assignment06_SafsafiAchraf

May 1, 2021

Achraf Safsafi

DSC650

Assignment 6

Assignment 6.1 :

Instantiating a small convnet

The model definition

[1]:
```python
import warnings
warnings.filterwarnings("ignore")

from keras import layers
from keras import models

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
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'))

model.summary()
```

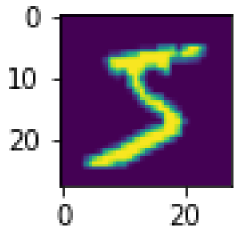
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 32)        320
_________________________________________________________________
```

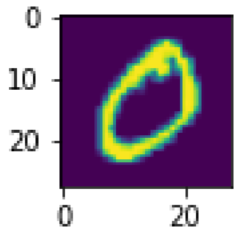
```
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0

_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 64)        18496

_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0

_________________________________________________________________
conv2d_2 (Conv2D)            (None, 3, 3, 64)          36928

_________________________________________________________________
flatten (Flatten)            (None, 576)               0

_________________________________________________________________
dense (Dense)                (None, 64)                36928

_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0

_________________________________________________________________
```

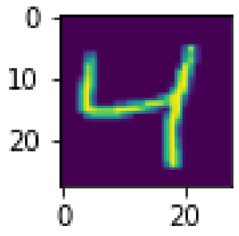
Training the convnet on MNIST images

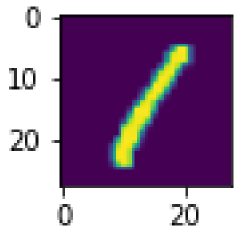
[2]:
```python
from keras.datasets import mnist
from keras.utils import to_categorical

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32') / 255

test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32') / 255

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

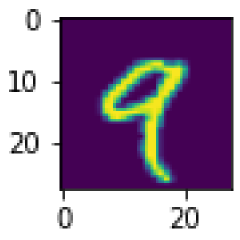
[3]:
```python
from matplotlib import pyplot

for i in range(9):
    pyplot.subplot(330 + 1 + i)
    pyplot.imshow(train_images[i])
    pyplot.show()
```

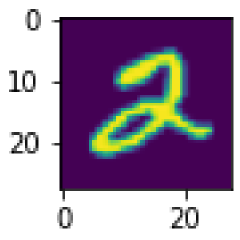

0
10
20
0
20

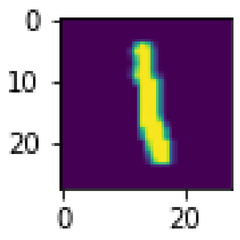

0
10
20
0
20

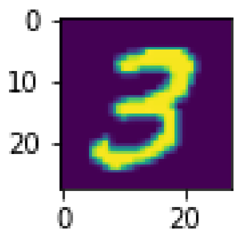

0
10
20
0
20

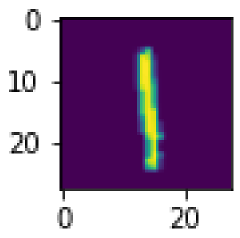

0
10
20
0
20


0
10
20
0
20


0
10
20
0
20


0
10
20
0
20


0
10
20
0
20



[4]:
```
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(train_images, train_labels, epochs=5,␣
 ↪batch_size=64,validation_data=(test_images,test_labels))
```

```
Epoch 1/5
938/938 [==============================] - 16s 16ms/step - loss: 0.3995 -
accuracy: 0.8737 - val_loss: 0.0416 - val_accuracy: 0.9856
Epoch 2/5
938/938 [==============================] - 14s 14ms/step - loss: 0.0538 -
accuracy: 0.9822 - val_loss: 0.0317 - val_accuracy: 0.9893
Epoch 3/5
938/938 [==============================] - 13s 14ms/step - loss: 0.0339 -
accuracy: 0.9893 - val_loss: 0.0320 - val_accuracy: 0.9901
Epoch 4/5
938/938 [==============================] - 13s 14ms/step - loss: 0.0245 -
accuracy: 0.9923 - val_loss: 0.0271 - val_accuracy: 0.9916
Epoch 5/5
938/938 [==============================] - 13s 14ms/step - loss: 0.0208 -
accuracy: 0.9936 - val_loss: 0.0340 - val_accuracy: 0.9905
```

Saving the model

[5]:
```
model.save('my_model_MNIST.h5')
```

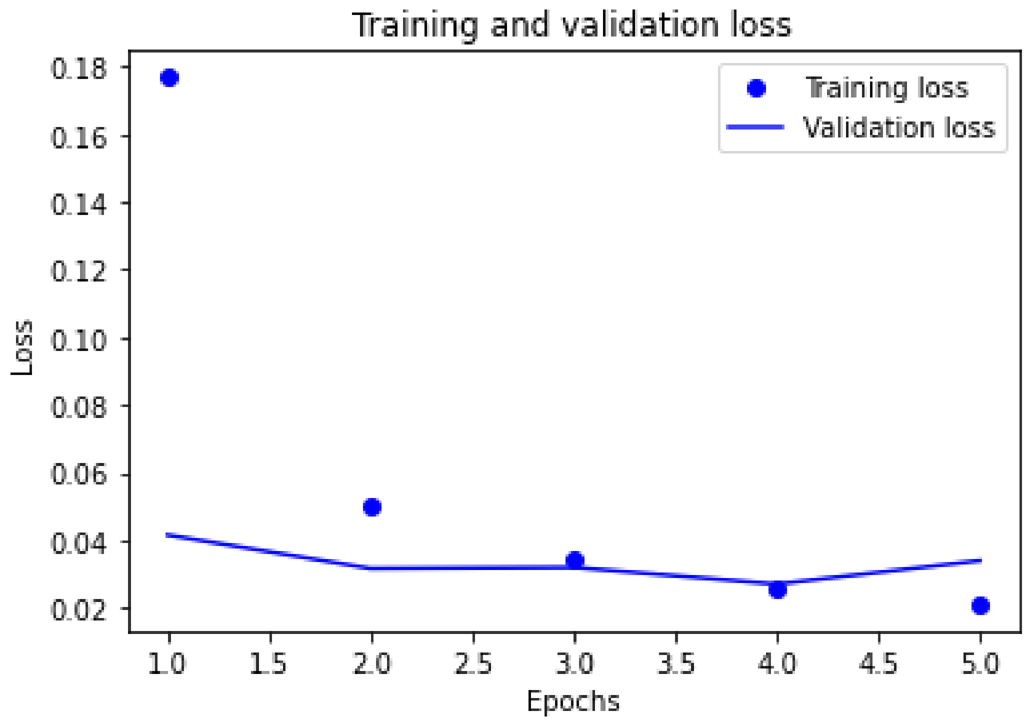
[6]:
```
history_dict = history.history
history_dict.keys()
```

[6]:
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

Plotting the training and validation loss

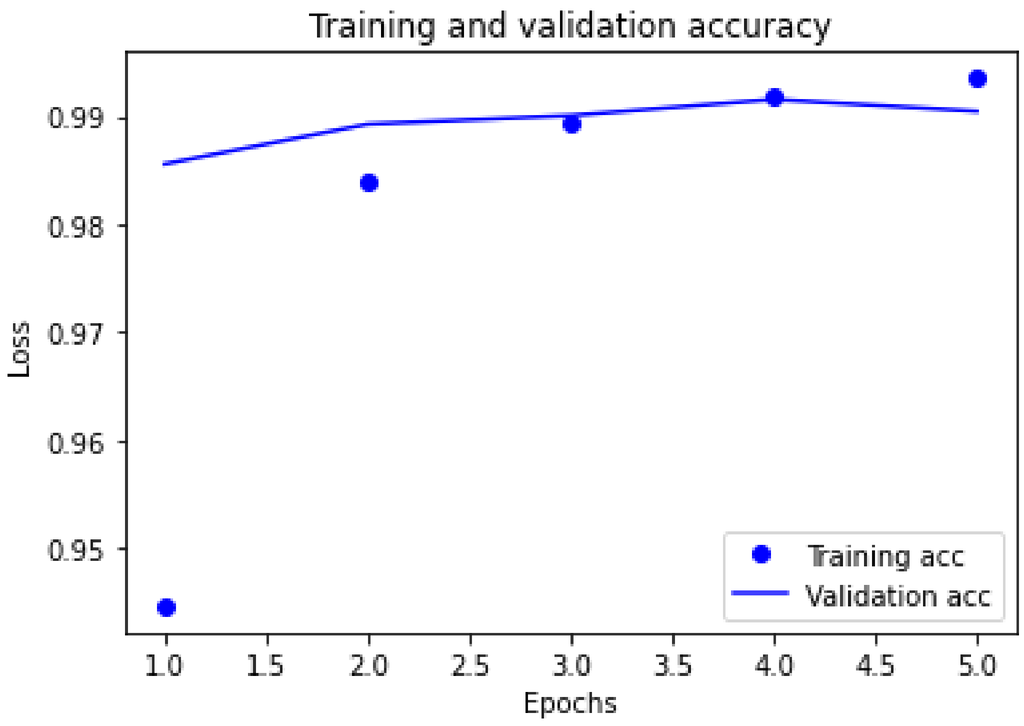
[7]:
```
import matplotlib.pyplot as plt

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
```

```
val_loss = history.history['val_loss']

epochs = range(1, len(loss) + 1)


plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



Plotting the training and validation accuracy

[8]:
```
plt.clf()
acc_values = history_dict['accuracy']
val_acc_values = 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('Loss')
plt.legend()

plt.show()
```



[9]:
```
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 1s 4ms/step - loss: 0.0340 -
accuracy: 0.9905
```

[10]:
```
test_loss
```

[10]:
```
0.034013763070106506
```

[11]:
```
test_acc
```

[11]:
```
0.9904999732971191
```

Using a trained network to generate predictions on new data

[12]:
```
predictions = model.predict(test_images)
predictions
```

[12]:
```
array([[2.57292715e-10, 3.30268479e-09, 1.31545805e-08, …,
         1.00000000e+00, 2.70648925e-09, 2.91721349e-08],
        [1.57501938e-06, 1.32001574e-07, 9.99998331e-01, …,
         1.36899884e-11, 2.23281602e-10, 2.09971878e-13],
        [9.15580056e-09, 9.99992847e-01, 4.73478217e-08, …,
         7.63992603e-07, 2.46768167e-07, 1.12914279e-07],
        …,
        [2.30144696e-15, 1.23220795e-10, 1.18460840e-12, …,
         8.22152589e-12, 2.47006625e-12, 7.60371988e-10],
        [1.12810064e-08, 1.40128664e-09, 2.90594892e-11, …,
         2.57040778e-10, 3.17143895e-05, 2.23177636e-08],
        [4.43960053e-05, 1.67791583e-07, 7.25055997e-08, …,
         1.02242442e-11, 1.80049039e-06, 4.92034246e-07]], dtype=float32)
```

Assignment 6.2 :

Assignment 6.2.A :

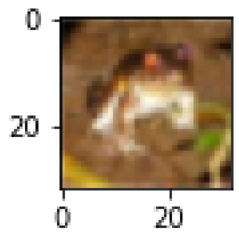
The model definition

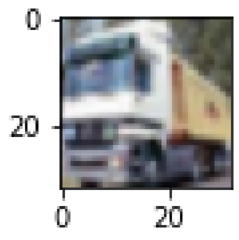
[13]:
```python
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same',␣
 ↪input_shape=(32, 32, 3)))
model.add(layers.Conv2D(32, (3, 3), activation='relu',padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu',padding='same'))
model.add(layers.Conv2D(64, (3, 3), activation='relu',padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu',padding='same'))
model.add(layers.Conv2D(128, (3, 3), activation='relu',padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
```

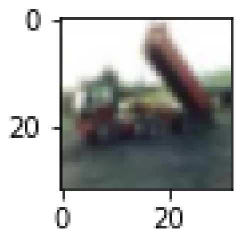
Training the convnet on CIFAR10 images

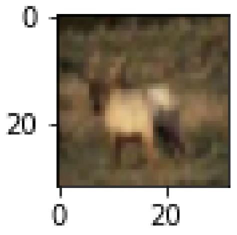
[14]:
```python
from keras.datasets import cifar10
from keras.utils import to_categorical

(train_images, train_labels), (test_images, test_labels) = cifar10.load_data()

train_images = train_images.reshape((50000,32,32, 3))
train_images = train_images.astype('float32') / 255

test_images = test_images.reshape((10000,32,32, 3))
test_images = test_images.astype('float32') / 255

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

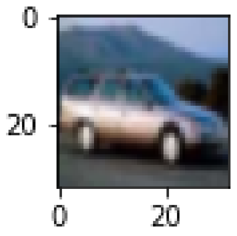
[15]:
```
for i in range(9):
    pyplot.subplot(330 + 1 + i)
    pyplot.imshow(train_images[i])
    pyplot.show()
```



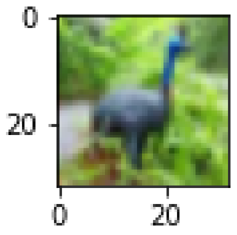


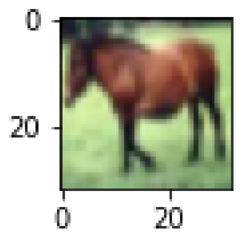


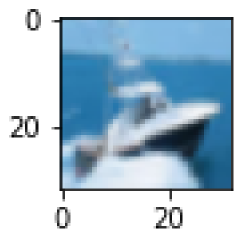

0
20
0
20


0
20
0
20

0
20
0
20


0
20
0
20





[16]:
```
model.compile(loss='categorical_crossentropy',
              optimizer= 'adam',
              metrics=['acc'])
history = model.fit(train_images, train_labels, epochs=100, batch_size=64,
                    validation_data= (test_images,test_labels))
```

```
Epoch 1/100
782/782 [==============================] - 39s 49ms/step - loss: 1.7857 - acc:
0.3325 - val_loss: 1.1783 - val_acc: 0.5765
Epoch 2/100
782/782 [==============================] - 39s 50ms/step - loss: 1.0540 - acc:
0.6242 - val_loss: 0.8873 - val_acc: 0.6885
Epoch 3/100
782/782 [==============================] - 38s 49ms/step - loss: 0.8031 - acc:
0.7192 - val_loss: 0.8202 - val_acc: 0.7216
Epoch 4/100
782/782 [==============================] - 38s 49ms/step - loss: 0.6617 - acc:
0.7679 - val_loss: 0.7497 - val_acc: 0.7409
Epoch 5/100
782/782 [==============================] - 38s 48ms/step - loss: 0.5502 - acc:
0.8065 - val_loss: 0.7326 - val_acc: 0.7527
Epoch 6/100
782/782 [==============================] - 38s 49ms/step - loss: 0.4465 - acc:
0.8409 - val_loss: 0.7185 - val_acc: 0.7641
Epoch 7/100
782/782 [==============================] - 38s 49ms/step - loss: 0.3662 - acc:
0.8708 - val_loss: 0.7162 - val_acc: 0.7785
```

```
Epoch 8/100
782/782 [==============================] - 38s 49ms/step - loss: 0.2908 - acc:
0.8992 - val_loss: 0.7726 - val_acc: 0.7620
Epoch 9/100
782/782 [==============================] - 38s 48ms/step - loss: 0.2460 - acc:
0.9144 - val_loss: 0.8753 - val_acc: 0.7657
Epoch 10/100
782/782 [==============================] - 39s 49ms/step - loss: 0.1979 - acc:
0.9295 - val_loss: 0.9156 - val_acc: 0.7691
Epoch 11/100
782/782 [==============================] - 38s 48ms/step - loss: 0.1625 - acc:
0.9412 - val_loss: 0.9787 - val_acc: 0.7635
Epoch 12/100
782/782 [==============================] - 38s 48ms/step - loss: 0.1292 - acc:
0.9543 - val_loss: 1.0722 - val_acc: 0.7643
Epoch 13/100
782/782 [==============================] - 38s 49ms/step - loss: 0.1290 - acc:
0.9558 - val_loss: 1.0732 - val_acc: 0.7701
Epoch 14/100
782/782 [==============================] - 38s 49ms/step - loss: 0.1188 - acc:
0.9582 - val_loss: 1.1087 - val_acc: 0.7649
Epoch 15/100
782/782 [==============================] - 38s 49ms/step - loss: 0.1107 - acc:
0.9614 - val_loss: 1.2222 - val_acc: 0.7655
Epoch 16/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0942 - acc:
0.9667 - val_loss: 1.3110 - val_acc: 0.7567
Epoch 17/100
782/782 [==============================] - 38s 49ms/step - loss: 0.1115 - acc:
0.9603 - val_loss: 1.2625 - val_acc: 0.7641
Epoch 18/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0857 - acc:
0.9710 - val_loss: 1.3475 - val_acc: 0.7623
Epoch 19/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0844 - acc:
0.9712 - val_loss: 1.3397 - val_acc: 0.7581
Epoch 20/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0909 - acc:
0.9685 - val_loss: 1.4138 - val_acc: 0.7572
Epoch 21/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0869 - acc:
0.9705 - val_loss: 1.3973 - val_acc: 0.7640
Epoch 22/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0831 - acc:
0.9716 - val_loss: 1.5002 - val_acc: 0.7690
Epoch 23/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0777 - acc:
0.9733 - val_loss: 1.5342 - val_acc: 0.7572
```

```
Epoch 24/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0790 - acc:
0.9734 - val_loss: 1.4064 - val_acc: 0.7670
Epoch 25/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0706 - acc:
0.9761 - val_loss: 1.4436 - val_acc: 0.7607
Epoch 26/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0666 - acc:
0.9782 - val_loss: 1.5694 - val_acc: 0.7530
Epoch 27/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0764 - acc:
0.9744 - val_loss: 1.5813 - val_acc: 0.7544
Epoch 28/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0807 - acc:
0.9734 - val_loss: 1.6454 - val_acc: 0.7549
Epoch 29/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0717 - acc:
0.9767 - val_loss: 1.4853 - val_acc: 0.7670
Epoch 30/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0777 - acc:
0.9750 - val_loss: 1.6182 - val_acc: 0.7648
Epoch 31/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0826 - acc:
0.9734 - val_loss: 1.5858 - val_acc: 0.7624
Epoch 32/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0599 - acc:
0.9797 - val_loss: 1.5675 - val_acc: 0.7576
Epoch 33/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0736 - acc:
0.9773 - val_loss: 1.7078 - val_acc: 0.7551
Epoch 34/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0690 - acc:
0.9780 - val_loss: 1.7029 - val_acc: 0.7476
Epoch 35/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0646 - acc:
0.9789 - val_loss: 1.7368 - val_acc: 0.7654
Epoch 36/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0719 - acc:
0.9781 - val_loss: 1.9076 - val_acc: 0.7623
Epoch 37/100
782/782 [==============================] - 37s 48ms/step - loss: 0.0701 - acc:
0.9788 - val_loss: 1.6966 - val_acc: 0.7599
Epoch 38/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0551 - acc:
0.9826 - val_loss: 1.7917 - val_acc: 0.7619
Epoch 39/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0740 - acc:
0.9758 - val_loss: 1.8276 - val_acc: 0.7593
```

```
Epoch 40/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0587 - acc:
0.9814 - val_loss: 1.7716 - val_acc: 0.7625
Epoch 41/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0681 - acc:
0.9792 - val_loss: 1.8606 - val_acc: 0.7590
Epoch 42/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0594 - acc:
0.9802 - val_loss: 1.8274 - val_acc: 0.7647
Epoch 43/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0561 - acc:
0.9832 - val_loss: 1.7834 - val_acc: 0.7494
Epoch 44/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0600 - acc:
0.9820 - val_loss: 1.7569 - val_acc: 0.7567
Epoch 45/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0524 - acc:
0.9834 - val_loss: 1.8909 - val_acc: 0.7606
Epoch 46/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0702 - acc:
0.9778 - val_loss: 1.8227 - val_acc: 0.7567
Epoch 47/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0581 - acc:
0.9818 - val_loss: 1.9713 - val_acc: 0.7531
Epoch 48/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0568 - acc:
0.9826 - val_loss: 1.9178 - val_acc: 0.7636
Epoch 49/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0566 - acc:
0.9829 - val_loss: 1.9116 - val_acc: 0.7594
Epoch 50/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0607 - acc:
0.9805 - val_loss: 1.8983 - val_acc: 0.7637
Epoch 51/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0623 - acc:
0.9801 - val_loss: 2.0249 - val_acc: 0.7560
Epoch 52/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0537 - acc:
0.9823 - val_loss: 1.9352 - val_acc: 0.7566
Epoch 53/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0511 - acc:
0.9853 - val_loss: 2.0432 - val_acc: 0.7581
Epoch 54/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0644 - acc:
0.9813 - val_loss: 1.9835 - val_acc: 0.7660
Epoch 55/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0681 - acc:
0.9797 - val_loss: 1.8886 - val_acc: 0.7633
```

```
Epoch 56/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0547 - acc:
0.9832 - val_loss: 2.0117 - val_acc: 0.7583
Epoch 57/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0517 - acc:
0.9840 - val_loss: 1.8840 - val_acc: 0.7605
Epoch 58/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0525 - acc:
0.9830 - val_loss: 1.9771 - val_acc: 0.7602
Epoch 59/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0512 - acc:
0.9843 - val_loss: 2.2913 - val_acc: 0.7524
Epoch 60/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0609 - acc:
0.9820 - val_loss: 2.1709 - val_acc: 0.7533
Epoch 61/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0596 - acc:
0.9820 - val_loss: 2.0797 - val_acc: 0.7558
Epoch 62/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0625 - acc:
0.9815 - val_loss: 2.1454 - val_acc: 0.7553
Epoch 63/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0596 - acc:
0.9815 - val_loss: 1.9716 - val_acc: 0.7591
Epoch 64/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0476 - acc:
0.9849 - val_loss: 2.0973 - val_acc: 0.7523
Epoch 65/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0592 - acc:
0.9826 - val_loss: 1.9834 - val_acc: 0.7648
Epoch 66/100
782/782 [==============================] - 37s 48ms/step - loss: 0.0509 - acc:
0.9839 - val_loss: 2.3586 - val_acc: 0.7607
Epoch 67/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0579 - acc:
0.9837 - val_loss: 2.1472 - val_acc: 0.7620
Epoch 68/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0469 - acc:
0.9849 - val_loss: 1.9565 - val_acc: 0.7547
Epoch 69/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0546 - acc:
0.9838 - val_loss: 2.1101 - val_acc: 0.7602
Epoch 70/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0522 - acc:
0.9842 - val_loss: 2.2212 - val_acc: 0.7621
Epoch 71/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0632 - acc:
0.9814 - val_loss: 2.3769 - val_acc: 0.7470
```

```
Epoch 72/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0608 - acc:
0.9817 - val_loss: 2.1986 - val_acc: 0.7604
Epoch 73/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0527 - acc:
0.9843 - val_loss: 2.4948 - val_acc: 0.7539
Epoch 74/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0691 - acc:
0.9809 - val_loss: 2.2164 - val_acc: 0.7630
Epoch 75/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0508 - acc:
0.9846 - val_loss: 2.1863 - val_acc: 0.7584
Epoch 76/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0446 - acc:
0.9860 - val_loss: 2.2156 - val_acc: 0.7563
Epoch 77/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0548 - acc:
0.9855 - val_loss: 2.4542 - val_acc: 0.7546
Epoch 78/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0516 - acc:
0.9847 - val_loss: 2.2392 - val_acc: 0.7620
Epoch 79/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0552 - acc:
0.9850 - val_loss: 2.2287 - val_acc: 0.7605
Epoch 80/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0524 - acc:
0.9849 - val_loss: 2.3654 - val_acc: 0.7534
Epoch 81/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0505 - acc:
0.9861 - val_loss: 2.5251 - val_acc: 0.7555
Epoch 82/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0662 - acc:
0.9810 - val_loss: 2.2846 - val_acc: 0.7520
Epoch 83/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0564 - acc:
0.9833 - val_loss: 2.2333 - val_acc: 0.7632
Epoch 84/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0517 - acc:
0.9857 - val_loss: 2.1705 - val_acc: 0.7627
Epoch 85/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0393 - acc:
0.9883 - val_loss: 2.2948 - val_acc: 0.7592
Epoch 86/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0579 - acc:
0.9838 - val_loss: 2.3332 - val_acc: 0.7629
Epoch 87/100
782/782 [==============================] - 39s 50ms/step - loss: 0.0450 - acc:
0.9871 - val_loss: 2.3041 - val_acc: 0.7578
```

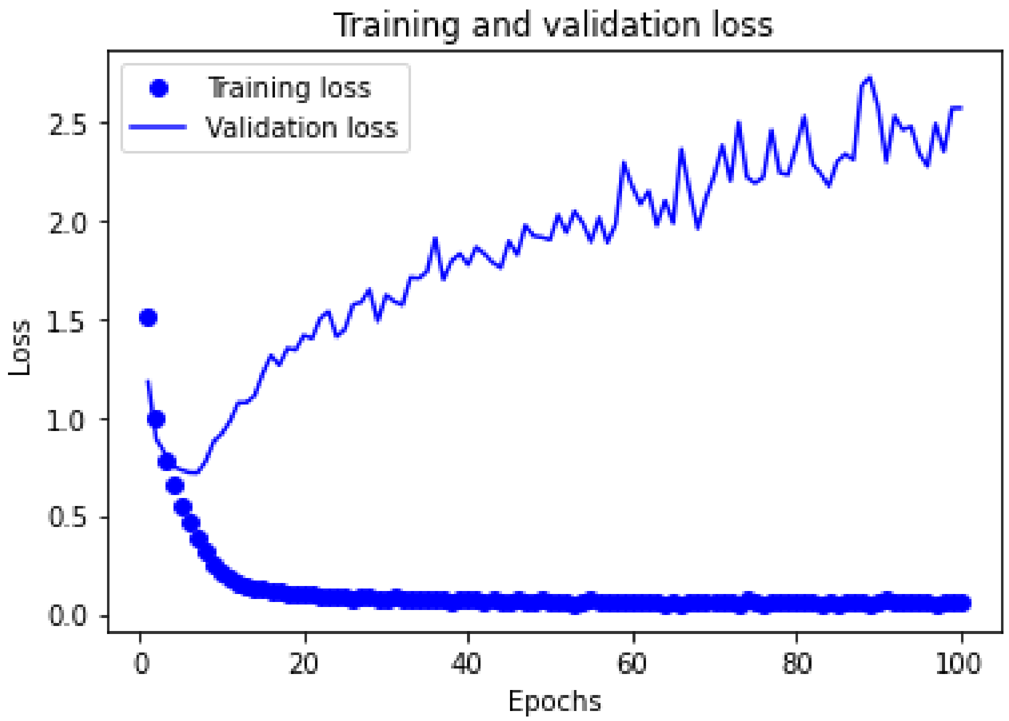
```
Epoch 88/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0545 - acc:
0.9852 - val_loss: 2.6798 - val_acc: 0.7538
Epoch 89/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0632 - acc:
0.9831 - val_loss: 2.7246 - val_acc: 0.7610
Epoch 90/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0587 - acc:
0.9834 - val_loss: 2.5711 - val_acc: 0.7487
Epoch 91/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0725 - acc:
0.9804 - val_loss: 2.2953 - val_acc: 0.7673
Epoch 92/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0598 - acc:
0.9832 - val_loss: 2.5258 - val_acc: 0.7586
Epoch 93/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0456 - acc:
0.9865 - val_loss: 2.4571 - val_acc: 0.7576
Epoch 94/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0515 - acc:
0.9858 - val_loss: 2.4731 - val_acc: 0.7524
Epoch 95/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0481 - acc:
0.9866 - val_loss: 2.3355 - val_acc: 0.7621
Epoch 96/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0567 - acc:
0.9839 - val_loss: 2.2723 - val_acc: 0.7597
Epoch 97/100
782/782 [==============================] - 38s 48ms/step - loss: 0.0429 - acc:
0.9884 - val_loss: 2.4865 - val_acc: 0.7545
Epoch 98/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0607 - acc:
0.9839 - val_loss: 2.3471 - val_acc: 0.7616
Epoch 99/100
782/782 [==============================] - 38s 49ms/step - loss: 0.0447 - acc:
0.9861 - val_loss: 2.5660 - val_acc: 0.7596
Epoch 100/100
782/782 [==============================] - 39s 49ms/step - loss: 0.0551 - acc:
0.9855 - val_loss: 2.5664 - val_acc: 0.7510
```

Saving the model model.save('my_model_CIFAR10_small_1.h5')

Plotting the training and validation loss

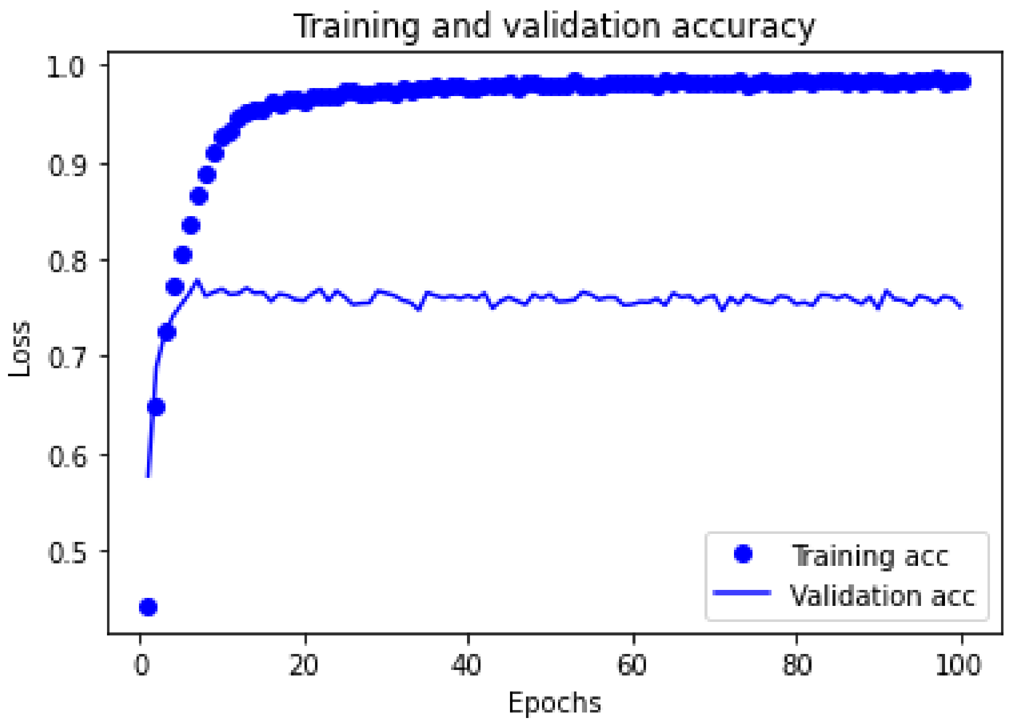
[17]:
```python
import matplotlib.pyplot as plt

acc = history.history['acc']
val_acc = history.history['val_acc']
```

```
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(1, len(loss) + 1)


plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



Plotting the training and validation accuracy

[18]:
```
plt.clf()
acc_values = history_dict['accuracy']
val_acc_values = 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('Loss')
plt.legend()

plt.show()
```



[19]:
```
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 4s 11ms/step - loss: 2.5664 - acc:
0.7510
```

[20]:
```
test_loss
```

[20]: 2.5664284229278564

[21]:
```
test_acc
```

[21]: 0.7509999871253967

Using a trained network to generate predictions on new data

[22]:
```
model.predict(test_images)
```

```
[22]: array([[9.83122349e-31, 0.00000000e+00, 3.76410218e-19, …,
         2.70899174e-25, 1.25453624e-34, 3.51887146e-22],
        [3.51257949e-24, 9.70938974e-09, 0.00000000e+00, …,
         0.00000000e+00, 1.00000000e+00, 4.83174063e-15],
        [2.43231189e-02, 8.85393694e-02, 2.66787225e-07, …,
         2.71037798e-02, 6.86487317e-01, 1.72860682e-01],
        …,
        [2.56901842e-29, 2.14824327e-26, 8.73304204e-16, …,
         2.02863690e-28, 2.73567756e-28, 1.81814868e-33],
        [2.34243433e-26, 1.00000000e+00, 3.03080940e-22, …,
         2.24590593e-22, 1.26092505e-23, 1.98170875e-16],
        [0.00000000e+00, 0.00000000e+00, 4.74353939e-34, …,
         1.00000000e+00, 0.00000000e+00, 0.00000000e+00]], dtype=float32)
```

Assignment 6.2.B :

Defining a new convnet that includes dropout

```
[23]: model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',padding='same',␣
 ↪input_shape=(32, 32, 3)))
model.add(layers.Conv2D(32, (3, 3), activation='relu',padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.2))
model.add(layers.Conv2D(64, (3, 3), activation='relu',padding='same'))
model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.2))
model.add(layers.Conv2D(128, (3, 3), activation='relu',padding='same'))
model.add(layers.Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.2))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(10, activation='softmax'))



model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['acc'])
```

Training the convnet using data-augmentation generators

[24]:
```
from keras.preprocessing.image import ImageDataGenerator

train_datagen =  ImageDataGenerator(
      rotation_range=20,
      width_shift_range=0.1,
      height_shift_range=0.1,
      shear_range=0.1,
      zoom_range=0.1,
      horizontal_flip=True,
      fill_mode='nearest')

train_datagen.fit(train_images)

train_generator = train_datagen.flow(train_images, train_labels,␣
 ↪batch_size=64,shuffle=False)

history = model.fit_generator(train_generator,
                    validation_data=(test_images,test_labels),
                    epochs=100)
```

```
Epoch 1/100
782/782 [==============================] - 52s 66ms/step - loss: 1.9380 - acc:
0.2705 - val_loss: 1.3633 - val_acc: 0.5061
Epoch 2/100
782/782 [==============================] - 51s 66ms/step - loss: 1.4272 - acc:
0.4771 - val_loss: 1.1614 - val_acc: 0.5886
Epoch 3/100
782/782 [==============================] - 52s 66ms/step - loss: 1.2321 - acc:
0.5569 - val_loss: 1.0104 - val_acc: 0.6405
Epoch 4/100
782/782 [==============================] - 51s 66ms/step - loss: 1.1325 - acc:
0.5990 - val_loss: 0.9315 - val_acc: 0.6764
Epoch 5/100
782/782 [==============================] - 51s 66ms/step - loss: 1.0405 - acc:
0.6357 - val_loss: 0.8756 - val_acc: 0.6861
Epoch 6/100
782/782 [==============================] - 51s 66ms/step - loss: 0.9894 - acc:
0.6508 - val_loss: 0.8638 - val_acc: 0.6903
Epoch 7/100
782/782 [==============================] - 52s 66ms/step - loss: 0.9393 - acc:
0.6672 - val_loss: 0.7676 - val_acc: 0.7290
Epoch 8/100
782/782 [==============================] - 52s 66ms/step - loss: 0.8936 - acc:
0.6886 - val_loss: 0.8164 - val_acc: 0.7248
Epoch 9/100
782/782 [==============================] - 52s 66ms/step - loss: 0.8597 - acc:
0.6991 - val_loss: 0.7028 - val_acc: 0.7561
```

```
Epoch 10/100
782/782 [==============================] - 52s 66ms/step - loss: 0.8469 - acc:
0.7054 - val_loss: 0.7763 - val_acc: 0.7343
Epoch 11/100
782/782 [==============================] - 52s 66ms/step - loss: 0.8265 - acc:
0.7121 - val_loss: 0.7457 - val_acc: 0.7453
Epoch 12/100
782/782 [==============================] - 52s 66ms/step - loss: 0.8036 - acc:
0.7227 - val_loss: 0.6858 - val_acc: 0.7695
Epoch 13/100
782/782 [==============================] - 51s 66ms/step - loss: 0.7884 - acc:
0.7236 - val_loss: 0.6946 - val_acc: 0.7676
Epoch 14/100
782/782 [==============================] - 52s 66ms/step - loss: 0.7617 - acc:
0.7377 - val_loss: 0.6467 - val_acc: 0.7817
Epoch 15/100
782/782 [==============================] - 51s 66ms/step - loss: 0.7511 - acc:
0.7393 - val_loss: 0.6596 - val_acc: 0.7766
Epoch 16/100
782/782 [==============================] - 52s 66ms/step - loss: 0.7496 - acc:
0.7448 - val_loss: 0.6597 - val_acc: 0.7749
Epoch 17/100
782/782 [==============================] - 52s 67ms/step - loss: 0.7274 - acc:
0.7491 - val_loss: 0.6570 - val_acc: 0.7776
Epoch 18/100
782/782 [==============================] - 52s 66ms/step - loss: 0.7191 - acc:
0.7496 - val_loss: 0.6795 - val_acc: 0.7761
Epoch 19/100
782/782 [==============================] - 52s 66ms/step - loss: 0.7051 - acc:
0.7573 - val_loss: 0.6685 - val_acc: 0.7799
Epoch 20/100
782/782 [==============================] - 52s 66ms/step - loss: 0.7110 - acc:
0.7562 - val_loss: 0.5796 - val_acc: 0.8039
Epoch 21/100
782/782 [==============================] - 51s 66ms/step - loss: 0.6948 - acc:
0.7588 - val_loss: 0.6085 - val_acc: 0.8050
Epoch 22/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6964 - acc:
0.7617 - val_loss: 0.6481 - val_acc: 0.7880
Epoch 23/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6890 - acc:
0.7622 - val_loss: 0.6627 - val_acc: 0.7806
Epoch 24/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6829 - acc:
0.7663 - val_loss: 0.6550 - val_acc: 0.7894
Epoch 25/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6814 - acc:
0.7635 - val_loss: 0.5907 - val_acc: 0.8015
```

```
Epoch 26/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6654 - acc:
0.7699 - val_loss: 0.6042 - val_acc: 0.8028
Epoch 27/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6621 - acc:
0.7712 - val_loss: 0.6464 - val_acc: 0.7927
Epoch 28/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6757 - acc:
0.7673 - val_loss: 0.6874 - val_acc: 0.7851
Epoch 29/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6619 - acc:
0.7716 - val_loss: 0.5593 - val_acc: 0.8189
Epoch 30/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6502 - acc:
0.7725 - val_loss: 0.5910 - val_acc: 0.8069
Epoch 31/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6495 - acc:
0.7756 - val_loss: 0.6284 - val_acc: 0.7936
Epoch 32/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6461 - acc:
0.7781 - val_loss: 0.6321 - val_acc: 0.7949
Epoch 33/100
782/782 [==============================] - 52s 67ms/step - loss: 0.6291 - acc:
0.7815 - val_loss: 0.6134 - val_acc: 0.8016
Epoch 34/100
782/782 [==============================] - 51s 66ms/step - loss: 0.6295 - acc:
0.7835 - val_loss: 0.6190 - val_acc: 0.8077
Epoch 35/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6290 - acc:
0.7823 - val_loss: 0.5677 - val_acc: 0.8148
Epoch 36/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6356 - acc:
0.7858 - val_loss: 0.5960 - val_acc: 0.8062
Epoch 37/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6180 - acc:
0.7887 - val_loss: 0.6270 - val_acc: 0.8005
Epoch 38/100
782/782 [==============================] - 51s 66ms/step - loss: 0.6316 - acc:
0.7857 - val_loss: 0.5824 - val_acc: 0.8112
Epoch 39/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6235 - acc:
0.7846 - val_loss: 0.5971 - val_acc: 0.8124
Epoch 40/100
782/782 [==============================] - 52s 67ms/step - loss: 0.6234 - acc:
0.7856 - val_loss: 0.5917 - val_acc: 0.8141
Epoch 41/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6275 - acc:
0.7849 - val_loss: 0.5999 - val_acc: 0.8089
```

```
Epoch 42/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6071 - acc:
0.7898 - val_loss: 0.6064 - val_acc: 0.8072
Epoch 43/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6094 - acc:
0.7914 - val_loss: 0.5499 - val_acc: 0.8203
Epoch 44/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6005 - acc:
0.7935 - val_loss: 0.5908 - val_acc: 0.8125
Epoch 45/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6038 - acc:
0.7914 - val_loss: 0.5355 - val_acc: 0.8284
Epoch 46/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5941 - acc:
0.7993 - val_loss: 0.5474 - val_acc: 0.8232
Epoch 47/100
782/782 [==============================] - 52s 66ms/step - loss: 0.6088 - acc:
0.7919 - val_loss: 0.6181 - val_acc: 0.8024
Epoch 48/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5882 - acc:
0.7978 - val_loss: 0.5470 - val_acc: 0.8182
Epoch 49/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5905 - acc:
0.7971 - val_loss: 0.5633 - val_acc: 0.8182
Epoch 50/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5973 - acc:
0.7943 - val_loss: 0.6132 - val_acc: 0.8087
Epoch 51/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5876 - acc:
0.7985 - val_loss: 0.5997 - val_acc: 0.8090
Epoch 52/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5816 - acc:
0.7991 - val_loss: 0.6580 - val_acc: 0.7986
Epoch 53/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5932 - acc:
0.7953 - val_loss: 0.6474 - val_acc: 0.7950
Epoch 54/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5979 - acc:
0.7955 - val_loss: 0.5484 - val_acc: 0.8238
Epoch 55/100
782/782 [==============================] - 52s 67ms/step - loss: 0.5824 - acc:
0.8031 - val_loss: 0.5586 - val_acc: 0.8202
Epoch 56/100
782/782 [==============================] - 52s 67ms/step - loss: 0.5938 - acc:
0.7987 - val_loss: 0.5508 - val_acc: 0.8266
Epoch 57/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5858 - acc:
0.7995 - val_loss: 0.5632 - val_acc: 0.8226
```

```
Epoch 58/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5925 - acc:
0.7993 - val_loss: 0.5552 - val_acc: 0.8238
Epoch 59/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5906 - acc:
0.7976 - val_loss: 0.5525 - val_acc: 0.8234
Epoch 60/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5837 - acc:
0.8010 - val_loss: 0.5556 - val_acc: 0.8243
Epoch 61/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5763 - acc:
0.8039 - val_loss: 0.5721 - val_acc: 0.8202
Epoch 62/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5786 - acc:
0.7990 - val_loss: 0.6523 - val_acc: 0.7966
Epoch 63/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5900 - acc:
0.7994 - val_loss: 0.5353 - val_acc: 0.8290
Epoch 64/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5877 - acc:
0.7992 - val_loss: 0.5443 - val_acc: 0.8249
Epoch 65/100
782/782 [==============================] - 52s 67ms/step - loss: 0.5791 - acc:
0.8010 - val_loss: 0.5714 - val_acc: 0.8226
Epoch 66/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5710 - acc:
0.8036 - val_loss: 0.5423 - val_acc: 0.8292
Epoch 67/100
782/782 [==============================] - 52s 67ms/step - loss: 0.5681 - acc:
0.8046 - val_loss: 0.5454 - val_acc: 0.8260
Epoch 68/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5665 - acc:
0.8053 - val_loss: 0.5648 - val_acc: 0.8193
Epoch 69/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5764 - acc:
0.8023 - val_loss: 0.5392 - val_acc: 0.8316
Epoch 70/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5561 - acc:
0.8111 - val_loss: 0.6852 - val_acc: 0.8025
Epoch 71/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5802 - acc:
0.8018 - val_loss: 0.5434 - val_acc: 0.8352
Epoch 72/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5765 - acc:
0.8026 - val_loss: 0.5458 - val_acc: 0.8249
Epoch 73/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5848 - acc:
0.8011 - val_loss: 0.5669 - val_acc: 0.8271
```

```
Epoch 74/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5672 - acc:
0.8071 - val_loss: 0.5459 - val_acc: 0.8280
Epoch 75/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5673 - acc:
0.8049 - val_loss: 0.5845 - val_acc: 0.8156
Epoch 76/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5630 - acc:
0.8076 - val_loss: 0.5738 - val_acc: 0.8182
Epoch 77/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5718 - acc:
0.8028 - val_loss: 0.6236 - val_acc: 0.8104
Epoch 78/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5710 - acc:
0.8078 - val_loss: 0.5763 - val_acc: 0.8187
Epoch 79/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5707 - acc:
0.8038 - val_loss: 0.5164 - val_acc: 0.8384
Epoch 80/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5691 - acc:
0.8074 - val_loss: 0.5611 - val_acc: 0.8212
Epoch 81/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5697 - acc:
0.8043 - val_loss: 0.5791 - val_acc: 0.8230
Epoch 82/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5744 - acc:
0.8053 - val_loss: 0.5606 - val_acc: 0.8264
Epoch 83/100
782/782 [==============================] - 51s 65ms/step - loss: 0.5723 - acc:
0.8058 - val_loss: 0.5010 - val_acc: 0.8436
Epoch 84/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5551 - acc:
0.8097 - val_loss: 0.5345 - val_acc: 0.8357
Epoch 85/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5656 - acc:
0.8050 - val_loss: 0.5895 - val_acc: 0.8168
Epoch 86/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5594 - acc:
0.8112 - val_loss: 0.5630 - val_acc: 0.8266
Epoch 87/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5507 - acc:
0.8132 - val_loss: 0.5259 - val_acc: 0.8310
Epoch 88/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5650 - acc:
0.8058 - val_loss: 0.5725 - val_acc: 0.8239
Epoch 89/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5627 - acc:
0.8084 - val_loss: 0.5802 - val_acc: 0.8196
```

```
Epoch 90/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5633 - acc:
0.8096 - val_loss: 0.5471 - val_acc: 0.8324
Epoch 91/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5633 - acc:
0.8084 - val_loss: 0.5796 - val_acc: 0.8259
Epoch 92/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5599 - acc:
0.8085 - val_loss: 0.5641 - val_acc: 0.8264
Epoch 93/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5567 - acc:
0.8122 - val_loss: 0.5697 - val_acc: 0.8265
Epoch 94/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5538 - acc:
0.8097 - val_loss: 0.5756 - val_acc: 0.8263
Epoch 95/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5585 - acc:
0.8087 - val_loss: 0.5772 - val_acc: 0.8202
Epoch 96/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5446 - acc:
0.8160 - val_loss: 0.5658 - val_acc: 0.8301
Epoch 97/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5562 - acc:
0.8128 - val_loss: 0.5395 - val_acc: 0.8334
Epoch 98/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5562 - acc:
0.8082 - val_loss: 0.5306 - val_acc: 0.8344
Epoch 99/100
782/782 [==============================] - 51s 66ms/step - loss: 0.5645 - acc:
0.8091 - val_loss: 0.5815 - val_acc: 0.8184
Epoch 100/100
782/782 [==============================] - 52s 66ms/step - loss: 0.5622 - acc:
0.8096 - val_loss: 0.5471 - val_acc: 0.8357
```

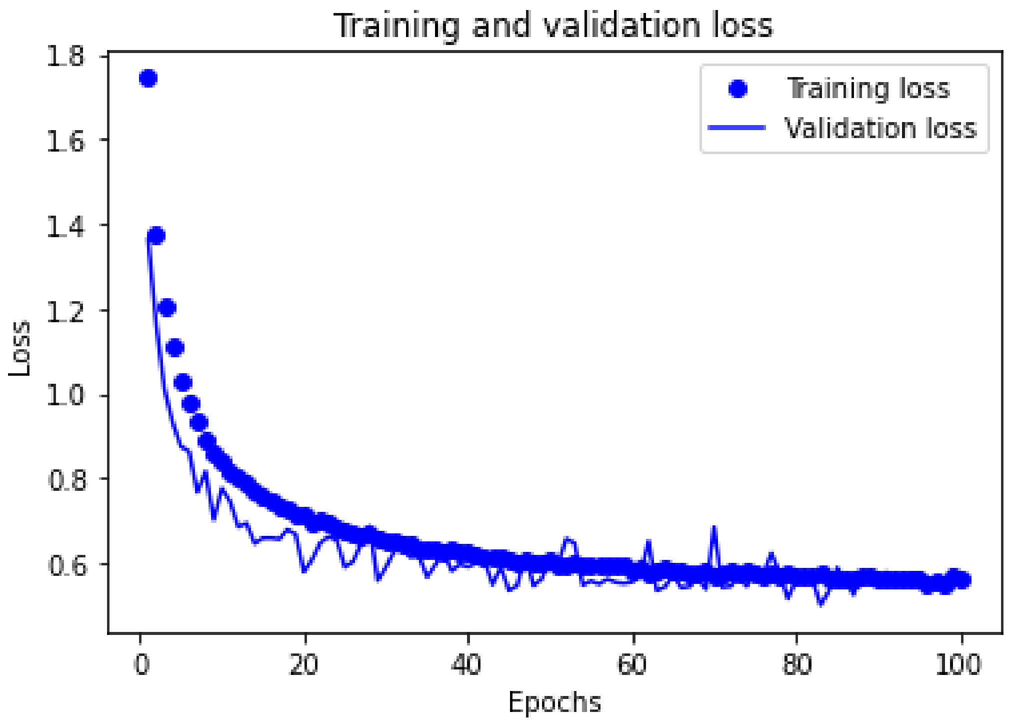
Saving the model

[25]:
```
model.save('my_model_CIFAR10_small_2.h5')
```

Plotting the training and validation loss

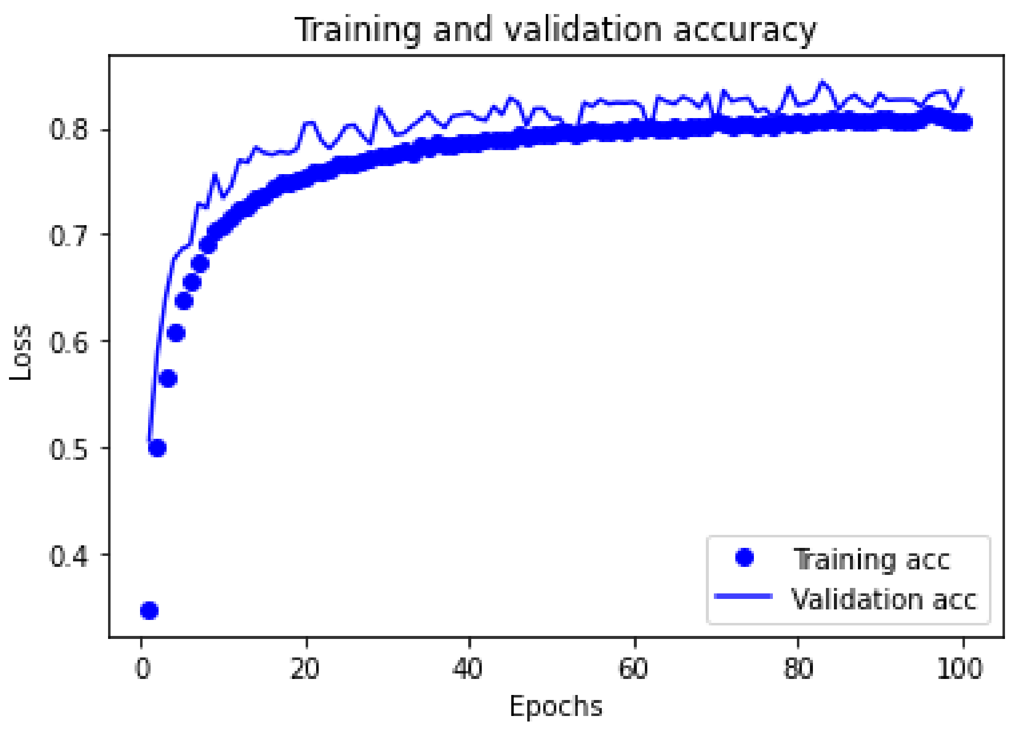
[26]:
```
import matplotlib.pyplot as plt

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(1, len(loss) + 1)
```

```
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



Plotting the training and validation accuracy

[27]:
```
plt.clf()
acc_values = history_dict['accuracy']
val_acc_values = 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('Loss')
plt.legend()
plt.show()
```



[28]:
```
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 3s 11ms/step - loss: 0.5471 - acc:
0.8357
```

[29]:
```
test_loss
```

[29]:
```
0.5471076369285583
```

[30]:
```
test_acc
```

[30]:
```
0.8356999754905701
```

Using a trained network to generate predictions on new data

[31]:
```
predictions = model.predict(test_images)
predictions
```

[31]:
```
array([[6.7784073e-04, 2.0172952e-01, 7.6297566e-04, …, 9.3408256e-05,
        2.6527985e-03, 2.1847911e-02],
       [4.4862385e-04, 8.0999220e-03, 1.1174266e-08, …, 4.3679316e-10,
        9.9143094e-01, 2.0585567e-05],
       [1.4015803e-02, 5.9949052e-01, 1.7706062e-06, …, 8.8142161e-07,
        3.8515681e-01, 1.3236572e-03],
       …,
       [2.4974351e-09, 2.6904849e-09, 6.7537404e-03, …, 3.4844122e-06,
        1.5670787e-07, 1.5309867e-10],
       [9.8884572e-09, 9.9999845e-01, 7.4739893e-12, …, 9.6070970e-13,
        5.2394855e-12, 1.5896888e-06],
       [6.5965121e-19, 2.2851819e-24, 2.7912929e-14, …, 9.9999988e-01,
        1.2999737e-23, 1.1006550e-22]], dtype=float32)
```

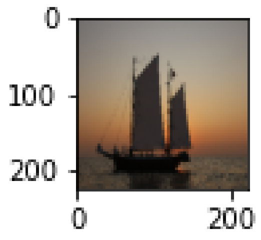
Assignment 6.3 :

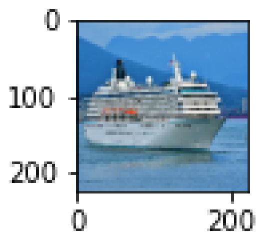
load the dataset

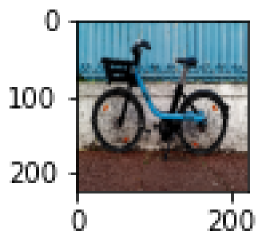
[32]:
```
from tensorflow.keras.preprocessing import image
```

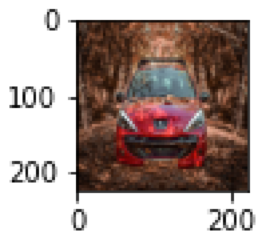
[33]:
```
images_paths = ['anders-thomasson-scZEblckRmM-unsplash.
↪jpg','dan-kb-UdxJSRD6UNM-unsplash.jpg',
               'francois-dallay-nB3GP4y2dlw-unsplash.
↪jpg','pedram-normohamadian-ts26KpllwY0-unsplash.jpg',
               'zetong-li-mVqTumQH-c0-unsplash.jpg']
```

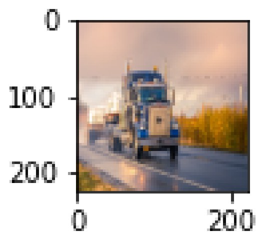
[34]:
```
from matplotlib import pyplot

for i in range(5):
    img = image.load_img(images_paths[i], target_size=(224,224))
    pyplot.subplot(330 + 1 + i)
    pyplot.imshow(img)
    pyplot.show()
```











Make predictions using the ResNet-50

[35]:
```
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input,␣
 ↪decode_predictions
```

```python
import numpy as np

model = ResNet50(weights='imagenet')

x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
print('Predicted:', decode_predictions(preds)[0])
```

```
Predicted: [('n04467665', 'trailer_truck', 0.9343032), ('n03417042',
'garbage_truck', 0.034359016), ('n04252225', 'snowplow', 0.009116761),
('n04461696', 'tow_truck', 0.006997567), ('n04146614', 'school_bus',
0.0062897047)]
```