MNIST Data Prediction With Different Types of Deep Learning Models

Objective

This notebook tests 3 different deep learning models on the MNIST hand written digits dataset to see which type of model is able to most accurately predict the digits into one of 10 categories (0 to 9) given an associated image. The test set accuracy will be used to assess which model performs the best, potential recommendations will also be outlined.

The three models that will be tested will be the following:

- Model 1: A densely connected deep neural network
- Model 2: A densely connected deep neural network with dropout applied to help with regularization
- Model 3: A convolutional neural network (CNN) with max pooling

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, precision_recall_curve, roc_auc_score, ro

#import keras objects for Deep Learning
from keras.models import Sequential
from keras.layers import Input,Dense, Flatten, Dropout, BatchNormalization
from keras.datasets import mnist
```

Description of the Data

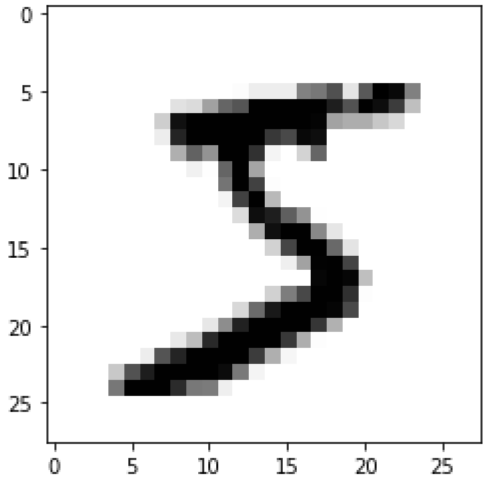
The MNIST handwritten digits dataset is one of the most popular datasets used in machine learning research and consists of grey-scale images (28x28 pixels). It has 60000 training set images and 10000 test set images and comes part of the Keras library. The following snippets of code loads the data, shows some examples of how it looks, shows the shape etc. and then does some pre-processing of the data to normalise to be used by the deep learning framework of Keras.

In [3]:

```
(train_images,y_train),(test_images,y_test) = mnist.load_data()
```

In [3]:

```
plt.imshow(train_images[0],cmap=plt.cm.binary)
plt.show()
```



In [21]:
```
y_train[0]
```
Out[21]:
```
5
```
In [14]:
```
train_images.shape
```
Out[14]:
```
(60000, 28, 28)
```
In [15]:
```
test_images.shape
```
Out[15]:
```
(10000, 28, 28)
```
In [19]:
```
x_train = train_images.reshape((60000,28*28))
x_train = x_train.astype('float32')/255

x_test = test_images.reshape((10000,28*28))
x_test = x_test.astype('float32')/255
```
In [22]:
```
y_train
```
Out[22]:
```
array([5, 0, 4, ..., 5, 6, 8], dtype=uint8)
```
In [11]:
```
y_train_df=pd.Series(y_train)
```
In [14]:
```
y_train_df.value_counts()
```
Out[14]:
```
1    6742
7    6265
3    6131
2    5958
9    5949
0    5923
6    5918
8    5851
```

```
4    5842
5    5421
dtype: int64
```

Model 1: Dense Deep Learning Model without Dropout

In [5]:

```
network = Sequential()
network.add(Dense(512,activation='relu',input_shape=(28*28,)))
network.add(Dense(10,activation='softmax'))
```

In [6]:

```
network.compile(optimizer='rmsprop', loss='categorical_crossentropy',metrics=['accuracy
```

In [9]:

```
from tensorflow.keras.utils import to_categorical

train_labels = to_categorical(y_train)
test_labels = to_categorical(y_test)
```

In [34]:

```
train_labels[0]
```

Out[34]:

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

In [32]:

```
network.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 512)               401920

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 407,050
Trainable params: 407,050
Non-trainable params: 0
_________________________________________________________________
```

In [31]:

```
run_hist_nn1=network.fit(x_train,train_labels,epochs=100,batch_size=128,validation_data
```

```
Epoch 1/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0023 - accuracy: 0.99
94 - val_loss: 0.0981 - val_accuracy: 0.9807
Epoch 2/100
469/469 [==============================] - 10s 22ms/step - loss: 0.0016 - accuracy: 0.99
96 - val_loss: 0.0881 - val_accuracy: 0.9822
Epoch 3/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0017 - accuracy: 0.99
95 - val_loss: 0.0834 - val_accuracy: 0.9833
Epoch 4/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0013 - accuracy: 0.99
96 - val_loss: 0.0909 - val_accuracy: 0.9828
Epoch 5/100
```

```
469/469 [==============================] - 11s 23ms/step - loss: 8.0959e-04 - accuracy: 
0.9998 - val_loss: 0.0957 - val_accuracy: 0.9819
Epoch 6/100
469/469 [==============================] - 11s 23ms/step - loss: 6.9963e-04 - accuracy: 
0.9999 - val_loss: 0.1025 - val_accuracy: 0.9815
Epoch 7/100
469/469 [==============================] - 11s 24ms/step - loss: 6.3591e-04 - accuracy: 
0.9998 - val_loss: 0.0989 - val_accuracy: 0.9828
Epoch 8/100
469/469 [==============================] - 10s 22ms/step - loss: 4.0429e-04 - accuracy: 
0.9999 - val_loss: 0.1148 - val_accuracy: 0.9807
Epoch 9/100
469/469 [==============================] - 10s 22ms/step - loss: 3.8987e-04 - accuracy: 
0.9999 - val_loss: 0.1083 - val_accuracy: 0.9820
Epoch 10/100
469/469 [==============================] - 11s 23ms/step - loss: 2.8428e-04 - accuracy: 
0.9999 - val_loss: 0.1093 - val_accuracy: 0.9820
Epoch 11/100
469/469 [==============================] - 11s 23ms/step - loss: 4.1039e-04 - accuracy: 
0.9999 - val_loss: 0.1041 - val_accuracy: 0.9834
Epoch 12/100
469/469 [==============================] - 11s 24ms/step - loss: 2.6296e-04 - accuracy: 
0.9999 - val_loss: 0.1108 - val_accuracy: 0.9839
Epoch 13/100
469/469 [==============================] - 11s 23ms/step - loss: 2.0038e-04 - accuracy: 
0.9999 - val_loss: 0.1155 - val_accuracy: 0.9828
Epoch 14/100
469/469 [==============================] - 10s 22ms/step - loss: 2.1510e-04 - accuracy: 
0.9999 - val_loss: 0.1148 - val_accuracy: 0.9826
Epoch 15/100
469/469 [==============================] - 10s 22ms/step - loss: 1.7144e-04 - accuracy: 
0.9999 - val_loss: 0.1156 - val_accuracy: 0.9844
Epoch 16/100
469/469 [==============================] - 11s 23ms/step - loss: 2.0484e-04 - accuracy: 
0.9999 - val_loss: 0.1210 - val_accuracy: 0.9828
Epoch 17/100
469/469 [==============================] - 12s 25ms/step - loss: 7.6356e-05 - accuracy: 
1.0000 - val_loss: 0.1251 - val_accuracy: 0.9833
Epoch 18/100
469/469 [==============================] - 12s 26ms/step - loss: 5.2416e-05 - accuracy: 
1.0000 - val_loss: 0.1259 - val_accuracy: 0.9820
Epoch 19/100
469/469 [==============================] - 11s 24ms/step - loss: 5.3242e-05 - accuracy: 
1.0000 - val_loss: 0.1214 - val_accuracy: 0.9839
Epoch 20/100
469/469 [==============================] - 11s 23ms/step - loss: 7.6576e-05 - accuracy: 
1.0000 - val_loss: 0.1310 - val_accuracy: 0.9829
Epoch 21/100
469/469 [==============================] - 10s 22ms/step - loss: 7.5973e-05 - accuracy: 
1.0000 - val_loss: 0.1262 - val_accuracy: 0.9840
Epoch 22/100
469/469 [==============================] - 12s 26ms/step - loss: 4.8590e-05 - accuracy: 
1.0000 - val_loss: 0.1334 - val_accuracy: 0.9833
Epoch 23/100
469/469 [==============================] - 12s 25ms/step - loss: 9.5414e-06 - accuracy: 
1.0000 - val_loss: 0.1353 - val_accuracy: 0.9830
Epoch 24/100
469/469 [==============================] - 11s 23ms/step - loss: 2.5416e-05 - accuracy: 
1.0000 - val_loss: 0.1373 - val_accuracy: 0.9836
Epoch 25/100
```

```
469/469 [==============================] - 11s 23ms/step - loss: 6.1681e-06 - accuracy:
1.0000 - val_loss: 0.1345 - val_accuracy: 0.9839
Epoch 26/100
469/469 [==============================] - 10s 22ms/step - loss: 5.3005e-06 - accuracy:
1.0000 - val_loss: 0.1380 - val_accuracy: 0.9831
Epoch 27/100
469/469 [==============================] - 10s 22ms/step - loss: 8.5327e-06 - accuracy:
1.0000 - val_loss: 0.1358 - val_accuracy: 0.9833
Epoch 28/100
469/469 [==============================] - 11s 24ms/step - loss: 1.8150e-07 - accuracy:
1.0000 - val_loss: 0.1378 - val_accuracy: 0.9835
Epoch 29/100
469/469 [==============================] - 11s 25ms/step - loss: 8.9969e-07 - accuracy:
1.0000 - val_loss: 0.1352 - val_accuracy: 0.9835
Epoch 30/100
469/469 [==============================] - 10s 22ms/step - loss: 2.2608e-07 - accuracy:
1.0000 - val_loss: 0.1364 - val_accuracy: 0.9834
Epoch 31/100
469/469 [==============================] - 10s 21ms/step - loss: 2.4698e-08 - accuracy:
1.0000 - val_loss: 0.1371 - val_accuracy: 0.9835
Epoch 32/100
469/469 [==============================] - 10s 21ms/step - loss: 1.9334e-08 - accuracy:
1.0000 - val_loss: 0.1381 - val_accuracy: 0.9835
Epoch 33/100
469/469 [==============================] - 10s 21ms/step - loss: 1.7150e-08 - accuracy:
1.0000 - val_loss: 0.1389 - val_accuracy: 0.9833
Epoch 34/100
469/469 [==============================] - 12s 26ms/step - loss: 1.5324e-08 - accuracy:
1.0000 - val_loss: 0.1389 - val_accuracy: 0.9835
Epoch 35/100
469/469 [==============================] - 11s 24ms/step - loss: 1.4353e-08 - accuracy:
1.0000 - val_loss: 0.1396 - val_accuracy: 0.9835
Epoch 36/100
469/469 [==============================] - 10s 22ms/step - loss: 1.3226e-08 - accuracy:
1.0000 - val_loss: 0.1399 - val_accuracy: 0.9836
Epoch 37/100
469/469 [==============================] - 10s 22ms/step - loss: 1.2443e-08 - accuracy:
1.0000 - val_loss: 0.1405 - val_accuracy: 0.9834
Epoch 38/100
469/469 [==============================] - 10s 21ms/step - loss: 1.1661e-08 - accuracy:
1.0000 - val_loss: 0.1407 - val_accuracy: 0.9839
Epoch 39/100
469/469 [==============================] - 10s 22ms/step - loss: 1.1086e-08 - accuracy:
1.0000 - val_loss: 0.1415 - val_accuracy: 0.9839
Epoch 40/100
469/469 [==============================] - 12s 26ms/step - loss: 1.0604e-08 - accuracy:
1.0000 - val_loss: 0.1418 - val_accuracy: 0.9837
Epoch 41/100
469/469 [==============================] - 11s 24ms/step - loss: 1.0037e-08 - accuracy:
1.0000 - val_loss: 0.1422 - val_accuracy: 0.9838
Epoch 42/100
469/469 [==============================] - 10s 22ms/step - loss: 9.7295e-09 - accuracy:
1.0000 - val_loss: 0.1425 - val_accuracy: 0.9835
Epoch 43/100
469/469 [==============================] - 12s 25ms/step - loss: 9.3301e-09 - accuracy:
1.0000 - val_loss: 0.1428 - val_accuracy: 0.9836
Epoch 44/100
469/469 [==============================] - 11s 23ms/step - loss: 8.9665e-09 - accuracy:
1.0000 - val_loss: 0.1431 - val_accuracy: 0.9834
Epoch 45/100
```

```
469/469 [==============================] - 13s 29ms/step - loss: 8.6486e-09 - accuracy:
1.0000 - val_loss: 0.1433 - val_accuracy: 0.9836
Epoch 46/100
469/469 [==============================] - 10s 22ms/step - loss: 8.4062e-09 - accuracy:
1.0000 - val_loss: 0.1437 - val_accuracy: 0.9830
Epoch 47/100
469/469 [==============================] - 10s 22ms/step - loss: 8.1420e-09 - accuracy:
1.0000 - val_loss: 0.1437 - val_accuracy: 0.9833
Epoch 48/100
469/469 [==============================] - 10s 22ms/step - loss: 7.9234e-09 - accuracy:
1.0000 - val_loss: 0.1440 - val_accuracy: 0.9838
Epoch 49/100
469/469 [==============================] - 10s 21ms/step - loss: 7.6950e-09 - accuracy:
1.0000 - val_loss: 0.1446 - val_accuracy: 0.9834
Epoch 50/100
469/469 [==============================] - 9s 20ms/step - loss: 7.4367e-09 - accuracy:
1.0000 - val_loss: 0.1447 - val_accuracy: 0.9835
Epoch 51/100
469/469 [==============================] - 10s 22ms/step - loss: 7.3075e-09 - accuracy:
1.0000 - val_loss: 0.1450 - val_accuracy: 0.9836
Epoch 52/100
469/469 [==============================] - 10s 21ms/step - loss: 7.1645e-09 - accuracy:
1.0000 - val_loss: 0.1453 - val_accuracy: 0.9831
Epoch 53/100
469/469 [==============================] - 10s 22ms/step - loss: 6.9817e-09 - accuracy:
1.0000 - val_loss: 0.1455 - val_accuracy: 0.9836
Epoch 54/100
469/469 [==============================] - 9s 20ms/step - loss: 6.7870e-09 - accuracy:
1.0000 - val_loss: 0.1459 - val_accuracy: 0.9835
Epoch 55/100
469/469 [==============================] - 9s 20ms/step - loss: 6.6400e-09 - accuracy:
1.0000 - val_loss: 0.1461 - val_accuracy: 0.9831
Epoch 56/100
469/469 [==============================] - 9s 20ms/step - loss: 6.5486e-09 - accuracy:
1.0000 - val_loss: 0.1463 - val_accuracy: 0.9834
Epoch 57/100
469/469 [==============================] - 10s 21ms/step - loss: 6.4294e-09 - accuracy:
1.0000 - val_loss: 0.1465 - val_accuracy: 0.9834
Epoch 58/100
469/469 [==============================] - 10s 21ms/step - loss: 6.2962e-09 - accuracy:
1.0000 - val_loss: 0.1466 - val_accuracy: 0.9835
Epoch 59/100
469/469 [==============================] - 9s 20ms/step - loss: 6.2068e-09 - accuracy:
1.0000 - val_loss: 0.1470 - val_accuracy: 0.9833
Epoch 60/100
469/469 [==============================] - 10s 21ms/step - loss: 6.0598e-09 - accuracy:
1.0000 - val_loss: 0.1469 - val_accuracy: 0.9836
Epoch 61/100
469/469 [==============================] - 9s 20ms/step - loss: 6.0618e-09 - accuracy:
1.0000 - val_loss: 0.1474 - val_accuracy: 0.9837
Epoch 62/100
469/469 [==============================] - 9s 20ms/step - loss: 5.8949e-09 - accuracy:
1.0000 - val_loss: 0.1473 - val_accuracy: 0.9833
Epoch 63/100
469/469 [==============================] - 12s 25ms/step - loss: 5.7836e-09 - accuracy:
1.0000 - val_loss: 0.1479 - val_accuracy: 0.9833
Epoch 64/100
469/469 [==============================] - 11s 24ms/step - loss: 5.7399e-09 - accuracy:
1.0000 - val_loss: 0.1479 - val_accuracy: 0.9833
Epoch 65/100
```

```
469/469 [==============================] - 11s 23ms/step - loss: 5.5810e-09 - accuracy: 1.0000 - val_loss: 0.1483 - val_accuracy: 0.9831
Epoch 66/100
469/469 [==============================] - 15s 31ms/step - loss: 5.5114e-09 - accuracy: 1.0000 - val_loss: 0.1486 - val_accuracy: 0.9832
Epoch 67/100
469/469 [==============================] - 13s 28ms/step - loss: 5.4240e-09 - accuracy: 1.0000 - val_loss: 0.1489 - val_accuracy: 0.9832
Epoch 68/100
469/469 [==============================] - 15s 31ms/step - loss: 5.4161e-09 - accuracy: 1.0000 - val_loss: 0.1490 - val_accuracy: 0.9832
Epoch 69/100
469/469 [==============================] - 17s 35ms/step - loss: 5.3465e-09 - accuracy: 1.0000 - val_loss: 0.1490 - val_accuracy: 0.9830
Epoch 70/100
469/469 [==============================] - 18s 38ms/step - loss: 5.2830e-09 - accuracy: 1.0000 - val_loss: 0.1493 - val_accuracy: 0.9834
Epoch 71/100
469/469 [==============================] - 18s 38ms/step - loss: 5.2293e-09 - accuracy: 1.0000 - val_loss: 0.1495 - val_accuracy: 0.9831
Epoch 72/100
469/469 [==============================] - 13s 28ms/step - loss: 5.1419e-09 - accuracy: 1.0000 - val_loss: 0.1496 - val_accuracy: 0.9833
Epoch 73/100
469/469 [==============================] - 12s 25ms/step - loss: 5.0704e-09 - accuracy: 1.0000 - val_loss: 0.1498 - val_accuracy: 0.9833
Epoch 74/100
469/469 [==============================] - 10s 22ms/step - loss: 5.0167e-09 - accuracy: 1.0000 - val_loss: 0.1501 - val_accuracy: 0.9835
Epoch 75/100
469/469 [==============================] - 10s 21ms/step - loss: 5.0247e-09 - accuracy: 1.0000 - val_loss: 0.1501 - val_accuracy: 0.9834
Epoch 76/100
469/469 [==============================] - 10s 22ms/step - loss: 4.9651e-09 - accuracy: 1.0000 - val_loss: 0.1503 - val_accuracy: 0.9834
Epoch 77/100
469/469 [==============================] - 11s 23ms/step - loss: 4.9015e-09 - accuracy: 1.0000 - val_loss: 0.1506 - val_accuracy: 0.9832
Epoch 78/100
469/469 [==============================] - 10s 21ms/step - loss: 4.8021e-09 - accuracy: 1.0000 - val_loss: 0.1508 - val_accuracy: 0.9834
Epoch 79/100
469/469 [==============================] - 10s 21ms/step - loss: 4.7624e-09 - accuracy: 1.0000 - val_loss: 0.1509 - val_accuracy: 0.9834
Epoch 80/100
469/469 [==============================] - 10s 20ms/step - loss: 4.7505e-09 - accuracy: 1.0000 - val_loss: 0.1512 - val_accuracy: 0.9831
Epoch 81/100
469/469 [==============================] - 10s 20ms/step - loss: 4.7465e-09 - accuracy: 1.0000 - val_loss: 0.1514 - val_accuracy: 0.9830
Epoch 82/100
469/469 [==============================] - 10s 21ms/step - loss: 4.6949e-09 - accuracy: 1.0000 - val_loss: 0.1515 - val_accuracy: 0.9833
Epoch 83/100
469/469 [==============================] - 11s 23ms/step - loss: 4.6114e-09 - accuracy: 1.0000 - val_loss: 0.1518 - val_accuracy: 0.9831
Epoch 84/100
469/469 [==============================] - 10s 21ms/step - loss: 4.6154e-09 - accuracy: 1.0000 - val_loss: 0.1516 - val_accuracy: 0.9833
Epoch 85/100
```

```
469/469 [==============================] - 10s 21ms/step - loss: 4.5915e-09 - accuracy:
1.0000 - val_loss: 0.1521 - val_accuracy: 0.9832
Epoch 86/100
469/469 [==============================] - 10s 20ms/step - loss: 4.5240e-09 - accuracy:
1.0000 - val_loss: 0.1520 - val_accuracy: 0.9834
Epoch 87/100
469/469 [==============================] - 10s 21ms/step - loss: 4.4862e-09 - accuracy:
1.0000 - val_loss: 0.1522 - val_accuracy: 0.9835
Epoch 88/100
469/469 [==============================] - 12s 26ms/step - loss: 4.4763e-09 - accuracy:
1.0000 - val_loss: 0.1526 - val_accuracy: 0.9831
Epoch 89/100
469/469 [==============================] - 13s 27ms/step - loss: 4.4703e-09 - accuracy:
1.0000 - val_loss: 0.1527 - val_accuracy: 0.9831
Epoch 90/100
469/469 [==============================] - 11s 23ms/step - loss: 4.4306e-09 - accuracy:
1.0000 - val_loss: 0.1529 - val_accuracy: 0.9830
Epoch 91/100
469/469 [==============================] - 11s 23ms/step - loss: 4.4386e-09 - accuracy:
1.0000 - val_loss: 0.1529 - val_accuracy: 0.9831
Epoch 92/100
469/469 [==============================] - 11s 23ms/step - loss: 4.4028e-09 - accuracy:
1.0000 - val_loss: 0.1532 - val_accuracy: 0.9833
Epoch 93/100
469/469 [==============================] - 11s 23ms/step - loss: 4.3313e-09 - accuracy:
1.0000 - val_loss: 0.1533 - val_accuracy: 0.9831
Epoch 94/100
469/469 [==============================] - 11s 24ms/step - loss: 4.3511e-09 - accuracy:
1.0000 - val_loss: 0.1535 - val_accuracy: 0.9831
Epoch 95/100
469/469 [==============================] - 12s 26ms/step - loss: 4.3650e-09 - accuracy:
1.0000 - val_loss: 0.1535 - val_accuracy: 0.9834
Epoch 96/100
469/469 [==============================] - 11s 24ms/step - loss: 4.3035e-09 - accuracy:
1.0000 - val_loss: 0.1537 - val_accuracy: 0.9831
Epoch 97/100
469/469 [==============================] - 11s 23ms/step - loss: 4.3015e-09 - accuracy:
1.0000 - val_loss: 0.1538 - val_accuracy: 0.9832
Epoch 98/100
469/469 [==============================] - 10s 22ms/step - loss: 4.2657e-09 - accuracy:
1.0000 - val_loss: 0.1542 - val_accuracy: 0.9831
Epoch 99/100
469/469 [==============================] - 11s 24ms/step - loss: 4.2319e-09 - accuracy:
1.0000 - val_loss: 0.1542 - val_accuracy: 0.9833
Epoch 100/100
469/469 [==============================] - 13s 27ms/step - loss: 4.2816e-09 - accuracy:
1.0000 - val_loss: 0.1545 - val_accuracy: 0.9830
```

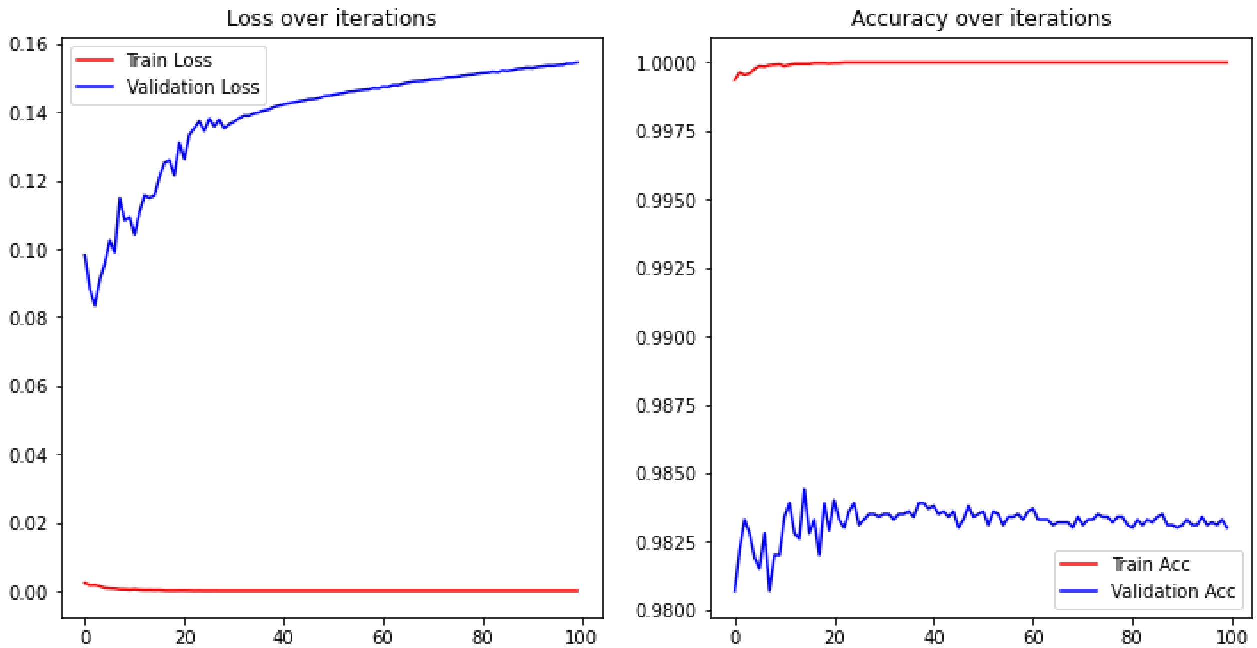
In [34]:

```
n = len(run_hist_nn1.history["loss"])

fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot(1, 2, 1)
ax.plot(range(n), (run_hist_nn1.history["loss"]),'r', label="Train Loss")
ax.plot(range(n), (run_hist_nn1.history["val_loss"]),'b', label="Validation Loss")
ax.legend()
ax.set_title('Loss over iterations')

ax = fig.add_subplot(1, 2, 2)
ax.plot(range(n), (run_hist_nn1.history["accuracy"]),'r', label="Train Acc")
```

```
ax.plot(range(n), (run_hist_nn1.history["val_accuracy"]),'b', label="Validation Acc")
ax.legend(loc='lower right')
ax.set_title('Accuracy over iterations')
```

Out[34]:

```
Text(0.5, 1.0, 'Accuracy over iterations')
```



Model starts plateauing after 15 epochs, retrain for 15 epochs and evaluate on test set:

In [21]:

```
network = Sequential()
network.add(Dense(512,activation='relu',input_shape=(28*28,)))
network.add(Dense(10,activation='softmax'))
network.compile(optimizer='rmsprop', loss='categorical_crossentropy',metrics=['accuracy
network.fit(x_train,train_labels,epochs=15,batch_size=128,validation_data=(x_test,test_
test_loss,test_acc = network.evaluate(x_test,test_labels)
print('test accuracy: ',test_acc)
```

```
Epoch 1/15
469/469 [==============================] - 11s 21ms/step - loss: 0.2571 - accuracy: 0.92
62 - val_loss: 0.1369 - val_accuracy: 0.9577
Epoch 2/15
469/469 [==============================] - 9s 20ms/step - loss: 0.1026 - accuracy: 0.969
7 - val_loss: 0.0910 - val_accuracy: 0.9733
Epoch 3/15
469/469 [==============================] - 11s 23ms/step - loss: 0.0681 - accuracy: 0.97
96 - val_loss: 0.0705 - val_accuracy: 0.9785
Epoch 4/15
469/469 [==============================] - 11s 24ms/step - loss: 0.0494 - accuracy: 0.98
50 - val_loss: 0.0691 - val_accuracy: 0.9790
Epoch 5/15
469/469 [==============================] - 9s 19ms/step - loss: 0.0370 - accuracy: 0.988
9 - val_loss: 0.0656 - val_accuracy: 0.9794
Epoch 6/15
469/469 [==============================] - 11s 25ms/step - loss: 0.0282 - accuracy: 0.99
13 - val_loss: 0.0628 - val_accuracy: 0.9816
Epoch 7/15
469/469 [==============================] - 9s 20ms/step - loss: 0.0214 - accuracy: 0.993
8 - val_loss: 0.0609 - val_accuracy: 0.9824
Epoch 8/15
469/469 [==============================] - 10s 20ms/step - loss: 0.0167 - accuracy: 0.99
```

```
49 - val_loss: 0.0666 - val_accuracy: 0.9818
Epoch 9/15
469/469 [==============================] - 9s 19ms/step - loss: 0.0126 - accuracy: 0.996
7 - val_loss: 0.0698 - val_accuracy: 0.9810
Epoch 10/15
469/469 [==============================] - 10s 21ms/step - loss: 0.0102 - accuracy: 0.99
73 - val_loss: 0.0678 - val_accuracy: 0.9827
Epoch 11/15
469/469 [==============================] - 9s 19ms/step - loss: 0.0073 - accuracy: 0.998
2 - val_loss: 0.0769 - val_accuracy: 0.9817
Epoch 12/15
469/469 [==============================] - 9s 20ms/step - loss: 0.0059 - accuracy: 0.998
3 - val_loss: 0.0744 - val_accuracy: 0.9818
Epoch 13/15
469/469 [==============================] - 10s 21ms/step - loss: 0.0046 - accuracy: 0.99
89 - val_loss: 0.0736 - val_accuracy: 0.9847
Epoch 14/15
469/469 [==============================] - 9s 20ms/step - loss: 0.0037 - accuracy: 0.999
0 - val_loss: 0.0837 - val_accuracy: 0.9804
Epoch 15/15
469/469 [==============================] - 9s 19ms/step - loss: 0.0027 - accuracy: 0.999
1 - val_loss: 0.0898 - val_accuracy: 0.9811
313/313 [==============================] - 2s 5ms/step - loss: 0.0898 - accuracy: 0.9811
test accuracy:  0.9811000227928162
```

In [22]:

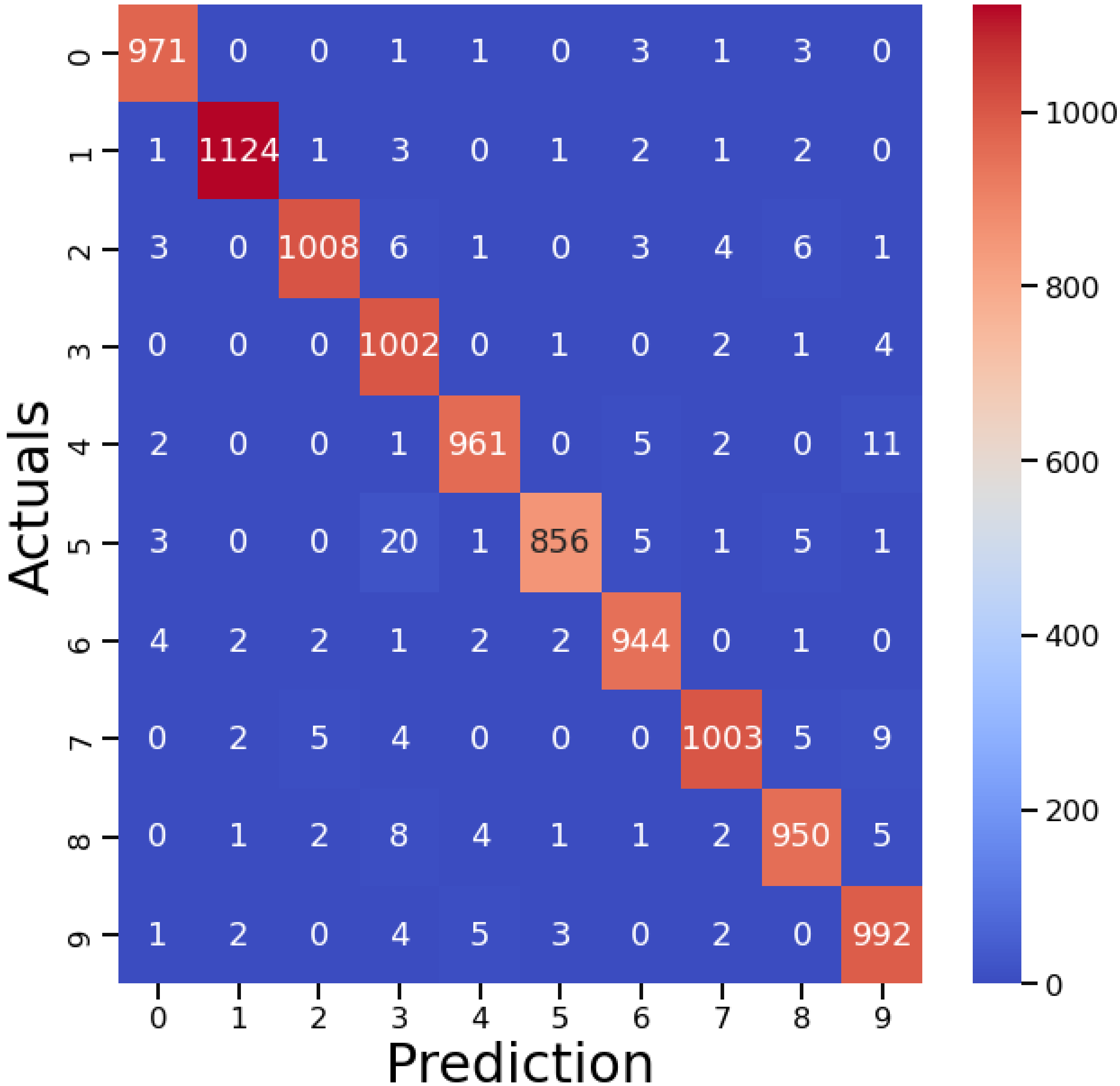
```
## we generate two kinds of predictions
#  One is a hard decision, the other is a probabilitistic score.

y_pred_prob_nn_1 = network.predict(x_test)
y_pred_class_nn_1 = np.argmax(y_pred_prob_nn_1,axis=1)
```

In [23]:

```
sns.set_context('talk')
cm = confusion_matrix(y_test, y_pred_class_nn_1)
_, ax = plt.subplots(figsize=(10,10))
ax = sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm')

ax.set_ylabel('Actuals', fontsize=30);
ax.set_xlabel('Prediction', fontsize=30)
```

Out[23]:

```
Text(0.5, 58.5, 'Prediction')
```



This model seems to be struggling with predicting 4, 5 and 8. The model also starts overfitting.

Model 2: Dense Deep Learning Model with Dropout

In [38]:

```
network = Sequential()
network.add(Dense(512,activation='relu',input_shape=(28*28,)))
network.add(Dropout(0.5))
network.add(Dense(10,activation='softmax'))
network.compile(optimizer='rmsprop', loss='categorical_crossentropy',metrics=['accuracy
run_hist_nn2=network.fit(x_train,train_labels,epochs=100,batch_size=128,validation_data
y_pred_prob_nn_2 = network.predict(x_test)
y_pred_class_nn_2 = np.argmax(y_pred_prob_nn_2,axis=1)
```

```
Epoch 1/100
469/469 [==============================] - 20s 38ms/step - loss: 0.3156 - accuracy: 0.90
71 - val_loss: 0.1443 - val_accuracy: 0.9578
Epoch 2/100
469/469 [==============================] - 18s 39ms/step - loss: 0.1542 - accuracy: 0.95
43 - val_loss: 0.1011 - val_accuracy: 0.9696
```

```
Epoch 3/100
469/469 [==============================] - 16s 35ms/step - loss: 0.1175 - accuracy: 0.96
52 - val_loss: 0.0886 - val_accuracy: 0.9732
Epoch 4/100
469/469 [==============================] - 18s 38ms/step - loss: 0.0997 - accuracy: 0.97
08 - val_loss: 0.0788 - val_accuracy: 0.9760
Epoch 5/100
469/469 [==============================] - 18s 38ms/step - loss: 0.0860 - accuracy: 0.97
39 - val_loss: 0.0742 - val_accuracy: 0.9783
Epoch 6/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0786 - accuracy: 0.97
71 - val_loss: 0.0736 - val_accuracy: 0.9795
Epoch 7/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0706 - accuracy: 0.97
88 - val_loss: 0.0740 - val_accuracy: 0.9788
Epoch 8/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0668 - accuracy: 0.98
03 - val_loss: 0.0732 - val_accuracy: 0.9809
Epoch 9/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0609 - accuracy: 0.98
22 - val_loss: 0.0717 - val_accuracy: 0.9802
Epoch 10/100
469/469 [==============================] - 13s 28ms/step - loss: 0.0584 - accuracy: 0.98
26 - val_loss: 0.0698 - val_accuracy: 0.9821
Epoch 11/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0525 - accuracy: 0.98
43 - val_loss: 0.0709 - val_accuracy: 0.9817
Epoch 12/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0514 - accuracy: 0.98
51 - val_loss: 0.0738 - val_accuracy: 0.9806
Epoch 13/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0492 - accuracy: 0.98
55 - val_loss: 0.0736 - val_accuracy: 0.9819
Epoch 14/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0460 - accuracy: 0.98
61 - val_loss: 0.0755 - val_accuracy: 0.9822
Epoch 15/100
469/469 [==============================] - 13s 28ms/step - loss: 0.0442 - accuracy: 0.98
66 - val_loss: 0.0791 - val_accuracy: 0.9824
Epoch 16/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0441 - accuracy: 0.98
69 - val_loss: 0.0736 - val_accuracy: 0.9821
Epoch 17/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0433 - accuracy: 0.98
73 - val_loss: 0.0733 - val_accuracy: 0.9836
Epoch 18/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0393 - accuracy: 0.98
77 - val_loss: 0.0790 - val_accuracy: 0.9831
Epoch 19/100
469/469 [==============================] - 14s 29ms/step - loss: 0.0382 - accuracy: 0.98
87 - val_loss: 0.0737 - val_accuracy: 0.9825
Epoch 20/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0366 - accuracy: 0.98
97 - val_loss: 0.0801 - val_accuracy: 0.9827
Epoch 21/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0357 - accuracy: 0.98
92 - val_loss: 0.0800 - val_accuracy: 0.9828
Epoch 22/100
469/469 [==============================] - 11s 25ms/step - loss: 0.0361 - accuracy: 0.98
98 - val_loss: 0.0748 - val_accuracy: 0.9841
```

```
Epoch 23/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0346 - accuracy: 0.99
01 - val_loss: 0.0823 - val_accuracy: 0.9830
Epoch 24/100
469/469 [==============================] - 13s 28ms/step - loss: 0.0348 - accuracy: 0.99
01 - val_loss: 0.0735 - val_accuracy: 0.9837
Epoch 25/100
469/469 [==============================] - 14s 29ms/step - loss: 0.0332 - accuracy: 0.99
07 - val_loss: 0.0774 - val_accuracy: 0.9834
Epoch 26/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0317 - accuracy: 0.99
07 - val_loss: 0.0807 - val_accuracy: 0.9839
Epoch 27/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0305 - accuracy: 0.99
08 - val_loss: 0.0860 - val_accuracy: 0.9828
Epoch 28/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0300 - accuracy: 0.99
10 - val_loss: 0.0837 - val_accuracy: 0.9842
Epoch 29/100
469/469 [==============================] - 14s 29ms/step - loss: 0.0306 - accuracy: 0.99
15 - val_loss: 0.0821 - val_accuracy: 0.9834
Epoch 30/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0274 - accuracy: 0.99
17 - val_loss: 0.0843 - val_accuracy: 0.9845
Epoch 31/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0277 - accuracy: 0.99
18 - val_loss: 0.0837 - val_accuracy: 0.9834
Epoch 32/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0276 - accuracy: 0.99
18 - val_loss: 0.0835 - val_accuracy: 0.9845
Epoch 33/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0253 - accuracy: 0.99
26 - val_loss: 0.0895 - val_accuracy: 0.9845
Epoch 34/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0265 - accuracy: 0.99
24 - val_loss: 0.0889 - val_accuracy: 0.9845
Epoch 35/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0241 - accuracy: 0.99
29 - val_loss: 0.0987 - val_accuracy: 0.9827
Epoch 36/100
469/469 [==============================] - 13s 29ms/step - loss: 0.0247 - accuracy: 0.99
26 - val_loss: 0.0919 - val_accuracy: 0.9844
Epoch 37/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0235 - accuracy: 0.99
33 - val_loss: 0.0933 - val_accuracy: 0.9846
Epoch 38/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0235 - accuracy: 0.99
32 - val_loss: 0.0909 - val_accuracy: 0.9842
Epoch 39/100
469/469 [==============================] - 12s 27ms/step - loss: 0.0227 - accuracy: 0.99
30 - val_loss: 0.0937 - val_accuracy: 0.9834
Epoch 40/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0230 - accuracy: 0.99
35 - val_loss: 0.0995 - val_accuracy: 0.9829
Epoch 41/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0231 - accuracy: 0.99
32 - val_loss: 0.1008 - val_accuracy: 0.9831
Epoch 42/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0262 - accuracy: 0.99
31 - val_loss: 0.1001 - val_accuracy: 0.9837
```

```
Epoch 43/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0214 - accuracy: 0.99
40 - val_loss: 0.0974 - val_accuracy: 0.9839
Epoch 44/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0224 - accuracy: 0.99
38 - val_loss: 0.0955 - val_accuracy: 0.9849
Epoch 45/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0223 - accuracy: 0.99
33 - val_loss: 0.0978 - val_accuracy: 0.9836
Epoch 46/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0199 - accuracy: 0.99
42 - val_loss: 0.1017 - val_accuracy: 0.9836
Epoch 47/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0210 - accuracy: 0.99
38 - val_loss: 0.0951 - val_accuracy: 0.9845
Epoch 48/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0206 - accuracy: 0.99
42 - val_loss: 0.0995 - val_accuracy: 0.9828
Epoch 49/100
469/469 [==============================] - 13s 29ms/step - loss: 0.0211 - accuracy: 0.99
40 - val_loss: 0.1041 - val_accuracy: 0.9836
Epoch 50/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0216 - accuracy: 0.99
40 - val_loss: 0.1048 - val_accuracy: 0.9845
Epoch 51/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0206 - accuracy: 0.99
45 - val_loss: 0.1021 - val_accuracy: 0.9850
Epoch 52/100
469/469 [==============================] - 13s 28ms/step - loss: 0.0213 - accuracy: 0.99
38 - val_loss: 0.1069 - val_accuracy: 0.9832
Epoch 53/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0199 - accuracy: 0.99
45 - val_loss: 0.1046 - val_accuracy: 0.9842
Epoch 54/100
469/469 [==============================] - 14s 30ms/step - loss: 0.0208 - accuracy: 0.99
44 - val_loss: 0.1042 - val_accuracy: 0.9834
Epoch 55/100
469/469 [==============================] - 13s 29ms/step - loss: 0.0205 - accuracy: 0.99
48 - val_loss: 0.1069 - val_accuracy: 0.9838
Epoch 56/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0170 - accuracy: 0.99
48 - val_loss: 0.1129 - val_accuracy: 0.9832
Epoch 57/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0199 - accuracy: 0.99
46 - val_loss: 0.1103 - val_accuracy: 0.9845
Epoch 58/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0172 - accuracy: 0.99
54 - val_loss: 0.1094 - val_accuracy: 0.9833
Epoch 59/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0177 - accuracy: 0.99
48 - val_loss: 0.1153 - val_accuracy: 0.9827
Epoch 60/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0185 - accuracy: 0.99
51 - val_loss: 0.1120 - val_accuracy: 0.9839
Epoch 61/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0205 - accuracy: 0.99
43 - val_loss: 0.1157 - val_accuracy: 0.9829
Epoch 62/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0154 - accuracy: 0.99
52 - val_loss: 0.1156 - val_accuracy: 0.9834
```

```
Epoch 63/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0194 - accuracy: 0.99
49 - val_loss: 0.1183 - val_accuracy: 0.9838
Epoch 64/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0172 - accuracy: 0.99
55 - val_loss: 0.1252 - val_accuracy: 0.9830
Epoch 65/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0173 - accuracy: 0.99
55 - val_loss: 0.1217 - val_accuracy: 0.9830
Epoch 66/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0185 - accuracy: 0.99
51 - val_loss: 0.1237 - val_accuracy: 0.9838
Epoch 67/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0172 - accuracy: 0.99
55 - val_loss: 0.1245 - val_accuracy: 0.9843
Epoch 68/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0172 - accuracy: 0.99
52 - val_loss: 0.1225 - val_accuracy: 0.9834
Epoch 69/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0157 - accuracy: 0.99
59 - val_loss: 0.1262 - val_accuracy: 0.9845
Epoch 70/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0188 - accuracy: 0.99
54 - val_loss: 0.1270 - val_accuracy: 0.9834
Epoch 71/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0163 - accuracy: 0.99
57 - val_loss: 0.1183 - val_accuracy: 0.9840
Epoch 72/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0160 - accuracy: 0.99
56 - val_loss: 0.1217 - val_accuracy: 0.9842
Epoch 73/100
469/469 [==============================] - 16s 35ms/step - loss: 0.0151 - accuracy: 0.99
55 - val_loss: 0.1256 - val_accuracy: 0.9838
Epoch 74/100
469/469 [==============================] - 17s 35ms/step - loss: 0.0154 - accuracy: 0.99
58 - val_loss: 0.1267 - val_accuracy: 0.9842
Epoch 75/100
469/469 [==============================] - 14s 30ms/step - loss: 0.0160 - accuracy: 0.99
56 - val_loss: 0.1233 - val_accuracy: 0.9838
Epoch 76/100
469/469 [==============================] - 23s 49ms/step - loss: 0.0151 - accuracy: 0.99
57 - val_loss: 0.1303 - val_accuracy: 0.9847
Epoch 77/100
469/469 [==============================] - 21s 44ms/step - loss: 0.0166 - accuracy: 0.99
54 - val_loss: 0.1338 - val_accuracy: 0.9831
Epoch 78/100
469/469 [==============================] - 18s 39ms/step - loss: 0.0163 - accuracy: 0.99
58 - val_loss: 0.1285 - val_accuracy: 0.9833
Epoch 79/100
469/469 [==============================] - 13s 28ms/step - loss: 0.0182 - accuracy: 0.99
54 - val_loss: 0.1229 - val_accuracy: 0.9838
Epoch 80/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0163 - accuracy: 0.99
56 - val_loss: 0.1313 - val_accuracy: 0.9837
Epoch 81/100
469/469 [==============================] - 13s 28ms/step - loss: 0.0151 - accuracy: 0.99
58 - val_loss: 0.1248 - val_accuracy: 0.9844
Epoch 82/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0149 - accuracy: 0.99
61 - val_loss: 0.1280 - val_accuracy: 0.9837
```

```
Epoch 83/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0155 - accuracy: 0.99
58 - val_loss: 0.1282 - val_accuracy: 0.9845
Epoch 84/100
469/469 [==============================] - 12s 26ms/step - loss: 0.0147 - accuracy: 0.99
57 - val_loss: 0.1289 - val_accuracy: 0.9841
Epoch 85/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0175 - accuracy: 0.99
56 - val_loss: 0.1228 - val_accuracy: 0.9843
Epoch 86/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0147 - accuracy: 0.99
57 - val_loss: 0.1303 - val_accuracy: 0.9840
Epoch 87/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0166 - accuracy: 0.99
58 - val_loss: 0.1273 - val_accuracy: 0.9850
Epoch 88/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0164 - accuracy: 0.99
57 - val_loss: 0.1407 - val_accuracy: 0.9831
Epoch 89/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0168 - accuracy: 0.99
59 - val_loss: 0.1345 - val_accuracy: 0.9834
Epoch 90/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0150 - accuracy: 0.99
59 - val_loss: 0.1324 - val_accuracy: 0.9848
Epoch 91/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0145 - accuracy: 0.99
64 - val_loss: 0.1380 - val_accuracy: 0.9833
Epoch 92/100
469/469 [==============================] - 12s 27ms/step - loss: 0.0157 - accuracy: 0.99
61 - val_loss: 0.1348 - val_accuracy: 0.9842
Epoch 93/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0154 - accuracy: 0.99
59 - val_loss: 0.1395 - val_accuracy: 0.9837
Epoch 94/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0167 - accuracy: 0.99
61 - val_loss: 0.1361 - val_accuracy: 0.9834
Epoch 95/100
469/469 [==============================] - 12s 25ms/step - loss: 0.0147 - accuracy: 0.99
61 - val_loss: 0.1357 - val_accuracy: 0.9836
Epoch 96/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0146 - accuracy: 0.99
63 - val_loss: 0.1350 - val_accuracy: 0.9850
Epoch 97/100
469/469 [==============================] - 13s 27ms/step - loss: 0.0134 - accuracy: 0.99
62 - val_loss: 0.1332 - val_accuracy: 0.9840
Epoch 98/100
469/469 [==============================] - 11s 24ms/step - loss: 0.0144 - accuracy: 0.99
61 - val_loss: 0.1408 - val_accuracy: 0.9839
Epoch 99/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0142 - accuracy: 0.99
64 - val_loss: 0.1413 - val_accuracy: 0.9835
Epoch 100/100
469/469 [==============================] - 11s 23ms/step - loss: 0.0161 - accuracy: 0.99
62 - val_loss: 0.1394 - val_accuracy: 0.9843
```

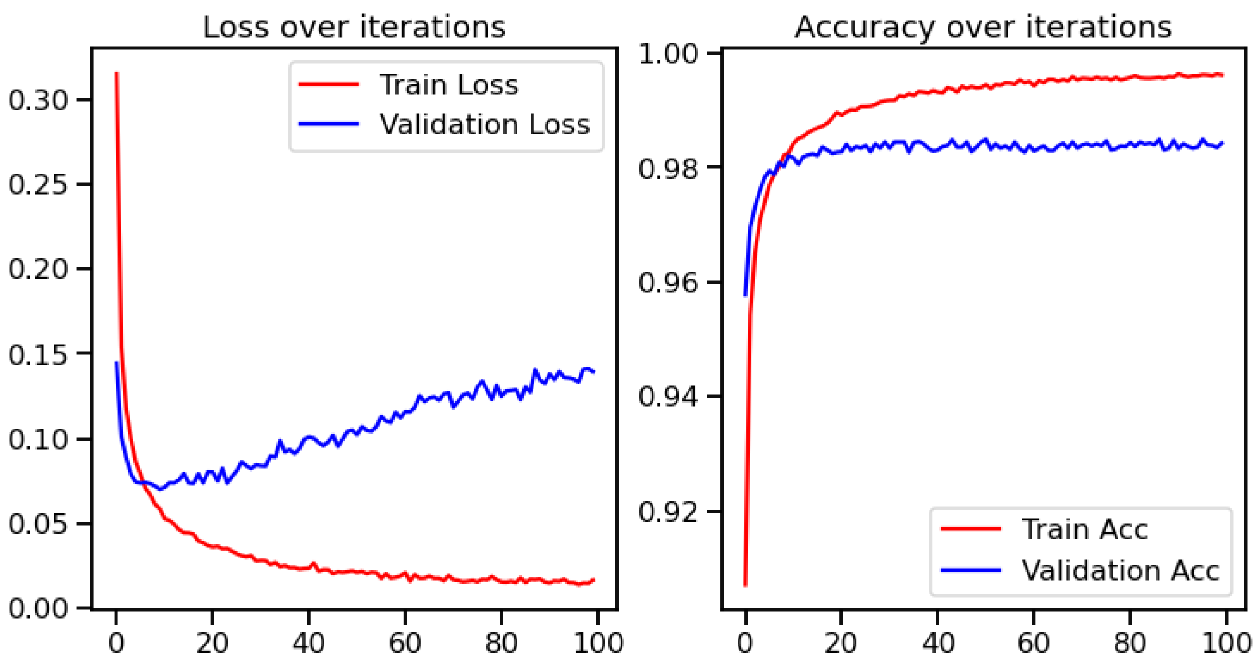
In [39]:

```
n = len(run_hist_nn2.history["loss"])

fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot(1, 2, 1)
```

```
ax.plot(range(n), (run_hist_nn2.history["loss"]),'r', label="Train Loss")
ax.plot(range(n), (run_hist_nn2.history["val_loss"]),'b', label="Validation Loss")
ax.legend()
ax.set_title('Loss over iterations')

ax = fig.add_subplot(1, 2, 2)
ax.plot(range(n), (run_hist_nn2.history["accuracy"]),'r', label="Train Acc")
ax.plot(range(n), (run_hist_nn2.history["val_accuracy"]),'b', label="Validation Acc")
ax.legend(loc='lower right')
ax.set_title('Accuracy over iterations')
```

Out[39]:

```
Text(0.5, 1.0, 'Accuracy over iterations')
```



After 44 epochs there isn't much improvement in the validation accuracy, retrain model for 44 epochs and evaluate on test set:

In [24]:

```
network = Sequential()
network.add(Dense(512,activation='relu',input_shape=(28*28,)))
network.add(Dropout(0.5))
network.add(Dense(10,activation='softmax'))
network.compile(optimizer='rmsprop', loss='categorical_crossentropy',metrics=['accuracy
run_hist_nn2=network.fit(x_train,train_labels,epochs=44,batch_size=128,validation_data=
y_pred_prob_nn_2 = network.predict(x_test)
y_pred_class_nn_2 = np.argmax(y_pred_prob_nn_2,axis=1)
```

```
Epoch 1/44
469/469 [==============================] - 13s 25ms/step - loss: 0.3163 - accuracy: 0.90
81 - val_loss: 0.1451 - val_accuracy: 0.9564
Epoch 2/44
469/469 [==============================] - 11s 24ms/step - loss: 0.1540 - accuracy: 0.95
43 - val_loss: 0.1073 - val_accuracy: 0.9677
Epoch 3/44
469/469 [==============================] - 10s 22ms/step - loss: 0.1166 - accuracy: 0.96
46 - val_loss: 0.0904 - val_accuracy: 0.9729
Epoch 4/44
469/469 [==============================] - 10s 22ms/step - loss: 0.1006 - accuracy: 0.97
01 - val_loss: 0.0833 - val_accuracy: 0.9752
Epoch 5/44
```

```
469/469 [==============================] - 10s 22ms/step - loss: 0.0857 - accuracy: 0.97
42 - val_loss: 0.0789 - val_accuracy: 0.9780
Epoch 6/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0788 - accuracy: 0.97
62 - val_loss: 0.0745 - val_accuracy: 0.9792
Epoch 7/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0738 - accuracy: 0.97
84 - val_loss: 0.0708 - val_accuracy: 0.9802
Epoch 8/44
469/469 [==============================] - 12s 26ms/step - loss: 0.0656 - accuracy: 0.98
06 - val_loss: 0.0720 - val_accuracy: 0.9807
Epoch 9/44
469/469 [==============================] - 11s 22ms/step - loss: 0.0611 - accuracy: 0.98
12 - val_loss: 0.0646 - val_accuracy: 0.9818
Epoch 10/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0574 - accuracy: 0.98
26 - val_loss: 0.0701 - val_accuracy: 0.9810
Epoch 11/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0543 - accuracy: 0.98
39 - val_loss: 0.0671 - val_accuracy: 0.9830
Epoch 12/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0521 - accuracy: 0.98
42 - val_loss: 0.0694 - val_accuracy: 0.9813
Epoch 13/44
469/469 [==============================] - 13s 28ms/step - loss: 0.0498 - accuracy: 0.98
55 - val_loss: 0.0760 - val_accuracy: 0.9811
Epoch 14/44
469/469 [==============================] - 14s 31ms/step - loss: 0.0480 - accuracy: 0.98
59 - val_loss: 0.0736 - val_accuracy: 0.9821
Epoch 15/44
469/469 [==============================] - 12s 24ms/step - loss: 0.0435 - accuracy: 0.98
72 - val_loss: 0.0751 - val_accuracy: 0.9823
Epoch 16/44
469/469 [==============================] - 12s 26ms/step - loss: 0.0425 - accuracy: 0.98
74 - val_loss: 0.0721 - val_accuracy: 0.9831
Epoch 17/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0429 - accuracy: 0.98
69 - val_loss: 0.0727 - val_accuracy: 0.9828
Epoch 18/44
469/469 [==============================] - 11s 24ms/step - loss: 0.0424 - accuracy: 0.98
78 - val_loss: 0.0735 - val_accuracy: 0.9832
Epoch 19/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0391 - accuracy: 0.98
89 - val_loss: 0.0761 - val_accuracy: 0.9827
Epoch 20/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0373 - accuracy: 0.98
85 - val_loss: 0.0748 - val_accuracy: 0.9839
Epoch 21/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0364 - accuracy: 0.98
91 - val_loss: 0.0738 - val_accuracy: 0.9844
Epoch 22/44
469/469 [==============================] - 13s 27ms/step - loss: 0.0348 - accuracy: 0.99
00 - val_loss: 0.0801 - val_accuracy: 0.9825
Epoch 23/44
469/469 [==============================] - 12s 26ms/step - loss: 0.0322 - accuracy: 0.99
04 - val_loss: 0.0789 - val_accuracy: 0.9836
Epoch 24/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0323 - accuracy: 0.98
95 - val_loss: 0.0820 - val_accuracy: 0.9826
Epoch 25/44
```

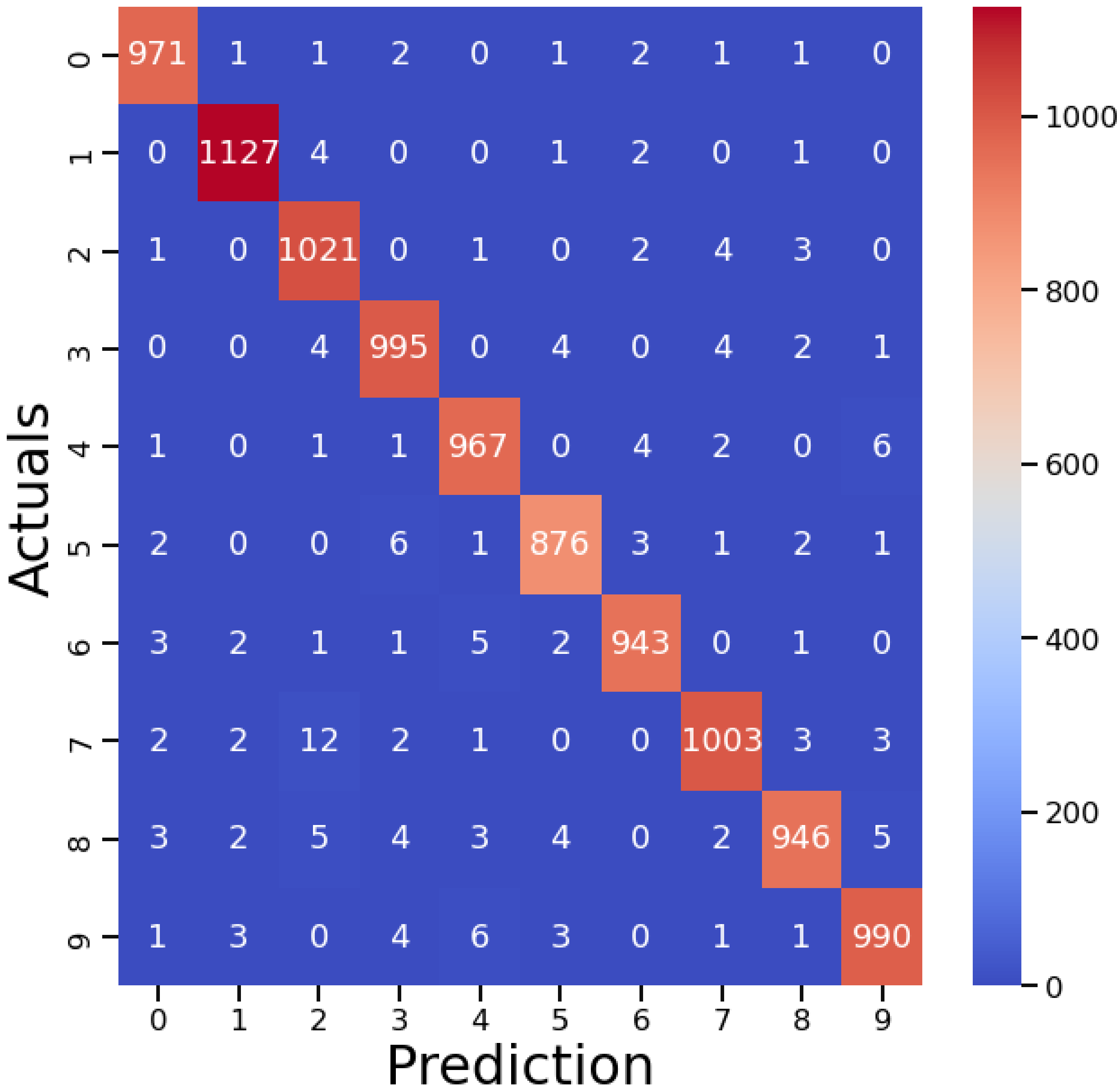
```
469/469 [==============================] - 11s 23ms/step - loss: 0.0338 - accuracy: 0.99
05 - val_loss: 0.0872 - val_accuracy: 0.9834
Epoch 26/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0308 - accuracy: 0.99
09 - val_loss: 0.0820 - val_accuracy: 0.9834
Epoch 27/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0297 - accuracy: 0.99
13 - val_loss: 0.0851 - val_accuracy: 0.9827
Epoch 28/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0299 - accuracy: 0.99
12 - val_loss: 0.0894 - val_accuracy: 0.9832
Epoch 29/44
469/469 [==============================] - 12s 25ms/step - loss: 0.0290 - accuracy: 0.99
19 - val_loss: 0.0860 - val_accuracy: 0.9821
Epoch 30/44
469/469 [==============================] - 10s 21ms/step - loss: 0.0288 - accuracy: 0.99
18 - val_loss: 0.0848 - val_accuracy: 0.9839
Epoch 31/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0265 - accuracy: 0.99
24 - val_loss: 0.0926 - val_accuracy: 0.9839
Epoch 32/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0245 - accuracy: 0.99
23 - val_loss: 0.0903 - val_accuracy: 0.9828
Epoch 33/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0286 - accuracy: 0.99
22 - val_loss: 0.0911 - val_accuracy: 0.9839
Epoch 34/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0260 - accuracy: 0.99
26 - val_loss: 0.0852 - val_accuracy: 0.9836
Epoch 35/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0259 - accuracy: 0.99
21 - val_loss: 0.0951 - val_accuracy: 0.9837
Epoch 36/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0255 - accuracy: 0.99
29 - val_loss: 0.0907 - val_accuracy: 0.9848
Epoch 37/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0247 - accuracy: 0.99
29 - val_loss: 0.0954 - val_accuracy: 0.9832
Epoch 38/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0249 - accuracy: 0.99
31 - val_loss: 0.0921 - val_accuracy: 0.9832
Epoch 39/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0243 - accuracy: 0.99
33 - val_loss: 0.0941 - val_accuracy: 0.9847
Epoch 40/44
469/469 [==============================] - 11s 24ms/step - loss: 0.0244 - accuracy: 0.99
29 - val_loss: 0.0960 - val_accuracy: 0.9832
Epoch 41/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0219 - accuracy: 0.99
39 - val_loss: 0.0969 - val_accuracy: 0.9832
Epoch 42/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0225 - accuracy: 0.99
34 - val_loss: 0.0933 - val_accuracy: 0.9843
Epoch 43/44
469/469 [==============================] - 10s 22ms/step - loss: 0.0234 - accuracy: 0.99
34 - val_loss: 0.0922 - val_accuracy: 0.9844
Epoch 44/44
469/469 [==============================] - 11s 23ms/step - loss: 0.0235 - accuracy: 0.99
33 - val_loss: 0.0976 - val_accuracy: 0.9839
```

In [25]:

```
sns.set_context('talk')
cm = confusion_matrix(y_test, y_pred_class_nn_2)
_, ax = plt.subplots(figsize=(10,10))
ax = sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm')

ax.set_ylabel('Actuals', fontsize=30);
ax.set_xlabel('Prediction', fontsize=30)
```

Out[25]:

```
Text(0.5, 58.5, 'Prediction')
```



In [26]:

```
test_loss,test_acc = network.evaluate(x_test,test_labels)
print('test accuracy: ',test_acc)
```

```
313/313 [==============================] - 2s 6ms/step - loss: 0.0976 - accuracy: 0.9839
test accuracy:  0.9839000105857849
```

Model 3: Convolutional Neural Network

Building a simple CNN to see if this will improve test accuracy

In [8]:

```
from keras import layers
from keras import models

network = models.Sequential()
network.add(layers.Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
network.add(layers.MaxPooling2D((2,2)))
network.add(layers.Conv2D(64,(3,3),activation='relu'))
network.add(layers.MaxPooling2D((2,2)))
network.add(layers.Conv2D(64,(3,3),activation='relu'))
network.add(layers.Flatten())
network.add(Dense(512,activation='relu'))
network.add(Dense(10,activation='softmax'))
network.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0         
 2D)                                                             
                                                                 
 conv2d_2 (Conv2D)           (None, 3, 3, 64)          36928     
                                                                 
 flatten (Flatten)           (None, 576)               0         
                                                                 
 dense_2 (Dense)             (None, 512)               295424    
                                                                 
 dense_3 (Dense)             (None, 10)                5130      
                                                                 
=================================================================
Total params: 356,298
Trainable params: 356,298
Non-trainable params: 0
_________________________________________________________________
```

In [10]:

```
x_train = train_images.reshape((60000,28,28,1))
x_train = x_train.astype('float32')/255

x_test = test_images.reshape((10000,28,28,1))
x_test = x_test.astype('float32')/255
```

In [11]:

```
network.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'
run_hist_nn3=network.fit(x_train,train_labels,epochs=100,batch_size=128,validation_data
y_pred_prob_nn_3 = network.predict(x_test)
y_pred_class_nn_3 = np.argmax(y_pred_prob_nn_3,axis=1)
```

```
Epoch 1/100
469/469 [==============================] - 76s 158ms/step - loss: 0.1880 - accuracy: 0.9
```

```
415 - val_loss: 0.0786 - val_accuracy: 0.9757
Epoch 2/100
469/469 [==============================] - 77s 165ms/step - loss: 0.0461 - accuracy: 0.9
853 - val_loss: 0.0510 - val_accuracy: 0.9841
Epoch 3/100
469/469 [==============================] - 74s 157ms/step - loss: 0.0304 - accuracy: 0.9
905 - val_loss: 0.0312 - val_accuracy: 0.9911
Epoch 4/100
469/469 [==============================] - 74s 157ms/step - loss: 0.0230 - accuracy: 0.9
930 - val_loss: 0.0373 - val_accuracy: 0.9888
Epoch 5/100
469/469 [==============================] - 68s 145ms/step - loss: 0.0172 - accuracy: 0.9
946 - val_loss: 0.0280 - val_accuracy: 0.9928
Epoch 6/100
469/469 [==============================] - 67s 143ms/step - loss: 0.0135 - accuracy: 0.9
959 - val_loss: 0.0359 - val_accuracy: 0.9905
Epoch 7/100
469/469 [==============================] - 69s 147ms/step - loss: 0.0109 - accuracy: 0.9
964 - val_loss: 0.0335 - val_accuracy: 0.9915
Epoch 8/100
469/469 [==============================] - 69s 146ms/step - loss: 0.0096 - accuracy: 0.9
969 - val_loss: 0.0397 - val_accuracy: 0.9907
Epoch 9/100
469/469 [==============================] - 71s 150ms/step - loss: 0.0077 - accuracy: 0.9
976 - val_loss: 0.0429 - val_accuracy: 0.9914
Epoch 10/100
469/469 [==============================] - 66s 141ms/step - loss: 0.0067 - accuracy: 0.9
979 - val_loss: 0.0442 - val_accuracy: 0.9913
Epoch 11/100
469/469 [==============================] - 68s 145ms/step - loss: 0.0059 - accuracy: 0.9
980 - val_loss: 0.0501 - val_accuracy: 0.9916
Epoch 12/100
469/469 [==============================] - 72s 153ms/step - loss: 0.0052 - accuracy: 0.9
982 - val_loss: 0.0415 - val_accuracy: 0.9926
Epoch 13/100
469/469 [==============================] - 72s 155ms/step - loss: 0.0047 - accuracy: 0.9
985 - val_loss: 0.0443 - val_accuracy: 0.9930
Epoch 14/100
469/469 [==============================] - 69s 147ms/step - loss: 0.0041 - accuracy: 0.9
988 - val_loss: 0.0469 - val_accuracy: 0.9930
Epoch 15/100
469/469 [==============================] - 72s 154ms/step - loss: 0.0040 - accuracy: 0.9
989 - val_loss: 0.0598 - val_accuracy: 0.9912
Epoch 16/100
469/469 [==============================] - 72s 154ms/step - loss: 0.0034 - accuracy: 0.9
988 - val_loss: 0.0641 - val_accuracy: 0.9917
Epoch 17/100
469/469 [==============================] - 73s 156ms/step - loss: 0.0036 - accuracy: 0.9
991 - val_loss: 0.0579 - val_accuracy: 0.9925
Epoch 18/100
469/469 [==============================] - 68s 145ms/step - loss: 0.0038 - accuracy: 0.9
990 - val_loss: 0.0663 - val_accuracy: 0.9905
Epoch 19/100
469/469 [==============================] - 71s 152ms/step - loss: 0.0031 - accuracy: 0.9
991 - val_loss: 0.0671 - val_accuracy: 0.9915
Epoch 20/100
469/469 [==============================] - 84s 180ms/step - loss: 0.0028 - accuracy: 0.9
991 - val_loss: 0.0677 - val_accuracy: 0.9913
Epoch 21/100
469/469 [==============================] - 87s 185ms/step - loss: 0.0023 - accuracy: 0.9
```

```
994 - val_loss: 0.0637 - val_accuracy: 0.9915
Epoch 22/100
469/469 [==============================] - 76s 162ms/step - loss: 0.0025 - accuracy: 0.9
993 - val_loss: 0.0693 - val_accuracy: 0.9909
Epoch 23/100
469/469 [==============================] - 77s 165ms/step - loss: 0.0025 - accuracy: 0.9
993 - val_loss: 0.0742 - val_accuracy: 0.9913
Epoch 24/100
469/469 [==============================] - 79s 169ms/step - loss: 0.0028 - accuracy: 0.9
993 - val_loss: 0.0787 - val_accuracy: 0.9915
Epoch 25/100
469/469 [==============================] - 79s 168ms/step - loss: 0.0022 - accuracy: 0.9
993 - val_loss: 0.0780 - val_accuracy: 0.9912
Epoch 26/100
469/469 [==============================] - 78s 166ms/step - loss: 0.0026 - accuracy: 0.9
994 - val_loss: 0.0923 - val_accuracy: 0.9916
Epoch 27/100
469/469 [==============================] - 75s 161ms/step - loss: 0.0023 - accuracy: 0.9
994 - val_loss: 0.0986 - val_accuracy: 0.9911
Epoch 28/100
469/469 [==============================] - 79s 169ms/step - loss: 0.0031 - accuracy: 0.9
992 - val_loss: 0.0845 - val_accuracy: 0.9915
Epoch 29/100
469/469 [==============================] - 78s 167ms/step - loss: 0.0022 - accuracy: 0.9
995 - val_loss: 0.1005 - val_accuracy: 0.9912
Epoch 30/100
469/469 [==============================] - 78s 166ms/step - loss: 0.0023 - accuracy: 0.9
994 - val_loss: 0.0820 - val_accuracy: 0.9922
Epoch 31/100
469/469 [==============================] - 84s 180ms/step - loss: 0.0023 - accuracy: 0.9
995 - val_loss: 0.0876 - val_accuracy: 0.9921
Epoch 32/100
469/469 [==============================] - 78s 167ms/step - loss: 0.0023 - accuracy: 0.9
995 - val_loss: 0.0930 - val_accuracy: 0.9925
Epoch 33/100
469/469 [==============================] - 76s 162ms/step - loss: 0.0021 - accuracy: 0.9
995 - val_loss: 0.0990 - val_accuracy: 0.9922
Epoch 34/100
469/469 [==============================] - 80s 170ms/step - loss: 0.0020 - accuracy: 0.9
996 - val_loss: 0.1002 - val_accuracy: 0.9915
Epoch 35/100
469/469 [==============================] - 86s 183ms/step - loss: 0.0019 - accuracy: 0.9
996 - val_loss: 0.0850 - val_accuracy: 0.9926
Epoch 36/100
469/469 [==============================] - 90s 191ms/step - loss: 0.0018 - accuracy: 0.9
996 - val_loss: 0.1037 - val_accuracy: 0.9929
Epoch 37/100
469/469 [==============================] - 79s 168ms/step - loss: 0.0025 - accuracy: 0.9
994 - val_loss: 0.1065 - val_accuracy: 0.9918
Epoch 38/100
469/469 [==============================] - 78s 167ms/step - loss: 0.0017 - accuracy: 0.9
996 - val_loss: 0.1307 - val_accuracy: 0.9918
Epoch 39/100
469/469 [==============================] - 79s 169ms/step - loss: 0.0018 - accuracy: 0.9
996 - val_loss: 0.1562 - val_accuracy: 0.9914
Epoch 40/100
469/469 [==============================] - 84s 179ms/step - loss: 0.0015 - accuracy: 0.9
996 - val_loss: 0.1410 - val_accuracy: 0.9915
Epoch 41/100
469/469 [==============================] - 50s 106ms/step - loss: 0.0015 - accuracy: 0.9
```

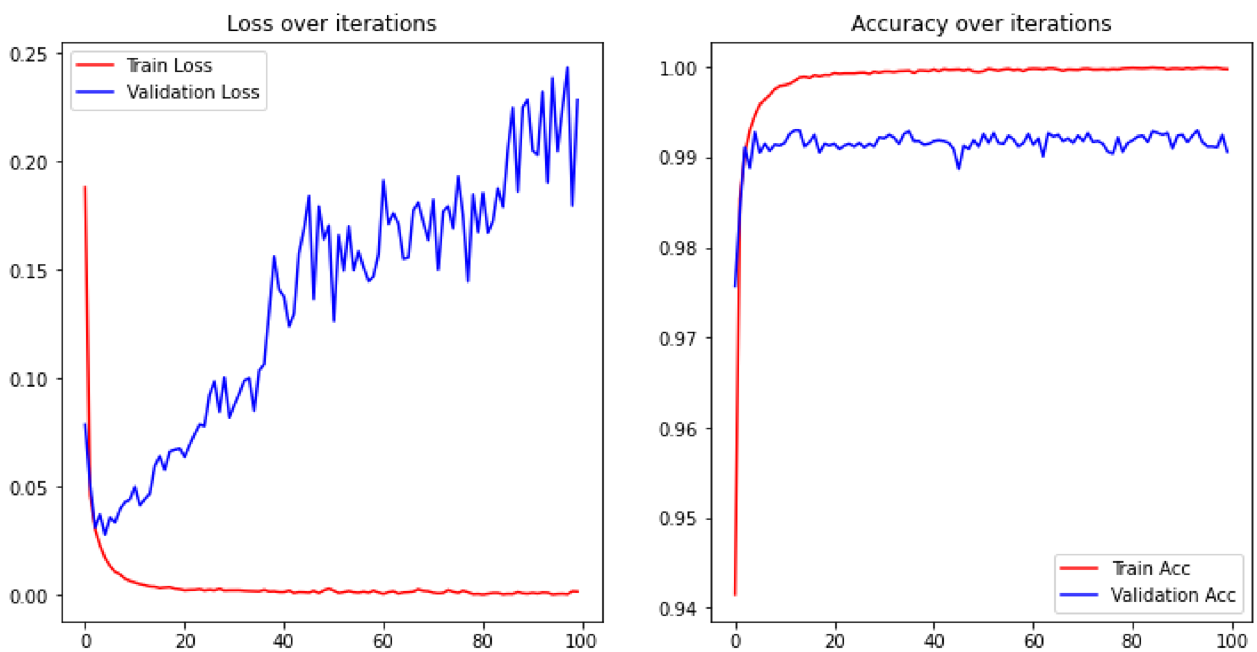
```
998 - val_loss: 0.1377 - val_accuracy: 0.9918
Epoch 42/100
469/469 [==============================] - 28s 61ms/step - loss: 0.0022 - accuracy: 0.99
95 - val_loss: 0.1240 - val_accuracy: 0.9919
Epoch 43/100
469/469 [==============================] - 29s 61ms/step - loss: 9.8487e-04 - accuracy:
0.9998 - val_loss: 0.1297 - val_accuracy: 0.9918
Epoch 44/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0015 - accuracy: 0.99
97 - val_loss: 0.1576 - val_accuracy: 0.9916
Epoch 45/100
469/469 [==============================] - 29s 61ms/step - loss: 0.0014 - accuracy: 0.99
97 - val_loss: 0.1688 - val_accuracy: 0.9909
Epoch 46/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0012 - accuracy: 0.99
98 - val_loss: 0.1841 - val_accuracy: 0.9887
Epoch 47/100
469/469 [==============================] - 29s 62ms/step - loss: 0.0021 - accuracy: 0.99
96 - val_loss: 0.1365 - val_accuracy: 0.9913
Epoch 48/100
469/469 [==============================] - 28s 61ms/step - loss: 9.6862e-04 - accuracy:
0.9998 - val_loss: 0.1793 - val_accuracy: 0.9909
Epoch 49/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0023 - accuracy: 0.99
95 - val_loss: 0.1641 - val_accuracy: 0.9919
Epoch 50/100
469/469 [==============================] - 28s 61ms/step - loss: 0.0031 - accuracy: 0.99
95 - val_loss: 0.1704 - val_accuracy: 0.9912
Epoch 51/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0021 - accuracy: 0.99
96 - val_loss: 0.1264 - val_accuracy: 0.9926
Epoch 52/100
469/469 [==============================] - 29s 61ms/step - loss: 0.0010 - accuracy: 0.99
98 - val_loss: 0.1660 - val_accuracy: 0.9916
Epoch 53/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0014 - accuracy: 0.99
97 - val_loss: 0.1496 - val_accuracy: 0.9927
Epoch 54/100
469/469 [==============================] - 29s 61ms/step - loss: 0.0020 - accuracy: 0.99
96 - val_loss: 0.1702 - val_accuracy: 0.9916
Epoch 55/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0014 - accuracy: 0.99
98 - val_loss: 0.1498 - val_accuracy: 0.9915
Epoch 56/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0012 - accuracy: 0.99
98 - val_loss: 0.1586 - val_accuracy: 0.9924
Epoch 57/100
469/469 [==============================] - 27s 58ms/step - loss: 0.0017 - accuracy: 0.99
97 - val_loss: 0.1510 - val_accuracy: 0.9920
Epoch 58/100
469/469 [==============================] - 28s 60ms/step - loss: 0.0012 - accuracy: 0.99
98 - val_loss: 0.1447 - val_accuracy: 0.9912
Epoch 59/100
469/469 [==============================] - 23s 50ms/step - loss: 0.0022 - accuracy: 0.99
96 - val_loss: 0.1470 - val_accuracy: 0.9916
Epoch 60/100
469/469 [==============================] - 23s 50ms/step - loss: 0.0013 - accuracy: 0.99
98 - val_loss: 0.1573 - val_accuracy: 0.9926
Epoch 61/100
469/469 [==============================] - 24s 51ms/step - loss: 5.8906e-04 - accuracy:
```

```
0.9998 - val_loss: 0.1913 - val_accuracy: 0.9914
Epoch 62/100
469/469 [==============================] - 24s 50ms/step - loss: 0.0015 - accuracy: 0.99
98 - val_loss: 0.1711 - val_accuracy: 0.9921
Epoch 63/100
469/469 [==============================] - 24s 51ms/step - loss: 0.0018 - accuracy: 0.99
98 - val_loss: 0.1761 - val_accuracy: 0.9901
Epoch 64/100
469/469 [==============================] - 24s 50ms/step - loss: 7.4369e-04 - accuracy:
0.9999 - val_loss: 0.1711 - val_accuracy: 0.9927
Epoch 65/100
469/469 [==============================] - 24s 50ms/step - loss: 0.0013 - accuracy: 0.99
99 - val_loss: 0.1551 - val_accuracy: 0.9923
Epoch 66/100
469/469 [==============================] - 24s 52ms/step - loss: 0.0014 - accuracy: 0.99
98 - val_loss: 0.1558 - val_accuracy: 0.9925
Epoch 67/100
469/469 [==============================] - 24s 51ms/step - loss: 0.0017 - accuracy: 0.99
96 - val_loss: 0.1776 - val_accuracy: 0.9918
Epoch 68/100
469/469 [==============================] - 24s 51ms/step - loss: 0.0028 - accuracy: 0.99
98 - val_loss: 0.1810 - val_accuracy: 0.9921
Epoch 69/100
469/469 [==============================] - 25s 53ms/step - loss: 0.0020 - accuracy: 0.99
97 - val_loss: 0.1719 - val_accuracy: 0.9916
Epoch 70/100
469/469 [==============================] - 25s 54ms/step - loss: 0.0019 - accuracy: 0.99
97 - val_loss: 0.1637 - val_accuracy: 0.9924
Epoch 71/100
469/469 [==============================] - 26s 55ms/step - loss: 0.0012 - accuracy: 0.99
98 - val_loss: 0.1824 - val_accuracy: 0.9918
Epoch 72/100
469/469 [==============================] - 25s 53ms/step - loss: 8.5786e-04 - accuracy:
0.9998 - val_loss: 0.1500 - val_accuracy: 0.9927
Epoch 73/100
469/469 [==============================] - 25s 54ms/step - loss: 0.0011 - accuracy: 0.99
97 - val_loss: 0.1769 - val_accuracy: 0.9917
Epoch 74/100
469/469 [==============================] - 25s 54ms/step - loss: 0.0024 - accuracy: 0.99
97 - val_loss: 0.1794 - val_accuracy: 0.9918
Epoch 75/100
469/469 [==============================] - 25s 54ms/step - loss: 0.0016 - accuracy: 0.99
98 - val_loss: 0.1691 - val_accuracy: 0.9915
Epoch 76/100
469/469 [==============================] - 25s 53ms/step - loss: 0.0011 - accuracy: 0.99
97 - val_loss: 0.1931 - val_accuracy: 0.9906
Epoch 77/100
469/469 [==============================] - 25s 53ms/step - loss: 0.0021 - accuracy: 0.99
98 - val_loss: 0.1750 - val_accuracy: 0.9904
Epoch 78/100
469/469 [==============================] - 25s 53ms/step - loss: 0.0014 - accuracy: 0.99
98 - val_loss: 0.1447 - val_accuracy: 0.9922
Epoch 79/100
469/469 [==============================] - 25s 53ms/step - loss: 4.7755e-04 - accuracy:
0.9998 - val_loss: 0.1848 - val_accuracy: 0.9906
Epoch 80/100
469/469 [==============================] - 26s 55ms/step - loss: 6.0331e-04 - accuracy:
0.9999 - val_loss: 0.1673 - val_accuracy: 0.9916
Epoch 81/100
469/469 [==============================] - 25s 54ms/step - loss: 3.0869e-04 - accuracy:
```

```
0.9999 - val_loss: 0.1854 - val_accuracy: 0.9919
Epoch 82/100
469/469 [==============================] - 26s 55ms/step - loss: 6.6706e-04 - accuracy:
0.9999 - val_loss: 0.1670 - val_accuracy: 0.9923
Epoch 83/100
469/469 [==============================] - 25s 52ms/step - loss: 0.0010 - accuracy: 0.99
99 - val_loss: 0.1724 - val_accuracy: 0.9924
Epoch 84/100
469/469 [==============================] - 25s 53ms/step - loss: 0.0010 - accuracy: 0.99
99 - val_loss: 0.1875 - val_accuracy: 0.9917
Epoch 85/100
469/469 [==============================] - 25s 54ms/step - loss: 4.0808e-04 - accuracy:
0.9999 - val_loss: 0.1790 - val_accuracy: 0.9929
Epoch 86/100
469/469 [==============================] - 25s 53ms/step - loss: 6.3818e-04 - accuracy:
0.9999 - val_loss: 0.2052 - val_accuracy: 0.9927
Epoch 87/100
469/469 [==============================] - 25s 53ms/step - loss: 5.6670e-04 - accuracy:
0.9999 - val_loss: 0.2248 - val_accuracy: 0.9925
Epoch 88/100
469/469 [==============================] - 25s 52ms/step - loss: 0.0015 - accuracy: 0.99
98 - val_loss: 0.1860 - val_accuracy: 0.9927
Epoch 89/100
469/469 [==============================] - 25s 54ms/step - loss: 8.9376e-04 - accuracy:
0.9998 - val_loss: 0.2250 - val_accuracy: 0.9910
Epoch 90/100
469/469 [==============================] - 25s 54ms/step - loss: 6.9421e-04 - accuracy:
0.9998 - val_loss: 0.2284 - val_accuracy: 0.9924
Epoch 91/100
469/469 [==============================] - 25s 54ms/step - loss: 0.0011 - accuracy: 0.99
98 - val_loss: 0.2049 - val_accuracy: 0.9925
Epoch 92/100
469/469 [==============================] - 25s 54ms/step - loss: 7.5969e-04 - accuracy:
0.9999 - val_loss: 0.2030 - val_accuracy: 0.9917
Epoch 93/100
469/469 [==============================] - 25s 54ms/step - loss: 0.0012 - accuracy: 0.99
98 - val_loss: 0.2321 - val_accuracy: 0.9924
Epoch 94/100
469/469 [==============================] - 26s 55ms/step - loss: 0.0012 - accuracy: 0.99
99 - val_loss: 0.1902 - val_accuracy: 0.9930
Epoch 95/100
469/469 [==============================] - 24s 52ms/step - loss: 3.2894e-04 - accuracy:
0.9999 - val_loss: 0.2385 - val_accuracy: 0.9917
Epoch 96/100
469/469 [==============================] - 25s 52ms/step - loss: 5.7841e-04 - accuracy:
0.9999 - val_loss: 0.2046 - val_accuracy: 0.9912
Epoch 97/100
469/469 [==============================] - 25s 53ms/step - loss: 6.7287e-04 - accuracy:
0.9999 - val_loss: 0.2253 - val_accuracy: 0.9912
Epoch 98/100
469/469 [==============================] - 24s 52ms/step - loss: 4.4055e-04 - accuracy:
0.9999 - val_loss: 0.2432 - val_accuracy: 0.9911
Epoch 99/100
469/469 [==============================] - 25s 54ms/step - loss: 0.0019 - accuracy: 0.99
98 - val_loss: 0.1796 - val_accuracy: 0.9925
Epoch 100/100
469/469 [==============================] - 25s 52ms/step - loss: 0.0017 - accuracy: 0.99
98 - val_loss: 0.2283 - val_accuracy: 0.9906
```

In [12]:
```
n = len(run_hist_nn3.history["loss"])

fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot(1, 2, 1)
ax.plot(range(n), (run_hist_nn3.history["loss"]),'r', label="Train Loss")
ax.plot(range(n), (run_hist_nn3.history["val_loss"]),'b', label="Validation Loss")
ax.legend()
ax.set_title('Loss over iterations')

ax = fig.add_subplot(1, 2, 2)
ax.plot(range(n), (run_hist_nn3.history["accuracy"]),'r', label="Train Acc")
ax.plot(range(n), (run_hist_nn3.history["val_accuracy"]),'b', label="Validation Acc")
ax.legend(loc='lower right')
ax.set_title('Accuracy over iterations')
```

Out[12]:
```
Text(0.5, 1.0, 'Accuracy over iterations')
```



The validation accuracy peaks at 5 epochs after that the model starts overfitting. Retrain the model on 5 epochs and evaluate:

In [15]:
```
network = models.Sequential()
network.add(layers.Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
network.add(layers.MaxPooling2D((2,2)))
network.add(layers.Conv2D(64,(3,3),activation='relu'))
network.add(layers.MaxPooling2D((2,2)))
network.add(layers.Conv2D(64,(3,3),activation='relu'))
network.add(layers.Flatten())
network.add(Dense(512,activation='relu'))
network.add(Dense(10,activation='softmax'))
network.summary()
network.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'
run_hist_nn3=network.fit(x_train,train_labels,epochs=5,batch_size=128,validation_data=(
y_pred_prob_nn_3 = network.predict(x_test)
y_pred_class_nn_3 = np.argmax(y_pred_prob_nn_3,axis=1)
```

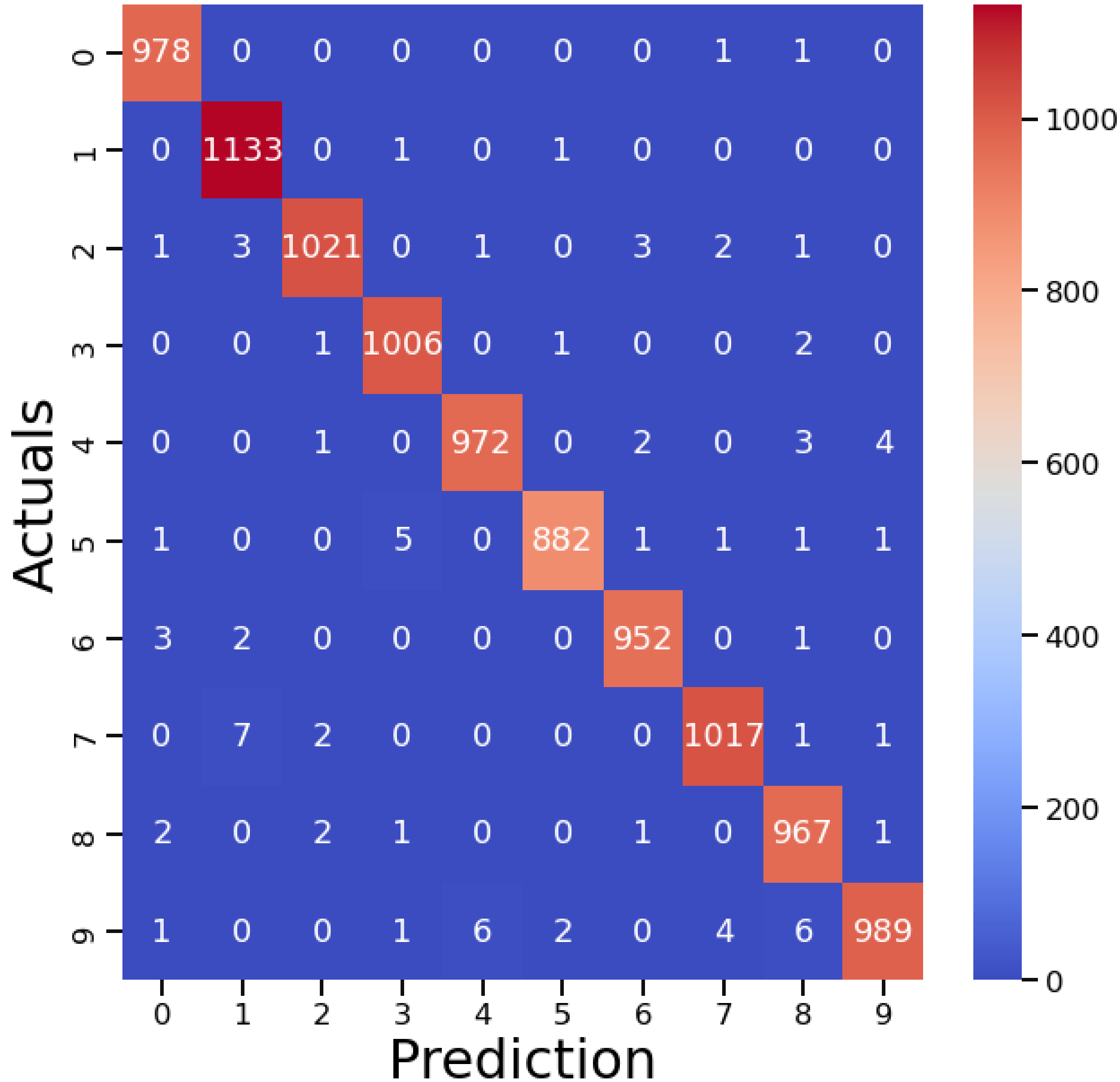
```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_3 (Conv2D)           (None, 26, 26, 32)        320       
                                                                 
 max_pooling2d_2 (MaxPooling  (None, 13, 13, 32)       0         
 2D)                                                             
                                                                 
 conv2d_4 (Conv2D)           (None, 11, 11, 64)        18496     
                                                                 
 max_pooling2d_3 (MaxPooling  (None, 5, 5, 64)         0         
 2D)                                                             
                                                                 
 conv2d_5 (Conv2D)           (None, 3, 3, 64)          36928     
                                                                 
 flatten_1 (Flatten)         (None, 576)               0         
                                                                 
 dense_4 (Dense)             (None, 512)               295424    
                                                                 
 dense_5 (Dense)             (None, 10)                5130      
                                                                 
=================================================================
Total params: 356,298
Trainable params: 356,298
Non-trainable params: 0
_________________________________________________________________
Epoch 1/5
469/469 [==============================] - 78s 163ms/step - loss: 0.1913 - accuracy: 0.9
394 - val_loss: 0.0601 - val_accuracy: 0.9797
Epoch 2/5
469/469 [==============================] - 77s 165ms/step - loss: 0.0467 - accuracy: 0.9
852 - val_loss: 0.0296 - val_accuracy: 0.9905
Epoch 3/5
469/469 [==============================] - 72s 153ms/step - loss: 0.0315 - accuracy: 0.9
903 - val_loss: 0.0440 - val_accuracy: 0.9859
Epoch 4/5
469/469 [==============================] - 72s 153ms/step - loss: 0.0238 - accuracy: 0.9
924 - val_loss: 0.0295 - val_accuracy: 0.9911
Epoch 5/5
469/469 [==============================] - 77s 164ms/step - loss: 0.0182 - accuracy: 0.9
942 - val_loss: 0.0243 - val_accuracy: 0.9917
```

In [16]:

```
sns.set_context('talk')
cm = confusion_matrix(y_test, y_pred_class_nn_3)
_, ax = plt.subplots(figsize=(10,10))
ax = sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm')

ax.set_ylabel('Actuals', fontsize=30);
ax.set_xlabel('Prediction', fontsize=30)
```

Out[16]:

```
Text(0.5, 58.5, 'Prediction')
```



In [17]:

```
test_loss,test_acc = network.evaluate(x_test,test_labels)
print('test accuracy: ',test_acc)
```

```
313/313 [==============================] - 7s 21ms/step - loss: 0.0243 - accuracy: 0.991
7
test accuracy:  0.9916999936103821
```

Findings:

The dense deep learning model without dropout started overfitting quite early and offered a test set accuracy of 98.11%, the same network with dropout applied wasn't overfitting as much and the test set accuracy improved to 98.39%. Model 1 struggled to accurately predict digits 4, 5 and 8 accurately. Classes 8, 4 and 5 were also the digits with the lowest number of samples in the training set in comparison to the other digits, with 5 having the least number of training images. Model 2 addressed some of these misclassifications but still struggled with class 8. Employing a CNN with maxpooling addresses these shortcomings and the model test accuracy improves to 99.17%, reducing the error rate by almost 50% compared to the dense deep learning model without dropout.

The CNN with maxpooling is the best model and should be used for predictions.

Recommendations

The test set accuracy of the CNN is already very good but to get even more improvement a potential idea is to use transfer learning and make use of feature extraction from pre-built large and popular CNN model architectures like inception or VGG.

In [ ]:

```

```