Index:190098M

Q01)

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

mnist·=·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
```

```
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,)
```

```
model=keras.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=keras.losses.SparseCategoricalCrossentropy(from_logits=True),metrics=['accuracy'])
print(model.summary())
model.fit(train_images,train_labels,epochs=5)

test_loss,test_acc=model.evaluate(test_images,test_labels,verbose=2)
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_2 (Conv2D)           (None, 28, 28, 6)         156       
                                                                 
 average_pooling2d_2 (Averag  (None, 14, 14, 6)        0         
 ePooling2D)                                                     
                                                                 
 conv2d_3 (Conv2D)           (None, 10, 10, 16)        2416      
                                                                 
 average_pooling2d_3 (Averag  (None, 5, 5, 16)         0         
 ePooling2D)                                                     
                                                                 
 flatten_1 (Flatten)         (None, 400)               0         
                                                                 
 dense_3 (Dense)             (None, 120)               48120     
                                                                 
 dense_4 (Dense)             (None, 84)                10164     
                                                                 
 dense_5 (Dense)             (None, 10)                850       
                                                                 
=================================================================
Total params: 61,706
Trainable params: 61,706
```

```
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/5
1875/1875 [==============================] - 39s 21ms/step - loss: 0.2127 - accuracy: 0.9358
Epoch 2/5
1875/1875 [==============================] - 39s 21ms/step - loss: 0.0756 - accuracy: 0.9766
Epoch 3/5
1875/1875 [==============================] - 38s 21ms/step - loss: 0.0526 - accuracy: 0.9837
Epoch 4/5
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0408 - accuracy: 0.9872
Epoch 5/5
1875/1875 [==============================] - 39s 21ms/step - loss: 0.0332 - accuracy: 0.9893
313/313 - 3s - loss: 0.0401 - accuracy: 0.9869 - 3s/epoch - 8ms/step
```

Q02)

```
(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','autombile','bird','cat','deer','dog','frog','horse','ship','truck']
model=keras.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=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])

print(model.summary())

model.fit(train_images,train_labels,epochs=5)

test_loss,test_acc=model.evaluate(test_images,test_labels,verbose=2)
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 2s 0us/step
170508288/170498071 [==============================] - 2s 0us/step
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_4 (Conv2D)           (None, 28, 28, 32)        2432      
                                                                 
 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0         
 )                                                               
                                                                 
 conv2d_5 (Conv2D)           (None, 12, 12, 64)        18496     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0         
 2D)                                                             
                                                                 
 conv2d_6 (Conv2D)           (None, 4, 4, 128)         73856     
                                                                 
 flatten_2 (Flatten)         (None, 2048)              0         
                                                                 
 dense_6 (Dense)             (None, 64)                131136    
                                                                 
 dense_7 (Dense)             (None, 10)                650       
                                                                 
=================================================================
Total params: 226,570
Trainable params: 226,570
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/5
1563/1563 [==============================] - 92s 59ms/step - loss: 1.5440 - accuracy: 0.4359
Epoch 2/5
1563/1563 [==============================] - 91s 58ms/step - loss: 1.1444 - accuracy: 0.5966
Epoch 3/5
1563/1563 [==============================] - 91s 59ms/step - loss: 0.9611 - accuracy: 0.6623
Epoch 4/5
1563/1563 [==============================] - 90s 58ms/step - loss: 0.8433 - accuracy: 0.7056
Epoch 5/5
1563/1563 [==============================] - 90s 58ms/step - loss: 0.7597 - accuracy: 0.7340
313/313 - 4s - loss: 0.9092 - accuracy: 0.6865 - 4s/epoch - 14ms/step
```

Q03)

```
import tensorflow as tf
from tensorflow import keras
from keras import layers,datasets
import numpy as np
import matplotlib.pyplot as plt
mnist = 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)

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=keras.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=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])

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/')
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_7 (Conv2D)           (None, 30, 30, 32)        320       
                                                                 
 max_pooling2d_2 (MaxPooling  (None, 15, 15, 32)       0         
 2D)                                                             
                                                                 
 conv2d_8 (Conv2D)           (None, 13, 13, 64)        18496     
                                                                 
 max_pooling2d_3 (MaxPooling  (None, 6, 6, 64)         0         
 2D)                                                             
                                                                 
 conv2d_9 (Conv2D)           (None, 4, 4, 64)          36928     
                                                                 
 flatten_3 (Flatten)         (None, 1024)              0         
                                                                 
 dense_8 (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 [==============================] - 79s 42ms/step - loss: 0.1321 - accuracy: 0.9591
Epoch 2/2
1875/1875 [==============================] - 78s 42ms/step - loss: 0.0412 - accuracy: 0.9866
313/313 - 4s - loss: 0.0313 - accuracy: 0.9904 - 4s/epoch - 11ms/step
```

Q04)

```
model_lw=keras.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=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])

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_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_10 (Conv2D)          (None, 30, 30, 32)        320       
                                                                 
 max_pooling2d_4 (MaxPooling  (None, 15, 15, 32)       0         
 2D)                                                             
                                                                 
 conv2d_11 (Conv2D)          (None, 13, 13, 64)        18496     
                                                                 
 max_pooling2d_5 (MaxPooling  (None, 6, 6, 64)         0         
 2D)                                                             
                                                                 
 conv2d_12 (Conv2D)          (None, 4, 4, 64)          36928     
                                                                 
 flatten_4 (Flatten)         (None, 1024)              0         
                                                                 
 dense_10 (Dense)            (None, 64)                65600     
                                                                 
 dense_11 (Dense)            (None, 10)                650       
                                                                 
=================================================================
Total params: 121,994
Trainable params: 121,994
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/2
1875/1875 [==============================] - 89s 47ms/step - loss: 0.0289 - accuracy: 0.9910
Epoch 2/2
1875/1875 [==============================] - 84s 45ms/step - loss: 0.0221 - accuracy: 0.9928
313/313 - 4s - loss: 0.0317 - accuracy: 0.9902 - 4s/epoch - 11ms/step
INFO:tensorflow:Assets written to: saved_model/assets
```

Q05)

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

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_10 (Conv2D)          (None, 30, 30, 32)        320       
                                                                 
 max_pooling2d_4 (MaxPooling  (None, 15, 15, 32)       0         
 2D)                                                             
                                                                 
 conv2d_11 (Conv2D)          (None, 13, 13, 64)        18496     
                                                                 
 max_pooling2d_5 (MaxPooling  (None, 6, 6, 64)         0         
 2D)                                                             
                                                                 
 conv2d_12 (Conv2D)          (None, 4, 4, 64)          36928     
                                                                 
 flatten_4 (Flatten)         (None, 1024)              0         
                                                                 
 dense_10 (Dense)            (None, 64)                65600     
 dense_11 (Dense)            (None, 10)                650

=================================================================
Total params: 121,994
Trainable params: 121,994
Non-trainable params: 0
_________________________________________________________________
None
313/313 - 4s - loss: 0.0317 - accuracy: 0.9902 - 4s/epoch - 13ms/step
[0.03165138512849808, 0.9901999831199646]
```

Q06)

```
base_innputs=model_ld.layers[0].input
base_ouputs=model_ld.layers[-2].output
output=layers.Dense(10)(base_ouputs)

new_model=keras.Model(inputs=base_innputs,outputs=output)
new_model.compile(optimizer=keras.optimizers.Adam(),loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),metrics=['accur
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_10_input (InputLayer  [(None, 32, 32, 1)]      0
 )

 conv2d_10 (Conv2D)          (None, 30, 30, 32)        320

 max_pooling2d_4 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_11 (Conv2D)          (None, 13, 13, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_12 (Conv2D)          (None, 4, 4, 64)          36928

 flatten_4 (Flatten)         (None, 1024)              0

 dense_10 (Dense)            (None, 64)                65600

 dense_12 (Dense)            (None, 10)                650

=================================================================
Total params: 121,994
Trainable params: 121,994
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/3
1875/1875 - 71s - loss: 0.0739 - accuracy: 0.9802 - 71s/epoch - 38ms/step
Epoch 2/3
1875/1875 - 69s - loss: 0.0188 - accuracy: 0.9938 - 69s/epoch - 37ms/step
Epoch 3/3
1875/1875 - 69s - loss: 0.0138 - accuracy: 0.9958 - 69s/epoch - 37ms/step
313/313 - 3s - loss: 0.0285 - accuracy: 0.9905 - 3s/epoch - 11ms/step
[0.028453951701521873, 0.9904999732971191]
```

Q07)

```
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_ouputs=model_for_tl.layers[-2].output
output=layers.Dense(10)(base_ouputs)

model_for_tl=keras.Model(inputs=base_inputs,outputs=output)
model_for_tl.compile(optimizer=keras.optimizers.Adam(),
    loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'])
print(model_for_tl.summary())
```

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

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_10_input (InputLayer  [(None, 32, 32, 1)]      0
 )

 conv2d_10 (Conv2D)          (None, 30, 30, 32)        320

 max_pooling2d_4 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_11 (Conv2D)          (None, 13, 13, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_12 (Conv2D)          (None, 4, 4, 64)          36928

 flatten_4 (Flatten)         (None, 1024)              0

 dense_10 (Dense)            (None, 64)                65600

 dense_13 (Dense)            (None, 10)                650

=================================================================
Total params: 121,994
Trainable params: 650
Non-trainable params: 121,344
_________________________________________________________________
None
Epoch 1/3
1875/1875 - 22s - loss: 0.1769 - accuracy: 0.9567 - 22s/epoch - 12ms/step
Epoch 2/3
1875/1875 - 22s - loss: 0.0158 - accuracy: 0.9959 - 22s/epoch - 12ms/step
Epoch 3/3
1875/1875 - 22s - loss: 0.0112 - accuracy: 0.9968 - 22s/epoch - 12ms/step
313/313 - 4s - loss: 0.0221 - accuracy: 0.9935 - 4s/epoch - 11ms/step
[0.022127004340291023, 0.9934999942779541]
```

Q08)

```
model_for_tl=keras.applications.resnet_v2.ResNet50V2()

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_ouputs=model_for_tl.layers[-2].output
output=layers.Dense(5)(base_ouputs)

model_for_tl=keras.Model(inputs=base_inputs,outputs=output)
model_for_tl.compile(optimizer=keras.optimizers.Adam(),
    loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'])
print(model_for_tl.summary())
```

```
 conv5_block1_3_conv (Conv2D)   (None, 7, 7, 2048)   1050624     ['conv5_block1_2_relu[0][0]']

 conv5_block1_out (Add)         (None, 7, 7, 2048)   0           ['conv5_block1_0_conv[0][0]',
                                                                  'conv5_block1_3_conv[0][0]']

 conv5_block2_preact_bn (BatchN  (None, 7, 7, 2048)  8192        ['conv5_block1_out[0][0]']
 ormalization)

 conv5_block2_preact_relu (Acti  (None, 7, 7, 2048)  0           ['conv5_block2_preact_bn[0][0]']
 vation)

 conv5_block2_1_conv (Conv2D)   (None, 7, 7, 512)    1048576     ['conv5_block2_preact_relu[0][0]'
                                                                 ]

 conv5_block2_1_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block2_1_conv[0][0]']
 ization)

 conv5_block2_1_relu (Activatio  (None, 7, 7, 512)   0           ['conv5_block2_1_bn[0][0]']
 n)

 conv5_block2_2_pad (ZeroPaddin  (None, 9, 9, 512)   0           ['conv5_block2_1_relu[0][0]']
 g2D)

 conv5_block2_2_conv (Conv2D)   (None, 7, 7, 512)    2359296     ['conv5_block2_2_pad[0][0]']
 conv5_block2_2_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block2_2_conv[0][0]']
 ization)

 conv5_block2_2_relu (Activatio  (None, 7, 7, 512)   0           ['conv5_block2_2_bn[0][0]']
 n)

 conv5_block2_3_conv (Conv2D)   (None, 7, 7, 2048)   1050624     ['conv5_block2_2_relu[0][0]']

 conv5_block2_out (Add)         (None, 7, 7, 2048)   0           ['conv5_block1_out[0][0]',
                                                                  'conv5_block2_3_conv[0][0]']

 conv5_block3_preact_bn (BatchN  (None, 7, 7, 2048)  8192        ['conv5_block2_out[0][0]']
 ormalization)

 conv5_block3_preact_relu (Acti  (None, 7, 7, 2048)  0           ['conv5_block3_preact_bn[0][0]']
 vation)

 conv5_block3_1_conv (Conv2D)   (None, 7, 7, 512)    1048576     ['conv5_block3_preact_relu[0][0]'
                                                                 ]

 conv5_block3_1_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block3_1_conv[0][0]']
 ization)

 conv5_block3_1_relu (Activatio  (None, 7, 7, 512)   0           ['conv5_block3_1_bn[0][0]']
 n)

 conv5_block3_2_pad (ZeroPaddin  (None, 9, 9, 512)   0           ['conv5_block3_1_relu[0][0]']
 g2D)

 conv5_block3_2_conv (Conv2D)   (None, 7, 7, 512)    2359296     ['conv5_block3_2_pad[0][0]']

 conv5_block3_2_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block3_2_conv[0][0]']
 ization)
```

```
train_images=tf.random.normal(shape=(5,224, 224, 3))
train_labels=tf.constant([0,1,2,3,4])

model_for_tl.fit(train_images,train_labels,epochs=30,verbose=2)
```

```
Epoch 1/30
1/1 - 1s - loss: 1.6908 - accuracy: 0.2000 - 1s/epoch - 1s/step
Epoch 2/30
1/1 - 1s - loss: 1.6799 - accuracy: 0.2000 - 1s/epoch - 1s/step
Epoch 3/30
1/1 - 1s - loss: 1.6604 - accuracy: 0.4000 - 703ms/epoch - 703ms/step
Epoch 4/30
1/1 - 1s - loss: 1.6359 - accuracy: 0.4000 - 696ms/epoch - 696ms/step
Epoch 5/30
1/1 - 1s - loss: 1.6090 - accuracy: 0.6000 - 717ms/epoch - 717ms/step
Epoch 6/30
1/1 - 1s - loss: 1.5815 - accuracy: 0.6000 - 696ms/epoch - 696ms/step
Epoch 7/30
1/1 - 1s - loss: 1.5542 - accuracy: 0.4000 - 692ms/epoch - 692ms/step
Epoch 8/30
1/1 - 1s - loss: 1.5272 - accuracy: 0.4000 - 693ms/epoch - 693ms/step
Epoch 9/30
1/1 - 1s - loss: 1.5003 - accuracy: 0.6000 - 686ms/epoch - 686ms/step
Epoch 10/30
1/1 - 1s - loss: 1.4735 - accuracy: 0.6000 - 691ms/epoch - 691ms/step
Epoch 11/30
1/1 - 1s - loss: 1.4469 - accuracy: 0.6000 - 694ms/epoch - 694ms/step
Epoch 12/30
1/1 - 1s - loss: 1.4209 - accuracy: 0.6000 - 682ms/epoch - 682ms/step
Epoch 13/30
1/1 - 1s - loss: 1.3958 - accuracy: 0.6000 - 695ms/epoch - 695ms/step
Epoch 14/30
1/1 - 1s - loss: 1.3713 - accuracy: 0.6000 - 705ms/epoch - 705ms/step
Epoch 15/30
1/1 - 1s - loss: 1.3469 - accuracy: 0.6000 - 682ms/epoch - 682ms/step
Epoch 16/30
1/1 - 1s - loss: 1.3222 - accuracy: 0.6000 - 695ms/epoch - 695ms/step
Epoch 17/30
1/1 - 1s - loss: 1.2965 - accuracy: 0.6000 - 698ms/epoch - 698ms/step
Epoch 18/30
1/1 - 1s - loss: 1.2697 - accuracy: 0.8000 - 684ms/epoch - 684ms/step
Epoch 19/30
1/1 - 1s - loss: 1.2422 - accuracy: 0.8000 - 688ms/epoch - 688ms/step
Epoch 20/30
1/1 - 1s - loss: 1.2143 - accuracy: 0.8000 - 703ms/epoch - 703ms/step
Epoch 21/30
1/1 - 1s - loss: 1.1866 - accuracy: 0.8000 - 683ms/epoch - 683ms/step
Epoch 22/30
1/1 - 1s - loss: 1.1595 - accuracy: 1.0000 - 700ms/epoch - 700ms/step
Epoch 23/30
1/1 - 1s - loss: 1.1335 - accuracy: 1.0000 - 695ms/epoch - 695ms/step
```

```
Epoch 24/30
1/1 - 1s - loss: 1.1085 - accuracy: 1.0000 - 681ms/epoch - 681ms/step
Epoch 25/30
1/1 - 1s - loss: 1.0845 - accuracy: 1.0000 - 702ms/epoch - 702ms/step
Epoch 26/30
1/1 - 1s - loss: 1.0612 - accuracy: 1.0000 - 703ms/epoch - 703ms/step
Epoch 27/30
1/1 - 1s - loss: 1.0383 - accuracy: 1.0000 - 695ms/epoch - 695ms/step
Epoch 28/30
1/1 - 1s - loss: 1.0155 - accuracy: 1.0000 - 686ms/epoch - 686ms/step
Epoch 29/30
1/1 - 1s - loss: 0.9928 - accuracy: 1.0000 - 698ms/epoch - 698ms/step
```

✓ 21s completed at 9:42 PM