National Taiwan University Applied Deep Learning
Assignment 3
Natural Language Understanding
Chia-Hsuan-Li, Electrical Engineering

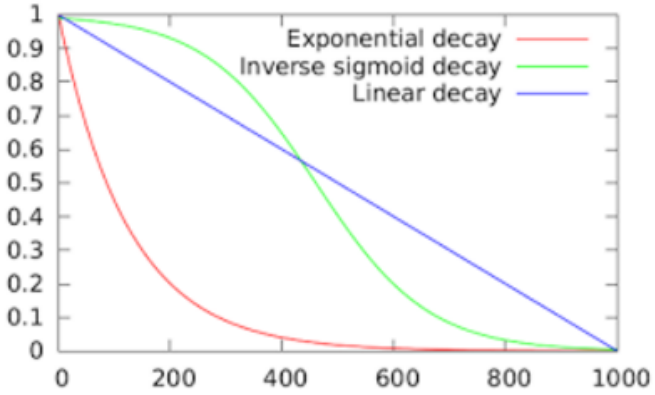
Overview

This is a recurrent neural network (RNN) model that jointly performs intent detection, slot filling and language modeling as the input word arrives.
The generated intent class and slot labels are useful for next word prediction
In this work, LSTM cell is used as the basic RNN unit for its stronger capability in capturing long-range dependencies in word sequence.

In my opinion, there are two novelties in this model.

(1) JOINTLY TRAIN：INTENT CLASS AND SLOT FILLING CAN BE CONTEXTUAL FEATURES TO MAKE EACH OTHER BETTER.
(2) SAMPLING METHOD TO MAKE NEURAL NETWORK MORE ROBUST FROM PREVIOUS PREDICTION MISTAKES IN THE SAME SENTENCE

Slot label dependencies can be modeled by feeding the output label from the previous time step to the current step hidden state.
In training phase, we can feed the last true label. And it will definitely help.
But in inference phase, there is only predicted label left to be used. This means that they never learn to recover from their own mistakes. If they make an error (which they will), they could end up in some distribution they've never seen at training time and behave arbitrarily badly.
So there is a compromised way to do the model training, we use a probability distribution to decide whether to use the true label or the predicted label.
Therefore the model are forced to take not perfectly correct information from the previous step and will become more robust.
Scheduled sampling considers two rates, one linear and one inverse sigmoid. The probability of using the true label decay when the training batches increase.



Model Architecture

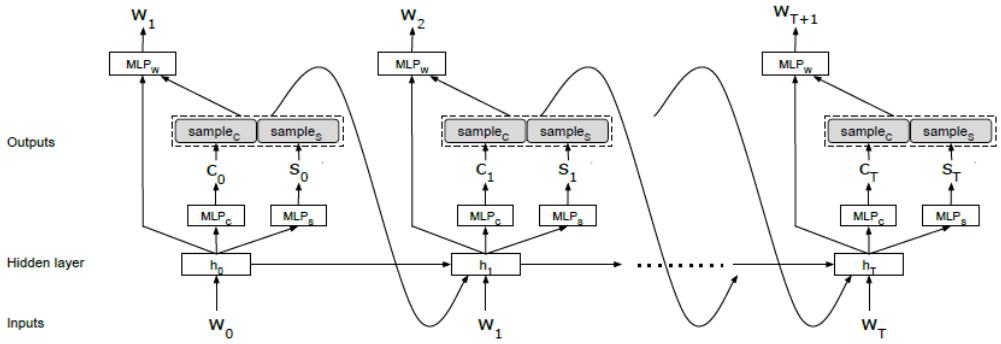


Figure 3: Proposed joint online RNN model for intent detection, slot filling, and next word prediction.

$$h_t = \text{LSTM}(h_{t-1}, [w_t, c_{t-1}, s_{t-1}]) \quad (7)$$

$$P(c_t | w_{\leq t}, c_{<t}, s_{<t}) = \text{IntentDist}(h_t) \quad (8)$$

$$P(s_t | w_{\leq t}, c_{<t}, s_{<t}) = \text{SlotLabelDist}(h_t) \quad (9)$$

$$P(w_{t+1} | w_{\leq t}, c_{\leq t}, s_{\leq t}) = \text{WordDist}(h_t, c_t, s_t) \quad (10)$$

The main cell is LSTM for its capability of capturing long sequence information. C is the prediction of intent class. S is the prediction of slot labeling. W is the prediction of the next word. (The language model output) Each hidden state is the production of LSTM based on the word input, last hidden state, and the output of last intent prediction、 last slot filling. IntentDist, SlotLabelDist and WordDist are three different Multi-layer perceptrons.

Code description

run_rnn_joint.py : The main program that uses codes from other python files to train and inference. It also receives the model hyper-parameters, print out the training history (perplexity and F1) in a csv file. Besides saving model in checkpoints, it also save the best model evaluated in the validation data.

multi_task_model.py :
(It calls functions from `seq_labeling.py and` generate_encoder_output.py)
Use basic LSTM cell to build the model. The three tasks are all separate in different functions. You can also see loss calculating, SGD update operation.

generate_encoder_output.py
The RNN cell receives word input , last hidden state , last prediction sample…to produce new hidden state.

seq_labeling.py
Implement the DNN (multilayer perceptrons on top of LSTM). Calculate the loss of tagging, intent classification.

Experiment

SlotFilling	Model_name	intent	Model_name
F1		acc	
0.791	step600	0.791	step600
0.948	step20000_nolocal	0.972	step20000_nolocal
0.946	step20000_local	0.962	step20000_local
0.948	step20000_local_BEST	0.972	step20000_local_BEST

First try is to increase the training steps. We can observe remarkable increase from training step only 600 to 20000.
Then the second try is to add local context in the model, but there is no obvious improvement.
The third try is inference with the model with the best validation F1 score (around 16000 training step), the intent better but slot filling is not obviously better.

Reference

A dagger by any other name: scheduled sampling
http://nlpers.blogspot.tw/2016/03/a-dagger-by-any-other-name-scheduled.html
Joint Online Spoken Language Understanding and Language Modeling with Recurrent Neural Networks
https://arxiv.org/abs/1609.01462