the input data.

The calculation of this correction introduces a dependency between the hidden units and therefore we sacrifice some efficiency for better performance on multi-cause data. The full, un-approximated correction calculation has been implemented and connected to the Restricted Boltzmann Machine.

Tests have been created to verify if the new approach is working. As RBMs are generative models, we can show them an input and have them generate their own recreation based on the internal representation. This recreation is dependant on the image it was created from and therefore could be used for a performance test. We can approximate the likely-hood of, given a noisy image, how well can we reconstruct the underlying clean image. This forms a 'score' for the *ith* pixel of an image.

$$P(v_i'|v_i^{composite}) = P(v_i'|v_i^{clean})$$

Where v_i' is the reconstruction based on the RBMs internal representation. $v_i^{composite}$ is the composite (noisy) image and v_i^{clean} is the clean image; the underlying model that noise is added to to create the composite image.

This score can then be approximated image-wise and compared to the traditional approach, by sampling (creating reconstructions). From this we can see which images the new approach performs better/worse.

3.2.1 Handwritten Digit Recognition

The performance tests described in section 3.2 have been implemented for all digits in the MNSIT Handwritten Digit dataset [7] and the results are not outstanding. A toy model of a horizontal bar acts as the noise and is composed with the digit dataset to create the noisy input. We see this in 3.1. The Noisy Input is the composite image, this is what the reconstructions pictured in New Approach and Traditional Approach were generated from. The target was generated from the underlying (non-composited) '2' images.

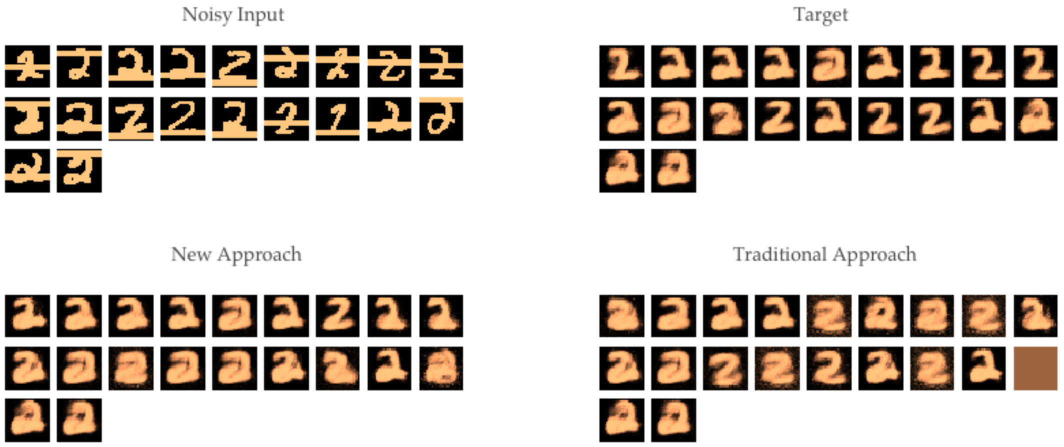

Figure 3.1: Initial results using new approach on noisy handwritten two images. The input was 20 MNIST two digits with a random 4 pixel high bar composited. The new technique does not perform better by much over the dataset.

Another performance test carried out has been classification performance. The hypothesis being that the new approach should be able to generate a better representation despite having noisy training and test sets. The actual classification is deferred to an AI classifier that is trained on the hidden representations. Fig 3.2 shows a visual representation of this. The RBM is trained on some data, then a hidden representation and known labels are used to create a transformed training and test set which a perceptron or other classifier is then trained on. The better the classification performance the better representation the model has achieved.

The non-outstanding results shown in 3.1 and the other digits, highlighted that some minimal test cases need to be satisfied before more complicated models like digits can be explored. The tests currently in place are not atomic enough.

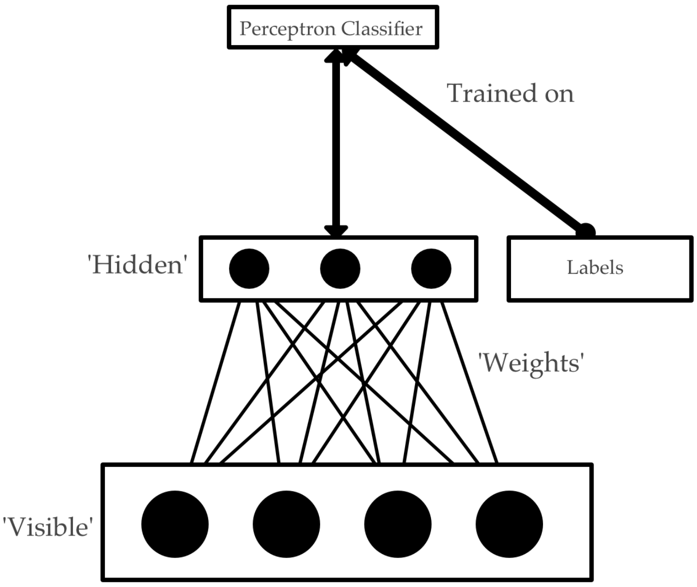

Figure 3.2: Figure demonstrating how classification can be applied using an RBM.

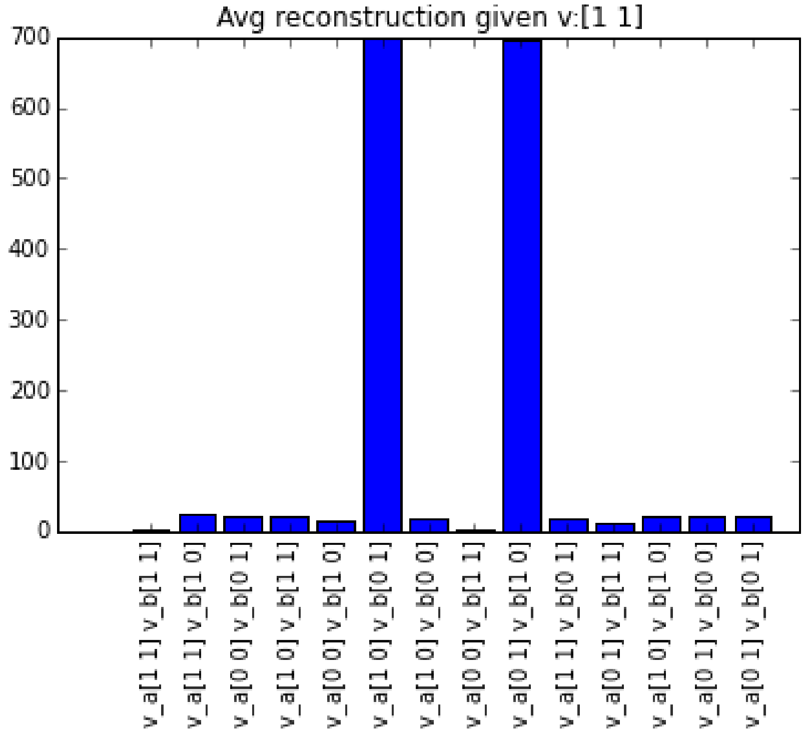

Figure 3.3: Figure showing the results of generating reconstructions using the new approach and two identical RBMs that can recognise a single pixel being on a time.

We arrived at the minimal test case, a two bit image, with binary pixels and a toy RBM model trained to understand one pixel being on at a time. This had the advantage of being able to be checked by hand and approximations of the correction could be explored in the minimal case. This is the smallest case possible for them to actually work and as a result, more robust correction tests have been constructed.

In the two bit scenario it is very easy to measure the performance of reconstructions, we would hope given an visible input of $v = [1, 1]$ that the new approach would most of the time result in reconstructions from the two models of $v'_A = [1, 0]$, $v'_B = [0, 1]$ and conversely $v'_A = [0, 1]$, $v'_B = [1, 0]$. Where v'_A is the reconstruction generated, given v from model A.

The results of doing this in a two bit system, that is two hidden and two visible units resulted in the plot in fig 3.3. We see the most common reconstructions are those defined above. It is important to look at the reconstructions from both models holistically to ensure they are actually separating the sources. The figure shows the average of 1000 runs of the new approach, the two bit example making this many runs quite feasible with so few features.

3.3 Evaluation Planning

As the output of the project is essentially a series of reproducible performance results, it is important that the algorithm is sound in practice and matches what the theory describes. The validity of the results hinge on the confidence that the code does what it should. This precondition is enforced by unit and integration testing the system. The third party libraries in use for linear algebra (NumPy) and plotting (Matplotlib) are well tested and feature in other peer reviewed literature in the field of Machine Learning [10].

As touched on in the scope changes in section 1.4 the evaluation phase of the project needs to be ongoing. The performance of the new approach needs to be checked at every point to ensure that cases where the new approach does and does not work can be pinpointed. It also makes tracking down what factors make it more or less effective easier. The actual approach to evaluation is touched on above, but it amounts to verifying if and when the new approach generates a better representation of a composite image versus that of the traditional approach.

Chapter 4

Future Work

Going forward it will be important to move from working with two bit image back up to larger images with non-toy models - i.e. the MNIST handwritten digit dataset. An outcome of the next step is being able to train two RBMs from purely composite data. For instance, being able to train an RBM that learnt a representation of a digit and another from noise from only seeing digits combined with noise. If this is fruitful, it will amount to blind source separation.

It is worth noting that out of scope at the moment is using actual photos, as this would require significantly more computing power. The runtime of this new approach needs to be tested for correctness on smaller images before this can occur. Also this might require aligning the content images and potentially acquiring a larger image dataset. It also raises questions around alpha matting or other techniques to preprocess the images. Working with datasets like MNIST where this has already been applied allows the focus to be on evaluating the new technique.

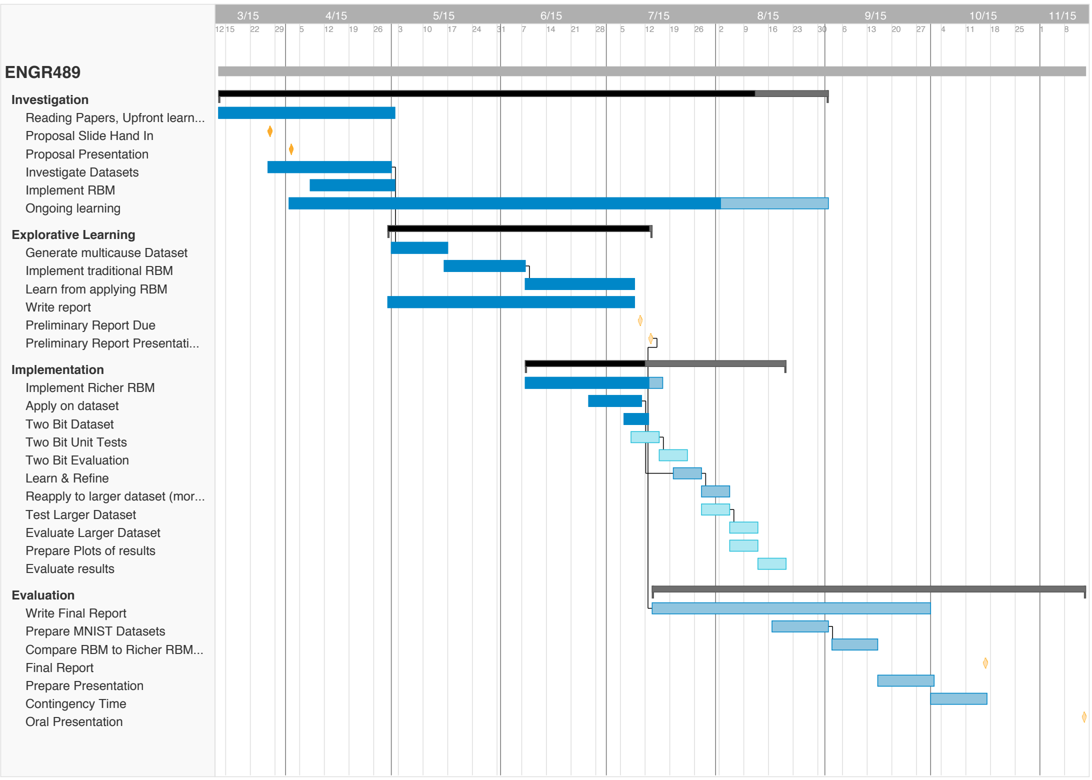

Figure 4.1: Revised Gantt chart

Bibliography

[1] History of SciPy. http://wiki.scipy.org/History_of_SciPy/. Accessed: 2015-07-20.

[2] BENGIO, Y., COURVILLE, A., AND VINCENT, P. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 8 (2013), 1798–1828.

[3] BERGSTRA, J., BREULEUX, O., BASTIEN, F., LAMBLIN, P., PASCANU, R., DESJARDINS, G., TURIAN, J., WARDE-FARLEY, D., AND BENGIO, Y. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)* (June 2010). Oral Presentation.

[4] COMMONS, W. Modern-captcha, 2007. File: Modern-captcha.jpg.

[5] HINTON, G., OSINDERO, S., AND TEH, Y. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation 18*, 7 (2006), 1527--1554.

[6] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278--2324.

[7] LECUN, Y., AND CORTES, C. The MNIST database of handwritten digits.

[8] LING, Z.-H., KANG, S.-Y., ZEN, H., SENIOR, A., SCHUSTER, M., QIAN, X.-J., MENG, H. M., AND DENG, L. Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine 32*, 3 (Apr. 2015), 41.

[9] MCDERMOTT, J. H. The cocktail party problem. *Current Biology 19*, 22 (Dec. 2009), R1024--R1027.

[10] MILLMAN, K. J., AND AIVAZIS, M. Python for scientists and engineers. *Computing in Science & Engineering* (2011).

[11] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825--2830.

[12] SHET, V. Google security blog, 2014.

[13] SMOLENSKY, P. *Foundations of harmony theory: Cognitive dynamical systems and the subsymbolic theory of information processing.* Parallel distributed processing: Explorations in the ..., 1986.