Week 6-2: Paper Summaries

CE-510 Seminar: Social Media Mining

Student Name: Jiaqi Guo NetID: JGR9647

■ Deep Speech: Scaling up end-to-end speech recognition

The author abandons the traditional speech recognition model framework and turns to the end-to-end speech recognition model framework based on deep learning:

1. Discard the original feature engineering, such as MFCC.

2. Discard phoneme dictionaries and the concept of phonemes.

3. Use RNN network

In general, the model structure consists of two parts:

1. AM (Acoustic Model): Bi-directional RNN (loss function CTC, Optimizer NAG, Nesterov Accelerated Gradient)

   The model has **five hidden layers**, none of which is a recurrent layer except the fourth, which is bi-Directional layer.

   The calculation formula of the first three layers is as follows:

$$h_t^{(l)} = g\left(W^{(l)} h_t^{(l-1)} + b^{(l)}\right)$$

   For the 4th layer:

$$h_t^{(f)} = g\left(W^{(4)} h_t^{(3)} + W_r^{(f)} h_{t-1}^{(f)} + b^{(4)}\right), h_t^{(b)} = g\left(W^{(4)} h_t^{(3)} + W_r^{(b)} h_{t+1}^{(b)} + b^{(4)}\right)$$

   5th layer:

$$h_t^{(5)} = g\left(W^{(4)} h_t^{(4)} + b^{(5)}\right)$$

$$h_t^{(4)} = h_t^{(f)} + h_t^{(b)}$$

   The output is a standard SoftMax:

$$h_{t,k}^{(6)} = \frac{\exp\left(W_k^{(6)} h_t^{(5)} + b_k^{(6)}\right)}{\sum_j \exp\left(W_j^{(6)} h_t^{(5)} + b_j^{(6)}\right)}$$

   Where, $h_{t,k}^{(6)}$ represents the probability that the model predicts that the letter corresponding to the speech in frame T will be the K^{th} letter in the dictionary.

2. LM (Language Model): N-gram, which is trained with KenLM.

Possible Improvement Directions:

1. The assumption that CTC is conditional independent is too strong to be true, so language models are needed to improve conditional dependence to achieve better results

Sequence to Sequence Learning with Neural Networks

Although DNNs performs well on large-scale labeled data sets, it cannot solve the SEQ2SEQ (sequence to sequence) problem. In this paper, the author proposed an end-to-end approach that makes minimal assumptions about sequence structure, using multi-level LSTM to map input sequences to fixed-dimensional vectors.

The LSTM model implemented in this paper is illustrated below:

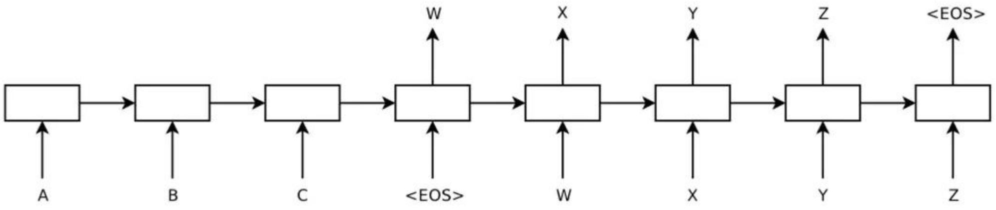

Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

The goal of LSTM is to optimize the conditional probabilities p:

$$p(y_1, \dots, y_T' \mid x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t \mid v, y_1, \dots, y_{t-1})$$

Here, x_i is the input sequence, and y_i is the corresponding output sequence. LTSM calculates this conditional probability by first obtaining a fixed dimensional vector representing v of the input sequence (derived from the last hidden state) and then calculating the probability of y_i by a standard LSTM-LM formula.

Possible Improvement Directions:

1. The model is insensitive to the active and passive voice of a sentence.