Assignment 2 - BONUS - Deep Learning

Alexandra Hotti

March 2019

1 Introduction

In the bonus part of assignment 2 the performance of the 2-layer neural network from the basic part is attempted to be optimized via different measures such as augmenting the training data and applying drop out.

2 Network Architecture

The following common hyper parameters were used in section REF:

$m = 50, \lambda = 0.000648717, trainingdatasize = 49000, batchsize = 100, n_s = 980, epochs = 12, cycles = 3$

3 Increasing the number of hidden nodes and the amount of regularization

Using the parameter setting in 2, except for λ and m which vary, a network was trained were the number of nodes as well as the regularization was increased. The results can be found in table 1.

Table 1: Test Accuracy for increasing regularization and an increasing number of nodes.

λ	Hidden nodes (m)	Test set Accuracy
$\lambda_{basic} = 0.000648717$	50	51.33 %
λ_{basic}	75	53.05 %
$\lambda_{basic} \cdot 1.5$	75	53.29 %
$\lambda_{basic} \cdot 1.6$	75	53.32 %
$\lambda_{basic} \cdot 1.75$	75	52.81 %
λ_{basic}	100	53.48 %
$\lambda_{basic} \cdot 1.5$	100	53.71 %
$\lambda_{basic} \cdot 1.6$	100	53.58 %
$\lambda_{basic} \cdot 1.75$	100	54.03 %
λ_{basic}	150	53.76 %
$\lambda_{basic} \cdot 1.5$	150	54.43 %
$\lambda_{basic} \cdot 1.6$	150	54.05 %
$\lambda_{basic} \cdot 1.75$	150	**54.52 %**

As can be seen in the table, increasing the number of hidden nodes and the amount of regularization had a relatively large impact on the final test set accuracy. The difference between the initial baseline case (see row 1 in table 1) and the final optimal networks (see the last row in table 1) test set accuracy was 3.19 percentage points.

4 Augmenting the training data with a random jitter

The training data was augmented by applying a random jitter to each image in a mini-batch before doing the forward and backward pass.

Again, this was performed on the hyper-parameters in section 2.

Table 2: Augmenting the training data with random jitter.

Uniform jitter	Test set Accuracy
-	51.33 %
+/- 2.0 %	50.94 %
+/- 1.5 %	51.19 %
+/- 1.0 %	51.58 %
+/- 0.5 %	**51.9 %**
+/- 0.25 %	51.83 %

Applying a random jitter to the training data did slightly improve the final test accuracy.

5 Applying drop out

Inverted dropout was implemented during the training of the network using the following implementation:

$$
\begin{aligned}
x^l &= max(0, W_l \cdot X^{l-1} + b_l) \\
u^l &= (rand(size(x^l)) < p)/p \\
x^l &= x^l \odot u^l
\end{aligned}
$$

Two different probabilities within the range (0, 1) where used for an increasing number of hidden nodes.

Table 3: Applying random Drop Out during training.

Hidden Nodes (m)	Probability (p)	Test set Accuracy
50	0.5	40.26 %
75	0.5	42.64 %
100	0.5	44.67 %
150	0.5	46.51 %
200	0.5	47.99 %
250	0.5	48.90 %
100	0.3	39.60 %
150	0.3	41.51 %
200	0.3	42.81 %
250	0.3	44.09 %

Implementing drop out did not improve the results on the final test accuracy, instead it dropped a bit.

6 Best performing Network

The best performing network had no drop out and no jitter applied during training. Using a large number of nodes was what had the most impact on the final test accuracy. The best performance was achieved by a network with 150 hidden nodes. The parameters of this network was:

$$m = 150, \lambda = 0.00113525475, trainingdatasize = 49000, batchsize = 100, n_s = 980, epochs = 12, cycles = 3$$