**Sentence-level Sentiment Classification with RNN**

**Introduction**

In this homework, we perform sentence-level sentiment classification problem on SST, with RNN models.

**Results Summary**

*Table1: Model details*

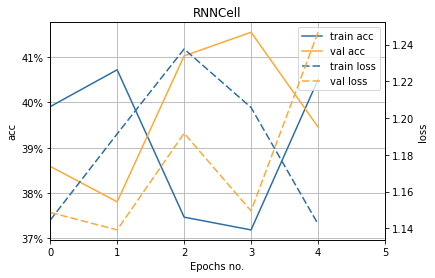
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Models | RNNCell | GRUCell | LSTMCell | GRUCell | GRUCell |
| Edmedding dim | 300 | 300 | 300 | **100** | 300 |
| Initialization of unknown | Normal | Normal | Normal | Normal | **None** |
| Epochs | 5 epochs | 5 epochs | 5 epochs | 5 epochs | 5 epochs |
| Hidden dim | 512 | 512 | 512 | 512 | 512 |
| Batch size | 64 | 64 | 64 | 64 | 64 |
| Test loss | 1.1710 | 1.5229 | 1.1096 | 0.9522 | 0.8881 |
| Test acc | 39.12% | 60.48 % | 41.17 % | 61.60% | 62.22% |

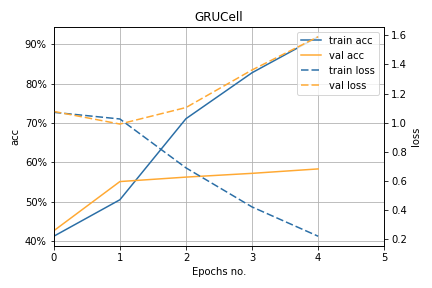
**Comments and Discussion**

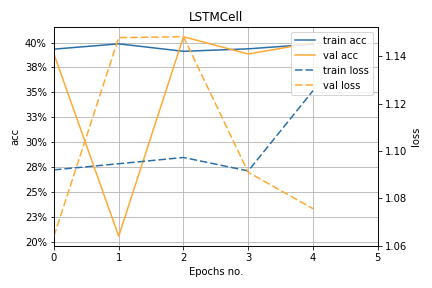
RNNCell vs. GRUCell vs. LSTMCell:

Out of the 3 RNN cells, **GRUCell** performs the best in training, validation and test datasets. It is used as benchmark for subsequent comparisons.

GRU is the most susceptible to overfitting. LSTM is in theory meant to deal with vanishing gradient, but performance seems unstable in the first couple of epochs.



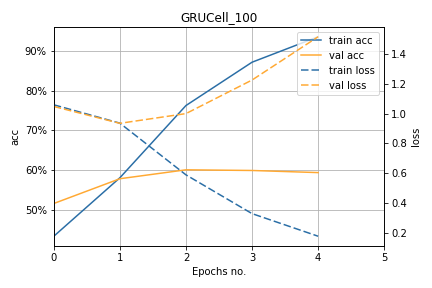




Embedding dim:

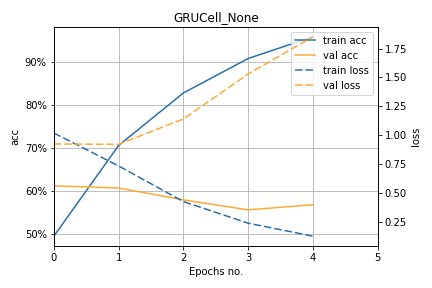
Decreasing embedding dim improves performance marginally. This means that 100 dim vectors describe the words better for this task.

However, it must be noted that we use embedding as fixed input. If we set embedding to be trainable, result could be different.



Normal initialization vs. None:

Not using normal initialization improves model performance marginally too.



**Reference:**

\* [Deep Learning](http://www.deeplearningbook.org/)