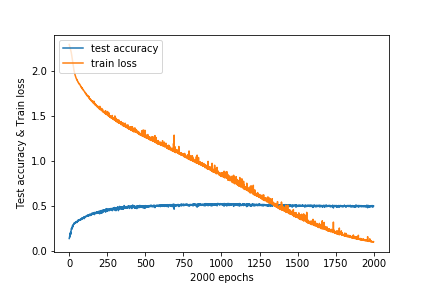
* 1. **Part A**
  2. **1. Train the network by using mini-batch gradient descent learning. Set batch size =128, and learning rate 𝛼=0.001. Images should be scaled.**
  3. **a. Plot the training cost and the test accuracy against learning epochs.**



*Figure 1a*

Test accuracy reached the maximum of around 0.5 at epoch 500. Afterwards, the loss is decreasing but accuracy stays the same, suggesting overfitting. Subsequent training will be done with 750 epochs.

* 1. **b. For any two test patterns, plot the feature maps at both convolution layers (𝐶1 and 𝐶2) and pooling layers (𝑆1 and 𝑆2) along with the test patterns.**

|  |  |  |
| --- | --- | --- |
|  | Test Pattern 1 | Test Pattern 2 |
| Raw |  |  |
| C1 |  |  |
| S1 |  |  |
| C2 |  |  |
| S2 |  |  |

**2. Using a grid search, find the optimal numbers of feature maps for part (1) at the convolution layers. Use the test accuracy to determine the optimal number of feature maps.**

Searching these grids:

C1 = {40,50,60}

C2 = {50,60,70}

|  |  |  |  |
| --- | --- | --- | --- |
| Test Accuracy | C1:40 | C1:50 | C1:60 |
| C2:50 |  |  |  |
| C2:60 |  | 0.516 |  |
| C2:70 |  |  |  |

From the table above, optimal number of feature maps {C1, C2} is { }.

**3. Using the optimal number of filters found in part (2), train the network by:**

**a. Adding the momentum term with momentum 𝛾=0.1.**

**b. Using RMSProp algorithm for learning**

**c. Using Adam optimizer for learning**

**d. Adding dropout to the layers**

**Plot the training costs and test accuracies against epochs for each case.**

**4. Compare the accuracies of all the models from parts (1) - (3) and discuss their performances.**