

Q2(a) Code (Digits, maxit=10,000, n=1000)

```

1  % initialize parameters
2  - maxit=10000;
3  - alpha=0.01;
4  - n=1000;
5  - n_test=10000;
6  - out_interval=1000;
7
8  % load the training and test images + labels
9  - trainimages=loadMNISTImagesAsVectors('Digits_train-images.idx3-ubyte');
10 - testimages=loadMNISTImagesAsVectors('Digits_t10k-images.idx3-ubyte');
11 - trainlabels=loadMNISTLabels('Digits_train-labels.idx1-ubyte');
12 - testlabels=loadMNISTLabels('Digits_t10k-labels.idx1-ubyte');
13
14 % generate 2 arrays of one-hot label vectors
15 - train_y_actual=(trainlabels==1:10)';
16 - test_y_actual=(testlabels==1:10)';
17
18 % randomly initialize W's and b's
19 - W1=randn(128,784)*0.3;
20 - W2=randn(32,128)*0.3;
21 - W3=randn(10,32)*0.3;
22 - b1=randn(128,1)*0.3;
23 - b2=randn(32,1)*0.3;
24 - b3=rand(10,1)*0.3;
25
26 % before SGD training
27 - before_accuracy=0;
28 - for i=1:n
29 -     h1=phi_ReLU(W1*(trainimages(:,i))+b1);
30 -     h2=phi_ReLU(W2*h1+b2);
31 -     yhat=phi_Softmax(W3*h2+b3);           % probability vector
32 -     yhat_label=find(yhat==max(yhat));    % find label
33 -     % loss and accuracy
34 -     if yhat_label==trainlabels(i)
35 -         before_accuracy = before_accuracy+1;
36 -     end
37 -     Fi_before(i)=norm(yhat-train_y_actual(:,i))^2;
38 - end
39
40 - F(1)=sum(Fi_before)/n;
41 - percent_accurate(1)=(before_accuracy)/n;
42 - fprintf(['Before SGD Training \nLoss: ' num2str(F(1)) '\nAccuracy: '...
43 -         num2str(percent_accurate(1)) '\n\n'])
44
45 % SGD training
46 - SGD_accuracy=0;
47 - for it=1:maxit
48 -     i=randi(n);
49 -     x_i=trainimages(:,i);
50 -     % forward pass for image i
51 -     [z1,h1,z2,h2,z3,y_i]=forward_pass(W1,W2,W3,b1,b2,b3,x_i);
52 -     y_i_label=find(y_i==max(y_i));
53 -     if y_i_label==trainlabels(i)
54 -         SGD_accuracy = SGD_accuracy+1;
55 -     end

```

```

56 - train_y_actual_i=train_y_actual(:,i);
57
58 % back-propagate to compute gradients of f_i
59 - [grad_W1_f,grad_W2_f,grad_W3_f,grad_b1_f,grad_b2_f,grad_b3_f]=...
60     back_prop(W1,W2,W3,z1,h1,z2,h2,z3,y_i,train_y_actual_i,x_i);
61
62 % SGD update of the weights
63 - b3=b3-(alpha*grad_b3_f);
64 - W3=W3-(alpha*grad_W3_f);
65
66 - b2=b2-(alpha*grad_b2_f);
67 - W2=W2-(alpha*grad_W2_f);
68
69 - b1=b1-(alpha*grad_b1_f);
70 - W1=W1-(alpha*grad_W1_f);
71
72 - Fi_SGD(it)=(norm(y_i-train_y_actual_i))^2;
73 - if mod(it,out_interval)==0
74 -     iteration=it/out_interval;
75 -     F(iteration+1)=sum(Fi_SGD(it))/it;
76 -     percent_accurate(iteration+1)=(SGD_accuracy)/it;
77 -     fprintf(['During SGD Training \nIteration: ' num2str(it) '\nLoss: '...
78 -         num2str(F(iteration+1)) '\nAccuracy: '...
79 -         num2str(percent_accurate(iteration+1)) '\n\n'])
80 - end
81 - end
82
83 % plot F as a function of iterations
84 - subplot(211);
85 - x=[0:out_interval:it];
86 - plot(x,F)
87 - xlabel('Iteration')
88 - ylabel('Loss')
89
90 - title('Training Progress: Loss')
91
92 % plot percent_accurate as a function of iterations
93 - subplot(212);
94 - plot(x,(percent_accurate*100))
95 - xlabel('Iteration')
96 - ylabel('Accuracy (%)')
97 - title('Training Progress: Accuracy')

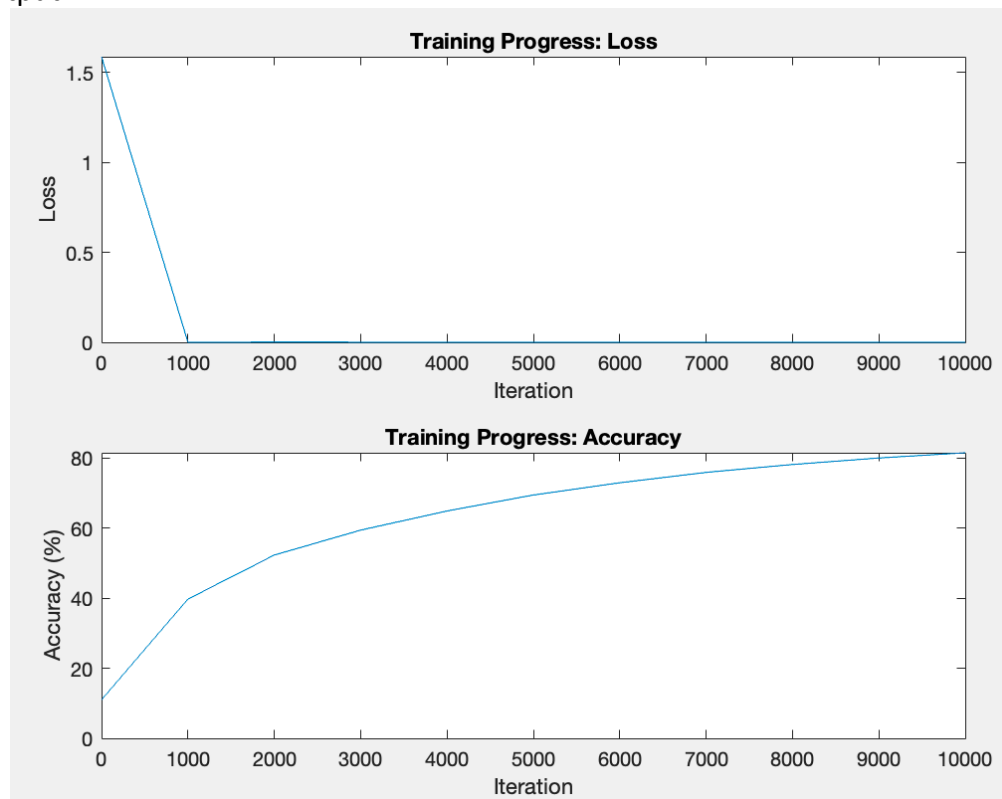
```

```

99 % test set
100 test_accuracy=0;
101 for j=1:n_test
102     x_j=testimages(:,j);
103     % forward pass for image j
104     [z1,h1,z2,h2,z3,y_j]=forward_pass(W1,W2,W3,b1,b2,b3,x_j);
105     y_j_label=find(y_j==max(y_j));
106     if y_j_label==testlabels(j)
107         test_accuracy = test_accuracy+1;
108     end
109     test_y_actual_j=test_y_actual(:,j);
110
111     Fi_test(j)=(norm(y_j-test_y_actual_j))^2;
112     if j==n_test
113         F_test=sum(Fi_test(j))/j;
114         percent_accurate_test=(test_accuracy)/j;
115         fprintf(['Test Iteration: ' num2str(j) '\nTest Loss: '...
116             num2str(F_test) '\nTest Accuracy: '...
117             num2str(percent_accurate_test) '\n\n'])
118     end
119 end

```

Q2(a) Output



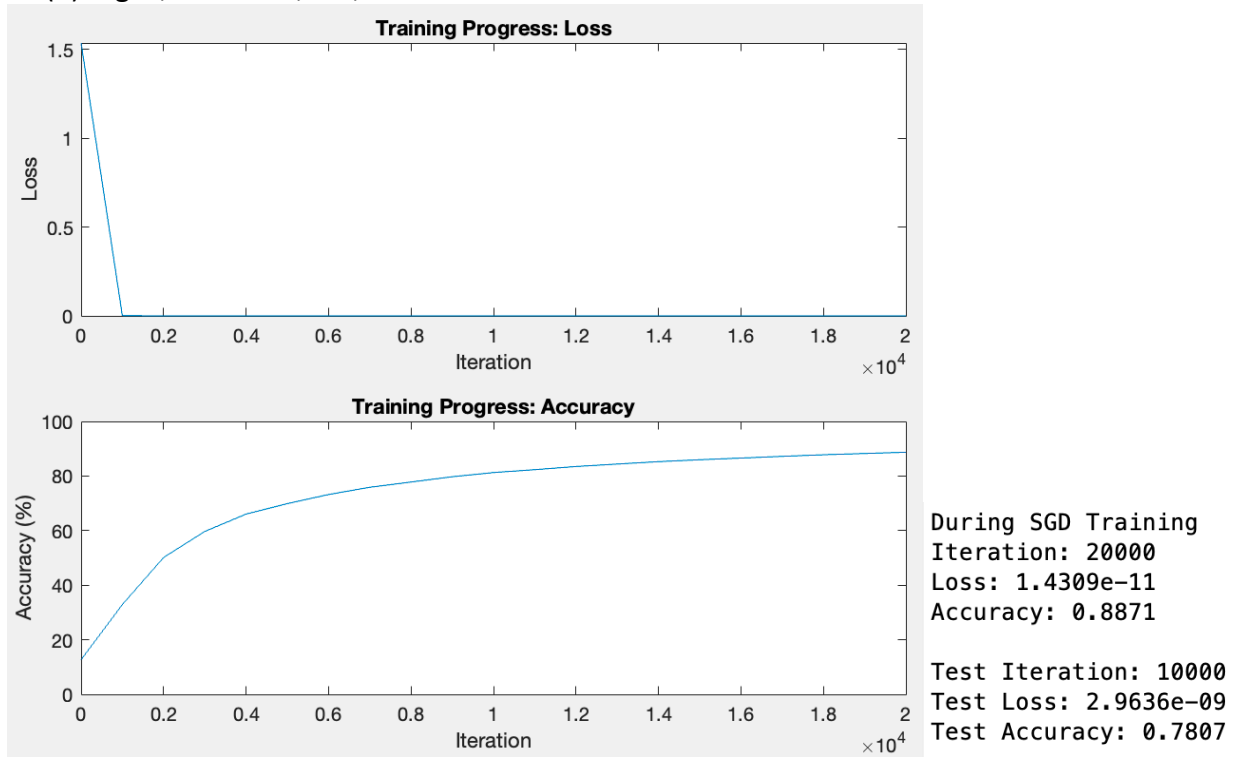
Selected outputs from the command window:

Before SGD Training
Loss: 1.5859
Accuracy: 0.11

During SGD Training
Iteration: 10000
Loss: 4.5287e-07
Accuracy: 0.8142

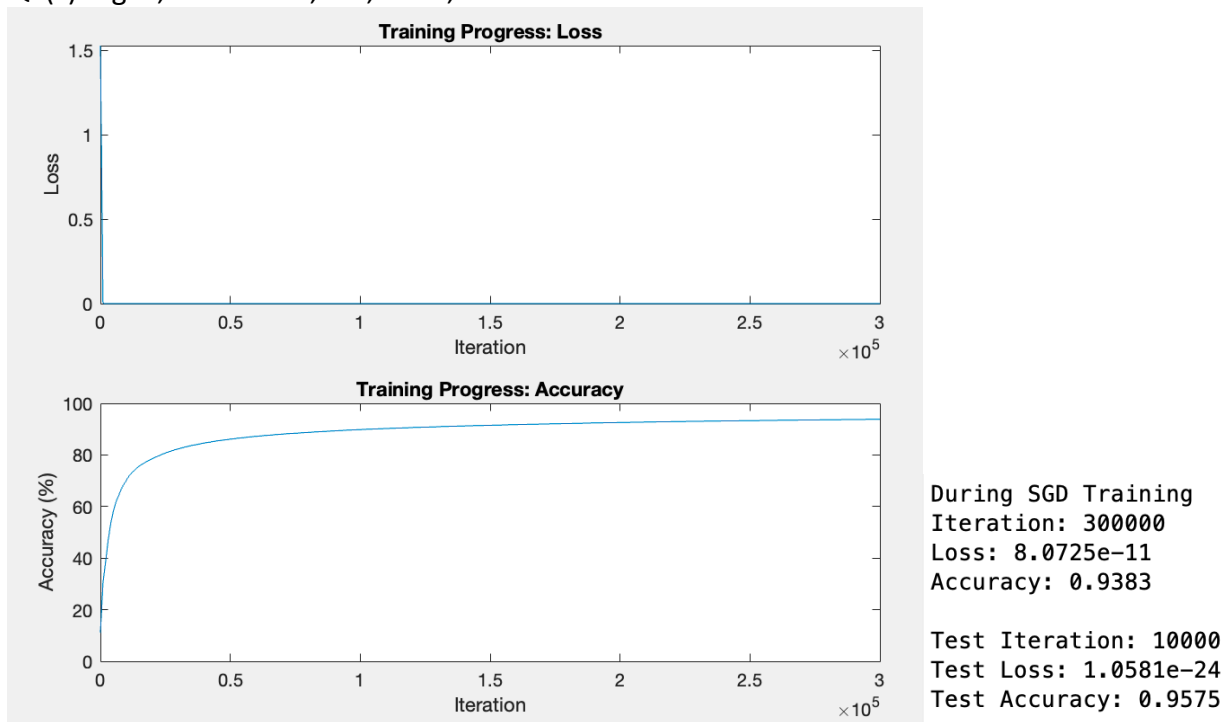
Test Iteration: 10000
Test Loss: 7.598e-11
Test Accuracy: 0.8041

Q2(b) Digits, maxit=20,000, n=1000



Final training accuracy: 88.71%, Test accuracy: 78.07%

Q2(c) Digits, maxit=300,000, n=60,000



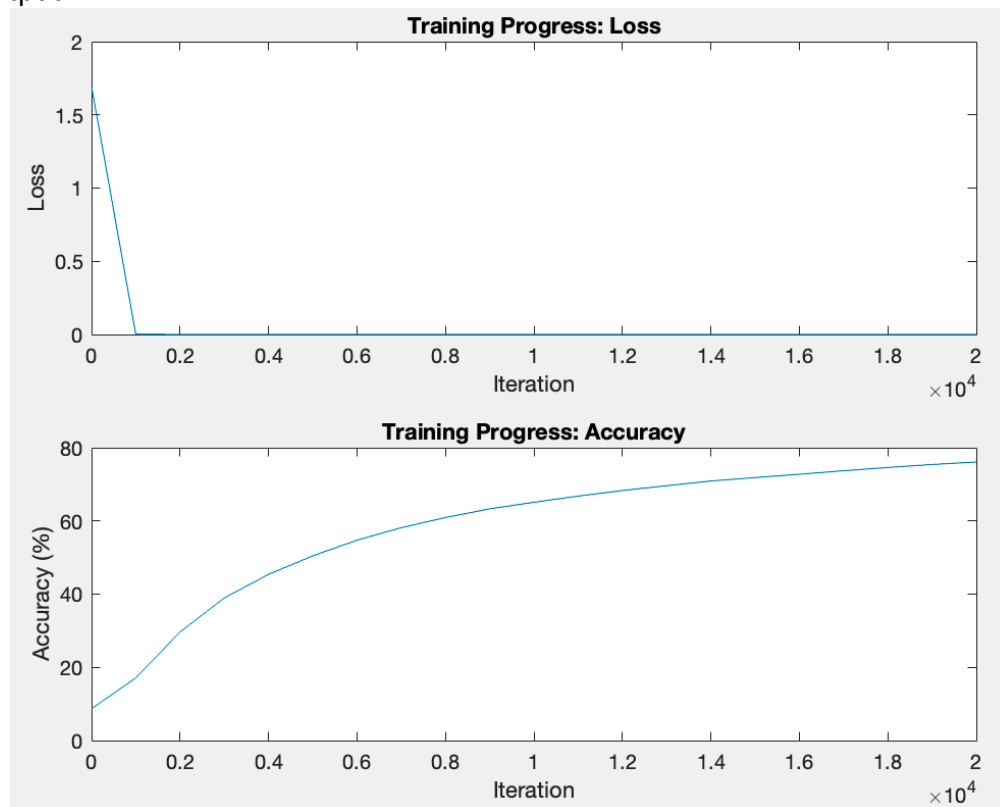
Final training accuracy: 93.83%, Test accuracy: 95.75%

Q2(d) Code (Fashion, maxit=20,000, n=1000)

```
1 % initialize parameters
2 - maxit=20000;
3 - alpha=0.01;
4 - n=1000;
5 - n_test=10000;
6 - out_interval=1000;
7
8 % load the training and test images + labels
9 - trainimages=loadMNISTImagesAsVectors('fashion_train-images.idx3-ubyte');
10 - testimages=loadMNISTImagesAsVectors('fashion_t10k-images.idx3-ubyte');
11 - trainlabels=loadMNISTLabels('fashion_train-labels.idx1-ubyte');
12 - testlabels=loadMNISTLabels('fashion_t10k-labels.idx1-ubyte');
```

The rest of the code is the same as Q2(a).

Q2(d) Output



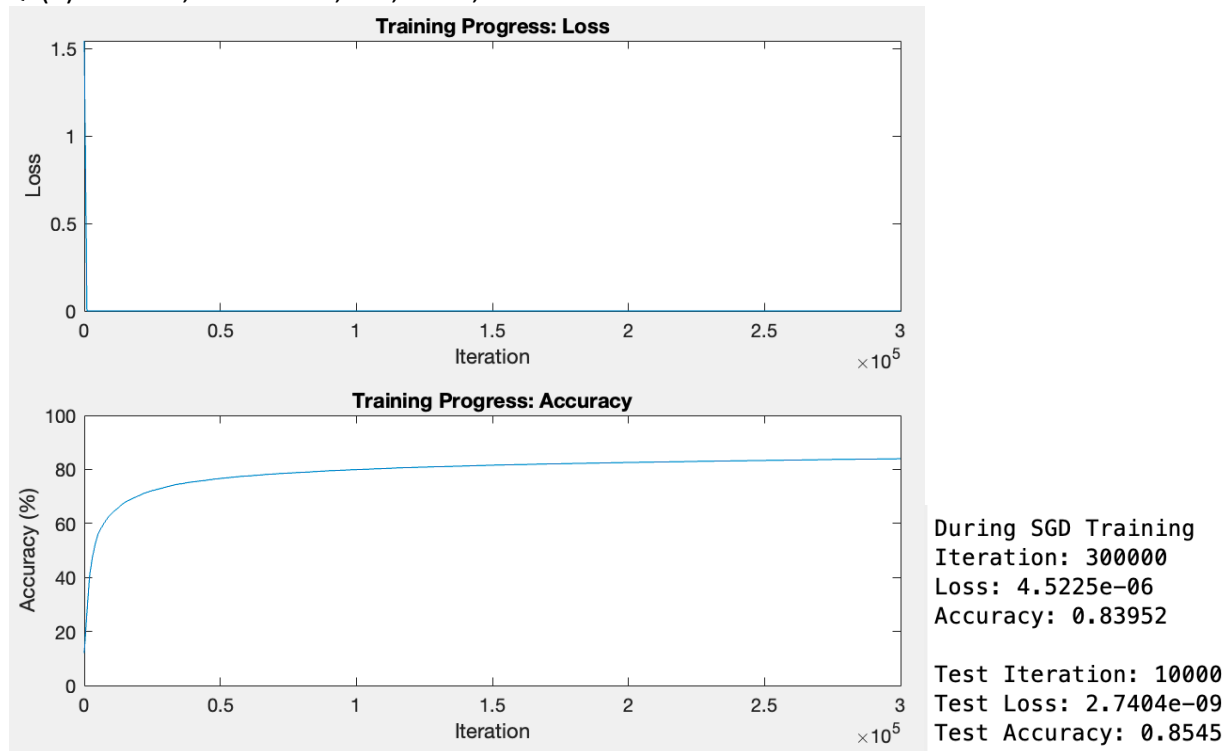
Selected outputs from the command window:

Before SGD Training
Loss: 1.6937
Accuracy: 0.087

During SGD Training
Iteration: 20000
Loss: 3.3106e-08
Accuracy: 0.76095

Test Iteration: 10000
Test Loss: 0.00010647
Test Accuracy: 0.7222

Q2(e) Fashion, maxit=300,000, n=60,000



Final training accuracy: 83.95%, Test accuracy: 85.45%

Comparing the digit images test accuracy (95.75%) with fashion images test accuracy (85.45%), using data with maxit=300,000 and training n=60,000, we can see that the digit images have achieved a higher accuracy than fashion images.

This is as expected because digits have less variation in its conformation compared to fashion images (refer to Q1b).