

# Implement CNN on mnist dataset

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, SimpleRNN, LSTM, GRU, Dense
```

WARNING:tensorflow:From C:\Users\user\anaconda3\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse\_softmax\_cross\_entropy is deprecated. Please use tf.compat.v1.losses.sparse\_softmax\_cross\_entropy instead.

```
In [3]: (train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images, test_images = train_images / 255.0, test_images / 255.0
```

```
In [4]: model=Sequential([
    layers.Input(shape=(28,28,1)),
    layers.Conv2D(32,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dense(10,activation='softmax')
])
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

WARNING:tensorflow:From C:\Users\user\anaconda3\Lib\site-packages\keras\src\backend.py:1398: The name tf.executing\_eagerly\_outside\_functions is deprecated. Please use tf.compat.v1.executing\_eagerly\_outside\_functions instead.

WARNING:tensorflow:From C:\Users\user\anaconda3\Lib\site-packages\keras\src\layers\pooling\max\_pooling2d.py:161: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

WARNING:tensorflow:From C:\Users\user\anaconda3\Lib\site-packages\keras\src\optimizers\\_\_init\_\_.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

In [5]: `model.summary()`

Model: "sequential"

| 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 (MaxPooling2D)      | (None, 5, 5, 64)   | 0       |
| conv2d_2 (Conv2D)                   | (None, 3, 3, 64)   | 36928   |
| flatten (Flatten)                   | (None, 576)        | 0       |
| dense (Dense)                       | (None, 64)         | 36928   |
| dense_1 (Dense)                     | (None, 10)         | 650     |
| =====                               |                    |         |
| Total params: 93322 (364.54 KB)     |                    |         |
| Trainable params: 93322 (364.54 KB) |                    |         |
| Non-trainable params: 0 (0.00 Byte) |                    |         |

```
In [6]: model.fit(train_images[...],np.newaxis],train_labels,epochs=1)
```

WARNING:tensorflow:From C:\Users\user\anaconda3\Lib\site-packages\keras\src\utils\tf\_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\user\anaconda3\Lib\site-packages\keras\src\engine\base\_layer\_utils.py:384: The name tf.executing\_eagerly\_outside\_functions is deprecated. Please use tf.compat.v1.executing\_eagerly\_outside\_functions instead.

1875/1875 [=====] - 31s 15ms/step - loss: 0.1413 - accuracy: 0.9565

Out[6]: <keras.src.callbacks.History at 0x1c26ac84210>

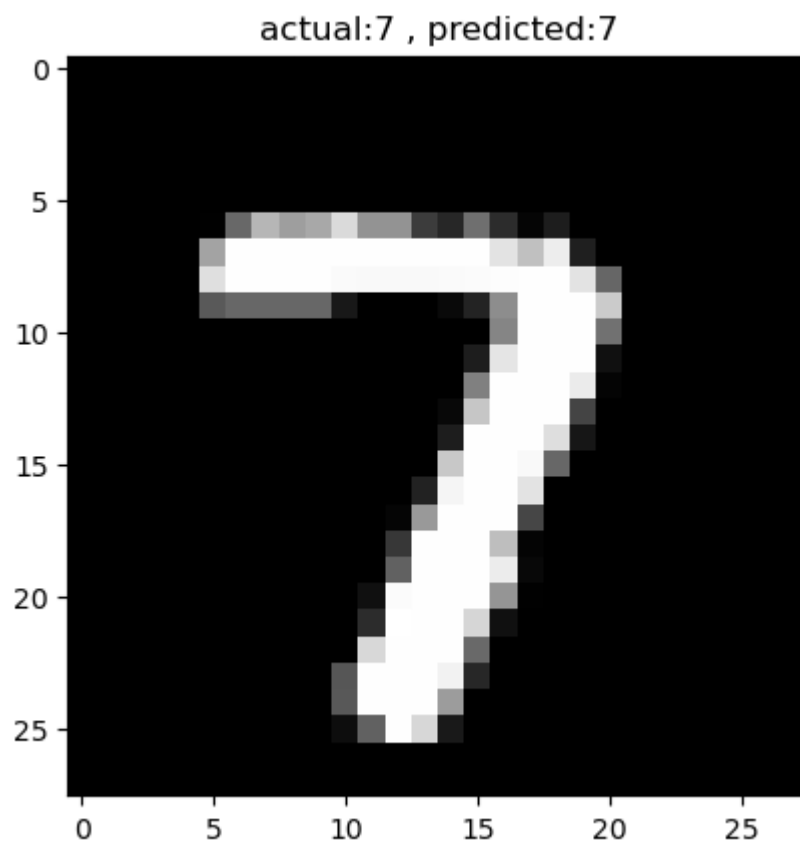
```
In [7]: loss,accuracy=model.evaluate(test_images,test_labels)
print(f'accuracy:{accuracy}')
print(f'loss:{loss}')
```

313/313 [=====] - 3s 7ms/step - loss: 0.0562 - accuracy: 0.9814  
accuracy:0.9814000129699707  
loss:0.05620864778757095

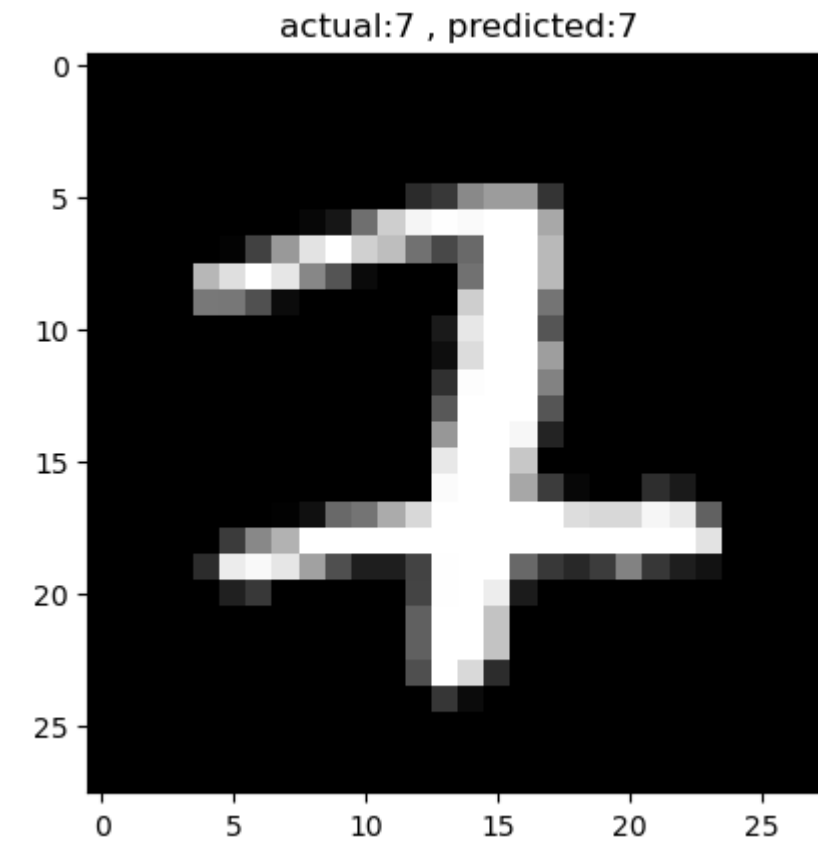
In [ ]:

```
In [20]: for _ in range(3):  
        index=np.random.randint(0,len(test_images))  
        actual_label=test_labels[index]  
        predicted_label=np.argmax(model.predict(test_images[index][np.newaxis,...,np.newaxis]))  
        plt.figure()  
        plt.title(f'actual:{actual_label} , predicted:{predicted_label}')  
        plt.imshow(test_images[index],cmap='gray')  
        plt.show()
```

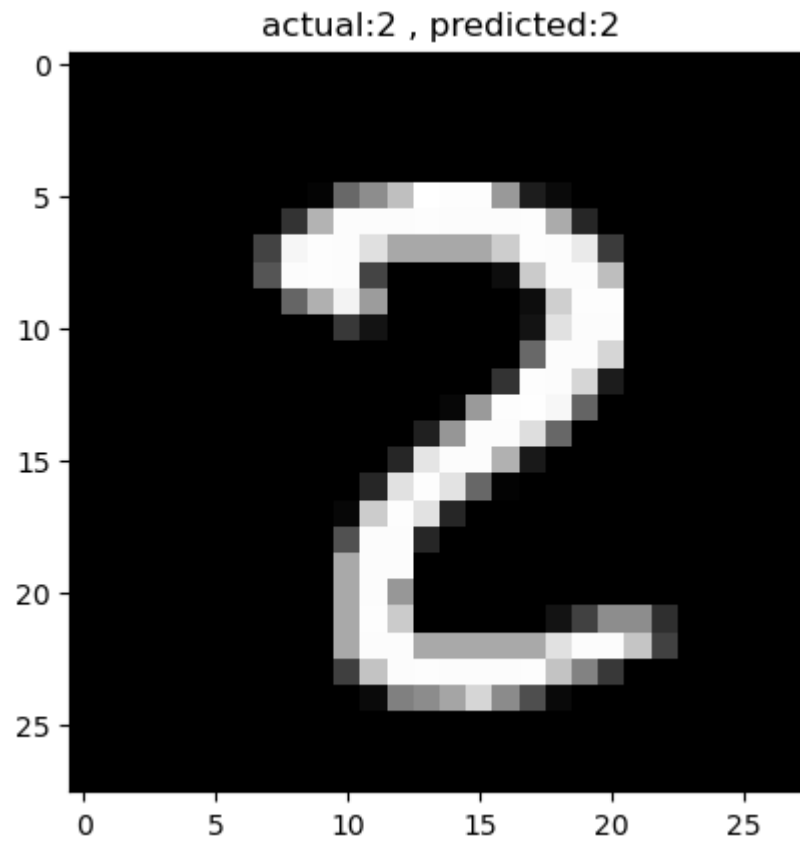
1/1 [=====] - 0s 23ms/step



1/1 [=====] - 0s 22ms/step



1/1 [=====] - 0s 21ms/step



In [ ]: