Assignment 1 Report

Max Dauber (dauber@kth.se)

DD2424 Deep Learning in Data Science - Spring 2020

Introduction

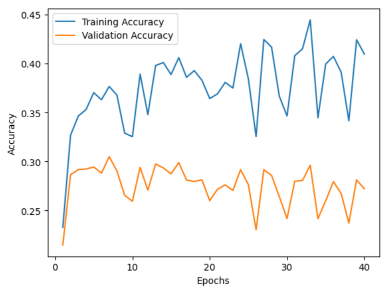
A 1-layer neural network was implemented from scratch using Python. The neural network is trained using MiniBatch gradient descent which is applied to a cost function that computes cross-entropy loss of the classifier applied to the labeled training data and an L2 regularization term related to the weight matrix. The dataset CIFAR10 is used, which consists of 10,000 32x32 images corresponding to 10 classes of objects.

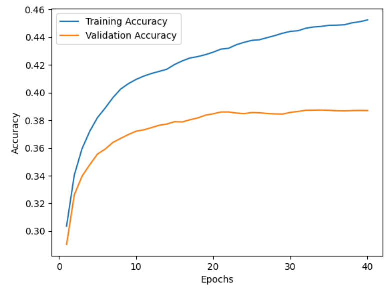
Training and Gradient Calculations

In Table 1 one can see the final classification accuracy of the model after training for 40 epochs, with a minibatch size of 100. Figure 1 and Figure 2 depict the evolution of the accuracy and loss across training epochs in the different setups of regularization (λ) and learning rate (η). The correctness of the analytically calculated gradients was checked using two numerical gradient calculation methods for comparison (ComputeGradsNum & ComputeGradsNumSlow). First, the gradients were checked using the finite difference method and then using the centered difference method which is slower but more accurate. In both comparisons, it was clear that the analytically computed gradients were very similar to the numerical ones (within 10^{-6}), which gives strong evidence that the gradient computation is mathematically correct. Below are the results for the different setups:

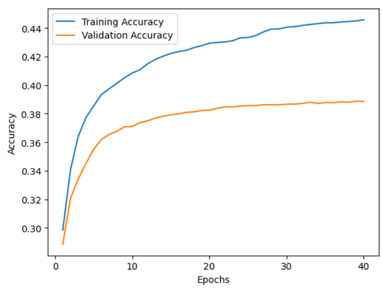
Setup	Train Accuracy	Test Accuracy
λ=0, η=0.1	40.97%	27.71%
λ=0, η=0.001	45.25%	38.90%
λ=0.1, η=0.001	44.58%	39.00%
λ=1, η=0.001	39.92%	37.08%

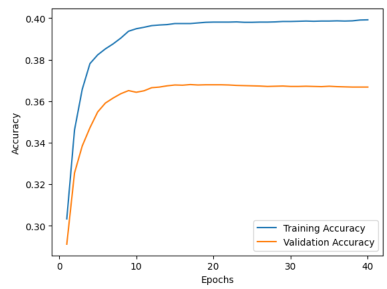
Table 1: Accuracy results on train and test set with different setups





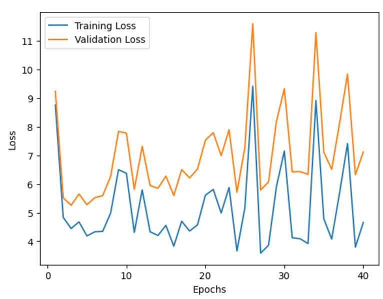
λ=0, η=0.1 λ=0, η=0.001

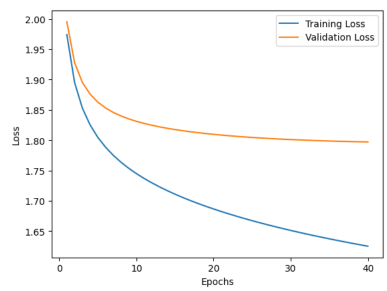




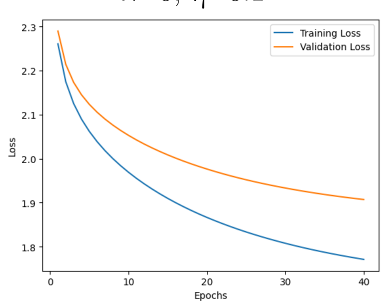
λ=0.1, η=0.001 λ=1, η=0.001

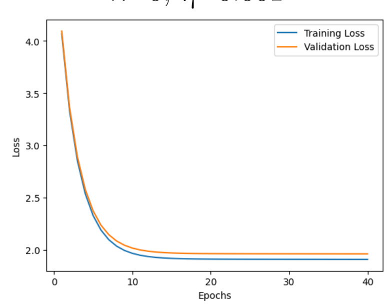
Figure 1: Training/Validation Accuracy Calculations





λ=0, η=0.1 λ=0, η=0.001





λ=0.1, η=0.001 λ=1, η=0.001

Figure 2: Training/Validation Loss Calculations

Learning Rate

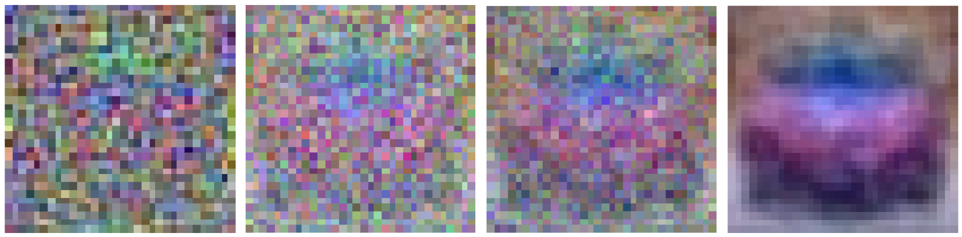
Three setups with learning rate (referred to as η) were tested with value 0.001, and one with value 0.1. In Figure 1 and Figure 2 it is clear that the learning rate of 0.1 is too high. The evolution and stability of the accuracy and loss across epochs is very poor, which is probably due to the fact that the the model will be overshooting or undershooting the minima. A learning rate of 0.001 causes much smoother behavior in the evolution of the loss and the accuracy, so this value works better in this context. More values should be tested if we wanted to determine the optimal learning rate, which would allow us to find more accurate minima and possibly speed up the training.

Regularization

Three different values were tested for the L2 regularization parameter λ: λ=0, λ=0.1 and λ=1. In Figure 2 it can be seen that the accuracy of the training set is generally better than the accuracy of the validation set. This implies that the model is overfitting when there is no regularization, which is very likely. In Figure 1, in the evolution of the loss, this is not as clearly visible. In Figure 2 it can also be seen that in the setup with λ=1, the train accuracy is well below the validation accuracy. Also, it can be seen in Table 1 that the accuracy for this setup is relatively low for both train and test set. This implies that the model is underfitting, so this value of $\lambda$is probably too high. λ=0.1 seems to be the best setup, but more values should be tested before the optimal value of $\lambda$is selected.

Weight Results

Figure 3 shows the progression of the visualization of the weight matrix after the training on each set of hyperparameters for one of the classes, corresponding to an automobile. In the two rightmost setups the car can be recognized clearly: the model has learned a good representation of this class. However, this does not completely match the performance reported in Table 1 for those configurations. For example, the setup with λ=1 provides a relatively good picture of a car, but has a lower accuracy. In Figure 4,5,6,7 you can see the weight matrix visualizations for all 10 classes of the CIFAR10 dataset across the different hyperparameter configurations.



λ=0, η=0.1 λ=0, η=0.001 λ=0.1, η=0.001 λ=1, η=0.001

Figure 3: Weight Visualizations for CIFAR10 Automobile Class

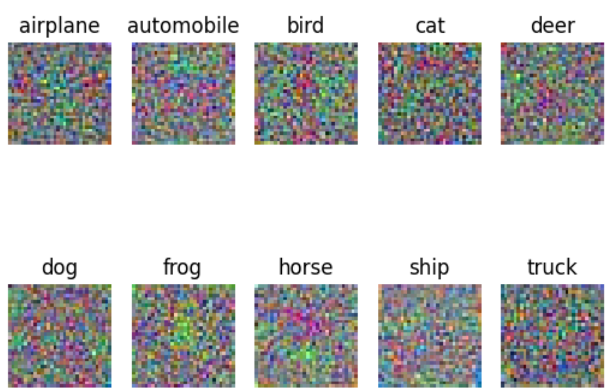


Figure 4: Weight Visualizations for λ=0, η=0.1

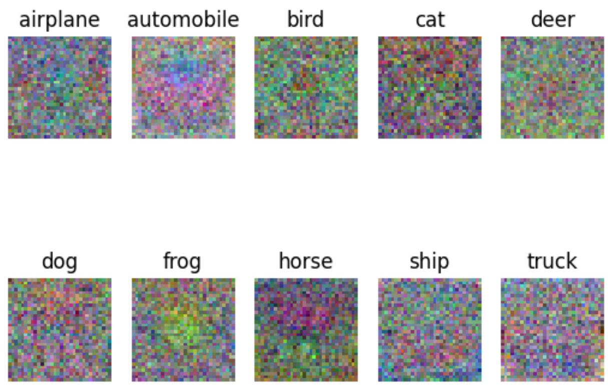


Figure 5: Weight Visualizations for λ=0, η=0.001

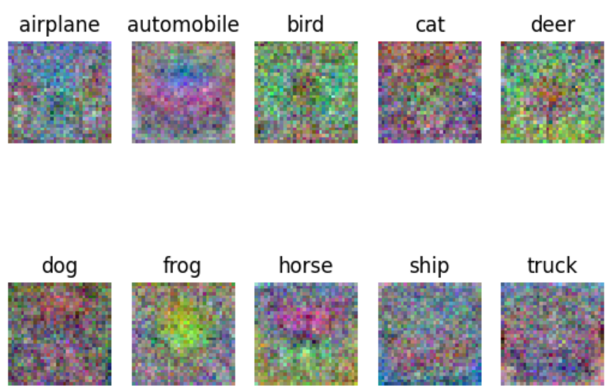


Figure 6: Weight Visualizations for λ=0.1, η=0.001

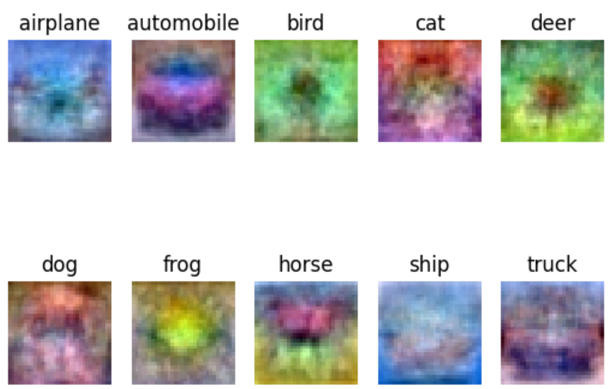


Figure 7: Weight Visualizations for λ=1, η=0.001