Asadullah Hill Galib
CSE 803: Computer Vision
Homework 4
November, 11

1 Optimization and Fitting

After fitting with L2 loss, the W and b are:

$$W = \begin{bmatrix} 1.0005 & -0.002995 \\ 0.000036 & 0.615 \end{bmatrix}$$
$$b = \begin{bmatrix} -0.00433 \\ 0.25339 \end{bmatrix}$$

I have used the following learning rate : 0.0002.
The output after fitting:

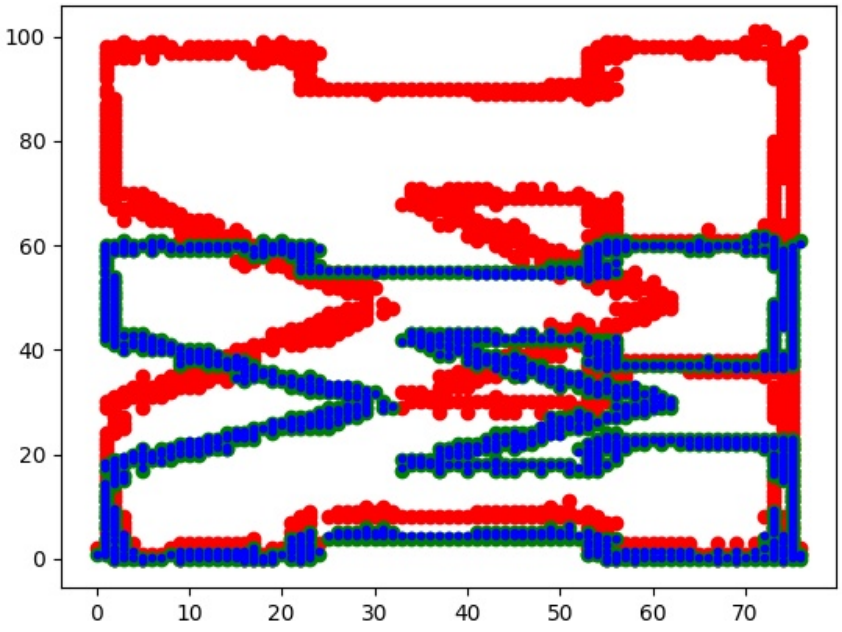

Figure 1: L2 Loss Fitting

2 Softmax Classifier with One Layer Neural Network

2.2

A fully connected neural network is implemented for this part. Softmax loss is used in the final layer of the network. There are two layers only: a fully connected layer followed by the softmax layer. Thers is not any hidden layer for this part. The network is optimized using gradient descent.

The hyperparameters used in this network:

- learning rate: 0.01
- batch size: 128
- learning decay: 0.95
- epochs: 100
- regularization factor: 0.5
- reg (Scalar giving L2 regularization strength): 0.0

Training and Validation accuracy:

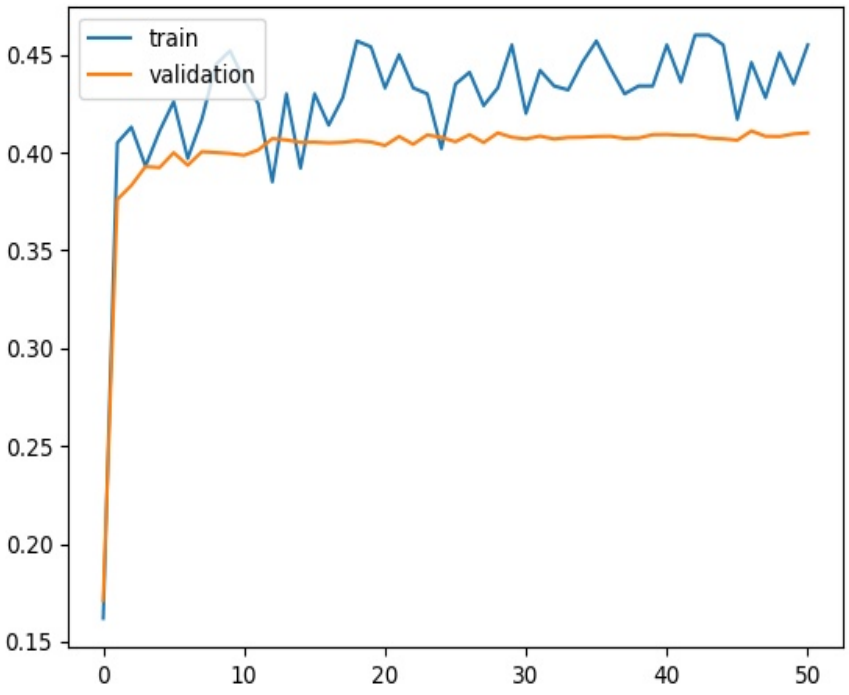

Figure 2: Softmax Classifier without hidden layer

2.3

Accuracy on test set: **41.18%**

```
(Epoch 46 / 50) train acc: 0.446000; val_acc: 0.411000
(Iteration 29001 / 31250) loss: 1.662461
(Epoch 47 / 50) train acc: 0.428000; val_acc: 0.408300
(Epoch 48 / 50) train acc: 0.451000; val_acc: 0.408200
(Iteration 30001 / 31250) loss: 1.837599
(Epoch 49 / 50) train acc: 0.435000; val_acc: 0.409500
(Iteration 31001 / 31250) loss: 1.450769
(Epoch 50 / 50) train acc: 0.455000; val_acc: 0.410000
Test accuracy: 0.4118
```

Figure 3: Accuracy on test set without hidden layer

2.4

The hyperparameters are analyzed in the experimentation:

- learning rate: 0.0001, 0.001, 0.01, 0.1, 1
- batch size: 64, 128, 256, 500, 1000
- learning decay: 0.95, 0.80, 0.90
- epochs: 100
- regularization factor: 0.5
- reg (Scalar giving L2 regularization strength): 0.0, 0.01, 0.1

Experimental plots with variation of hyperparameters:

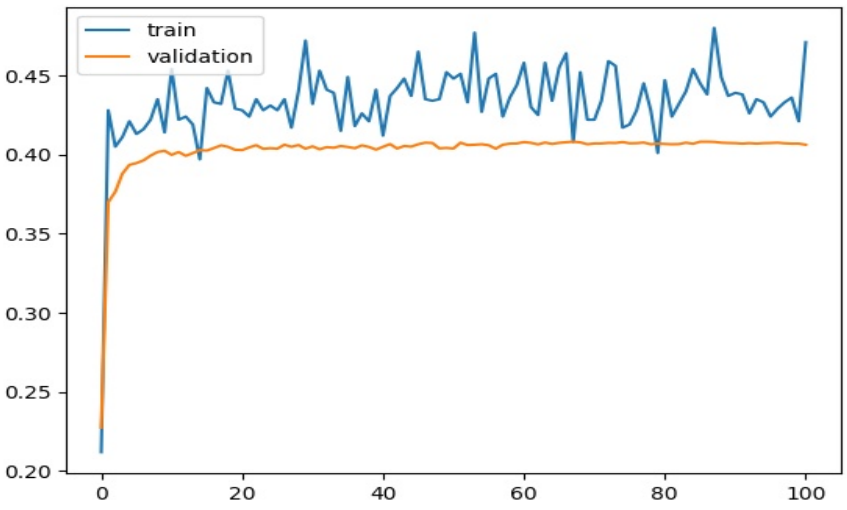

Figure 4: Using smaller batch size (32): More fluctuation while training

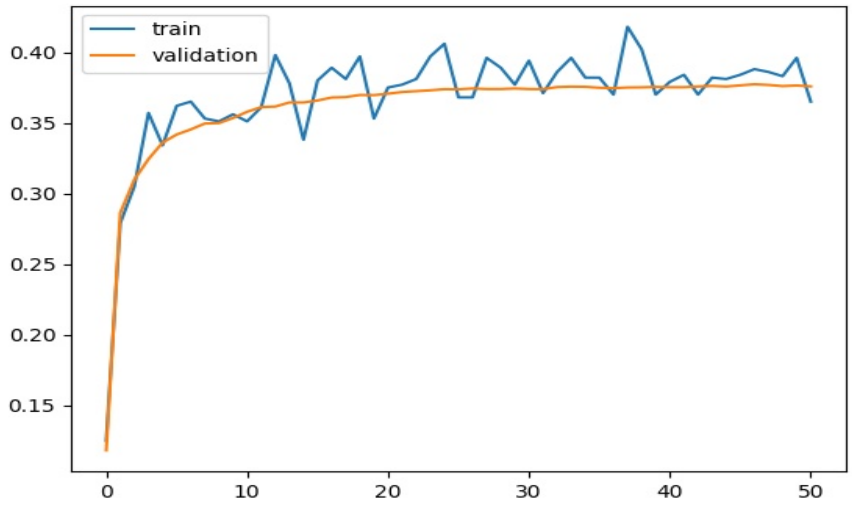

Figure 5: Using smaller learning rate (0.0001): Performance is not as good as lr=0.01. Test accuracy: 36.05%

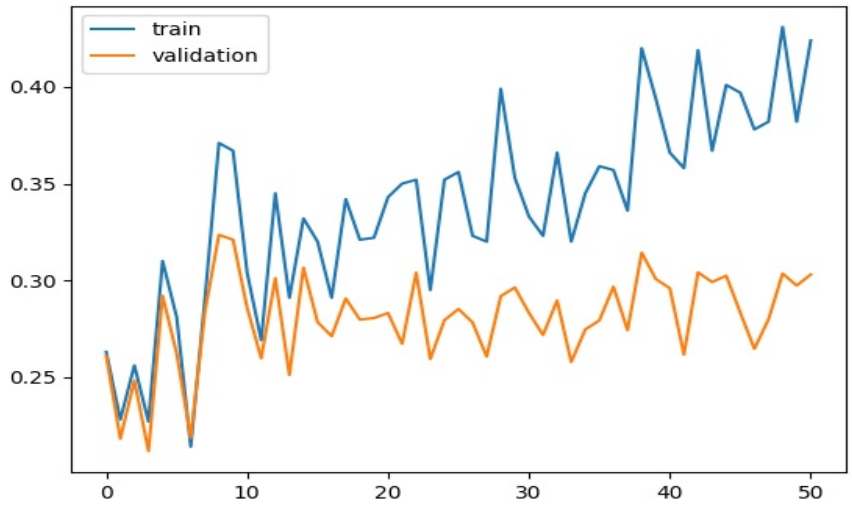

Figure 6: Using bigger learning rate (1.0): Model does not generalize well. Test accuracy: 29.78%

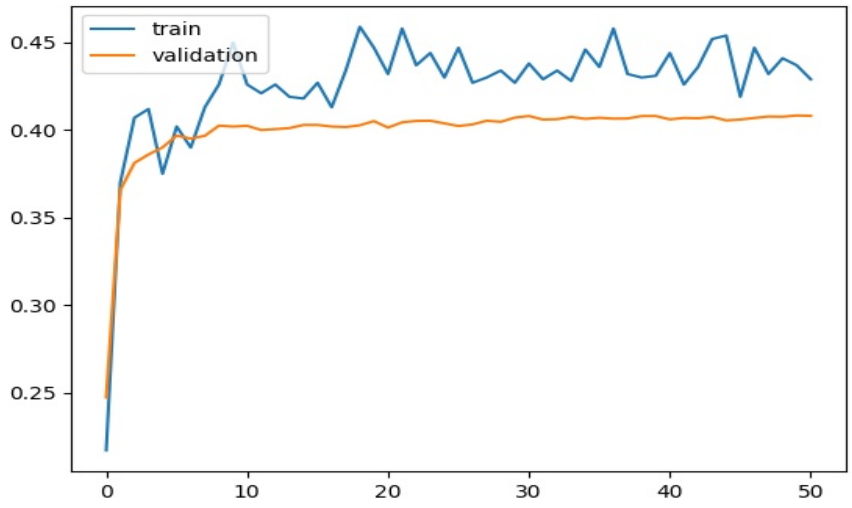

Figure 7: Using bigger batch size: Bigger batch size does not make any difference. Test accuracy: 40.08%

3 Softmax Classifier with Hidden Layers

3.2

A fully connected neural network with hidden layers is implemented for this part. Softmax loss is used in the final layer of the network. There are four layers: a fully connected layer followed by a relu layer, followed by another fully connected network followed by the softmax layer. Thers is an hidden layer of dimension = 64 used for this part. The network is optimized using gradient descent.

The hyperparameters used in this network:

- hidden size: 512
- learning rate: 0.1
- batch size: 500
- learning decay: 0.95
- epochs: 100
- regularization factor: 0.5
- reg (Scalar giving L2 regularization strength): 0.0

Training and Validation accuracy:

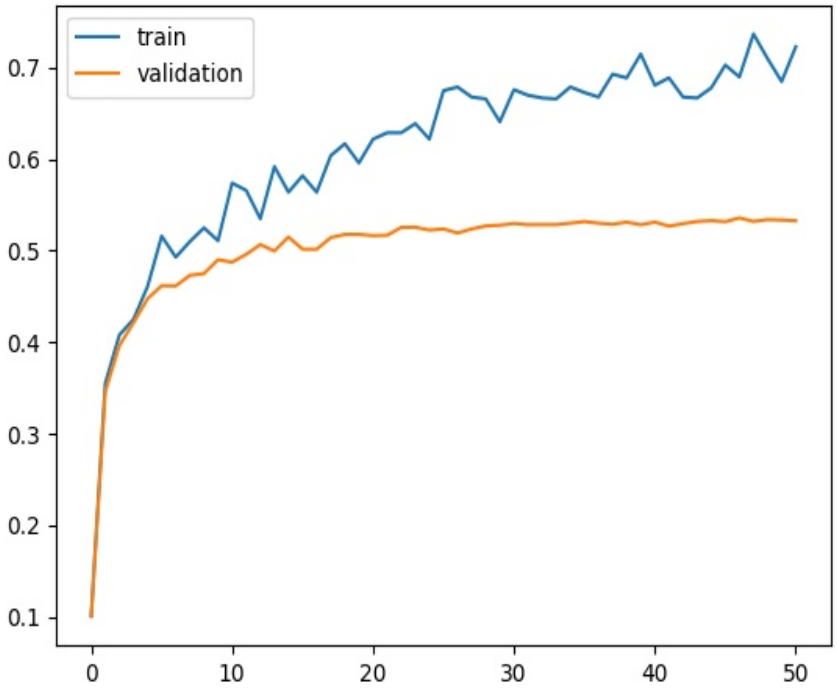

Figure 8: Softmax Classifier with hidden layers

3.3

Accuracy on test set: **52.72%**

```
(Epoch 45 / 50) train acc: 0.703000; val_acc: 0.531800
(Epoch 46 / 50) train acc: 0.690000; val_acc: 0.535900
(Epoch 47 / 50) train acc: 0.737000; val_acc: 0.532100
(Epoch 48 / 50) train acc: 0.710000; val_acc: 0.534000
(Epoch 49 / 50) train acc: 0.685000; val_acc: 0.533600
(Epoch 50 / 50) train acc: 0.723000; val_acc: 0.532800
Test accuracy: 0.5272
```

Figure 9: Accuracy on test set with hidden layer

3.4

The hyperparameters are analyzed in the experimentation:

- hidden size: 32, 64, 128, 256, 512
- learning rate: 0.0001, 0.001, 0.01, 0.1, 1

- batch size: 64, 128, 256, 500, 1000
- learning decay: 0.95, 0.80, 0.90
- epochs: 100
- regularization factor: 0.5
- reg (Scalar giving L2 regularization strength): 0.0, 0.01, 0.1

Experimental plots with variation of hyperparameters:

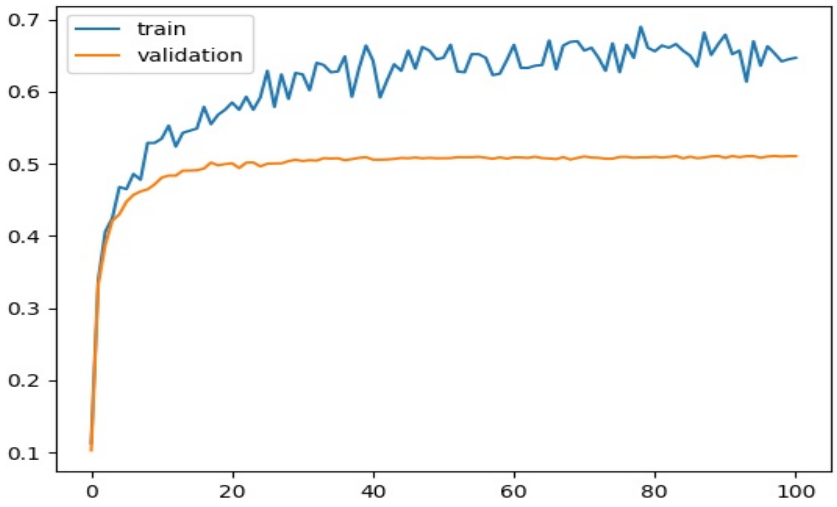

Figure 10: Hidden size (128): Performance is poorer with smaller hidden size than hidden size of 512. Test accuracy: 50.03%

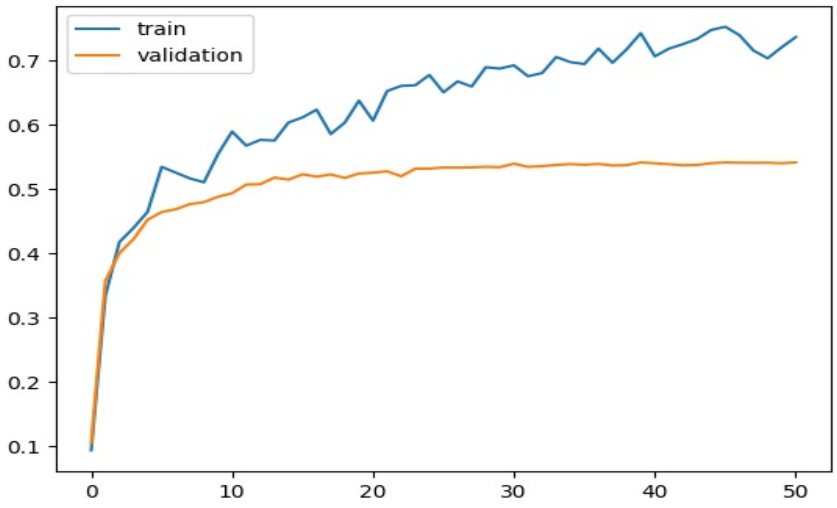

Figure 11: Hidden size (32): Performance is the worst with the smalles hidden size (32). Test accuracy: 48.43%

4 Fooling Images

4.1

Corresponding code added

4.2

Using gradient ascent, a fooled image is produced from a correctly classified image.

In the iterations of gradient ascent, the score (the softmax scores) of the correct class is being reduced and the score of the target class is increased. In the end, the score of the target class exceeds the score of the correct class, so the target class is prevailed and the fooled image is classified as the target class.

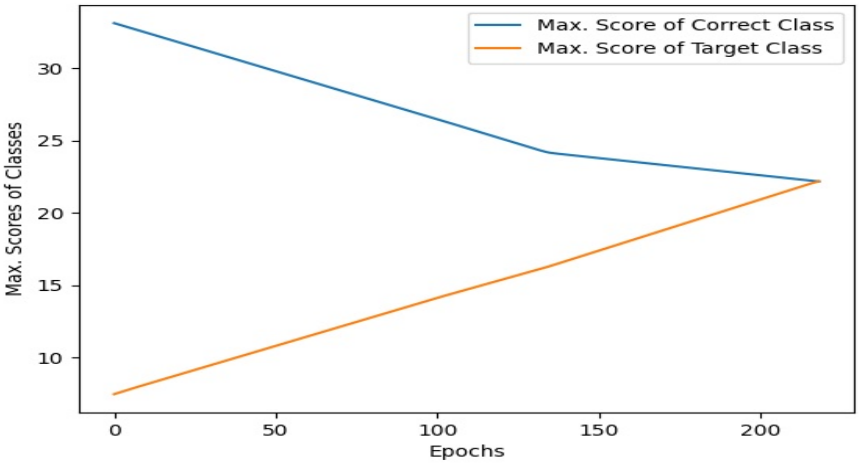

Figure 12: Softmax score of the correct class and target class
*NB: While reading the labels, please ignore the "Max." word in all cases, that is a typo

It suggests that using proper hyperparameters, the fooling model can create fake image close to real image, which can mislead the classifier. The fooling model is fairly robust as we have tried with different images and it works well.

The fooling image, real image, and the 10 times magnified difference image are shown below. The fooling and the real one are very close. Unfortunately, I can't interpret the difference image, I struggled with this part (RGB value ranging from 0 to 1, but it requires 0 to 255. I kind of messed up the conversions here).

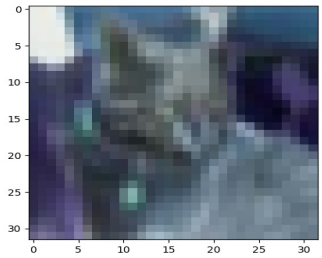

Figure 13: Original Image

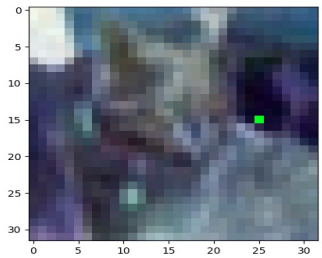

Figure 14: Fooling Image

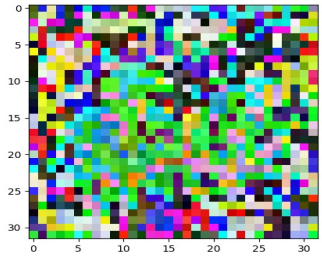

Figure 15: 10 times magnified Difference Image