Name: Gajaanan S. Index No: 190185D

In [ ]:

```
#Q1
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import datasets, layers, models
import numpy as np
import matplotlib.pyplot as plt

mnist = keras.datasets.mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

# Padding
paddings = tf.constant([[0, 0], [2, 2], [2, 2]])
train_images = tf.pad(train_images, paddings, constant_values=0)
test_images = tf.pad(test_images, paddings, constant_values=0)

print('train_images.shape: ', train_images.shape)
print('train_labels.shape: ', train_labels.shape)
print('test_images.shape:', test_images.shape)
print('test_labels.shape:', test_labels.shape)
class_names = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

train_images = tf.dtypes.cast(train_images, tf.float32)
test_images = tf.dtypes.cast(test_images, tf.float32)
train_images, test_images = train_images[..., np.newaxis]/255.0, test_images[..., np.ne
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.
npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
train_images.shape:  (60000, 32, 32)
train_labels.shape:  (60000,)
test_images.shape: (10000, 32, 32)
test_labels.shape: (10000,)
```

In [ ]:

```
model = models.Sequential()
model.add(layers.Conv2D(6,(5,5),activation='relu',input_shape=(32,32,1)))
model.add(layers.AveragePooling2D((2,2)))
model.add(layers.Conv2D(16,(5,5),activation='relu'))
model.add(layers.AveragePooling2D((2,2)))
model.add(layers.Flatten())
model.add(layers.Dense(120,activation='relu'))
model.add(layers.Dense(84,activation='relu'))
model.add(layers.Dense(10))
model.compile(optimizer='adam',loss=tf.keras.losses.SparseCategoricalCrossentropy(from_
print(model.summary())
model.fit(train_images,train_labels,epochs=5)
test_loss,train_loss=model.evaluate(test_images,test_labels,verbose=2)
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 6)         156

 average_pooling2d (AverageP  (None, 14, 14, 6)        0
 ooling2D)
```

```
 conv2d_1 (Conv2D)           (None, 10, 10, 16)        2416

 average_pooling2d_1 (Averag  (None, 5, 5, 16)         0
 ePooling2D)

 flatten (Flatten)           (None, 400)               0

 dense (Dense)               (None, 120)               48120

 dense_1 (Dense)             (None, 84)                10164

 dense_2 (Dense)             (None, 10)                850

=================================================================
Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/5
1875/1875 [==============================] - 38s 20ms/step - loss: 0.2113 - accuracy: 0.
9346
Epoch 2/5
1875/1875 [==============================] - 37s 20ms/step - loss: 0.0653 - accuracy: 0.
9800
Epoch 3/5
1875/1875 [==============================] - 36s 19ms/step - loss: 0.0465 - accuracy: 0.
9855
Epoch 4/5
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0361 - accuracy: 0.
9886
Epoch 5/5
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0289 - accuracy: 0.
9910
313/313 - 3s - loss: 0.0402 - accuracy: 0.9874 - 3s/epoch - 9ms/step
```

In [ ]:

```python
#Q2
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import cifar10,mnist
(train_images,train_labels),(test_images,test_labels) = datasets.cifar10.load_data()

train_images,test_images = train_images / 255.0, test_images / 255.0
class_names = ['airplane','automobile','bird','cat','deer','dog','frog','horse','ship',
print('train_images.shape: ', train_images.shape)
print('train_labels.shape: ', train_labels.shape)
print('test_images.shape:', test_images.shape)
print('test_labels.shape:', test_labels.shape)
```

```
train_images.shape:  (50000, 32, 32, 3)
train_labels.shape:  (50000, 1)
test_images.shape: (10000, 32, 32, 3)
test_labels.shape: (10000, 1)
```

In [ ]:

```python
model = models.Sequential()
model.add(layers.Conv2D(32,(5,5),activation='relu',input_shape=(32,32,3)))
model.add(layers.MaxPool2D((2,2)))
model.add(layers.Conv2D(64,(5,5),activation='relu'))
model.add(layers.MaxPool2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPool2D((2,2)))
```

```
model.add(layers.Flatten())
model.add(layers.Dense(64,activation='relu'))
model.add(layers.Dense(10))

model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),loss=tf.keras.losses
print(model.summary())
model.fit(train_images,train_labels,epochs=5)
test_loss,test_acc = model.evaluate(test_images,test_labels,verbose=2)
print(test_acc)
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 28, 28, 32)        2432

 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0
 )

 conv2d_3 (Conv2D)           (None, 10, 10, 64)        51264

 max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 conv2d_4 (Conv2D)           (None, 3, 3, 128)         73856

 max_pooling2d_2 (MaxPooling  (None, 1, 1, 128)        0
 2D)

 flatten_1 (Flatten)         (None, 128)               0

 dense_3 (Dense)             (None, 64)                8256

 dense_4 (Dense)             (None, 10)                650

=================================================================
Total params: 136,458
Trainable params: 136,458
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/5
1563/1563 [==============================] - 98s 63ms/step - loss: 1.6396 - accuracy: 0.
3928
Epoch 2/5
1563/1563 [==============================] - 106s 68ms/step - loss: 1.2922 - accuracy:
0.5383
Epoch 3/5
1563/1563 [==============================] - 111s 71ms/step - loss: 1.1434 - accuracy:
0.5958
Epoch 4/5
1563/1563 [==============================] - 122s 78ms/step - loss: 1.0543 - accuracy:
0.6294
Epoch 5/5
1563/1563 [==============================] - 104s 67ms/step - loss: 0.9717 - accuracy:
0.6580
313/313 - 5s - loss: 1.0763 - accuracy: 0.6260 - 5s/epoch - 15ms/step
0.6259999871253967
```

In [ ]:

```
#Q3
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import datasets, layers, models
```

```
import numpy as np
import matplotlib.pyplot as plt

mnist = keras.datasets.mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

# Padding
paddings = tf.constant([[0, 0], [2, 2], [2, 2]])
train_images = tf.pad(train_images, paddings, constant_values=0)
test_images = tf.pad(test_images, paddings, constant_values=0)

print('train_images.shape: ', train_images.shape)
print('train_labels.shape: ', train_labels.shape)
print('test_images.shape:', test_images.shape)
print('test_labels.shape:', test_labels.shape)
class_names = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

train_images = tf.dtypes.cast(train_images, tf.float32)
test_images = tf.dtypes.cast(test_images, tf.float32)
train_images, test_images = train_images[..., np.newaxis]/255.0, test_images[..., np.ne

model_base = models.Sequential()
model_base.add(layers.Conv2D(32,(3,3),activation='relu',input_shape=(32,32,1)))
model_base.add(layers.MaxPool2D((2,2)))
model_base.add(layers.Conv2D(64,(3,3),activation='relu'))
model_base.add(layers.MaxPool2D((2,2)))
model_base.add(layers.Flatten())
model_base.add(layers.Dense(64,activation='relu'))
model_base.add(layers.Dense(10))

model_base.compile(optimizer=keras.optimizers.Adam() ,loss=tf.keras.losses.SparseCatego
print(model_base.summary())
model_base.fit(train_images,train_labels,epochs=2)
test_loss,test_acc = model_base.evaluate(test_images,test_labels,verbose=2)
model_base.save_weights('saved_weights/')
```

```
train_images.shape:  (60000, 32, 32)
train_labels.shape:  (60000,)
test_images.shape: (10000, 32, 32)
test_labels.shape: (10000,)
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 30, 30, 32)        320

 max_pooling2d_5 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_8 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_6 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 flatten_3 (Flatten)         (None, 2304)              0

 dense_7 (Dense)             (None, 64)                147520

 dense_8 (Dense)             (None, 10)                650

=================================================================
Total params: 166,986
```

```
Trainable params: 166,986
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/2
1875/1875 [==============================] - 66s 35ms/step - loss: 0.1416 - accuracy: 0.
9553
Epoch 2/2
1875/1875 [==============================] - 76s 41ms/step - loss: 0.0470 - accuracy: 0.
9857
313/313 - 3s - loss: 0.0391 - accuracy: 0.9882 - 3s/epoch - 9ms/step
```

In [ ]:

```
#Q4
model_lw = models.Sequential()
model_lw.add(layers.Conv2D(32,(3,3),activation='relu',input_shape=(32,32,1)))
model_lw.add(layers.MaxPool2D((2,2)))
model_lw.add(layers.Conv2D(64,(3,3),activation='relu'))
model_lw.add(layers.MaxPool2D((2,2)))
model_lw.add(layers.Flatten())
model_lw.add(layers.Dense(64,activation='relu'))
model_lw.add(layers.Dense(10))

model_lw.compile(optimizer=keras.optimizers.Adam() ,loss=tf.keras.losses.SparseCategori
print(model_lw.summary())

model_lw.load_weights('saved_weights/')

model_lw.fit(train_images,train_labels,epochs=2)
test_loss,test_acc = model_lw.evaluate(test_images,test_labels,verbose=2)
model_lw.save('saved_model/')
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_13 (Conv2D)          (None, 30, 30, 32)        320

 max_pooling2d_11 (MaxPoolin  (None, 15, 15, 32)       0
 g2D)

 conv2d_14 (Conv2D)          (None, 13, 13, 64)        18496

 max_pooling2d_12 (MaxPoolin  (None, 6, 6, 64)         0
 g2D)

 flatten_6 (Flatten)         (None, 2304)              0

 dense_13 (Dense)            (None, 64)                147520

 dense_14 (Dense)            (None, 10)                650

=================================================================
Total params: 166,986
Trainable params: 166,986
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/2
1875/1875 [==============================] - 68s 36ms/step - loss: 0.0320 - accuracy: 0.
9905
Epoch 2/2
1875/1875 [==============================] - 66s 35ms/step - loss: 0.0232 - accuracy: 0.
```

```
9926
313/313 - 3s - loss: 0.0299 - accuracy: 0.9900 - 3s/epoch - 10ms/step
INFO:tensorflow:Assets written to: saved_model/assets
```

In [ ]:

```
#Q5
model_id = keras.models.load_model('saved_model/')
print(model_id.summary())
model_id.evaluate(test_images,test_labels,verbose=2)
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_13 (Conv2D)          (None, 30, 30, 32)        320

 max_pooling2d_11 (MaxPoolin  (None, 15, 15, 32)       0
 g2D)

 conv2d_14 (Conv2D)          (None, 13, 13, 64)        18496

 max_pooling2d_12 (MaxPoolin  (None, 6, 6, 64)         0
 g2D)

 flatten_6 (Flatten)         (None, 2304)              0

 dense_13 (Dense)            (None, 64)                147520

 dense_14 (Dense)            (None, 10)                650

=================================================================
Total params: 166,986
Trainable params: 166,986
Non-trainable params: 0
_________________________________________________________________
None
313/313 - 4s - loss: 0.0299 - accuracy: 0.9900 - 4s/epoch - 14ms/step
```

Out[ ]:

```
[0.02990245446562767, 0.9900000095367432]
```

In [ ]:

```
#Q6
base_inputs = model_id.layers[0].input
base_outputs = model_id.layers[-2].output
output = layers.Dense(10)(base_outputs)

new_model = keras.Model(inputs=base_inputs,outputs=output)
new_model.compile(optimizer=keras.optimizers.Adam() ,loss=tf.keras.losses.SparseCategor
print(new_model.summary())

new_model.fit(train_images,train_labels,epochs=3, verbose=2)
new_model.evaluate(test_images,test_labels,verbose=2)
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_13_input (InputLayer  [(None, 32, 32, 1)]      0
 )

 conv2d_13 (Conv2D)          (None, 30, 30, 32)        320

 max_pooling2d_11 (MaxPoolin  (None, 15, 15, 32)       0
 g2D)
```

```
 conv2d_14 (Conv2D)          (None, 13, 13, 64)        18496

 max_pooling2d_12 (MaxPoolin  (None, 6, 6, 64)         0
 g2D)

 flatten_6 (Flatten)         (None, 2304)              0

 dense_13 (Dense)            (None, 64)                147520

 dense_15 (Dense)            (None, 10)                650

=================================================================
Total params: 166,986
Trainable params: 166,986
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/3
1875/1875 - 67s - loss: 0.0915 - accuracy: 0.9759 - 67s/epoch - 36ms/step
Epoch 2/3
1875/1875 - 69s - loss: 0.0204 - accuracy: 0.9934 - 69s/epoch - 37ms/step
Epoch 3/3
1875/1875 - 75s - loss: 0.0146 - accuracy: 0.9950 - 75s/epoch - 40ms/step
313/313 - 3s - loss: 0.0324 - accuracy: 0.9899 - 3s/epoch - 10ms/step
```

Out[ ]: `[0.03241473808884621, 0.9898999929428101]`

In [ ]:

```python
#Q7
model_for_tl = keras.models.load_model('saved_model/')
model_for_tl.trainable = False
for layer in model_for_tl.layers:
  assert layer.trainable == False

base_inputs = model_for_tl.layers[0].input
base_outputs = model_for_tl.layers[-2].output
output = layers.Dense(10)(base_outputs)

new_model = keras.Model(inputs=base_inputs,outputs=output)
new_model.compile(optimizer=keras.optimizers.Adam() ,loss=tf.keras.losses.SparseCategor

new_model.fit(train_images,train_labels,epochs=3, verbose=2)
new_model.evaluate(test_images,test_labels,verbose=2)
```

```
Epoch 1/3
1875/1875 - 19s - loss: 0.3359 - accuracy: 0.9257 - 19s/epoch - 10ms/step
Epoch 2/3
1875/1875 - 18s - loss: 0.0203 - accuracy: 0.9943 - 18s/epoch - 10ms/step
Epoch 3/3
1875/1875 - 19s - loss: 0.0134 - accuracy: 0.9963 - 19s/epoch - 10ms/step
313/313 - 3s - loss: 0.0250 - accuracy: 0.9913 - 3s/epoch - 10ms/step
```

Out[ ]: `[0.025021089240908623, 0.9912999868392944]`

In [3]:

```python
#Q8
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import datasets, layers, models

a = tf.random.normal(shape = (5, 224, 224, 3))
b = tf.constant([0, 1, 2, 3, 4])
ResNet = keras.applications.resnet_v2.ResNet50V2(include_top=True)
```

```
ResNet.trainable = False
for layer in ResNet.layers:
    assert layer.trainable == False

base_inputs = ResNet.layers[0].input
base_outputs = ResNet.layers[-2].output
output = layers.Dense(5)(base_outputs)

New_model = keras.Model(inputs = base_inputs, outputs = output)
New_model.compile(optimizer = keras.optimizers.Adam(), loss = tf.keras.losses.SparseCat
New_model.fit(a, b, epochs=15)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resne
t/resnet50v2_weights_tf_dim_ordering_tf_kernels.h5
102875136/102869336 [==============================] - 4s 0us/step
102883328/102869336 [==============================] - 4s 0us/step
Epoch 1/15
1/1 [==============================] - 5s 5s/step - loss: 1.9619 - accuracy: 0.0000e+00
Epoch 2/15
1/1 [==============================] - 1s 745ms/step - loss: 1.8368 - accuracy: 0.0000e+
00
Epoch 3/15
1/1 [==============================] - 1s 739ms/step - loss: 1.7402 - accuracy: 0.0000e+
00
Epoch 4/15
1/1 [==============================] - 1s 770ms/step - loss: 1.6703 - accuracy: 0.2000
Epoch 5/15
1/1 [==============================] - 1s 738ms/step - loss: 1.6217 - accuracy: 0.4000
Epoch 6/15
1/1 [==============================] - 1s 750ms/step - loss: 1.5856 - accuracy: 0.2000
Epoch 7/15
1/1 [==============================] - 1s 722ms/step - loss: 1.5542 - accuracy: 0.2000
Epoch 8/15
1/1 [==============================] - 1s 745ms/step - loss: 1.5224 - accuracy: 0.4000
Epoch 9/15
1/1 [==============================] - 1s 738ms/step - loss: 1.4877 - accuracy: 0.4000
Epoch 10/15
1/1 [==============================] - 1s 714ms/step - loss: 1.4495 - accuracy: 0.4000
Epoch 11/15
1/1 [==============================] - 1s 731ms/step - loss: 1.4088 - accuracy: 0.6000
Epoch 12/15
1/1 [==============================] - 1s 719ms/step - loss: 1.3670 - accuracy: 0.6000
Epoch 13/15
1/1 [==============================] - 1s 716ms/step - loss: 1.3258 - accuracy: 0.8000
Epoch 14/15
1/1 [==============================] - 1s 702ms/step - loss: 1.2860 - accuracy: 0.8000
Epoch 15/15
1/1 [==============================] - 1s 741ms/step - loss: 1.2483 - accuracy: 0.8000
```

Out[3]: <keras.callbacks.History at 0x7f5379461b50>