

Exercise 11

May 30, 2022

Name: B.S.V.W. Munasinghe

Index Number: 190397E

```
[ ]: #Importing Libraries
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import datasets, layers, models

%matplotlib inline
```

1 Question 1

```
[ ]: 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,)
```

```
[ ]: 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),metrics=['accuracy'])
print(model.summary())

model.fit(train_images,train_labels,epochs=5)
test_loss,train_accuracy=model.evaluate(test_images,test_labels,verbose=2)
```

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: 0.2242 -
accuracy: 0.9316
Epoch 2/5
1875/1875 [=====] - 17s 9ms/step - loss: 0.0676 -
accuracy: 0.9788
Epoch 3/5
1875/1875 [=====] - 18s 10ms/step - loss: 0.0488 -
accuracy: 0.9845
Epoch 4/5
1875/1875 [=====] - 16s 9ms/step - loss: 0.0375 -
accuracy: 0.9880
Epoch 5/5
1875/1875 [=====] - 17s 9ms/step - loss: 0.0307 -
accuracy: 0.9902
313/313 - 1s - loss: 0.0370 - accuracy: 0.9878 - 1s/epoch - 4ms/step

```

2 Question 2

```

[ ]: from tensorflow.keras.datasets import cifar10, mnist
      (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', 'truck']

```

```

[ ]: 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(64,activation='relu'))
      model.add(layers.Dense(10))

      model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),loss = tf.
      ↪keras.losses.
      ↪SparseCategoricalCrossentropy(from_logits=True),metrics=['accuracy'])
      print(model.summary())

      model.fit(train_images,train_labels,epochs=5)
      test_loss,train_accuracy=model.evaluate(test_images,test_labels,verbose=2)

```

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, 64)	131136
dense_4 (Dense)	(None, 10)	650

Total params: 226,570
 Trainable params: 226,570
 Non-trainable params: 0

Epoch 1/5
 1563/1563 [=====] - 30s 19ms/step - loss: 1.5492 - accuracy: 0.4368
 Epoch 2/5
 1563/1563 [=====] - 32s 20ms/step - loss: 1.1972 - accuracy: 0.5759
 Epoch 3/5
 1563/1563 [=====] - 32s 20ms/step - loss: 1.0430 - accuracy: 0.6349
 Epoch 4/5
 1563/1563 [=====] - 33s 21ms/step - loss: 0.9348 - accuracy: 0.6685
 Epoch 5/5
 1563/1563 [=====] - 32s 21ms/step - loss: 0.8409 - accuracy: 0.7063
 313/313 - 2s - loss: 0.9337 - accuracy: 0.6762 - 2s/epoch - 6ms/step

3 Question 3

```
[ ]: 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

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.
    ↪SparseCategoricalCrossentropy(from_logits=True),metrics=['accuracy'])
print(model_base.summary())

model_base.fit(train_images,train_labels,epochs=2)
test_loss,train_accuracy=model_base.evaluate(test_images,test_labels,verbose=2)
model_base.save_weights('saved_model_weights/')
```

```
train_images.shape: (60000, 32, 32)
train_labels.shape: (60000,)
test_images.shape: (10000, 32, 32)
test_labels.shape: (10000,)
Model: "sequential_2"
```

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 30, 30, 32)	320

max_pooling2d_2 (MaxPooling 2D)	(None, 15, 15, 32)	0
conv2d_6 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_3 (MaxPooling 2D)	(None, 6, 6, 64)	0
conv2d_7 (Conv2D)	(None, 4, 4, 64)	36928
flatten_2 (Flatten)	(None, 1024)	0
dense_5 (Dense)	(None, 64)	65600
dense_6 (Dense)	(None, 10)	650

```

=====
Total params: 121,994
Trainable params: 121,994
Non-trainable params: 0
-----
None
Epoch 1/2
1875/1875 [=====] - 29s 15ms/step - loss: 0.1411 -
accuracy: 0.9557
Epoch 2/2
1875/1875 [=====] - 33s 18ms/step - loss: 0.0433 -
accuracy: 0.9868
313/313 - 2s - loss: 0.0301 - accuracy: 0.9910 - 2s/epoch - 6ms/step

```

4 Question 4

```

[ ]: 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.
    ↪SparseCategoricalCrossentropy(from_logits=True),metrics=['accuracy'])
print(model_lw.summary())

```

```

model_lw.load_weights('saved_model_weights/')

model_lw.fit(train_images,train_labels,epochs=2)
test_loss,train_accuracy=model_lw.evaluate(test_images,test_labels,verbose=2)

model_lw.save('saved_model/')

```

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_7 (Dense)	(None, 64)	65600
dense_8 (Dense)	(None, 10)	650

```

=====
Total params: 121,994
Trainable params: 121,994
Non-trainable params: 0

```

```

-----
None
Epoch 1/2
1875/1875 [=====] - 29s 15ms/step - loss: 0.0297 -
accuracy: 0.9903
Epoch 2/2
1875/1875 [=====] - 33s 18ms/step - loss: 0.0228 -
accuracy: 0.9925
313/313 - 2s - loss: 0.0353 - accuracy: 0.9880 - 2s/epoch - 6ms/step

```

WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 3 of 3). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: saved_model/assets

INFO:tensorflow:Assets written to: saved_model/assets

5 Question 5

```
[ ]: model_ld = keras.models.load_model('saved_model/')
      print(model_ld.summary())
      model_ld.evaluate(test_images, test_labels, verbose=2)
```

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 30, 30, 32)	320
max_pooling2d_4 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_9 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_5 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_10 (Conv2D)	(None, 4, 4, 64)	36928
flatten_3 (Flatten)	(None, 1024)	0
dense_7 (Dense)	(None, 64)	65600
dense_8 (Dense)	(None, 10)	650

=====
Total params: 121,994
Trainable params: 121,994
Non-trainable params: 0
=====
None
313/313 - 2s - loss: 0.0353 - accuracy: 0.9880 - 2s/epoch - 6ms/step

```
[ ]: [0.035340044647455215, 0.9879999756813049]
```

6 Question 6

```
[ ]: 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.
    ↳SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])
print(new_model.summary())

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

```

Model: "model"

Layer (type)	Output Shape	Param #
conv2d_8_input (InputLayer)	[(None, 32, 32, 1)]	0
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_7 (Dense)	(None, 64)	65600
dense_9 (Dense)	(None, 10)	650

=====
Total params: 121,994

Trainable params: 121,994

Non-trainable params: 0

None

Epoch 1/2

1875/1875 - 26s - loss: 0.0850 - accuracy: 0.9773 - 26s/epoch - 14ms/step

Epoch 2/2

1875/1875 - 29s - loss: 0.0192 - accuracy: 0.9942 - 29s/epoch - 16ms/step

313/313 - 2s - loss: 0.0302 - accuracy: 0.9910 - 2s/epoch - 6ms/step

[]: [0.030212702229619026, 0.9909999966621399]

7 Question 7

```
[ ]: 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.
    ↪SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])

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

Epoch 1/2

1875/1875 - 10s - loss: 0.2110 - accuracy: 0.9537 - 10s/epoch - 6ms/step

Epoch 2/2

1875/1875 - 10s - loss: 0.0156 - accuracy: 0.9954 - 10s/epoch - 5ms/step

313/313 - 2s - loss: 0.0272 - accuracy: 0.9918 - 2s/epoch - 5ms/step

```
[ ]: [0.027205144986510277, 0.9918000102043152]
```

8 Question 8

```
[ ]: resnet_model = models.Sequential()
#Import the pre-trained model as non trainable layers
pretrained_model= tf.keras.applications.ResNet50(include_top=False,
                                                    input_shape=(32,32,3),
                                                    pooling='avg',
                                                    weights='imagenet')

for layer in pretrained_model.layers:
    layer.trainable=False

#Add the downloaded model to the newly created model
resnet_model.add(pretrained_model)
```

```
[ ]: # Flatten the model and add 5 node layers to it
resnet_model.add(layers.Flatten())
resnet_model.add(layers.Dense(5, activation='relu'))
```

```
[ ]: resnet_model.compile(optimizer=keras.optimizers.Adam(), loss = tf.keras.losses.
    ↪SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])
```

```
print(resnet_model.summary())
```

Model: "sequential_3"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2048)	23587712
flatten_6 (Flatten)	(None, 2048)	0
dense_6 (Dense)	(None, 5)	10245

=====
Total params: 23,597,957
Trainable params: 10,245
Non-trainable params: 23,587,712
=====

None