

Index No: 190643G

Name: Vakeesan.K

Q1)

```
In [1]: 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.newaxis]/255.0

train_images.shape: (60000, 32, 32)
train_labels.shape: (60000,)
test_images.shape: (10000, 32, 32)
test_labels.shape: (10000,)

In [2]: 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_logits=True))
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"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 28, 28, 6)	156
average_pooling2d (AveragePooling2D)	(None, 14, 14, 6)	0
conv2d_1 (Conv2D)	(None, 10, 10, 16)	2416
average_pooling2d_1 (AveragePooling2D)	(None, 5, 5, 16)	0
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 [=====] - 16s 8ms/step - loss: 0.2205 - accuracy: 0.9314
Epoch 2/5
1875/1875 [=====] - 14s 8ms/step - loss: 0.0694 - accuracy: 0.9790
Epoch 3/5
1875/1875 [=====] - 15s 8ms/step - loss: 0.0495 - accuracy: 0.9850
Epoch 4/5
1875/1875 [=====] - 15s 8ms/step - loss: 0.0387 - accuracy: 0.9883
Epoch 5/5
1875/1875 [=====] - 15s 8ms/step - loss: 0.0322 - accuracy: 0.9900
313/313 - 1s - loss: 0.0349 - accuracy: 0.9884 - 1s/epoch - 4ms/step
0.9883999824523926
```

Q2)

```
In [3]: #CIFAR10
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import cifar10, mnist
import tensorflow as tf
import matplotlib.pyplot as plt
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship',
```

```
In [4]: 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,(3,3),activation='relu'))
model.add(layers.MaxPool2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(10))

model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),loss=tf.keras.losses.SparseCa

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, 12, 12, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_4 (Conv2D)	(None, 4, 4, 128)	73856
flatten_1 (Flatten)	(None, 2048)	0
dense_3 (Dense)	(None, 10)	20490

Total params: 115,274
Trainable params: 115,274
Non-trainable params: 0

None
Epoch 1/5
1563/1563 [=====] - 30s 19ms/step - loss: 1.4941 - accuracy: 0.4582
Epoch 2/5
1563/1563 [=====] - 30s 19ms/step - loss: 1.1357 - accuracy: 0.6019
Epoch 3/5
1563/1563 [=====] - 30s 19ms/step - loss: 0.9686 - accuracy: 0.6642
Epoch 4/5
1563/1563 [=====] - 30s 19ms/step - loss: 0.8652 - accuracy: 0.6984
Epoch 5/5
1563/1563 [=====] - 29s 18ms/step - loss: 0.7762 - accuracy: 0.7305
313/313 - 2s - loss: 0.9207 - accuracy: 0.6843 - 2s/epoch - 7ms/step
0.6843000054359436

Q3)

```
In [6]: 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[...]/255.0, test_images[...]/255.0

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.Conv2D(64,(3,3),activation='relu'))
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.SparseCategoricalCross
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_8 (Conv2D)	(None, 30, 30, 32)	320
max_pooling2d_4 (MaxPooling 2D)	(None, 15, 15, 32)	0
conv2d_9 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_5 (MaxPooling 2D)	(None, 6, 6, 64)	0
conv2d_10 (Conv2D)	(None, 4, 4, 64)	36928
flatten_3 (Flatten)	(None, 1024)	0
dense_6 (Dense)	(None, 64)	65600
dense_7 (Dense)	(None, 10)	650
=====		
Total params: 121,994		
Trainable params: 121,994		
Non-trainable params: 0		

```

None
Epoch 1/2
1875/1875 [=====] - 31s 16ms/step - loss: 0.1332 - accuracy: 0.9585
Epoch 2/2
1875/1875 [=====] - 30s 16ms/step - loss: 0.0423 - accuracy: 0.9870
313/313 - 2s - loss: 0.0395 - accuracy: 0.9878 - 2s/epoch - 6ms/step

```

Q4)

```

In [13]: 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.Conv2D(64,(3,3),activation='relu'))
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.SparseCategoricalCrossen
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_8"

Layer (type)	Output Shape	Param #
=====		
conv2d_23 (Conv2D)	(None, 30, 30, 32)	320
max_pooling2d_14 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_24 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_15 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_25 (Conv2D)	(None, 4, 4, 64)	36928
flatten_8 (Flatten)	(None, 1024)	0
dense_16 (Dense)	(None, 64)	65600
dense_17 (Dense)	(None, 10)	650
=====		
Total params: 121,994		
Trainable params: 121,994		
Non-trainable params: 0		

None
Epoch 1/2
1875/1875 [=====] - 30s 16ms/step - loss: 0.0305 - accuracy: 0.9901
Epoch 2/2
1875/1875 [=====] - 29s 16ms/step - loss: 0.0235 - accuracy: 0.9925
313/313 - 2s - loss: 0.0294 - accuracy: 0.9896 - 2s/epoch - 6ms/step
INFO:tensorflow:Assets written to: saved_model/assets

Q5)

```
In [19]: #Loading the model
model_ld = keras.models.load_model("saved_model/")
print(model_ld.summary())
model_ld.evaluate(test_images, test_labels, verbose=2)
```

Model: "sequential_8"

Layer (type)	Output Shape	Param #
=====		
conv2d_23 (Conv2D)	(None, 30, 30, 32)	320
max_pooling2d_14 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_24 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_15 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_25 (Conv2D)	(None, 4, 4, 64)	36928
flatten_8 (Flatten)	(None, 1024)	0
dense_16 (Dense)	(None, 64)	65600
dense_17 (Dense)	(None, 10)	650
=====		
Total params: 121,994		
Trainable params: 121,994		
Non-trainable params: 0		

None
313/313 - 2s - loss: 0.0294 - accuracy: 0.9896 - 2s/epoch - 7ms/step
[0.02937854640185833, 0.9896000027656555]

Out[19]:

Q6)

```
In [24]: #fine tuning
base_inputs = model_ld.layers[0].input
base_outputs= model_ld.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.SparseCategoricalCrosse
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_2"

Layer (type)	Output Shape	Param #
conv2d_23_input (InputLayer)	[(None, 32, 32, 1)]	0
conv2d_23 (Conv2D)	(None, 30, 30, 32)	320
max_pooling2d_14 (MaxPoolin g2D)	(None, 15, 15, 32)	0
conv2d_24 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_15 (MaxPoolin g2D)	(None, 6, 6, 64)	0
conv2d_25 (Conv2D)	(None, 4, 4, 64)	36928
flatten_8 (Flatten)	(None, 1024)	0
dense_16 (Dense)	(None, 64)	65600
dense_26 (Dense)	(None, 10)	650
=====		
Total params: 121,994		
Trainable params: 121,994		
Non-trainable params: 0		

```
None
Epoch 1/3
1875/1875 - 29s - loss: 0.0788 - accuracy: 0.9791 - 29s/epoch - 16ms/step
Epoch 2/3
1875/1875 - 29s - loss: 0.0183 - accuracy: 0.9943 - 29s/epoch - 16ms/step
Epoch 3/3
1875/1875 - 28s - loss: 0.0137 - accuracy: 0.9957 - 28s/epoch - 15ms/step
313/313 - 2s - loss: 0.0299 - accuracy: 0.9912 - 2s/epoch - 6ms/step
[0.02991761453449726, 0.9911999702453613]
```

Out[24]:

Q7)

```
In [26]: #transfer Learning
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.SparseCategoricalCrosse
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_3"

Layer (type)	Output Shape	Param #
=====		
conv2d_23_input (InputLayer)	[(None, 32, 32, 1)]	0
conv2d_23 (Conv2D)	(None, 30, 30, 32)	320
max_pooling2d_14 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_24 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_15 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_25 (Conv2D)	(None, 4, 4, 64)	36928
flatten_8 (Flatten)	(None, 1024)	0
dense_16 (Dense)	(None, 64)	65600
dense_27 (Dense)	(None, 10)	650

```
=====
Total params: 121,994
Trainable params: 650
Non-trainable params: 121,344
```

```
None
Epoch 1/3
1875/1875 - 12s - loss: 0.2314 - accuracy: 0.9488 - 12s/epoch - 6ms/step
Epoch 2/3
1875/1875 - 10s - loss: 0.0157 - accuracy: 0.9959 - 10s/epoch - 6ms/step
Epoch 3/3
1875/1875 - 10s - loss: 0.0112 - accuracy: 0.9968 - 10s/epoch - 6ms/step
313/313 - 2s - loss: 0.0226 - accuracy: 0.9925 - 2s/epoch - 6ms/step
[0.022625740617513657, 0.9925000071525574]
```

Out[26]: