MSoS: A Transfer Learning based approach

Souvic Chakraborty

[1] IIT Kharagpur
[2] chakra.souvic@gmail.com

Abstract. We present four submissions of the Making Sense of Sound challenge here. As the data-set used for training is small we had to go through a lot of trouble to avoid over-fitting on the data set. We tried popular pre-processing techniques like mel-spectrogram and stft of the signal in dB scale. We could only reach 65% validation accuracy that way. So, we finally decided to use transfer learning and used a vgg-like model and google's audio-set data for training on and obtained 128 point embedding for each second of data to use various techniques on that and obtained up to 88% validation accuracy.

Keywords: Neural Network · Transfer Learning · Data Augmentation.

1 Methods and Results

1.1 Transfer Learning

After failing to solve the problem with over-fitting due to very small size of the data-set, we decided to make use of transfer learning to learn the lower level filter weights easily on an extended data-set.

So we used google's audio-set dataset and trained our vgg-like system on that. Getting the trained net from their removing the dense layers we got a machine that translates 5 second data to 5 embeddings of size 128 points. We used a feed-forward neural net on that, a CNN architecture on that and a CRNN architecture. The results were promising with upto 82% validation accuracy.

However, to achieve higher accuracy than that was being a tough task with any neural net architecture. So we concluded that, as the training data accuracy was still improving after 80-82% but the validation accuracy was not improving, we must be over-fitting on the data-set as still the training error was decreasing even after the testing accuracy saturates.

So, we tried augmenting the dataset with data generated from the given data as the final tagged dataset is limited. So, we used affine transforms which can be used in case of sound data to enlarge the training data.

1.2 Affine Transform

To increase the size of the dataset, we changed the position of the sound wave by 10% or 0.5 seconds in either direction. o from 1500 samples we created 15000 samples and with these data going through the pretrained VGG like model we obtained 5 vectors of size 125 dimensions for each of the datapoint. Now,in order to avoid overlap we divided the data in two parts where all affine transforms of a particular original datapoint will fall in the same division(training or validation). Here our goal is not to achieve high training accuracy in the augmented dataset but to use augmented dataset to train for even worse scenarios so that it gives high accuracy in case of original validation data.

1.3 Models

Pre-processing All audio is resampled to 16 kHz mono. spectrogram is then computed using magnitudes of the Short-Time Fourier Transform with a window size of 25 ms, a window hop of 10 ms, and a periodic Hann window. A mel spectrogram is computed by mapping the spectrogram to 64 mel bins covering the range 125-7500 Hz. Now,a stabilized log mel spectrogram is computed by applying log(mel-spectrum + 0.01) where the offset is used to avoid taking a logarithm of zero. These features are then framed into non-overlapping examples of 0.96 seconds, where each example covers 64 mel bands and 96 frames of 10 ms each.

Models We used these layers in sequential order to get the embedding for each 0.96 seconds: 64 filter conv2D(3x3), Maxpool(2x2),128 filter conv2D(3x3), Maxpool(2x2), 256 filter conv2D(3x3), 256 filter conv2D(3x3), Maxpool(2x2), 512 filter conv2D(3x3), 512 filter conv2D(3x3), Maxpool(2x2), flattened, Dense(4096),Dense(4096), Dense(128) All activation used are ReLU. On these embeddings we used 3 best performing models (FFNN with 128 hidden nodes, dropout of 0.7 and BatchNormalization in both layers, CNN with 512,128 and 64 filter single channel 1D convolution and then the FFNN architecture as the fully connected layer and for the third an elaborate biLSTM with 128 hidden nodes connected with the FFNN architecture).The results are mentioned below: We also used a PCA applied embeddings with FFNN to get best validation accuracy of 90% In the final submission we use ensemble of PCA and raw input FFNN output ensemble.

Table 1. Results

Heading level	Without Affine Transform	With Affine Transform
0.77 FFNN	0.78	0.77
CNN	0.72	0.73
CRNN	0.75	0.77

References

1. Audio Dataset used for pre-training https://github.com/tensorflow/models/tree/master/research/audioset