Audio Classification with Convolutional Neural Networks

Chibuzor John Amadi[502623]

[1] University of Pavia, Pavia
[2] University of Milano-Biccoca, Milan
[3] University of Milano-Statale, Milan

Abstract. Convolutional Neural Networks(CNN) in deep learning over the years has been well known for image classification with its ability to learn features from pixel data. In recent years, CNN has shown quite some exciting promise in audio classification. The dataset[1] contains the audio of spoken words which we will train and experiment to carry out keyword spotting. The CNN will use batch-normalization and down-sampling in the initial layers to perform classification of the raw waveforms of the audio data. Our evaluation of our model, being a predictive model, will be done with the accuracy metric.

Keywords: Audio processing · Raw Waveform · Keyword Spotting.

1 Introduction

Keyword spotting[2], the task of our project, aims to provide a way to build and test small models that detect when a single word is spoken, trying to minimizing the occurrence of false positives from background noise or unrelated speech. This project is centered around acoustic modeling, which can be described as the process of establishing statistical representations for the feature vector sequences computed from the speech waveform. Two major parts of this project is transforming and processing our audio data, the other is defining and refining a suitable model to carry out classification. In audio processing, the spectrogram and the raw waveform are the two major visual representation. Raw waveforms are favored when the time-domain is more relevant, while the spectrogram is favored when the frequency is more relevant. For the case of this project, we will study a deep CNN using time-series waveforms as the input. Building a suitable model, CNN to be more precise, will entail carefully designing a convolutional layer, an detector layer and a pooling layer. The deep network are to be efficient to optimize over long sequences necessary for processing raw audio waveform. The architecture of the model will be described a very deep convolutional neural network[4]. Batch normalization is applied to overcome training difficulty and keep the computational cost low. The choice to use a deeper CNN is inspired be its likelihood to perform better due to its ability to learn more complex, abstract representations of the audio data, handle greater variations within the

data, and generalize better from these representations to make accurate classifications. This of course was experimented in this project as the very deep CNN was compared to a more swallow neural network.

2 Model Architecture

2.1 THE CONVOLUTIONAL NEURAL NETWORK

In this work, 2 architectures will be considered namely the M5 and subsequently the M3 following [4] , the breakdown of these architectures are displayed in Figure 1 1 and Figure 2 2. The input to our model time-series waveforms, represented as a long 1D vector.To design very deep networks, we use very small receptive field 3 for all but the first 1D convolutional layers. This was done to reduce the number of parameters in each layer and control the model sizes and cost of computation.We used a large convoltional and max pooling stride of 16x in the first two layers to reduce the temporal resolution with a consideration of the computation cost in the rest of the deep network [9]. After the first two layers, the reduction of resolution is complemented by a doubling in the number of feature maps2 . We use rectified linear units (ReLU) for lower computation cost, following [7,6].

Many deep convolutional networks for classification use two or more dense layers with high dimensions, significantly increasing the number of parameters. We believe that critical learning occurs in the convolutional layers, and if these are sufficiently expressive, dense layers might be unnecessary. Therefore, we've designed our network to be fully convolutional, eliminating dense layers in favor of a single global average pooling layer that reduces each feature map to a single value by averaging activations across the temporal dimension. This design encourages better representation learning in the convolutional layers and potentially improves generalization.

Waveforms sampled at rates like 8000Hz produce large single-dimension sample sets. Using small receptive fields across all layers, as seen in some models with 3x3 fields, necessitates many layers to abstract high-level features, increasing computational costs. Moreover, the actual length of a receptive field such as 80 varies with the sampling rate, being different at 8kHz versus 16kHz. Our tests, which we will display in the evaluation section, show that much smaller or larger fields yield poorer performance.

We use batch normalization (BN) layers, as described in [8], to address the issues of exploding and vanishing gradients often encountered in deep network optimization. BN normalizes the outputs from each convolutional layer, stabilizing the gradients before we apply the ReLU non-linearity.

3 Evaluation

3.1 Experiment details

The loss function of of the model we will be using is the Negative Log Likelihood (NLL) which is suitable for our multi-class classification. The NLL loss measures

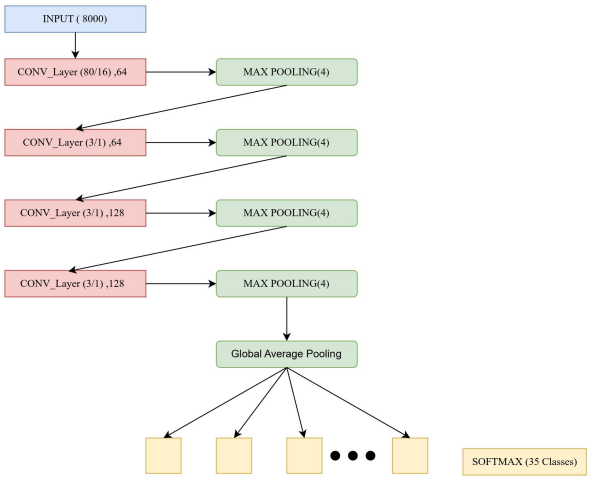

Fig. 1. M5 model. Description: The input audio is represented by a single channel.(80/16), 64 denotes a convolutional layer with receptive field 80 and 64 filters, with stride 16. Stride is omitted for stride 1 example, (3/1), 64 has stride 1. Finally its passed to a global average pooling and then to a softmax activation function

the performance of a classification model whose output is a probability value between 0 and 1, its equation can be defined as

$$NLL = -L(y) - (1)log(y) \tag{1}$$

To train the M5 model I utilized Adam[3], a variant of the stochastic gradient descent. Adam is an optimization method used to minimize the Negative Log Likelihood loss. The results of the M5 model was improved on training by adjusting the learning rate, a scheduler was also introduced to adjust the learning rate of the optimizer over time. The CNN model was evaluated finally on an accuracy performance metrics. Table 1 1 shows the result of the M5 model on different learning rates.

Table 1. *Results of M5 Model on different Learning rates. Note: the learning rate shows the initial learning rate of the model before introduced to the scheduler, the results of the accuracy is in a percentage value*

LR = 0.1	LR = 0.01	LR = 0.001
69	74	80

4 Further Experiments

Experiments details With the optimal M5 model, We decided to experiment with inspiration from[4], the effect of the architecture of these models. By introducing the M3 model, we will alter the architecture of the M5, train with the best proven optimizers and study its outcome. Different from the M5, the M3 will have less layer, and a different startup on the number of filters. The M3 will have 2 layers and as such with have about 60 percent less total parameters than the M5.Figure 2 2 derived from [4] shows the architecture of the M3.

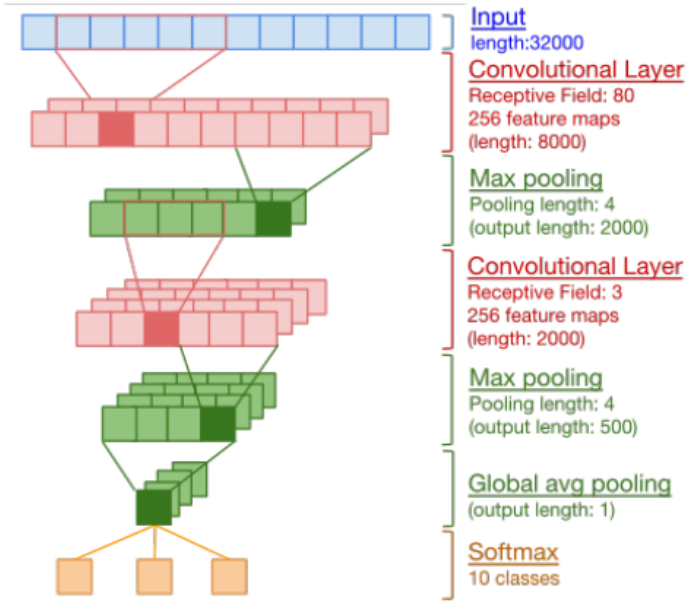

Fig. 2. M3 model Architecture

4.1 Result Analysis

Analysis the results as shown in Table1 1 derived from the different variations of our M5 model with a change in the learning rate. From the table we can observe that when the model is trained with an initial learning rate of 0.1 achieved considerably poor result of 69 percent on accuracy. However, the baseline on 0.01 learning rate achieved a regular accuracy score of 74 percent on several testing. The best result however came from the learning rate of 0.001, this achieved an accuracy score of 80 percent on consistent. With this we can see that the lower the learning rate on the adam optimizer the better the performance accuaracy is of the model.

The M3 model was trained and tested on the optimal parameters we used on the M5. The learning rate was set to 0,001 and we maintained the adam optimizer. The result however for these two models differs by a good margin as the best performance of the M3 model was the 47 percent of various training occasions. the M3 performing very poorly compared with the other model,indicates that 2-layered Convolutional Neural Networks are insufficient to extract discriminative features from raw waveforms for sound recognition [5].

5 Conclusion

Concluding that the M5 is the suitable model for the task, we decide to assess the performance on single classes by plotting out a confusion matrix. The confusion matrix as shown in Figure 3 3 has a clear diagonal line that goes across it. The line in its different colours shows the performance of the predicted label against the true label. We can see that the best performances occurred with words of less letters or single syllables. This shows that the raw waveform of single syllable were easier to distinguish and predict compared to longer words or double syllables. Another observation is that there were some common mis-classification with similar words like "three", "two" and "tree" , "zero" and "go". We can attribute this mis-classification to the basic notation these words sound very similar. With further training and more computational resources, we can be optimistic to say that the M5 model can achieve better classification.

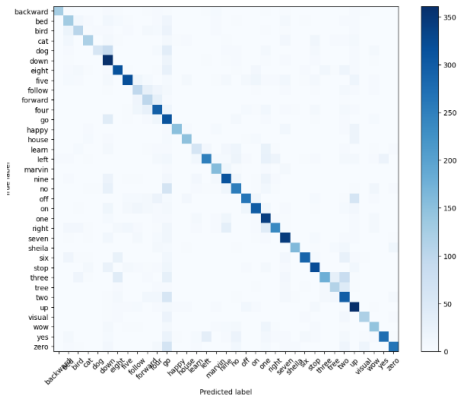

Fig. 3. Confusion Matrix on Single Class Performance

6 References

1 "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition". Pete Warden, Google Brain, Mountain View, California, June 2019.

2 "Keyword Spotting" Wikipedia, en.wikipedia.org/wiki/Keyword-spotting
3 "Stochastic Gradient Descent." Wikipedia, en.wikipedia.org/wiki/Stochastic-gradient-descent/Adam
4 "Very Deep Convolutional Neural Networks for raw waveforms" Wei Dai*, Chia Dai*, Shuhui Qu, Juncheng Li, Samarjit Das, 2016
5 "Very Deep Convolutional Neural Networks for raw waveforms" Wei Dai*, Chia Dai*, Shuhui Qu, Juncheng Li, Samarjit Das, 2016 (section 4)
6 Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
7 Matthew D et al. Zeiler, "On rectified linear units for speech processing," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 3517–3521.
8 Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.
9 Christian et al Szegedy, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.