CS229 Final Project Report, Stanford University

Phoneme Recognition Using Deep Neural Networks

John Labiak

December 16, 2011

1 Introduction

Deep architectures, such as multilayer neural networks, can be used to learn highly-complex, highly-nonlinear functions by mapping inputs to outputs through multiple layers of nonlinear transformations. Problems in artificial intelligence (AI) are filled with very complex, poorly understood processes, and deep architectures have shown promise when applied to a variety of problems in AI, such as visual object recognition [1], and natural language processing [2]. However, very little has been done to explore the benefits of deep architectures for automatic speech recognition (ASR). In a typical speech recognition system, a hidden Markov model (HMM) is used to model the sequential structure of speech, and Gaussian mixture models (GMMs) are used as density models of acoustic feature vectors to estimate the state-dependent probability distributions of the HMM. Recently, researchers have begun exploring ways to leverage the modeling capacity of deep neural networks (DNNs) for automatic speech recognition. For example, it is possible to replace GMMs with DNNs for acoustic modeling within the HMM framework [3]. Deep neural networks have also been applied within a new paradigm for ASR, which replaces the traditionally used HMM with segmental conditional random fields (SCRFs). Within this framework, DNNs have been used to construct phoneme recognizers, which are then fed as an additional feature to the SCRF model.

Deep neural networks are typically constructed by stacking multilayer neural networks, such as denoising autoencoders. Previous work has suggested the importance of context on recognition performance [4]. One approach to generating a large context window within a stacked architecture is to concatenate the posterior outputs of a classifier, and then use these posterior features as inputs to a second classifier. In this work, we test the effect of larger context on phoneme recognition for a softmax classifier. In particular, we construct a phoneme recognizer by stacking softmax classifiers, using concatenated posterior outputs from a softmax classifier as posterior features for a second softmax classifier.

The contribution of this work is twofold. Firstly, we gain insight into the importance of using a large context for phoneme recognition. In particular, we test the idea that using a large context improves phoneme recognition by enabling a softmax classifier to learn temporal patterns and phonotactics from the training set. Secondly, the work done here sets the stage for future work in deep learning. In particular, the insight gained into the role of context can be used to build a better phone detector, which can then be used as an additional feature for an ASR system based on the SCRF model.

The remainder of this paper is outlined as follows: First, we describe the methods used in our work. Next, we present the results of our experiments. Finally, we conclude with a discussion of the results and future directions of our work.

2 Methods

2.1 Softmax Regression

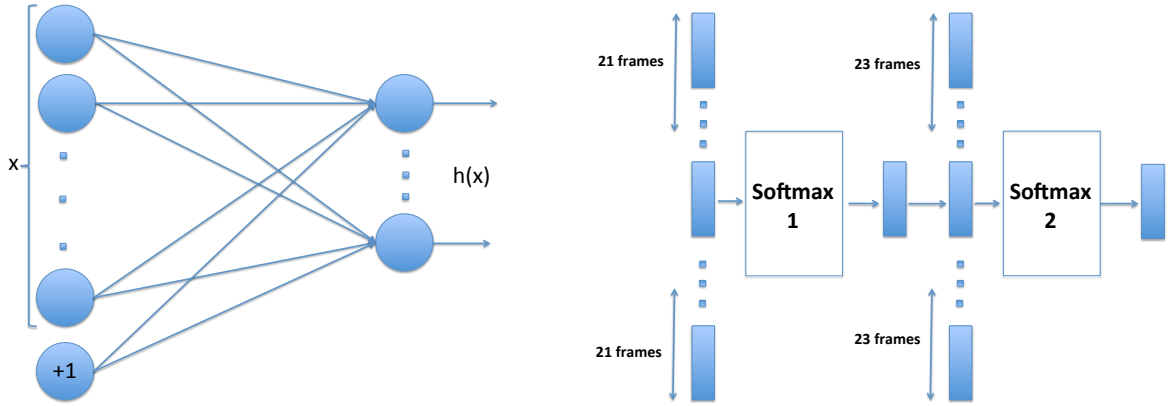
Softmax regression generalizes logistic regression for classification problems in which there are more than two classes. Suppose we have k classes (i.e. we have $y^{(i)} \in \{1, 2, ..., k\}$). Then, for a given feature vector $x^{(i)} \in \mathbb{R}^{n+1}$ (where $x_0 = 1$), the softmax classifier outputs a vector of posterior probabilites:

$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} p\left(y^{(i)}=1|x^{(i)};\theta\right) \\ p\left(y^{(i)}=2|x^{(i)};\theta\right) \\ \vdots \\ p\left(y^{(i)}=k|x^{(i)};\theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

where $\theta_1, \theta_2, ..., \theta_k \in \mathbb{R}^{n+1}$ are chosen to minimize:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2$$

over a set of training examples $\{(x^{(1)}, y^{(1)}), ..., (x^{(m)}, y^{(m)})\}$. The softmax classifer then assigns $x^{(i)}$ to the class with the largest posterior probability. That is, it sets $y^{(i)} = j$ where $j = \arg\max_j p\left(y^{(i)} = j|x^{(i)};\theta\right)$. Figure 1(a) shows a diagram of the softmax classifier, where we have $n+1$ input nodes and k output nodes.



(a) Softmax classifier

(b) Stacked softmax classifier with 23 frame posterior feature vectors

Figure 1: Overview of the architectures used in our experiments.

2.2 Stacked Softmax Classifier

For the stacked softmax classifier, the posterior probability vectors output by a softmax classifier (Softmax 1) are concatenated for a window of speech frames and used as input to a second softmax classifier (Softmax 2). Figure 2(a) shows an overview of the architecture for a stacked softmax classifier in which a 23 frame window is used to construct the posterior features. In particular, note that we use two sets of context frames within this stacked architecture: we use 21 frame MFCC feature vectors as input to the first softmax classifier and 23 frame posterior features as input to the second softmax classifier. The details of the construction of these two sets of features are included below.

2.3 Experimental Setup

Our experiments are based on phonetic classification of frames of speech from the Broadcast News database (approximately 430 hours of data). For the first softmax classifier, we construct acoustic feature vectors by concatenating 13-dimensional Mel-frequency cepstral coefficients (MFCCs) for a window of 21 speech frames. We preprocess the data using PCA to whiten the features, thus yielding 139-dimensional feature vectors for the first softmax classifier. For the second softmax classifier, we construct posterior feature vectors by concatenating the posterior outputs from the first softmax classifier for a window of speech frames. We perform experiments using 3, 13, and 23 frame context windows. We use force-aligned outputs of a speech recognizer as ground-truth for the phoneme labels. The set of phonetic labels contains 42 classes, including 41 non-silence classes and 1 silence class. For the silence class, we map silence, background noise, and voiced noise to a single silence token.

We divide the data into subsets for training and testing the performance of the classifiers. We train both softmax classifiers using mini-batch LBFGS. We use a batch size of 20 and 5 files (each file contains approximately 1 hour of speech), for training the first and second softmax classifiers, respectively. For both classifiers, we regularize the cost function using a weight decay parameter of $\lambda = 1e^{-4}$. Furthermore, for each classifier, we initialize the parameters for each batch using the average parameter values across all previous batches, and train on each batch for 20 iterations. After training is complete, we test the classifiers by recording frame level accuracies for the phonetic classification task on the held out test set.

3 Results

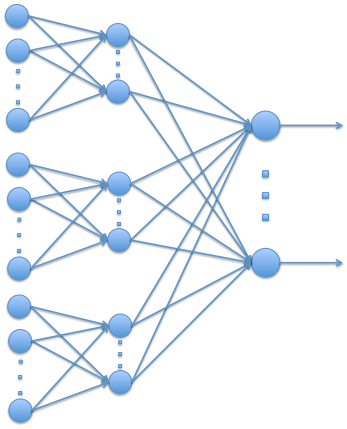
Table 1 displays the results of our experiments on the Broadcast News data. In particular, Table 1 displays the frame level accuracies for our phoneme recognition task. Referring to Table 1, we see that using a large context improves the performance of our softmax classifier, with the largest gains seen as we increase the context window from 3 to 13 frames, and leveling off thereafter.

Classifier	**Accuracy (%)**
Softmax	37.94
Stacked-softmax w/ 3 frame context	40.77
Stacked-softmax w/ 13 frame context	44.34
Stacked-softmax w/ 23 frame context	45.49

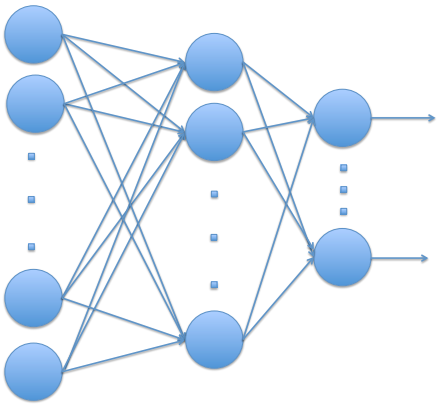
Table 1: Frame level accuracies for phoneme recognition on the Broadcast News data.

4 Discussion

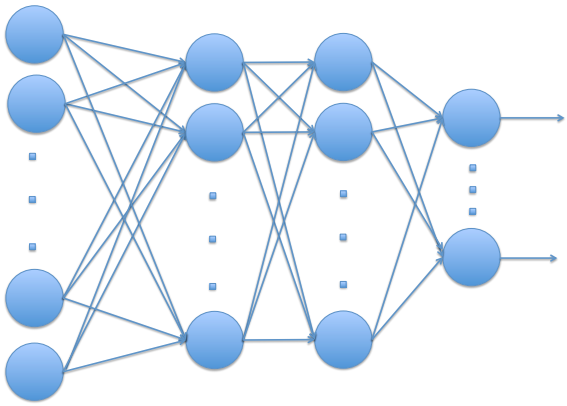
In this work, we tested the effect of context on phoneme recognition for a softmax classifier. In particular, we constructed a simple neural network by stacking softmax classifiers. Within this architecture, we used concatenated posterior outputs of a softmax classifier as inputs to a second softmax classifier. By stacking softmax classifiers in this manner, we gained insight into the importance of context for classification of phonetically labeled speech frames. In particular, we found that stacking softmax classifiers improves frame level accuracy over a single softmax classifier, and the accuracy improves with increasing context, in the range of 3 to 23 context frames for the posterior features.



(a) Stacked softmax classifier w/ 3 frame context



(b) Two-layer neural network



(c) Deep neural network

Figure 2: Neural network architectures.

The work considered herein is part of a larger project to construct a phoneme recognizer which can be fed as a detector stream to an ASR system based on the SCRF model. We have, therefore, provided insight into which neural network architectures work well for phoneme recognition. The stacked softmax classifier created a large context by using a window of concatenated posterior feature vectors. Figure 2(a) shows a stacked softmax classifier in which a 3 frame context window is used to construct posterior features. This architecture is a special case of a two-layer neural network (Figure 2(b)). Future work will consider alternative architectures for constructing a phone recognizer. For example, we might consider deeper architectures, such as that in Figure 2(c), which can be constructed by stacking many multilayer neural networks. Deeper neural networks include additional layers of nonlinearity, and experiments with deeper architectures will give us insight into the effects of adding these additional layers of nonlinearity. In particular, there is great interest in comparing the effects of context versus additional layers of nonlinearity for phoneme recognition. We might also consider alternative approaches for generating a large context, such as constructing acoustic features by concatenating a much larger window of MFCC features. Finally, future work may examine the effects of context and additional layers of nonlinearity together by considering alternative deep network architectures which generate large context by using concatenated posterior features.

References

[1] M.A. Ranzato, F.J. Huang, Y.L. Boureau, and Y. LeCun, Unsupervised learning of invariant feature hierarchies with applications to object recognition., in Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-07), 2007, pp. 18.

[2] R. Collobert and J. Weston, A unified architecture for natural language processing: deep neural networks with multitask learning, in Proceedings of the 25th International Conference on Machine Learning (ICML-08), 2008, pp. 160167.

[3] G. Zweig, P. Nguyen, D. Van Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, F. Sha, M. Wang, A. Jansen, H. Hermansky, D. Karakos, S. Thomas, G.S.V.S. Sivaram, K. Kintzley, S. Bowman, and J. Kao, Speech Recognition with Segmental Conditional Random Fields: Final Report from the 2010 JHU Summer Workshop, no. MSR-TR-2010-173, November 2010.

[4] S. Thomas, P. Nguyen, G. Zweig, H. Hermansky, MLP Based Phoneme Detectors for Automatic Speech Recognition. Microsoft Research, 2010.