DD2424 Deep Learning Assignment 1

Mikael Ljung (milju@kth.se)
940312–7435

April 2020

1. Introduction

In this paper a one-layer network with multiple outputs was trained to classify images from the CIFAR10-dataset. The CIFAR100 consists of 60 000 32x32 color images in 10 classes, with 6000 images per class.

2. Methods

The one-layer network was built from scratch in Python using mini-batch gradient descent on a cost function that calculated the cross-entropy loss. To make sure that the analytical gradient was calculated correctly it was compared to a numerical calculated gradient. The network was evaluated on different values on the models two parameters: the learning rate and the regularization term (lambda).

3. Results

3.1. Gradients

Batchsize	10	100
$\sum gn_W$	6.66136656946037e-09	4.1744385725905886e-08
$\sum gnb$	2.220446049250313e-10	-4.440892098500626e-10
$\sum ga_W$	0	-4.440892098500626e-10
$\sum gab$	4.163336342344337e-17	-2.42861286636753e-17
$\sum \varepsilon w$	8.100611964785012e-10	2.137742263423408e-09
$\sum \varepsilon b$	1.2242895362562212e-09	2.252276231236979e-09

3.2. Parameter settings

The following parameter settings were applied:

	Test 1	Test 2	Test 3	Test 4
λ	0	0	0.1	1
epochs	40	40	40	40
batch	100	100	100	100
eta	0.1	0,001	0,001	0,001

3.3. Losses and costs

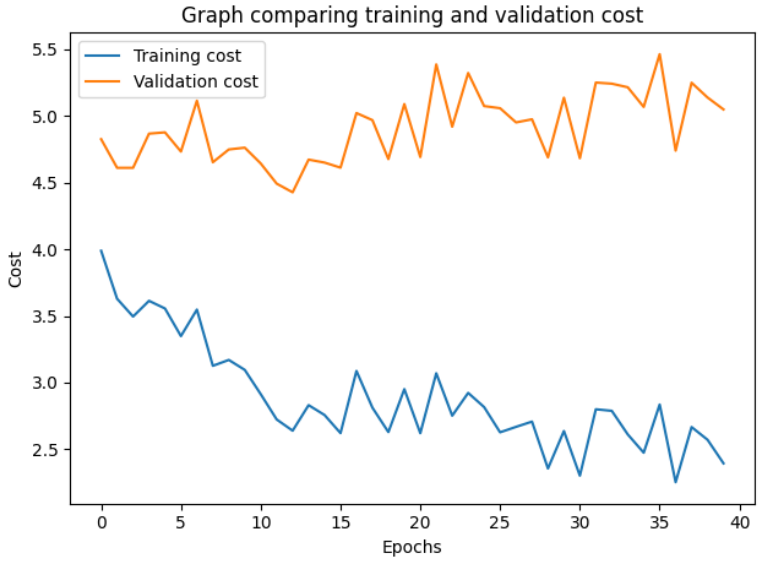


Figure 1: Learning rate = 0.1 & λ =0

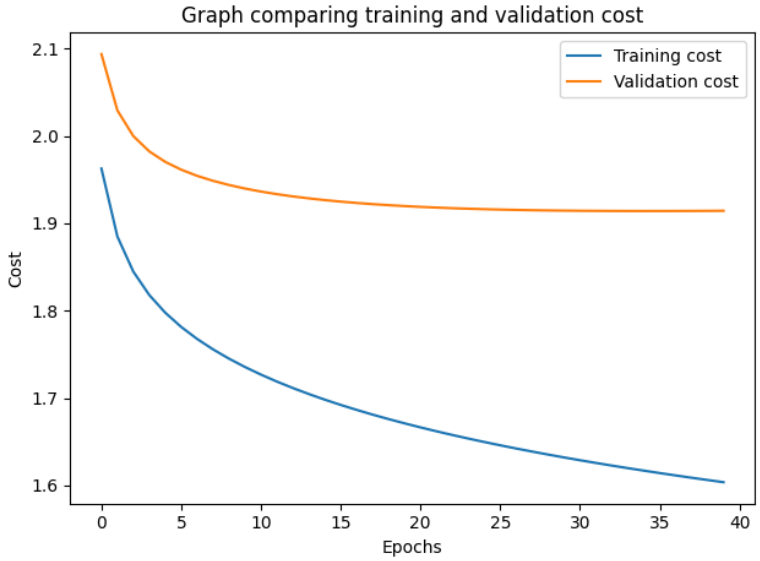


Figure 2: Learning rate = 0,001 & λ =0

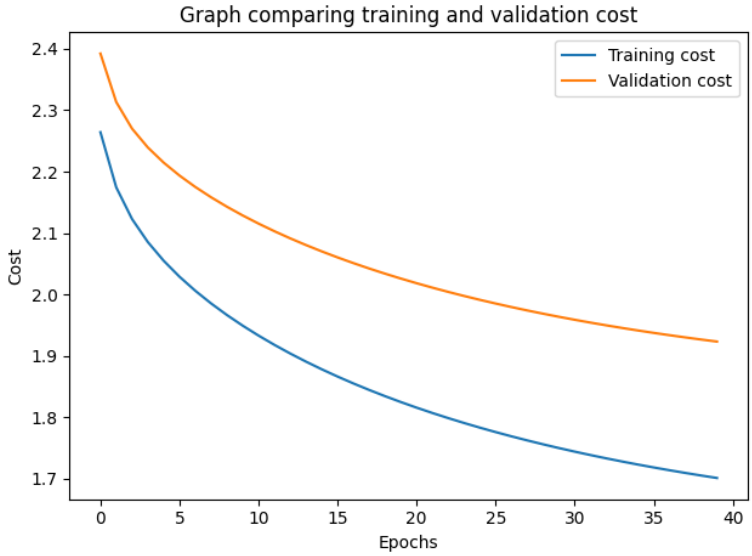


Figure 3: Learning rate = 0,001 & λ =0.1

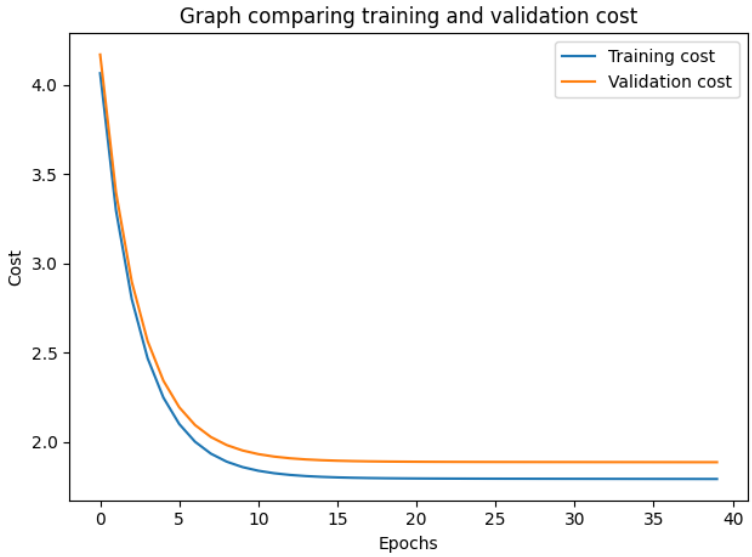


Figure 4: Learning rate = 0,001 & λ =1

3.4. Accuracies

	Accuracy on training	Accuracy on validation	Accuracy on test
Test 1	46,39%	27,53%	28,6%
Test 2	49,31%	38,36%	38,45%
Test 3	46,13%	38,91%	39,43%
Test 4	39,88%	36,48%	37,78%

3.5. Weight matrix

By looking at the four weight matrices we can see that the classes start to show up when the parameters changes.

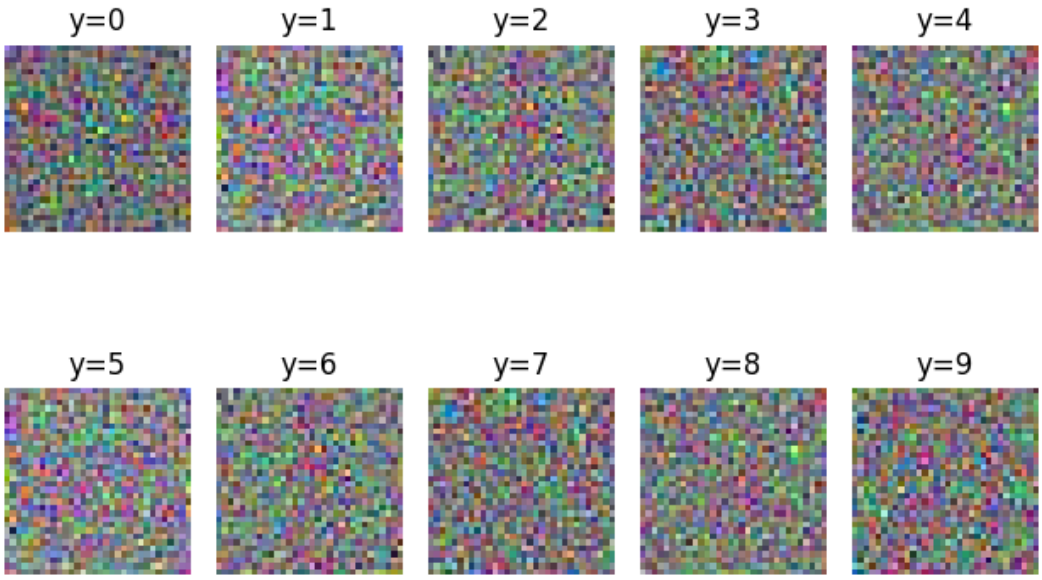


Figure 5: Weight matrix on test 1

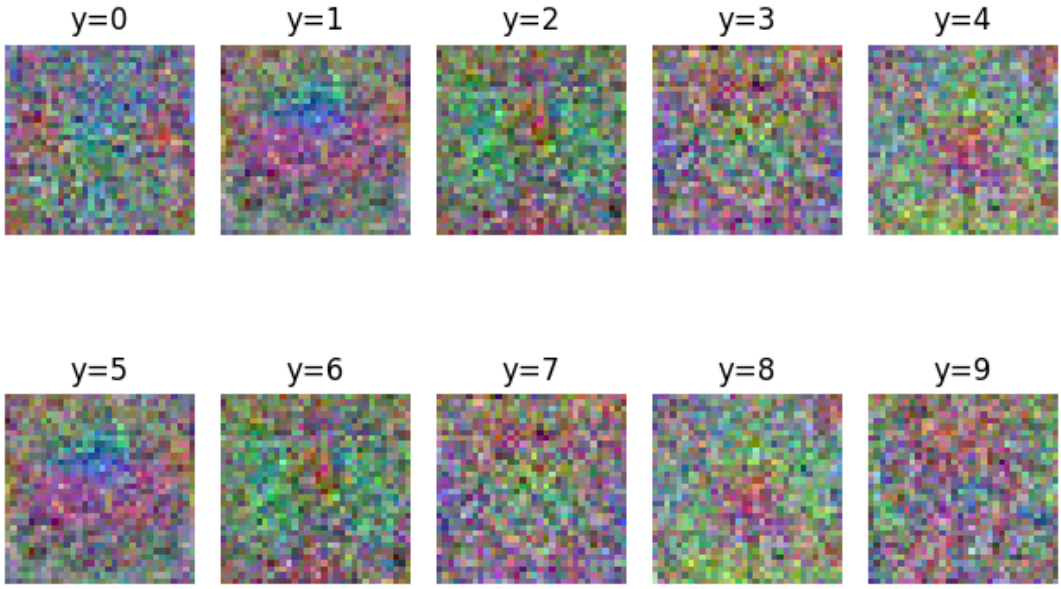
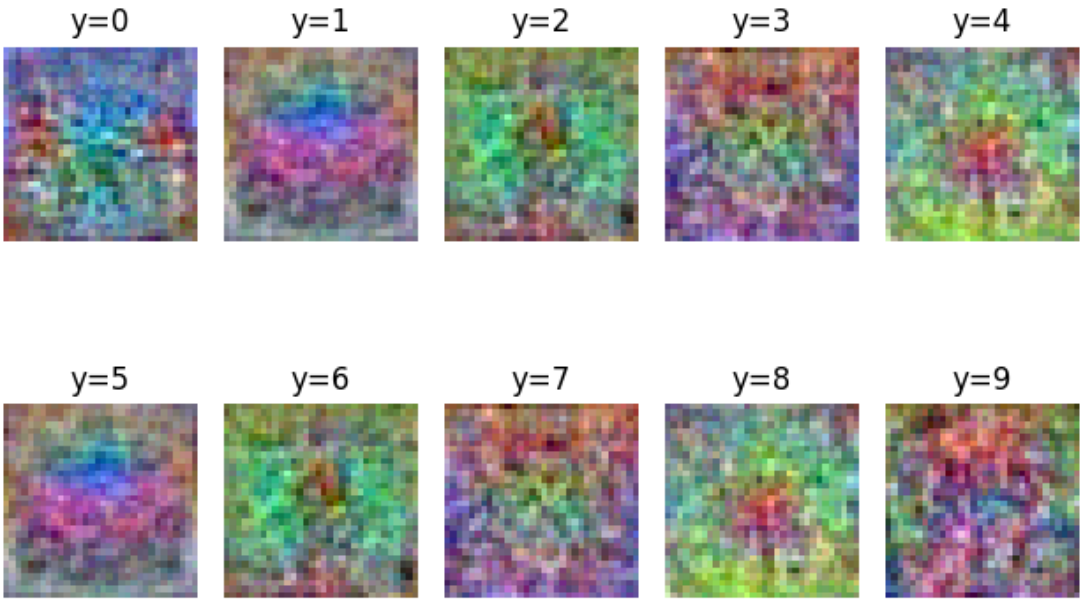
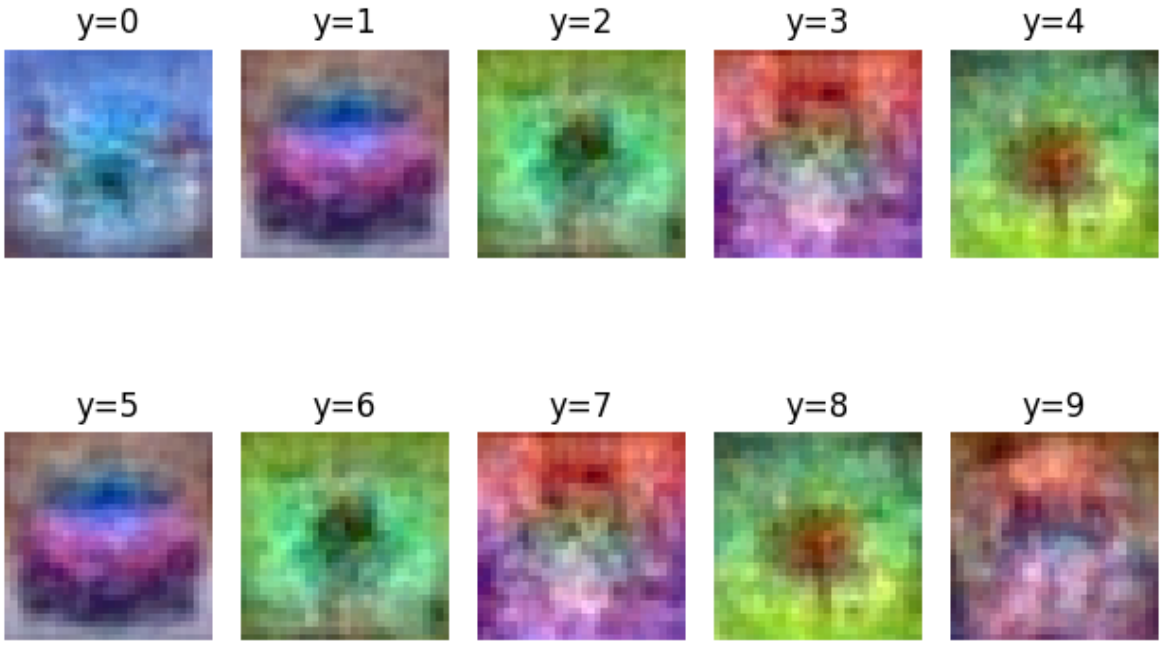


Figure 6: Weight matrix on test 2



Figur 7: Weight matrix on test 3



Figur 8: Weight matrix on test 4

4. Conclusion

When choosing the parameters for the model one has to think of the variance-bias tradeoff. By increasing lambda, the model scores lower on the training data and higher on the validation and the test data. This is because this parameter makes the model less complex, meaning it will not overfit as much on the training data. The second parameter, the learning rate, decides how fast the gradient will converge. By having a higher learning rate, the model will converge quicker, but instead risks not converging towards the local minima. It is therefore important to choose the right values on both the regularization term and the learning rate when building your machine learning model.