Training a Convolutional Neural Network for Phonemes Classification

Mohamed Abdulaziz Ali Haseeb moaah@kth.se
Omar El-shenawy omares@kth.se

Abstract

Convolutional Neural Networks has interesting properties that make them more suitable to cope with spectral variations and model spectral correlations. In this report we present our work on training a deep convolutional neural network CNN for phoneme classification acoustic task. CNNs with different configurations are trained and compared.

1 Introduction

Neural networks have always been an attractive area of research since 1960. The attempt at simulating the human brain has always been intriguing. Since the first perceptron model, Neural Networks have evolved in many ways, in which layers of perceptrons grew wider and deeper. However, until recently, it was only possible to train shallow networks, because of the vanishing gradient problem. The vanishing gradient is a phenomena where the error information starts to decay when propagated through many layers, and therefore the learning process is no longer doable. A remedy was made by Hinton[7], in which the network is trained a layer at a time, instead of trying to train all layers at once.

Deep Learning is the new trend in Machine Learning field. Recently, there has been many applications that uses Deep Learning. Training these deep networks is very expensive computationally, they require heavy computations on the GPU, and so far, several frameworks that facilitate training Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) —a deep network that uses Convolutional filters— have been developed by many research labs around the world. CNNs are usually very popular with the computer vision applications.

Recently, DNNs and CNNs have been applied to the field of speech recognition with very promising results. CNNs have the ability to reduce spectral variations and model spectral correlations which exist in signals, therefore CNNs are a more effective model for speech compared to DNNs [12]. In this work, we experiment with CNNs and apply them to a small scope of acoustic modelling which is phoneme classification, using the TIMIT dataset. We use the CNN training library, Caffe to train our network. We train on spectrograms i.e. images of Fast Fourier Transform applied on the phonemes acoustic data. We base our work on the architecture described in [12].

2 Related Work

There has been many approaches for speech recognition with Neural Networks. The classical approach was always to combine Hidden Markov Models (HMMs) with NNs, such as [11]. This approach has been very successful and popular. Recently, there has been attempts to remove the need for HMMs. [6] used Recurrent Neural Networks (RNNs) with good results for speech recognition and has yielded promising result. [5] have done similar work

with RNNs.

[12] uses HMMs in their model, however, we only build a CNN based on their architecture, and since we do not do speech recognition, there is no need for an HMM.

There has been attempts to use both DNNs and CNNs in speech recognition, however, DNNs have difficulty modeling transitional variance within speech signals, which exists due to difference in speaking styles [9]. Various speaker adaptation techniques are required to reduce this variation. Therefore, we have preferred to use CNNs for this task, since TIMIT consists of a wide range of speakers.

3 Method

In the section the approach followed to build the CCN will be described.

3.1 Overall setup

A deep CNN will be trained to give a probability distribution over the phonemes labels given the acoustic input. The acoustic input will be converted into a sequence of fixed size frames windows, that is converted into spectrograms. The deep CNN will then generate probability distributions over the possible phone labels for each spectrogram. The sequence of the probability distributions will then be used to compute the emission probabilities of the HMM states on a Viterbi decoder that can generate the expected phones sequence.

In this work, the CCN network ability to predict the correct phone label given an input spectrogram was tested, and no Viterbi decoder was used.

3.2 Feature representation

As in [5] we have chosen to use the spectrograms as inputs to the CCN. The acoustic input are split into smaller frames chunks which are then converted into fixed size spectrogram images. Section 4 detail the spectrograms generation process for this phone recognition task.

3.3 Network Architecture

As suggested in [12], a CNN network with both convolutional layers and fully connected layers will be used. Convolutional layers will be used at the first (bottom) layers of the network, while fully connected layers will be used at the last (top) layers of the network. The convolutional layers will sometimes be followed with a pooling layer. Having the convolutional layers at the bottom of the network helps with the spectral variation, and the fully connected layer are used to discriminate between the different phonemes using the convoluted-pooled input from the convolution and pooling layers.
Our architecture follows after the famous AlexNet[8], a very successful network used in Computer Vision literature to do object recognition on more than 1,000 classes. Figure 1 shows the architecture of the best performing network according to our experiments. As we will see in Section 5.2, we have tried slightly modified versions of this architecture for different experiment.

3.4 Evaluation

A number of CNN with different number of layers will be trained. The performance of these networks will be compared using the classification error rate. Due to the limitation of the computing resources, a relatively small networks will be trained, also the network will be trained to classify over a subset of the phonemes.

4 Dataset

The Timit corpus [4] was used for verifying the proposed solution. It contains recordings of 630 speakers from 8 different regions in the united states. Each speaker read 10 sentences, recorded at 16 KHz frequency. Beside the audio recordings, the corpus contains time-aligned orthographic, word and phonetic transcription.

To create the spectrogram images that are used as input to our phone recognition network, the phonetic time alignment information are used to extract the frames associated with each phoneme. The frames of each phoneme are then split into 16ms Hanning windows with 15.5 ms overlap. A Fast Fourier Transform is then applied on the frames to generate a spectrogram, which are then padded to create a 128 by 128 pixels images. Figures 2 and 3 shows example spectrograms.

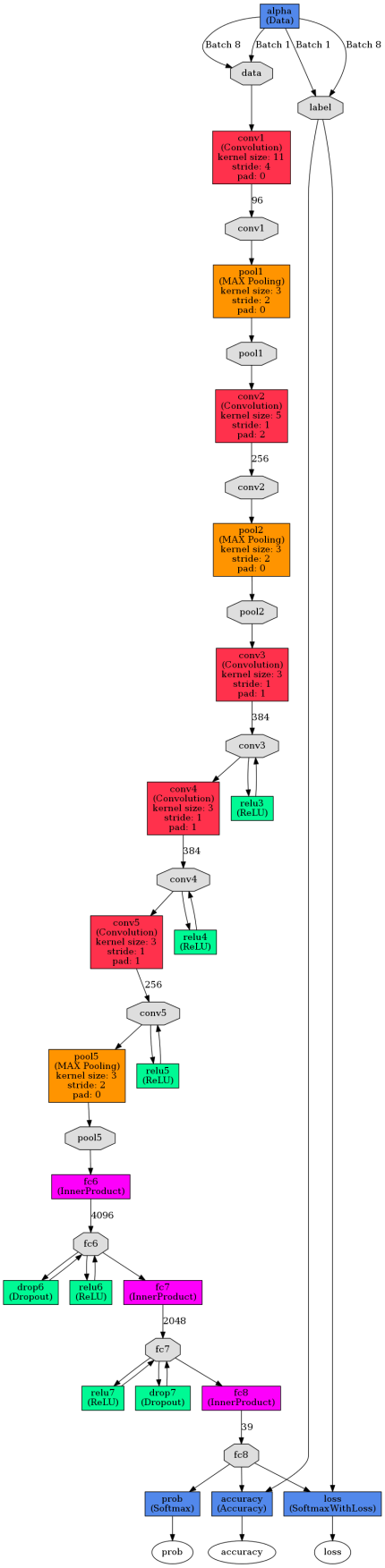

Figure 1: Best performing network architecture, it follows after AlexNet.

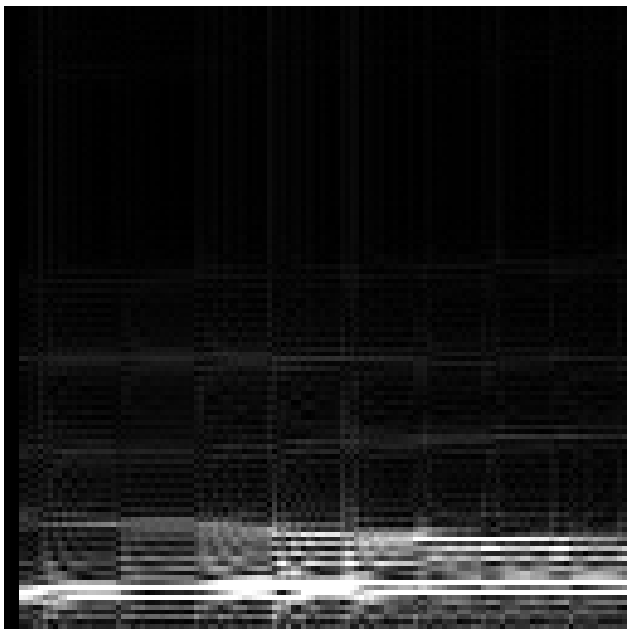

Figure 2: Spectrogram of phoneme **ao**

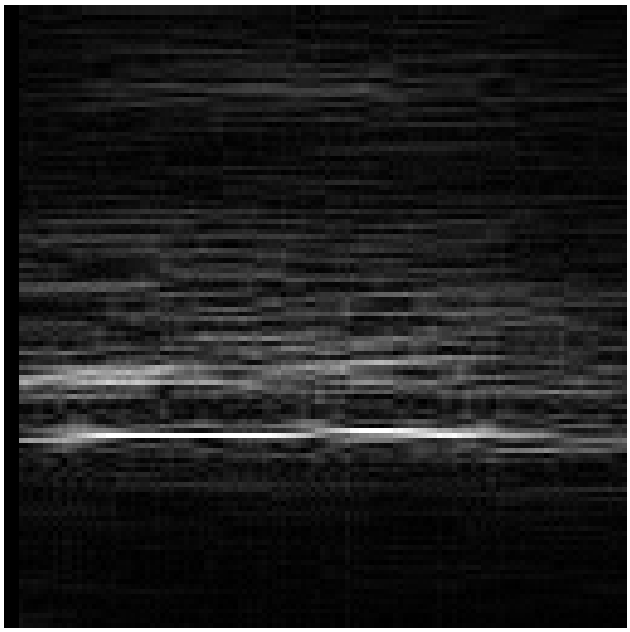

Figure 3: Spectrogram of phoneme **sh**

5 Experiments and Results

5.1 Implementation

The Timit corpus contains 6300 **wav** files corresponding to the 6300 sentences uttered by the 630 different speakers. The **Sndfile** from **scikits.audiolab** library is used to read the **wav** files. Using the phones time alignment information, the frames of each phoneme are extracted and saved as **wav** files, resulting a **wav** for each phoneme. The **wav** files are then converted into spectrogram images using a script provided by [3].

The CCN network was built and trained using the NVIDIA DIGITS deep learning system [2]. DIGITS providesa web based user interface that wraps the Caffe deep learning framework [1]. The training was done on a rented Amazon instance that has a GPU with around 1500 cores. The trained model is then tested using Caffe pycaffe module, since DIGITS provides poor testing facilities.

5.2 Experiments

The original 61 phonemes in the TIMIT corpus was mapped into 39 phonemes as suggested in mapping. This mapping is shown in table 1. The 39 phonemes are then used to both train and evaluate the model.

aa, ao	aa
ah, ax, ax-h	ah
er, axr	er
hh, hv	hh
ih, ix	ih
l, el	l
m, em	m
n, en, nx	n
ng, eng	ng
sh, zh	sh
uw, ux	uw
pcl, tcl, kcl, bcl, dcl, gcl, h#, pau, epi	sil
q	-

Table 1: Mapping the 61 original TIMIT phonemes(left) into 39 phonemes(right) as suggested in [10]

We have tried different networks with different data sets (train/validation

Name	Val Split %	Phone Count	Conv. Layers Setup	Fully Con. Setup	Accuracy	Epochs
EXP1	25	2 (dr1)	96, 256, 384, 384, 256, 256	256, 128, 2	100%	30
EXP2	25	61 (dr1,dr2)	96, 256, 384, 384, 256, 256	256, 128, 61	60%	30
EXP3	25	39 (all)	96, 256, 384, 384, 256, 256	4096, 2048, 39	65%	50

Table 2: Details of different experiments. Columns from left to right: experiment code-name, percentage of validation split, number of phonemes used and the regions included, number of nodes in different Convolutional layers, number of nodes in Fully Connected layers, Validation Accuracy and number of Epochs of the training.

Experiment	Unseen Test Error
EXP2	81%
EXP3	79%

Table 3: Error results on unseen test dataset.

split). As a proof of concept, we trained the network for only two phonemes. Figure 4 shows the graphs of training error and validation error as well as accuracy at each epoch for this network. Reaching 100% accuracy after 5 epochs, suggested that our data representation, along with the chosen network architecture, was in deed capable of separating different classes. Figure 5 shows a similar graph for our best performing method on all 39 classes.

Table 2 shows a detailed listing of the setup and performance of different experiments.

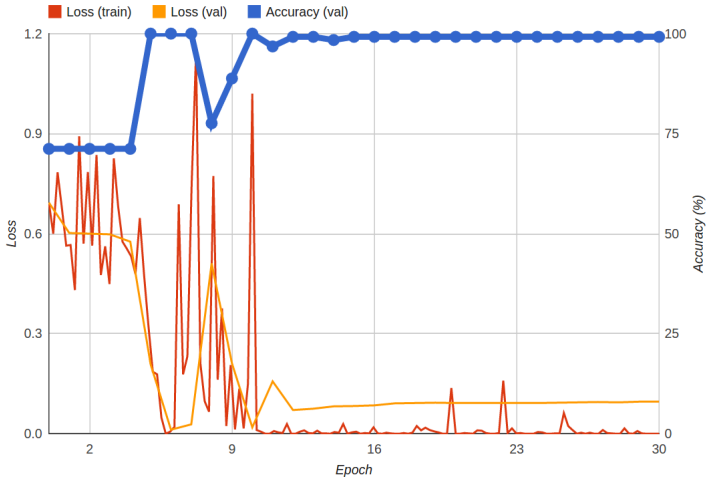

Figure 4: Training error, Validation error and Accuracy for each epoch of training the AlexNet on two classes.

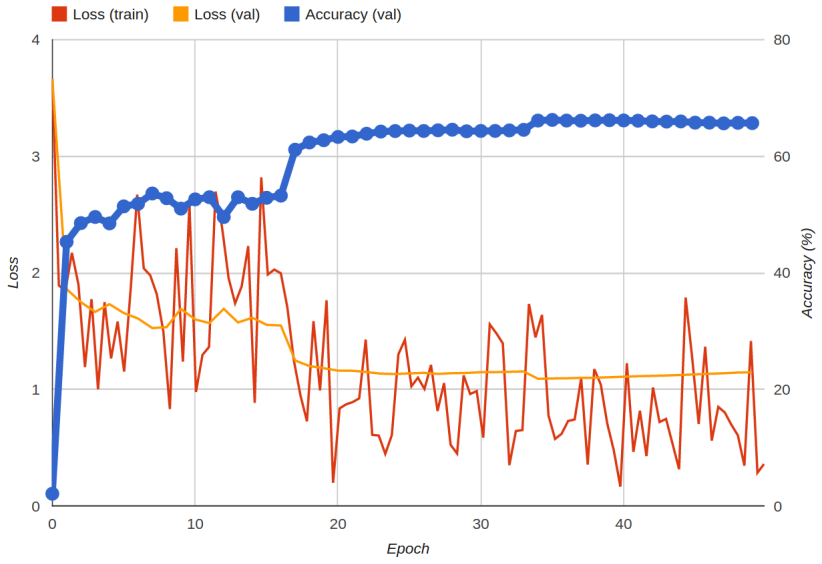

Figure 5: Training error, Validation error and Accuracy for each epoch of training the best model on all 39 classes.

6 Discussion and Conclusions

The method showed good potential in separating two phonemes, this was very beneficial as a proof of concept to show that the model works. However,

as seen from results in Table 3, our method was unable to generalize with good performance. However on the validation set, it was able to perform very well. The reasons behind this difference in performance is still unknown to us, however we propose some possible explanations and future work.

- The size and complexity of the network. Through all experiments, we were never able to get our model to overfit the data, this suggests that our model might not be complex enough. Another explanation is that the model needs much more epochs to learn.
- It might be the size of the spectrograms used. The size we used is small and does not capture variations that differentiate the different phone classes, for instance, Google successfully used words spectrograms.
- The representation of the data, while it was easily separated in the two phoneme case, the representation might not be separable for the 39 phoneme case. Therefore we might want to experiment with raw sound waves.
- The amount of data used. Unfortunately, due to the expensiveness of the training, we were only able to train only on 39 phonemes for all regions. Most of the literature trains on all 61 phonemes, and tests on a smaller subset (39 phonemes).

References

[1] Caffe deep learning framework. `http://caffe.berkeleyvision.org/`.

[2] Nvidia digits interactive deep learning gpu training system. `https://developer.nvidia.com/digits`.

[3] Speech recognition with bvlc caffe. `https://github.com/pannous/caffe-speech-recognition`.

[4] Garofolo, J., et al. Timit acoustic-phonetic continuous speech corpus. `https://catalog.ldc.upenn.edu/LDC93S1`.

[5] Graves, A., and Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (2014), pp. 1764–1772.

[6] Graves, A., Mohamed, A.-R., and Hinton, G. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (2013), IEEE, pp. 6645–6649.

[7] Hinton, G. E., and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science 313*, 5786 (2006), 504–507.

[8] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[9] LeCun, Y., and Bengio, Y. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks 3361* (1995), 310.

[10] Lee, K. F., and Hon, H. W. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on Audio Speech & Language Processing 37* (1989), 1641–1648.

[11] Robinson, A. J. An application of recurrent nets to phone probability estimation. *Neural Networks, IEEE Transactions on 5*, 2 (1994), 298–305.

[12] Sainath, T. N., Mohamed, A.-r., Kingsbury, B., and Ramabhadran, B. Deep convolutional neural networks for lvcsr. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (2013), IEEE, pp. 8614–8618.