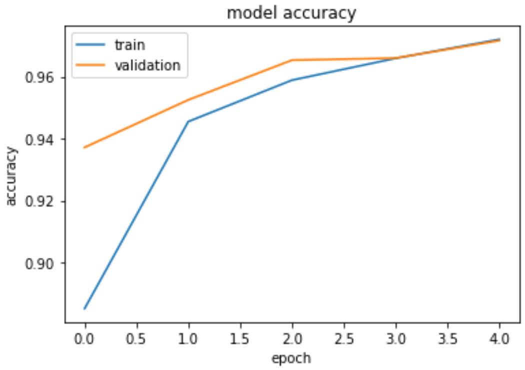
Tensforwflow example with two hidden layers. The MNIST dataset

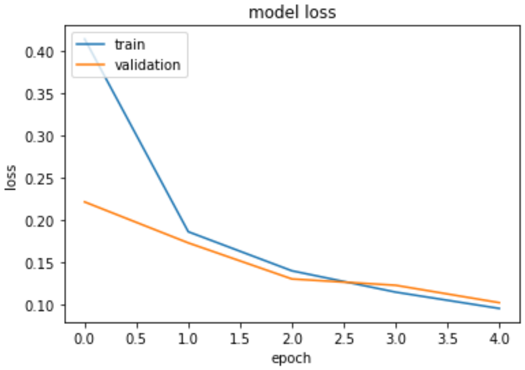
Out[1]: Click here to show or hide your raw code.

Q1: (hidden layer size)

Model: Try a hidden layer size of 50

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten (Flatten)            (None, 784)               0         
_________________________________________________________________
dense (Dense)                (None, 50)                39250     
_________________________________________________________________
dense_1 (Dense)              (None, 50)                2550      
_________________________________________________________________
dense_2 (Dense)              (None, 10)                510       
=================================================================
Total params: 42,310
Trainable params: 42,310
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```

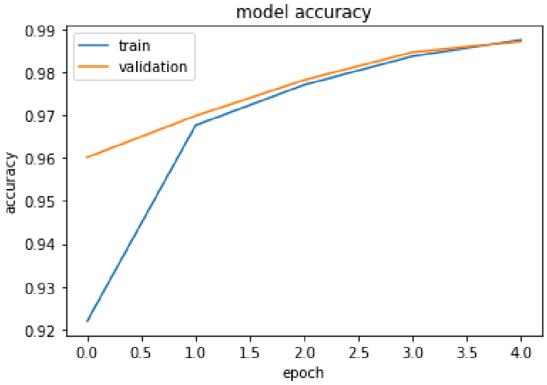


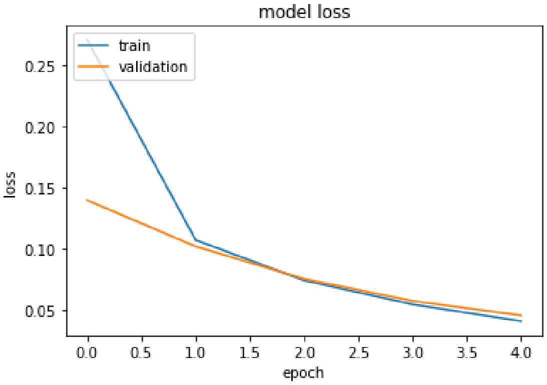


```
-------------------------------------------------------------
Train Accuracy = 0.972
Train Loss = 0.095
-------------------------------------------------------------
Validation Accuracy = 0.972
Validation Loss = 0.102
-------------------------------------------------------------
Test Accuracy = 0.970
Test Loss = 0.104
-------------------------------------------------------------
Taining Time = 7.903 second
```

Model: Try a hidden layer size of 200

```
------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_1 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_3 (Dense)              (None, 200)               157000    
_________________________________________________________________
dense_4 (Dense)              (None, 200)               40200     
_________________________________________________________________
dense_5 (Dense)              (None, 10)                2010      
=================================================================
Total params: 199,210
Trainable params: 199,210
Non-trainable params: 0
_________________________________________________________________
None
------------------------------------------------------------
```

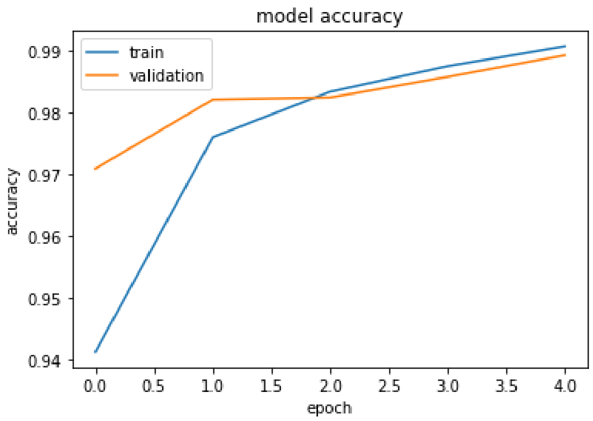


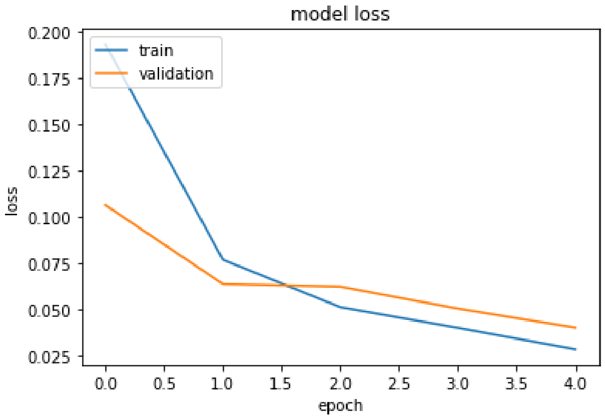


```
------------------------------------------------------------
Train Accuracy = 0.988
Train Loss = 0.041
------------------------------------------------------------
Validation Accuracy = 0.987
Validation Loss = 0.045
------------------------------------------------------------
Test Accuracy = 0.976
Test Loss = 0.080
------------------------------------------------------------
Taining Time = 7.000 second
```

Model: Try a Custom hidden layer size

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_2 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_6 (Dense)              (None, 1000)              785000    
_________________________________________________________________
dense_7 (Dense)              (None, 1000)              1001000   
_________________________________________________________________
dense_8 (Dense)              (None, 10)                10010     
=================================================================
Total params: 1,796,010
Trainable params: 1,796,010
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```





```
-------------------------------------------------------------
Train Accuracy = 0.991
Train Loss = 0.029
-------------------------------------------------------------
Validation Accuracy = 0.989
Validation Loss = 0.040
-------------------------------------------------------------
Test Accuracy = 0.979
Test Loss = 0.073
-------------------------------------------------------------
Taining Time = 7.385 second
```

Answering Q1: (hidden layer size)

1. Try a hidden layer size of 200. How does the validation accuracy of the model change? What about the time it took the algorithm to train? Can you find a hidden layer size that does better?

How does the validation accuracy of the model change?

When increase the hidden layer from 50 to 200, the validation accuracy also increased from 0.970 to 0.988

What about the time it took the algorithm to train?

Train time increased from 7.396 to 7.856 seconds ¶

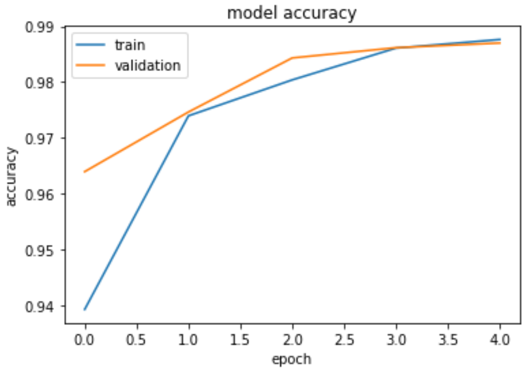
Can you find a hidden layer size that does better?

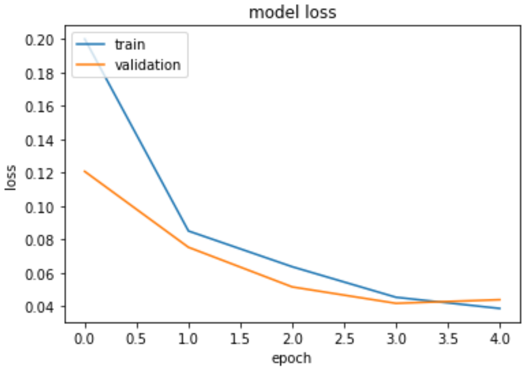
hidden layer size = 1000, I can notice that increasing the hidden layer is increaseing the validation accuracy and testing accuracy, but if you keep increasing hidden layer more than 1000 the accuracy almost the same but the training time increaseing, until you the model will overfit and the accuracy will start in increaseing.

Q2: (The Depth of hidden layers)

Model: Add another hidden layer

```
------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_3 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_9 (Dense)              (None, 1000)              785000    
_________________________________________________________________
dense_10 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_11 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_12 (Dense)             (None, 10)                10010     
=================================================================
Total params: 2,797,010
Trainable params: 2,797,010
Non-trainable params: 0
_________________________________________________________________
None
------------------------------------------------------------
```





```
------------------------------------------------------------
Train Accuracy = 0.988
Train Loss = 0.039
------------------------------------------------------------
Validation Accuracy = 0.987
Validation Loss = 0.044
------------------------------------------------------------
Test Accuracy = 0.980
Test Loss = 0.080
------------------------------------------------------------
Taining Time = 7.768 second
```

Answering Q2: (The Depth of hidden layers)

2. The *depth* of the algorithm. Add another hidden layer to the algorithm. This is an extremely important exercise! How does the validation accuracy change? What about the time it took the algorithm to train?

How does the validation accuracy change?

When Adding another hidden layer, the validation accuracy decreased from 0.987 to 0.982

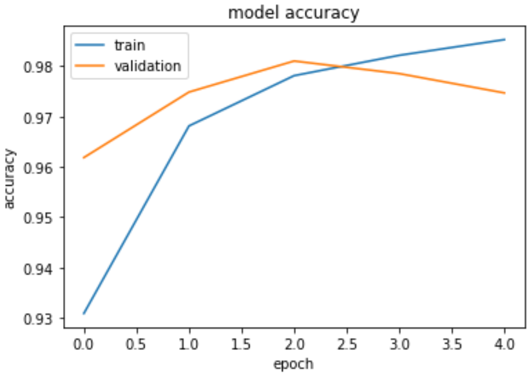
What about the time it took the algorithm to train?

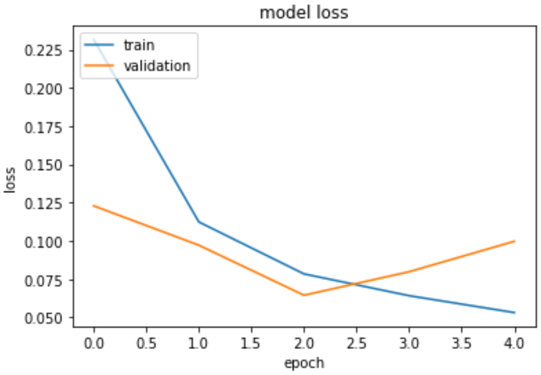
Train time increased from 7.678 to 8.179 seconds

Q3: (Width and Depth of the Model)

Model: (5 Hidden layers, 1000 width)

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_4 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_13 (Dense)             (None, 1000)              785000    
_________________________________________________________________
dense_14 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_15 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_16 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_17 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_18 (Dense)             (None, 10)                10010     
=================================================================
Total params: 4,799,010
Trainable params: 4,799,010
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```





```
-------------------------------------------------------------
Train Accuracy = 0.985
Train Loss = 0.053
-------------------------------------------------------------
Validation Accuracy = 0.975
Validation Loss = 0.100
-------------------------------------------------------------
Test Accuracy = 0.964
Test Loss = 0.136
-------------------------------------------------------------
Taining Time = 8.959 second
```

3. The *width and depth* of the algorithm. Add as many additional layers as you need to reach 5 hidden layers. Moreover, adjust the width of the algorithm as you find suitable. How does the validation accuracy change? What about the time it took the algorithm to train?

How does the validation accuracy of the model change?

When use (5 hidden layers, 1000 width) the validation accuracy also increased from 0.982 to 0.987.

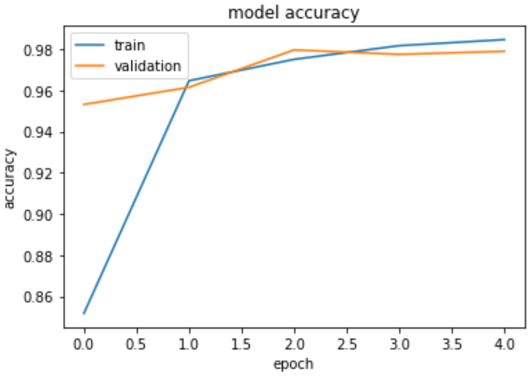
What about the time it took the algorithm to train?

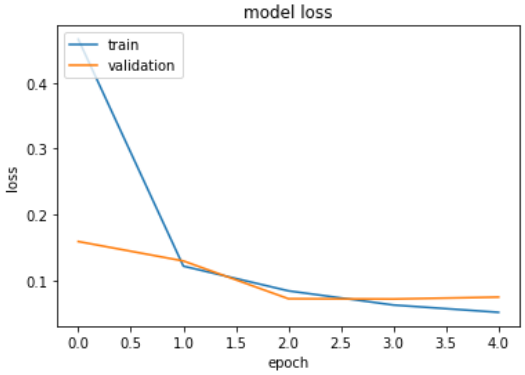
Train time increased from 8.179 to 9.193 seconds

Q4: (Fiddle with the activation functions.)

Model: (use Sigmoid Activation Funcation)

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_5 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_19 (Dense)             (None, 1000)              785000    
_________________________________________________________________
dense_20 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_21 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_22 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_23 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_24 (Dense)             (None, 10)                10010     
=================================================================
Total params: 4,799,010
Trainable params: 4,799,010
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```





```
-------------------------------------------------------------
Train Accuracy = 0.985
Train Loss = 0.051
-------------------------------------------------------------
Validation Accuracy = 0.979
Validation Loss = 0.075
-------------------------------------------------------------
Test Accuracy = 0.973
Test Loss = 0.097
-------------------------------------------------------------
Taining Time = 8.934 second
```

4. Fiddle with the activation functions. Try applying sigmoid transformation to bothlayers. The sigmoid activation is given by the string 'sigmoid'.

How does the validation accuracy of the model change?

use Sigmoid in 2 hidden layers, the validation accuracy also decreased from 0. 987 to 0.986. Test Accuracy decreased from 0.979 to 0.977

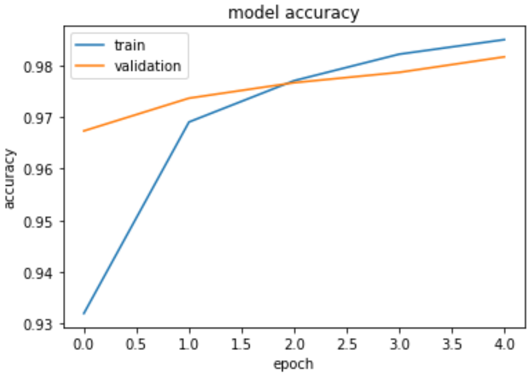
What about the time it took the algorithm to train?

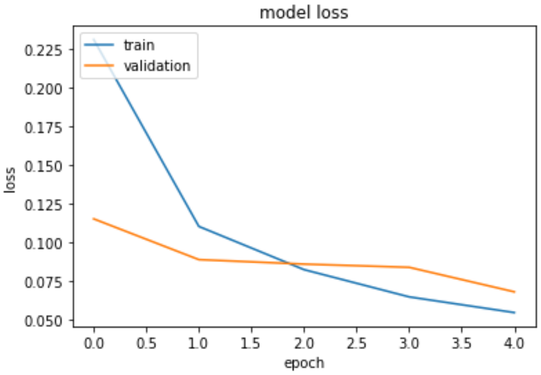
Train time decreased from 9.193 to 9.190 seconds

Q5: (Fiddle with the activation functions.)

Model: (use Tanh Activation Funcation)

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_6 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_25 (Dense)             (None, 1000)              785000    
_________________________________________________________________
dense_26 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_27 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_28 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_29 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_30 (Dense)             (None, 10)                10010     
=================================================================
Total params: 4,799,010
Trainable params: 4,799,010
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```





```
-------------------------------------------------------------
Train Accuracy = 0.985
Train Loss = 0.055
-------------------------------------------------------------
Validation Accuracy = 0.982
Validation Loss = 0.068
-------------------------------------------------------------
Test Accuracy = 0.975
Test Loss = 0.103
-------------------------------------------------------------
Taining Time = 8.911 second
```

5. Fiddle with the activation functions. Try applying sigmoid transformation to bothlayers. The sigmoid activation is given by the string 'sigmoid'.

How does the validation accuracy of the model change?

use Sigmoid in 2 hidden layers, the validation accuracy also decreased from 0.986 to 0.980. Test Accuracy decreased from 0.977 to 0.972

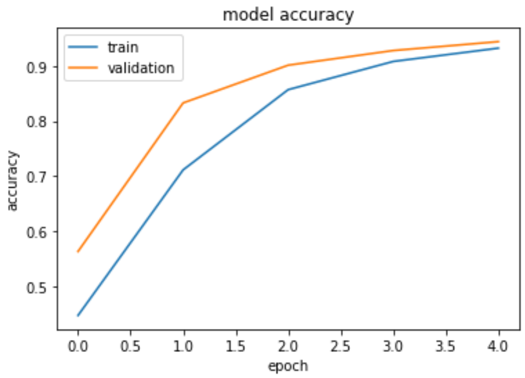
What about the time it took the algorithm to train?

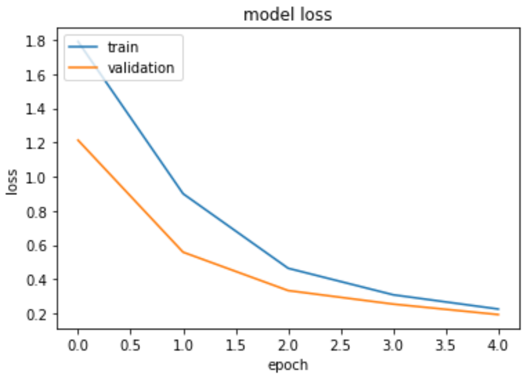
Train time increased from 9.190 to 9.380 seconds

Q6: (Adjust the batch size)

Model: (use batch size 10000)

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 10000, NUM_EPOCHS = 5
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_7 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_31 (Dense)             (None, 1000)              785000    
_________________________________________________________________
dense_32 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_33 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_34 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_35 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_36 (Dense)             (None, 10)                10010     
=================================================================
Total params: 4,799,010
Trainable params: 4,799,010
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```





```
-------------------------------------------------------------
Train Accuracy = 0.933
Train Loss = 0.226
-------------------------------------------------------------
Validation Accuracy = 0.945
Validation Loss = 0.194
-------------------------------------------------------------
Test Accuracy = 0.940
Test Loss = 0.189
-------------------------------------------------------------
Taining Time = 6.075 second
```

6. Adjust the batch size. Try a batch size of 10000. How does the required time change? What about the accuracy?

How does the required time change?

Train time decreased from 9.193 to 6.044 seconds

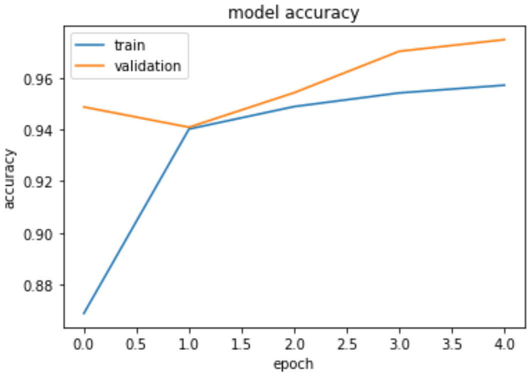
What about the accuracy?

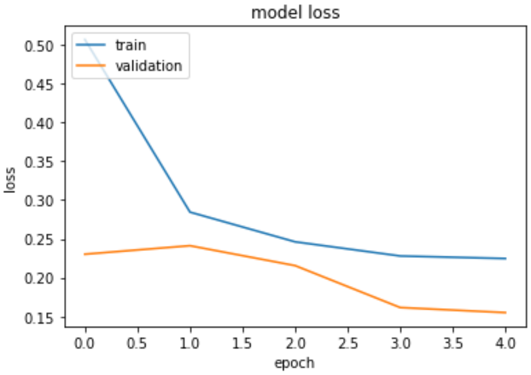
validation Accuracy decreased from 0.987 to 0.948, Test Accuracy decreased from 0.979 to 0.945,

Q7: (Adjust the batch size)

Model: (use batch size 1)

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 1, NUM_EPOCHS = 5
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_8 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_37 (Dense)             (None, 1000)              785000    
_________________________________________________________________
dense_38 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_39 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_40 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_41 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_42 (Dense)             (None, 10)                10010     
=================================================================
Total params: 4,799,010
Trainable params: 4,799,010
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```





```
-------------------------------------------------------------
Train Accuracy = 0.957
Train Loss = 0.225
-------------------------------------------------------------
Validation Accuracy = 0.975
Validation Loss = 0.155
-------------------------------------------------------------
Test Accuracy = 0.969
Test Loss = 0.201
-------------------------------------------------------------
Taining Time = 449.658 second
```

7. Adjust the batch size. Try a batch size of 1. How does the required time change? What about the accuracy?

How does the required time change?

Train time significantly Increased from 9.193 to 449.6 seconds

---

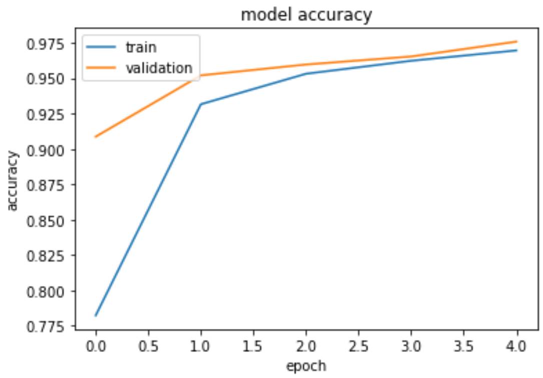
What about the accuracy?

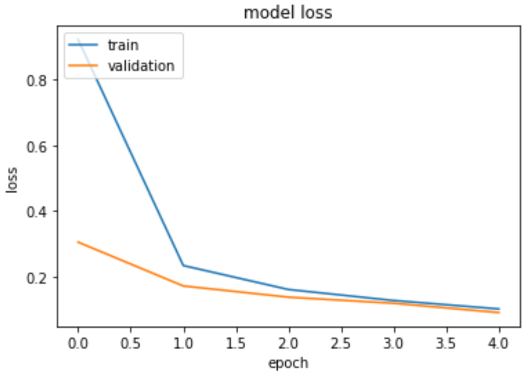
validation Accuracy significantly decreased from 0.987 to 0.975 . , Test Accuracy decreased from 0.979 to 0.969.

Q8: (Adjust the Learning rate)

Model: (use Learning rate 0.0001)

```
------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 1000, NUM_EPOCHS = 5
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_9 (Flatten)          (None, 784)               0         
_________________________________________________________________
dense_43 (Dense)             (None, 1000)              785000    
_________________________________________________________________
dense_44 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_45 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_46 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_47 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_48 (Dense)             (None, 10)                10010     
=================================================================
Total params: 4,799,010
Trainable params: 4,799,010
Non-trainable params: 0
_________________________________________________________________
None
------------------------------------------------------------
```





```
------------------------------------------------------------
Train Accuracy = 0.970
Train Loss = 0.102
------------------------------------------------------------
Validation Accuracy = 0.976
Validation Loss = 0.091
------------------------------------------------------------
Test Accuracy = 0.967
Test Loss = 0.105
------------------------------------------------------------
Taining Time = 7.548 second
```

8. Adjust the learning rate. Try a value of 0.0001. Does it make a difference?

Does it make a difference?

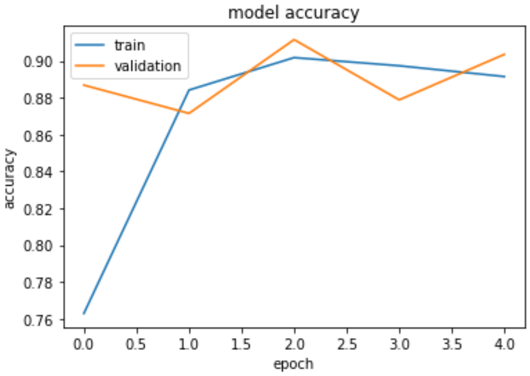
Hint : default learning rate is 0.01 ¶

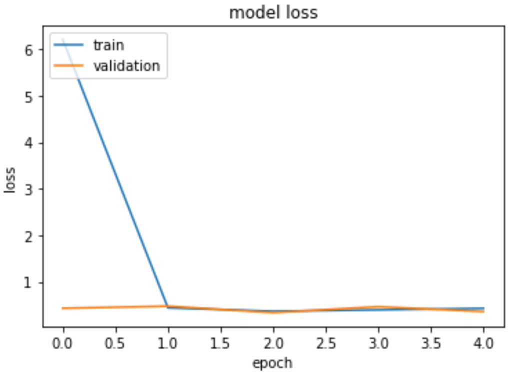
validation Accuracy decreased from 0.987 to 0.976, testing Accuracy also decreased from 0.979 to 0.967

Q9: (Adjust the Learning rate)

Model: (use Learning rate 0.02)

```
-------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 5
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_10 (Flatten)         (None, 784)               0         
_________________________________________________________________
dense_49 (Dense)             (None, 1000)              785000    
_________________________________________________________________
dense_50 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_51 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_52 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_53 (Dense)             (None, 1000)              1001000   
_________________________________________________________________
dense_54 (Dense)             (None, 10)                10010     
=================================================================
Total params: 4,799,010
Trainable params: 4,799,010
Non-trainable params: 0
_________________________________________________________________
None
-------------------------------------------------------------
```





```
-------------------------------------------------------------
Train Accuracy = 0.891
Train Loss = 0.429
-------------------------------------------------------------
Validation Accuracy = 0.904
Validation Loss = 0.362
-------------------------------------------------------------
Test Accuracy = 0.901
Test Loss = 0.370
-------------------------------------------------------------
Taining Time = 10.235 second
```

9. Adjust the learning rate. Try a value of 0.02. Does it make a difference?

Does it make a difference?

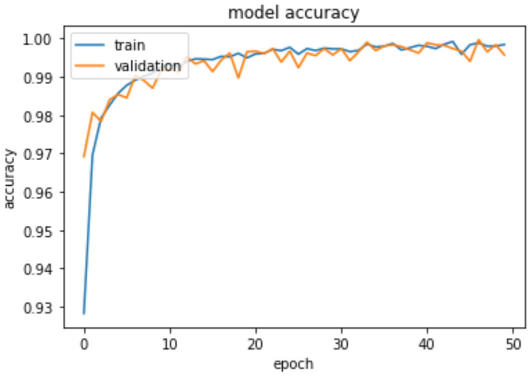
Hint : default learning rate is 0.01 ¶

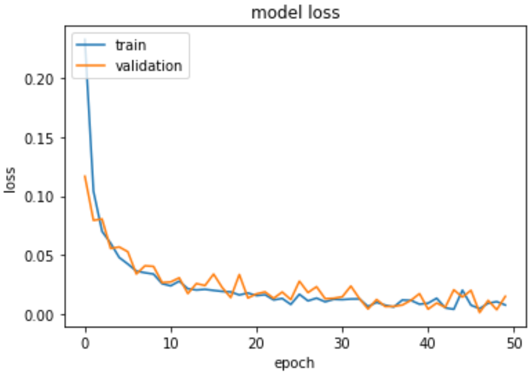
validation Accuracy decreased from 0.987 to 0.904, testing Accuracy also decreased from 0.979 to 0.901

Q10: (Combining all together)

Model: (Customized)

```
------------------------------------------------------------
HyperParamters:
BUFFER SIZE =10000, BATCH SIZE = 100, NUM_EPOCHS = 50
Model: "sequential_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten_12 (Flatten)         (None, 784)               0         
_________________________________________________________________
dense_61 (Dense)             (None, 600)               471000    
_________________________________________________________________
dense_62 (Dense)             (None, 600)               360600    
_________________________________________________________________
dense_63 (Dense)             (None, 600)               360600    
_________________________________________________________________
dense_64 (Dense)             (None, 600)               360600    
_________________________________________________________________
dense_65 (Dense)             (None, 600)               360600    
_________________________________________________________________
dense_66 (Dense)             (None, 10)                6010      
=================================================================
Total params: 1,919,410
Trainable params: 1,919,410
Non-trainable params: 0
_________________________________________________________________
None
------------------------------------------------------------
```





```
------------------------------------------------------------
Train Accuracy = 0.998
Train Loss = 0.008
------------------------------------------------------------
Validation Accuracy = 0.996
Validation Loss = 0.015
------------------------------------------------------------
Test Accuracy = 0.982
Test Loss = 0.145
------------------------------------------------------------
Taining Time = 78.653 second
```

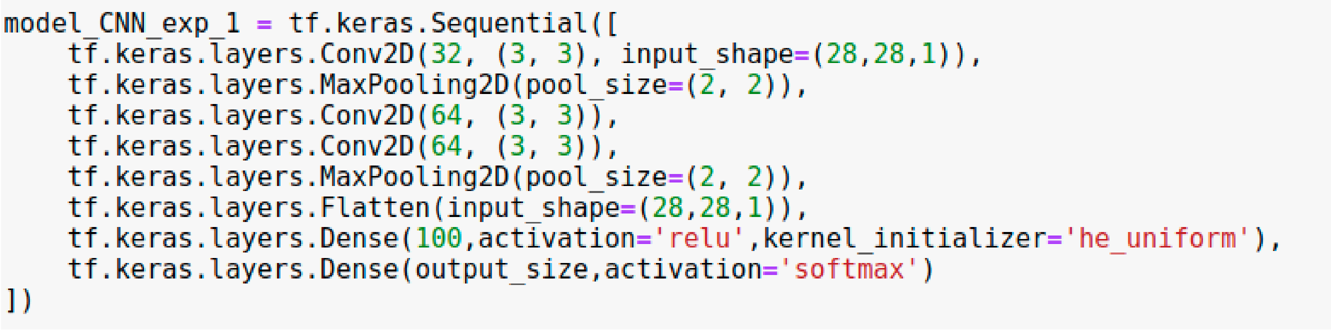
Final Model

Validation Accuracy = 0.998

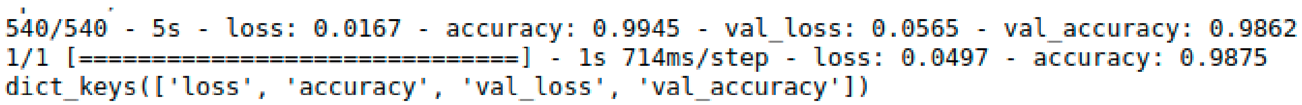
Test Accuracy = 0.982

CNN Network

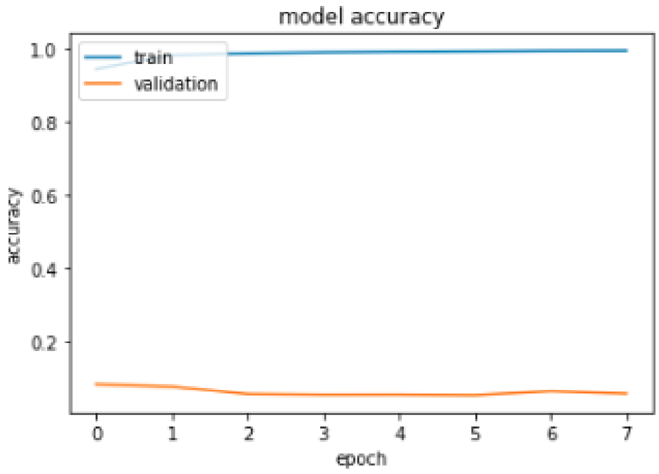
CNN: Experiment 1

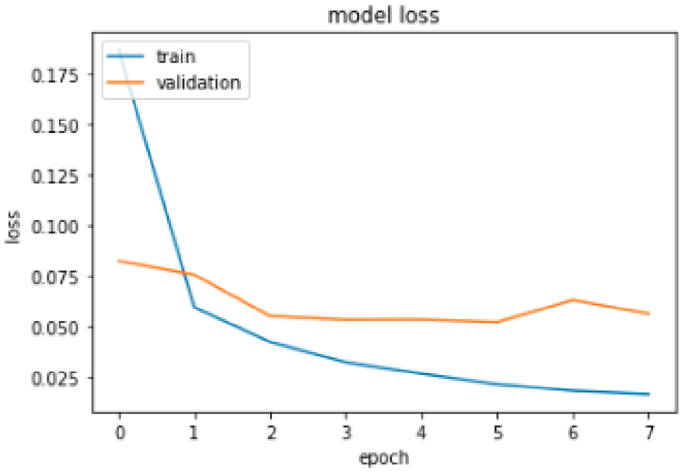


```
model_CNN_exp_1 = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), input_shape=(28,28,1)),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Conv2D(64, (3, 3)),
    tf.keras.layers.Conv2D(64, (3, 3)),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Flatten(input_shape=(28,28,1)),
    tf.keras.layers.Dense(100,activation='relu',kernel_initializer='he_uniform'),
    tf.keras.layers.Dense(output_size,activation='softmax')
])
```

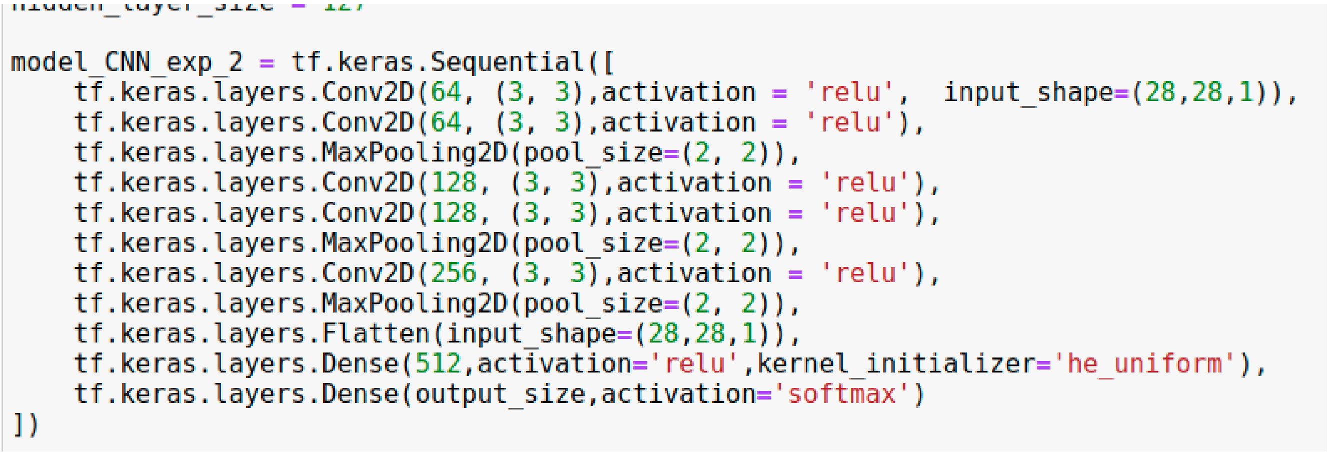


```
540/540 - 5s - loss: 0.0167 - accuracy: 0.9945 - val_loss: 0.0565 - val_accuracy: 0.9862
1/1 [==============================] - 1s 714ms/step - loss: 0.0497 - accuracy: 0.9875
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

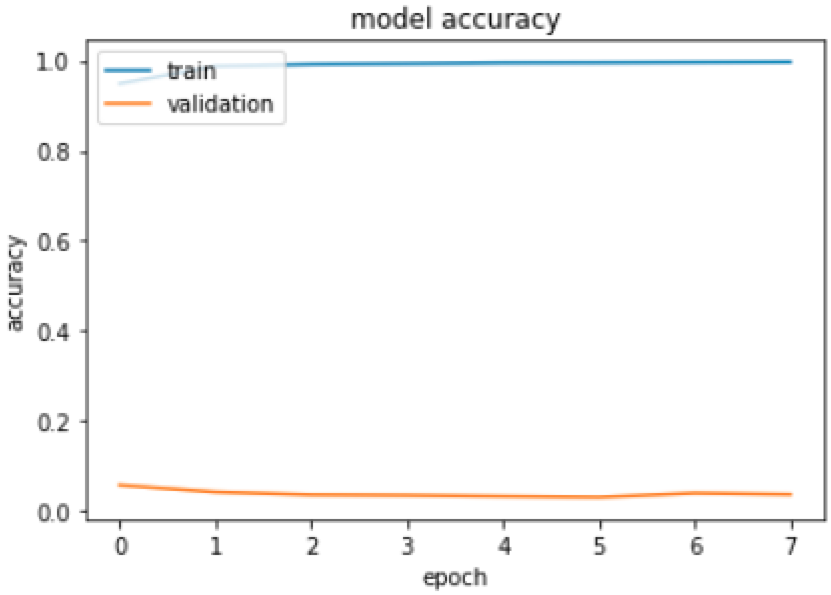


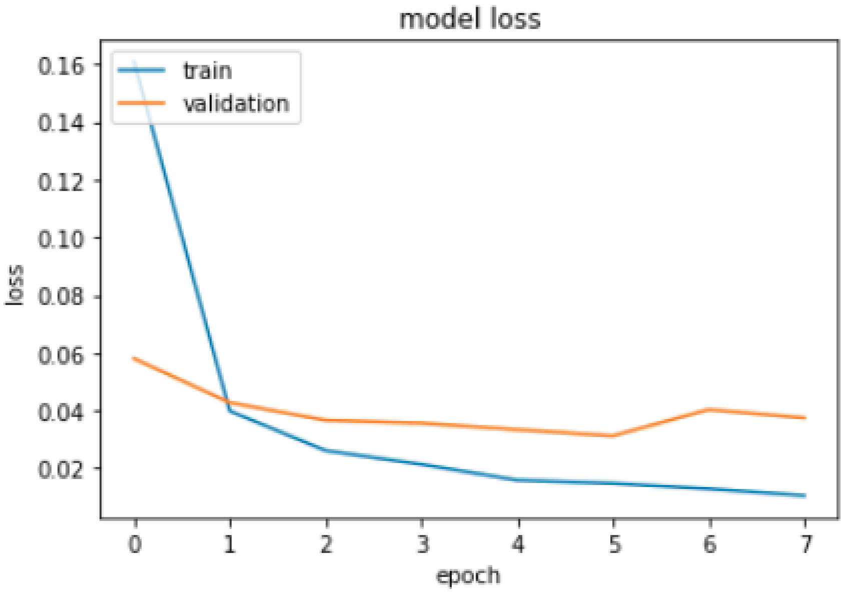


CNN: Experiment 2

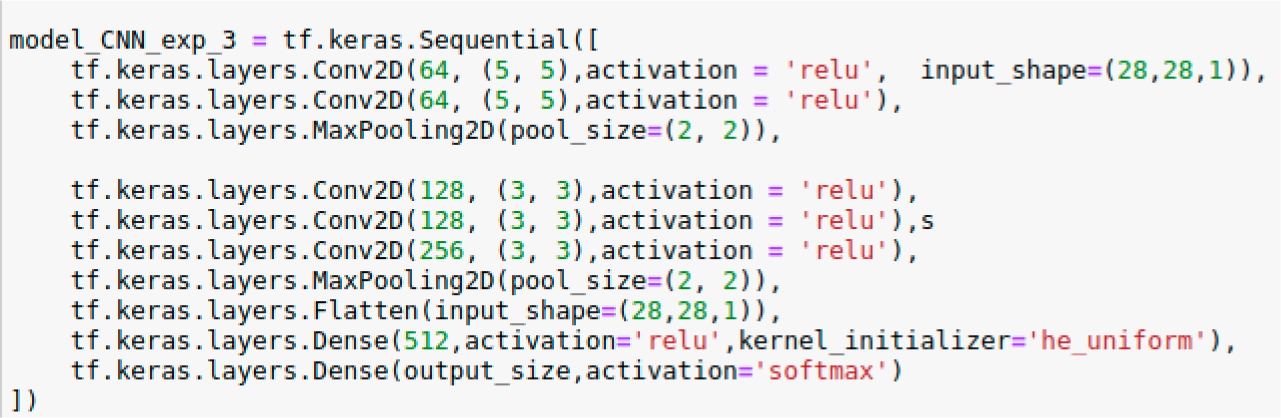


```
hidden_layer_size = 127

model_CNN_exp_2 = tf.keras.Sequential([
    tf.keras.layers.Conv2D(64, (3, 3),activation = 'relu',  input_shape=(28,28,1)),
    tf.keras.layers.Conv2D(64, (3, 3),activation = 'relu'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Conv2D(128, (3, 3),activation = 'relu'),
    tf.keras.layers.Conv2D(128, (3, 3),activation = 'relu'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Conv2D(256, (3, 3),activation = 'relu'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Flatten(input_shape=(28,28,1)),
    tf.keras.layers.Dense(512,activation='relu',kernel_initializer='he_uniform'),
    tf.keras.layers.Dense(output_size,activation='softmax')
])
```

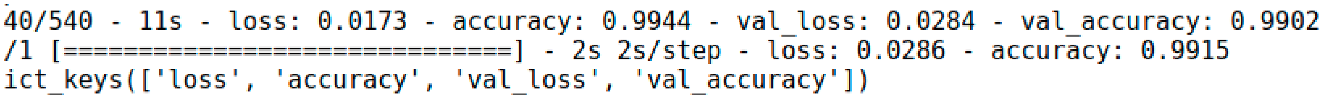




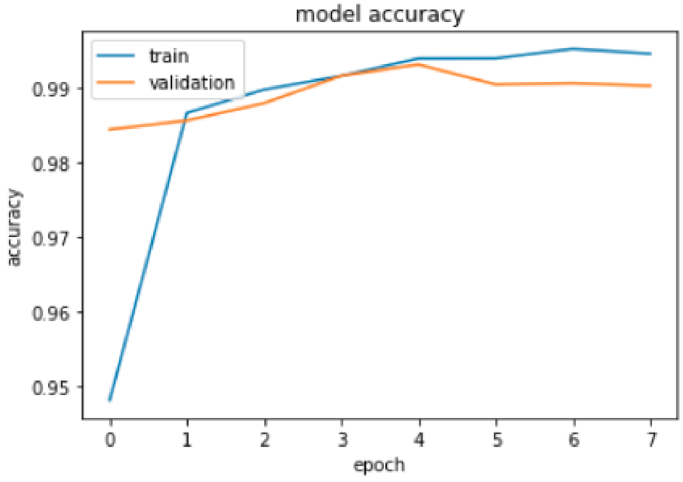
CNN: Experiment 3

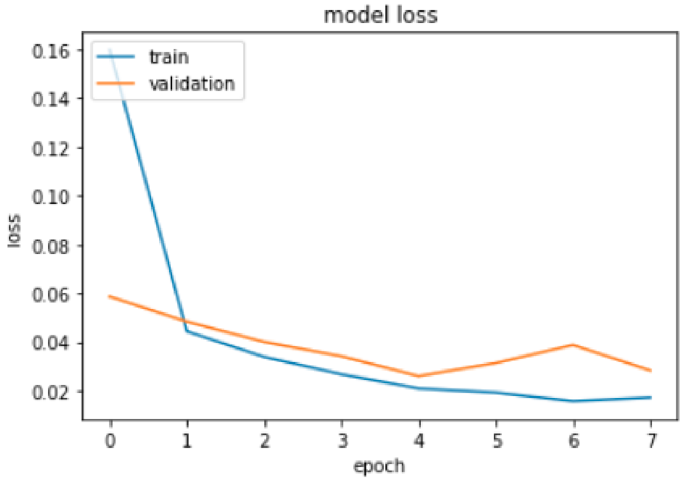


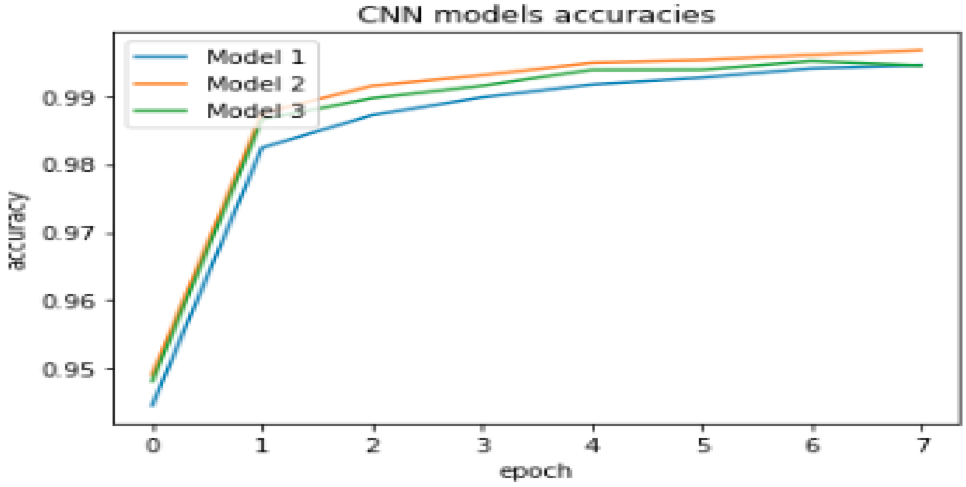
```
model_CNN_exp_3 = tf.keras.Sequential([
    tf.keras.layers.Conv2D(64, (5, 5),activation = 'relu',  input_shape=(28,28,1)),
    tf.keras.layers.Conv2D(64, (5, 5),activation = 'relu'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),

    tf.keras.layers.Conv2D(128, (3, 3),activation = 'relu'),
    tf.keras.layers.Conv2D(128, (3, 3),activation = 'relu'),s
    tf.keras.layers.Conv2D(256, (3, 3),activation = 'relu'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Flatten(input_shape=(28,28,1)),
    tf.keras.layers.Dense(512,activation='relu',kernel_initializer='he_uniform'),
    tf.keras.layers.Dense(output_size,activation='softmax')
])
```

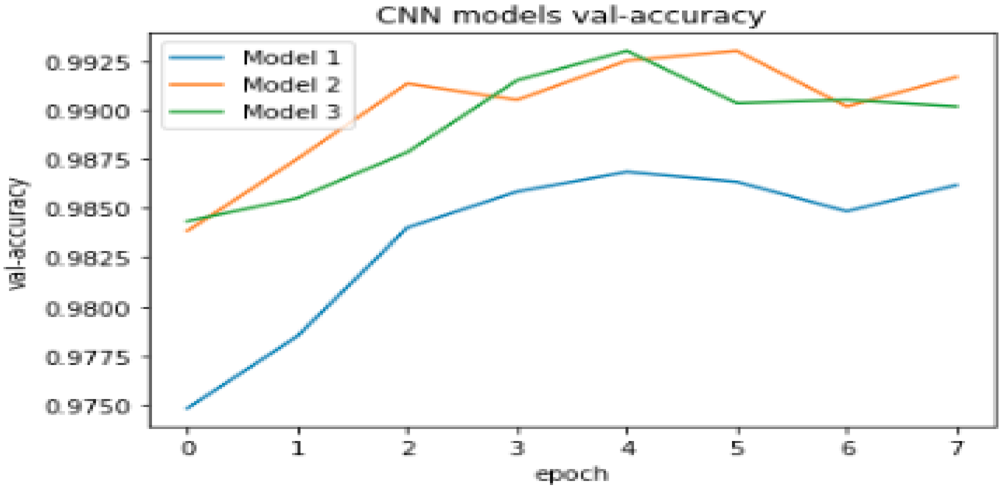


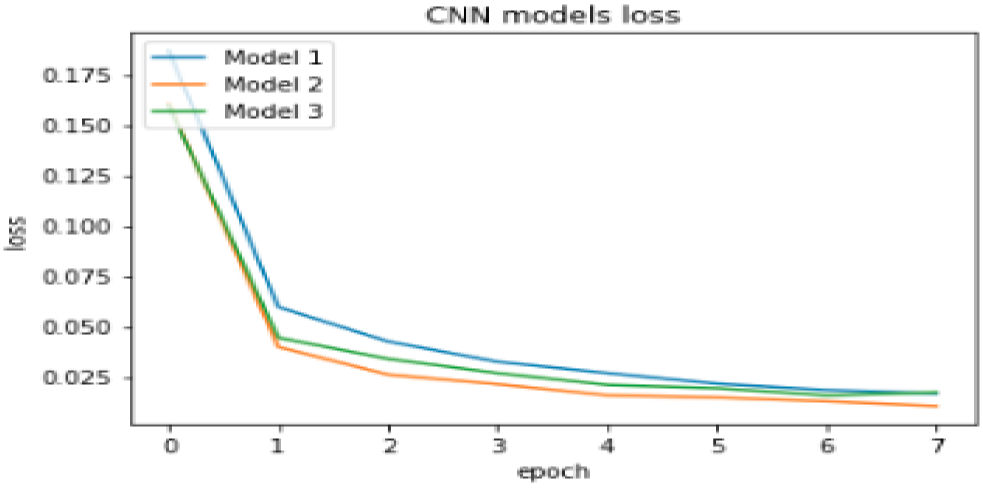
```
40/540 - 11s - loss: 0.0173 - accuracy: 0.9944 - val_loss: 0.0284 - val_accuracy: 0.9902
/1 [==============================] - 2s 2s/step - loss: 0.0286 - accuracy: 0.9915
ict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```






CNN models accuracies
Model 1
Model 2
Model 3
accuracy
0.99
0.98
0.97
0.96
0.95
0
1
2
3
4
5
6
7
epoch


CNN models val-accuracy
Model 1
Model 2
Model 3
val-accuracy
0.9925
0.9900
0.9875
0.9850
0.9825
0.9800
0.9775
0.9750
0
1
2
3
4
5
6
7
epoch


CNN models loss
Model 1
Model 2
Model 3
loss
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0
1
2
3
4
5
6
7
epoch