DD2424 Deep Learning in Data Science - Assignment 1

Bas Straathof — btstr@kth.se

April 5, 2019

1 Introduction

The functions that were required for this assignment were all successfully implemented in Python, including the one to correctly compute the gradient analytically (`ComputeGradient`). A separate `TextMethods` class was established for testing several aspects of the data set and the methods of the `Classifier` class (with a random seed of `numpy.random.seed(0)`). A custom test case was created to test the similarity of arrays resulting from computing the gradients numerically and analytically. Using the `assert_almost_equal` function from the `numpy.testing` library, it was found that the arrays were equal up to 4 decimals.

2 Classification results

See Figures 1-4 for the cost plots of the training and validation sets for the different parameter settings (with a random seed of `numpy.random.seed(0)`). From these plots we can tell that when the learning rate η is too big, such as is the case in Figure 1, the training of the model is unstable. When the value of the regularization term is too high, such as is the case in Figure 4, the model will not be able to learn much from the data. The following mean classification results (with $\pm$ the standard deviation) were obtained after running the models 10 times:

1 Settings: **lambda=0, n epochs=40, n batch=100, eta=.1**

- The accuracy on the training set is: 0.2932 $\pm$ 0.0479
- The accuracy on the validation set is: 0.2297 $\pm$ 0.0332
- The accuracy on the testing set is: 0.2309 $\pm$ 0.0330

2 Settings: **lambda=0, n epochs=40, n batch=100, eta=.01**

- The accuracy on the training set is: 0.5879 $\pm$ 0.0103
- The accuracy on the validation set is: 0.3128 $\pm$ 0.0024
- The accuracy on the testing set is: 0.3143 $\pm$ 0.0033

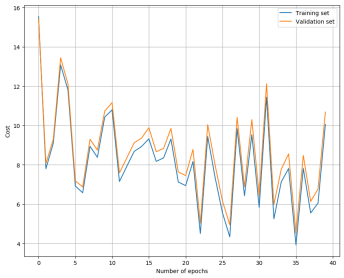

Figure 1: Graph of the total cost on the training and validation data for the parameter settings `lambda=0, n epochs=40, n batch=100, eta=.1`.

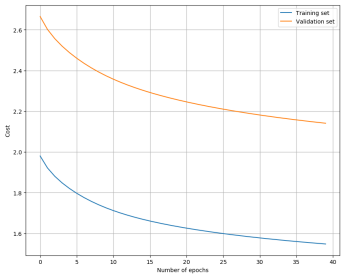

Figure 2: Graph of the total cost on the training and validation data for the parameter settings `lambda=0, n epochs=40, n batch=100, eta=.01`.

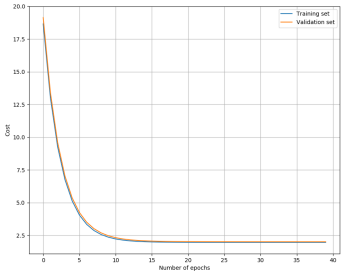

Figure 3: Graph of the total cost on the training and validation data for the parameter settings `lambda=.1, n epochs=40, n batch=100, eta=.01`.

3 Settings: **lambda=.1, n epochs=40, n batch=100, eta=.01**

- The accuracy on the training set is: 0.3401 $\pm$ 0.0022
- The accuracy on the validation set is: 0.3180 $\pm$ 0.0021
- The accuracy on the testing set is: 0.3297 $\pm$ 0.0022

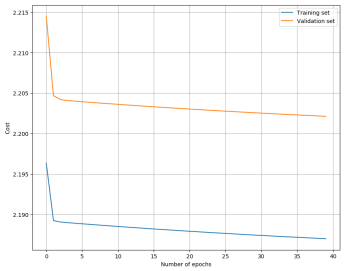

Figure 4: Graph of the total cost on the training and validation data for the parameter settings `lambda=1, n epochs=40, n batch=100, eta=.01`.

4 Settings: **lambda=1, n epochs=40, n batch=100, eta=.01**

- The accuracy on the training set is: 0.2274 ± 0.0010
- The accuracy on the validation set is: 0.2166 ± 0.0008
- The accuracy on the testing set is: 0.2212 ± 0.0009

See Figures 5-8 to see what images the various models have learned. As can be observed, the more regularization is applied, the smoother the learned images become.

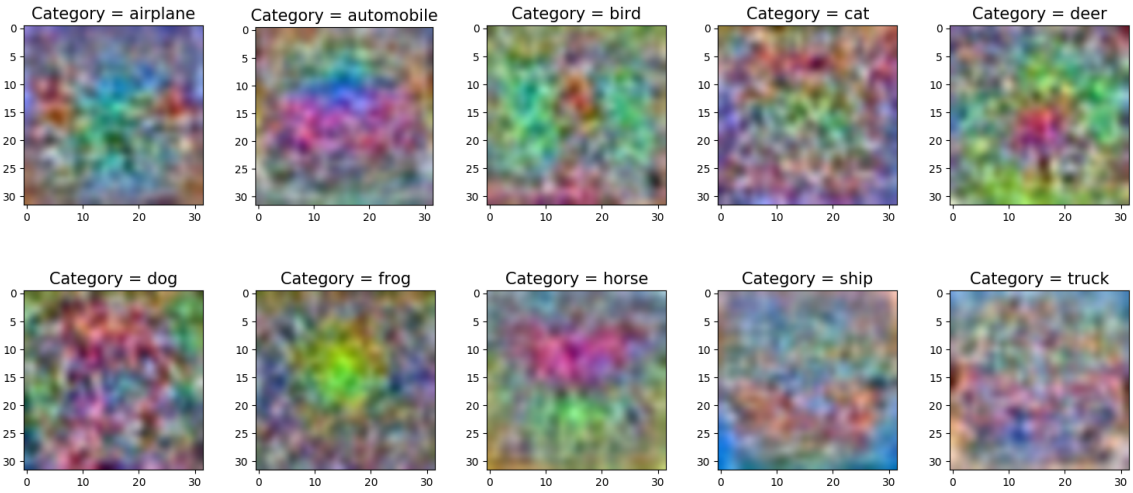

Figure 5: Images representing the learnt weight matrix after the completion of training for the parameter settings `lambda=0, n epochs=40, n batch=100, eta=.1`.

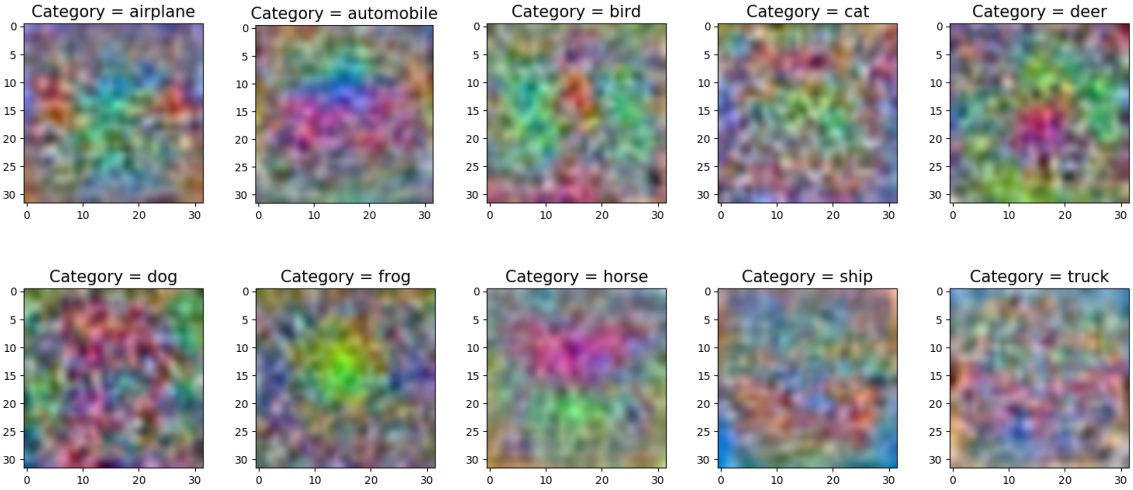

Figure 6: Images representing the learnt weight matrix after the completion of training for the parameter settings `lambda=0, n epochs=40, n batch=100, eta=.01`.

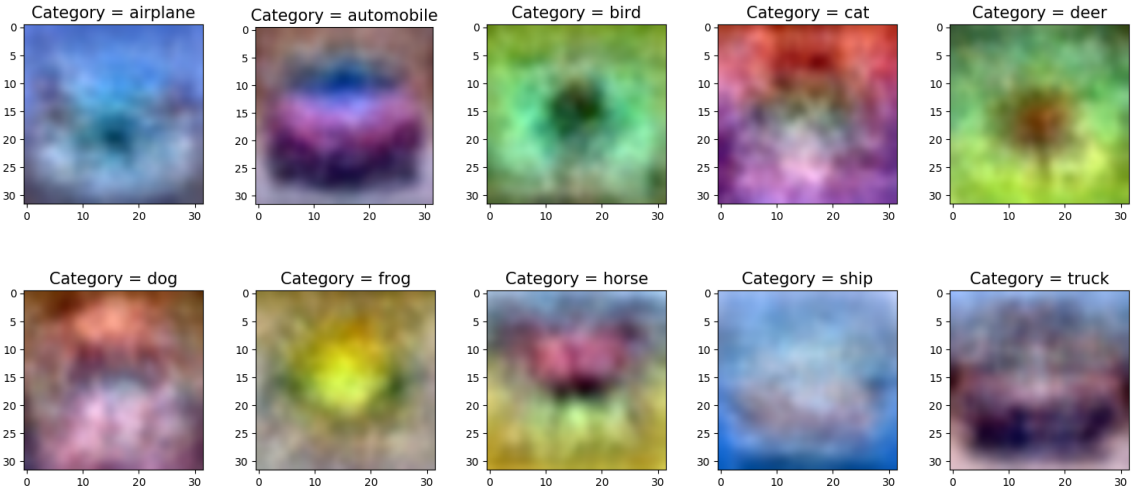

Figure 7: Images representing the learnt weight matrix after the completion of training for the parameter settings `lambda=.1, n epochs=40, n batch=100, eta=.01`.

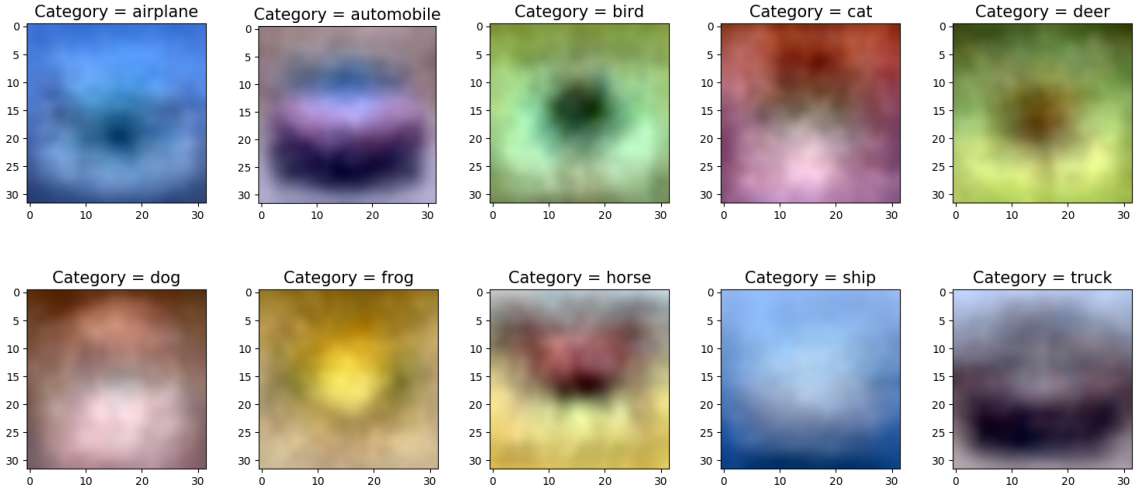

Figure 8: Images representing the learnt weight matrix after the completion of training for the parameter settings `lambda=1, n epochs=40, n batch=100, eta=.01`.