ID- 112688862
Aditya Choudhary

NLP Hw-2 Report

1. Model Implementation

a. DAN

- First I initialized model architecture in __init__ method. ‘Num_layers’ of Dense layers with Relu activation were added along with a final Dense layer with softmax activation function.
- In the ‘call’ method, I first iterated over the batch_size to pre-process the data. Masking is done for both training and evaluation mode, **special cases** when all words in a sentence are masked are also handled. In this case I am using the whole sentence without removing any word.
- Dropout is applied only for training mode. Since it was given in paper to use bernoulli distribution, I generated a probability vector from a uniform distribution U (0,1) of size equal to remaining words after masking.Select the word If the probability corresponding to each word is greater than ‘dropout’ given in __init__.
- This processed input is then passed through four Relu layers, each layer’s representation was saved in a tensor to for ‘layer_representations’ variable.
- Final layer’s representation is stored in ‘combined_vector’.

b. GRU

- ‘Num_layers’ of GRUrs with return_sequences=True were initialized.
- Sequence_mask is passed to first layer only, rest layers do not require masking.
- Each layer’s last cell representation is stored in ‘layer_representations’
- Final layer’s last cell representation is stored in ‘combined_vector’
- In keras weights initializer is by default ‘glorot_uniform’, therefore no need to specifically do it.

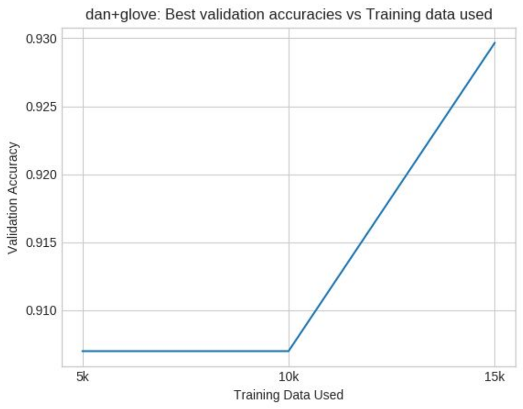
c. Probing

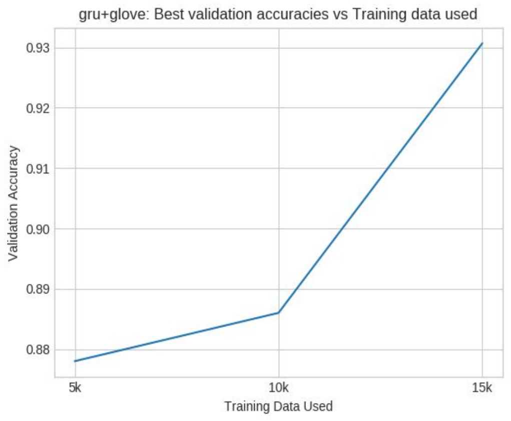
- Initialized a linear classification layer of size ‘classes_num’
- Got outputs from pretrained_model with setting training as False.
- Passed ‘nth’ layer representation to classification layer and returned the ‘logits’

3. Analysis

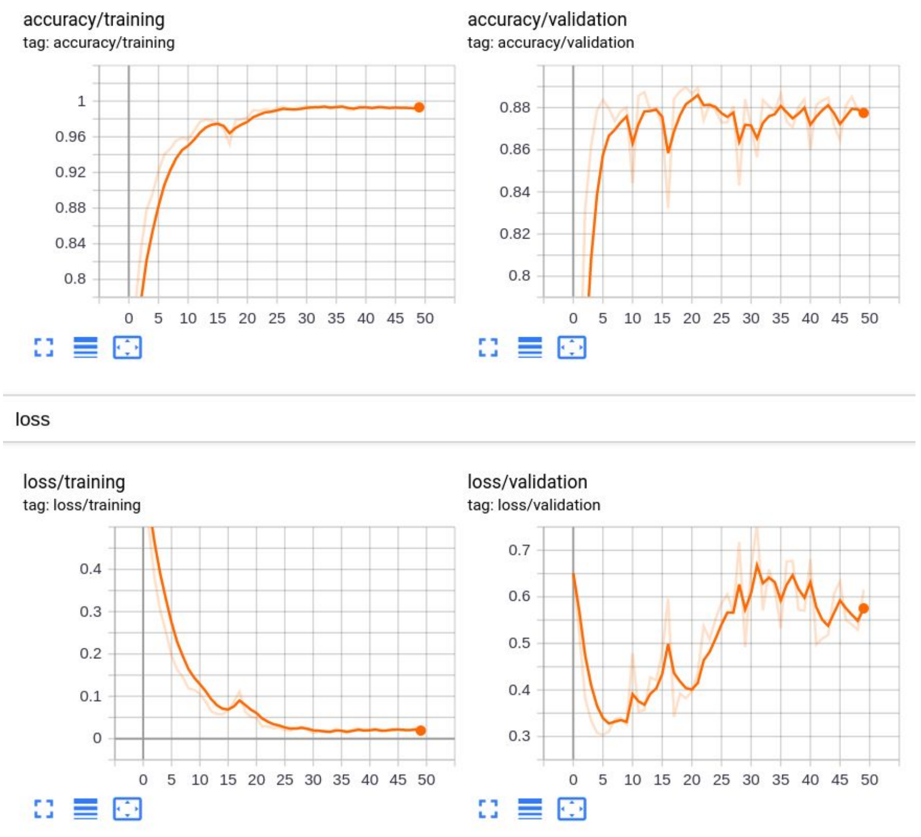
3.1 Learning Curves

Increasing the training data





For both models when increasing training data, the training time increased. For DAN Validation accuracy remained almost same for 10k size but increased significantly (0.929) for 15k size. For GRU there is a notable improvement for 10k size and significant increase to 0.931 for 15k size.



Increase training time (number of epochs)

This is an interesting graph, On increasing the no. of epochs we observed that training loss decreased continuously and reached almost zero levels, but the corresponding Validation loss first decreased till a level and then kept on increasing. This shows a typical case of Model overfitting.

We can also see that the 'validation' accuracy has some spikes but did not increase with epoch numbers although training accuracy kept on improving in fractions. According to the graphs the ideal epoch number is around 8-9.

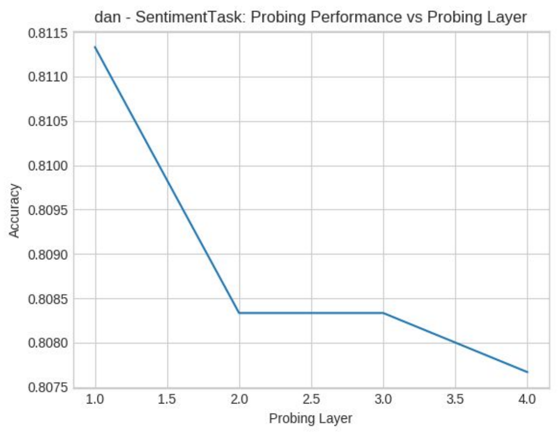
3.2 Error Analysis:

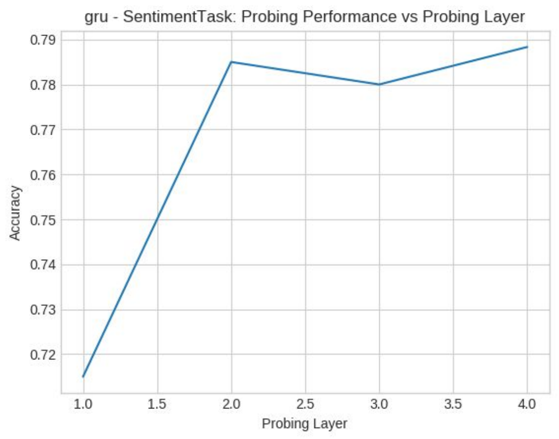
one advantage of DAN over GRU - positive and negative word sentiment differences gets better represented in DAN. Also training time of DAN is much less than GRU.
From our Perturbation task, we can see that DAN has represented the differences between positive and negative words better, if we see Layer#3 and #4 from the plots we can clearly see that difference amplify in DAN better than GRU.

one advantage of GRU over DAN - DAN ignores word order in sentences which GRU incorporates well. From our Bi-gram training task, when we reversed the word order from 'New York' to 'York New' DAN generated the same representations(Since the averaging from individual word vectors give same values) and hence it's accuracy is 0.5. But GRU differentiated between the two, since it processes in sequential manner and has a better accuracy ~0.63. model used - main_dan_5k_with_emb and main_gru_5k_with_emb

4. Probing Task

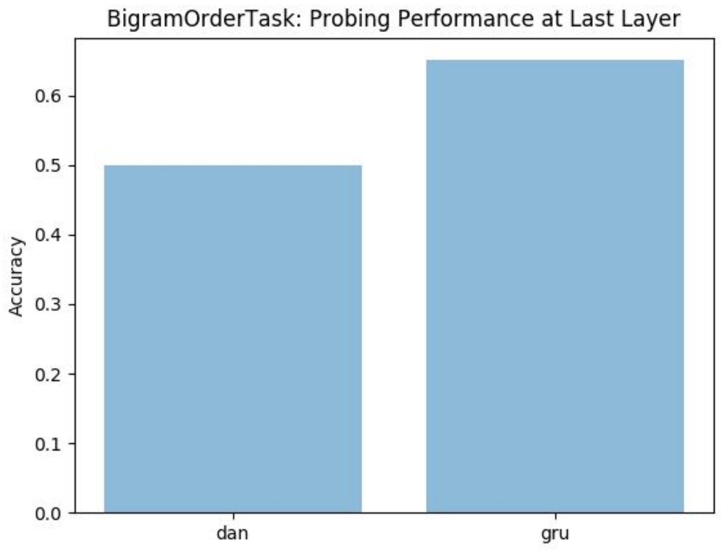
4.1 Probing Sentence Representation for Sentiment Task





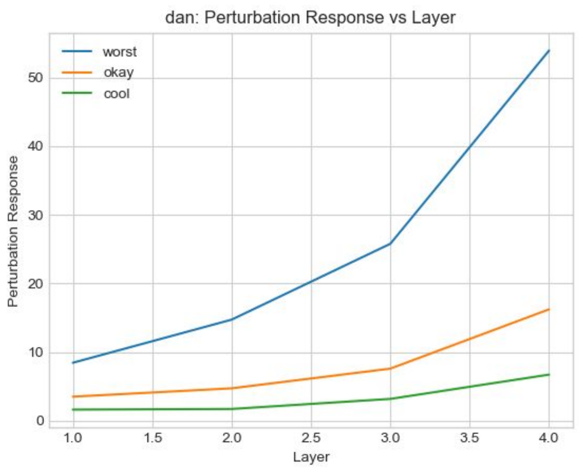
In GRU we observe that accuracy increment from 1st layer to 2nd layer is significant (~0.07), and there is a general trend of increment till 4th layer, stating that the optimal layer maybe 4th or yet to come.
In DAN we observe accuracy has decreased from 1st layer to corresponding layers, but the drop is very low , around 0.03 . Although we expect the accuracy in 2nd or 3rd layer to be greater than 1st layer, this kind of cases can occur sometimes because randomness of weights initializer in neural networks and when evaluation data distribution is different from training data.

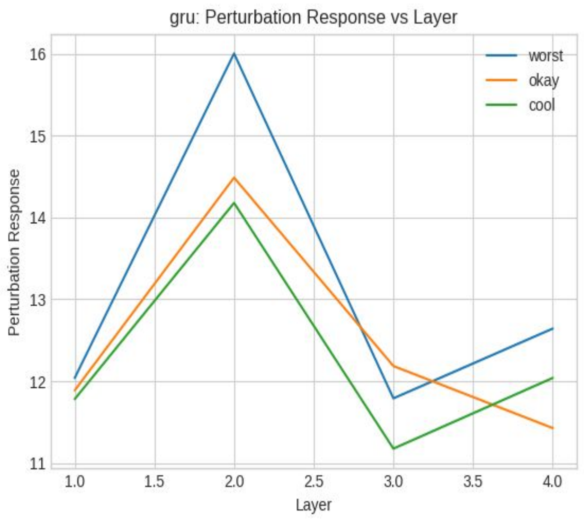
4.2 Probing Sentence Representations for Bigram Order Task



Observations - GRU performs better that DAN for these kind of datasets. Since DAN ignores word orders and average them to form sentence representations, it loses out to incorporate meaning changes on reversing words. Whereas GRU processing sentences sequentially , can retain the order difference in words. DAN has accuracy of 0.5 which is what we expected , GRU has performed at ~0.63 accuracy.

4.3 Analysing Perturbation Response of Representations





In DAN initially the differences between ‘worst’ and ‘cool’ was less but as we go deeper in DAN the difference between the positive and negative sentence gets amplified. This is according to our expectations of the network.
In GRU we notice that the difference b/w ‘worst’ and ‘cool’ increases slightly in Layer#2 and then decreases slightly in Layer#3 but it is almost the same across the layers, which is also expected from GRU.