HW2

xy990

October 2018

1 Hyper-parameters Tuning

1.1 Hidden Size in CNN

For the parameter hidden size in CNN, I choose hidden size is 100, 200 and 300. Hidden size means the number of hidden neurons. Compared with the training accuracy and validation accuracy, the CNN model with hidden size 300 performs better than other two models, but it seems that CNN model with hidden size 300 overfits the data a little because the training accuracy for the model with hidden size 300 is much higher than other models, but the validation accuracy does not outperforms others much. Please see results in figure 1 and 2 .

1.2 Hidden Size in RNN

For the parameter hidden size in RNN, I also choose 100, 200 and 300 as hidden size. Compared with the training accuracy and validation accuracy, the RNN model with hidden size 300 performs better than other two models. Please see results in figure 3 and 4 .

1.3 Drop out in CNN

For the parameter drop out in CNN, I choose 0.0, 0.1, 0.5, 0.8 as drop out. Drop out means every time does not activate all cells in the model to avoid overfitting. Compared with the validation accuracy, the CNN model with drop

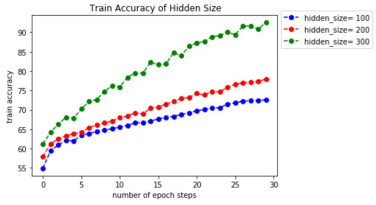

Figure 1: train accuracy of hidden size for cnn

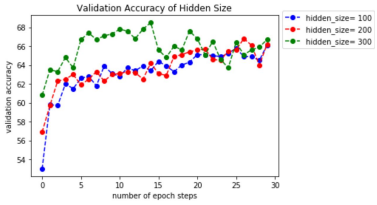

Figure 2: validation accuracy of hidden size for cnn

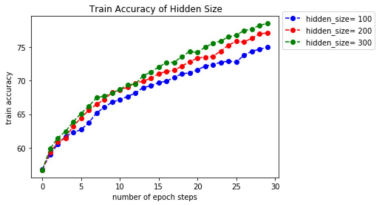

Figure 3: train acc of hidden size for rnn

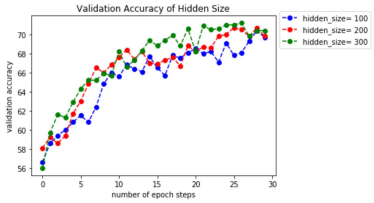

Figure 4: validation accuracy of hidden size for rnn

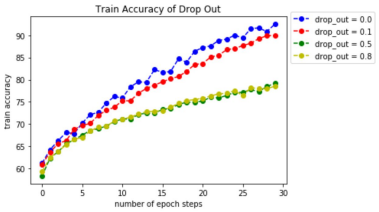

Figure 5: train acc of drop out for cnn

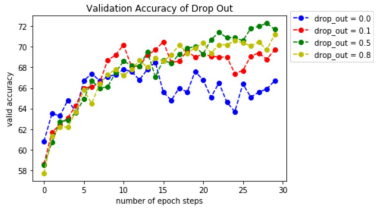

Figure 6: validation accuracy of drop out for cnn

out 0.5 outperforms than other two models. The training accuracy of CNN model without drop out and with drop out 0.1 are much higher than CNN model with drop out 0.5 and 0.8, but the validation accuracy of CNN model without drop out and with drop out 0.1 is much lower than CNN model with drop out 0.5 and 0.8. It shows CNN model without drop out overfits the data. Please see results in figure 5 and 6 .

1.4 Drop Out in RNN

For the parameter drop out in RNN, I choose 0.0, 0.1, 0.5 as drop out. Compared with the train accuracy and validation accuracy, the RNN model with drop out 0.1 and without drop out performs similarly and outperforms the model with drop out 0.5. It probably because the RNN model does not overfit the data, there is no need to use drop out. Please see results in figure 5 and 6 .

1.5 Kernel Size in CNN

For parameter kernel size in CNN model, I choose 2, 3, 5 and 7. Kernel size in CNN is very similar to the function of n-gram. Compared with training accuracy and validation accuracy, CNN model with kernel size 5 and 7 is a little bit overfitting, the training accuracy for CNN model with kernel size 5 and 7 is higher than model with kernel size 3. However, the validation accuracy for CNN model with kernel size 3 is higher than other models. Due to early stopping, model with kernel size 2 early stops after 7 epochs, so I fill out all latter values with the last values. Please see results in figure 9 and 10 .

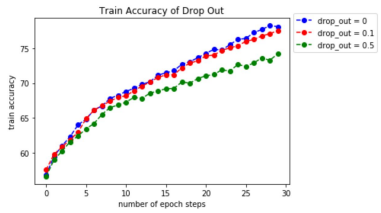

Figure 7: training loss of Learning Rate

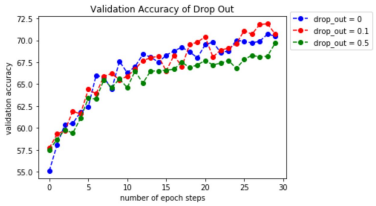

Figure 8: validation accuracy of Learning Rate

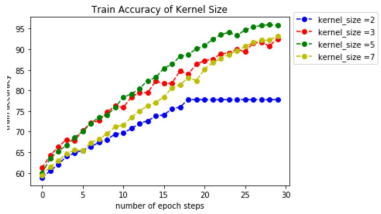

Figure 9: training accuracy of kernel size

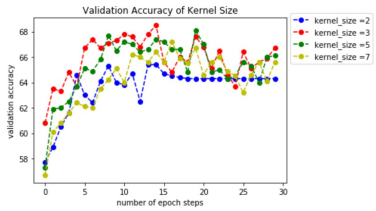

Figure 10: validation accuracy of kernel Size

2 Best Model and Error Analysis

After tuning the parameters, I choose the RNN model without drop out and with hidden size 300, which produces the best validation accuracy 71.8. For the best trained model, I select 3 correct prediction and 3 wrong prediction. please see table 11 and 12. In the first incorrect prediction, Premise: A women wearing a white shirt and green shorts, sitting on a rock in a beautiful body of water. Hypothesis: The woman is barefoot. The label is neutral and the prediction is Contradiction. Why the model misclassifies this probably because in premise there appears 'a person is climbing. . . ' and in hypothesis there appears 'a man is climbing'. The model wrongly thinks the hypothesis entails the premise. In the second incorrect prediction, Premise: A person is climbing a cliff wall that overlooks water. Hypothesis: A man is climbing: The label is neutral and the prediction is entailment. Why the model misclassifies this probably because in premise there appears 'a person is climbing. . . ' and in hypothesis there appears 'a man is climbing'. The model thinks the hypothesis entails the premise. In the third incorrect prediction, Premise: A young adolescent is jumping into a pool. Hypothesis: The child is sacred to jump in the water. The label is contradiction and prediction is neutral. Why the model misclassifies this probably because in premise there appears ' a young adolescent', 'jumping into a pool' and in hypothesis there appears ' the child', ' jump in the water'. The model is not smart to utilize the word 'scared' to know it is a contradiction, but predicts neutral.

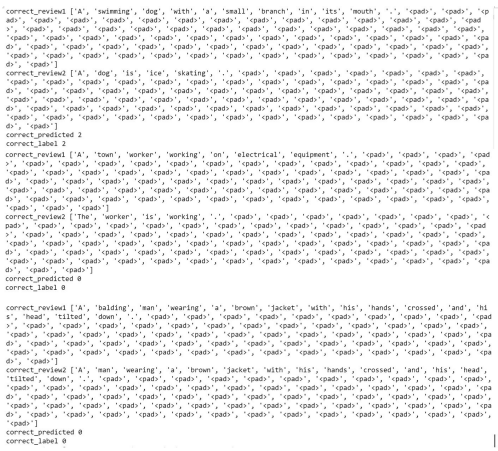

Figure 11: correct prediction

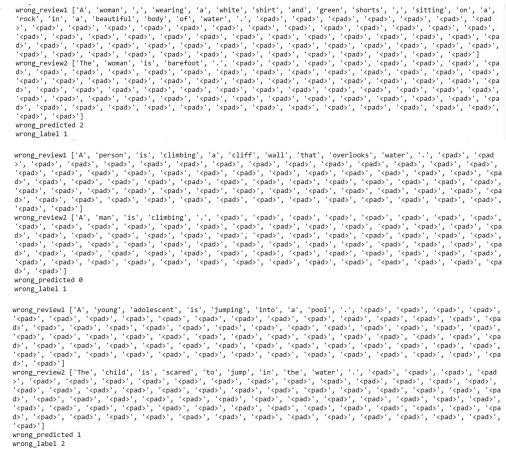

Figure 12: wrong prediction

CNN Best Model	
Val Acc	71.5
Fiction Test Acc	46.83417085
Government Test Acc	46.35826772
Slate Test Acc	42.81437126
Telephone Test Acc	48.85572139
Travel Test Acc	46.53767821

Figure 13: CNN Test Accuracy on Multinli

RNN Best Model	
Val Acc	71.8
Fiction Test Acc	50.75376884
Government Test Acc	47.93307087
Slate Test Acc	44.01197605
Telephone Test Acc	49.45273632
Travel Test Acc	45.51934827

Figure 14: RNN Test Accuracy on Multinli

3 Evaluation on Multinli

For the best trained CNN model and RNN model, I used both of them to test Multinli dataset. Compared with validation accuracy and test accuracy get from CNN model and RNN model, it is obvious that the validation accuracy using snli datasets is much higher than the test accuracy using multinli dataset. Compared all test accuracy in 5 genres, the test accuracy for fiction and telephone set is a little better than other genres and the test accuracy for the slate set is lower than other genres, it probably because these fiction and telephone dataset are more similar to training snli dataset and slate dataset is less similar to the training snli dataset. Hence, the classifier is not good at generalization, and dependent to the training set used. please see table 13 and 14 .

4 Explanation

For all figures, train/validation accuracy tables, you might see in jupyter notebook and excel in Github. This is my link: https://github.com/xy990/nlphw2