HW6 Report

Rui Huang, Zonglin Peng

The implementation is written in* ***homework6_rhuang2_zpeng.py*

Part One: 3-layer neural network

Check gradient and hyperparameters optimizations

After the stochastic gradient descent and the cost function was implemented correctly, the grade reported from the check_grad was supposed to and became small enough, as shown below.

Given arrays of possible hyperparameters (neuron numbers, epsilon, mini-batch size, and epoches), we used nested for loops to find the best ones that will improve the fCE on validation set.

Note that the regularization strength uses the default value of 0.001 for both L1 and L2. After tried 2 neuron numbers, 2 epsilons, 3 mini-batch sizes, and 2 epoches, which are 24 different hyperparameter settings, the best fCE is 0.053 and fPC 0.9922 on validation set. The optimized process is shown below as a print stream, where each of the 4 columns matches the title, and the line saying "OPTIMIZED" indicates better fCE and fPC achieved by the new hyperparameters.

```
In [34]: run homework6_final.py
_____ Check Gradient _____
1.4679559150366342e-06
_____ Find best hyperparameters _____
Neuron  Epsilon Batch   Epoch
40      0.005   32      100
OPTIMIZED:  | fCE: 0.2047186006567583  | fPC: 0.9458
40      0.005   32      200
OPTIMIZED:  | fCE: 0.12769252652731727  | fPC: 0.9694
40      0.005   64      100
40      0.005   64      200
40      0.005   128     100
40      0.005   128     200
40      0.01    32      100
40      0.01    32      200
OPTIMIZED:  | fCE: 0.05759823697587674  | fPC: 0.9918
40      0.01    64      100
40      0.01    64      200
40      0.01    128     100
40      0.01    128     200
50      0.005   32      100
50      0.005   32      200
50      0.005   64      100
50      0.005   64      200
50      0.005   128     100
50      0.005   128     200
50      0.01    32      100
50      0.01    32      200
OPTIMIZED:  | fCE: 0.05542571148987705  | fPC: 0.9918
50      0.01    64      100
50      0.01    64      200
50      0.01    128     100
50      0.01    128     200
BEST HYPER PARAMETERS:
50      0.01    32      200
```

Screenshot of check_grad and hyperparameters optimizations

As the screenshot above shows, the best hyperparameters are:
Hidden neurons = 50; Epsilon = 0.01; Batch size = 32; Epoch = 200

Neural networks training

Training set and training labels are passed into the training process using SGD, and the best set of hyperparameters are used to train the NN during the SGD process.

fCE and fPC are used to evaluate the accuracy of the NN, and the values are found from both testing and training set. The values are shown below.

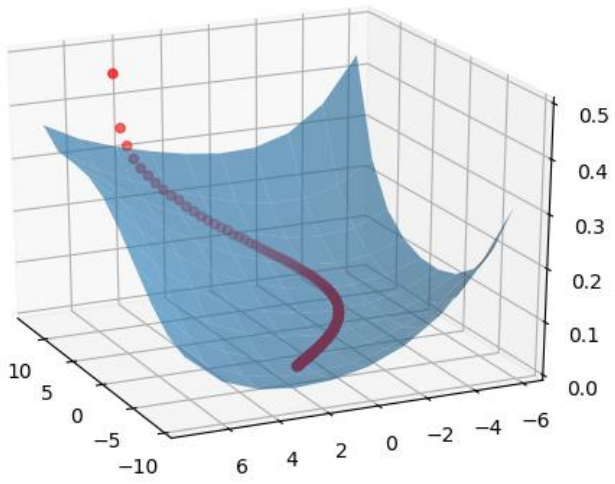
```
In [36]: run homework6_rhuang2_zpeng.py
_____ Starts Training _____
Iteration: 180  | fCE: 0.07976843458368874  | fPC: 0.9757
Iteration: 181  | fCE: 0.07969005104722399  | fPC: 0.9766
Iteration: 182  | fCE: 0.07944327626061967  | fPC: 0.9761
Iteration: 183  | fCE: 0.07916798382323288  | fPC: 0.9757
Iteration: 184  | fCE: 0.07859489111029554  | fPC: 0.9765
Iteration: 185  | fCE: 0.07961766367209532  | fPC: 0.976
Iteration: 186  | fCE: 0.07917193949770378  | fPC: 0.9762
Iteration: 187  | fCE: 0.07960047981875873  | fPC: 0.9764
Iteration: 188  | fCE: 0.0791388758956667  | fPC: 0.9758
Iteration: 189  | fCE: 0.08041470239222703  | fPC: 0.9754
Iteration: 190  | fCE: 0.07898756380438916  | fPC: 0.9767
Iteration: 191  | fCE: 0.07943164692220196  | fPC: 0.9763
Iteration: 192  | fCE: 0.07977725998250118  | fPC: 0.9762
Iteration: 193  | fCE: 0.08009863172482806  | fPC: 0.9761
Iteration: 194  | fCE: 0.07983986705662767  | fPC: 0.976
Iteration: 195  | fCE: 0.07908655211287803  | fPC: 0.9756
Iteration: 196  | fCE: 0.07948187067007265  | fPC: 0.9755
Iteration: 197  | fCE: 0.07889656760617438  | fPC: 0.9767
Iteration: 198  | fCE: 0.07837482697781047  | fPC: 0.9769
Iteration: 199  | fCE: 0.07959018867337513  | fPC: 0.9762
_____ Training Results _____
Training Accuracy: 0.9951636363636364
Testing Accuracy: 0.9762
Training fCE: 0.026873617600201512
Testing fCE: 0.07959018867337513
```

Screenshot of training process and results

As the screenshot shows, the accuracy is 97.62% and fCE is 0.0796 for testing images, which is above the 95% and 0.15 described in the requirements.

Part Two: Mountains and valleys

Trajectory plot



Trajectory plot of weights

Explanations

In this plot, we could see that the sequence of weights (red dots) are decreasing along the weights grid (blue surface). The x and y coordinates is the transformed coordinates from pca.transform from the original 39760 features and the z coordinates is the cross-entropy loss function with respect to the weights.