Natural Language Processing - Assignment 2

Weicheng Zhu, wz727

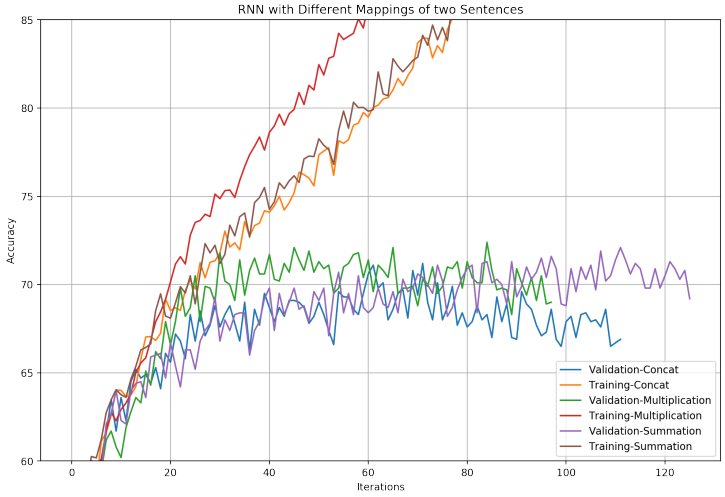
October 31, 2018

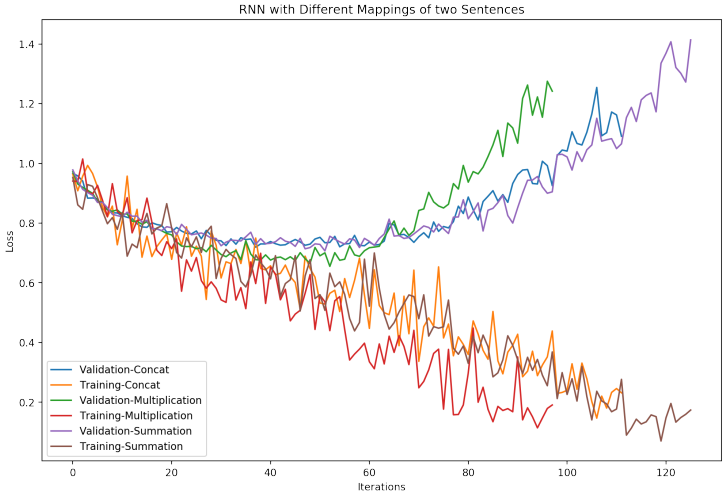
Source code repository: https://github.com/jackzhu727/NLP-1011/blob/master/HW2.ipynb

1 Train on SNLI

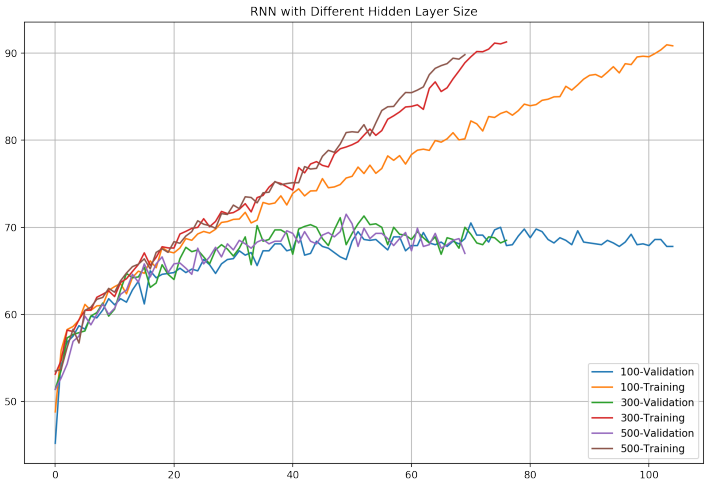
RNN Encoder I built RNN encoder with bidirectional GRU. I trained the model with Adam optimizer with learning rate 0.001. To avoid overfitting and training unnecessary additional epochs, I set a threshold of difference between training accuracy and validation accuracy for early stopping at 25. Because my best validation accuracy are above 70 and when the difference reaches 25, the training accuracy wll be close to 100. In this case, the model will overfit. With the early stopping, you may observed that curves in figures do not have same iterations.

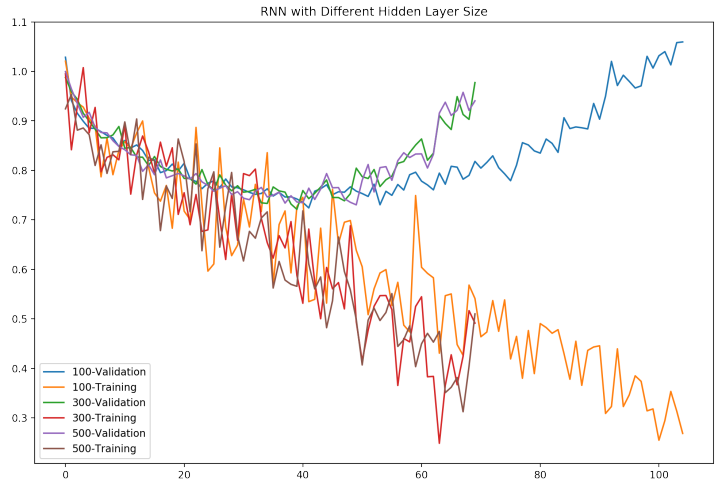
Interacting two encoded sentences I experimented with different mapping schemes: multiplying, summing or concatenating two encoded sentences. The number of trained parameters in the model of multiplication and summation is 2348403, because they have same hidden size; the number of trained parameters of concatenation is 2889603. From figures blow, we learnt that summation and multiplication have higher accuracy than concatenation, and multiplication has faster convergence. So I choose elementwise multiplication to interact between two encoded sentences.



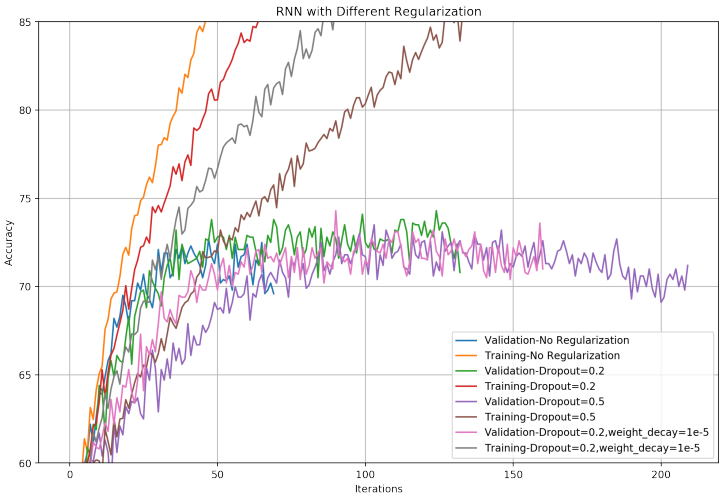


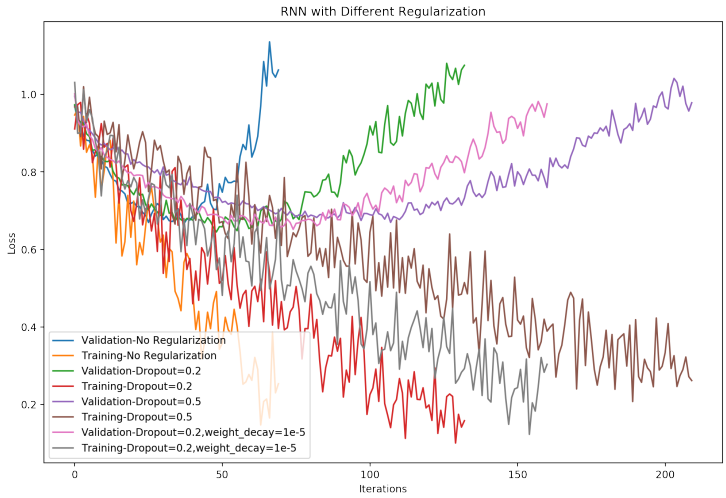
Hidden size I experimented with different sizes for hidden layer of GRU 100, 300 and 500. The numbers of trained parameters are 563203, 2348403, and 6816003. The accuacy and loss in these two plots for 300 and 500 are better than 100 hidden size. The plot also showed that the training and validation curve of 300 and 500 greatly overlapped, but cost of training 500 hidden size is much greater since the number of parameters to be trained is exponentially greater than 300. Since we cannot significantly improve performance of model adding 300 hidden size to 500 hidden size, I prefer to choose 300 as size of hidden layer.





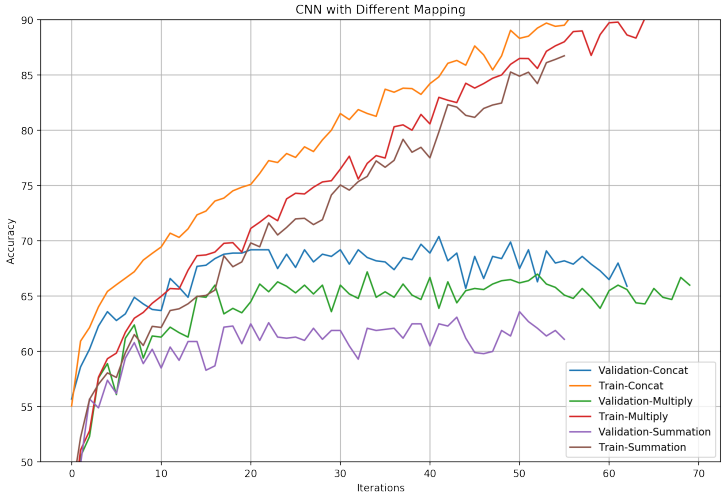
Regularization I experimented with different ways of regulation including dropouts of 0.2 and 0.5 and weight decay of 1e-5 at optimizer. For tuning regularization, the number of trained parameters is fixed at 2348403. The dropout layer and weight decay can avoid fast overfitting. The figure tells that model dropout of 0.2 and weight decay of 1e-5 can obtain lowest validation loss and highest accuracy. Also, we cannot dropout too much; otherwise, the model will underfit just like the dropout 0.5 case. So I choose dropout of 0.2 and weight decay of 1e-5.

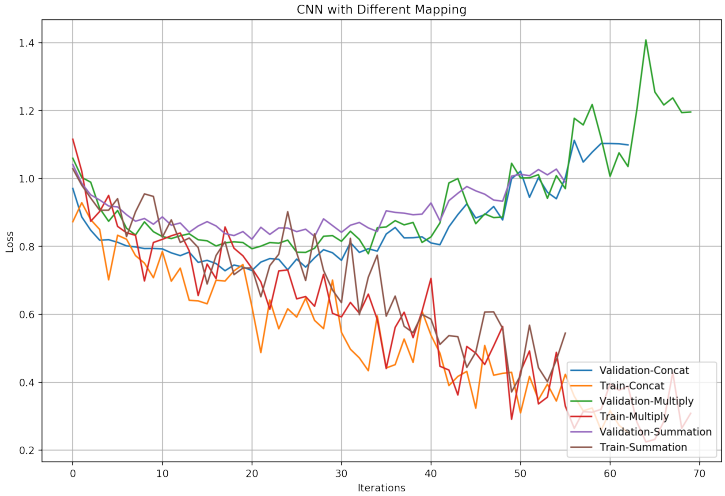




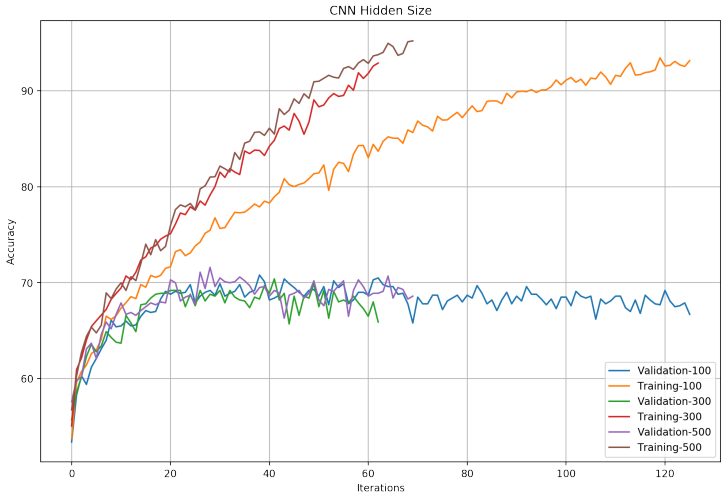
CNN Encoder I encoded two sentences with two convolution layer. Same to training RNN models, I choose Adam optimizer with learning rate of 0.001 and early stopping when training accuarcy is 25 more than validation accuracy.

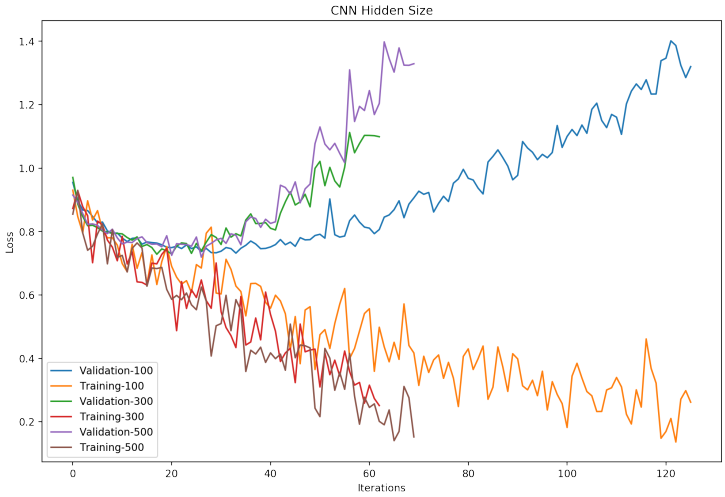
Interacting two encoded sentences Similar to fine tuning in RNN model, I experimented with mapping schemes of multiplication, summation and concatenation. The numbers of trained parameters are 631803, 631803 and 721803, respectively. From figures below, the concatenation of has highest validation accuracy and validation loss. Therefore, unlike RNN, I choose concatenation to map two encoded sentences in CNN.



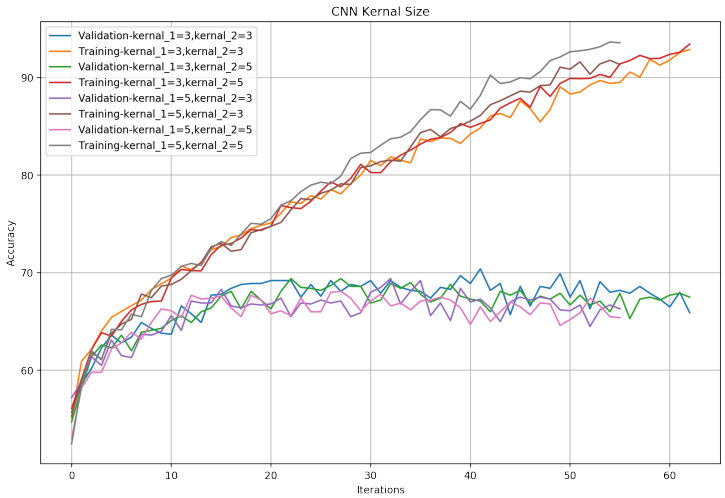


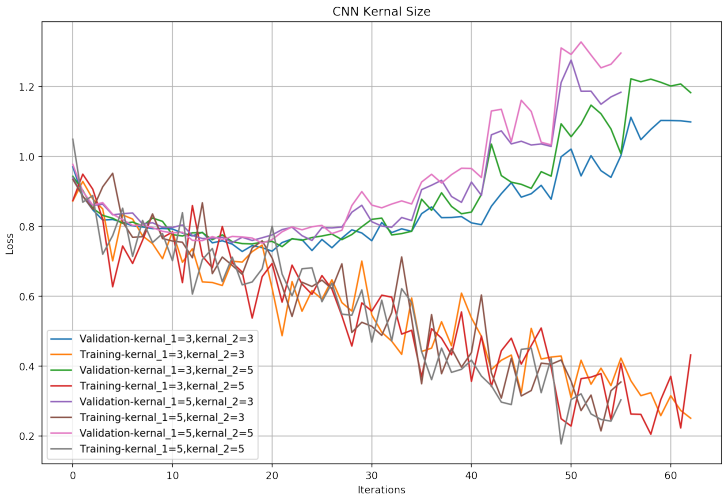
Hidden size I experimented with different sizes for hidden layer of convolution layers:100, 300, 500. The numbers of trained parameters are 140603, 721803 and 1703603, respectively. 500 and 300 hidden size leads to same validation accuracy, but 500 hidden size training loss raises faster. So a more complicated model is more likely to overfit and cost more time to train the model. So I choose hidden size of 300.





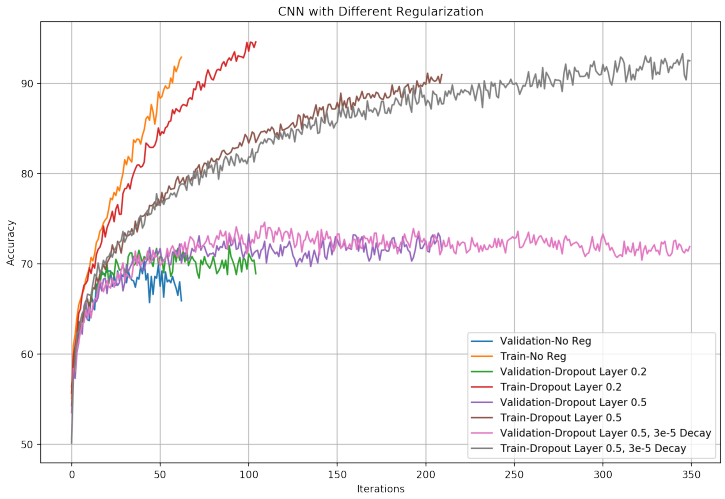
Kernal Size I experimented with different combination of filter kernal sizes in two layers, including (3,3), (3,5), (5,3), (5,5). From loss and accuracy curves, I learnt that the more words we put in the same filter, the faster the training accuracy and validation loss grow. i.e. the model is more likely to overfit with larger kernal sizes. So I choose two layers of convolution filter with size 3.

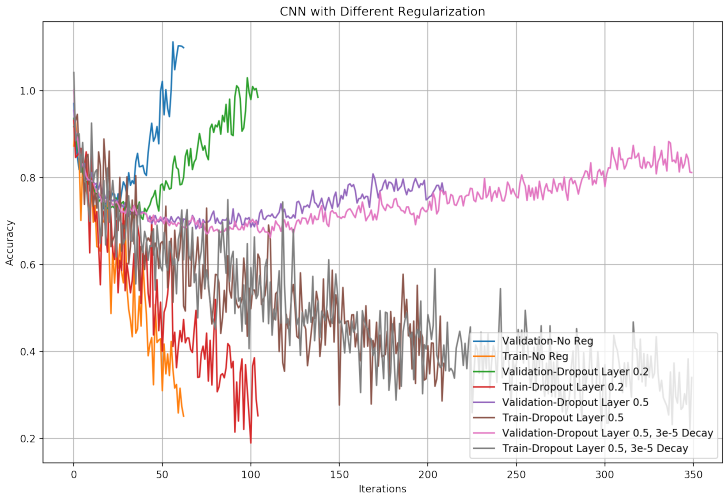




Regularization I experimented with different ways of regulation including dropouts of 0.2 and 0.5 and weight decay of 1e-5 at optimizer. By adding regularization in the model, the train accuracy curve has less

slope, so we trained more epoches before overfitting and obtain a better result. These curves indicate that a dropout layer of 0.5 and weight decay of 1e-5 keep a good balance between underfitting and overfitting and obtained highest validation accuracy. So I choose dropout of 0.5 and weight decay of 1e-5.





Correct Samples :

1. Contradiction:

–Three women on a stage , one wearing red shoes , black pants , and a gray shirt is sitting on a prop , another is sitting on the floor , and the third wearing a black shirt and pants is standing , as a gentleman in the back tunes an instrument .

–There are two women standing on the stage

2. Entailment:

–An Asian man in colorful robes holds a bell and leans toward a small table , with Buddhist or Hindu decorations on the table and the walls .

–An Asian man in colorful light robes holds a bell .

3. Neutral:

–Three cheerful ladies sitting at a table doing a yarn work in a room , at the background are similar groups of ladies doing similar work .

–The ladies are discussing what they are going to do tonight .

Wrong Samples Overall, the hypothesis in wrong samples are short, providing less information.

1. Label: Contradiction, predict: entailment

–A lone , 2-3 year old blond child in a blue jacket is putting a small black plastic item in his mouth as he kneels on a waiting room couch pointed toward the back while looking at something or someone not in the room .

–The couch is pointed toward the front .

Reason: The model probably did not recognize back and front are opposite words, just learnt that all the other words of hypo are contained in premise.

2. Label: Neutral, predict: contradiction

–A group of people dressed in Santa Claus suits are looking towards an audience while a DJ runs a sound board and another person throws green balls into the air .

–A band plays at a beach party .

Reason: The model may think beach party and Santa Claus were two distant stuffs in summer and winter, so the model judged them as contradiction. The dataset labeling it as neutral may consider Australia.

3. Label: Neutral, predict: entailment

–A woman walking a dog on a leash at the beach , trailing behind as a pug follows another unseen woman .

–a large woman walks a dog

Reason: Most of words in hypothesis are included in premise except large, so the model mistook them as entailment.

Conclusion For RNN, I used 300 hidden size and dropout 0.2 with RNN encoder, multiplication to map two sentences and 1e-5 weight decay in Adam to train the model. As a result, I got training accuracy 89.483 and validation accuracy 75.0.
For CNN, I used 300 hidden size, two convolution filter of size 3 and dropout 0.5 with CNN encoder, concatenation to map two sentences and 1e-5 weight decay in Adam to train the model. As a result, I got training accuracy 93.032 and validation accuracy 73.8.

2 Evaluating on MultiNLI

I applied trained CNN and RNN model on the MNLI dataset grouped by genres and calculated the validation accuracy of different genres. From the following accuracy table, I learnt that even though I achieve a validation accuracy at 75 at SNLI dataset, the test accuracy was rather low in totally different genres. This problem indicates that my model trained on SNLI was conditioned on a certain genre and cannot be generalized to other genres. Also, for MNLI, we can see some sentences much longer than SNLI; they might lead to the low accuarcy.
Among these five genres, I observed that accuarcy of fiction, telephone, government and travel are around 50, but slate is around 45. So I think the data distribution of slate might be more diverging from distribution of SNLI than other four genres.

	RNN	CNN
fiction	48.543	49.648
telephone	46.567	51.94
slate	44.711	44.91
government	45.472	49.311
travel	43.381	50.713

3 Fine-tuning on MultiNLI(Bonus)

Transfer learning was applied to train a model more specific to different genres. I loaded pre-trained CNN model in SNLI and implemented a new dataloader with greater maxsentence length for MNLI. Then I continue training based on small MNLI genre dataset for 20 epochs and obtained $5-10\%$ increases in validation accuracy for each genre.

	Before Fine-tuning		After Fine-tuning	
	Train	Val	Train	Val
fiction	48.67	49.648	90.355	55.578
telephone	47.518	51.94	83.841	57.015
slate	44.66	44.91	77.049	49.002
government	48.777	49.311	95.364	59.35
travel	48.407	50.713	99.573	55.397

Then I applied fine tuned models of each genre to all the genres for validation accuracies. From the following table, we can learn that the accuracy on diagonal entries are greatest in either its row or column, so fine-tuning on CNN model can improve the performance of classification in specific genre.

Model \ Val	fiction	telephone	slate	government	travel
fiction	55.578	51.841	47.006	52.56	51.426
telephone	53.768	57.015	44.81	51.28	51.426
slate	53.166	52.239	49.002	53.642	50.102
government	51.357	55.224	46.806	59.35	52.342
travel	51.357	51.343	45.21	53.839	55.397