NLP Final Assignment - Explanation/Summary of Model
Leigh Yeh

Hello Everyone!

So, this was a super fun and challenging assignment for me, in many ways. As I'm sure many of you realized throughout this process, there's a LOT of things to think about when it comes to designing your own model. There are a lot of hyperparameters to tune, a lot of variables and functions to think about, and a lot of different ways to optimize and improve accuracy. It took me *weeks* (and a **ton** of trial and error) for me to settle on this particular model.

I used a BiLSTM architecture, with sparse softmax cross entropy (with logits) for my loss measurement, and AdamOptimizer for my optimizer. I knew from the getgo that I wanted to use an LSTM or BiLSTM model, because of the accuracy that both have with sequential/text data. I found multiple research papers on BiLSTM's and sequence tagging, (https://arxiv.org/abs/1508.01991, https://arxiv.org/pdf/1607.01133.pdf, and more!), which prompted me to choose this particular structure. Although the model/features used in both papers are different than mine, it made sense to me to use a structure that takes into consideration surrounding linguistic context to classify a unit in a sentence, so I went with it!

The final set of parameters I used and submitted had a learning rate of 0.01, a hidden size of 86, an embedding size of 80, a batch size of 32, and tf.zeros for my embedding variable. I know that the hidden size is a super random number; I originally chose 85, then played with it a bit more and just accepted the score that came with the hidden size of 86. I found that this combination of hidden size and embedding size worked really well for both the Italian and Japanese datasets. If I made the embedding size too big, the Japanese set wouldn't train well, and if I made it too small, Italian wouldn't train well. If I made the hidden size too large, then it would take too long for both to train. I also chose a larger learning rate ("larger" in my opinion, I don't think I would typically choose anything bigger than 0.001) because of the time constraints.

This took a really long time for me to figure out and learn, but I feel like I understand so much more about neural networks/architectures now! I want to say a quick, massive thank you to Sami for being one of the best TA's I've ever had! It's incredible how much time he's put into his students and writing up these projects/grading scripts, and it's so rare to find a TA who is willing to spend the time to encourage and motivate students to truly learn the material. Anyway, this was a really long summary, but I had a ton of fun in this class! Hope everyone has a fabulous summer!

Accuracy:

- Italian: 95.2%
- Japanese: 95%
- Secret language: > 90th

Final list of hyperparameters:

- Learning rate: 0.01
- Hidden (state) size: 86
- Embedding size: 80
- Batch size: 32

Final model:

- BiLSTM (with a “forward” and “backward” LSTMCell)
- Loss measurement: sparse_softmax_cross_entropy_with_logits (with tf.reduce_mean)
- Optimizer: AdamOptimizer
- As stated above, I actually didn’t use tf.get_embedding, I used tf.Variable(tf.zeros(...)) for my embedding matrix/placeholder, then I did tf.nn.embedding_lookup

Things I tried:

- I originally just used the SimpleRNNCell implementation, which didn’t give me a high enough accuracy on both languages
- I also tried Keras’ BiLSTM and CRF implementation, but couldn’t get it to run fast enough (but when I ran it with no time limits, it gave really good accuracies!)
- I tried using different features (3 words before/after, 2 words before/after, etc.) but it was hard for me to fit it in the structure of the assignment so I scrapped that