2024-Spring EE738 Project Report

Donguk Kim
Department of Electric and Electrical Engineering
KAIST
kdu3613@kaist.ac.kr

Abstract

To enhance the speech recognition acoustic model performance, I tried to do my best effort on the model architecture, data augmentation, and training strategy. I used conformer blocks and Transformer layers in the architecture of my acoustic model. I utilized conformer layers and transformer layers from the PyTorch library to enhance the acoustic model's resolution capability. For data augmentation, I add noise using the PyTorch library. In the training strategy, I incorporated a learning rate scheduler and included pre-training and fine-tuning steps.

1 Baseline Result

In the given skeleton code, after filling in the blank code parts and training based on the given hyperparameters, the Character Error Rate (CER), the baseline's training log and the validation dataset was as follows:

```
logs > train.log
Namespace(max_length=10, train_list='data/ks_train.json', val_list='data/ks_val.json', labels_path='data/label.json', train_path='data/kspon_train',
Epoch 000, train loss 5.611, val loss 4.847
Epoch 001, train loss 4.439, val loss 4.182
Epoch 002, train loss 2.971, val loss 2.550
Epoch 003, train loss 2.133, val loss 2.107
Epoch 004, train loss 1.820, val loss 1.902
Epoch 005, train loss 1.637, val loss 1.728
Epoch 006, train loss 1.515, val loss 1.604
Epoch 007, train loss 1.419, val loss 1.530
Epoch 008, train loss 1.348, val loss 1.454
Epoch 009, train loss 1.290, val loss 1.402
```

Figure 1: **Baseline training log**

After completing 10 epochs of training, the model achieved a train loss of **1.290**, a validation loss of **1.402**, and a CER of **35.81%**.

2 Model Architecture

To improve the sound model's ability to capture details, I modify the model structure using the PyTorch library. Fig. 2 illustrates the overall architecture of my refined acoustic speech recogition model.

2.1 Conformer Layer

First, conformer layers that combines convolutional and transformer modules is added to enhance the quality of representations. Fig. 3 is detail code set up for conformer layers.

Preprint. Under review.

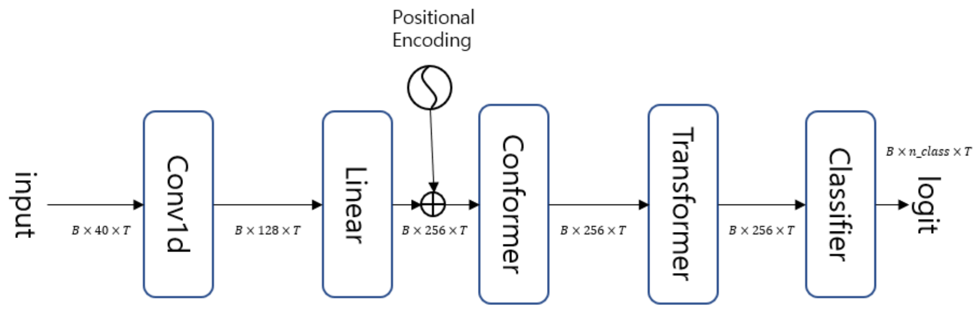

Figure 2: **Overall framework the acoustic model**

2.2 Transformer Layer

To better analyze the interaction between sound chunk vectors (tokens), I incorporated transformer layers into my acoustic model.Fig. 3 is detail code set up for Transformer layers.

```
self.CE_Block = torchaudio.models.Conformer(
    input_dim=256,
    num_heads=4,
    ffn_dim=512,
    num_layers=4,
    depthwise_conv_kernel_size=11,
    dropout=0.1
)
```

```
self.T_Block = nn.TransformerEncoder(
    nn.TransformerEncoderLayer(
        d_model=256,
        nhead=4,
        dim_feedforward=512,
        dropout=0.1),
    num_layers=4
)
```

Figure 3: **Conformer Layer Set up**

3 Training Details

To perform data augmentation, I include a 2dB background noise in the input sound. For the learning rate scheduler, I utilized the PyTorch LambdaLR scheduler, which employs a lambda expression $\lambda\ epoch :\ 0.95^{epoch}$. In the first pre-training step, the learning rate is set to $1e^{-4}$. I trained the model for 20 epochs using an un-augmented dataset. In the second fine-tuning step, I utilized learning rate scheduler and training data is added noise with probability 0.3. I fine-tuned the model with 10 epochs.

4 Results

```
Namespace(max_length=10, train_list='data/ks_train.json', val_list='data/ks_val.json', labels_path='data/label.json', train_path='data/
Epoch 000, train loss 1.074, val loss 1.146
Epoch 001, train loss 1.079, val loss 1.143
Epoch 002, train loss 1.078, val loss 1.153
Epoch 003, train loss 1.056, val loss 1.169
Epoch 004, train loss 1.041, val loss 1.116
Epoch 005, train loss 1.002, val loss 1.111
Epoch 006, train loss 0.993, val loss 1.103
Epoch 007, train loss 0.977, val loss 1.105
Epoch 008, train loss 0.977, val loss 1.093
Epoch 009, train loss 0.967, val loss 1.063
```

Figure 4: **Fine-tuning log**

Fig.4 is training log for second step fine-tuning and the result CER is **26.74%**.