UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Master in Telecommunications Engineering

Deep Learning for Speech and Language
Speech Commands Challenge

Johan Bender Koch and Miquel Llobet

January 30, 2018

Contents

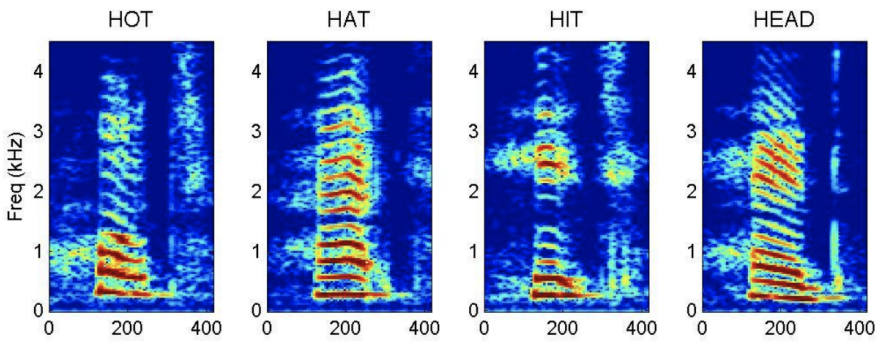
1 | Introduction

Computation systems for converting speech to text have challenged researchers for decades. Over the last few years, major improvements in Word Error Rate have been achieved by utilizing deep neural network (DNN) architectures [1]. When converting speech to text, some of the main challenges are the fact that humans pronounce words differently depending on several factors such as origin, age and intonation among others. Background noise noise can also make words difficult to interpret, and human error-rate is at around 5.1% on the hardest challenges [2]. By using non-linear functions, DNNs are able to separate noise and different forms of pronunciation when provided with enough diverse data. For our final project, we have chosen to work with the TensorFlow Speech Recognition Challenge [3]. In short, the objective is to classify 30 different voice commands from a dataset consisting of one-second long utterances. Using the provided code by professor Jose A. R. Fonollosa, we intend to test various experiments for improving the classification accuracy and understanding the underlying challenges of the dataset.

1.1 Dataset

The TensorFlow Speech Recognition Challenge [4] includes 65,000 one-second-long utterances, recorded by thousands of different speakers. Speakers are from all around the world and recordings are generally made in a noiseless environment such as a room. The files are encoded in WAVE form at a 16000 sample rate 1.1. For this project the recordings are pre-processed to the standard 40 log-mel frequencies. Padding is also added to fit all samples to the same size and z-score is normalized.

Figure 1.1: Sample dataset spectrograms



80% of the dataset is used for training with the remaining 20% evenly split in validation and testing.

1.2 Methodology

Although, speech recognition is known as a sequence challenge, we can exploit the fact that the dataset consists of one-second voice samples, which can be treated as images in the format of a spectogram. As such, it's possible to train Convolutional Neural Networks (CNNs) to output the correct label. Our work uses the provided implementation of standard CNNs in the pytorch Deep Learning framework and uses python and matplotlib visualizations.

1.3 Experiment Lists

For training purposes a total of 51,093 samples were used for every experiment. Given the time constraints, we opted for testing different hyper parameters for CNN architectures that have been known for high performance. Below is a list of the experiments and results 1.1:

1. Try default models
2. Try SGD instead of Adam
3. Try training with augmented data
4. Try an ensemble model
5. Visualize with Confusion Matrix

Table 1.1: Experiment Results

Accuracy	TDNN	VGG11 SGD	VGG11	VGG16	VGG16 *	Ensemble	Ensemble**
Validation	89.95%	90.80%	95.5%	91.13%	94.76%	N/A	N/A
Test	89.53%	90.40%	93.47%	90.84%	94.70%	95.19%	95.54%

* Trained on augmented dataset. ** Weighted voting ensemble.

1.4 Experiment 1: Baseline model architectures

Initially, we trained the proposed models, including TDNN [5], VGG11 and VGG16 [6] with the Adam optimizer. TDNN is a network proposed for phoneme detection due to it's shift-invariance (it delays and accelerates samples in time). VGGs are a set of CNNs used for generic image recognition tasks and are increasing in depth. The results of TDNN were worse than expected, but perhaps the most surprising result was VGG11 outperforming VGG16 - we expected the added depth (8 extra hidden layers) to more accurately recognize the voice patterns in the commands. This could be due to the complexity of VGG16 needing more time to train or more data to do well, we tested this hypothesis in future experiments.

1.5 Experiment 2: Optimizer tuning

Instead of training with the adaptive optimizer Adam, we tried using stochastic gradient descent on the VGG11 architecture. Generally, adaptive optimizers are used today, as they are able to converge faster. However, a recent paper from Berkeley [7] outlined that SGD, when tested on datasets such as CIFAR10, are able to generalize better. As seen in figure 1.2 and figure 1.2, the training accuracy of Adam converges faster and also generalize well.

Figure 1.2: Training Adaptive and Non-adaptive Optimizers

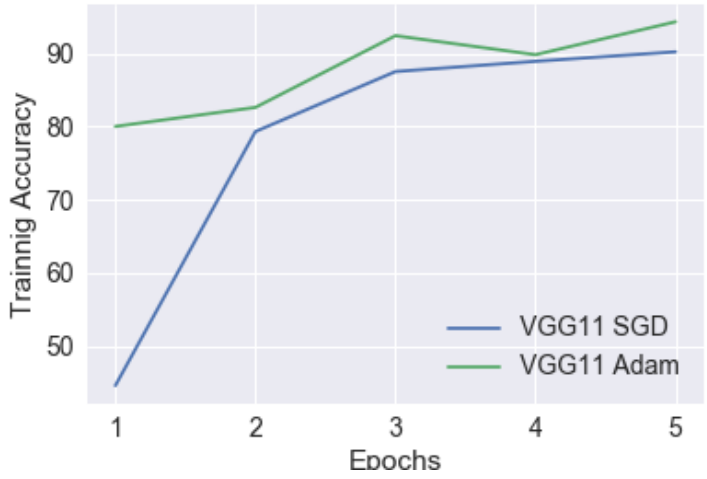


Table 1.2: Training Adaptive and Non-adaptive Optimizers

	VGG11 Adam	VGG11 SGD
Validation Accuracy	94.3%	90.80%
Test Accuracy	94.5%	90.40%

Given our resources, we were not able to run experiments of up to 250 epochs as in the Berkeley paper [7]. Since Adam performed best of the two optimizers, we decided to use Adam for the rest of our experiments.

1.6 Experiment 3: Data Augmentation

Sometimes better performances can be achieved by training on a larger dataset. Rather than collecting or generating new data from scratch, it is possible to augment the original data and thereby develop a better generalizable model. In our case, we had access to a dataset in which the original spectrograms were distorted in the time domain as if the durations were 0.9 seconds of 1.1 seconds long instead of 1 second. From this a dataset 3 times the size was generated. We decided to test on VGG16, which had shown a disappointing performance before. We believed training this more complex network on more data could result in better accuracy than previously observed. The results from the experiment proved that using the augmented data

improved the test accuracy - from 90.84% to 94.70%. We planned to test training VGG11 with the augmented dataset as well, but each run took significantly longer than before (4x the time).

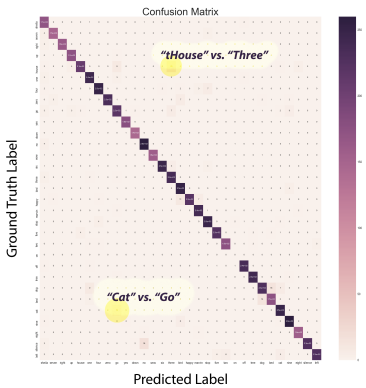
1.7 Experiment 4: Ensemble model

Instead of relying on the prediction of one model, results can sometimes be improved by combining the results of multiple models. This is a technique often used in competitions but rarely in the industry due to the extra resources required. There are several methods to implement ensembles, but we tried two: majority voting and weighted majority voting. In majority voting each classification is ran through all the models and the majority winner is selected, in case of tie one is selected at random. In the weighted variant the best performing models are added a voting boost that an be fine-tuned in testing. Since all our models have performed quite well, none of them were excluded, our ensemble contained TDNN, VGG11, VGG16 and VGG16 trained on augmented data. Results yielded better accuracy than any other model, with the weighted version (boosting VGG16 trained on augmented data) reaching 95.54% accuracy.

1.8 Experiment 5: Visualize with Confusion Matrix

In order to identify which words the model is not able to generalize well, we implemented a confusion matrix of the VGG11 SGD test set 1.3. The intensity of the color reflects the number of labels. The straight diagonal reflects a good prediction power. That said, the diagonal line does have an empty space for the label "on". This could be a mistake on our part, but also reflect that the model doesn't predict this particular label well. Among some of the labels that the model miss classifies are "Three" instead of "House" and "Go" instead of "cat". While "Cat" and "Go" sound similar pronunciation, "Three" and "House" are surprising miss matches. One possible explanation could be that both "House" and "Three" are words that take longer to pronounce compared to the other words. Therefore, it could be possible that the model simply learned to classify longer words. To explore further, we would have liked to visualize the labels with image embeddings as well.

Figure 1.3: Confusion Matrix of VGG11 SGD



1.9 Conclusion

The experiments showed some promising results, but the performance is suspiciously good. At 95.54% accuracy it is significantly higher than Kaggle's top team (91%). This is probably due to our test set being significantly easier than the competition. That aside, we believe there are several unexploited performance gains possible in this challenge:

- Explore further data augmentation
- Train all ensemble models with augmented dataset
- Explore other architectures

Top teams in the Kaggle competition explored advanced data augmentation techniques: from standardizing peak volumes to normalizing for vocal tract length perturbations [8]. The voice spectrum has many variation factors that make this a challenging task and data augmentation and normalization are very promising next steps to try.

Due to lack of time not all models could be trained with the augmented dataset, the ensemble model could benefit from this and yield better results. Other ensemble methods could be tried as well as only voting (simple and weighted) was tried. This would be a possible improvement avenue as the winning team had over 30 models in ensemble.

Finally other architectures could be explored. We restricted ourselves to the provided architectures, given our time-constraints, but there are many more standard image recognition with great results in the literature that could be tried for this task.

Bibliography

[1] G. Saon and M. Picheny. Recent advances in conversational speech recognition using convolutional and recurrent neural networks. *IBM Journal of Research and Development*, 61(4):1:1–1:10, July 2017.

[2] THE MICROSOFT 2017 CONVERSATIONAL SPEECH RECOGNITION SYSTEM W . Xiong , L . Wu , F . Alleva , J . Droppo , X . Huang , A . Stolcke Technical Report MSR-TR-2017-39 August 2017. (August), 2017.

[3] Kaggle speech commands challenge. https://www.kaggle.com/c/tensorflow-speech-recognition-challenge. Accessed: 2018-29-01.

[4] Google Brain. Research Blog: Launching the Speech Commands Dataset, 2017.

[5] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, Mar 1989.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[7] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning. pages 1–14, 2017.

[8] Navdeep Jaitly and Geoffrey E Hinton. Vocal Tract Length Perturbation (VTLP) improves speech recognition. *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013.