Middle East Technical University Northern Cyprus Campus
Computer Engineering Program

CNG 409 Special Topics: Introduction to Machine Learning
Fall 2021-2022

Homework 1
Artificial Neural Networks

2243392 Pınar Dilbaz

Chapter 1

INTRODUCTION

Thanks to this assignment, I had the opportunity to work on Artificial Neural Network. I gained experience in knowing and coding ANNs closely. In this assignment we were expected to design various image classifiers and optimize hyperparameters. In this way, I had the chance to experience what I could use for hyperparameter optimization for ANNs, and when I changed which metrics, I could get better results. At the same time, I had never used the Pytorch library before, although I was a bit of a stranger at first, I was able to advance my homework in line with my research. Additionally, the challenge I struggled with with the homework was that it took too long for the results to come out. It was a very interesting and enjoyable assignment for me. While doing my homework, I first glanced at the reproducibility page of PyTorch to get more familiar the library. Then I loaded and manipulated the CIFAR-10 dataset which is used for our homework. I changed some hyperparameters and created my models, and train them. Finally, I evaluated all my results, I had a lot of results, you can see some of them in the following chapters.

Chapter 2

SANITY CHECK

2.1 Accuracy

Since we have 10 labels, we expect 0.1 because of 1/10.

2.2 Loss

We expect the loss to be 2.302 using the Cross Entropy loss function formula. In addition, loss value from Pytorch is 2.3139 . So, they are similar.

2.3 Dataset

I separated the validation set via train set using torch.utils.data.random split() as said in the assignment pdf.

Chapter 3

HYPERPARAMETER OPTIMIZATION

3.1 1-Layer(0-Hidden Layer) Network

Below, you can see the validation accuracy and validation loss values I have received for 1-Layer Network in a table.

Hyperparameters;

- *Learning Rate:* 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7

Learning Rate	Accuracy	Validation Loss
1e-2	14.57	3.1102
1e-3	38.020	1.8560
1e-4	41.070	1.7512
1e-5	36.850	1.8575
1e-6	26.040	2.1293
1e-7	15.820	2.2729

Figure 3.1: Table of Accuracy and Loss for 1-Layer Network

3.2 2-Layer(1-Hidden Layer) Network

Below, you can see the validation accuracy and validation loss values I have received for 2-Layer Network in a table.

Hyperparameters;

- *The number of neurons in each layer:* 150, 200, 350
- *Activation Functions:* relu, hardwish, tanh
- *Learning Rate:* 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7

Activation Function	Layer Size	Learning Rate	Accuracy	Validation Loss
Relu	150	1e-2	30.730	4.0534
Hardwish	150	1e-2	29.930	3.0604
Tanh	150	1e-2	29.990	1.9685
Relu	200	1e-2	32.480	3.7775
Hardwish	200	1e-2	34.010	3.6356
Tanh	200	1e-2	31.210	1.9540
Relu	350	1e-2	32.360	4.9737
Hardwish	350	1e-2	28.800	6.6861
Tanh	350	1e-2	31.760	2.0022
Relu	150	1e-3	48.880	1.4941
Hardwish	150	1e-3	48.380	1.5012
Tanh	150	1e-3	40.430	1.6940
Relu	200	1e-3	49.020	1.4716
Hardwish	200	1e-3	48.650	1.5011
Tanh	200	1e-3	40.250	1.6951
Relu	350	1e-3	47.180	1.5794
Hardwish	350	1e-3	48.810	1.4783
Tanh	350	1e-3	41.530	1.6840

Figure 3.2: Table of Accuracy and Loss for 2-Layer Network

Relu	150	1e-4	48.540	1.4898
Hardwish	150	1e-4	48.680	1.4941
Tanh	150	1e-4	45.900	1.5528
Relu	200	1e-4	49.010	1.4754
Hardwish	200	1e-4	48.420	1.4971
Tanh	200	1e-4	46.810	1.5313
Relu	350	1e-4	48.350	1.4553
Hardwish	350	1e-4	48.380	1.4527
Tanh	350	1e-4	47.790	1.4935
Relu	150	1e-5	40.480	1.7340
Hardwish	150	1e-5	40.320	1.7444
Tanh	150	1e-5	38.700	1.7985
Relu	200	1e-5	41.170	1.7131
Hardwish	200	1e-5	40.940	1.7180
Tanh	200	1e-5	39.130	1.7738
Relu	350	1e-5	42.290	1.6820
Hardwish	350	1e-5	41.730	1.6992
Tanh	350	1e-5	39.730	1.7469

Figure 3.3: Table of Accuracy and Loss for 2-Layer Network

Relu	150	1e-6	27.450	2.0851
Hardwish	150	1e-6	26.840	2.0963
Tanh	150	1e-6	28.960	2.0642
Relu	200	1e-6	28.900	2.0662
Hardwish	200	1e-6	27.450	2.0921
Tanh	200	1e-6	30.140	2.0407
Relu	350	1e-6	30.330	2.0391
Hardwish	350	1e-6	29.110	2.0570
Tanh	350	1e-6	30.990	2.0071
Relu	150	1e-7	18.340	2.2676
Hardwish	150	1e-7	16.720	2.2760
Tanh	150	1e-7	21.270	2.2453
Relu	200	1e-7	19.650	2.2646
Hardwish	200	1e-7	19.050	2.2729
Tanh	200	1e-7	19.870	2.2533
Relu	350	1e-7	21.880	2.2578
Hardwish	350	1e-7	21.990	2.2548
Tanh	350	1e-7	22.700	2.2246

Figure 3.4: Table of Accuracy and Loss for 2-Layer Network

3.3 3-Layer(2-Hidden Layer) Network

Below, you can see the validation accuracy and validation loss values I have received for 3-Layer Network in a table.

Hyperparameters;

- *The number of neurons in each layer:* 150, 200, 350
- *Activation Functions:* relu, hardwish, tanh
- *Learning Rate:* 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7

Activation Function	Layer Size	Learning Rate	Accuracy	Validation Loss
Relu	150	1e-2	27.980	3.2263
Hardwish	150	1e-2	28.230	3.7530
Tanh	150	1e-2	31.110	1.9555
Relu	200	1e-2	28.620	3.8055
Hardwish	200	1e-2	29.670	3.7254
Tanh	200	1e-2	27.410	2.0625
Relu	350	1e-2	28.520	5.5653
Hardwish	350	1e-2	28.590	6.7969
Tanh	350	1e-2	31.160	2.1178
Relu	150	1e-3	49.070	1.4748
Hardwish	150	1e-3	49.080	1.4845
Tanh	150	1e-3	40.310	1.7079
Relu	200	1e-3	49.070	1.4925
Hardwish	200	1e-3	48.960	1.4992
Tanh	200	1e-3	40.730	1.6948
Relu	350	1e-3	47.340	1.5779
Hardwish	350	1e-3	47.290	1.5433
Tanh	350	1e-3	41.550	1.6766

Figure 3.5: Table of Accuracy and Loss for 3-Layer Network

Relu	150	1e-4	46.840	1.5327
Hardwish	150	1e-4	46.610	1.5488
Tanh	150	1e-4	43.990	1.6252
Relu	200	1e-4	47.830	1.5218
Hardwish	200	1e-4	48.270	1.5074
Tanh	200	1e-4	44.570	1.6016
Relu	350	1e-4	48.480	1.4910
Hardwish	350	1e-4	48.290	1.4910
Tanh	350	1e-4	46.130	1.5459
Relu	150	1e-5	39.260	1.8568
Hardwish	150	1e-5	35.880	2.1367
Tanh	150	1e-5	39.760	1.8149
Relu	200	1e-5	38.920	1.9053
Hardwish	200	1e-5	32.990	2.3912
Tanh	200	1e-5	41.160	1.7720
Relu	350	1e-5	40.590	1.8486
Hardwish	350	1e-5	35.550	2.1669
Tanh	350	1e-5	36.470	2.0567

Figure 3.6: Table of Accuracy and Loss for 3-Layer Network

Relu	150	1e-6	23.460	3.9546
Hardwish	150	1e-6	22.460	4.0434
Tanh	150	1e-6	24.970	4.3327
Relu	200	1e-6	23.160	3.8603
Hardwish	200	1e-6	23.480	4.0431
Tanh	200	1e-6	23.400	4.4728
Relu	350	1e-6	27.110	3.8861
Hardwish	350	1e-6	24.400	4.2170
Tanh	350	1e-6	23.960	4.6745
Relu	150	1e-7	2.740	4.9498
Hardwish	150	1e-7	4.540	4.9686
Tanh	150	1e-7	10.450	4.9046
Relu	200	1e-7	6.650	5.2416
Hardwish	200	1e-7	6.290	5.2390
Tanh	200	1e-7	8.820	5.2075
Relu	350	1e-7	11.170	5.7411
Hardwish	350	1e-7	14.020	5.7794
Tanh	350	1e-7	16.340	5.7306

Figure 3.7: Table of Accuracy and Loss for 3-Layer Network

3.4 The Best Results

3.4.1 1. Result

As we can clearly see when we look at the tables above, the best validation accuracy is 49.080 .
Hyperparameters;

- *Test Accuracy:* 48.630
- *The number of hidden layers:* 2
- *The number of neurons in each layer:* 150
- *Activation Functions:* hardwish
- *Learning Rate:* 1e-3

3.4.2 2. Result

Second best validation accuracy is 49.070 .
Hyperparameters;

- *Test Accuracy:* 49.180
- *The number of hidden layers:* 2
- *The number of neurons in each layer:* 150
- *Activation Functions:* relu
- *Learning Rate:* 1e-3

3.4.3 3. Result

Third best validation accuracy is also 49.070 .

3.4.4 2. Result

Second best validation accuracy is 49.070 .
Hyperparameters;

- *Test Accuracy:* 47.810
- *The number of hidden layers:* 2
- *The number of neurons in each layer:* 200
- *Activation Functions:* relu
- *Learning Rate:* 1e-3

You can see the training and validation losses graph in the figure below. I found it sufficient to use 50 epochs.

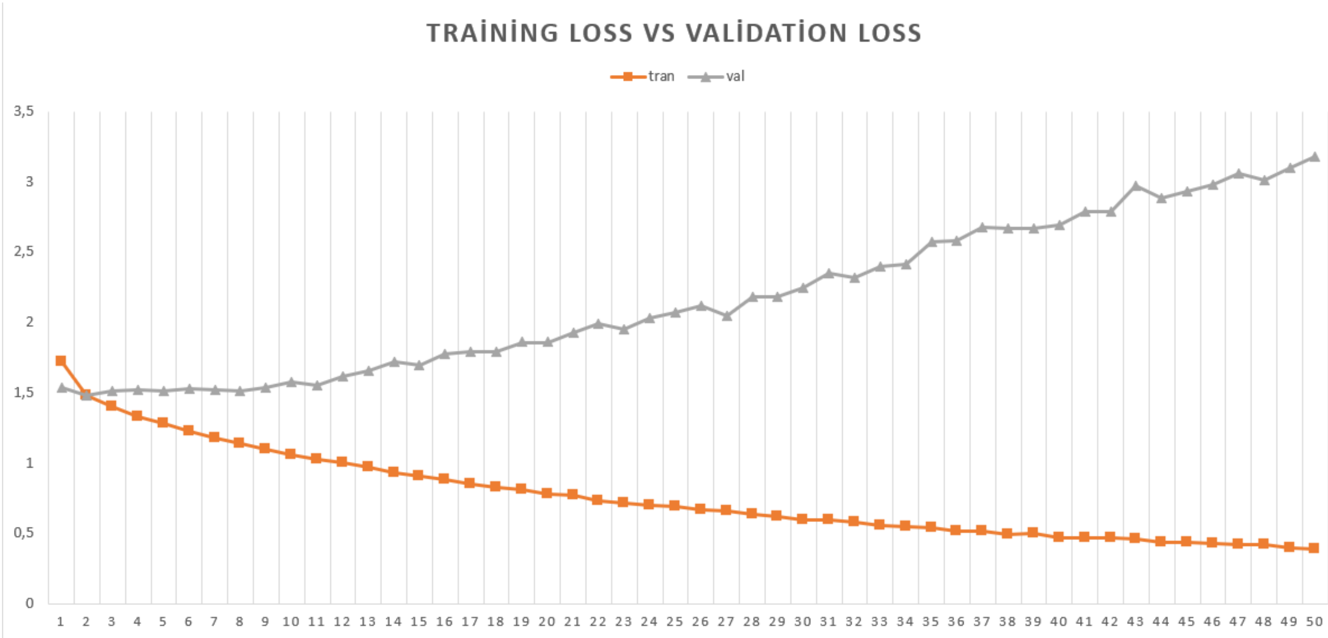

Figure 3.8: Train Loss vs Validation Loss for the Best Hyperparameter

I looked at the validation loss and training loss to see if it's overfitting. If the epoch number is increasing and the losses are decreasing, if they suddenly start to increase, it may be overfitting. If my model starts to memorize the data set I use for training more than necessary, it can be possible to overfitting. I separated my dataset as validation set and train set, and in this way, the possibility of overfitting decreased. Also, to avoid overfitting, I created an uncomplicated model, thus reducing the possibility of overfitting. Training should stop when the gap between validation loss and train loss starts to increase and reaches a certain level. If the performance of the model on the validation set starts to decrease , then we understand that it is time for the training process to stop. In addition, increasing the training data can be used as an overfitting prevention method.

Since our dataset is balanced, accuracy is a good metric to measure performance. Conversely, different metrics would be a better option if we had an unbalanced dataset.

3.5 Answers

1. Advantages/disadvantages of using a small learning rate
   - A learning rate that is too small can cause the process to get stuck.
2. Advantages/disadvantages of using a big learning rate
   - A learning rate that is too large can cause the model to converge too quickly to a suboptimal solution.
3. Advantages/disadvantages of small batch size
   - Reduced amount of Work in Process and reduced cycle time. Since the batch is smaller, it's done faster, thus reducing the cycle time.
   - Decreased risk and variability. Since the batch contains less content, it's by essence easier to control and validate.
   - Higher flexibility. Already-delivered-batches being smaller, the set of assumptions they are based on are smaller, and it's easier to adjust it.
4. Advantages/disadvantages of using a big batch size
   - If batch sizes are too large, it leads to poor generalization.
   - Large batch sizes are slower and heavier than small batch sizes.