DD2424 Deep Learning in Data Science

Assignment 1 Report

Yage Hao (yage@kth.se)

I. Introduction

In this assignment, we intend to classify images into multiple categories by constructing a one-layer neural network. Our dataset is CIFAR-10 which has already been split into training set, validation set and testing set. Each one includes data (observations) and labels (results).

II. Code Explanation

In general, we applied mini-batch gradient descent algorithm on the cross-entropy loss with L2 regularization term. Full detailed mathematical formulas are provided in the 'Background' part of the assignment document.

In detail, we first defined three function during the data preprocessing step. They are 'loadBatch', 'unpickle' and 'proprocess'.

Then, we defined a class named 'Classifier', in which we realized the neural network and the mini-batch gradient descent algorithm. In this class, we defined the following methods:

- evaluateClassifier: Compute p=softmax(s).
- computeCost: Compute cost function.
- computeAccuracy: Compute the accuracy of the network's predictions.
- computeGradients: Evaluate the gradients of the cost function w.r.t. W and b.
- computeGradientsNum: Numerically evaluate gradients.
- performPlot: Plot performance curve of training set and validation set.
- minibatchGD: Model training with mini-batch gradient descent.
- Visualization: visualize the weight matrix.

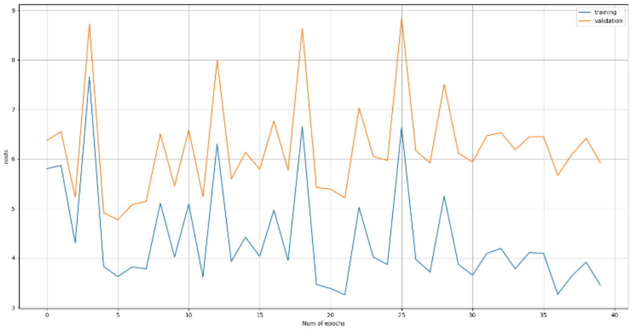
Specially, by comparing the results obtained by function 'computeGradients' and 'computeGradientsNum', we received the same derivatives (in the level of 10^{-6}) through numerical method and analytical method.

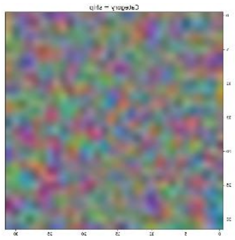
Finally, in the main function, we picked four sets of parameters for the model and output the training accuracy, validation accuracy, testing accuracy and plots of the weight matrix for different classes.

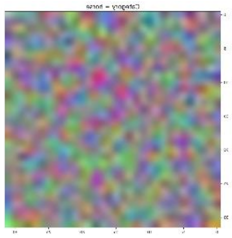
III. Results

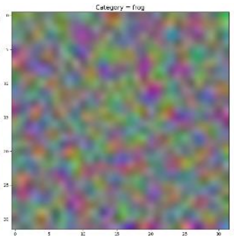
- Case 1: Lambda=0, n_epochs=40, n_batch=100, eta=.1

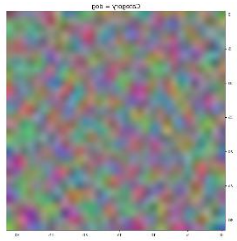
```
TRAINING MEAN: 0.50916
TRAINING STD: 0.03605676635529037
VALIDATION MEAN: 0.27493
VALIDATION STD: 0.007513461252977887
TESTING MEAN: 0.27874000000000004
TESTING STD: 0.007938160996099794
```

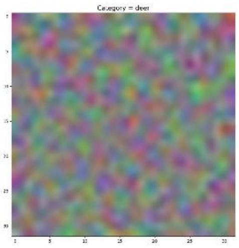


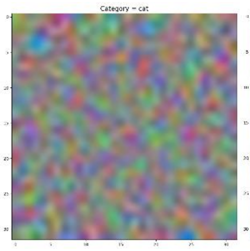


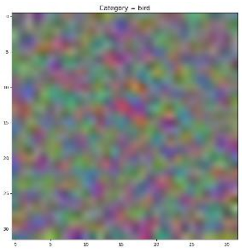


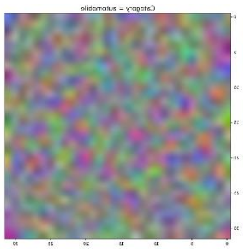


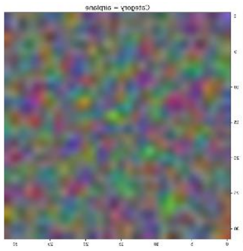


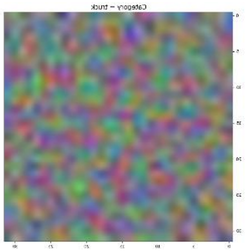






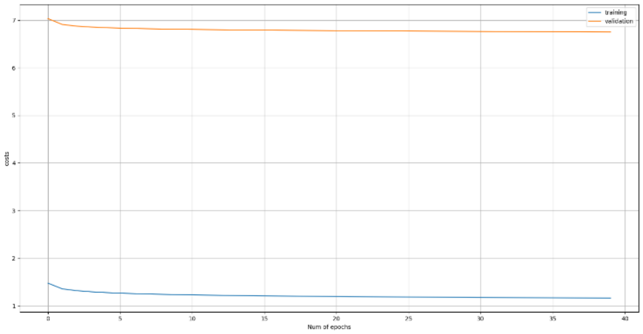


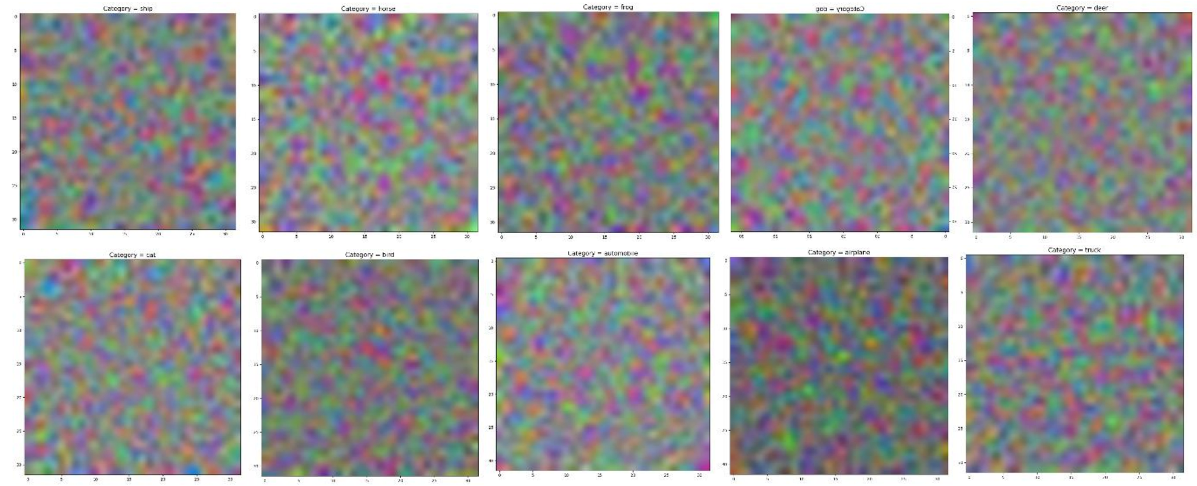




- Case 2: Lambda=0, n_epochs=40, n_batch=100, eta=.001

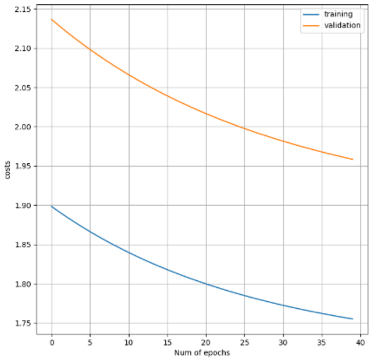
TRAINING MEAN: 0.7042900000000001
TRAINING STD: 0.007924449507694522
VALIDATION MEAN: 0.27424000000000004
VALIDATION STD: 0.0014691494137765598
TESTING MEAN: 0.28001999999999994
TESTING STD: 0.0017837040113202656

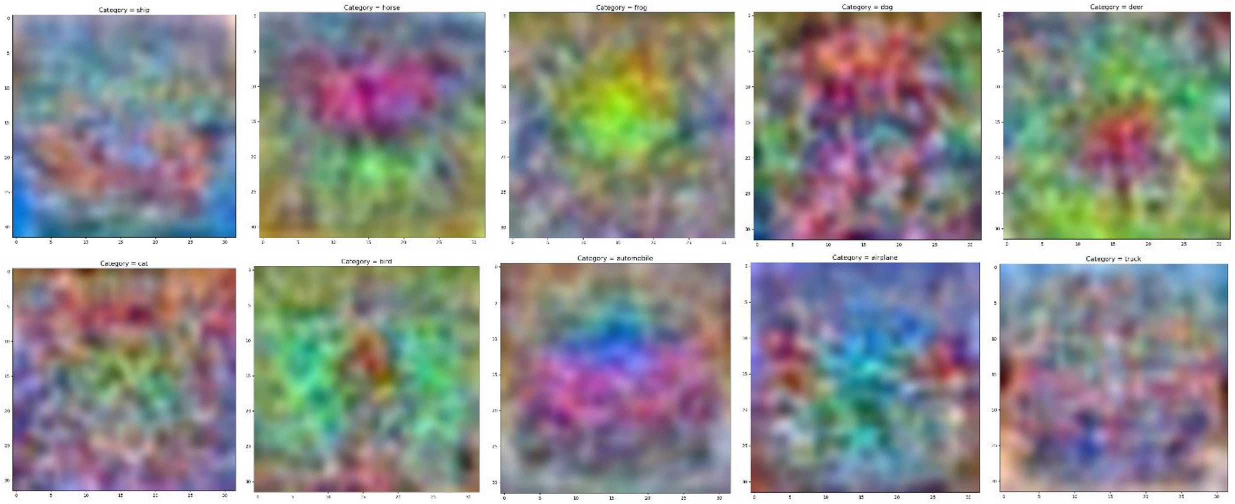




- Case 3: Lambda=.1, n_epochs=40, n_batch=100, eta=.001

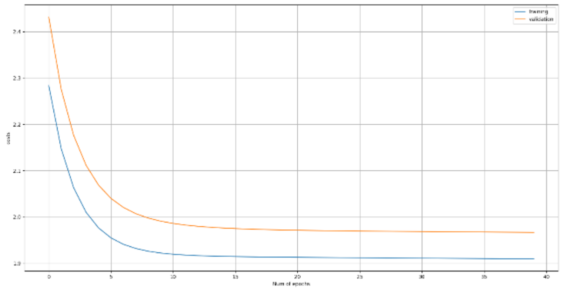
TRAINING MEAN: 0.5267999999999999
TRAINING STD: 0.08386338891316045
VALIDATION MEAN: 0.36319
VALIDATION STD: 0.03479555862462909
TESTING MEAN: 0.36891
TESTING STD: 0.03292337923117857

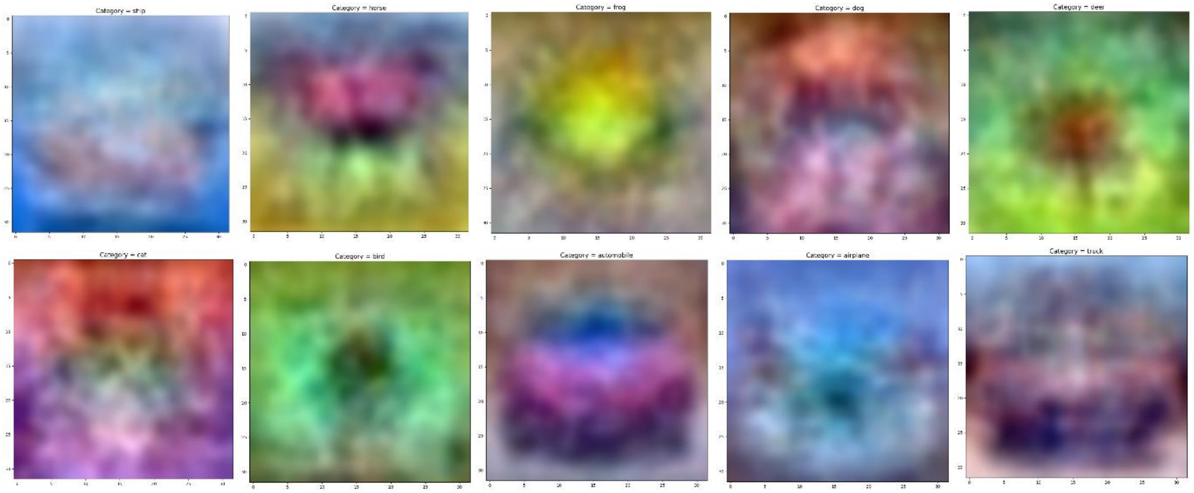




- Case 4: Lambda=1, n_epochs=40, n_batch=100, eta=.001

TRAINING MEAN: 0.39948
TRAINING STD: 0.0008749857141690874
VALIDATION MEAN: 0.36696
VALIDATION STD: 0.0014582180906846521
TESTING MEAN: 0.37781
TESTING STD: 0.002151023012429204





IV. Conclusion

The most important conclusion is that the performance of using a one-layer neural network with mini-batch gradient descent algorithm to classify multiple categorical images is not good as shown in the test accuracy with different parameters.

Comparing the results of Case 3 and Case 4, we noticed that increasing the regularization parameter lambda led to a poor classification power, while we also know that the overfitting problem can be avoided through adding the regularization term.

Comparing the results of Case 1 and Case 2, we noticed that reducing the learning rate eta led to a stronger classification power. Especially by observing the unstable cost plot in Case 1, we find that a large learning rate can hardly descent to a local optimum point.