DS-GA 1011 Natural Language Processing HW 1

Muyang Jin mj1477; Yixuan Wang yw1708; Meiyi He mh5275; Jiayu Qiu jq429

3.1 Training on SNLI

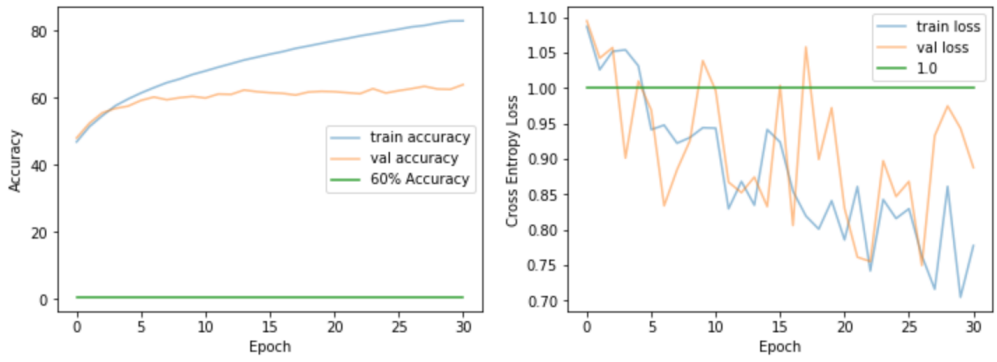
3.1.1 Validation Result

Below is the hyper-parameter tuning result for both model. We've included the plot for the best model and the rest of the graphs can be found in appendix.

- **Logistic Regression**

The best performance (validation accuracy = 64.0) is achieved using parameters max_vocab = 8000, emb_dim = 100, interaction = product, learning_rate = 0.001, max_sentence_length = 100, batch_size = 32.

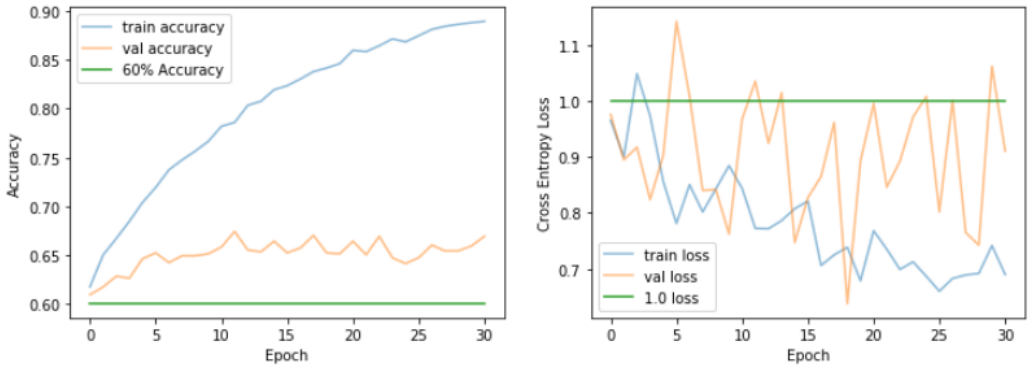
	max_vocab	emb_dim	interaction	training accuracy	val accuracy	training loss	val loss
0	8000	80	concat	70.078	62.1	0.844	0.968
1	10000	80	concat	70.925	61.7	0.833	0.714
2	12000	80	concat	71.783	61.8	0.878	0.999
3	10000	60	concat	70.466	61.5	0.854	0.838
4	10000	80	concat	71.151	59.7	0.937	1.032
5	10000	100	concat	71.538	62.6	0.831	0.876
6	10000	80	concat	71.126	60.7	0.888	1.038
7	10000	80	product	81.975	62.6	0.782	0.891
8	8000	100	product	83.023	64.0	0.777	0.887



- **Neural Network**

The best performance (validation accuracy = 66.9) is achieved using parameters max_vocab = 10000, emb_dim = 80, interaction = product, learning_rate = 0.01, max_sentence_length = 100, batch_size = 32.

	max_vocab	emb_dim	interaction	training accuracy	val accuracy	training loss	val loss
0	8000	80	concat	87.619	63.9	0.733	0.825
1	10000	80	concat	89.130	65.8	0.685	0.871
2	12000	80	concat	89.537	65.1	0.678	0.907
3	10000	60	concat	89.351	65.4	0.713	0.864
4	10000	100	concat	90.023	65.5	0.667	0.845
5	10000	80	concat	88.969	66.9	0.690	0.911
6	10000	80	product	95.922	63.0	0.582	1.142
7	8000	80	product	95.755	63.6	0.586	0.969
8	10000	80	product	95.594	63.0	0.597	0.696
9	12000	80	product	96.109	64.8	0.574	0.910



3.1.2 Parameter Implication

We decided to tune on three of the hyper-parameters, we will discuss the effect of each one of them and provide with the best hyper-parameters combination.

- **Size of the vocabulary** :

  Larger size of vocabulary tends to provide context, in a sense that less words will be mapped to unknown and the size of the embedding matrix will be larger, which will improve the complexity of the model. Refer to the first three rows of both tables we can see that although the thresholds are different between logistic regression and neural network, both models are getting higher training accuracy and lower validation accuracy as the size of the vocabulary exceed some thresholds. This means there exists over-fitting. Our best sizes of vocabulary are 8000 for logistic regression and 10000 for neural network.

- **Embedding dimension**:

  Embedding indicates the relevance of a token to the document and larger embedding dimension tends to reveal more implicit message one token is bearing with respect to the document. Our best embedding dimensions are 100 for logistic regression and 80 for neural network.

- **Two ways of interacting the two encoded sentences**:

  - **Concatenation**
  - **Element-wise multiplication**

  From our validation result, we found that for both types of models, Element-wise multiplication(PRODUCT) has a significant higher training accuracy than Concatenation(CONCAT). Our interpretation for this phenomenon is that multiplication can capture more interaction between the two sentences than concatenation. Using multiplication, if two sentences have the same sign on i_{th} row of the vector embedding , their product will be positive and negative if the signs are opposite. Also the scale of the term might also be affected. If we want the same interaction to be captured using concatenation, it will need the model to identity: i_{th} and the $(i + emb\_dim)_{th}$ terms of the concatenated vector are both greater than 0. Neural network naturally can do a better job of finding these type of interaction than logistic regression so we see that NN has a higher training accuracy than LR when using

concatenation as the interaction method and the increases in training accuracy when switching from concatenation to multiplication are higher for LR than that of NN.

3.1.3 Example Explanation

- Correct prediction:

Correct example #1:
Premise: A man in a blue helmet jumping off of a hill on a dirt bike.
Hypothesis: The man is a professional athlete.
Label: Neutral

Correct example #2:
Premise: A lady sitting on a bench that is against a building and under a poster of a man in a uniform waving.
Hypothesis: Nobody is sitting.
Label: Contradiction

Correct example #3:
Premise: A woman in a blue shirt talking to a baby.
Hypothesis: A young woman talks to her baby at the park.
Label: Neutral

- Incorrect prediction:

Incorrect example #1:
Premise: Various people shop at a street clothing sale.
Hypothesis: Various cats shop at a street clothing sale.
Label: Contradiction
Prediction: Entailment

Incorrect example #2:
Premise: Football players playing in mud.
Hypothesis: Football players are playing chess.
Label: Contradiction
Prediction: Entailment

Incorrect example #3:
Premise: Three women in festive dresses are dancing in a large city street while a large crowd is behind them holding up a banner.
Hypothesis: Three women dance in front of a large crowd.
Label: Entailment
Prediction: Neutral

Example #1 and #2 follow the same logic where only 1 or 2 words are different in both sentences, therefore rendering the prediction both as entailment where the labels are both contradiction. Since this incorrect sample is generated using logistic regression model with element-wise product method, it's possible that when the ratio of same words to different words bigger than 1 and there's no clear negation word, the embedding might imply high sentence similarity which lead to incorrect prediction as entailment. For example #3, the length of two sentences is of large difference in which the first sentence contains lots of details information that the second sentence do not have, thus these pair might be deemed low relevance by model and be predicted to neutral while the correct label should be entailment.

3.2 Evaluating on MultiNLI

In this section, we tokenized each genre of the MultiNLI data, then took the best SNLI model to get the validation accuracy for each genre, the result can be found on next page in the left table.

Within MultiNLI, validation accuracy of telephone genre is a bit higher for both model. It could be explained by the fact that words used over phone are more colloquial therefore bearing a better match with corpus in SNLI, while it's not the case for other genres where language used tends to be more formal or expert that requires certain industry knowledge.

Compared to SNLI, the performance of MultiNLI is significantly lower only achieving validation accuracy around .35. It makes sense as the model is trained on SNLI corpus and directly put into use without training process on this specific corpus.

3.3 Fine-tuning on MultiNLI

Since the Neural Net Classifier is the best performing model in section 3.1, we decided to fine-tune the model for each of the genre, using smaller learning rate of 0.001, 0.003, 0.005 and kept the number of epoch to 10. We picked the best validation accuracy for each genre and compared to their performance before fine-tuning, the result is recorded next page in the right table.

	genre	lr val acc	nn val acc
0	fiction	34.371859	35.075377
1	slate	34.431138	31.437126
2	government	36.811024	33.070866
3	telephone	36.915423	35.621891
4	travel	35.437882	33.706721

	Genre	Fine-tune	Val accuracy
0	fiction	Before	35.075
1	fiction	After	36.080
2	slate	Before	31.437
3	slate	After	37.525
4	government	Before	33.071
5	government	After	37.500
6	telephone	Before	35.621
7	telephone	After	37.711
8	travel	Before	33.707
9	travel	After	35.438

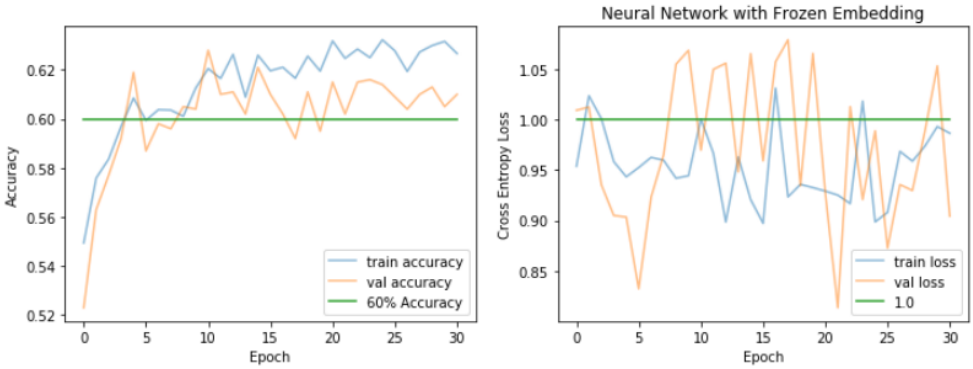
After the fine-tuning, the validation accuracy in all genres have been improved.

3.4 Pre-Trained Word Embedding

For this task, we downloaded the preembedding "wiki-news-300d-1M.vec"
Repeat 3.1:
The validation result shows that the performance has gone down significantly after applying frozen pre-trained embedding.

One reason could be that the relationship between words are different between our data sets and the text source where the embedding come from - Wikipedia. This is a reasonable guess since our texts, based on our observation, are simpler, informal and story-telling. The Wikipedia texts, on the other hand, are more formal.

Because the embedding dimension and the size of vocabulary are much higher than that of the original embedding we trained, one could argue that the drop of the accuracy is due to over-fitting, but because the training accuracy also dropped, we believe the main reason is the one above.

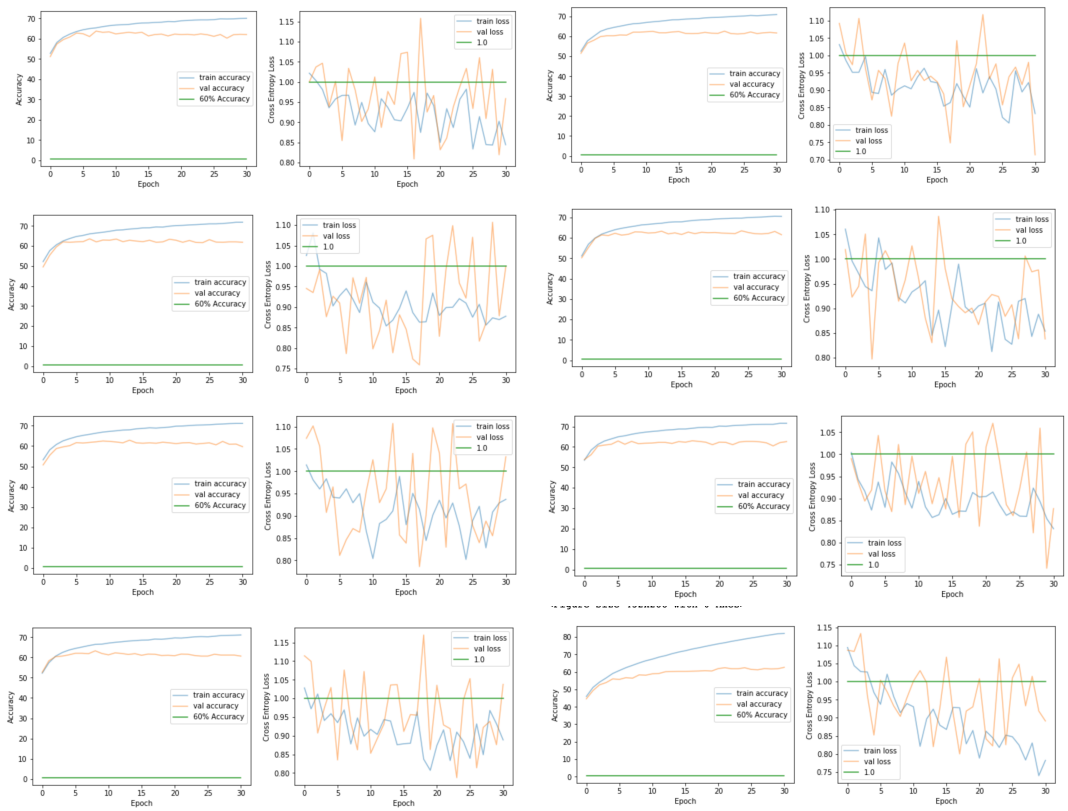


Repeat 3.2:
There is no significant increase or decrease in term of the validation accuracy using frozen pre-trained embedding. The result is on the next page.

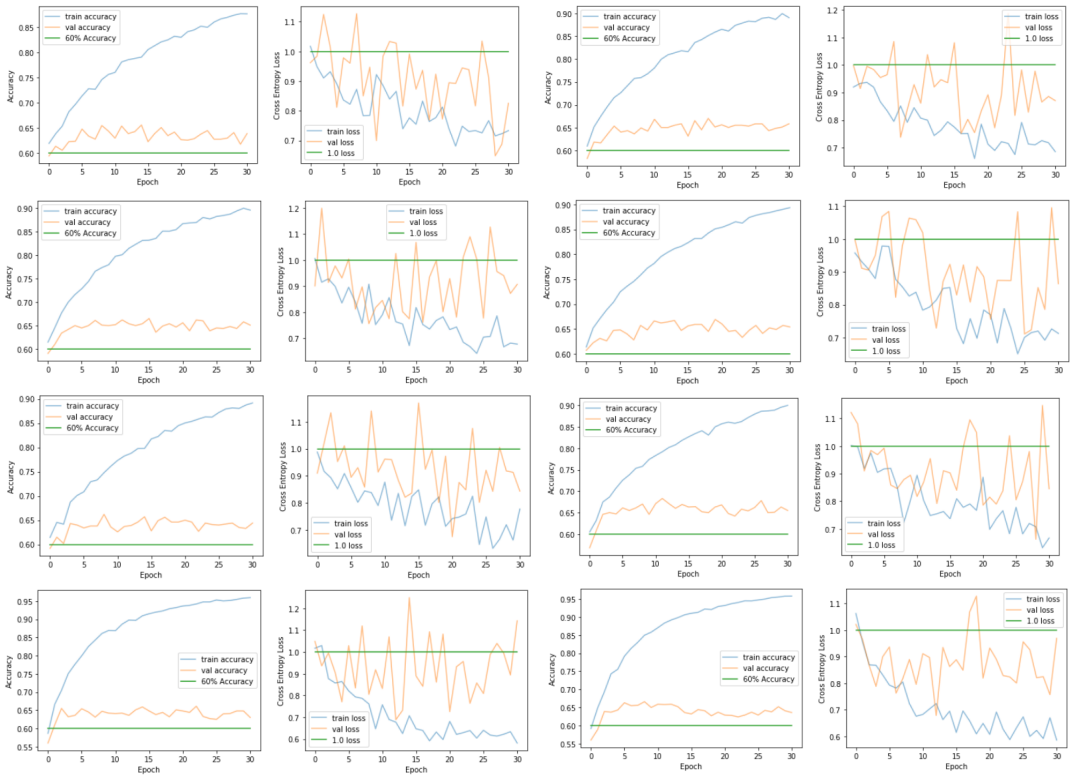
	Genre	Trained From	Val accuracy
0	fiction	Pre-trained	34.573
1	fiction	SNLI	35.075
2	slate	Pre-trained	33.633
3	slate	SNLI	31.437
4	government	Pre-trained	31.594
5	government	SNLI	33.071
6	telephone	Pre-trained	32.338
7	telephone	SNLI	35.622
8	travel	Pre-trained	31.161
9	travel	SNLI	33.707

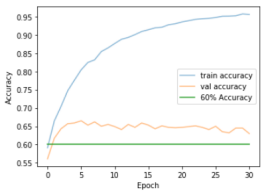
4 Appendix

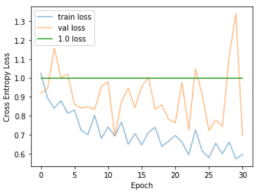
4.1 Appendix A - Logistic Regression Validation Result

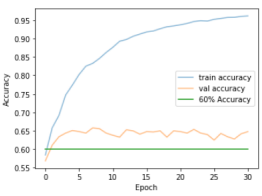


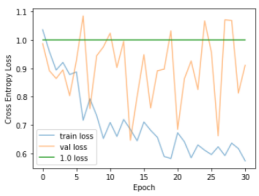
4.2 Appendix B - Neural Network Validation Result




Accuracy
Epoch
train accuracy
val accuracy
60% Accuracy


Cross Entropy Loss
Epoch
train loss
val loss
1.0 loss


Accuracy
Epoch
train accuracy
val accuracy
60% Accuracy


Cross Entropy Loss
Epoch
train loss
val loss
1.0 loss