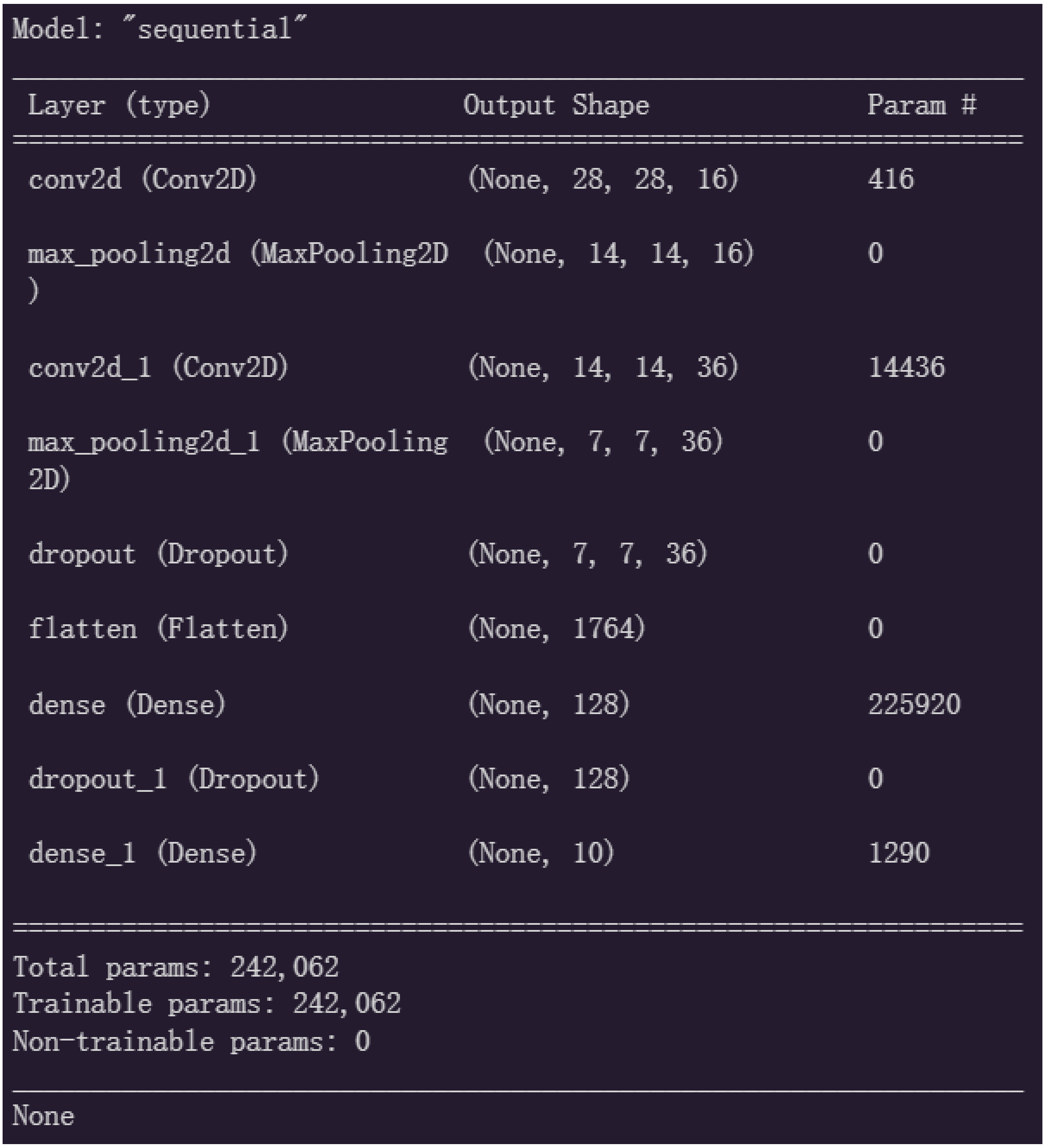
IC HW5

B07611001 Li-Wei Yang, B07611002 Po-Sheng Cheng, B07611016 Ping-Jian Huang

Problem 1

Original Setup

Model

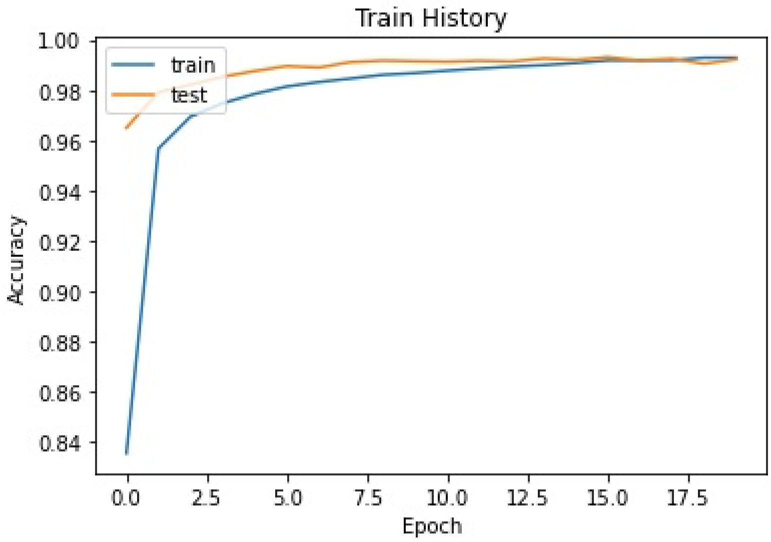


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 16)        416

 max_pooling2d (MaxPooling2D  (None, 14, 14, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 36)        14436

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 36)         0
 2D)

 dropout (Dropout)           (None, 7, 7, 36)          0

 flatten (Flatten)           (None, 1764)              0

 dense (Dense)               (None, 128)               225920

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 242,062
Trainable params: 242,062
Non-trainable params: 0
_________________________________________________________________
None
```

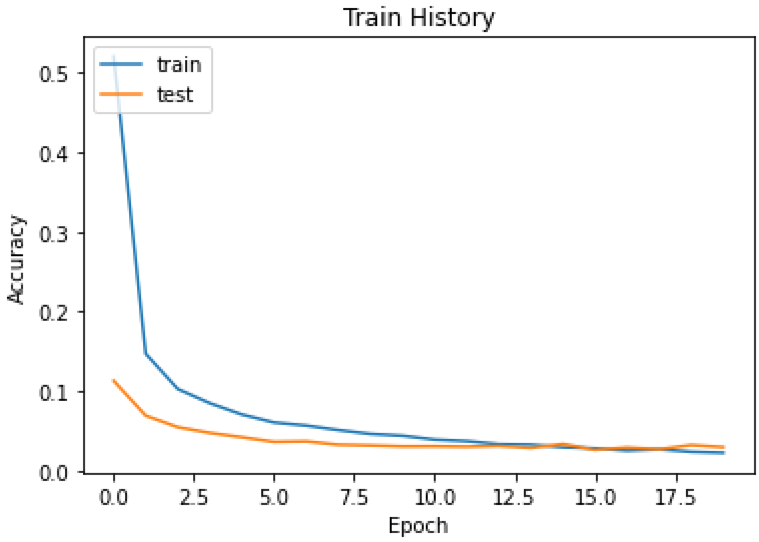
Hyperparameter

Item	Value
Optimizer	Adam
validation_split	0.2
epochs	20
batch_size	300

Accuracy Train History



Loss Train History



Test Accuracy

0.9919999837875366

Confusion Matrix

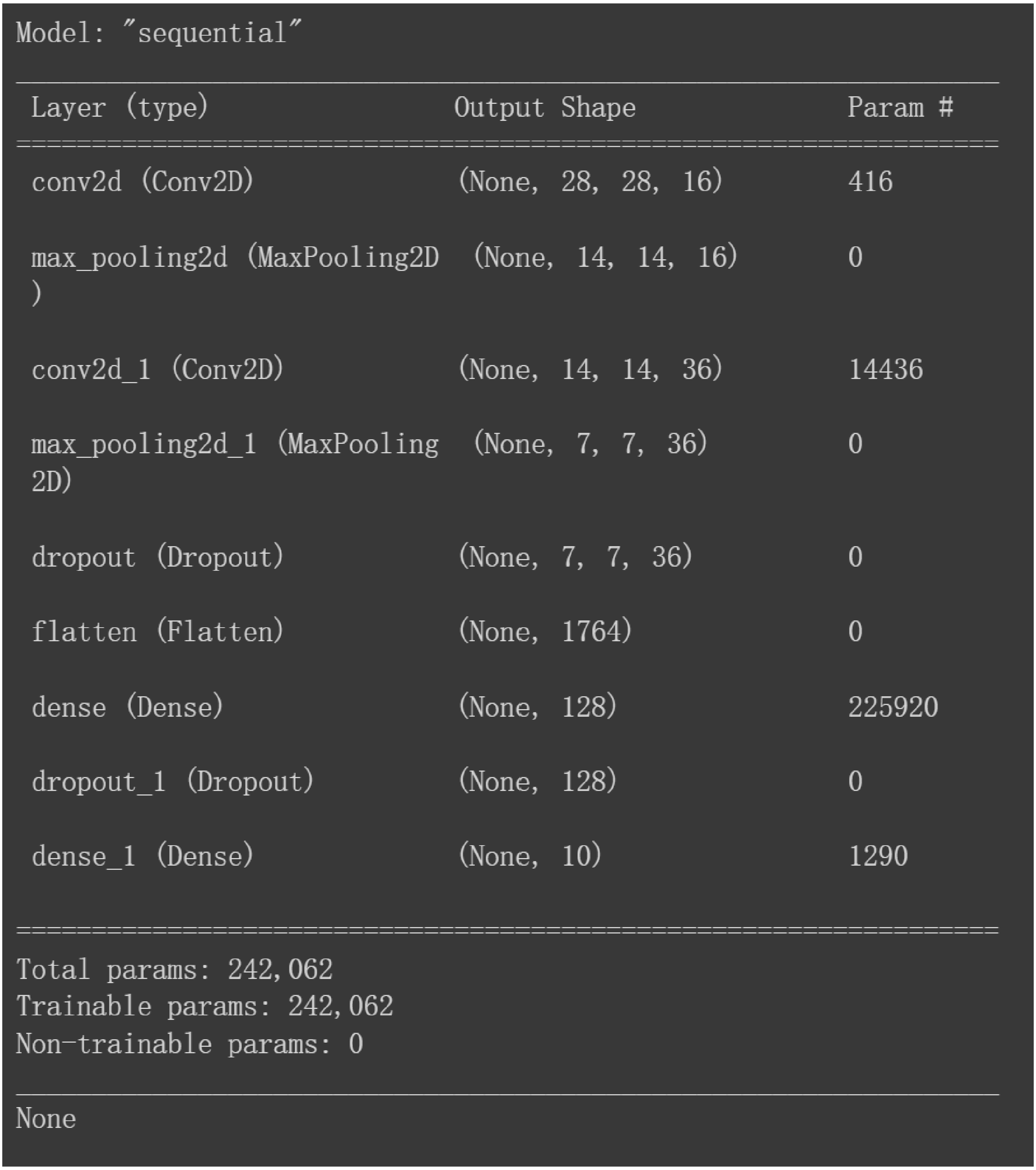
label	0	1	2	3	4	5	6	7	8	9
0	975	0	1	0	0	1	1	1	1	0
1	0	1129	0	2	0	0	0	3	1	0
2	0	0	1025	0	0	0	0	6	1	0
3	0	0	1	1006	0	1	0	0	2	0
4	0	0	0	0	980	0	1	0	0	1
5	2	0	0	8	0	880	1	0	0	1

label	0	1	2	3	4	5	6	7	8	9
6	3	2	1	0	2	2	945	0	3	0
7	0	1	1	1	0	0	0	1023	1	1
8	3	0	2	0	0	0	0	0	968	1
9	0	1	1	1	10	4	0	3	2	987

Modified Setup

- Replacing all "relu" activation with "elu" could increase accuracy to 0.9933.
- Adding another hidden layer with 64 unit doesn't seems to help.
- Reducing dropout rate also has no positive effect.

Model

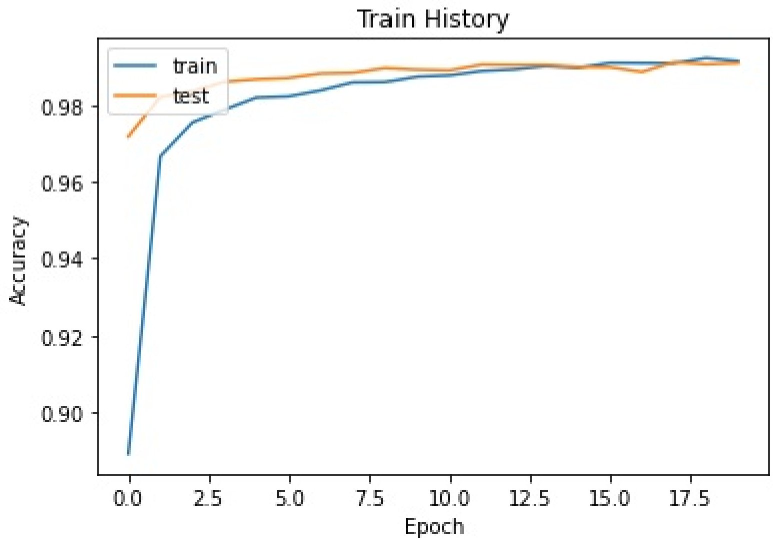


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 16)        416

 max_pooling2d (MaxPooling2D  (None, 14, 14, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 36)        14436

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 36)         0
 2D)

 dropout (Dropout)           (None, 7, 7, 36)          0

 flatten (Flatten)           (None, 1764)              0

 dense (Dense)               (None, 128)               225920

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 242,062
Trainable params: 242,062
Non-trainable params: 0
_________________________________________________________________
None
```

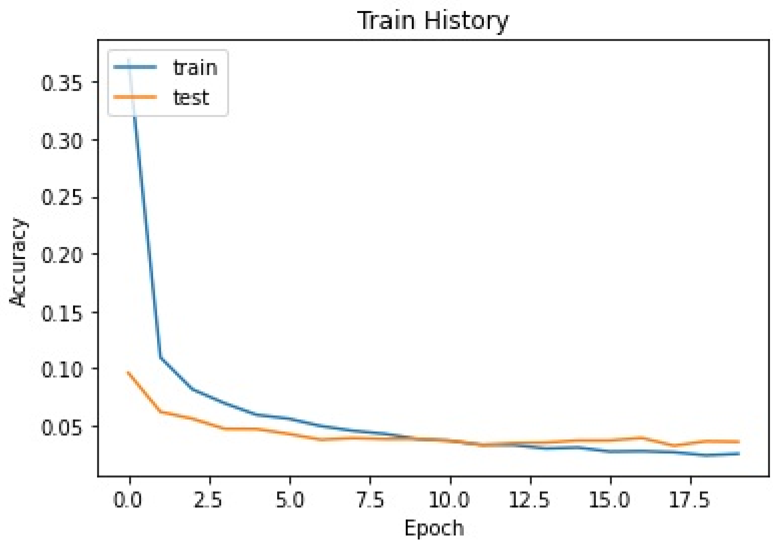
Hyperparameter

Item	Value
Optimizer	Adam
validation_split	0.2
epochs	20
batch_size	300

Accuracy Train History



Loss Train History



Test Accuracy

0.9933000206947327

Confusion Matrix

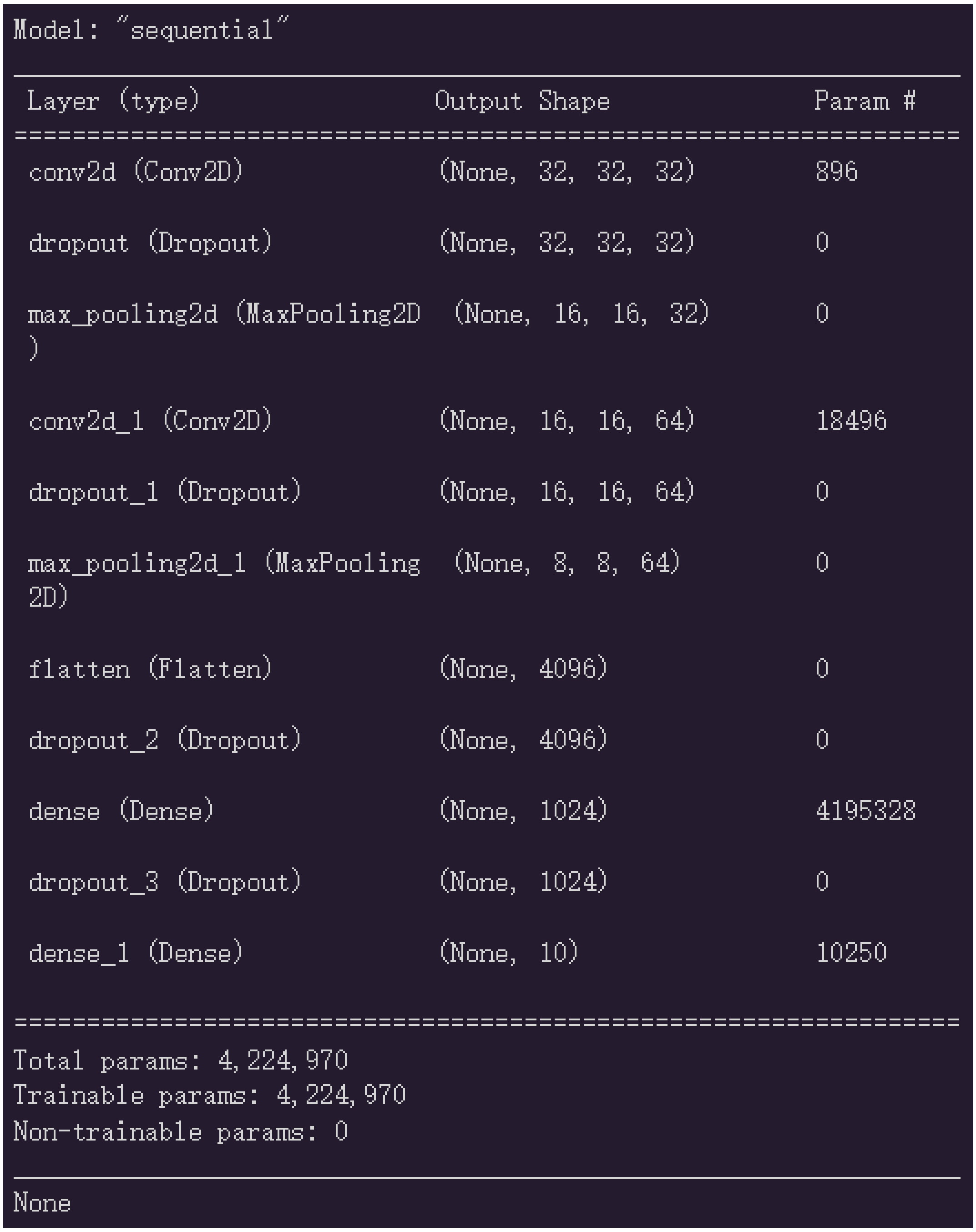
label	0	1	2	3	4	5	6	7	8	9
0	9	65	248	1	33	62	6	550	0	6
1	0	1114	0	0	0	4	0	17	0	0
2	0	40	875	8	0	1	0	108	0	0
3	0	0	0	988	0	7	0	15	0	0
4	0	9	1	0	773	1	0	190	0	8

label	0	1	2	3	4	5	6	7	8	9
5	0	0	1	5	0	883	0	1	0	2
6	0	10	6	0	3	167	769	3	0	0
7	0	0	0	0	0	0	0	1028	0	0
8	0	40	26	152	22	255	0	383	60	36
9	0	0	2	4	1	20	0	115	20	867

Problem 2

Original Setup

Model

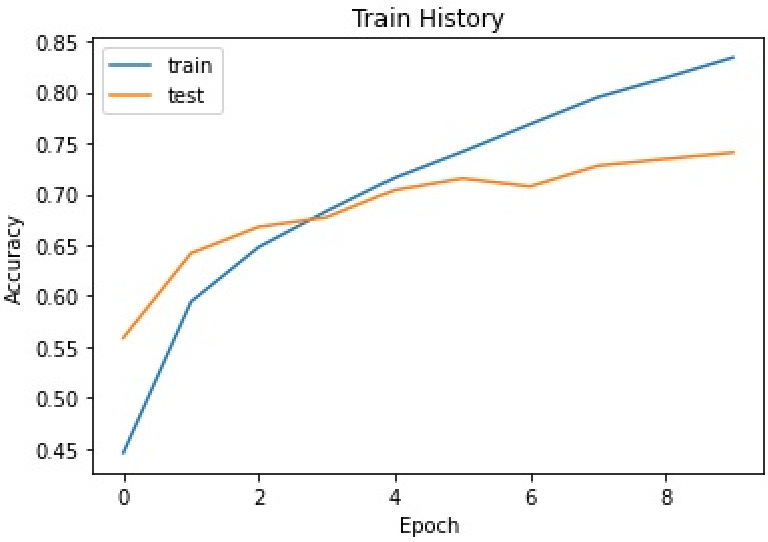


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 dropout (Dropout)           (None, 32, 32, 32)        0

 max_pooling2d (MaxPooling2D  (None, 16, 16, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 16, 16, 64)        18496

 dropout_1 (Dropout)         (None, 16, 16, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 64)         0
 2D)

 flatten (Flatten)           (None, 4096)              0

 dropout_2 (Dropout)         (None, 4096)              0

 dense (Dense)               (None, 1024)              4195328

 dropout_3 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 10)                10250

=================================================================
Total params: 4,224,970
Trainable params: 4,224,970
Non-trainable params: 0
_________________________________________________________________
None
```

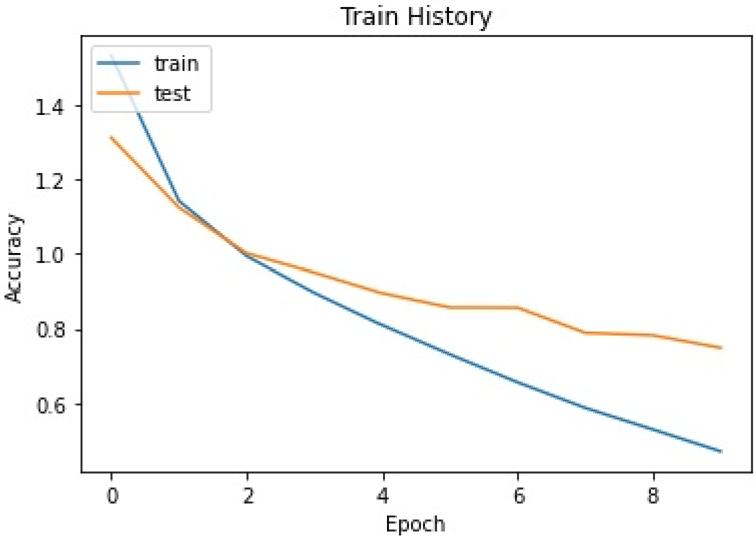
Hyperparameter

Item	Value
Optimize	Adam
validation_split	0.2
epochs	10
batch_size	128

Accuracy Train History



Loss Train History



Test Accuracy

0.7432000041007996

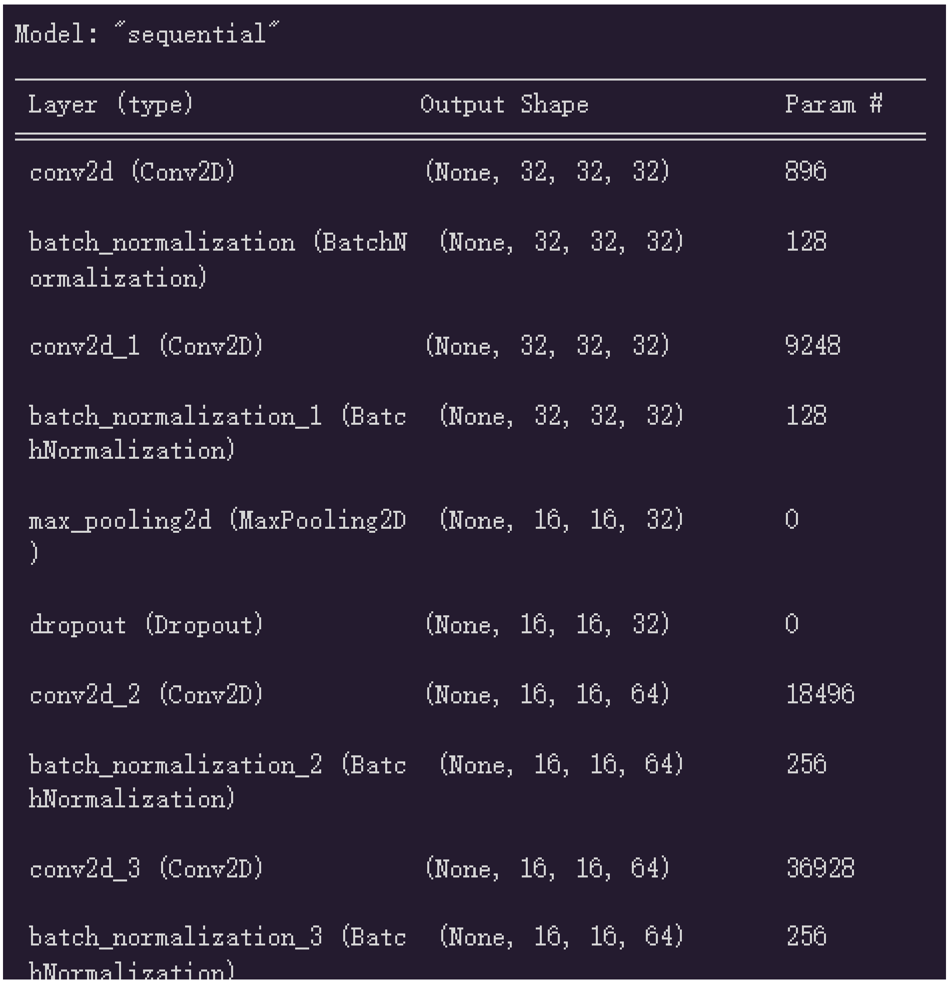
Confusion Matrix

label	0	1	2	3	4	5	6	7	8	9
0	749	14	54	14	10	9	13	8	91	38
1	11	848	13	6	4	5	10	2	32	69
2	57	3	701	37	86	33	49	16	10	8
3	18	9	102	523	81	160	54	20	22	11
4	16	3	89	54	742	18	33	32	12	1
5	12	1	79	165	65	609	24	36	8	1
6	2	4	59	46	36	20	823	1	8	1
7	15	3	51	27	70	46	9	766	5	8
8	38	17	25	6	10	9	5	1	872	17
9	21	71	15	13	6	13	9	13	40	799

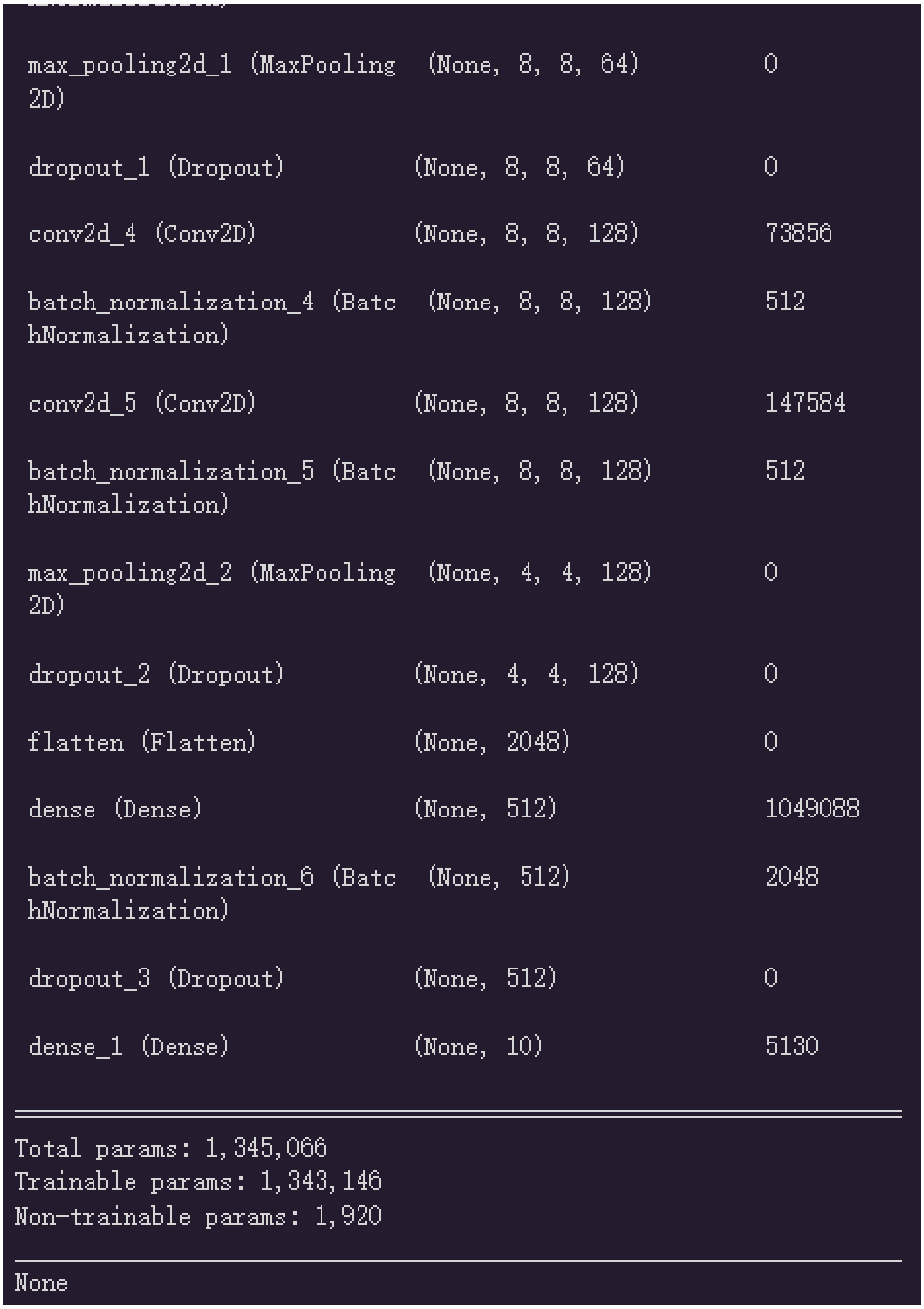
Modified Setup

- Change Model structure.
- Data augmentation, including rotation, flip, shift and zoom.
- Train for more epochs.

Model



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 batch_normalization (BatchN  (None, 32, 32, 32)       128
 ormalization)

 conv2d_1 (Conv2D)           (None, 32, 32, 32)        9248

 batch_normalization_1 (Batc  (None, 32, 32, 32)       128
 hNormalization)

 max_pooling2d (MaxPooling2D  (None, 16, 16, 32)       0
 )

 dropout (Dropout)           (None, 16, 16, 32)        0

 conv2d_2 (Conv2D)           (None, 16, 16, 64)        18496

 batch_normalization_2 (Batc  (None, 16, 16, 64)       256
 hNormalization)

 conv2d_3 (Conv2D)           (None, 16, 16, 64)        36928

 batch_normalization_3 (Batc  (None, 16, 16, 64)       256
 hNormalization)
```

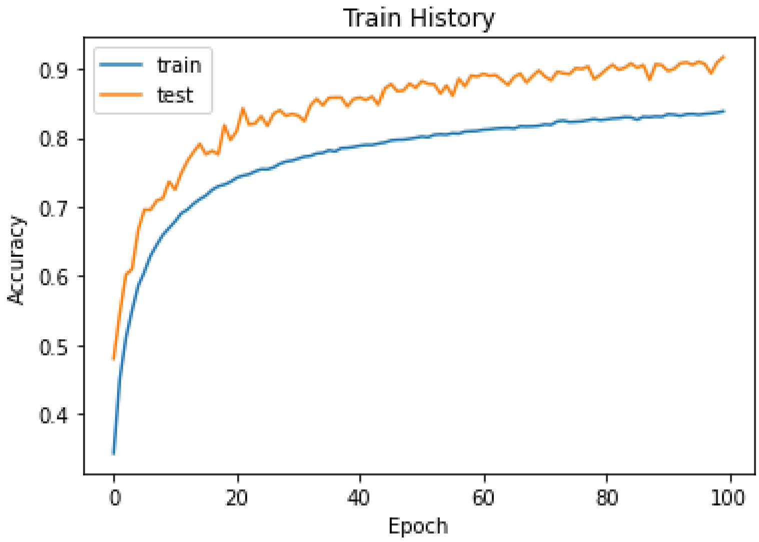


```
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 8, 8, 64)          0

 conv2d_4 (Conv2D)           (None, 8, 8, 128)         73856

 batch_normalization_4 (Batc  (None, 8, 8, 128)        512
 hNormalization)

 conv2d_5 (Conv2D)           (None, 8, 8, 128)         147584

 batch_normalization_5 (Batc  (None, 8, 8, 128)        512
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 dropout_2 (Dropout)         (None, 4, 4, 128)         0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 batch_normalization_6 (Batc  (None, 512)              2048
 hNormalization)

 dropout_3 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 1,345,066
Trainable params: 1,343,146
Non-trainable params: 1,920
_________________________________________________________________
None
```

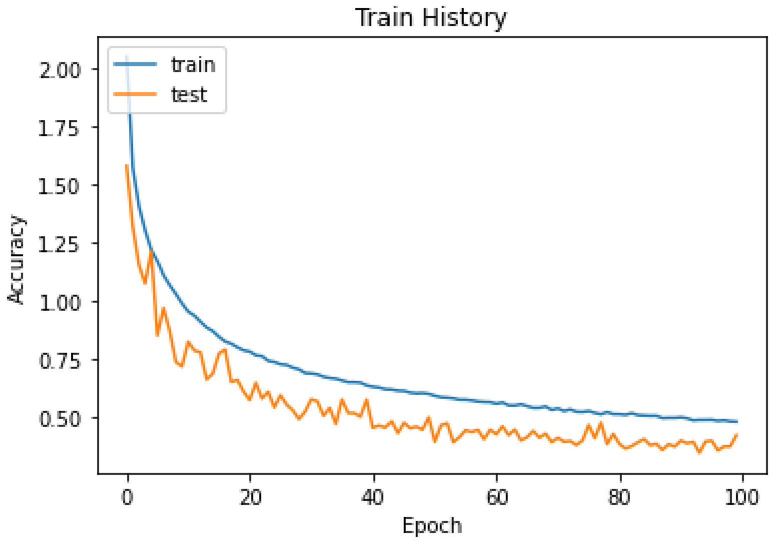
Hyperparameter

Item	Value
Optimize	Adam
validation_split	0.2
epochs	100
batch_size	128

Accuracy Train History



Loss Train History



Test Accuracy

0.8543000221252441

Confusion Matrix

label	0	1	2	3	4	5	6	7	8	9
0	870	25	7	10	6	0	4	8	17	53
1	2	944	0	1	0	0	0	0	1	52
2	56	9	750	35	39	13	54	19	3	22
3	7	10	23	747	25	53	54	22	11	48
4	7	2	29	16	842	8	40	44	2	10
5	7	6	15	115	18	734	43	25	5	32

label	0	1	2	3	4	5	6	7	8	9
6	4	6	7	9	8	1	945	3	3	14
7	9	7	7	12	14	15	8	896	2	30
8	49	41	4	1	1	0	4	0	865	35
9	5	39	0	0	1	1	1	1	2	950

Non-programing Problems

Explain the function of convolution layer, pooling layer and fully connected layer in CNN model. Please attach your result with the discussion.

Convolution layer（卷積層）：

The major building blocks used in CNNs.

A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image.

Pooling layer（池化層）：

Summarize the presence of features in an input image.

Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively.

Fully connected layer（全連接層）：

Form the last few layers in the network.

The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.