assignment06-2a_muley_tushar

January 8, 2022

Name: Tushar Muley

Assignment: Assignment 6-2a

Date:January 9, 2022

Assignment 6.2a Using section 5.2 in Deep Learning with Python as a guide, create a ConvNet model that classifies images CIFAR10 small images classification dataset. Do not use dropout or data-augmentation in this part.

[1]:
```
# get data
from keras.datasets import cifar10
from keras.utils import to_categorical
```

[2]:
```
# library
import pandas as pd
from keras import layers
from keras import models
```

[3]:
```
# breakout data
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 14s 0us/step
```

[4]:
```
# check the training data
x_train.shape, y_train.shape
```

[4]:
```
((50000, 32, 32, 3), (50000, 1))
```

[5]:
```
# check the test data
x_test.shape, y_test.shape
```

[5]:
```
((10000, 32, 32, 3), (10000, 1))
```

[6]:
```
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.MaxPooling2D(2,2))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(10, activation='sigmoid'))
```

[7]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 64)          36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 2, 2, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 512)               131584
_________________________________________________________________
dense_1 (Dense)              (None, 10)                5130
=================================================================
Total params: 193,034
Trainable params: 193,034
Non-trainable params: 0
_________________________________________________________________
```

[8]:
```
model.compile(optimizer='rmsprop',
loss='categorical_crossentropy',
metrics=['accuracy'])
```

[9]:
```
# preprocess the data
x_train = x_train.astype("float32") / 255
x_test = x_test.astype("float32") / 255
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)

# reserve 10K for validation
x_val = x_train[-10000:]
y_val = y_train[-10000:]
```

```python
x_train = x_train[:-10000]
y_train = y_train[:-10000]
```

[10]:
```python
# check sample
x_val.shape, y_val.shape
```

[10]: ((10000, 32, 32, 3), (10000, 10))

[11]:
```python
history = model.fit(x_train, y_train, epochs=100, validation_data=(x_val,y_val))
```

```
Epoch 1/100
1250/1250 [==============================] - 45s 33ms/step - loss: 1.8359 -
accuracy: 0.3269 - val_loss: 1.4621 - val_accuracy: 0.4850
Epoch 2/100
1250/1250 [==============================] - 43s 34ms/step - loss: 1.2246 -
accuracy: 0.5634 - val_loss: 1.0588 - val_accuracy: 0.6308
Epoch 3/100
1250/1250 [==============================] - 40s 32ms/step - loss: 1.0348 -
accuracy: 0.6326 - val_loss: 1.0182 - val_accuracy: 0.6515
Epoch 4/100
1250/1250 [==============================] - 45s 36ms/step - loss: 0.9155 -
accuracy: 0.6751 - val_loss: 1.0605 - val_accuracy: 0.6403
Epoch 5/100
1250/1250 [==============================] - 28s 22ms/step - loss: 0.8219 -
accuracy: 0.7108 - val_loss: 0.9920 - val_accuracy: 0.6637
Epoch 6/100
1250/1250 [==============================] - 30s 24ms/step - loss: 0.7485 -
accuracy: 0.7379 - val_loss: 1.1147 - val_accuracy: 0.6302
Epoch 7/100
1250/1250 [==============================] - 37s 30ms/step - loss: 0.6865 -
accuracy: 0.7629 - val_loss: 1.2606 - val_accuracy: 0.6178
Epoch 8/100
1250/1250 [==============================] - 31s 25ms/step - loss: 0.6568 -
accuracy: 0.7755 - val_loss: 0.9155 - val_accuracy: 0.7007
Epoch 9/100
1250/1250 [==============================] - 29s 23ms/step - loss: 0.6263 -
accuracy: 0.7859 - val_loss: 1.0432 - val_accuracy: 0.6926
Epoch 10/100
1250/1250 [==============================] - 29s 23ms/step - loss: 0.6255 -
accuracy: 0.7854 - val_loss: 1.0788 - val_accuracy: 0.6995
Epoch 11/100
1250/1250 [==============================] - 28s 23ms/step - loss: 0.6033 -
accuracy: 0.7927 - val_loss: 1.0889 - val_accuracy: 0.6526
Epoch 12/100
1250/1250 [==============================] - 29s 23ms/step - loss: 0.6105 -
accuracy: 0.7911 - val_loss: 1.1144 - val_accuracy: 0.6942
Epoch 13/100
1250/1250 [==============================] - 32s 25ms/step - loss: 0.5880 -
```

```
accuracy: 0.8033 - val_loss: 1.1963 - val_accuracy: 0.6676
Epoch 14/100
1250/1250 [==============================] - 30s 24ms/step - loss: 0.6048 -
accuracy: 0.7946 - val_loss: 1.1257 - val_accuracy: 0.6785
Epoch 15/100
1250/1250 [==============================] - 31s 25ms/step - loss: 0.5957 -
accuracy: 0.8022 - val_loss: 1.1461 - val_accuracy: 0.6815
Epoch 16/100
1250/1250 [==============================] - 36s 28ms/step - loss: 0.5900 -
accuracy: 0.8020 - val_loss: 1.2578 - val_accuracy: 0.6584
Epoch 17/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.5980 -
accuracy: 0.7992 - val_loss: 1.2337 - val_accuracy: 0.6879
Epoch 18/100
1250/1250 [==============================] - 36s 29ms/step - loss: 0.5820 -
accuracy: 0.8052 - val_loss: 1.2124 - val_accuracy: 0.6905
Epoch 19/100
1250/1250 [==============================] - 29s 23ms/step - loss: 0.5783 -
accuracy: 0.8104 - val_loss: 1.2467 - val_accuracy: 0.6834
Epoch 20/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.5973 -
accuracy: 0.8028 - val_loss: 1.3658 - val_accuracy: 0.6705
Epoch 21/100
1250/1250 [==============================] - 29s 23ms/step - loss: 0.5939 -
accuracy: 0.8062 - val_loss: 1.2275 - val_accuracy: 0.6880
Epoch 22/100
1250/1250 [==============================] - 32s 26ms/step - loss: 0.6042 -
accuracy: 0.8045 - val_loss: 1.3989 - val_accuracy: 0.6955
Epoch 23/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6111 -
accuracy: 0.8016 - val_loss: 1.3010 - val_accuracy: 0.7018
Epoch 24/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6028 -
accuracy: 0.8052 - val_loss: 1.1473 - val_accuracy: 0.6558
Epoch 25/100
1250/1250 [==============================] - 36s 28ms/step - loss: 0.6097 -
accuracy: 0.8038 - val_loss: 1.6226 - val_accuracy: 0.6797
Epoch 26/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6076 -
accuracy: 0.8048 - val_loss: 1.3226 - val_accuracy: 0.6620
Epoch 27/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.5925 -
accuracy: 0.8108 - val_loss: 1.4596 - val_accuracy: 0.6563
Epoch 28/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6117 -
accuracy: 0.8090 - val_loss: 1.5850 - val_accuracy: 0.6884
Epoch 29/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6254 -
```

```
accuracy: 0.8032 - val_loss: 1.6789 - val_accuracy: 0.6827
Epoch 30/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6130 -
accuracy: 0.8051 - val_loss: 1.2540 - val_accuracy: 0.6500
Epoch 31/100
1250/1250 [==============================] - 37s 30ms/step - loss: 0.6196 -
accuracy: 0.8011 - val_loss: 1.6076 - val_accuracy: 0.6927
Epoch 32/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6343 -
accuracy: 0.8020 - val_loss: 1.2851 - val_accuracy: 0.6581
Epoch 33/100
1250/1250 [==============================] - 37s 30ms/step - loss: 0.6173 -
accuracy: 0.8018 - val_loss: 1.4253 - val_accuracy: 0.6842
Epoch 34/100
1250/1250 [==============================] - 37s 30ms/step - loss: 0.6591 -
accuracy: 0.7905 - val_loss: 1.2248 - val_accuracy: 0.6288
Epoch 35/100
1250/1250 [==============================] - 37s 30ms/step - loss: 0.6345 -
accuracy: 0.8044 - val_loss: 1.4401 - val_accuracy: 0.6584
Epoch 36/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6484 -
accuracy: 0.7996 - val_loss: 1.4441 - val_accuracy: 0.6581
Epoch 37/100
1250/1250 [==============================] - 34s 27ms/step - loss: 0.6506 -
accuracy: 0.8023 - val_loss: 1.6454 - val_accuracy: 0.6527
Epoch 38/100
1250/1250 [==============================] - 38s 30ms/step - loss: 0.6489 -
accuracy: 0.7993 - val_loss: 1.5955 - val_accuracy: 0.5976
Epoch 39/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6675 -
accuracy: 0.7947 - val_loss: 2.0528 - val_accuracy: 0.6854
Epoch 40/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6773 -
accuracy: 0.7924 - val_loss: 1.7016 - val_accuracy: 0.6676
Epoch 41/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6704 -
accuracy: 0.7948 - val_loss: 1.4472 - val_accuracy: 0.6163
Epoch 42/100
1250/1250 [==============================] - 35s 28ms/step - loss: 0.6953 -
accuracy: 0.7855 - val_loss: 2.3117 - val_accuracy: 0.6828
Epoch 43/100
1250/1250 [==============================] - 33s 27ms/step - loss: 0.6832 -
accuracy: 0.7930 - val_loss: 1.3057 - val_accuracy: 0.6521
Epoch 44/100
1250/1250 [==============================] - 32s 26ms/step - loss: 0.6691 -
accuracy: 0.7942 - val_loss: 1.8548 - val_accuracy: 0.6347
Epoch 45/100
1250/1250 [==============================] - 32s 25ms/step - loss: 0.6860 -
```

```
accuracy: 0.7892 - val_loss: 1.7781 - val_accuracy: 0.6739
Epoch 46/100
1250/1250 [==============================] - 32s 26ms/step - loss: 0.6897 -
accuracy: 0.7904 - val_loss: 1.5058 - val_accuracy: 0.6674
Epoch 47/100
1250/1250 [==============================] - 32s 26ms/step - loss: 0.7068 -
accuracy: 0.7849 - val_loss: 1.7811 - val_accuracy: 0.6577
Epoch 48/100
1250/1250 [==============================] - 33s 26ms/step - loss: 0.7003 -
accuracy: 0.7878 - val_loss: 1.4099 - val_accuracy: 0.6699
Epoch 49/100
1250/1250 [==============================] - 30s 24ms/step - loss: 0.7012 -
accuracy: 0.7868 - val_loss: 1.6122 - val_accuracy: 0.6636
Epoch 50/100
1250/1250 [==============================] - 30s 24ms/step - loss: 0.7121 -
accuracy: 0.7827 - val_loss: 1.2997 - val_accuracy: 0.6009
Epoch 51/100
1250/1250 [==============================] - 27s 22ms/step - loss: 0.7033 -
accuracy: 0.7796 - val_loss: 1.5868 - val_accuracy: 0.6712
Epoch 52/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7234 -
accuracy: 0.7801 - val_loss: 2.0812 - val_accuracy: 0.6593
Epoch 53/100
1250/1250 [==============================] - 26s 21ms/step - loss: 0.7357 -
accuracy: 0.7782 - val_loss: 1.3825 - val_accuracy: 0.6335
Epoch 54/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7190 -
accuracy: 0.7765 - val_loss: 1.7406 - val_accuracy: 0.6805
Epoch 55/100
1250/1250 [==============================] - 26s 21ms/step - loss: 0.7115 -
accuracy: 0.7814 - val_loss: 1.5623 - val_accuracy: 0.6572
Epoch 56/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7293 -
accuracy: 0.7786 - val_loss: 1.4875 - val_accuracy: 0.6389
Epoch 57/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7283 -
accuracy: 0.7759 - val_loss: 1.8414 - val_accuracy: 0.6602
Epoch 58/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7229 -
accuracy: 0.7781 - val_loss: 1.3578 - val_accuracy: 0.6666
Epoch 59/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7084 -
accuracy: 0.7866 - val_loss: 1.7043 - val_accuracy: 0.6717
Epoch 60/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7240 -
accuracy: 0.7799 - val_loss: 1.8645 - val_accuracy: 0.6750
Epoch 61/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7294 -
```

```
accuracy: 0.7777 - val_loss: 1.4217 - val_accuracy: 0.6399
Epoch 62/100
1250/1250 [==============================] - 26s 21ms/step - loss: 0.7086 -
accuracy: 0.7818 - val_loss: 1.7239 - val_accuracy: 0.6453
Epoch 63/100
1250/1250 [==============================] - 29s 23ms/step - loss: 0.7189 -
accuracy: 0.7781 - val_loss: 1.2878 - val_accuracy: 0.6216
Epoch 64/100
1250/1250 [==============================] - 30s 24ms/step - loss: 0.7400 -
accuracy: 0.7767 - val_loss: 1.3993 - val_accuracy: 0.6432
Epoch 65/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7198 -
accuracy: 0.7855 - val_loss: 1.2909 - val_accuracy: 0.6489
Epoch 66/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7096 -
accuracy: 0.7765 - val_loss: 1.2580 - val_accuracy: 0.6741
Epoch 67/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7202 -
accuracy: 0.7767 - val_loss: 1.3813 - val_accuracy: 0.6738
Epoch 68/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7086 -
accuracy: 0.7787 - val_loss: 1.4085 - val_accuracy: 0.6528
Epoch 69/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6995 -
accuracy: 0.7807 - val_loss: 1.5318 - val_accuracy: 0.6491
Epoch 70/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7013 -
accuracy: 0.7826 - val_loss: 2.0221 - val_accuracy: 0.6489
Epoch 71/100
1250/1250 [==============================] - 28s 22ms/step - loss: 0.7239 -
accuracy: 0.7776 - val_loss: 1.3145 - val_accuracy: 0.6558
Epoch 72/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6977 -
accuracy: 0.7835 - val_loss: 1.5881 - val_accuracy: 0.6540
Epoch 73/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6955 -
accuracy: 0.7827 - val_loss: 1.5039 - val_accuracy: 0.6407
Epoch 74/100
1250/1250 [==============================] - 26s 21ms/step - loss: 0.6956 -
accuracy: 0.7821 - val_loss: 1.3095 - val_accuracy: 0.6561
Epoch 75/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6929 -
accuracy: 0.7818 - val_loss: 1.6257 - val_accuracy: 0.5840
Epoch 76/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6991 -
accuracy: 0.7810 - val_loss: 1.2542 - val_accuracy: 0.6420
Epoch 77/100
1250/1250 [==============================] - 24s 20ms/step - loss: 0.6962 -
```

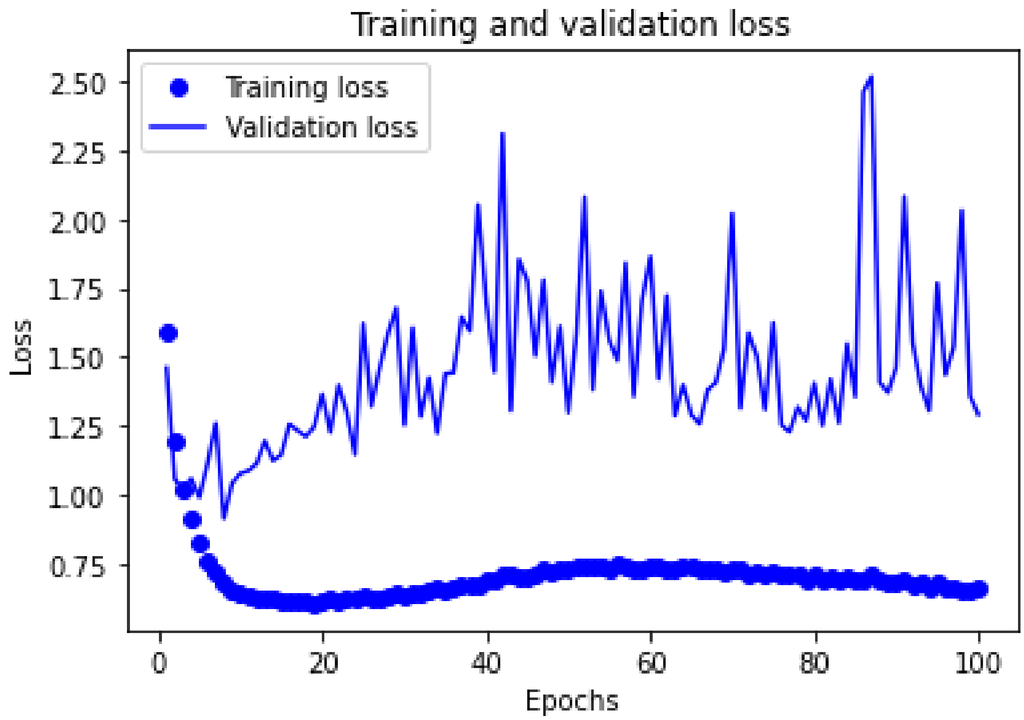
```
accuracy: 0.7809 - val_loss: 1.2289 - val_accuracy: 0.6609
Epoch 78/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.7004 -
accuracy: 0.7830 - val_loss: 1.3193 - val_accuracy: 0.6469
Epoch 79/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6674 -
accuracy: 0.7896 - val_loss: 1.2697 - val_accuracy: 0.6825
Epoch 80/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6884 -
accuracy: 0.7826 - val_loss: 1.4066 - val_accuracy: 0.6833
Epoch 81/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6760 -
accuracy: 0.7857 - val_loss: 1.2521 - val_accuracy: 0.6512
Epoch 82/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6919 -
accuracy: 0.7790 - val_loss: 1.4214 - val_accuracy: 0.6362
Epoch 83/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6849 -
accuracy: 0.7825 - val_loss: 1.2615 - val_accuracy: 0.6520
Epoch 84/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6905 -
accuracy: 0.7844 - val_loss: 1.5496 - val_accuracy: 0.6694
Epoch 85/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6801 -
accuracy: 0.7848 - val_loss: 1.3567 - val_accuracy: 0.6598
Epoch 86/100
1250/1250 [==============================] - 27s 22ms/step - loss: 0.6757 -
accuracy: 0.7864 - val_loss: 2.4631 - val_accuracy: 0.6636
Epoch 87/100
1250/1250 [==============================] - 29s 23ms/step - loss: 0.7057 -
accuracy: 0.7804 - val_loss: 2.5179 - val_accuracy: 0.6797
Epoch 88/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6747 -
accuracy: 0.7866 - val_loss: 1.4089 - val_accuracy: 0.6470
Epoch 89/100
1250/1250 [==============================] - 24s 19ms/step - loss: 0.6834 -
accuracy: 0.7843 - val_loss: 1.3715 - val_accuracy: 0.6728
Epoch 90/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6661 -
accuracy: 0.7875 - val_loss: 1.4632 - val_accuracy: 0.6247
Epoch 91/100
1250/1250 [==============================] - 24s 20ms/step - loss: 0.6779 -
accuracy: 0.7893 - val_loss: 2.0835 - val_accuracy: 0.6677
Epoch 92/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6625 -
accuracy: 0.7908 - val_loss: 1.5524 - val_accuracy: 0.6594
Epoch 93/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6661 -
```

```
accuracy: 0.7937 - val_loss: 1.3946 - val_accuracy: 0.6622
Epoch 94/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6510 -
accuracy: 0.7958 - val_loss: 1.3060 - val_accuracy: 0.6333
Epoch 95/100
1250/1250 [==============================] - 24s 20ms/step - loss: 0.6662 -
accuracy: 0.7895 - val_loss: 1.7688 - val_accuracy: 0.6423
Epoch 96/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6523 -
accuracy: 0.7934 - val_loss: 1.4366 - val_accuracy: 0.6609
Epoch 97/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6445 -
accuracy: 0.7922 - val_loss: 1.5374 - val_accuracy: 0.6284
Epoch 98/100
1250/1250 [==============================] - 24s 20ms/step - loss: 0.6415 -
accuracy: 0.7932 - val_loss: 2.0326 - val_accuracy: 0.6223
Epoch 99/100
1250/1250 [==============================] - 25s 20ms/step - loss: 0.6436 -
accuracy: 0.7908 - val_loss: 1.3589 - val_accuracy: 0.6638
Epoch 100/100
1250/1250 [==============================] - 24s 20ms/step - loss: 0.6616 -
accuracy: 0.7964 - val_loss: 1.2924 - val_accuracy: 0.6511
```

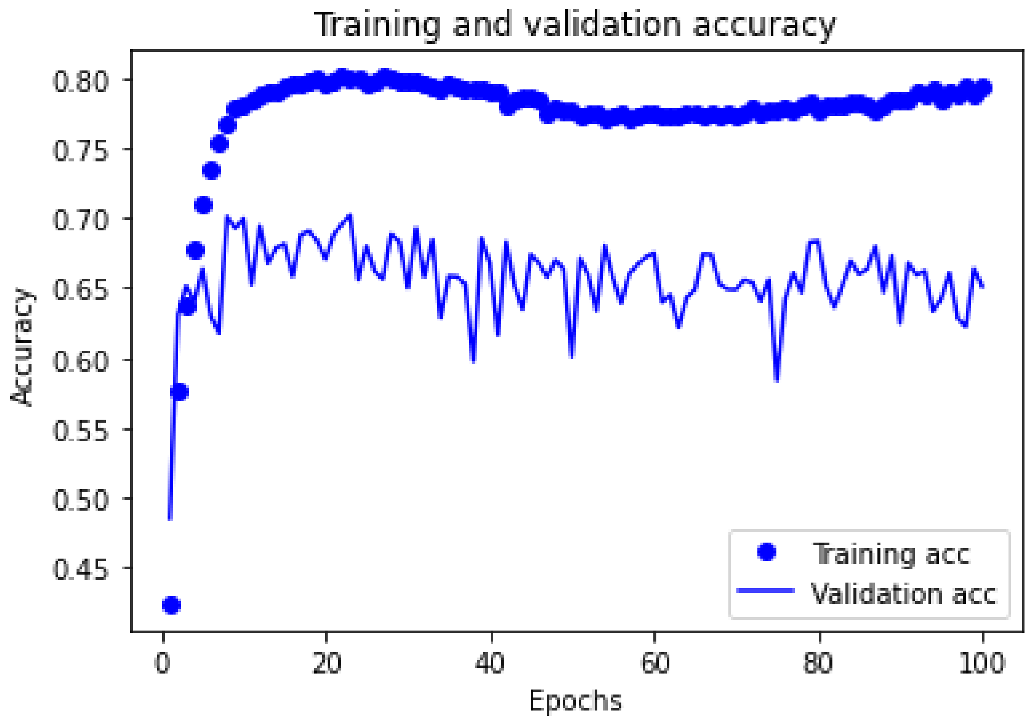
[12]:
```python
history_dict = history.history
history_dict.keys()
```

[12]:
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

[14]:
```python
# plot the training and validation loss
import matplotlib.pyplot as plt
history_dict = history.history
loss_values = history_dict["loss"]
val_loss_values = history_dict["val_loss"]
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, "bo", label="Training loss")
plt.plot(epochs, val_loss_values, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



[15]:
```
# plot the training and validation accuracy
plt.clf()
acc = history_dict["accuracy"]
val_acc = history_dict["val_accuracy"]
plt.plot(epochs, acc, "bo", label="Training acc")
plt.plot(epochs, val_acc, "b", label="Validation acc")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```



[16]:
```
# retrain the model
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# preprocess the data
x_train = x_train.astype("float32") / 255
x_test = x_test.astype("float32") / 255
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
model.compile(optimizer='rmsprop',
loss='categorical_crossentropy',
metrics=['accuracy'])
history = model.fit(x_train, y_train, epochs=10)
results = model.evaluate(x_test, y_test)
```

```
Epoch 1/10
1563/1563 [==============================] - 30s 18ms/step - loss: 0.8621 -
accuracy: 0.7425
Epoch 2/10
1563/1563 [==============================] - 33s 21ms/step - loss: 0.7850 -
accuracy: 0.75360s - loss: 0.7849 - accura
Epoch 3/10
1563/1563 [==============================] - 36s 23ms/step - loss: 0.7789 -
accuracy: 0.7581
Epoch 4/10
```

```
1563/1563 [==============================] - 33s 21ms/step - loss: 0.7755 -
accuracy: 0.7540
Epoch 5/10
1563/1563 [==============================] - 30s 19ms/step - loss: 0.7620 -
accuracy: 0.7608
Epoch 6/10
1563/1563 [==============================] - 31s 20ms/step - loss: 0.7477 -
accuracy: 0.7637
Epoch 7/10
1563/1563 [==============================] - 30s 19ms/step - loss: 0.7574 -
accuracy: 0.7581
Epoch 8/10
1563/1563 [==============================] - 31s 20ms/step - loss: 0.7640 -
accuracy: 0.7569
Epoch 9/10
1563/1563 [==============================] - 32s 21ms/step - loss: 0.7653 -
accuracy: 0.7579
Epoch 10/10
1563/1563 [==============================] - 30s 19ms/step - loss: 0.7346 -
accuracy: 0.7614
313/313 [==============================] - 3s 8ms/step - loss: 1.4350 -
accuracy: 0.6515
```

[17]:
```
# print the results
results
```

[17]:
```
[1.4349656105041504, 0.6514999866485596]
```

[18]:
```
# using a trained model to generate predictions on new data
model.predict(x_test)
```

[18]:
```
array([[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, …, 0.0000000e+00,
        0.0000000e+00, 0.0000000e+00],
       [3.0394677e-31, 1.7475647e-31, 4.7103405e-35, …, 6.9648780e-37,
        2.3962413e-30, 8.8907534e-32],
       [4.8058961e-22, 5.4047130e-22, 2.2406346e-22, …, 2.4171372e-22,
        6.1508722e-21, 6.1663518e-22],
       …,
       [0.0000000e+00, 0.0000000e+00, 0.0000000e+00, …, 0.0000000e+00,
        0.0000000e+00, 0.0000000e+00],
       [5.2398946e-34, 2.3454804e-36, 5.3544291e-32, …, 5.3570209e-33,
        9.0837745e-37, 1.8028205e-36],
       [0.0000000e+00, 0.0000000e+00, 0.0000000e+00, …, 0.0000000e+00,
        0.0000000e+00, 0.0000000e+00]], dtype=float32)
```