HW1 Report

Yusuf Eren Tunç

27/10/2020

1 Sanity Checks

1.1 Loss

It is almost similar to what I expected because at the beginning weights are distributed around 0.01. When I check log(0.01), I saw that my values are quite close to that value.

1.2 Accuracy

As I expected accuracy was really close to zero due to equal weight distribution without any training. I was think that for each image there is a 0.01 chance to predict it true and accuracy should be around 10^{-6} and that is what I get.

2 Hyperparameter optimization

2.1 1-layer (0-hidden-layer) network

I will try 1-layer model which doesn't has any hidden layer and so any activation function. I create a loop in my train.py script. Moreover, that script keeps loss and accuracy history of validation test data which is %20 percent of training data. After that, I took maximum accuracy value divide it with maximum loss value. Script tries that with different hidden layer numbers, layer sizes and learning rates.

AF and HS	Learning Rate					
	0.01	0.003	0.001	0.0003	0.0001	0.00003
-, 256	0.69	1.39	1.68	1.67	1.41	0.88
-, 512	0.61	1.39	1.69	1.59	1.38	0.95
-, 1024	0.57	1.34	1.66	1.61	1.39	1.04

Table 1: 1-layer network

2.2 2-layer (1-hidden-layer) network

I will try 2-layer model which has one hidden layer and so there is activation function. I create a loop in my train.py script. Moreover, that script keeps loss and accuracy history of validation test data which is %20 percent of training data. After that, I took maximum accuracy value divide it with maximum loss value. Script tries that with different hidden layer numbers, layer sizes, activation functions and learning rates.

Layer Activations	Learning Rate					
	0.01	0.003	0.001	0.0003	0.0001	0.00003
S, 256	0.18	1.32	2.36	1.84	0.93	0.65
S, 512	0.16	1.46	2.34	2.02	1.27	0.76
S, 1024	0.15	1.75	2.67	2.12	1.42	0.84
T, 256	0.17	0.55	2.21	2.17	1.71	1.07
T, 512	0.12	0.63	2.19	2.23	1.98	1.29
T, 1024	0.08	0.44	2.45	2.29	2.16	1.43
R, 256	0.16	0.96	2.15	1.90	1.50	1.09
R, 512	0.16	1.43	2.39	2.00	1.71	1.25
R, 1024	0.16	1.59	2.40	2.39	1.92	1.43

Table 2: 2-layer network

2.3 3-layer (2-hidden-layer) network

I will try 3-layer model which has two hidden layer and so there is activation function. I create a loop in my train.py script. Moreover, that script keeps loss and accuracy history of validation test data which is %20 percent of training data. After that, I took maximum accuracy value divide it with maximum loss value. Script tries that with different hidden layer numbers, layer sizes, activation functions and learning rates.

Layer	Learning Rate					
Activations	0.01	0.003	0.001	0.0003	0.0001	0.00003
S, 256	0.16	0.29	2.33	1.46	0.67	0.35
S, 512	0.16	0.16	2.82	1.88	0.85	0.34
S, 1024	0.18	0.15	2.69	2.27	1.26	0.61
T, 256	0.17	0.23	3.02	2.58	2.04	1.11
T, 512	0.15	0.18	2.67	2.72	2.36	1.54
T, 1024	0.08	0.17	2.30	2.82	2.38	1.91
R, 256	0.16	1.63	2.57	2.23	1.74	0.85
R, 512	0.33	2.07	3.14	2.85	1.97	1.25
R, 1024	0.78	1.32	3.42	3.46	2.50	1.72

Table 3: 3-layer network

3 The best hyperparameter

3.1 Results

Best hyperparameters can be received from tables. Therefore, learning rate is 0.0003, hidden layer is 2, activation function is ReLu and layer size is 1024.
Test accuracy is 0.33

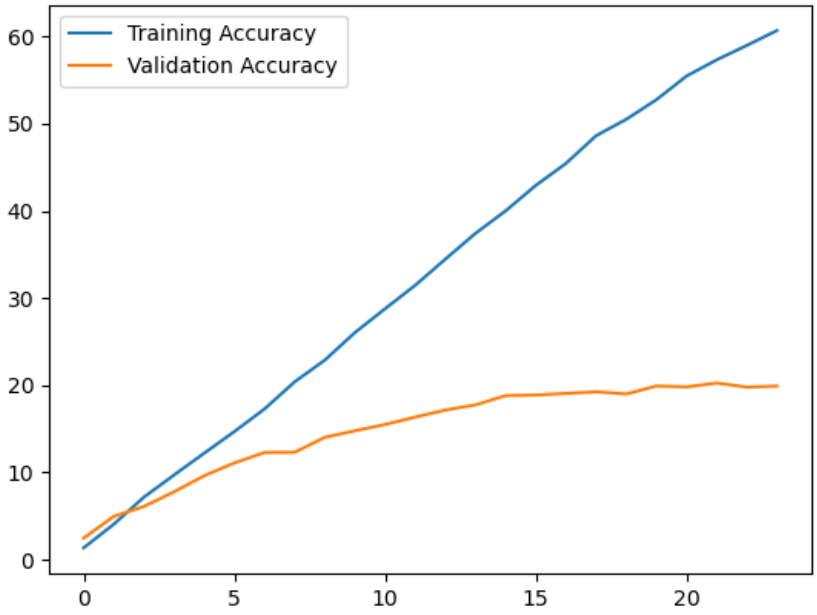

Figure 1: Final Accuracy Graph

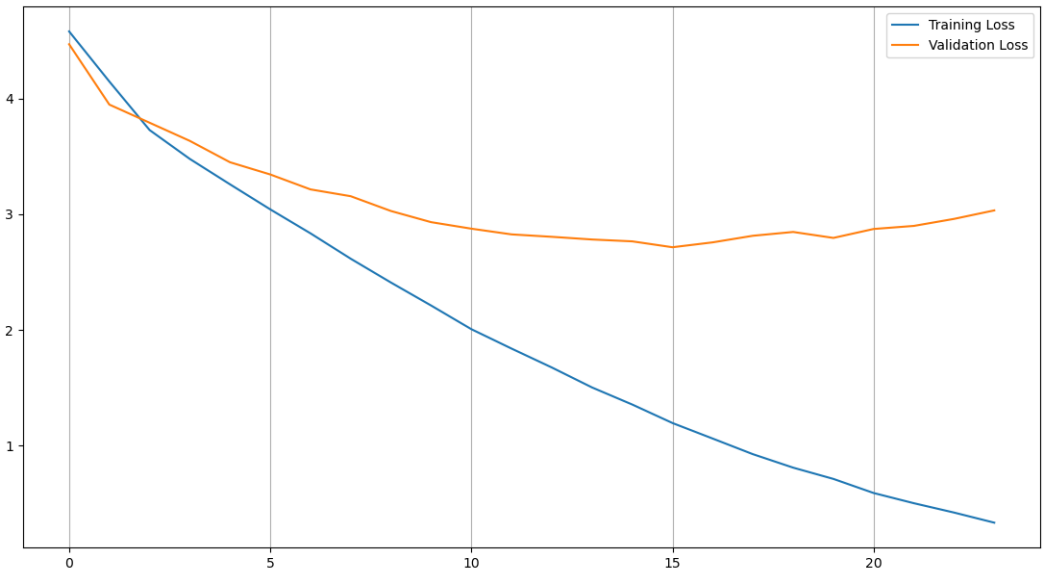

Figure 2: Final Loss Graph

3.2 Overfitting countermeasures

To prevent overfitting, I examine loss graph that created in training part. At best hyperparameters, I give 32 epoch to system then examined loss graph. Overfitting can be seen at graph. If training loss line is much lower then validation loss line, model is overfitting. When I give 32 epoch to system, training cause overfitting at model so I decide to give 24 epoch to system to prevent overfitting.

You can use the train/validation loss graph in your explanation if you want.

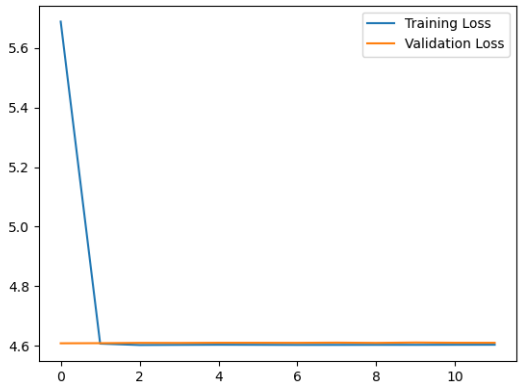

Figure 3: A Fail Loss Graph

4 Comments

Homework was very confusing at the beginning. After training videos, I got some ideas about to what to do and how to do and I tried to make that ideas real. With human mistakes and failures, I think that homework improve myself.