HW2 Report

bz957@nyu.edu

October 2018

https://github.com/abenpy/DS1011_NLP.git

1 Training on SNLI

1.1 Build pretrained word embedding matrix

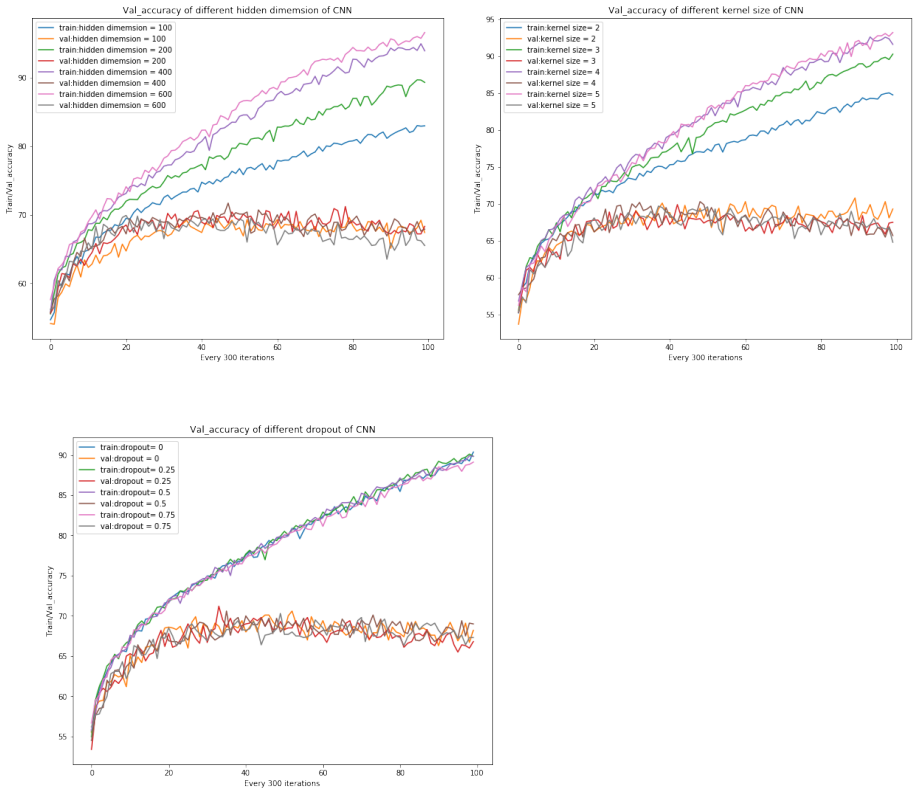
Load 50000 pretrained word embedding vectors from 'wiki-news-300d-1M.vec',and embedding $' < pad >'$ and $' < unk >'$, formed a 50002*300 matrix. Preprocess SNLI, Build CNN and Bi-directional RNN model.

1.2 Tunning Parameters on CNN

Hidden dimension size Hidden size list = $[100, 200, 400, 600]$. When hidden size = 200, the val-accuracy is a little bit higher than other parameters. It is because CNN model can keep most important information to the next hidden layers, as the input dimemsion is 300, slightly smaller hidden layer can remove trivial information. But too small hidden layer dimmension is not bigger enough to hold important information

Kernel size Kernel size list = $[2, 3, 4, 5]$. When hidden size = 3, the val-accuracy is slightly better at most time.According to empirical study, too large kernel can lose some information, while too small kernel can not decrease input dimension. The kernel size I tested is too close, that's why there is no big difference on val-accuracy

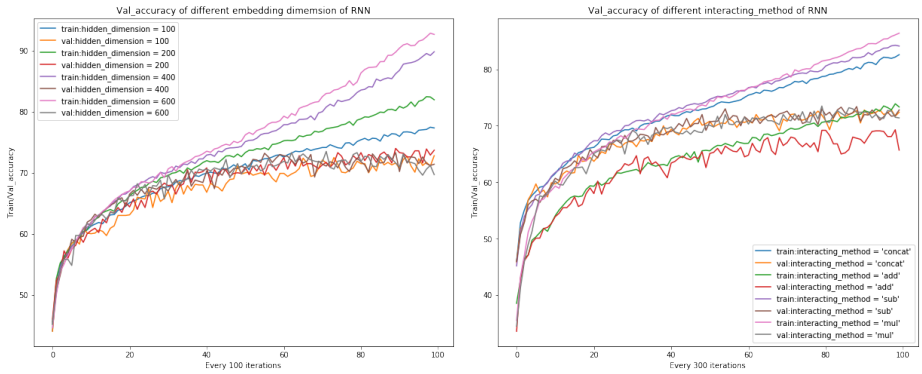
Dropout rate Dropout rate list = $[0, 0.25, 0.5, 0.75]$. When hidden size = 0.5, the val-accuracy is slightly better.According to previous study, dropout effectively allows to train a probability distribution of network architectures. Dropping a neuron with 0.5 probability gets the highest variance for this distribution. However in this model dropout rate didn't make a big change, because it is only two hidden layers model and not complex enough to let dropout takes effect



1.3 Tunning Parameters on RNN

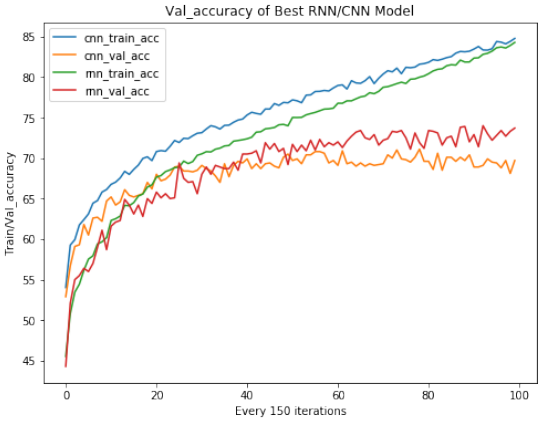
Hidden dimension size Hidden size list = $[100, 200, 400, 600]$. When hidden size= 200/400/600, the mean val-accuracy of 10 epochs is a little better than hidden size= 100. When hidden size= 200, the maximum val-accuracy=74 of 10 epochs is highest. As input dimension is 300, hidden layer dimension which is close to 300 or slightly larger than 300 achieve better val-accuracy. Because Bi-Gru model has a gate to decide which information in H_{t-1} to be kept into H_t, so large hidden dimension size keep more information. But too large will lead to sparse problem

Interacting Methods Interacting Methods list = $['concat', 'add', 'sub', 'mul']$. 'concat','sub','mul' achieve much higher val-accuracy than 'add', however the val-accuracy of method 'concat','sub','mul' are very close. 'add' causes element wise information lost or confusion. 'concat' keeps all the element wise information from both sentence.'sub' can delete trivial information in each sentence.'mul' strengthen the important element wise information.Here I choose 'concat'.



1.4 Best model parameters and 3 correct and incorrect predictions in the validation set

Best mode Best model is RNN with parameters: emb_size=30, embedding_dim=300, hidden_size=200, num_layers=2, num_classes=3,method='concat'
In the 10 epochs training, the Max_val_acc of RNN is 74.0,Mean_val_acc of RNN after 6000 iterations is 71. Max_val_acc of CNN is 71.1,Mean_val_acc of CNN after 6000 iterations is 69.33. RNN model outperform CNN.
According to exiting study and practict, RNN works well for tasks where length and relationship of text is important,these types of tasks include: question-answering, translation etc.But RNN is much slower. For tasks where feature detection in text is more important, for example, searching for sentiment detection, named entities etc.CNN may work well.This preject is to detect the relation between two sentence, so the length and the context relation of words are important.That is why RNN works better.



3 correct and incorrect predictions For the 1st wrong prediction, the model predict as entailment but indeed it is contradiction. Even 'young-old' are contradicting but other words are two similar. The model needs to be improved the distribution of weight on different words. For the 2nd wrong prediction,

the model predict as contradiction but indeed it is neutral. The last half of sentence 1 is contradicted with sentence2 but the first half let their relationship becomes neural. Besides there is unknow words may also affect the prediction. For the 3nd wrong prediction, the model predict as contradiction but indeed it is neutral. The model can not tell the relationship between verb 'wreck' and adj 'dusty' and 'clumsy'. Maybe the reason also lies in the pretrained embeddings.

```
3 Right Predictions
Two men standing near a metal structure in front of a brick wall .
Men stand in line at a water fountain in front of a brick building .
target:  2
predicted:  2
two girls sitting by a tree while playing .
The girls are not near a tree .
target:  1
predicted:  1
A man talks on his cellphone in private .
A man is on his cellphone .
target:  0
predicted:  0

3 Wrong Predictions
A young man wearing goggles , is jumping out of a pool , splashing water everywhere .
An old man jumping into a pool .
target:  1
predicted:  0
A building that portrays beautiful architecture stands in the sunlight as somebody on a bike passes by .
A <unk> rides past an abandoned warehouse on a rainy day
target:  2
predicted:  1
An old dusty car is half way in the brown water .
Someone wrecked their car a long time ago .
target:  2
predicted:  1
```

2 Evaluating on MultiNLI

2.1 Performance

RNN performans better than CNN on almost every genre of MultiNLI dataset.However, both models' val-accuracies are from 45-50, which is much lower than 70 in SNLI dataset

```
CNN for MNLI
travel 46.537678207739305
government 47.73622047244095
slate 45.808383233532936
telephone 49.15422885572139
fiction 45.82914572864322
```

```
RNN for MNLI
travel 46.23217922606925
government 52.75590551181102
slate 47.90419161676647
telephone 50.049751243781095
fiction 48.24120603015076
```

2.2 Analysis

A model classifier is highly dependent and conditioned on the training data, and may not generalize well to data from another domain/genre or in general another data distribution. So the model built based on SNLI is not generalize well on MNLI dataset. In real life we need to build a set of realistic test cases to ensure the model's generalization in deployment
Even though, RNN text classifier model can be generalized better than CNN, because RNN's can keep the connect between words in the whole context.