Deep learning - Assignment 1

Gabriel Carrizo 9205043152

April 2017

Gradient function

The following is the function i used to compute the gradient:

```
%mygrad
function [grad_W, grad_b]=MyComputeGrads(X,Y,W,b,lambda)
N = size(X,2);
P = EvaluateClassifier(X,W,b);
dldb = zeros(size(b));
dldW = zeros(size(W));
for i=1:N
    g = (-Y(:,i)'/(Y(:,i)'*P(:,i)))*(diag(P(:,i))-(P(:,i)
        *P(:,i)'));
    dldb = (dldb + g');
    dldW = (dldW + g'*X(:,i)');
end

grad_W = dldW/N + 2*lambda*W;
grad_b = dldb/N;
end
```

With this function my gradient errors are at the worst $\approx 10^{-8}$ when compared to the slow numerical gradient function and 10^{-10}

1 First settings:

lambda=0, n_epochs=40, n_batch=100, eta=.1

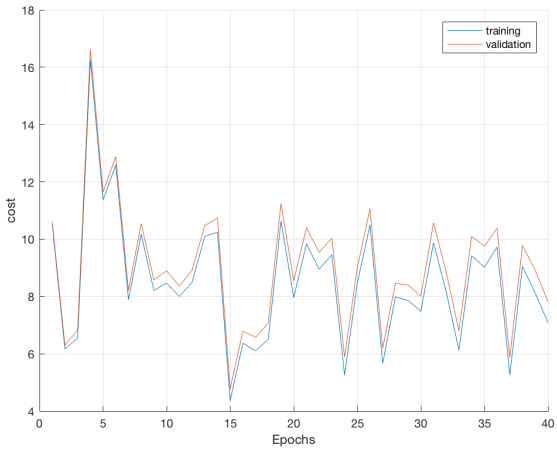

Figure 1: Caption

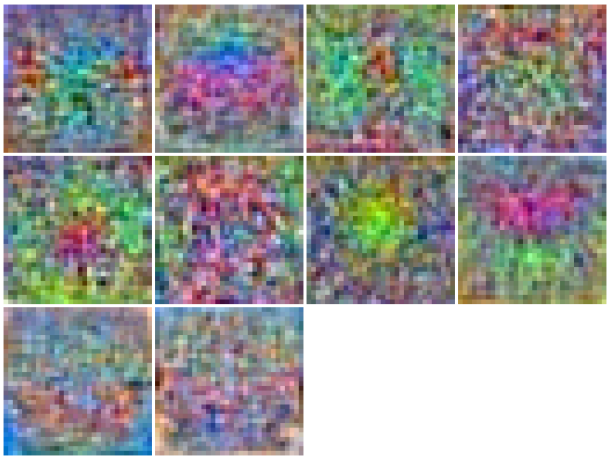

Figure 2: Cost function of training and validation data sets

Accuracy on test set: 0.2446

2 Second settings:

lambda=0, n epochs=40, n batch=100, eta=.01

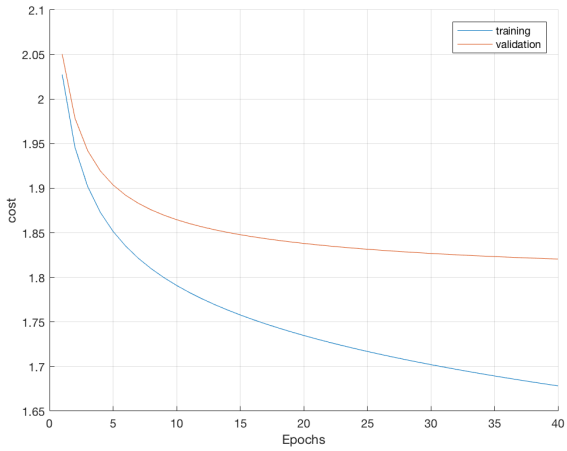

Figure 3: Caption

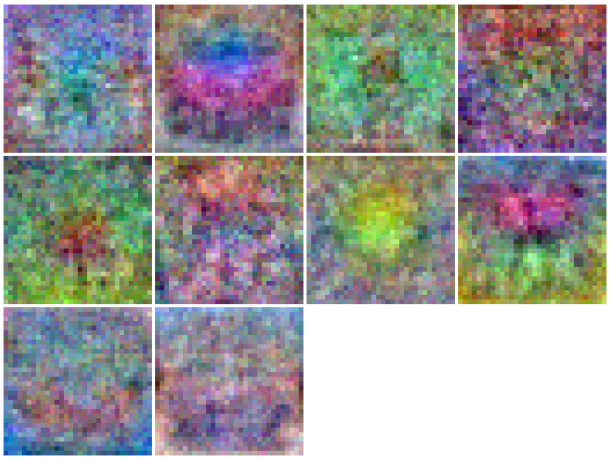

Figure 4: Cost function of training and validation data sets

Accuracy on test set: 0.3665

3 Third settings

lambda=.1, n_epochs=40, n_batch=100, eta=.01

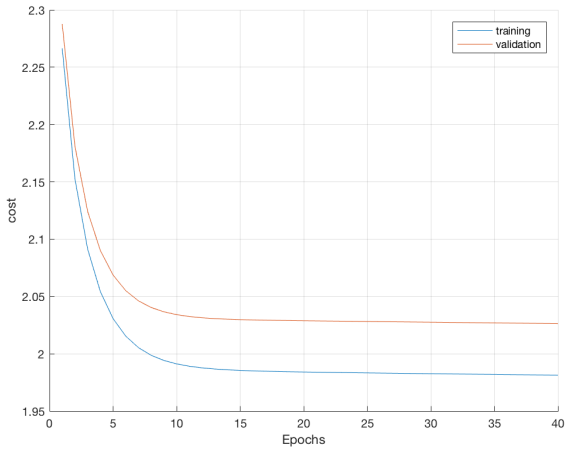

Figure 5: Caption

Figure 6: Cost function of training and validation data sets

Accuracy on test set: 0.3337

4 Fourth settings

lambda=1, n_epochs=40, n_batch=100, eta=.01

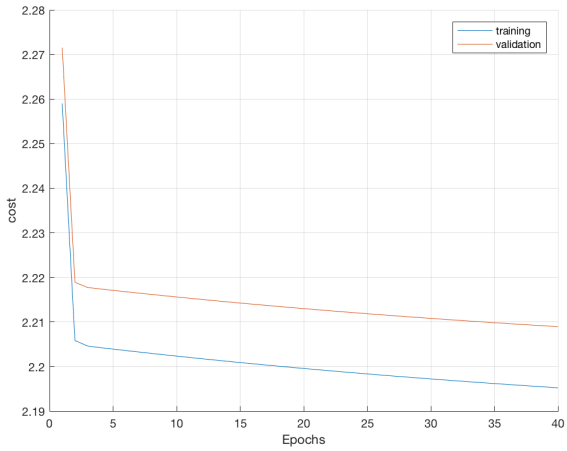

Figure 7: Caption

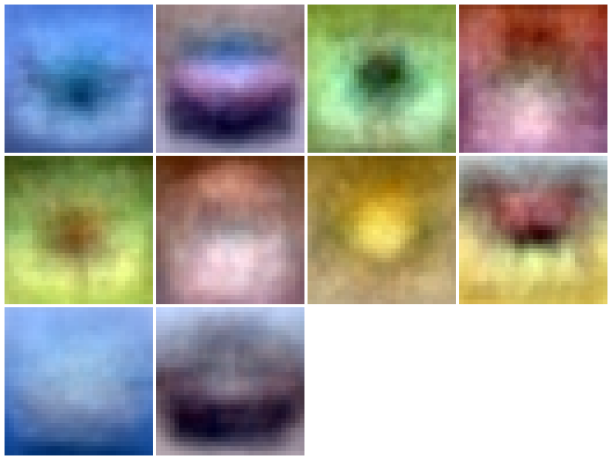

Figure 8: Cost function of training and validation data sets

Accuracy on test set: 0.2192

5 Comments on results:

In theory, the regularization term reduces complexity of the classifier and since this is a fairly low complexity classifier the results are worse with a regularisation term. This can be observed in Figure 9.
When increasing eta we can see that the gradient overshoots past the local minimum and keeps doing this which means it will not necessarily converge. Lower eta might not find minimum if we do not run the algorithm for enough epochs.

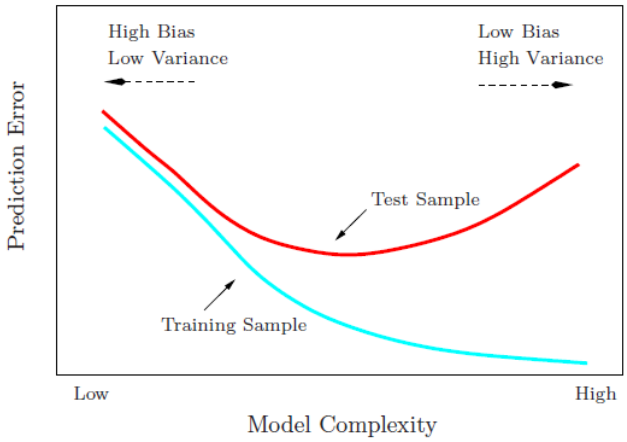

Figure 9: How prediction error depends on the complexity of the learner.