NLU course projects

Amir Gheser (247173)

University of Trento

amir.gheser@studenti.unitn.it

1. Introduction (approx. 100 words)

To improve the baseline of the IAS model on the ATIS dataset the LSTM layer was replaced with a bidirectional LSTM. To account for both directions the last two hidden layers were concatenated before passing them to the linear output layers. Consequently, dropout was integrated which is applied right before the concatenation of the last two hidden layers, opting for a standard dropout probability value of 0.5. In the second part of the assignment, a BERT-base model was fine tuned on the same ATIS dataset. Like the LSTM model evaluation was performed on the aforementioned dataset confronting accuracy and F1-score using the provided conll library [**?**].

2. Implementation details (max approx. 200-300 words)

For Part 1, the implementation of the data loader and instantiation of Lang class were given. The dropout is applied before the linear layer and, as previously stated, to implement bidirectionality the last two hidden layers were concatenated. To train the model I used the Adam optimizer with a learning rate of 1e-4 with batch size 64 over 100 epochs with early stopping a patience 3. The trained network has hidden and embedding size of 200 and 300, respectively.
For Part 2, since BERT uses byte-pair encoding the previous index-based encoding is no longer valid. Hence, to address the tokenization issue I tokenized utterances word by word as done by Chen et al. [1]. This involves creating three tensors to be passed to the BERT model: 'input_ids', the 'attention_mask' and lastly, 'token_type_ids', despite not being required in this specific task. By tokenizing word by word I avoided code complications but memory isn't well exploited causing a slower performances during the embedding phase. Conversely, to preprocess slot and intent labels I followed the same procedure in part 1.

To adapt the architecture to this task I added two linear layers with dropout as done by Chen et al. [1]. The first linear layer for slots takes the last hidden layer of BERT as input, whereas the second linear layer for intent classification requires the pooled output of the BERT network. I opted for AdamW,as it is optimizer of choice in Huggingface BERT fine tuning example, and tested multiple schedulers with varying starting learning rates. Specifically:

- Reduce Learning Rate on Plateau
- Linear Schedule with Warm-up
- No Scheduler

In post processing, to properly evaluate the model I extracted indices of relevant values by masking all special tokens ([PAD] tokens, [CLS] tokens and [SEP] tokens).

3. Results

According to the graphs, choosing a larger model yields similar results but is more prone to overfitting. It would be interesting to experiment higher learning rates and larger models whilst implementing additional regularization techniques.

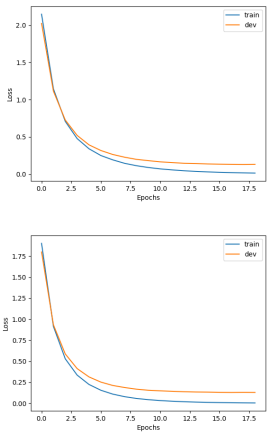

Figure 1: *Plots of loss on small (hid=250, emb=300) and large model (hid=400, emb=500)*

	Slot F1	Intent Acc.
Small		
BiDir	0.937±0.002	0.946±0.002
BiDir+Drop	0.94±0.003	0.944±0.003
Large		
BiDir	0.937±0.002	0.944±0.002
BiDir+Drop	0.942±0.001	0.944±0.001

In part 2, the go-to choice for the optimizer was AdamW and I tested a range of different learning rates and schedulers as can be seen from the Slot-Intent F1 table. F1-scores of intent and slot classification in task2 with different scheduler and varying learning rates. As can be seen from the chart higher learning rates tend to higher rates of overfitting whilst achieving similar results. Also, the choice of scheduler does not seem to influence training.

	Slot-Intent F1			
Scheduler	1e-5	2e-5	3e-5	4e-5
ReduceLR on Plateau	0.93-0.99	0.95-0.99	0.94-0.98	0.96-0.99
Linear with warm-up	0.93-0.98	0.95-0.99	0.95-0.99	0.95-0.99
None	0.94-0.97	0.95-0.99	0.95-0.99	0.95-0.98

4. References

[1] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

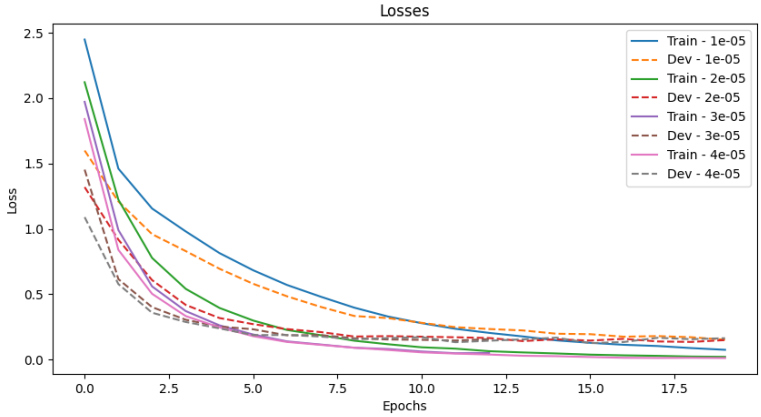

Figure 2: *Reducing Learning Rate on Plateau*

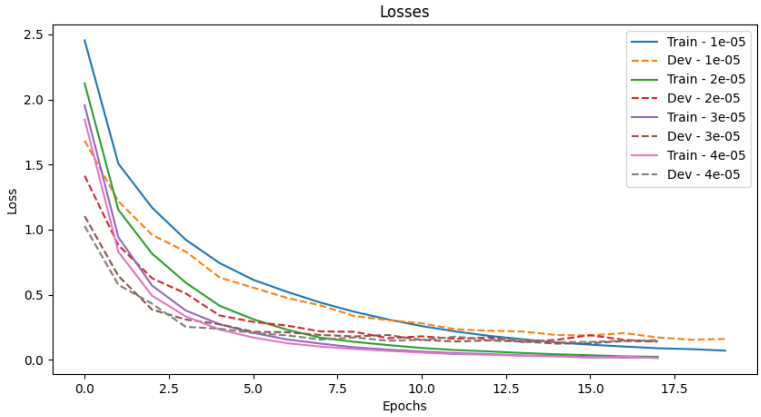

Figure 3: *Linear Schedule with Warm-up*

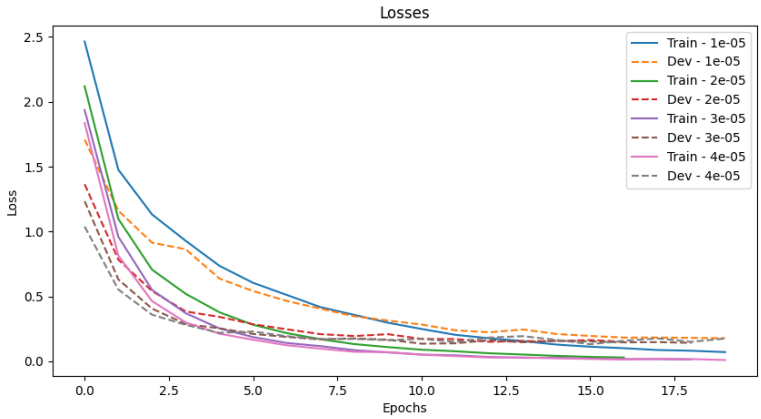

Figure 4: *No schedule*