RNN/CNN-based Natural Language Inference Report

Zhihan Li

October 31, 2018

1 Training on SNLI

1.1 Tuning Mode 1

Hyperparameters:
CNN, hidden size = 200, kernel size = 3, dropout = FALSE

epoch	train loss	val loss	train acc	val acc
0	0.832963	1.109433	62.5	56.2
1	0.921935	1.080448	58.3	59.1
2	0.662638	0.636873	70.8	61.0
3	0.671945	0.656248	70.8	61.6
4	0.760337	0.630832	62.5	61.9
5	0.669956	1.413036	70.8	60.3
6	0.520316	0.604200	79.2	61.5
7	0.575984	1.079817	79.2	61.9
8	0.841169	1.215599	66.7	63.0
9	0.539128	0.703030	70.8	60.7

1.2 Tuning Mode 2

Hyperparameters:
CNN , hidden size = 100, kernel size = 3, dropout = FALSE

epoch	train loss	val loss	train acc	val acc
0	0.944406	0.982141	58.3	57.5
1	0.827740	0.909768	62.5	60.5
2	0.874666	0.917046	54.2	60.3
3	0.634986	0.857405	75.0	62.0
4	0.650674	0.515622	70.8	62.5
5	0.961410	0.519126	66.7	62.6
6	0.658704	0.755453	75.0	63.1
7	0.755238	0.738587	62.5	63.7
8	0.317213	0.866739	87.5	62.3
9	0.670436	0.942639	75.0	63.1

1.3 Tuning Mode 3

Hyperparameters:
CNN, hidden size = 100, kernel size = 5, dropout = FALSE

epoch	train loss	val loss	train acc	val acc
0	0.830431	0.815015	54.2	57.8
1	0.766540	0.666479	62.5	60.1
2	0.966648	0.630540	54.2	61.4
3	0.831226	0.846666	58.3	63.0
4	0.666718	1.890501	75.0	63.2
5	0.595547	1.344122	83.3	63.0
6	0.651425	1.344152	70.8	63.2
7	0.482722	0.713448	75.0	63.2
8	0.479897	1.153279	83.3	63.7
9	0.603719	0.892604	79.2	62.7

1.4 Tuning Mode 4

Hyperparameters:
CNN, hidden size = 100, kernel size = 3, dropout = TRUE

epoch	train loss	val loss	train acc	val acc
0	0.857222	0.765864	65.6	58.0
1	0.877461	0.903923	50.0	61.2
2	0.874572	1.027983	59.4	63.5
3	0.856565	0.751945	62.5	62.7
4	0.545737	1.270057	78.1	64.0
5	0.877181	0.743008	56.3	63.2
6	0.714999	0.759234	62.5	63.5
7	0.715825	0.628397	68.8	64.1
8	0.701119	0.507828	68.8	63.7
9	0.967130	0.698065	62.5	64.4

1.5 Tuning Mode 5

Hyperparameters:
RNN, hidden size = 100, dropout = FALSE

epoch	train loss	val loss	train acc	val acc
0	0.899823	0.994854	71.9	52.3
1	0.919549	0.893461	62.5	57.1
2	0.848188	0.768986	62.5	58.4
3	0.783004	0.704093	62.5	59.9
4	0.780601	0.783618	68.8	59.4
5	0.682274	1.044791	75.0	60.2
6	0.826559	0.771843	68.8	60.7
7	0.832078	1.279301	59.4	59.9
8	0.859190	1.118668	68.8	59.6
9	0.957020	1.126307	50.0	58.6

1.6 Tuning Mode 6

Hyperparameters:
RNN, hidden size = 100, dropout = TRUE

epoch	train loss	val loss	train acc	val acc
0	0.962078	1.161669	46.9	52.0
1	1.042803	1.023585	50.0	55.6
2	0.959237	0.947626	56.3	57.6
3	0.818148	1.052669	68.8	59.4
4	0.968561	0.779647	53.1	59.4
5	0.815410	0.431272	65.6	60.9
6	0.936348	1.098801	62.5	60.2
7	0.826626	0.934144	65.6	61.5
8	0.806180	1.267943	59.4	59.3
9	0.852467	0.839534	65.6	60.7

1.7 Conclusion

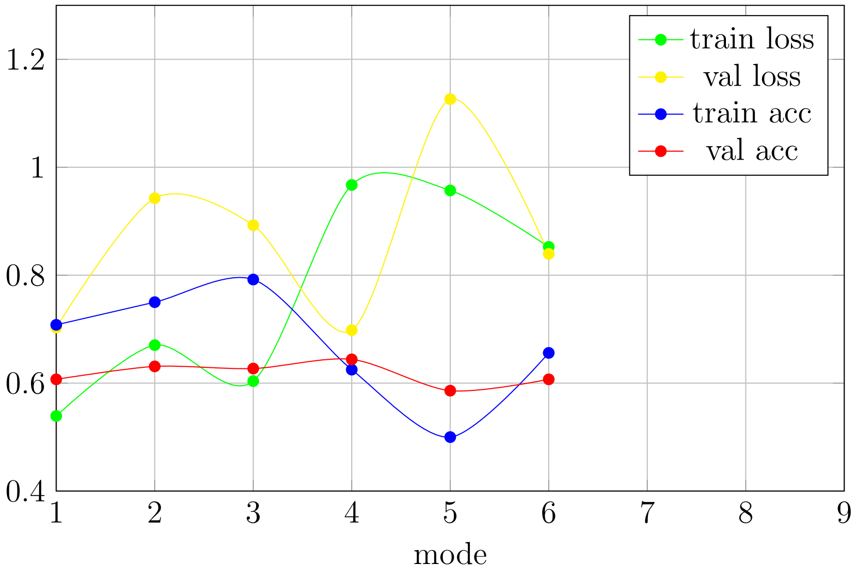

Figure 1: Tuning Curve

The hyperparameters I have trained are:

1. Hidden dimension of CNN and RNN (either 100 or 200)

With higher hidden dimension size, we can have more information of the data passed through layers but we need to have more data to train it. If the hidden size is low, the model might not be able to catch enough information regardless of the amount of data we use for training. Here hidden size of 100 is better on both CNN and RNN. This might be because model with 200 hidden features overfits the training data and does not generalize on the validation dataset.

2. Kernel size of CNN (either 3 or 5)

The kernel size defines the stride when we pass outputs from one layer to another. Smaller kernel size gives more details then the larger kernel size. Here kernel size of 3 is better since more number of neuron from the previous layer gives more details of the image.

3. Dropout (either TRUE or FALSE)

The dropout step with 0.2 probability improves my model for both RNN and CNN as it helps reduce the problem of overfitting.

My best model is mode 4, a CNN with hidden size = 100, kernel size = 3 and dropout = TRUE. Its validation accuracy is 64.4.

Correct samples:

index	sentence 1	sentence 2	label
92	An older man in an apron cleaning	A man is mopping the floor	neutral
99	The boy wearing the blue hooded top is holding a baby goat in his arms	A boy ran from a goat	contradiction
100	A team of people on a bike race	The people are riding bikes	entailment

Incorrect samples:

idx	sentence 1	sentence 2	true	predicted
84	A man and woman are sitting at a restaurant table holding hands	a man and a woman are celebrating a birthday	neutral	contradiction
91	A waitress is serving customers at a restaurant	The waitress is sitting in a chair ignoring the customers around her	contradiction	neutral
97	A black dog running through the forest	A dog playing outside	entailment	contradiction

1. The first incorrect sample is incorrect because the network does not learn the implicit relationship between hands holding and birthday celebration.

2. The second incorrect sample is incorrect because the model might not identify that serving customers and sitting in a chair contradicts.

3. The third incorrect sample is incorrect because a dog running through the forest can contradict to the dog playing because it could also run for other reasons. The true label provided is insufficient.

2 Evaluating on MultiNLI

Evaluation on the validation data set for MultiNLI separately for each genre.

Genre	RNN loss	CNN loss	RNN acc	CNN acc
government	1.194837	1.447474	37.598425	41.929134
slate	1.267342	1.219725	38.622754	40.818363
fiction	1.995804	0.991552	35.879397	40.301508
travel	1.391345	0.904735	38.289206	41.955193
telephone	1.428192	1.381033	38.109453	42.686567

The validation accuracy on MultiNLI is largely smaller than that on SNLI. This is because the model is dependent on the SNLI data since we use SNLI to train it. To generalize the classifier, we can use test dataset on further evaluation.

The CNN accuracy reaches its maximum in the telephone genre and minimum in the fiction genre. These could result from that the relationship between sentences in fiction is more complicated than that in telephone related text and it might be harder to identify the relationship between premise and hypothesis in fiction text.