Computer Vision Lab 1 Report

Jingyan Li

1. SIMPLE 2D CLASSIFIER

The given sample to train the 2D classifier is not linearly separable. The data in each class compose a ring according to the plot. Three classifiers are trained and tested, namely linear classifier, multi-layer perceptron and a transformed linear classifier.

1.1 Linear Classifier

The loss and accuracy of the linear classifier is shown in Table 1. We can see the training loss keeps going down to around 0.69 as a suggestion of convergence. But the validation accuracy varies around 50% rather than increases. This is not the expected result. Linear classifier can only fit a line in 2D space which cannot successfully separate the two classes in the shape of ring.

Table 1. The loss and validation accuracy per epoch of the linear classifier.

Epoch	Loss	Acc
01	1.4926	51.1905%
02	0.9954	53.3730%
03	0.7577	60.5159%
04	0.7051	55.1587%
05	0.6969	57.1429%
06	0.6945	62.5000%
07	0.6938	55.1587%
08	0.6933	52.1825%
09	0.6932	49.8016%
10	0.6928	50.7937%

1.2 Multi-layer Perceptron

The training loss keeps going down and the validation accuracy goes up from 72.4% to 99.8% as shown in Table 2. The result performs better than the linear classifier because multi-layer perceptron can distinguish data that are non-linear separable due to its non-linear activations.

1.3 Feature Transformation

We can utilize feature engineering to make the non-linearly separable data into linearly separable. Since the two classes are shown as ring in 2D space, I assume transform the features into the second order is helpful. Let (x, y) be original input of a data point, I transformed the input into (x^2, y^2) and continued to conduct the prediction in 2D space. The loss and accuracy of the transformation are shown in Table 3.

2. DIGIT CLASSIFIER

In the digit classification task, multi-layer perceptron and convolutional network are compared.

2.1 Multi-layer Perceptron

The loss and accuracy of linear MLP and MLP of one hidden layer are shown in Table 4 and Table 5 respectively. We can see the hidden layer which is able distinguish non-linear separable data performs better than the linear model.

2.2 Convolutional Network

The convolutional network performs better than the previous two networks. The result is shown in Table 6.

jingyli@ethz.ch

Table 2. The loss and validation accuracy per epoch of the multi-layer perceptron.

Epoch	Loss	Acc
01	0.6000	72.4206%
02	0.3606	99.0079%
03	0.1398	100.0000%
04	0.0522	99.8016%
05	0.0276	99.8016%
06	0.0179	99.8016%
07	0.0121	99.8016%
08	0.0091	99.8016%
09	0.0071	99.8016%
10	0.0062	99.8016%

Table 3. The loss and validation accuracy per epoch of feature transformed linear classifier. The transformation is (x^2, y^2) .

Epoch	Loss	Acc
01	3.5559	50.0000%
02	0.8525	50.1984%
03	0.5004	53.1746%
04	0.4644	62.8968%
05	0.4283	84.9206%
06	0.3932	91.0714%
07	0.3593	95.6349%
08	0.3299	95.4365%
09	0.3012	97.8175%
10	0.2751	99.0079%

Table 4. The loss and validation accuracy per epoch of the linear MLP.

Epoch	Loss	Acc
01	0.4099	89.6700%
02	0.3337	90.9700%
03	0.3231	90.6300%
04	0.3174	91.3500%
05	0.3088	91.1300%

Table 5. The loss and validation accuracy per epoch of the MLP with one hidden layer of dimension 32.

Epoch	Loss	Acc
01	0.4219	90.1900%
02	0.2804	92.6300%
03	0.2431	93.5000%
04	0.2203	92.5900%
05	0.2054	93.9500%

2.3 Number of Parameters

The number of parameters of a neuron network is calculated by the sum of all connections between layers. The number of connections between two layers equal to `(number of neurons in the previous layer + bias)*(number of neurons in the later layer)`.

For MLP, it is fully connected. Thus each pixel represents a neuron. The number of parameters of MLP with

Table 6. The loss and validation accuracy per epoch of the CNN.

Epoch	**Loss**	**Acc**
01	0.2861	96.7200%
02	0.1014	97.8000%
03	0.0737	97.9700%
04	0.0587	98.2600%
05	0.0486	98.0700%

one hidden layer is calculated as,

$$MLPparameters = (28 * 28 + 1) * 32 + (32 + 1) * 10 = 25450 \tag{1}$$

For CNN, the number of neurons for each layer is equal to `(kernal window size * kernal window size * channels + bias)`. The number of parameters of convolution network is calculated as,

$$CNNparameters = (3 * 3 * 1 + 1) * 8 + (3 * 3 * 8 + 1) * 16 + (3 * 3 * 16 + 1) * 32 + (32 + 1) * 10 = 6218 \tag{2}$$

2.4 Confusion matrix

The two confusion matrices of MLP and CNN are given in Figure 1 and Figure 2. We can find that the MLP cannot predict as many true positives (i.e., the sum of diagonal values) as CNN does. Some classes are not well predicted by MLP. For example, for class 3, 55 samples are mis-predicted as 5. While by CNN only 13 samples from class 3 are mispredicted as 5.

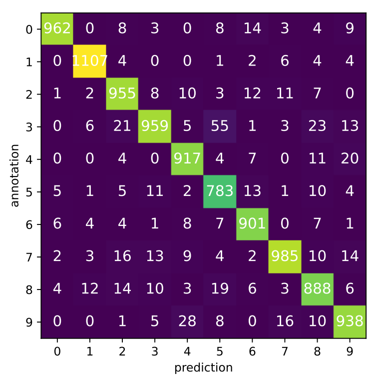

Figure 1. Confusion matrix of the multi-layer perceptron of one hidden layer.

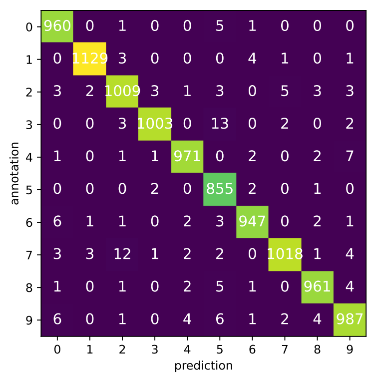

Figure 2. Confusion matrix of the convolution network.