

Untitled1

October 23, 2024

```
[1]: import tensorflow as tf
      from tensorflow.keras import datasets, layers, models
```

```
[2]: (train_images, train_labels), (test_images, test_labels) = datasets.cifar10.
      ↪load_data()
```

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>
170498071/170498071 32s
0us/step

```
[3]: train_images = train_images / 255.0
      test_images = test_images / 255.0
```

```
[4]: model = models.Sequential()

      model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
      model.add(layers.MaxPooling2D((2, 2)))

      model.add(layers.Conv2D(64, (3, 3), activation='relu'))
      model.add(layers.MaxPooling2D((2, 2)))

      model.add(layers.Conv2D(64, (3, 3), activation='relu'))

      model.add(layers.Flatten())

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

D:\anaconda3\envs\dp_env\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

```
    super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
[5]: model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
      ↪metrics=['accuracy'])
```

```
[6]: history = model.fit(train_images, train_labels, epochs=10,  
    ↪ validation_data=(test_images, test_labels))
```

```
Epoch 1/10  
1563/1563          12s 7ms/step -  
accuracy: 0.3288 - loss: 1.7996 - val_accuracy: 0.5259 - val_loss: 1.2858  
Epoch 2/10  
1563/1563          10s 7ms/step -  
accuracy: 0.5584 - loss: 1.2331 - val_accuracy: 0.5730 - val_loss: 1.2130  
Epoch 3/10  
1563/1563          10s 7ms/step -  
accuracy: 0.6306 - loss: 1.0497 - val_accuracy: 0.6475 - val_loss: 1.0036  
Epoch 4/10  
1563/1563          12s 7ms/step -  
accuracy: 0.6689 - loss: 0.9441 - val_accuracy: 0.6681 - val_loss: 0.9484  
Epoch 5/10  
1563/1563          14s 9ms/step -  
accuracy: 0.6970 - loss: 0.8624 - val_accuracy: 0.6633 - val_loss: 0.9751  
Epoch 6/10  
1563/1563          14s 9ms/step -  
accuracy: 0.7186 - loss: 0.8014 - val_accuracy: 0.6914 - val_loss: 0.9031  
Epoch 7/10  
1563/1563          14s 9ms/step -  
accuracy: 0.7343 - loss: 0.7562 - val_accuracy: 0.6861 - val_loss: 0.9164  
Epoch 8/10  
1563/1563          13s 8ms/step -  
accuracy: 0.7466 - loss: 0.7215 - val_accuracy: 0.6965 - val_loss: 0.9170  
Epoch 9/10  
1563/1563          13s 8ms/step -  
accuracy: 0.7643 - loss: 0.6730 - val_accuracy: 0.7100 - val_loss: 0.8612  
Epoch 10/10  
1563/1563          15s 10ms/step -  
accuracy: 0.7748 - loss: 0.6473 - val_accuracy: 0.7011 - val_loss: 0.8816
```

```
[7]: test_loss, test_acc = model.evaluate(test_images, test_labels)  
    print(f"Test accuracy: {test_acc}")
```

```
313/313           1s 2ms/step -  
accuracy: 0.7100 - loss: 0.8602  
Test accuracy: 0.7010999917984009
```

```
[ ]:
```