the IMDB class

In [ ]:

```
#import os
#os.environ['CUDA_VISIBLE_DEVICES'] = '-1'

# Import packages
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.datasets import imdb
#from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

# default values
NUM_WORDS = 10000
BATCH_SIZE = 512

class Imdb:
    def __init__(self, num_words = NUM_WORDS):
        self.x_train, self.y_train, \
        self.x_test, self.y_test = self._load(num_words)
        self.model = self._model()


    def _load(self, num_words):
        (x_train_raw, y_train_raw), (x_test_raw, y_test_raw) \
            = imdb.load_data(num_words=num_words)

        # vectorize reviews
        x_train = self._vectorize_sequences(x_train_raw, num_words)
        y_train = np.asarray(y_train_raw).astype('float32')

        x_test = self._vectorize_sequences(x_test_raw, num_words)
        y_test = np.asarray(y_test_raw).astype('float32')

        #x_train, x_val, y_train, y_val \
        #    = train_test_split(x_train_vec, y_train, test_size=0.2)

        return x_train, y_train, x_test, y_test


    def _model(self):
        model = keras.Sequential([
            layers.Dense(32, activation="relu"),
            layers.Dense(16, activation="relu"),
            layers.Dense(1, activation="sigmoid")
        ])

        # model compilation
        model.compile(optimizer="rmsprop",
                      loss="binary_crossentropy",
                      metrics=["accuracy"])
        return model
```

```
    def plot_loss(self, history):
        # Plotting the training and validation loss
        history_dict = history.history
        loss_values = history_dict["loss"]
        val_loss_values = history_dict["val_loss"]
        epochs = range(1, len(loss_values) + 1)
        plt.figure(1)
        plt.plot(epochs, loss_values, "r", label="Training loss")
        plt.plot(epochs, val_loss_values, "r--", label="Validation loss")
        plt.title("Training and validation loss")
        plt.xlabel("Epochs")
        plt.ylabel("Loss")
        plt.legend()
        plt.show()


    def plot_accuracy(self, history):
        history_dict = history.history
        acc = history_dict["accuracy"]
        val_acc = history_dict["val_accuracy"]
        epochs = range(1, len(acc) + 1)
        plt.figure(2)
        plt.plot(epochs, acc, "b", 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()


    def train(self, epochs=20):
        if self.model is None:
            print('[INFO] model is not defined.')
            return

        history = self.model.fit(self.x_train, self.y_train, \
                                 epochs = epochs, validation_split = 0.2,
                                 batch_size = BATCH_SIZE)
        self.plot_loss(history)
        self.plot_accuracy(history)
        return history


    def evaluate(self):
        score = self.model.evaluate(self.x_test, self.y_test)
        print(f'[INFO] Test loss: {score[0]}')
        print(f'[INFO] Test accuracy: {score[1]}')


    def _vectorize_sequences(self, sequences, dimension):
        results = np.zeros((len(sequences), dimension))
        for i, sequence in enumerate(sequences):
            for j in sequence:
                results[i, j] = 1.
        return results
```

Calling the Imdb class and loading the dataset

In [ ]:
```
imdb = Imdb()
```

```
Metal device set to: Apple M1 Pro

systemMemory: 16.00 GB
maxCacheSize: 5.33 GB

2022-11-14 22:51:41.939096: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:306] Could not identify NUMA node of platform GPU ID 0, defaulting to 0. Your kernel may not have been built with NUMA support.
2022-11-14 22:51:41.939210: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:272] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 0 MB memory) -> physical PluggableDevice (device: 0, name: METAL, pci bus id: <undefined>)
```

Traing the model

In [ ]:
```
history = imdb.train(epochs=100)
```

```
Epoch 1/100
2022-11-14 22:52:02.372578: W tensorflow/core/platform/profile_utils/cpu_utils.cc:128] Failed to get CPU frequency: 0 Hz
2022-11-14 22:52:02.667980: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
40/40 [==============================] - 2s 28ms/step - loss: 0.4691 - accuracy: 0.8031 - val_loss: 0.3330 - val_accuracy: 0.8842
Epoch 2/100
 1/40 [..............................] - ETA: 0s - loss: 0.2740 - accuracy: 0.9199
2022-11-14 22:52:03.909305: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
40/40 [==============================] - 1s 14ms/step - loss: 0.2628 - accuracy: 0.9100 - val_loss: 0.4011 - val_accuracy: 0.8324
Epoch 3/100
40/40 [==============================] - 1s 13ms/step - loss: 0.1992 - accuracy: 0.9288 - val_loss: 0.2736 - val_accuracy: 0.8906
Epoch 4/100
40/40 [==============================] - 1s 13ms/step - loss: 0.1636 - accuracy: 0.9419 - val_loss: 0.2803 - val_accuracy: 0.8920
Epoch 5/100
40/40 [==============================] - 1s 13ms/step - loss: 0.1307 - accuracy: 0.9556 - val_loss: 0.3165 - val_accuracy: 0.8808
Epoch 6/100
40/40 [==============================] - 1s 14ms/step - loss: 0.1096 - accuracy: 0.9631 - val_loss: 0.3237 - val_accuracy: 0.8852
Epoch 7/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0885 - accuracy: 0.9725 - val_loss: 0.3560 - val_accuracy: 0.8788
Epoch 8/100
40/40 [==============================] - 1s 14ms/step - loss: 0.0756 - a
```

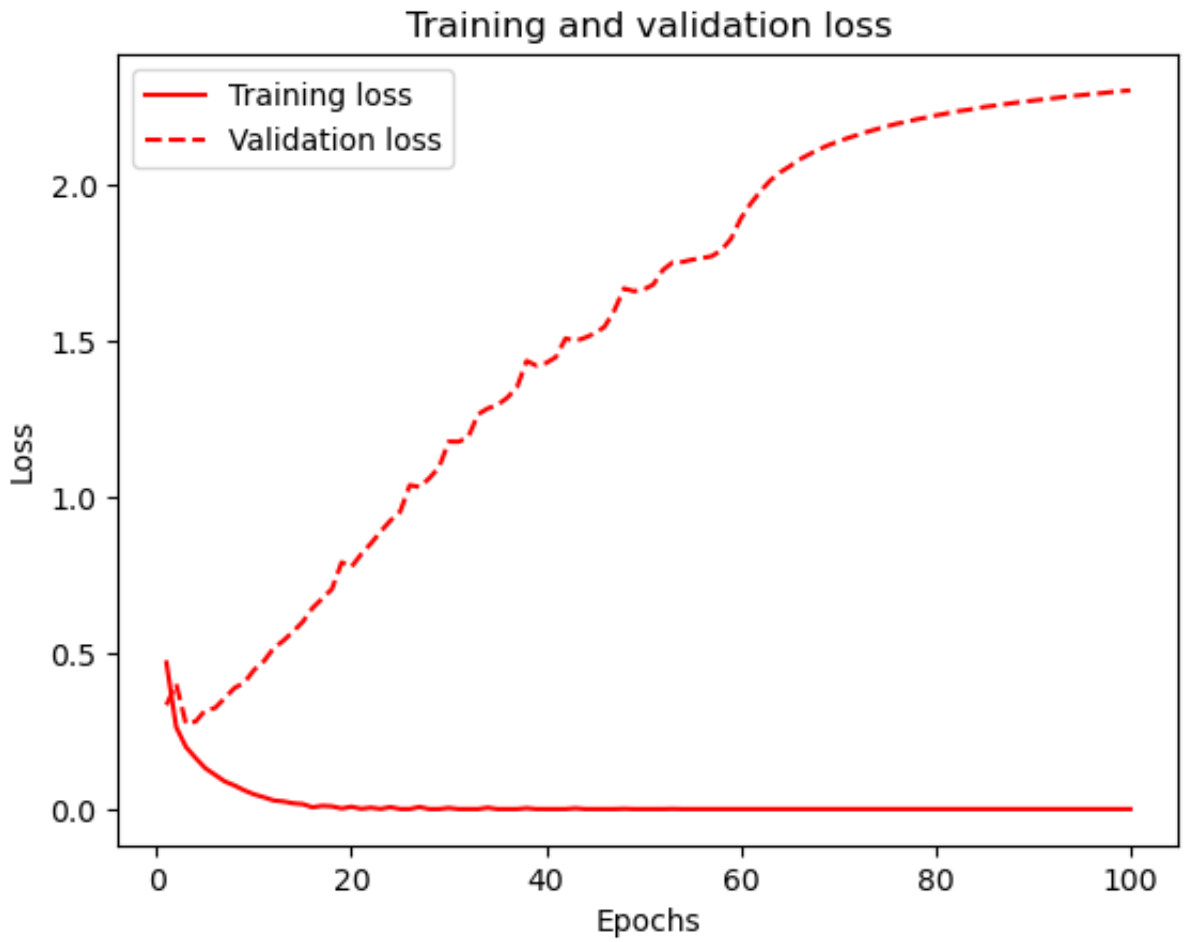
```
ccuracy: 0.9768 - val_loss: 0.3880 - val_accuracy: 0.8748
Epoch 9/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0600 - a
ccuracy: 0.9818 - val_loss: 0.4049 - val_accuracy: 0.8812
Epoch 10/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0471 - a
ccuracy: 0.9865 - val_loss: 0.4456 - val_accuracy: 0.8802
Epoch 11/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0380 - a
ccuracy: 0.9896 - val_loss: 0.4731 - val_accuracy: 0.8724
Epoch 12/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0278 - a
ccuracy: 0.9929 - val_loss: 0.5147 - val_accuracy: 0.8712
Epoch 13/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0245 - a
ccuracy: 0.9934 - val_loss: 0.5391 - val_accuracy: 0.8738
Epoch 14/100
40/40 [==============================] - 0s 13ms/step - loss: 0.0186 - a
ccuracy: 0.9951 - val_loss: 0.5678 - val_accuracy: 0.8726
Epoch 15/100
40/40 [==============================] - 0s 12ms/step - loss: 0.0160 - a
ccuracy: 0.9963 - val_loss: 0.5996 - val_accuracy: 0.8714
Epoch 16/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0058 - a
ccuracy: 0.9996 - val_loss: 0.6435 - val_accuracy: 0.8698
Epoch 17/100
40/40 [==============================] - 1s 14ms/step - loss: 0.0106 - a
ccuracy: 0.9973 - val_loss: 0.6740 - val_accuracy: 0.8694
Epoch 18/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0088 - a
ccuracy: 0.9975 - val_loss: 0.7047 - val_accuracy: 0.8692
Epoch 19/100
40/40 [==============================] - 0s 13ms/step - loss: 0.0020 - a
ccuracy: 1.0000 - val_loss: 0.7902 - val_accuracy: 0.8656
Epoch 20/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0076 - a
ccuracy: 0.9981 - val_loss: 0.7754 - val_accuracy: 0.8716
Epoch 21/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0010 - a
ccuracy: 1.0000 - val_loss: 0.8159 - val_accuracy: 0.8702
Epoch 22/100
40/40 [==============================] - 1s 15ms/step - loss: 0.0047 - a
ccuracy: 0.9987 - val_loss: 0.8504 - val_accuracy: 0.8690
Epoch 23/100
40/40 [==============================] - 1s 14ms/step - loss: 5.5643e-04
- accuracy: 1.0000 - val_loss: 0.8883 - val_accuracy: 0.8668
Epoch 24/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0065 - a
ccuracy: 0.9985 - val_loss: 0.9237 - val_accuracy: 0.8666
Epoch 25/100
40/40 [==============================] - 1s 13ms/step - loss: 2.6478e-04
- accuracy: 1.0000 - val_loss: 0.9521 - val_accuracy: 0.8674
Epoch 26/100
40/40 [==============================] - 1s 13ms/step - loss: 2.0017e-04
- accuracy: 1.0000 - val_loss: 1.0373 - val_accuracy: 0.8656
Epoch 27/100
40/40 [==============================] - 0s 12ms/step - loss: 0.0070 - a
```

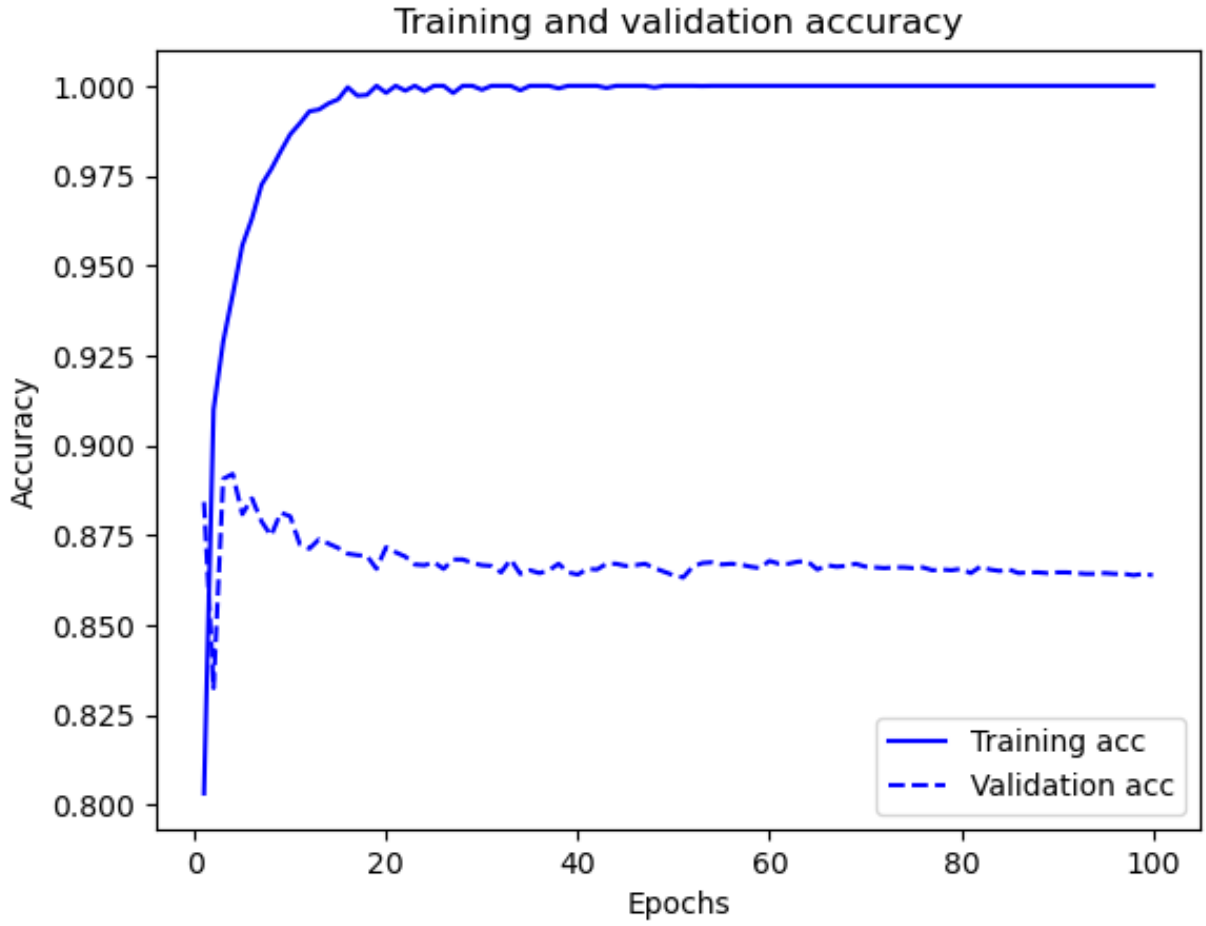
```
ccuracy: 0.9980 - val_loss: 1.0314 - val_accuracy: 0.8682
Epoch 28/100
40/40 [==============================] - 1s 13ms/step - loss: 1.0695e-04
- accuracy: 1.0000 - val_loss: 1.0594 - val_accuracy: 0.8682
Epoch 29/100
40/40 [==============================] - 1s 13ms/step - loss: 7.8676e-05
- accuracy: 1.0000 - val_loss: 1.0946 - val_accuracy: 0.8672
Epoch 30/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0038 - a
ccuracy: 0.9989 - val_loss: 1.1769 - val_accuracy: 0.8666
Epoch 31/100
40/40 [==============================] - 1s 13ms/step - loss: 9.1042e-05
- accuracy: 1.0000 - val_loss: 1.1762 - val_accuracy: 0.8664
Epoch 32/100
40/40 [==============================] - 1s 13ms/step - loss: 4.1568e-05
- accuracy: 1.0000 - val_loss: 1.1910 - val_accuracy: 0.8646
Epoch 33/100
40/40 [==============================] - 1s 13ms/step - loss: 2.6037e-05
- accuracy: 1.0000 - val_loss: 1.2639 - val_accuracy: 0.8682
Epoch 34/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0046 - a
ccuracy: 0.9988 - val_loss: 1.2828 - val_accuracy: 0.8642
Epoch 35/100
40/40 [==============================] - 0s 13ms/step - loss: 1.7938e-05
- accuracy: 1.0000 - val_loss: 1.2940 - val_accuracy: 0.8652
Epoch 36/100
40/40 [==============================] - 1s 13ms/step - loss: 1.1039e-05
- accuracy: 1.0000 - val_loss: 1.3171 - val_accuracy: 0.8644
Epoch 37/100
40/40 [==============================] - 0s 12ms/step - loss: 7.8016e-06
- accuracy: 1.0000 - val_loss: 1.3502 - val_accuracy: 0.8652
Epoch 38/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0030 - a
ccuracy: 0.9993 - val_loss: 1.4351 - val_accuracy: 0.8670
Epoch 39/100
40/40 [==============================] - 1s 13ms/step - loss: 9.6232e-06
- accuracy: 1.0000 - val_loss: 1.4198 - val_accuracy: 0.8646
Epoch 40/100
40/40 [==============================] - 1s 13ms/step - loss: 4.9873e-06
- accuracy: 1.0000 - val_loss: 1.4284 - val_accuracy: 0.8640
Epoch 41/100
40/40 [==============================] - 1s 13ms/step - loss: 3.3318e-06
- accuracy: 1.0000 - val_loss: 1.4469 - val_accuracy: 0.8656
Epoch 42/100
40/40 [==============================] - 1s 13ms/step - loss: 2.2776e-06
- accuracy: 1.0000 - val_loss: 1.5073 - val_accuracy: 0.8654
Epoch 43/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0026 - a
ccuracy: 0.9994 - val_loss: 1.4987 - val_accuracy: 0.8670
Epoch 44/100
40/40 [==============================] - 1s 13ms/step - loss: 1.7544e-06
- accuracy: 1.0000 - val_loss: 1.5094 - val_accuracy: 0.8670
Epoch 45/100
40/40 [==============================] - 1s 13ms/step - loss: 1.3672e-06
- accuracy: 1.0000 - val_loss: 1.5232 - val_accuracy: 0.8664
Epoch 46/100
40/40 [==============================] - 0s 12ms/step - loss: 1.1113e-06
```

```
- accuracy: 1.0000 - val_loss: 1.5443 - val_accuracy: 0.8666
Epoch 47/100
40/40 [==============================] - 1s 13ms/step - loss: 8.3635e-07
- accuracy: 1.0000 - val_loss: 1.5963 - val_accuracy: 0.8670
Epoch 48/100
40/40 [==============================] - 1s 13ms/step - loss: 0.0011 - a
ccuracy: 0.9996 - val_loss: 1.6673 - val_accuracy: 0.8658
Epoch 49/100
40/40 [==============================] - 1s 13ms/step - loss: 1.5762e-06
- accuracy: 1.0000 - val_loss: 1.6582 - val_accuracy: 0.8648
Epoch 50/100
40/40 [==============================] - 1s 13ms/step - loss: 7.9948e-07
- accuracy: 1.0000 - val_loss: 1.6634 - val_accuracy: 0.8638
Epoch 51/100
40/40 [==============================] - 1s 13ms/step - loss: 5.3333e-07
- accuracy: 1.0000 - val_loss: 1.6794 - val_accuracy: 0.8632
Epoch 52/100
40/40 [==============================] - 1s 13ms/step - loss: 3.8262e-07
- accuracy: 1.0000 - val_loss: 1.7265 - val_accuracy: 0.8662
Epoch 53/100
40/40 [==============================] - 1s 13ms/step - loss: 7.2209e-04
- accuracy: 1.0000 - val_loss: 1.7506 - val_accuracy: 0.8672
Epoch 54/100
40/40 [==============================] - 0s 12ms/step - loss: 2.9245e-07
- accuracy: 1.0000 - val_loss: 1.7520 - val_accuracy: 0.8674
Epoch 55/100
40/40 [==============================] - 1s 13ms/step - loss: 2.1573e-07
- accuracy: 1.0000 - val_loss: 1.7591 - val_accuracy: 0.8668
Epoch 56/100
40/40 [==============================] - 1s 13ms/step - loss: 1.8506e-07
- accuracy: 1.0000 - val_loss: 1.7643 - val_accuracy: 0.8670
Epoch 57/100
40/40 [==============================] - 0s 13ms/step - loss: 1.5928e-07
- accuracy: 1.0000 - val_loss: 1.7701 - val_accuracy: 0.8668
Epoch 58/100
40/40 [==============================] - 1s 13ms/step - loss: 1.3408e-07
- accuracy: 1.0000 - val_loss: 1.7893 - val_accuracy: 0.8662
Epoch 59/100
40/40 [==============================] - 1s 13ms/step - loss: 1.0243e-07
- accuracy: 1.0000 - val_loss: 1.8266 - val_accuracy: 0.8658
Epoch 60/100
40/40 [==============================] - 1s 13ms/step - loss: 6.7401e-08
- accuracy: 1.0000 - val_loss: 1.8912 - val_accuracy: 0.8678
Epoch 61/100
40/40 [==============================] - 1s 13ms/step - loss: 4.2054e-08
- accuracy: 1.0000 - val_loss: 1.9381 - val_accuracy: 0.8670
Epoch 62/100
40/40 [==============================] - 0s 13ms/step - loss: 2.8776e-08
- accuracy: 1.0000 - val_loss: 1.9778 - val_accuracy: 0.8670
Epoch 63/100
40/40 [==============================] - 0s 12ms/step - loss: 2.1099e-08
- accuracy: 1.0000 - val_loss: 2.0121 - val_accuracy: 0.8676
Epoch 64/100
40/40 [==============================] - 0s 12ms/step - loss: 1.6241e-08
- accuracy: 1.0000 - val_loss: 2.0383 - val_accuracy: 0.8676
Epoch 65/100
40/40 [==============================] - 0s 12ms/step - loss: 1.3014e-08
```

```
- accuracy: 1.0000 - val_loss: 2.0578 - val_accuracy: 0.8654
Epoch 66/100
40/40 [==============================] - 1s 13ms/step - loss: 1.0523e-08
- accuracy: 1.0000 - val_loss: 2.0796 - val_accuracy: 0.8666
Epoch 67/100
40/40 [==============================] - 1s 13ms/step - loss: 8.9033e-09
- accuracy: 1.0000 - val_loss: 2.0962 - val_accuracy: 0.8662
Epoch 68/100
40/40 [==============================] - 1s 13ms/step - loss: 7.6598e-09
- accuracy: 1.0000 - val_loss: 2.1124 - val_accuracy: 0.8664
Epoch 69/100
40/40 [==============================] - 0s 13ms/step - loss: 6.6310e-09
- accuracy: 1.0000 - val_loss: 2.1259 - val_accuracy: 0.8670
Epoch 70/100
40/40 [==============================] - 0s 12ms/step - loss: 5.6353e-09
- accuracy: 1.0000 - val_loss: 2.1377 - val_accuracy: 0.8662
Epoch 71/100
40/40 [==============================] - 0s 13ms/step - loss: 4.9784e-09
- accuracy: 1.0000 - val_loss: 2.1488 - val_accuracy: 0.8660
Epoch 72/100
40/40 [==============================] - 1s 13ms/step - loss: 4.3422e-09
- accuracy: 1.0000 - val_loss: 2.1585 - val_accuracy: 0.8658
Epoch 73/100
40/40 [==============================] - 1s 13ms/step - loss: 4.0695e-09
- accuracy: 1.0000 - val_loss: 2.1686 - val_accuracy: 0.8660
Epoch 74/100
40/40 [==============================] - 1s 13ms/step - loss: 3.5613e-09
- accuracy: 1.0000 - val_loss: 2.1776 - val_accuracy: 0.8660
Epoch 75/100
40/40 [==============================] - 0s 12ms/step - loss: 3.1606e-09
- accuracy: 1.0000 - val_loss: 2.1863 - val_accuracy: 0.8658
Epoch 76/100
40/40 [==============================] - 0s 12ms/step - loss: 2.8920e-09
- accuracy: 1.0000 - val_loss: 2.1943 - val_accuracy: 0.8660
Epoch 77/100
40/40 [==============================] - 1s 13ms/step - loss: 2.6359e-09
- accuracy: 1.0000 - val_loss: 2.2005 - val_accuracy: 0.8652
Epoch 78/100
40/40 [==============================] - 1s 14ms/step - loss: 2.5326e-09
- accuracy: 1.0000 - val_loss: 2.2082 - val_accuracy: 0.8654
Epoch 79/100
40/40 [==============================] - 0s 13ms/step - loss: 2.2930e-09
- accuracy: 1.0000 - val_loss: 2.2143 - val_accuracy: 0.8652
Epoch 80/100
40/40 [==============================] - 0s 12ms/step - loss: 2.0203e-09
- accuracy: 1.0000 - val_loss: 2.2210 - val_accuracy: 0.8656
Epoch 81/100
40/40 [==============================] - 1s 13ms/step - loss: 1.8757e-09
- accuracy: 1.0000 - val_loss: 2.2266 - val_accuracy: 0.8644
Epoch 82/100
40/40 [==============================] - 0s 12ms/step - loss: 1.7104e-09
- accuracy: 1.0000 - val_loss: 2.2335 - val_accuracy: 0.8662
Epoch 83/100
40/40 [==============================] - 1s 13ms/step - loss: 1.5948e-09
- accuracy: 1.0000 - val_loss: 2.2376 - val_accuracy: 0.8654
Epoch 84/100
40/40 [==============================] - 0s 12ms/step - loss: 1.5121e-09
```

```
- accuracy: 1.0000 - val_loss: 2.2425 - val_accuracy: 0.8650
Epoch 85/100
40/40 [==============================] - 1s 13ms/step - loss: 1.4336e-09
- accuracy: 1.0000 - val_loss: 2.2482 - val_accuracy: 0.8654
Epoch 86/100
40/40 [==============================] - 1s 13ms/step - loss: 1.3634e-09
- accuracy: 1.0000 - val_loss: 2.2522 - val_accuracy: 0.8644
Epoch 87/100
40/40 [==============================] - 0s 12ms/step - loss: 1.2684e-09
- accuracy: 1.0000 - val_loss: 2.2565 - val_accuracy: 0.8646
Epoch 88/100
40/40 [==============================] - 1s 13ms/step - loss: 1.2188e-09
- accuracy: 1.0000 - val_loss: 2.2610 - val_accuracy: 0.8646
Epoch 89/100
40/40 [==============================] - 1s 13ms/step - loss: 1.1486e-09
- accuracy: 1.0000 - val_loss: 2.2646 - val_accuracy: 0.8644
Epoch 90/100
40/40 [==============================] - 1s 13ms/step - loss: 1.0783e-09
- accuracy: 1.0000 - val_loss: 2.2686 - val_accuracy: 0.8646
Epoch 91/100
40/40 [==============================] - 1s 13ms/step - loss: 1.0577e-09
- accuracy: 1.0000 - val_loss: 2.2722 - val_accuracy: 0.8646
Epoch 92/100
40/40 [==============================] - 1s 13ms/step - loss: 1.0287e-09
- accuracy: 1.0000 - val_loss: 2.2759 - val_accuracy: 0.8644
Epoch 93/100
40/40 [==============================] - 1s 13ms/step - loss: 9.9569e-10
- accuracy: 1.0000 - val_loss: 2.2796 - val_accuracy: 0.8642
Epoch 94/100
40/40 [==============================] - 1s 13ms/step - loss: 9.4611e-10
- accuracy: 1.0000 - val_loss: 2.2830 - val_accuracy: 0.8642
Epoch 95/100
40/40 [==============================] - 1s 13ms/step - loss: 9.0893e-10
- accuracy: 1.0000 - val_loss: 2.2862 - val_accuracy: 0.8644
Epoch 96/100
40/40 [==============================] - 0s 13ms/step - loss: 8.9653e-10
- accuracy: 1.0000 - val_loss: 2.2895 - val_accuracy: 0.8642
Epoch 97/100
40/40 [==============================] - 0s 12ms/step - loss: 8.4695e-10
- accuracy: 1.0000 - val_loss: 2.2922 - val_accuracy: 0.8642
Epoch 98/100
40/40 [==============================] - 0s 12ms/step - loss: 7.7672e-10
- accuracy: 1.0000 - val_loss: 2.2957 - val_accuracy: 0.8638
Epoch 99/100
40/40 [==============================] - 1s 13ms/step - loss: 8.0151e-10
- accuracy: 1.0000 - val_loss: 2.2987 - val_accuracy: 0.8642
Epoch 100/100
40/40 [==============================] - 0s 12ms/step - loss: 7.5193e-10
- accuracy: 1.0000 - val_loss: 2.3014 - val_accuracy: 0.8638
```

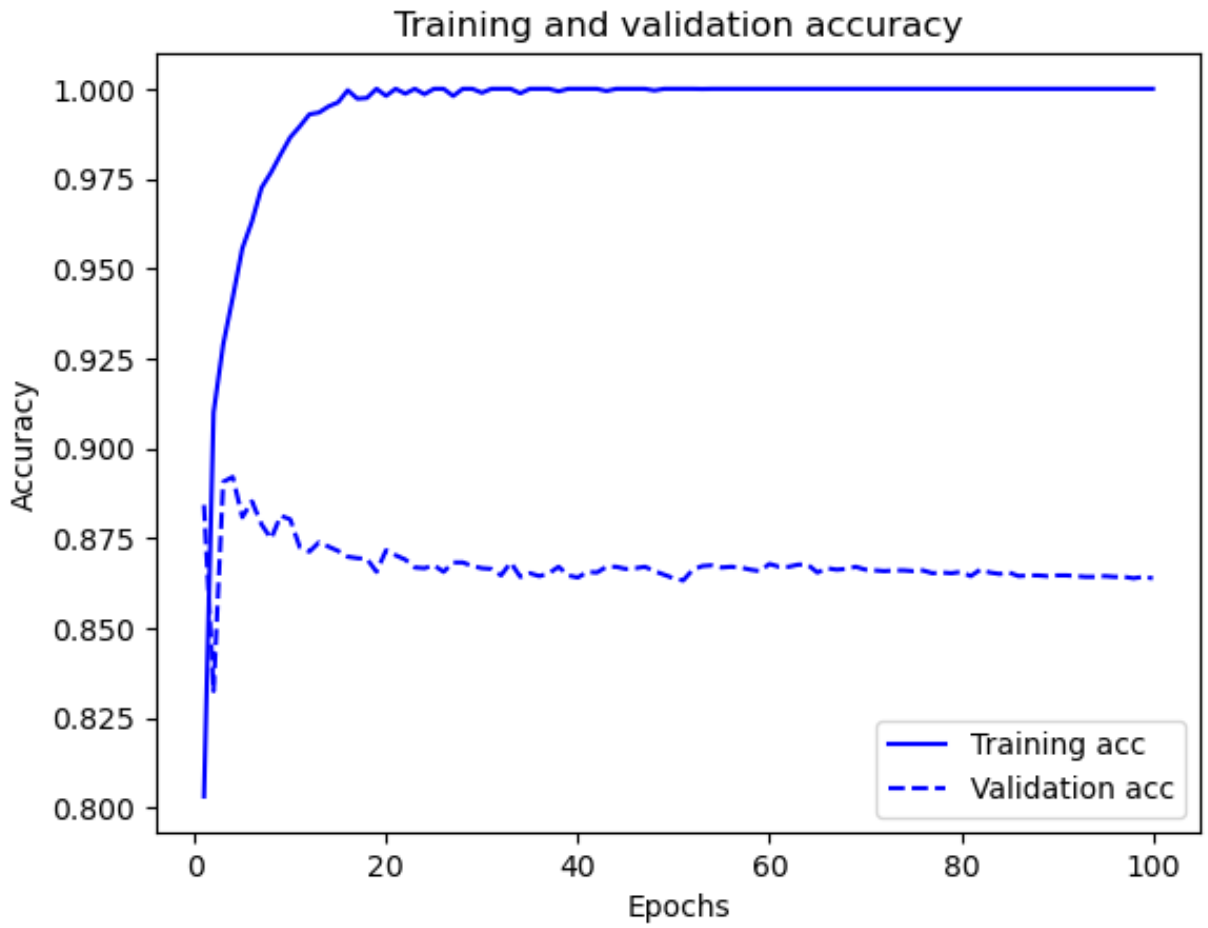




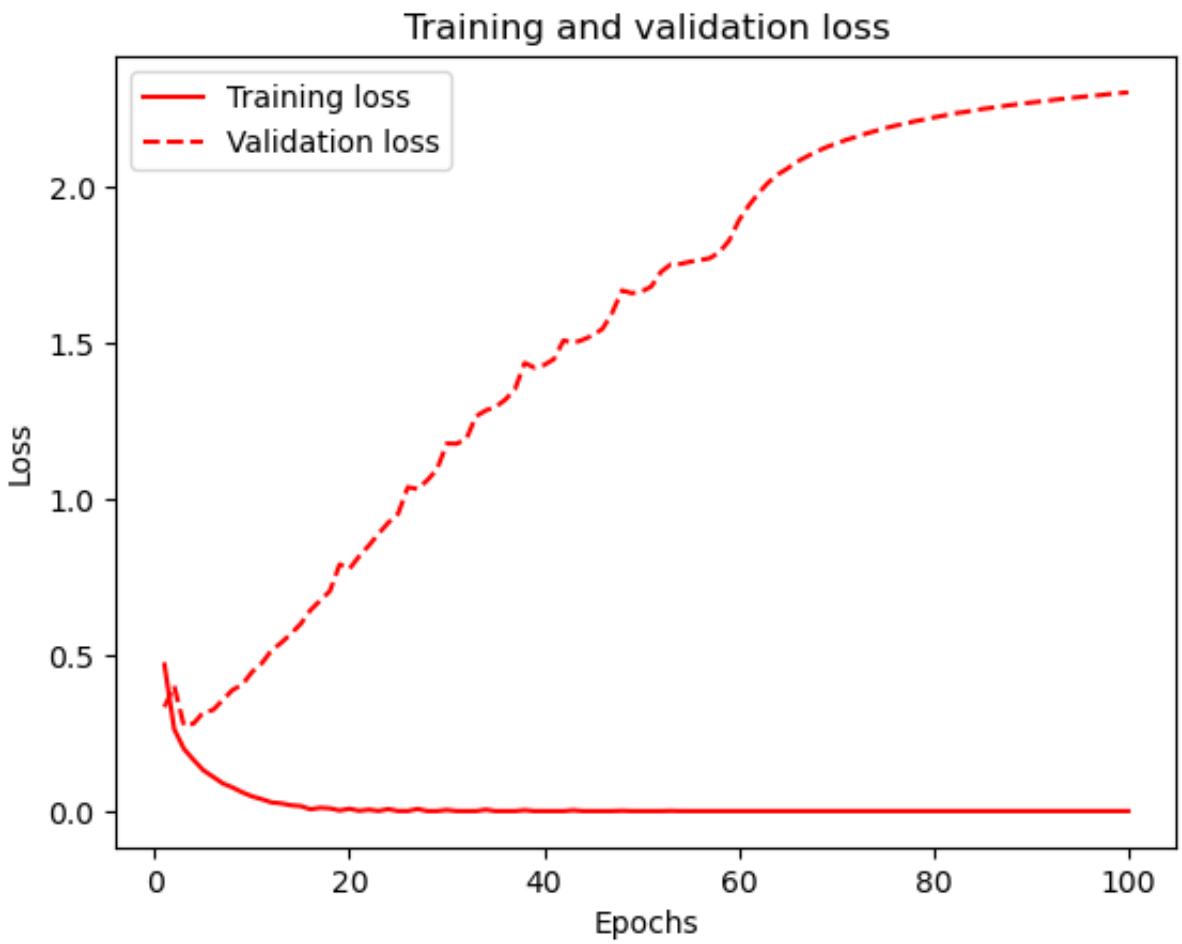
In [ ]:
```
# Evalluting the model
imdb.evaluate()
```

```
782/782 [==============================] - 5s 6ms/step - loss: 2.4979 -
accuracy: 0.8496
[INFO] Test loss: 2.4978740215301514
[INFO] Test accuracy: 0.8495600819587708
```

In [ ]:
```
# Plotting IMDB model accuracy history
imdb.plot_accuracy(history)
```



In [ ]:
```
# Plotiing model loss history
imdb.plot_loss(history)
```



Reuters Class

In [ ]:

```
#import os
#os.environ['CUDA_VISIBLE_DEVICES'] = '-1'

# Import packages
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
#from tensorflow.keras.datasets import reuters
#from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from keras.utils.np_utils import to_categorical

# default values
NUM_WORDS = 10000
BATCH_SIZE = 512

class Reuters:
    def __init__(self, num_words = NUM_WORDS):
        self.x_train, self.y_train, \
        self.x_test, self.y_test = self._load(num_words)
        self.model = self._model()


    def _load(self, num_words):
        (x_train_raw, y_train_raw), (x_test_raw, y_test_raw) = keras.data
```

```
        # vectorize reviews
        x_train = self._vectorize_sequences(x_train_raw, num_words)
        y_train = np.asarray(y_train_raw).astype('float32')

        x_test = self._vectorize_sequences(x_test_raw, num_words)
        y_test = np.asarray(y_test_raw).astype('float32')
        y_train=to_categorical(y_train)
        y_test=to_categorical(y_test)
        #x_train, x_val, y_train, y_val \
        #    = train_test_split(x_train_vec, y_train, test_size=0.2)

        return x_train, y_train, x_test, y_test


    def _model(self):
        model = keras.Sequential([
            layers.Dense(64, activation="relu"),
            layers.Dense(64, activation="relu"),
            layers.Dense(46, activation="softmax")
        ])

        # model compilation
        model.compile(optimizer="rmsprop",
                      loss="categorical_crossentropy",
                      metrics=["accuracy"])
        return model


    def plot_loss(self, history):
        # Plotting the training and validation loss
        history_dict = history.history
        loss_values = history_dict["loss"]
        val_loss_values = history_dict["val_loss"]
        epochs = range(1, len(loss_values) + 1)
        plt.figure(1)
        plt.plot(epochs, loss_values, "r", label="Training loss")
        plt.plot(epochs, val_loss_values, "r--", label="Validation loss")
        plt.title("Training and validation loss")
        plt.xlabel("Epochs")
        plt.ylabel("Loss")
        plt.legend()
        plt.show()


    def plot_accuracy(self, history):
        history_dict = history.history
        acc = history_dict["accuracy"]
        val_acc = history_dict["val_accuracy"]
        epochs = range(1, len(acc) + 1)
        plt.figure(2)
        plt.plot(epochs, acc, "b", 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()
```

```
    def train(self, epochs=20):
        if self.model is None:
            print('[INFO] model is not defined.')
            return

        history = self.model.fit(self.x_train, self.y_train, \
                                 epochs = epochs, validation_split = 0.2,
                                 batch_size = BATCH_SIZE)
        self.plot_loss(history)
        self.plot_accuracy(history)
        return history


    def evaluate(self):
        score = self.model.evaluate(self.x_test, self.y_test)
        print(f'[INFO] Test loss: {score[0]}')
        print(f'[INFO] Test accuracy: {score[1]}')


    def _vectorize_sequences(self, sequences, dimension=10000):
        results = np.zeros((len(sequences), dimension))
        for i, sequence in enumerate(sequences):
            for j in sequence:
                results[i, j] = 1.
        return results
    def to_one_hot(labels, dimension=46):
        results = np.zeros((len(labels), dimension))
        for i,label in enumerate(labels):
            results[i,label] = 1
        return results
```

In [ ]:
```
# Calling Reuters class
reuters = Reuters()
```

In [ ]:
```
# training Reuters model
reuters.train(epochs = 100)
```

```
Epoch 1/100
2022-11-14 22:54:49.838113: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
15/15 [==============================] - 1s 33ms/step - loss: 2.6243 - a
ccuracy: 0.5113 - val_loss: 1.8076 - val_accuracy: 0.6316
Epoch 2/100
 1/15 [=>............................] - ETA: 0s - loss: 1.6648 - accura
cy: 0.6660
2022-11-14 22:54:50.381809: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
15/15 [==============================] - 0s 22ms/step - loss: 1.4732 - a
ccuracy: 0.6928 - val_loss: 1.3861 - val_accuracy: 0.7084
Epoch 3/100
15/15 [==============================] - 0s 17ms/step - loss: 1.1056 - a
ccuracy: 0.7692 - val_loss: 1.2254 - val_accuracy: 0.7407
```

```
Epoch 4/100
15/15 [==============================] - 0s 17ms/step - loss: 0.8738 - a
ccuracy: 0.8173 - val_loss: 1.1613 - val_accuracy: 0.7234
Epoch 5/100
15/15 [==============================] - 0s 19ms/step - loss: 0.7089 - a
ccuracy: 0.8498 - val_loss: 1.1089 - val_accuracy: 0.7401
Epoch 6/100
15/15 [==============================] - 0s 18ms/step - loss: 0.5772 - a
ccuracy: 0.8777 - val_loss: 1.0099 - val_accuracy: 0.7807
Epoch 7/100
15/15 [==============================] - 0s 17ms/step - loss: 0.4665 - a
ccuracy: 0.9040 - val_loss: 0.9789 - val_accuracy: 0.7974
Epoch 8/100
15/15 [==============================] - 0s 18ms/step - loss: 0.3789 - a
ccuracy: 0.9236 - val_loss: 1.0399 - val_accuracy: 0.7846
Epoch 9/100
15/15 [==============================] - 0s 17ms/step - loss: 0.3134 - a
ccuracy: 0.9350 - val_loss: 0.9930 - val_accuracy: 0.7813
Epoch 10/100
15/15 [==============================] - 0s 16ms/step - loss: 0.2612 - a
ccuracy: 0.9442 - val_loss: 1.0420 - val_accuracy: 0.7757
Epoch 11/100
15/15 [==============================] - 0s 18ms/step - loss: 0.2229 - a
ccuracy: 0.9534 - val_loss: 1.0512 - val_accuracy: 0.7735
Epoch 12/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1962 - a
ccuracy: 0.9549 - val_loss: 0.9683 - val_accuracy: 0.7991
Epoch 13/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1680 - a
ccuracy: 0.9591 - val_loss: 1.0314 - val_accuracy: 0.7885
Epoch 14/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1506 - a
ccuracy: 0.9594 - val_loss: 1.0356 - val_accuracy: 0.8036
Epoch 15/100
15/15 [==============================] - 0s 16ms/step - loss: 0.1396 - a
ccuracy: 0.9606 - val_loss: 1.0655 - val_accuracy: 0.7846
Epoch 16/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1255 - a
ccuracy: 0.9633 - val_loss: 1.0840 - val_accuracy: 0.7935
Epoch 17/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1213 - a
ccuracy: 0.9624 - val_loss: 1.1875 - val_accuracy: 0.7819
Epoch 18/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1138 - a
ccuracy: 0.9628 - val_loss: 1.1112 - val_accuracy: 0.8013
Epoch 19/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1099 - a
ccuracy: 0.9623 - val_loss: 1.1089 - val_accuracy: 0.7986
Epoch 20/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1008 - a
ccuracy: 0.9640 - val_loss: 1.1800 - val_accuracy: 0.7830
Epoch 21/100
15/15 [==============================] - 0s 17ms/step - loss: 0.1005 - a
ccuracy: 0.9637 - val_loss: 1.2333 - val_accuracy: 0.7802
Epoch 22/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0975 - a
ccuracy: 0.9635 - val_loss: 1.2402 - val_accuracy: 0.7807
```

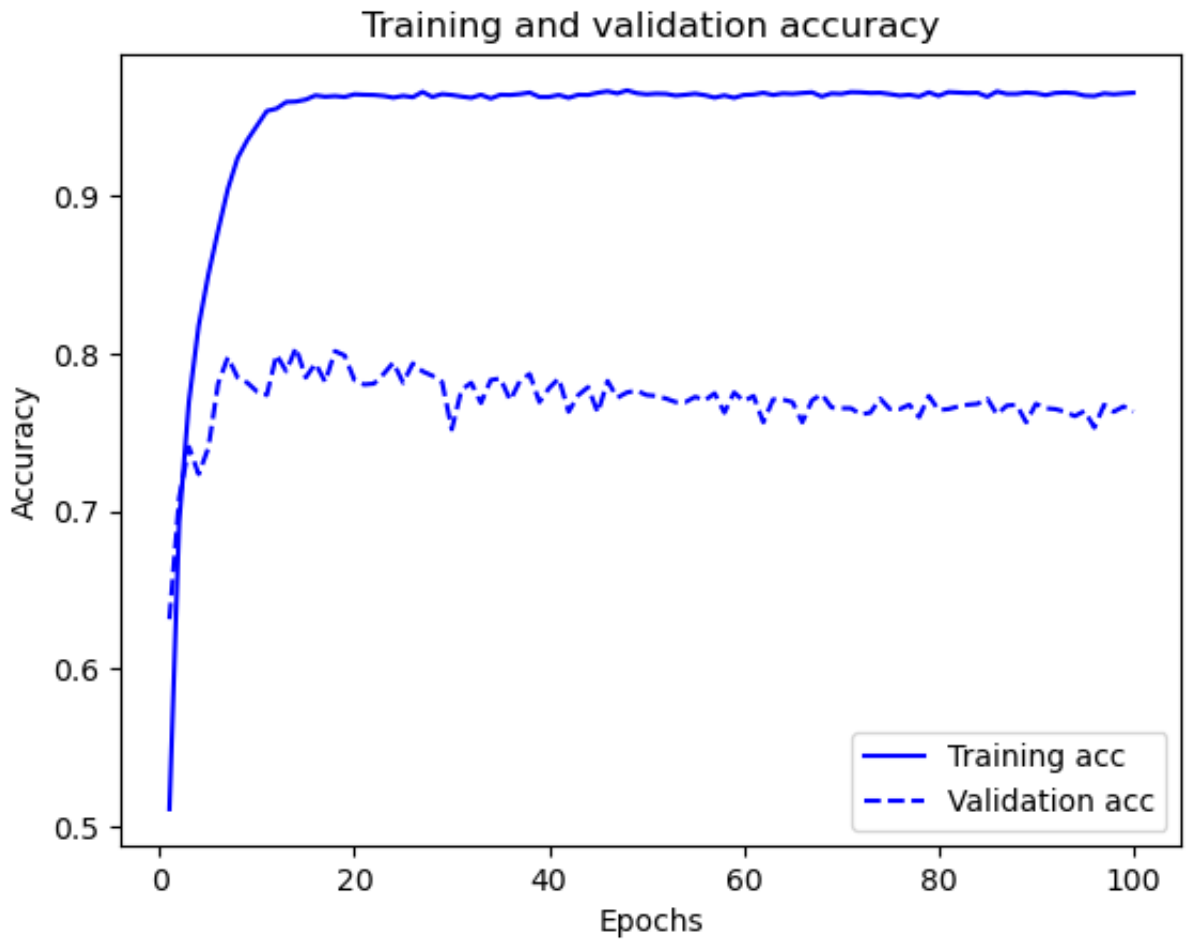
```
Epoch 23/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0961 - a
ccuracy: 0.9630 - val_loss: 1.2104 - val_accuracy: 0.7869
Epoch 24/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0927 - a
ccuracy: 0.9619 - val_loss: 1.2301 - val_accuracy: 0.7941
Epoch 25/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0902 - a
ccuracy: 0.9630 - val_loss: 1.2515 - val_accuracy: 0.7813
Epoch 26/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0930 - a
ccuracy: 0.9621 - val_loss: 1.2706 - val_accuracy: 0.7935
Epoch 27/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0859 - a
ccuracy: 0.9655 - val_loss: 1.3550 - val_accuracy: 0.7885
Epoch 28/100
15/15 [==============================] - 0s 18ms/step - loss: 0.0895 - a
ccuracy: 0.9623 - val_loss: 1.3014 - val_accuracy: 0.7858
Epoch 29/100
15/15 [==============================] - 0s 18ms/step - loss: 0.0894 - a
ccuracy: 0.9641 - val_loss: 1.4132 - val_accuracy: 0.7819
Epoch 30/100
15/15 [==============================] - 0s 19ms/step - loss: 0.0868 - a
ccuracy: 0.9635 - val_loss: 1.5711 - val_accuracy: 0.7518
Epoch 31/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0860 - a
ccuracy: 0.9626 - val_loss: 1.4222 - val_accuracy: 0.7769
Epoch 32/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0846 - a
ccuracy: 0.9617 - val_loss: 1.3394 - val_accuracy: 0.7813
Epoch 33/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0855 - a
ccuracy: 0.9637 - val_loss: 1.5161 - val_accuracy: 0.7685
Epoch 34/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0862 - a
ccuracy: 0.9613 - val_loss: 1.3610 - val_accuracy: 0.7830
Epoch 35/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0793 - a
ccuracy: 0.9637 - val_loss: 1.4450 - val_accuracy: 0.7841
Epoch 36/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0799 - a
ccuracy: 0.9635 - val_loss: 1.5104 - val_accuracy: 0.7702
Epoch 37/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0818 - a
ccuracy: 0.9641 - val_loss: 1.4478 - val_accuracy: 0.7824
Epoch 38/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0791 - a
ccuracy: 0.9651 - val_loss: 1.4280 - val_accuracy: 0.7869
Epoch 39/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0816 - a
ccuracy: 0.9623 - val_loss: 1.4982 - val_accuracy: 0.7691
Epoch 40/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0775 - a
ccuracy: 0.9623 - val_loss: 1.4946 - val_accuracy: 0.7780
Epoch 41/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0787 - a
ccuracy: 0.9635 - val_loss: 1.4620 - val_accuracy: 0.7841
```

```
Epoch 42/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0808 - a
ccuracy: 0.9617 - val_loss: 1.6595 - val_accuracy: 0.7629
Epoch 43/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0796 - a
ccuracy: 0.9637 - val_loss: 1.5817 - val_accuracy: 0.7735
Epoch 44/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0764 - a
ccuracy: 0.9635 - val_loss: 1.4956 - val_accuracy: 0.7785
Epoch 45/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0731 - a
ccuracy: 0.9649 - val_loss: 1.6329 - val_accuracy: 0.7624
Epoch 46/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0756 - a
ccuracy: 0.9659 - val_loss: 1.5085 - val_accuracy: 0.7824
Epoch 47/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0755 - a
ccuracy: 0.9645 - val_loss: 1.6346 - val_accuracy: 0.7718
Epoch 48/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0743 - a
ccuracy: 0.9665 - val_loss: 1.6052 - val_accuracy: 0.7752
Epoch 49/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0763 - a
ccuracy: 0.9646 - val_loss: 1.6118 - val_accuracy: 0.7763
Epoch 50/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0789 - a
ccuracy: 0.9640 - val_loss: 1.6418 - val_accuracy: 0.7735
Epoch 51/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0734 - a
ccuracy: 0.9644 - val_loss: 1.6833 - val_accuracy: 0.7730
Epoch 52/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0731 - a
ccuracy: 0.9642 - val_loss: 1.6105 - val_accuracy: 0.7713
Epoch 53/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0748 - a
ccuracy: 0.9631 - val_loss: 1.7441 - val_accuracy: 0.7691
Epoch 54/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0740 - a
ccuracy: 0.9637 - val_loss: 1.6961 - val_accuracy: 0.7691
Epoch 55/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0724 - a
ccuracy: 0.9644 - val_loss: 1.6884 - val_accuracy: 0.7724
Epoch 56/100
15/15 [==============================] - 0s 18ms/step - loss: 0.0714 - a
ccuracy: 0.9634 - val_loss: 1.7313 - val_accuracy: 0.7702
Epoch 57/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0746 - a
ccuracy: 0.9619 - val_loss: 1.7050 - val_accuracy: 0.7746
Epoch 58/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0714 - a
ccuracy: 0.9633 - val_loss: 1.7204 - val_accuracy: 0.7624
Epoch 59/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0716 - a
ccuracy: 0.9617 - val_loss: 1.6829 - val_accuracy: 0.7752
Epoch 60/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0698 - a
ccuracy: 0.9635 - val_loss: 1.7759 - val_accuracy: 0.7696
```

```
Epoch 61/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0698 - a
ccuracy: 0.9638 - val_loss: 1.6649 - val_accuracy: 0.7730
Epoch 62/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0693 - a
ccuracy: 0.9649 - val_loss: 1.8181 - val_accuracy: 0.7563
Epoch 63/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0700 - a
ccuracy: 0.9635 - val_loss: 1.6959 - val_accuracy: 0.7707
Epoch 64/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0689 - a
ccuracy: 0.9645 - val_loss: 1.7979 - val_accuracy: 0.7707
Epoch 65/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0689 - a
ccuracy: 0.9642 - val_loss: 1.8425 - val_accuracy: 0.7691
Epoch 66/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0654 - a
ccuracy: 0.9648 - val_loss: 1.8229 - val_accuracy: 0.7563
Epoch 67/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0690 - a
ccuracy: 0.9652 - val_loss: 1.7454 - val_accuracy: 0.7702
Epoch 68/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0676 - a
ccuracy: 0.9626 - val_loss: 1.8052 - val_accuracy: 0.7741
Epoch 69/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0656 - a
ccuracy: 0.9645 - val_loss: 1.8582 - val_accuracy: 0.7657
Epoch 70/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0649 - a
ccuracy: 0.9642 - val_loss: 1.8221 - val_accuracy: 0.7652
Epoch 71/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0678 - a
ccuracy: 0.9653 - val_loss: 1.8333 - val_accuracy: 0.7652
Epoch 72/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0651 - a
ccuracy: 0.9652 - val_loss: 1.9107 - val_accuracy: 0.7613
Epoch 73/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0678 - a
ccuracy: 0.9648 - val_loss: 1.9037 - val_accuracy: 0.7624
Epoch 74/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0654 - a
ccuracy: 0.9649 - val_loss: 1.8622 - val_accuracy: 0.7713
Epoch 75/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0661 - a
ccuracy: 0.9641 - val_loss: 1.8870 - val_accuracy: 0.7646
Epoch 76/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0672 - a
ccuracy: 0.9631 - val_loss: 1.9351 - val_accuracy: 0.7646
Epoch 77/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0659 - a
ccuracy: 0.9637 - val_loss: 1.9130 - val_accuracy: 0.7674
Epoch 78/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0635 - a
ccuracy: 0.9626 - val_loss: 1.9478 - val_accuracy: 0.7596
Epoch 79/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0618 - a
ccuracy: 0.9652 - val_loss: 1.8416 - val_accuracy: 0.7730
```

```
Epoch 80/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0643 - a
ccuracy: 0.9630 - val_loss: 1.9713 - val_accuracy: 0.7641
Epoch 81/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0619 - a
ccuracy: 0.9653 - val_loss: 1.9284 - val_accuracy: 0.7646
Epoch 82/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0613 - a
ccuracy: 0.9651 - val_loss: 2.0020 - val_accuracy: 0.7663
Epoch 83/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0626 - a
ccuracy: 0.9648 - val_loss: 1.9761 - val_accuracy: 0.7674
Epoch 84/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0620 - a
ccuracy: 0.9649 - val_loss: 2.0416 - val_accuracy: 0.7679
Epoch 85/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0620 - a
ccuracy: 0.9624 - val_loss: 2.0183 - val_accuracy: 0.7713
Epoch 86/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0621 - a
ccuracy: 0.9658 - val_loss: 2.0991 - val_accuracy: 0.7613
Epoch 87/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0618 - a
ccuracy: 0.9642 - val_loss: 1.9240 - val_accuracy: 0.7668
Epoch 88/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0608 - a
ccuracy: 0.9642 - val_loss: 2.0874 - val_accuracy: 0.7674
Epoch 89/100
15/15 [==============================] - 0s 18ms/step - loss: 0.0599 - a
ccuracy: 0.9651 - val_loss: 2.1437 - val_accuracy: 0.7563
Epoch 90/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0614 - a
ccuracy: 0.9646 - val_loss: 2.0809 - val_accuracy: 0.7679
Epoch 91/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0611 - a
ccuracy: 0.9634 - val_loss: 2.0892 - val_accuracy: 0.7652
Epoch 92/100
15/15 [==============================] - 0s 18ms/step - loss: 0.0598 - a
ccuracy: 0.9648 - val_loss: 2.1635 - val_accuracy: 0.7646
Epoch 93/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0591 - a
ccuracy: 0.9651 - val_loss: 2.1794 - val_accuracy: 0.7629
Epoch 94/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0581 - a
ccuracy: 0.9646 - val_loss: 2.1834 - val_accuracy: 0.7602
Epoch 95/100
15/15 [==============================] - 0s 20ms/step - loss: 0.0601 - a
ccuracy: 0.9631 - val_loss: 2.2312 - val_accuracy: 0.7635
Epoch 96/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0603 - a
ccuracy: 0.9628 - val_loss: 2.2918 - val_accuracy: 0.7529
Epoch 97/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0601 - a
ccuracy: 0.9645 - val_loss: 2.1830 - val_accuracy: 0.7674
Epoch 98/100
15/15 [==============================] - 0s 16ms/step - loss: 0.0587 - a
ccuracy: 0.9640 - val_loss: 2.1945 - val_accuracy: 0.7629
```

```
Epoch 99/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0607 - a
ccuracy: 0.9645 - val_loss: 2.1968 - val_accuracy: 0.7663
Epoch 100/100
15/15 [==============================] - 0s 17ms/step - loss: 0.0593 - a
ccuracy: 0.9649 - val_loss: 2.1742 - val_accuracy: 0.7629
```

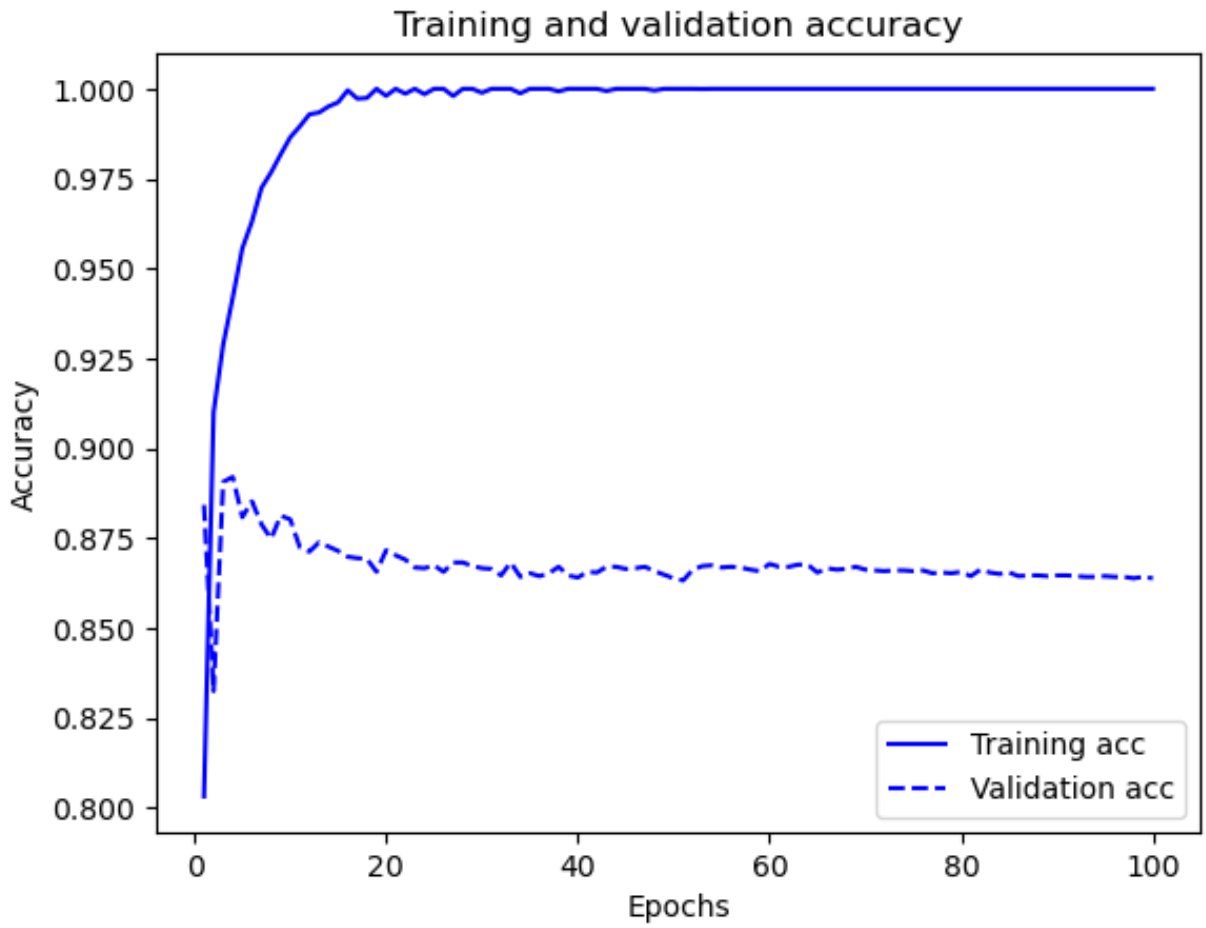


Out[ ]: <keras.callbacks.History at 0x2cb18bfd0>

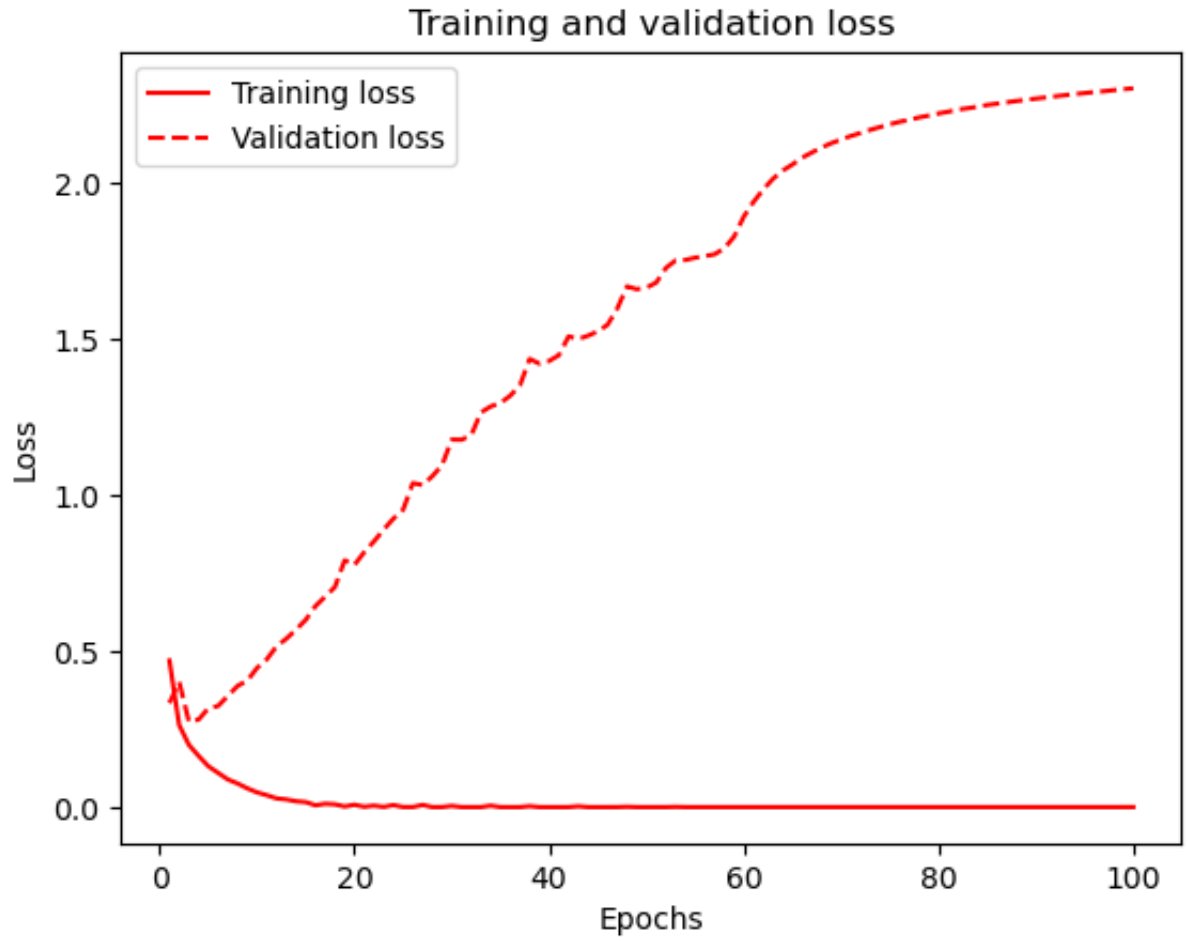
In [ ]:
```
# evaluting the Reuters model
reuters.evaluate()
```

```
71/71 [==============================] - 1s 10ms/step - loss: 2.2141 - a
ccuracy: 0.7542
[INFO] Test loss: 2.214082956314087
[INFO] Test accuracy: 0.7542297840118408
```

In [ ]:
```
# plotting Reuters accuracy
reuters.plot_accuracy(history)
```



In [ ]:
```
# plotting Reuters loss history
reuters.plot_loss(history)
```



the Boston_Housing class

In [ ]:
```
#import os
#os.environ['CUDA_VISIBLE_DEVICES'] = '-1'

# Import packages
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
#from tensorflow.keras.datasets import boston_housing
#from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt


# default values
NUM_WORDS = 10000
BATCH_SIZE = 512

class Boston_Housing:
    def __init__(self):
        self.x_train, self.y_train, \
        self.x_test, self.y_test = self._load()
        self.model = self._model()


    def _load(self):
        (x_train, y_train), (x_test, y_test) = keras.datasets.boston_hous

        # vectorize reviews
        mean=x_train.mean(axis=0)
        x_train-=mean
        std=x_train.std(axis=0)
        x_train/=std
        x_test-=mean
        x_test/=std

        return x_train, y_train, x_test, y_test


    def _model(self):
        model = keras.Sequential([
            layers.Dense(64, activation="relu",input_shape=(self.x_train.
            layers.Dense(64, activation="relu"),
            layers.Dense(1)
        ])

        # model compilation
        model.compile(optimizer="rmsprop",
                      loss="mse",
                      metrics=["mae"])
        return model


    def plot_loss(self, history):
        # Plotting the training and validation loss
        history_dict = history.history
```

```
        loss_values = history_dict["loss"]
        val_loss_values = history_dict["val_loss"]
        epochs = range(1, len(loss_values) + 1)
        plt.figure(1)
        plt.plot(epochs, loss_values, "r", label="Training loss")
        plt.plot(epochs, val_loss_values, "r--", label="Validation loss")
        plt.title("Training and validation loss")
        plt.xlabel("Epochs")
        plt.ylabel("Loss")
        plt.legend()
        plt.show()


    def plot_accuracy(self, history):
        history_dict = history.history
        acc = history_dict["mae"]
        val_acc = history_dict["val_mae"]
        epochs = range(1, len(acc) + 1)
        plt.figure(2)
        plt.plot(epochs, acc, "b", 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()


    def train(self, epochs=20):
        if self.model is None:
            print('[INFO] model is not defined.')
            return

        history = self.model.fit(self.x_train, self.y_train, \
                                 epochs = epochs, validation_split = 0.2,
                                 batch_size = BATCH_SIZE)
        self.plot_loss(history)
        self.plot_accuracy(history)
        return history


    def evaluate(self):
        score = self.model.evaluate(self.x_test, self.y_test)
        print(f'[INFO] Test loss: {score[0]}')
        print(f'[INFO] Test accuracy: {score[1]}')


    def _vectorize_sequences(self, sequences, dimension=10000):
        results = np.zeros((len(sequences), dimension))
        for i, sequence in enumerate(sequences):
            for j in sequence:
                results[i, j] = 1.
        return results
    def k_fold(self):
          k = 3
          num_val_samples = len(self.x_train) // k
          num_epochs = 40
```

```
        all_scores = []
        for i in range(self, k):
            print(f'Processing fold # {i}')
            val_data = self.x_train[i * num_val_samples: (i+1) * num_val
            val_targets = self.y_train[i * num_val_samples: (i+1) * num_
            partial_train_data = np.concatenate(
                                    [self.x_train[:i * num_val_samples],
                                    self.x_train[(i+1) * num_val_samples
                                    axis=0)
            partial_train_targets = np.concatenate(
                                    [self.y_train[:i * num_val_samples],
                                    self.y_train[(i+1)*num_val_samples:]
                                    axis=0)
            model = self.train()
            model.fit(partial_train_data,
                      partial_train_targets,
                      epochs=num_epochs,
                      batch_size=16,
                      verbose=0)
            val_mse, val_mae = model.evaluate(val_data, val_targets, ver
            all_scores.append(val_mae)
```

In [ ]:
```
# calling Boston_Housing class
boston_housing = Boston_Housing()
```

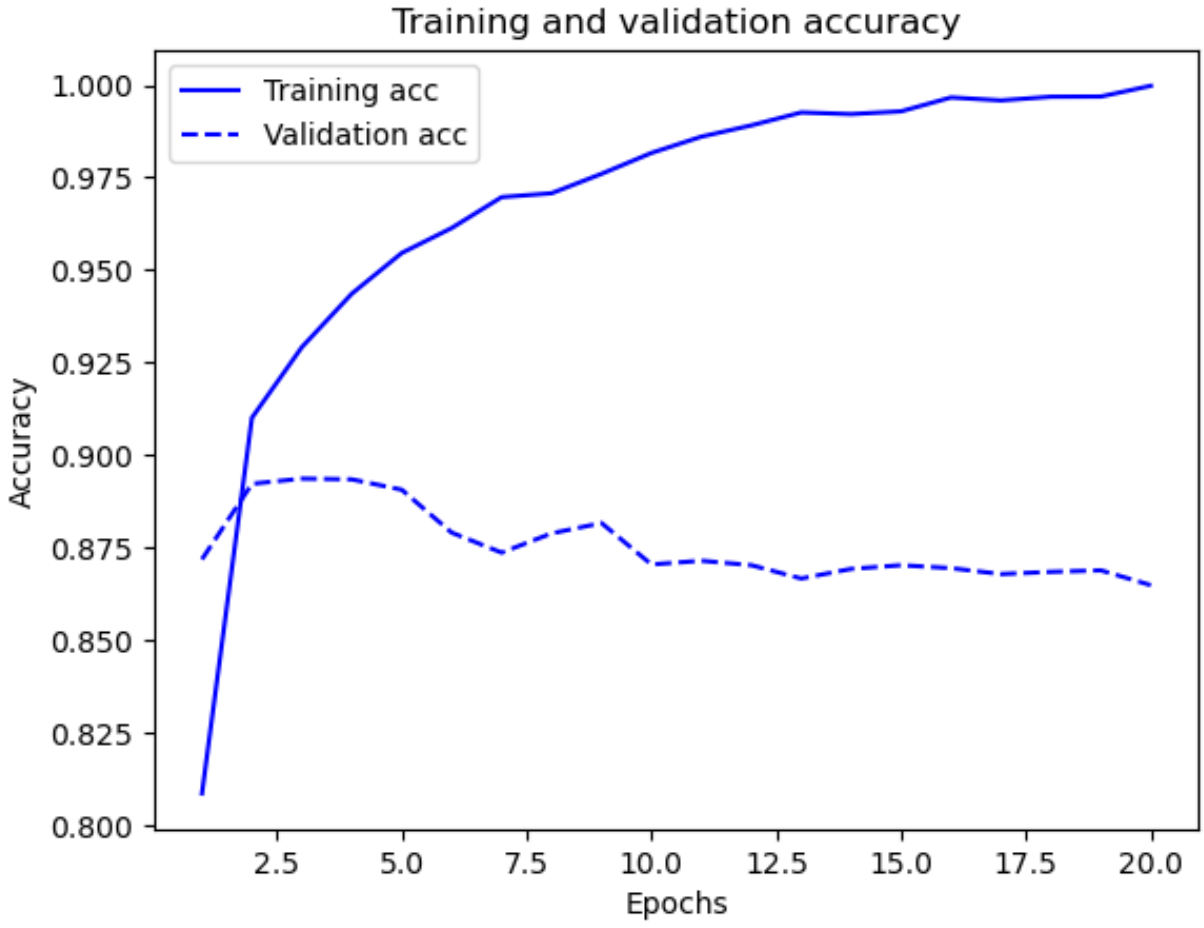
In [ ]:
```
from imdb import Imdb
imdb=Imdb()
imdb._load(num_words=10000)
imdb.train(epochs=20)
imdb.evaluate()

    #%%
#from reuters import Reuters
reuters=Reuters()
reuters._load(num_words=10000)
reuters.train(epochs=30)
reuters.evaluate()
#%%
# Boston Housing price analysis



# from boston_housing import Boston_Housing

boston_housing= Boston_Housing()
boston_housing._load()
boston_housing.train(epochs=30)
boston_housing.evaluate()
```

```
Epoch 1/20
2022-11-14 22:36:28.753666: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
```

```
40/40 [==============================] - 1s 24ms/step - loss: 0.4593 - a
ccuracy: 0.8085 - val_loss: 0.3423 - val_accuracy: 0.8718
Epoch 2/20
 1/40 [..............................] - ETA: 0s - loss: 0.2746 - accura
cy: 0.9199
2022-11-14 22:36:29.736919: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
```

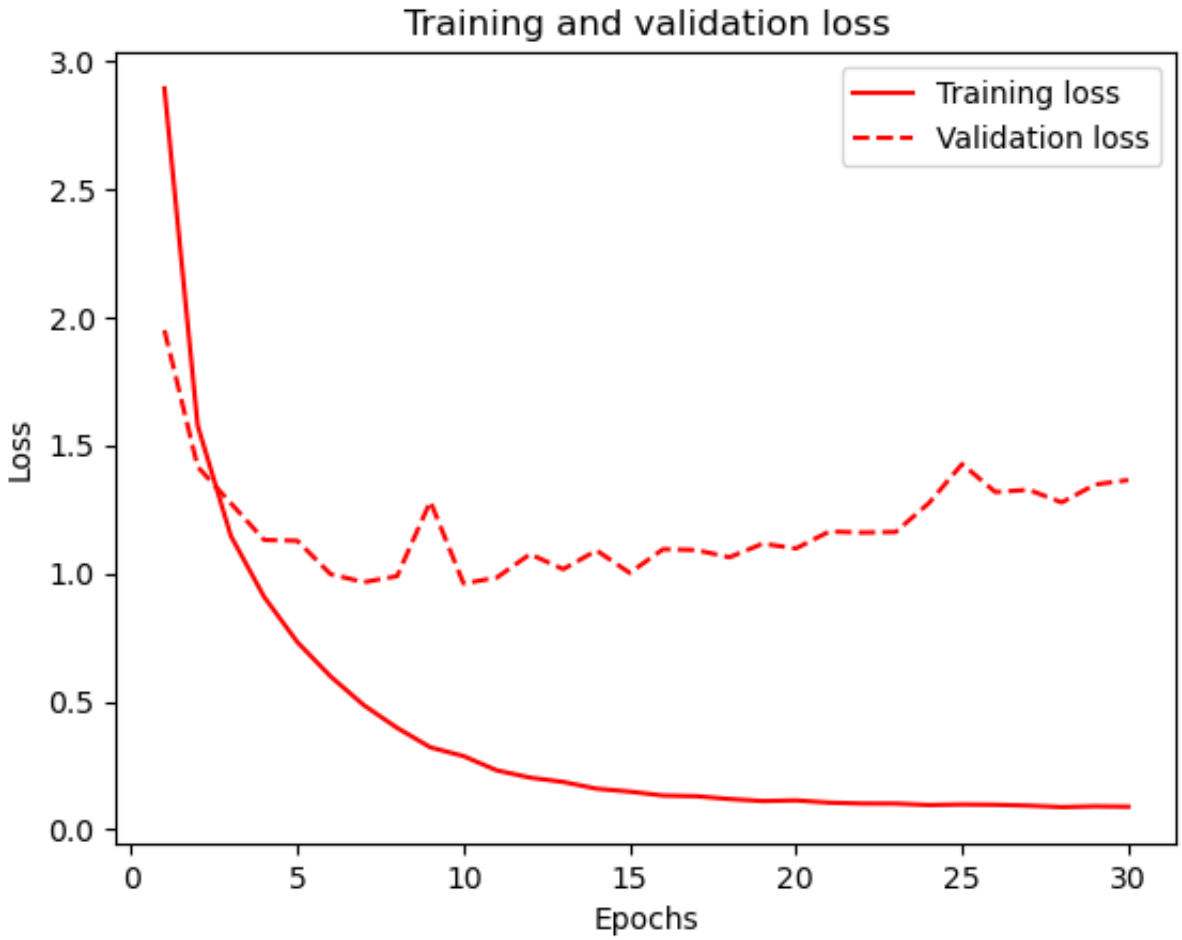
```
40/40 [==============================] - 1s 13ms/step - loss: 0.2620 - a
ccuracy: 0.9101 - val_loss: 0.2768 - val_accuracy: 0.8922
Epoch 3/20
40/40 [==============================] - 1s 15ms/step - loss: 0.1973 - a
ccuracy: 0.9292 - val_loss: 0.2718 - val_accuracy: 0.8936
Epoch 4/20
40/40 [==============================] - 1s 13ms/step - loss: 0.1594 - a
ccuracy: 0.9436 - val_loss: 0.2801 - val_accuracy: 0.8934
Epoch 5/20
40/40 [==============================] - 1s 13ms/step - loss: 0.1323 - a
ccuracy: 0.9546 - val_loss: 0.2993 - val_accuracy: 0.8906
Epoch 6/20
40/40 [==============================] - 1s 13ms/step - loss: 0.1129 - a
ccuracy: 0.9614 - val_loss: 0.3603 - val_accuracy: 0.8790
Epoch 7/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0918 - a
ccuracy: 0.9697 - val_loss: 0.3860 - val_accuracy: 0.8736
Epoch 8/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0831 - a
ccuracy: 0.9708 - val_loss: 0.3903 - val_accuracy: 0.8788
Epoch 9/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0709 - a
ccuracy: 0.9760 - val_loss: 0.4083 - val_accuracy: 0.8816
Epoch 10/20
40/40 [==============================] - 1s 15ms/step - loss: 0.0568 - a
ccuracy: 0.9817 - val_loss: 0.4769 - val_accuracy: 0.8704
Epoch 11/20
40/40 [==============================] - 1s 14ms/step - loss: 0.0475 - a
ccuracy: 0.9861 - val_loss: 0.4852 - val_accuracy: 0.8714
Epoch 12/20
40/40 [==============================] - 1s 14ms/step - loss: 0.0381 - a
ccuracy: 0.9891 - val_loss: 0.5238 - val_accuracy: 0.8702
Epoch 13/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0311 - a
ccuracy: 0.9926 - val_loss: 0.6067 - val_accuracy: 0.8666
Epoch 14/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0302 - a
ccuracy: 0.9922 - val_loss: 0.5974 - val_accuracy: 0.8692
Epoch 15/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0235 - a
ccuracy: 0.9929 - val_loss: 0.6334 - val_accuracy: 0.8702
Epoch 16/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0161 - a
ccuracy: 0.9967 - val_loss: 0.6571 - val_accuracy: 0.8694
Epoch 17/20
40/40 [==============================] - 0s 13ms/step - loss: 0.0159 - a
ccuracy: 0.9959 - val_loss: 0.7014 - val_accuracy: 0.8678
Epoch 18/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0137 - a
ccuracy: 0.9969 - val_loss: 0.7337 - val_accuracy: 0.8684
Epoch 19/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0111 - a
ccuracy: 0.9969 - val_loss: 0.7681 - val_accuracy: 0.8688
Epoch 20/20
40/40 [==============================] - 1s 13ms/step - loss: 0.0041 - a
ccuracy: 0.9998 - val_loss: 0.8404 - val_accuracy: 0.8648
```

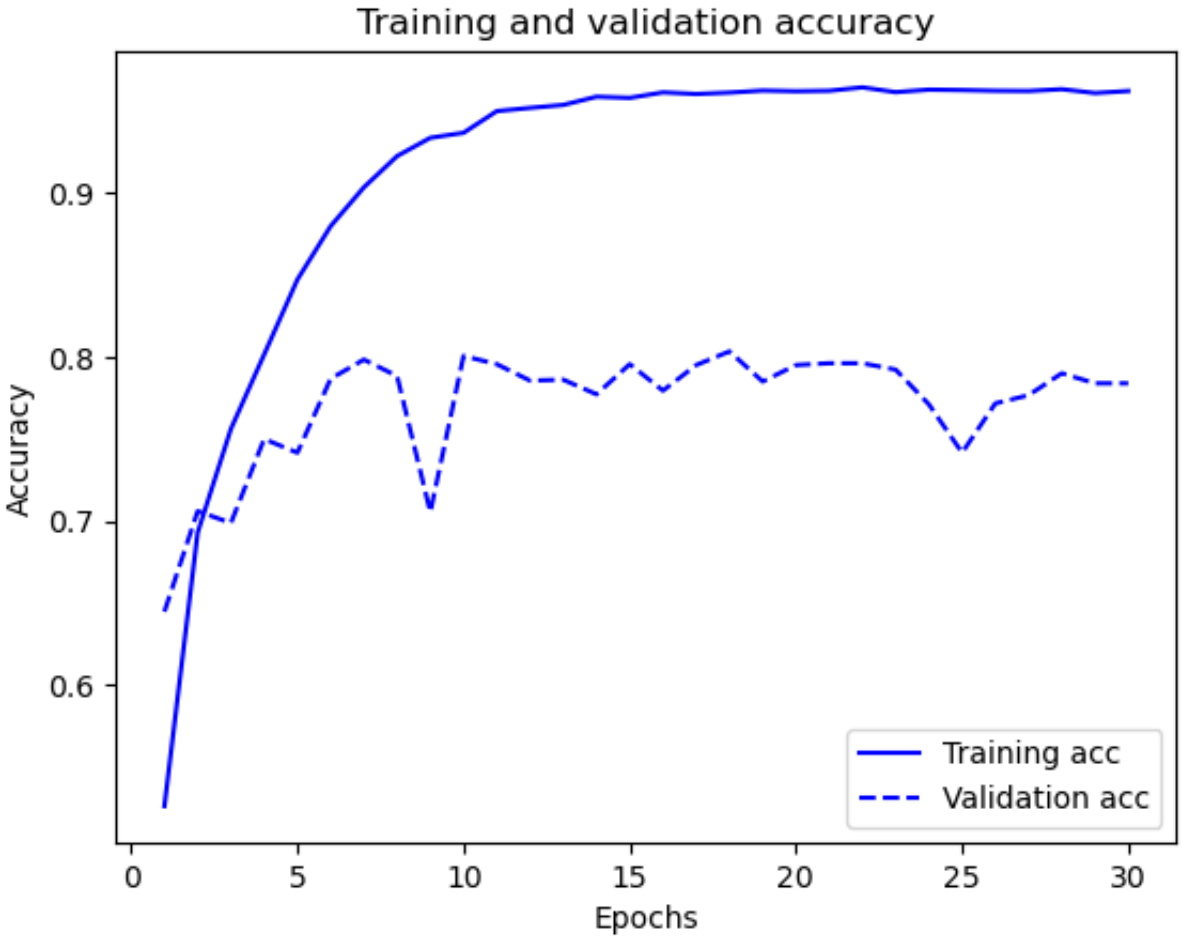

Training and validation loss
Training loss
Validation loss
Loss
Epochs
0.0
0.2
0.4
0.6
0.8
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Training and validation accuracy
Training acc
Validation acc
Accuracy
Epochs
0.800
0.825
0.850
0.875
0.900
0.925
0.950
0.975
1.000
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

```
782/782 [==============================] - 4s 5ms/step - loss: 0.9276 -
accuracy: 0.8432
[INFO] Test loss: 0.9275516271591187
[INFO] Test accuracy: 0.8432000279426575
Epoch 1/30
2022-11-14 22:36:46.892253: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
15/15 [==============================] - 1s 35ms/step - loss: 2.8928 - a
ccuracy: 0.5264 - val_loss: 1.9488 - val_accuracy: 0.6450
Epoch 2/30
 5/15 [=========>....................] - ETA: 0s - loss: 1.7266 - accura
cy: 0.6762
2022-11-14 22:36:47.566312: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
15/15 [==============================] - 0s 19ms/step - loss: 1.5767 - a
ccuracy: 0.6931 - val_loss: 1.4160 - val_accuracy: 0.7062
Epoch 3/30
15/15 [==============================] - 0s 18ms/step - loss: 1.1454 - a
ccuracy: 0.7563 - val_loss: 1.2732 - val_accuracy: 0.6989
Epoch 4/30
15/15 [==============================] - 0s 17ms/step - loss: 0.9079 - a
ccuracy: 0.8017 - val_loss: 1.1303 - val_accuracy: 0.7501
Epoch 5/30
15/15 [==============================] - 0s 18ms/step - loss: 0.7314 - a
ccuracy: 0.8473 - val_loss: 1.1269 - val_accuracy: 0.7418
Epoch 6/30
15/15 [==============================] - 0s 18ms/step - loss: 0.5974 - a
ccuracy: 0.8799 - val_loss: 0.9965 - val_accuracy: 0.7869
Epoch 7/30
15/15 [==============================] - 0s 18ms/step - loss: 0.4852 - a
ccuracy: 0.9035 - val_loss: 0.9655 - val_accuracy: 0.7986
Epoch 8/30
15/15 [==============================] - 0s 18ms/step - loss: 0.3968 - a
ccuracy: 0.9226 - val_loss: 0.9888 - val_accuracy: 0.7885
Epoch 9/30
15/15 [==============================] - 0s 19ms/step - loss: 0.3207 - a
ccuracy: 0.9338 - val_loss: 1.2804 - val_accuracy: 0.7056
Epoch 10/30
15/15 [==============================] - 0s 18ms/step - loss: 0.2865 - a
ccuracy: 0.9368 - val_loss: 0.9601 - val_accuracy: 0.8008
Epoch 11/30
15/15 [==============================] - 0s 18ms/step - loss: 0.2307 - a
ccuracy: 0.9500 - val_loss: 0.9833 - val_accuracy: 0.7958
Epoch 12/30
15/15 [==============================] - 0s 18ms/step - loss: 0.2026 - a
ccuracy: 0.9520 - val_loss: 1.0748 - val_accuracy: 0.7858
Epoch 13/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1856 - a
ccuracy: 0.9538 - val_loss: 1.0161 - val_accuracy: 0.7863
Epoch 14/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1593 - a
ccuracy: 0.9588 - val_loss: 1.0894 - val_accuracy: 0.7774
Epoch 15/30
15/15 [==============================] - 0s 17ms/step - loss: 0.1479 - a
```

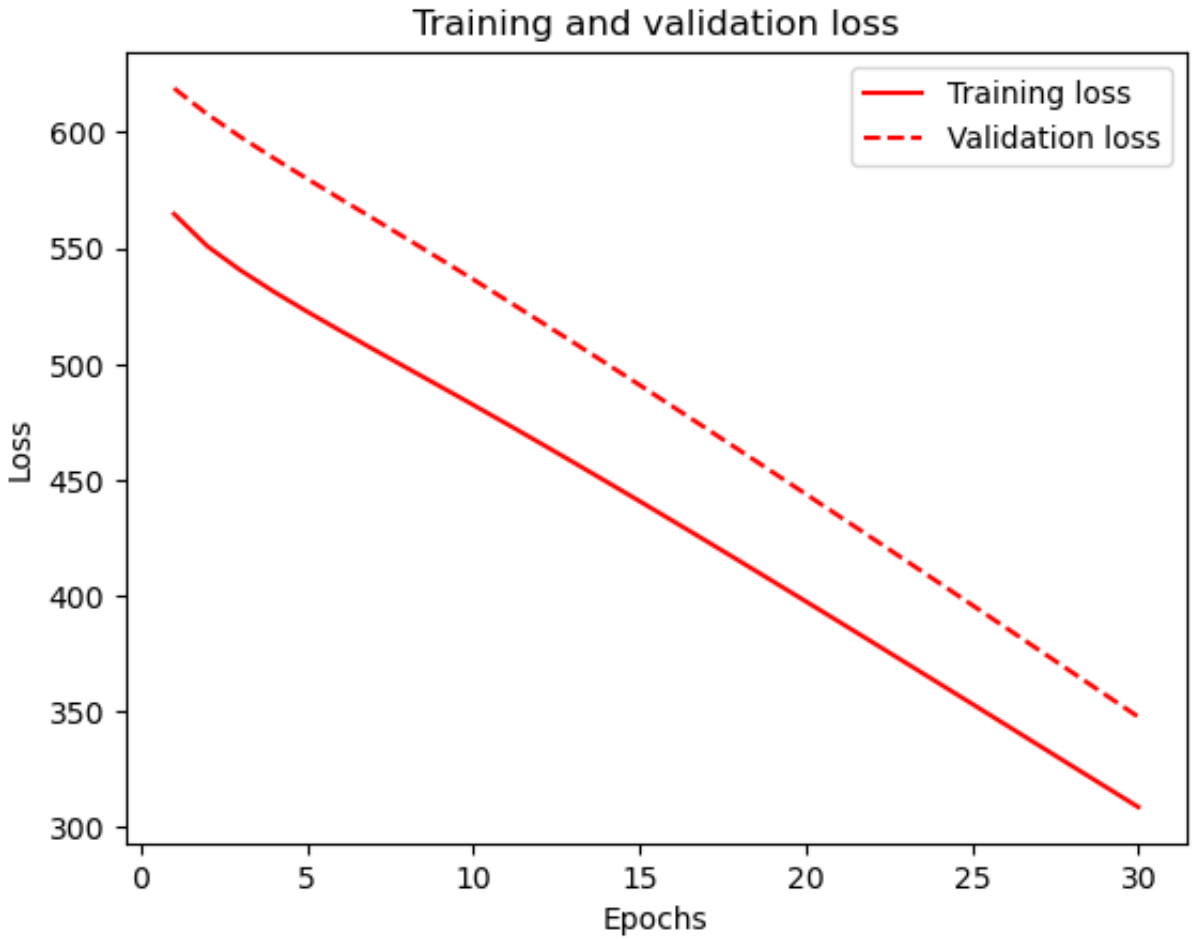
```
ccuracy: 0.9581 - val_loss: 1.0015 - val_accuracy: 0.7958
Epoch 16/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1326 - a
ccuracy: 0.9614 - val_loss: 1.0939 - val_accuracy: 0.7796
Epoch 17/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1299 - a
ccuracy: 0.9605 - val_loss: 1.0909 - val_accuracy: 0.7952
Epoch 18/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1189 - a
ccuracy: 0.9613 - val_loss: 1.0626 - val_accuracy: 0.8036
Epoch 19/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1111 - a
ccuracy: 0.9626 - val_loss: 1.1155 - val_accuracy: 0.7852
Epoch 20/30
15/15 [==============================] - 0s 17ms/step - loss: 0.1137 - a
ccuracy: 0.9621 - val_loss: 1.0971 - val_accuracy: 0.7952
Epoch 21/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1048 - a
ccuracy: 0.9624 - val_loss: 1.1629 - val_accuracy: 0.7963
Epoch 22/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1014 - a
ccuracy: 0.9645 - val_loss: 1.1591 - val_accuracy: 0.7963
Epoch 23/30
15/15 [==============================] - 0s 18ms/step - loss: 0.1012 - a
ccuracy: 0.9616 - val_loss: 1.1614 - val_accuracy: 0.7924
Epoch 24/30
15/15 [==============================] - 0s 18ms/step - loss: 0.0955 - a
ccuracy: 0.9631 - val_loss: 1.2738 - val_accuracy: 0.7713
Epoch 25/30
15/15 [==============================] - 0s 18ms/step - loss: 0.0978 - a
ccuracy: 0.9628 - val_loss: 1.4268 - val_accuracy: 0.7418
Epoch 26/30
15/15 [==============================] - 0s 17ms/step - loss: 0.0966 - a
ccuracy: 0.9624 - val_loss: 1.3171 - val_accuracy: 0.7718
Epoch 27/30
15/15 [==============================] - 0s 18ms/step - loss: 0.0931 - a
ccuracy: 0.9623 - val_loss: 1.3261 - val_accuracy: 0.7769
Epoch 28/30
15/15 [==============================] - 0s 18ms/step - loss: 0.0875 - a
ccuracy: 0.9634 - val_loss: 1.2766 - val_accuracy: 0.7902
Epoch 29/30
15/15 [==============================] - 0s 18ms/step - loss: 0.0906 - a
ccuracy: 0.9610 - val_loss: 1.3465 - val_accuracy: 0.7841
Epoch 30/30
15/15 [==============================] - 0s 17ms/step - loss: 0.0891 - a
ccuracy: 0.9623 - val_loss: 1.3637 - val_accuracy: 0.7841
```

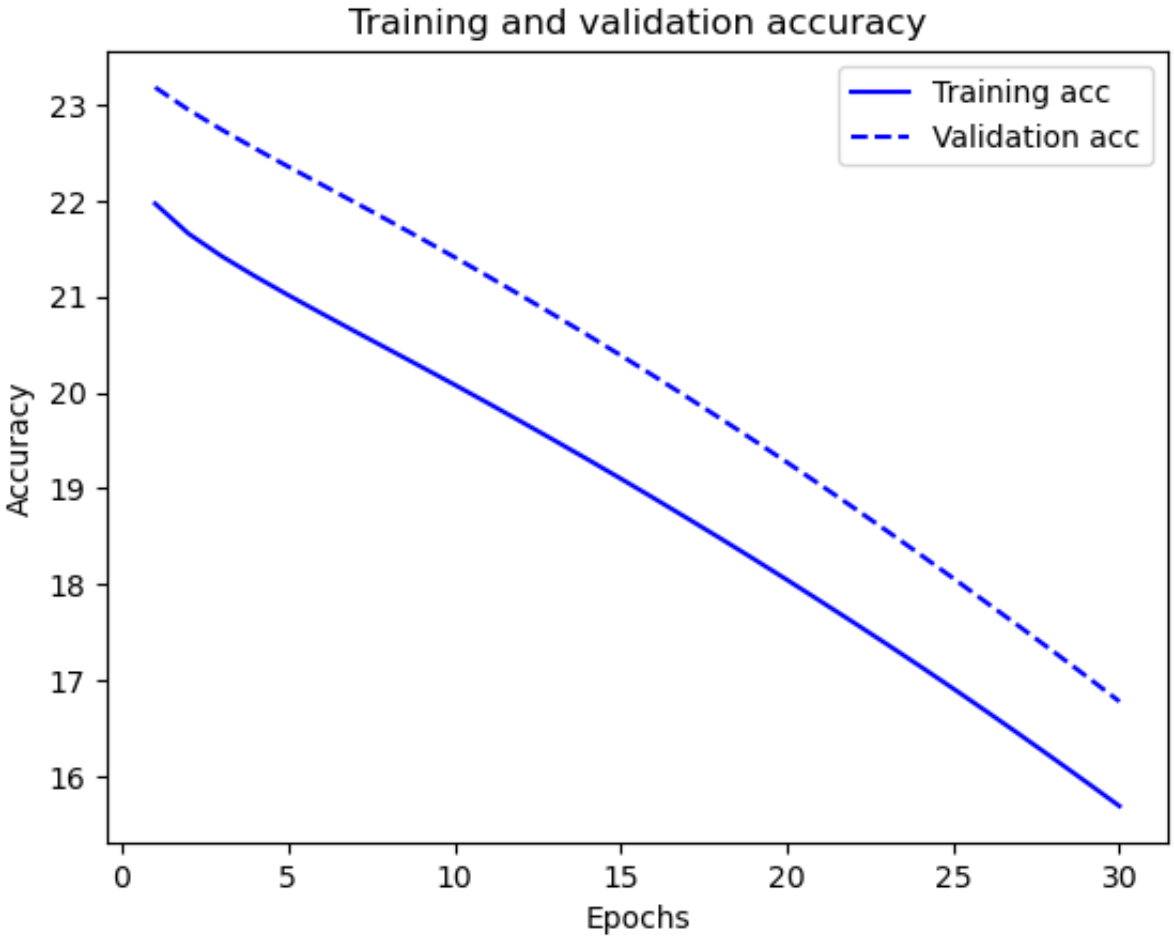

Training and validation loss
Training loss
Validation loss
Loss
Epochs
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
5
10
15
20
25
30


Training and validation accuracy
Training acc
Validation acc
Accuracy
Epochs
0.9
0.8
0.7
0.6
0
5
10
15
20
25
30

```
71/71 [==============================] - 1s 10ms/step - loss: 1.4665 - a
ccuracy: 0.7729
[INFO] Test loss: 1.4664809703826904
[INFO] Test accuracy: 0.7729296684265137
Epoch 1/30
1/1 [==============================] - ETA: 0s - loss: 564.7747 - mae: 2
1.9693
2022-11-14 22:36:56.467337: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
1/1 [==============================] - 0s 409ms/step - loss: 564.7747 -
mae: 21.9693 - val_loss: 618.9827 - val_mae: 23.1810
Epoch 2/30
1/1 [==============================] - 0s 50ms/step - loss: 550.8596 - m
ae: 21.6566 - val_loss: 607.8082 - val_mae: 22.9473
Epoch 3/30
1/1 [==============================] - 0s 30ms/step - loss: 540.4371 - m
ae: 21.4212 - val_loss: 597.9879 - val_mae: 22.7399
Epoch 4/30
1/1 [==============================] - 0s 34ms/step - loss: 531.2383 - m
ae: 21.2123 - val_loss: 588.8207 - val_mae: 22.5445
Epoch 5/30
1/1 [==============================] - ETA: 0s - loss: 522.6472 - mae: 2
1.0154
2022-11-14 22:36:56.694447: I tensorflow/core/grappler/optimizers/custom
_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU i
s enabled.
1/1 [==============================] - 0s 34ms/step - loss: 522.6472 - m
ae: 21.0154 - val_loss: 579.9846 - val_mae: 22.3551
Epoch 6/30
1/1 [==============================] - 0s 29ms/step - loss: 514.4047 - m
ae: 20.8254 - val_loss: 571.3124 - val_mae: 22.1685
Epoch 7/30
1/1 [==============================] - 0s 28ms/step - loss: 506.3159 - m
ae: 20.6384 - val_loss: 562.7026 - val_mae: 21.9825
Epoch 8/30
1/1 [==============================] - 0s 28ms/step - loss: 498.3360 - m
ae: 20.4535 - val_loss: 554.0555 - val_mae: 21.7953
Epoch 9/30
1/1 [==============================] - 0s 28ms/step - loss: 490.3787 - m
ae: 20.2687 - val_loss: 545.3280 - val_mae: 21.6052
Epoch 10/30
1/1 [==============================] - 0s 28ms/step - loss: 482.3489 - m
ae: 20.0815 - val_loss: 536.4844 - val_mae: 21.4111
Epoch 11/30
1/1 [==============================] - 0s 29ms/step - loss: 474.2122 - m
ae: 19.8909 - val_loss: 527.5212 - val_mae: 21.2130
Epoch 12/30
1/1 [==============================] - 0s 27ms/step - loss: 465.9836 - m
ae: 19.6973 - val_loss: 518.4819 - val_mae: 21.0118
Epoch 13/30
1/1 [==============================] - 0s 29ms/step - loss: 457.6729 - m
ae: 19.5018 - val_loss: 509.3884 - val_mae: 20.8075
Epoch 14/30
1/1 [==============================] - 0s 44ms/step - loss: 449.2865 - m
ae: 19.3037 - val_loss: 500.2132 - val_mae: 20.5994
```

```
Epoch 15/30
1/1 [==============================] - 0s 31ms/step - loss: 440.8206 - m
ae: 19.1022 - val_loss: 490.9606 - val_mae: 20.3874
Epoch 16/30
1/1 [==============================] - 0s 28ms/step - loss: 432.2888 - m
ae: 18.8975 - val_loss: 481.6207 - val_mae: 20.1713
Epoch 17/30
1/1 [==============================] - 0s 28ms/step - loss: 423.6924 - m
ae: 18.6892 - val_loss: 472.2291 - val_mae: 19.9516
Epoch 18/30
1/1 [==============================] - 0s 29ms/step - loss: 415.0319 - m
ae: 18.4788 - val_loss: 462.8207 - val_mae: 19.7287
Epoch 19/30
1/1 [==============================] - 0s 31ms/step - loss: 406.3150 - m
ae: 18.2647 - val_loss: 453.3689 - val_mae: 19.5021
Epoch 20/30
1/1 [==============================] - 0s 29ms/step - loss: 397.5673 - m
ae: 18.0472 - val_loss: 443.8889 - val_mae: 19.2720
Epoch 21/30
1/1 [==============================] - 0s 28ms/step - loss: 388.7836 - m
ae: 17.8259 - val_loss: 434.3673 - val_mae: 19.0378
Epoch 22/30
1/1 [==============================] - 0s 26ms/step - loss: 379.9724 - m
ae: 17.6031 - val_loss: 424.8190 - val_mae: 18.7998
Epoch 23/30
1/1 [==============================] - 0s 27ms/step - loss: 371.1253 - m
ae: 17.3770 - val_loss: 415.2525 - val_mae: 18.5583
Epoch 24/30
1/1 [==============================] - 0s 26ms/step - loss: 362.2518 - m
ae: 17.1471 - val_loss: 405.6491 - val_mae: 18.3128
Epoch 25/30
1/1 [==============================] - 0s 26ms/step - loss: 353.3479 - m
ae: 16.9131 - val_loss: 396.0240 - val_mae: 18.0634
Epoch 26/30
1/1 [==============================] - 0s 26ms/step - loss: 344.4354 - m
ae: 16.6754 - val_loss: 386.3694 - val_mae: 17.8097
Epoch 27/30
1/1 [==============================] - 0s 26ms/step - loss: 335.5133 - m
ae: 16.4337 - val_loss: 376.6990 - val_mae: 17.5580
Epoch 28/30
1/1 [==============================] - 0s 27ms/step - loss: 326.5851 - m
ae: 16.1879 - val_loss: 367.0369 - val_mae: 17.3025
Epoch 29/30
1/1 [==============================] - 0s 28ms/step - loss: 317.6822 - m
ae: 15.9391 - val_loss: 357.3693 - val_mae: 17.0426
Epoch 30/30
1/1 [==============================] - 0s 27ms/step - loss: 308.7940 - m
ae: 15.6878 - val_loss: 347.7158 - val_mae: 16.7785
```


Training and validation loss
Training loss
Validation loss
Loss
600
550
500
450
400
350
300
0
5
10
15
20
25
30
Epochs


Training and validation accuracy
Training acc
Validation acc
Accuracy
23
22
21
20
19
18
17
16
0
5
10
15
20
25
30
Epochs

```
4/4 [==============================] - 0s 11ms/step - loss: 326.2187 - m
ae: 16.1766
[INFO] Test loss: 326.21868896484375
[INFO] Test accuracy: 16.176576614379883
```

In [ ]:

```python
import cv2
import mediapipe as mp
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_pose = mp.solutions.pose

# For webcam input:
cap = cv2.VideoCapture(0)
with mp_pose.Pose(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as pose:
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      # If loading a video, use 'break' instead of 'continue'.
      continue

    # To improve performance, optionally mark the image as not writeable
    # pass by reference.
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = pose.process(image)

    # Draw the pose annotation on the image.
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    mp_drawing.draw_landmarks(
        image,
        results.pose_landmarks,
        mp_pose.POSE_CONNECTIONS,
        landmark_drawing_spec=mp_drawing_styles.get_default_pose_landmark
    # Flip the image horizontally for a selfie-view display.
    cv2.imshow('MediaPipe Pose', cv2.flip(image, 1))
    if cv2.waitKey(5) & 0xFF == 27:
      break
cap.release()
```

```
OpenCV: not authorized to capture video (status 0), requesting...
OpenCV: camera failed to properly initialize!
```

In [ ]:
```
import cv2
import mediapipe as mp
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_hands = mp.solutions.hands

# For webcam input:
cap = cv2.VideoCapture(0)
with mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      # If loading a video, use 'break' instead of 'continue'.
      continue

    # To improve performance, optionally mark the image as not writeable
    # pass by reference.
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = hands.process(image)

    # Draw the hand annotations on the image.
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    if results.multi_hand_landmarks:
      for hand_landmarks in results.multi_hand_landmarks:
        mp_drawing.draw_landmarks(
            image,
            hand_landmarks,
            mp_hands.HAND_CONNECTIONS,
            mp_drawing_styles.get_default_hand_landmarks_style(),
            mp_drawing_styles.get_default_hand_connections_style())
    # Flip the image horizontally for a selfie-view display.
    cv2.imshow('MediaPipe Hands', cv2.flip(image, 1))
    if cv2.waitKey(5) & 0xFF == 27:
      break
cap.release()
```

In [ ]: