

h2-written

kgz2437

February 2023

1 Question 1

Question 1b: This controls for confounding factors such as the quality of label collection + cleaning, underlying distributions. And it allows for fair comparison of different approaches. Additionally, it also makes it easier to reproduce other's results, as a person can always claim their approach achieves perfect performance on their own private dataset.

Question 1c: For fairly evaluating out of sample performance, all methods must have the same test set. Therefore we have to use a portion of training for validation

Question 1d: Negative reviews have a score less than or equal to 4 while positive reviews have a score greater than or equal to 7.

2 Question 2

$$\frac{\partial}{\partial \hat{y}_i} L(\hat{y}, y) = -1(i - 1 = y) + \frac{e^{\hat{y}_i}}{\hat{y}_1 + \hat{y}_2} = -1(i - 1 = y) + \mathbf{softmax}(\hat{y})$$

The derivative contains the output of the softmax, therefore combining the two makes the derivative very easy to compute. However, if we were to separate them: then we would need to compute the derivative of the softmax, which involves computing an additional term.

$$\frac{\partial}{\partial \hat{y}_i} \mathbf{softmax}(\hat{y}) = \frac{e^{\hat{y}_1 + \hat{y}_2}}{(e^{\hat{y}_1} + e^{\hat{y}_2})^2}$$

3 Question 3

4 Question 4

LSTM using the pretrained GLOVE embeddings achieved a test performance of 0.878, while without GLOVE it only achieved 0.677, using max epochs of 30 and

a patience of 3 (see Table 1). Therefore pretraining leads to much better performance most likely because the GLOVE encodes information that's not learnable just from the IMBD dataset. Typically, the training ends after less than 10 epochs. So, Additionally, after obtaining the optimal validation, I also tuned the max number of epochs and patience to obtain a slight improvement in performance to 0.7 for LSTM with random embeddings, achieved using maximum of 10 epochs with patience of 2 (see Table 2).

			val_acc	test_acc
pretrained	batch_size	lr		
False	16	0.005	0.5896	0.59104
		0.010	0.5652	0.56944
		0.050	0.5126	0.50000
	32	0.005	0.6262	0.62216
		0.010	0.5784	0.58084
		0.050	0.5142	0.51952
	64	0.005	0.6822	0.66968
		0.010	0.6146	0.61444
		0.050	0.5124	0.50912
True	16	0.005	0.8750	0.87756
		0.010	0.8542	0.86020
		0.050	0.7288	0.72872
	32	0.005	0.8724	0.87804
		0.010	0.5350	0.53500
		0.050	0.6196	0.61348
	64	0.005	0.8592	0.86648
		0.010	0.8090	0.80968
		0.050	0.7896	0.79688

Table 1: Results from hyperparameter search.

			lr	batch_size	val_acc	test_acc
pretrained	max_epochs	patience				
False	10	2	0.005	64	0.7012	0.70040
	15	3	0.005	64	0.6672	0.65340
	30	5	0.005	64	0.6516	0.63088
True	10	2	0.005	32	0.8702	0.87040
	15	3	0.005	32	0.8710	0.88140
	30	5	0.005	32	0.8440	0.84248

Table 2: Parameter search over patience and max epochs