Audio Tagging: LSTM and Gated CNN Model
MS318 Deep Learning Course Project

Author
Jinning Li, 515030910592
Shanghai Jiao Tong University
lijinning@sjtu.edu.cn, http://jinningli.cn

Abstract

In this paper, the LSTM model and gated Convolutional Nerual Network model are used to solve the audio utterance tagging task. Pytorch, TFLearn, and Keras are used to complete these model. Two baselines of fully connected model are built to evaluate the result. Experiments are conducted to analyze the speed and performance of different models. Results proves that the LSTM model built with TFLearn achieves the best performance and the TFLearn platform is the fastest.

1 Background

Sounds carry a large amount of information about our everyday environment and physical events that take place in it. We can perceive the sound scene we are within (busy street, office, etc.), and recognize individual sound sources (car passing by, footsteps, etc.).

Developing signal processing methods to automatically extract this information has huge potential in several applications, for example searching for multimedia based on its audio content, making context-aware mobile devices, robots, cars etc., and intelligent monitoring systems to recognize activities in their environments using acoustic information.

However, a significant amount of research is still needed to reliably recognize sound scenes and individual sound sources in realistic soundscapes, where multiple sounds are present, often simultaneously, and distorted by the environment.

2 Task Definition

In this task, we are going to tagging a large scale of utterances. Each utterance is about 10 seconds, features are extracted at 1Hz. The features are 128 dimensional vectors, extracted from a bottleneck layer of ResNet. Each utterance may have several labels and there are 527 kinds of labels in total.

The dataset we use is part of the Audio Set [2] built by google. There are 22160 pieces of utterances and labels in the training data and 20371 pieces of utterances and labels in the evaluation data. The ontology is specified as a hierarchical graph of event categories, covering a wide range of human and animal sounds, musical instruments and genres, and common everyday environmental sounds, for example, human voices, engines, and wild animals. In a word, our task is to predict the label when givn an utterance.

3 Data Preprocess

The timestep of the utterance is not certain, so that a padding operation need to be conducted. In my model, zero padding is employed at the end of the utterances. In the LSTM and GCNN model, all the utterances are padded to size of 10.

4 LSTM Model

The architecture of LSTM Model is very simple. There are three LSTM layers, which receive the input of size $[N, T, D]$, where N is the mini-batch size, T is the time step. D represents the dimension. In this model, $T = 10$ and $D = 128$. The first LSTM layer includes 512 hidden layers. The second one includes 256 hidden layers. And 128 for the third layer. Then, the output of the third LSTM layer at each timestep $t \in T$ are concatenated and flattened and thrown to three fully connected layers with the size of 2048, 1024, and 527.

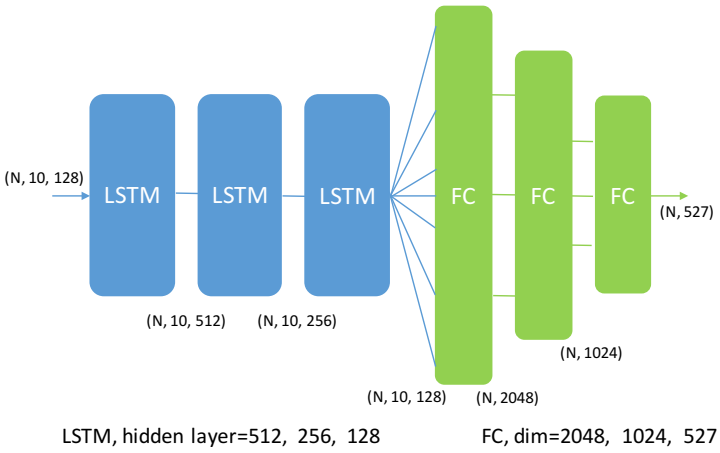

Figure 1: Architecture of LSTM model

The reason why I choose LSTM is that LSTM is an effective architecture solving time series data. When I consider this tagging problem, the simplest solution is using several fully connected(FC) layers. However, the performance of FC model is not good enough. Then I decide to add some LSTM layer before that to settle the time series information.

However, some problem also appear. One is the gradient explosion. It seems that in the backward process, the gradient of parameters of LSTM often explode to an very large scale. So I employed the gradient clip method to control the gradient.

5 Gated Convolutional Neural Network Model

In [3], the author propose the *gated convolutional neural network*(GCNN) model. I changed some architecture of this model, and find it performs well.

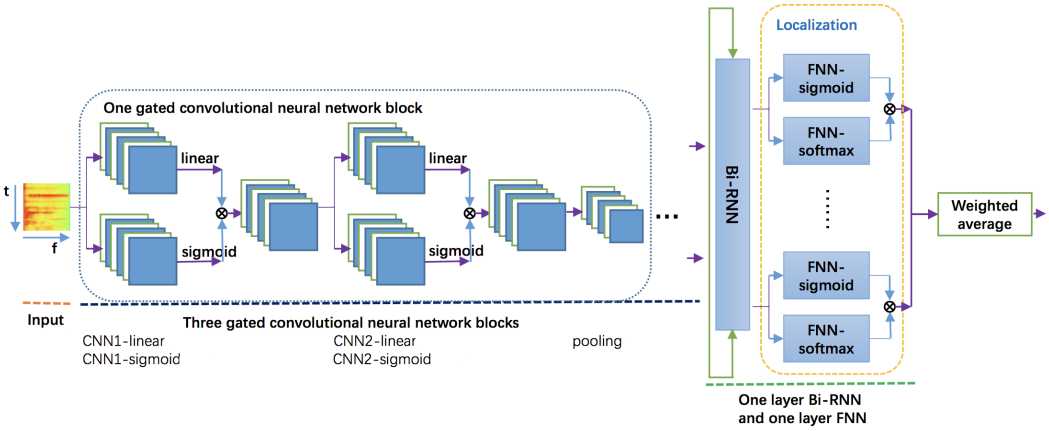

Figure 2: Architecture of GCNN

The input of GCNN is the tensor of utterance. In our problem, the size of input is $(N, 10, 128)$. Then, the tensor go through some GCNN blocks.

In the beginning of each GCNN block, there is a CNN layer with kernal size of 2 and filter number pf 128. Padding is used to maintain the shape of tensor. Then, the output is sliced into two part with a dimension of 64. One is applied with linear activation and another with sigmoid activation. Then these two part is combined.

In my model, there is two GCNN blocks. Then, I let the output go through two pairs of CNN layers and maxpooling layers for down sampling. Then, the size of tensor become $[N, 256, 128]$, where N is the size of mini-batch, 256 is the dimension of features, 128 is the dimension of audio. Intuitively, this is an extraction of time series information.

Then, there is two bi-directional recurrent neural network (Bi-RNN) adopted to capture the temporal context information followed by a FC layer with the number of audio classes to predict the tags. In the paper, the author propose the *gated linear units*(GLUS) as the final activation.

GLUs are first proposed in [1] for language modeling. The motivation of using GLUs in audio classification is to introduce the attention mechanism to all the layers of the neural network. The GLUs can control the amount of information of a T-F unit flow to the next layer. By this means the network will learn to attend to the audio events and ignore the unrelated sounds.

6 Evaluation Metrics

mAP and mAUC are adopted as the evaluation metrics. Average precision(AP) computes the area under precisionrecall curve. AUC, computes the area under the ROC curve, which is a TPR-FPR curve.

7 Experiments

7.1 Environment

All the experiments are conducted on a PC device (Intel(R) Core(TM) i7-6900K 3.2GHz, 16GB memory, NVIDIA GeForce(R) GTX 1080, CUDA 8.0). All the code are completed with Python 3.6.1. The versions of used platforms are: Pytorch 0.3.0, TFLearn 0.3.2, Keras 2.0.8, Tensorflow 1.3.0.

7.2 Baselines

Two baselines are set up. The first one is that with only one fully connected layers of 527 dimensions, denoted by FC-1, The second one is that with two fully connected layers of 1024 and 527, denoted by FC-2.

Because of the limitation of time, the big training set is not used. The models I implemented includes: LSTM-TFLearn, LSTM-Keras, LSTM-Pytorch, GCNN-Keras, FC-1-TFLearn, and FC-2-TFLearn.

7.3 Speed Evaluation

To evluate the speed of different model, I test them in the same devices, set the hyperparameters as: epoch=10, learning rate=0.001, optimizer=*Adam*, batch size=64. The result is shwon in Fig.3. The runtime is measured by the occupation of CPU and GPU.

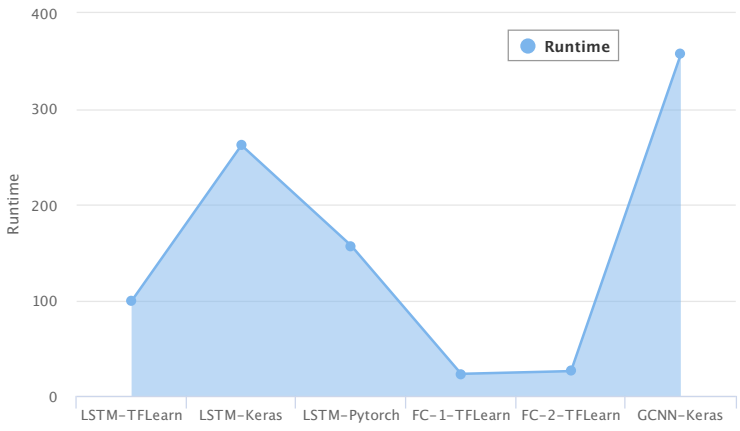

Figure 3: Runtime of each model

From Fig.3, we can see that when running LSTM model, the TFLearn platform is the fastest, it cost only 99.13 units of time. The Keras platform is the slowest. It cost 261.93 units of time. When comparing different model, the FC-1 is the fatest, costing only 23.01 units of time. The GCNN is the slowest, it cost up tp 357.20 units of time.

7.4 Performance Evaluation

When evaluating the performance, each model should have different hyperparameters that most suitable for it. So I run them with different hyperparameters and draw the maximum, minimum, and average mAUC on the evaluation set in Fig.4

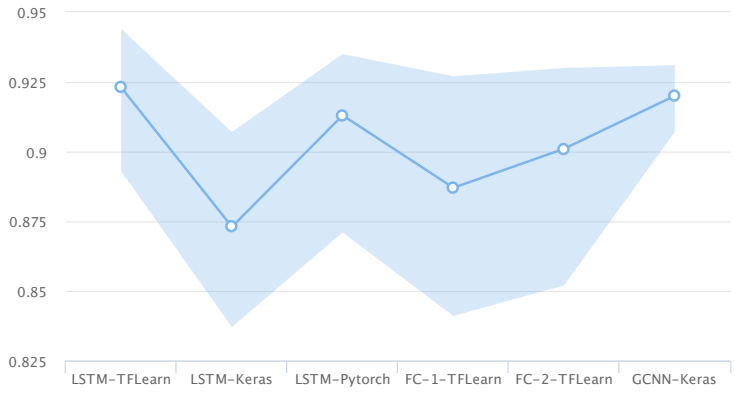

Figure 4: mAUC value of each model

As is shown in Fig.4, the LSTM-TFLearn model achieves the best performance with 0.945 maximum mAUC value. The LSTM model in Keras performs not so good, the average mAUC is about 0.873, this might be because I use different loss function for TFLearn and Keras platforms. And actually, the fully connected model performs not bad. The FC-1 achieves 0.927 of the maximum. For the GCNN model, the result is not the best. But I think the GCNN model will get a better perfromance on the big dataset, since its architecture is better to solve a large scale dataset.

8 Conclusion

The audio tagging task is really interesting. LSTM and GCNN model perform very well on the audio dataset. As for the future work, I hope to process the real audio recodings and build a complete system. I have learnt much by doing this task. Last but not the least, thanks our hard-working assistant and prof. Yu.

References

[1] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

[2] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE ICASSP*, 2017.

[3] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley. Large-scale weakly supervised audio classification using gated convolutional neural network. *arXiv preprint arXiv:1710.00343*, 2017.