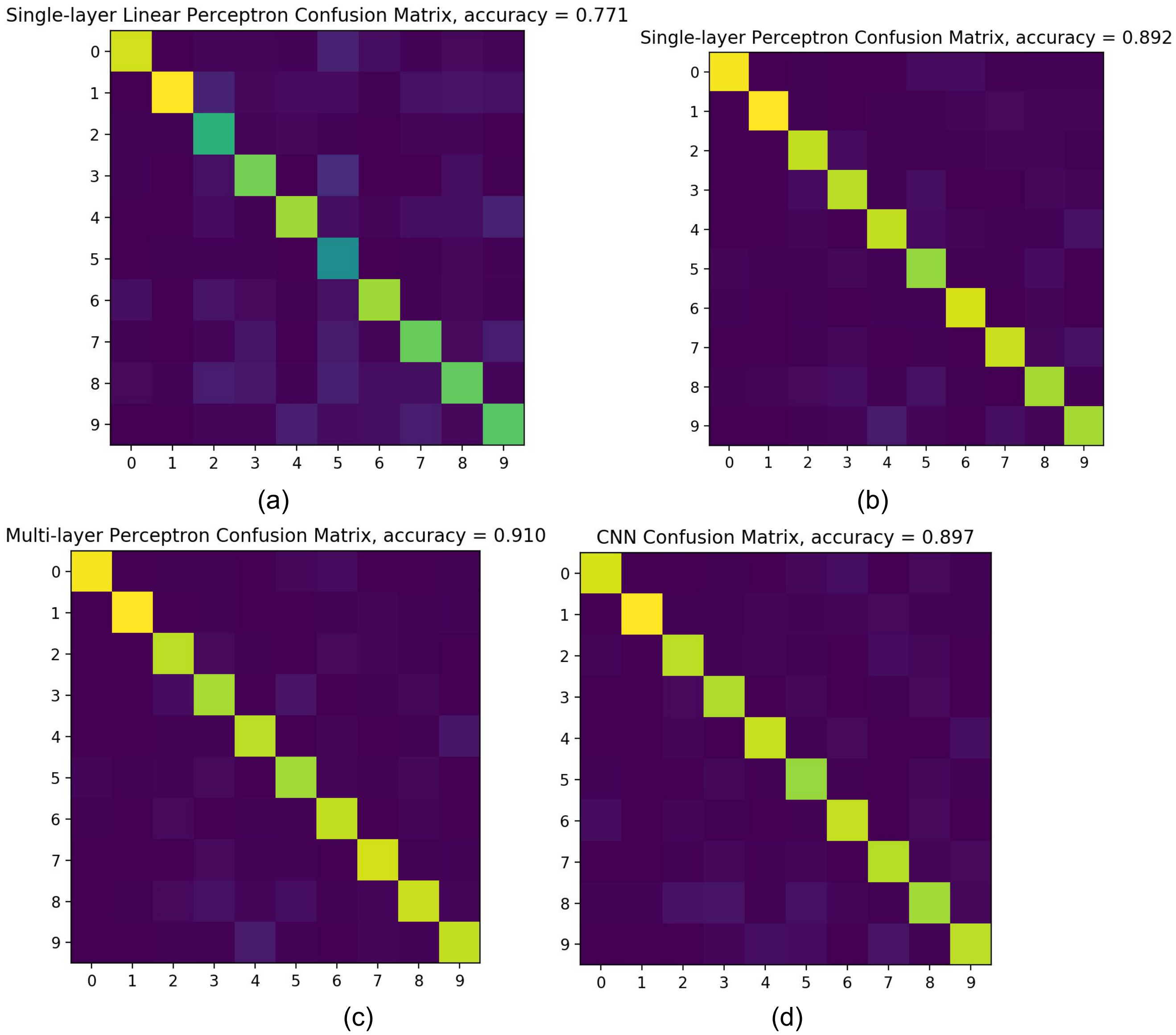
Homework-4



Single-layer Linear Perceptron Confusion Matrix, accuracy = 0.771

Single-layer Perceptron Confusion Matrix, accuracy = 0.892

(a)

(b)

Multi-layer Perceptron Confusion Matrix, accuracy = 0.910

CNN Confusion Matrix, accuracy = 0.897

(c)

(d)

Figure (a) –**Confusion Matrix(SLP-Linear)**; *Figure (b)* – **Confusion Matrix(SLP)**; *Figure (c)* – **Confusion Matrix(MLP)**; *Figure (d)* – **Confusion Matrix(CNN)**

4125 0.893 LR=0.15 6751 0.898 LR=0.2

Summary:

Three different classification algorithms are implemented in this homework.

1. Single Layer Linear Perceptron:

   The initial weights and bias are set using a gaussian with mean 0 and std 1. Fc method does the forward propagation. Loss_euclidean method calculates the loss and differential with respect to y. Fc_backward method does the backward propagation. After tuning the parameters, learning rate = 0.1, decay = 0.5 and number of iterations = 3000 gave an accuracy of 0.771. The confusion matrix is as shown in figure(a).

2. Single Layer Perceptron:

   All the details are the same as single layer linear perceptron except a softmax layer is added. Loss_cross_entropy_softmax method calculates the loss and differential with respect to y. After tuning the parameters, learning rate = 0.9, decay = 0.05 and number of iterations = 10000 gave an accuracy of 0.892. The confusion matrix is as shown in figure(b).

3. Multi Layer Perceptron:

   Relu method implements the Relu activation function. Relu_backward functions implements the backward propagation in the Relu layer. After tuning the parameters, learning rate = 1.5, decay = 0.4 and number of iterations = 20000 gave an accuracy of 0.91. The confusion matrix is as shown in figure(c).

4. Convolutional Neural Net:

   The conv method implements the convolution of the input tensor with convolutional weights. The conv_backward method implements the backward propagation in the convolutional layer. pool2*2 method implements max-pooling with window size=2 and stride=2. Flattening method flattens the tensor for a fully connected layer. After tuning the parameters, learning rate = 0.2, decay = 0.7 and number of iterations = 9000 gave an accuracy of 0.897. The confusion matrix is as shown in figure(d).