HW1

September 21, 2023

[1]:
```
#Loading the IMDB dataset
```

[2]:
```
from tensorflow.keras.datasets import imdb
(train_data, train_labels), (test_data, test_labels) = imdb.load_data(
    num_words=10000)
```

[3]:
```
train_data[0]
train_labels[0]
max([max(sequence) for sequence in train_data])
```

[3]:
```
9999
```

[4]:
```
#Decoding reviews back to text
```

[5]:
```
word_index = imdb.get_word_index()
reverse_word_index = dict(
    [(value, key) for (key, value) in word_index.items()])
decoded_review = " ".join(
    [reverse_word_index.get(i - 3, "?") for i in train_data[0]])
```

[6]:
```
#Encoding the integer sequences via multi-hot encoding
```

[7]:
```
import numpy as np
def vectorize_sequences(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
x_train = vectorize_sequences(train_data)
x_test = vectorize_sequences(test_data)
```

[8]:
```
x_train[0]
```

[8]:
```
array([0., 1., 1., …, 0., 0., 0.])
```

[9]:
```
y_train = np.asarray(train_labels).astype("float32")
y_test = np.asarray(test_labels).astype("float32")
```

[10]:
```
#Building your model
```

[11]:
```
#Model definition
```

[12]:
```
#original model from textbook
from tensorflow import keras
from tensorflow.keras import layers

model = keras.Sequential([
    layers.Dense(16, activation="relu"),
    layers.Dense(16, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
```

[14]:
```
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

[15]:
```
#Setting aside a validation set
```

[16]:
```
x_val = x_train[:10000]
partial_x_train = x_train[10000:]
y_val = y_train[:10000]
partial_y_train = y_train[10000:]
```

[17]:
```
#Training your model
```

[18]:
```
history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(x_val, y_val))
```

```
Epoch 1/20
30/30 [==============================] - 1s 28ms/step - loss: 0.5240 - accuracy:
0.7886 - val_loss: 0.4108 - val_accuracy: 0.8526
Epoch 2/20
30/30 [==============================] - 0s 14ms/step - loss: 0.3224 - accuracy:
0.8939 - val_loss: 0.3151 - val_accuracy: 0.8827
Epoch 3/20
30/30 [==============================] - 0s 13ms/step - loss: 0.2440 - accuracy:
0.9179 - val_loss: 0.2846 - val_accuracy: 0.8889
Epoch 4/20
30/30 [==============================] - 0s 12ms/step - loss: 0.1949 - accuracy:
0.9353 - val_loss: 0.2753 - val_accuracy: 0.8888
Epoch 5/20
30/30 [==============================] - 0s 12ms/step - loss: 0.1602 - accuracy:
0.9485 - val_loss: 0.2866 - val_accuracy: 0.8849
```

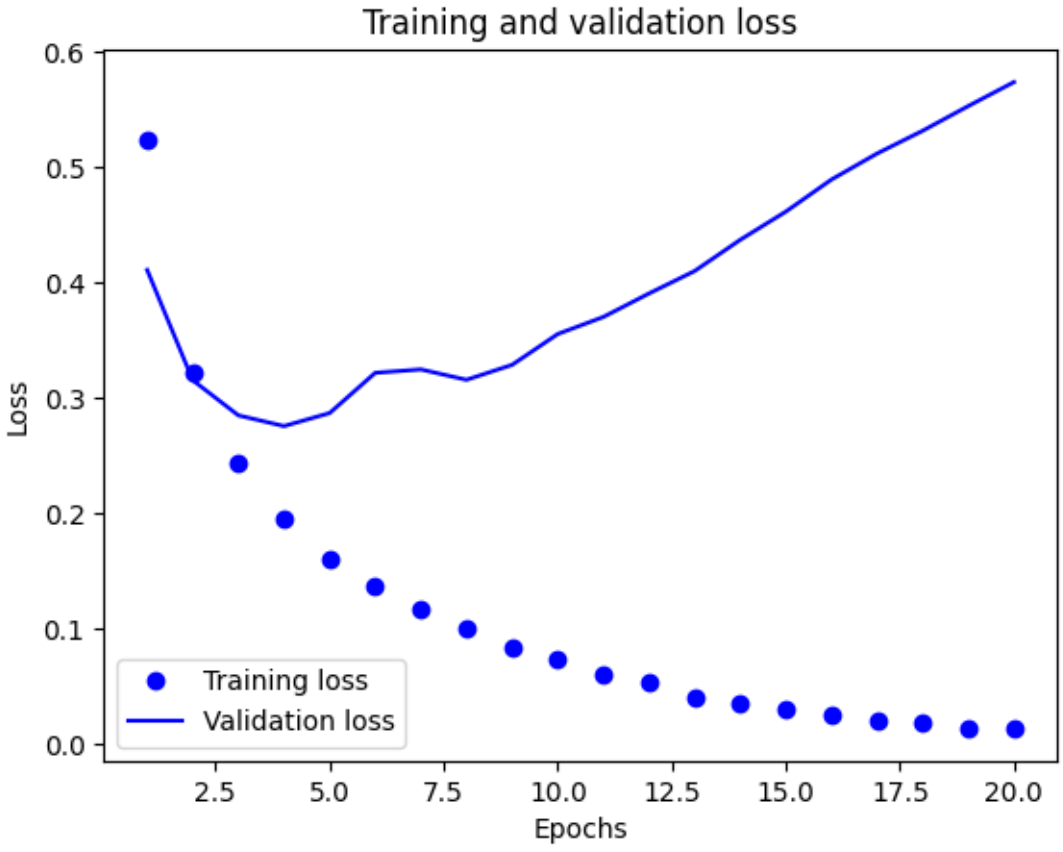
```
Epoch 6/20
30/30 [==============================] - 0s 12ms/step - loss: 0.1367 - accuracy:
0.9564 - val_loss: 0.3215 - val_accuracy: 0.8737
Epoch 7/20
30/30 [==============================] - 0s 12ms/step - loss: 0.1169 - accuracy:
0.9625 - val_loss: 0.3245 - val_accuracy: 0.8789
Epoch 8/20
30/30 [==============================] - 0s 11ms/step - loss: 0.1000 - accuracy:
0.9707 - val_loss: 0.3155 - val_accuracy: 0.8841
Epoch 9/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0833 - accuracy:
0.9771 - val_loss: 0.3284 - val_accuracy: 0.8814
Epoch 10/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0733 - accuracy:
0.9801 - val_loss: 0.3552 - val_accuracy: 0.8739
Epoch 11/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0597 - accuracy:
0.9847 - val_loss: 0.3700 - val_accuracy: 0.8771
Epoch 12/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0527 - accuracy:
0.9866 - val_loss: 0.3903 - val_accuracy: 0.8787
Epoch 13/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0401 - accuracy:
0.9926 - val_loss: 0.4098 - val_accuracy: 0.8764
Epoch 14/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0353 - accuracy:
0.9928 - val_loss: 0.4369 - val_accuracy: 0.8761
Epoch 15/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0293 - accuracy:
0.9943 - val_loss: 0.4613 - val_accuracy: 0.8760
Epoch 16/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0255 - accuracy:
0.9947 - val_loss: 0.4891 - val_accuracy: 0.8745
Epoch 17/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0199 - accuracy:
0.9967 - val_loss: 0.5117 - val_accuracy: 0.8725
Epoch 18/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0183 - accuracy:
0.9967 - val_loss: 0.5315 - val_accuracy: 0.8719
Epoch 19/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0133 - accuracy:
0.9983 - val_loss: 0.5529 - val_accuracy: 0.8722
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0135 - accuracy:
0.9981 - val_loss: 0.5737 - val_accuracy: 0.8712
```

[19]:
```
history_dict = history.history
history_dict.keys()
```

[19]:
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

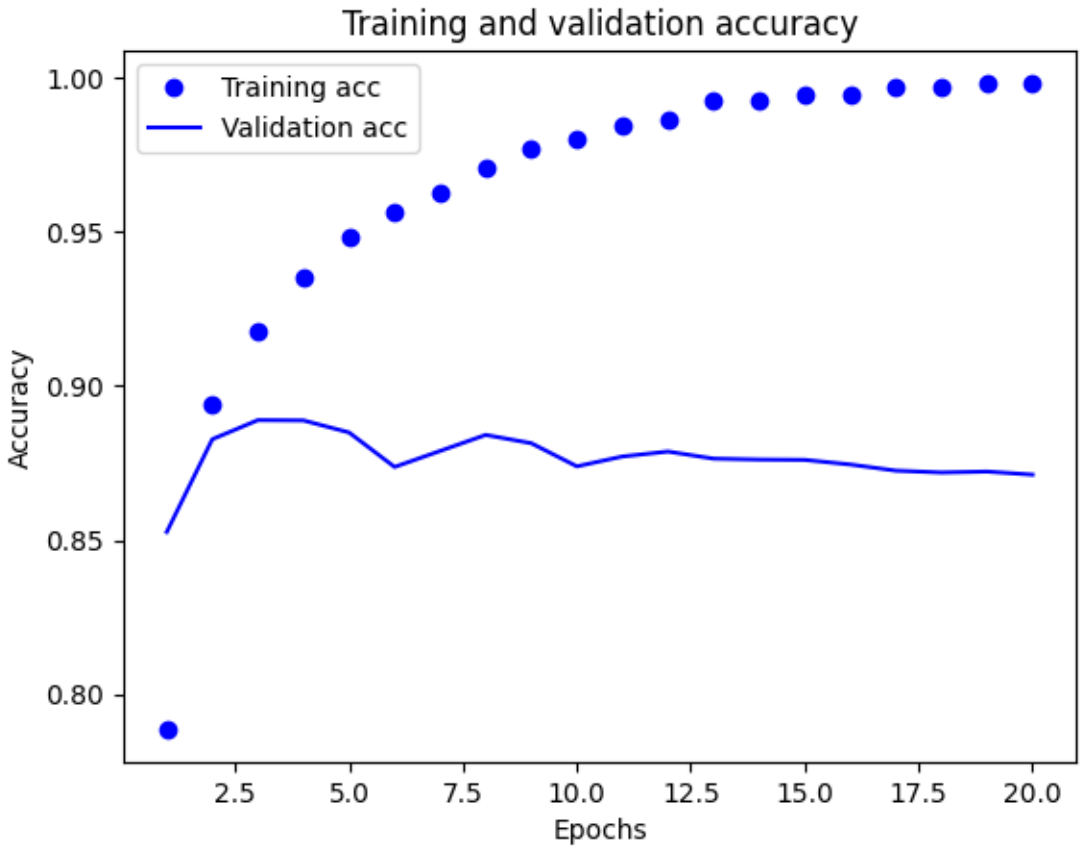
[20]:
```
#Plotting the training and validation loss
```

[21]:
```
import matplotlib.pyplot as plt
history_dict = history.history
loss_values = history_dict["loss"]
val_loss_values = history_dict["val_loss"]
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, "bo", label="Training loss")
plt.plot(epochs, val_loss_values, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



[21]:
```
#Plotting the training and validation accuracy
```

[22]:
```
plt.clf()
acc = history_dict["accuracy"]
val_acc = history_dict["val_accuracy"]
plt.plot(epochs, acc, "bo", label="Training acc")
plt.plot(epochs, val_acc, "b", label="Validation acc")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```



[23]:
```
#######retrian model######
```

[23]:
```
# the first one:
#four layers
                #16,16,16,1 nodes
model = keras.Sequential([
    layers.Dense(16, activation="relu"),
```

```
    layers.Dense(16, activation="relu"),
    layers.Dense(16, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

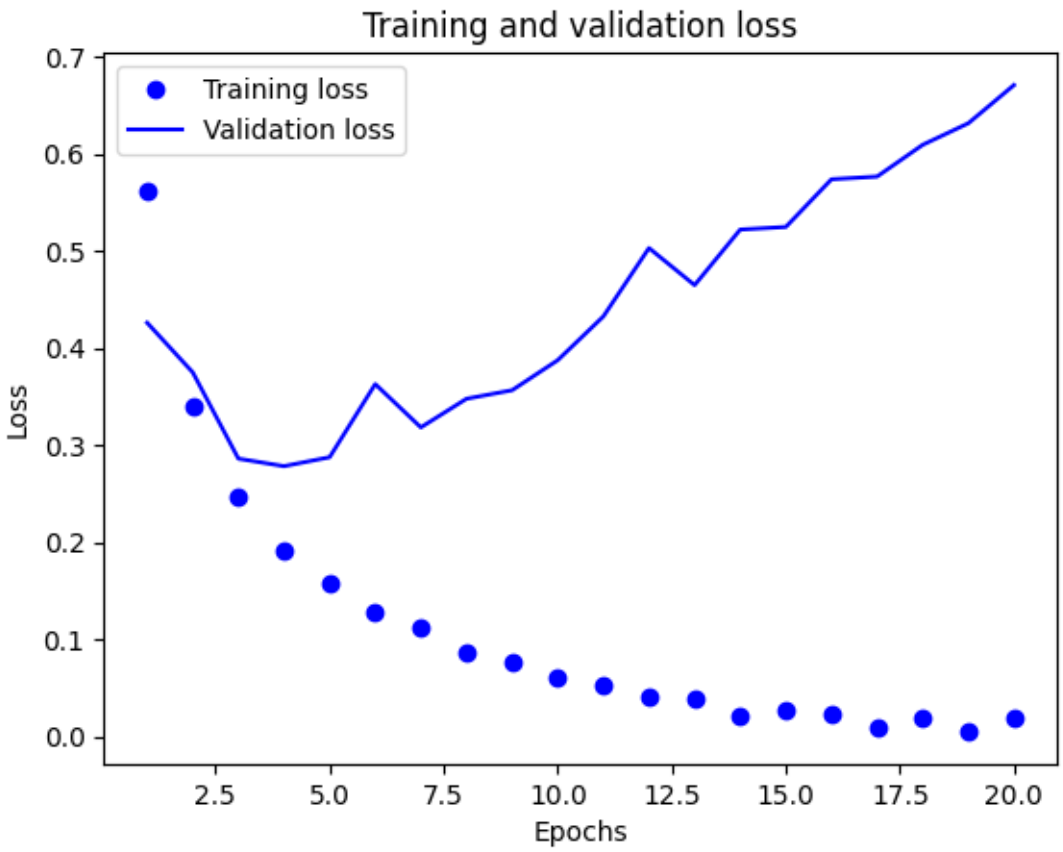
[24]:
```
x_val = x_train[:10000]
partial_x_train = x_train[10000:]
y_val = y_train[:10000]
partial_y_train = y_train[10000:]
```

[25]:
```
history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(x_val, y_val))
```

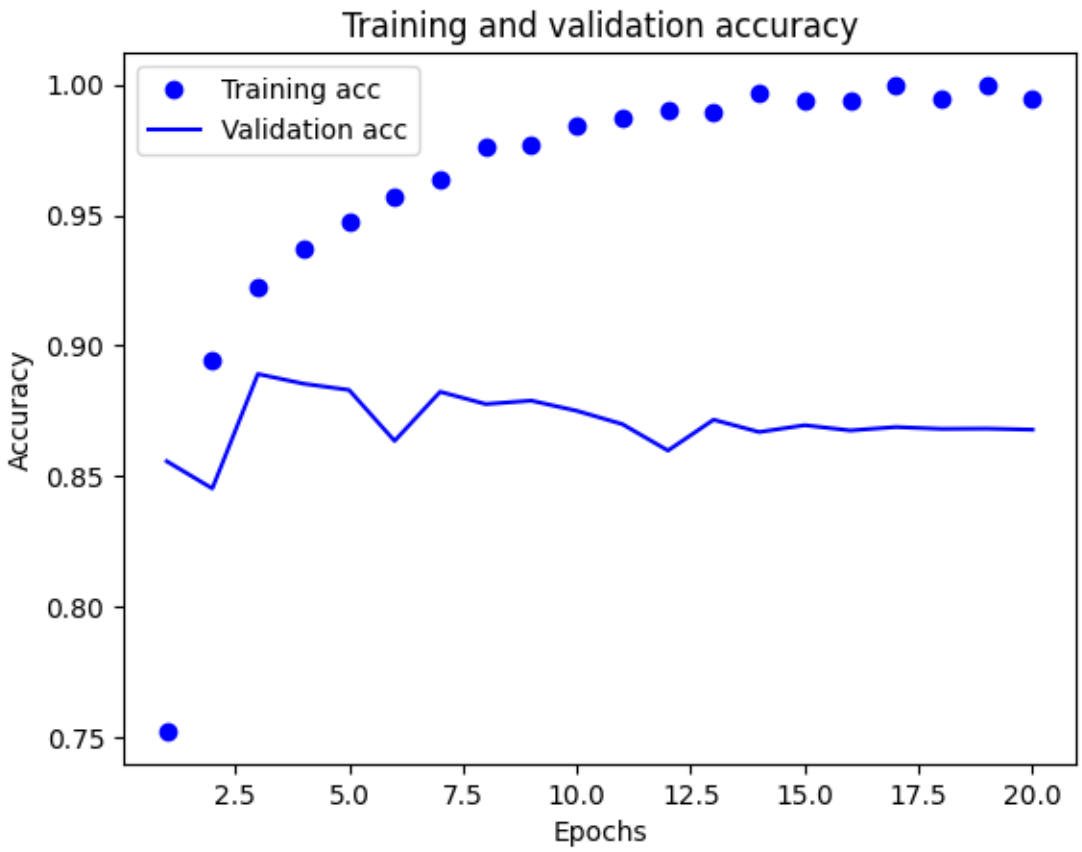
```
Epoch 1/20
30/30 [==============================] - 2s 51ms/step - loss: 0.5606 - accuracy:
0.7525 - val_loss: 0.4259 - val_accuracy: 0.8558
Epoch 2/20
30/30 [==============================] - 0s 13ms/step - loss: 0.3394 - accuracy:
0.8942 - val_loss: 0.3748 - val_accuracy: 0.8454
Epoch 3/20
30/30 [==============================] - 0s 13ms/step - loss: 0.2460 - accuracy:
0.9226 - val_loss: 0.2859 - val_accuracy: 0.8892
Epoch 4/20
30/30 [==============================] - 0s 11ms/step - loss: 0.1907 - accuracy:
0.9370 - val_loss: 0.2782 - val_accuracy: 0.8855
Epoch 5/20
30/30 [==============================] - 0s 12ms/step - loss: 0.1578 - accuracy:
0.9478 - val_loss: 0.2874 - val_accuracy: 0.8831
Epoch 6/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1285 - accuracy:
0.9573 - val_loss: 0.3627 - val_accuracy: 0.8636
Epoch 7/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1129 - accuracy:
0.9640 - val_loss: 0.3181 - val_accuracy: 0.8824
Epoch 8/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0861 - accuracy:
0.9759 - val_loss: 0.3476 - val_accuracy: 0.8777
Epoch 9/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0755 - accuracy:
0.9767 - val_loss: 0.3564 - val_accuracy: 0.8790
Epoch 10/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0614 - accuracy:
```

```
0.9840 - val_loss: 0.3873 - val_accuracy: 0.8751
Epoch 11/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0516 - accuracy:
0.9869 - val_loss: 0.4323 - val_accuracy: 0.8700
Epoch 12/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0404 - accuracy:
0.9905 - val_loss: 0.5028 - val_accuracy: 0.8599
Epoch 13/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0395 - accuracy:
0.9898 - val_loss: 0.4646 - val_accuracy: 0.8717
Epoch 14/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0203 - accuracy:
0.9971 - val_loss: 0.5218 - val_accuracy: 0.8670
Epoch 15/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0270 - accuracy:
0.9939 - val_loss: 0.5245 - val_accuracy: 0.8696
Epoch 16/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0231 - accuracy:
0.9940 - val_loss: 0.5736 - val_accuracy: 0.8676
Epoch 17/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0087 - accuracy:
0.9995 - val_loss: 0.5764 - val_accuracy: 0.8689
Epoch 18/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0188 - accuracy:
0.9945 - val_loss: 0.6091 - val_accuracy: 0.8682
Epoch 19/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0052 - accuracy:
0.9998 - val_loss: 0.6313 - val_accuracy: 0.8683
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0185 - accuracy:
0.9945 - val_loss: 0.6706 - val_accuracy: 0.8679
```

[26]:
```
import matplotlib.pyplot as plt
history_dict = history.history
loss_values = history_dict["loss"]
val_loss_values = history_dict["val_loss"]
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, "bo", label="Training loss")
plt.plot(epochs, val_loss_values, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



[27]:
```
plt.clf()
acc = history_dict["accuracy"]
val_acc = history_dict["val_accuracy"]
plt.plot(epochs, acc, "bo", label="Training acc")
plt.plot(epochs, val_acc, "b", label="Validation acc")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```



[28]:
```
# the second one:
#three layers
                # 64,64,1 nodes
model = keras.Sequential([
    layers.Dense(64, activation="relu"),
    layers.Dense(64, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

[29]:
```
x_val = x_train[:10000]
partial_x_train = x_train[10000:]
y_val = y_train[:10000]
partial_y_train = y_train[10000:]
```

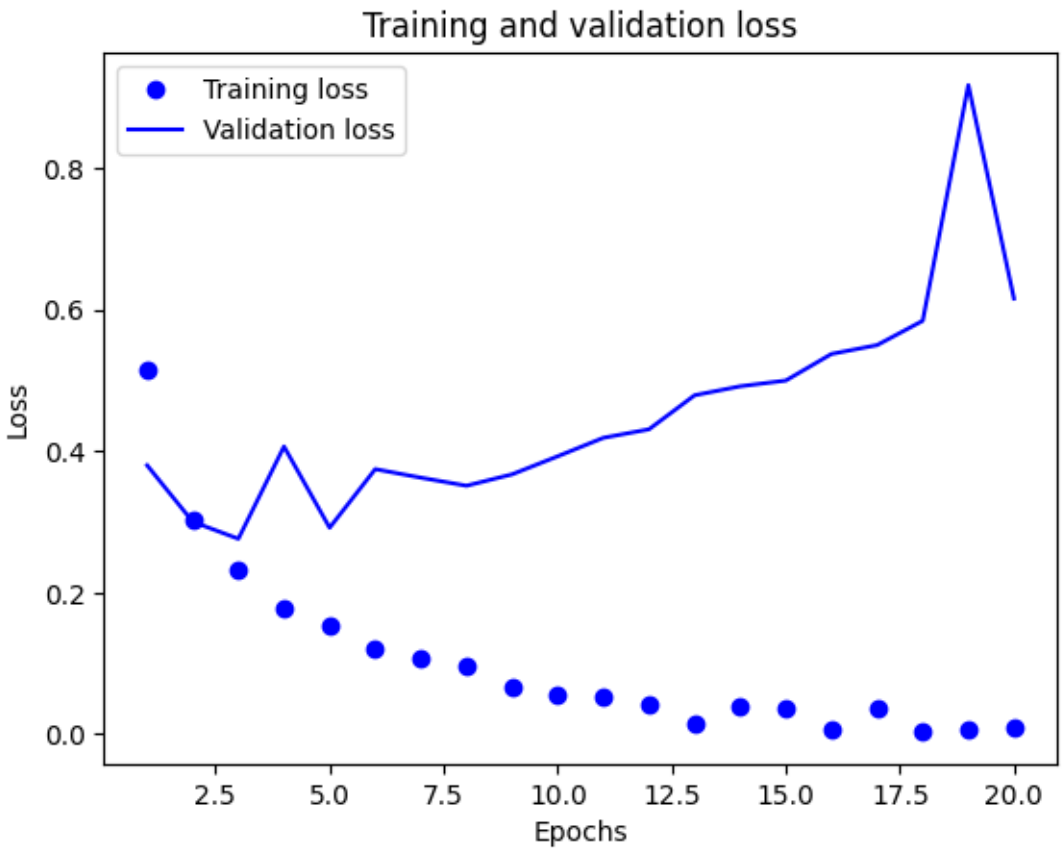
[30]:
```
history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs=20,
```

```
                    batch_size=512,
                    validation_data=(x_val, y_val))
```

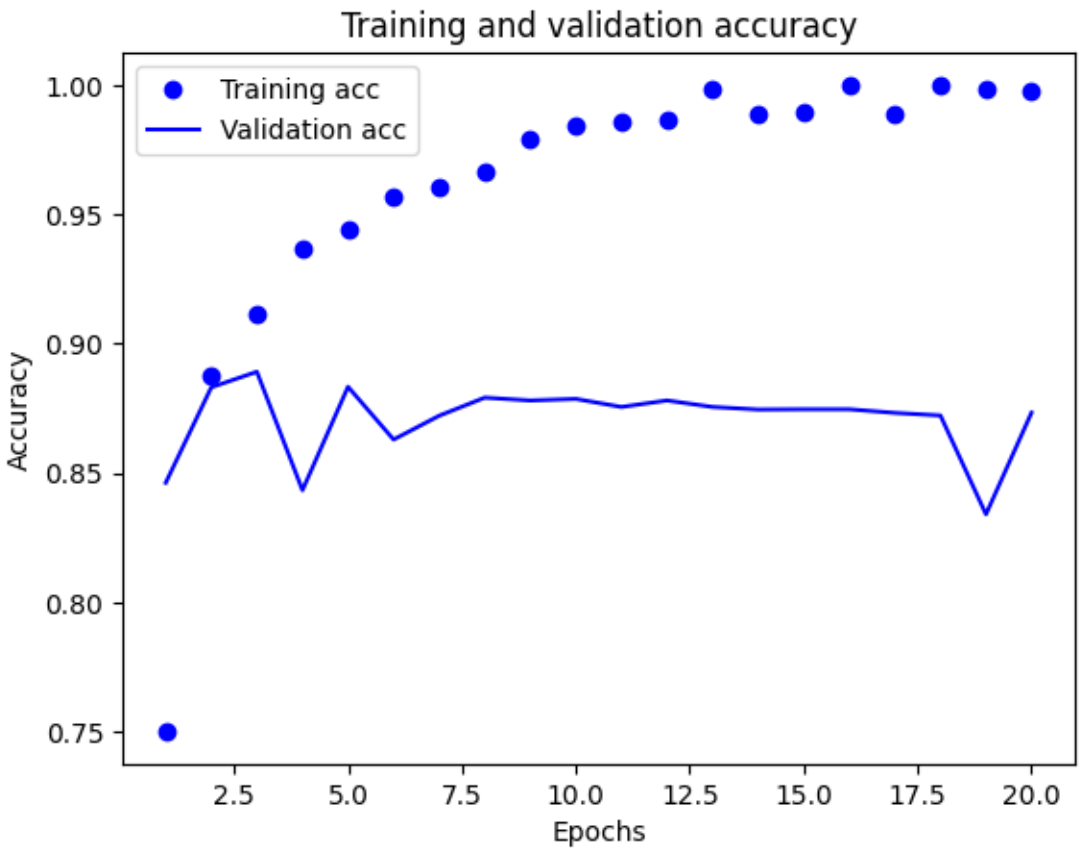
```
Epoch 1/20
30/30 [==============================] - 1s 21ms/step - loss: 0.5141 - accuracy:
0.7501 - val_loss: 0.3797 - val_accuracy: 0.8462
Epoch 2/20
30/30 [==============================] - 0s 15ms/step - loss: 0.3017 - accuracy:
0.8879 - val_loss: 0.3000 - val_accuracy: 0.8831
Epoch 3/20
30/30 [==============================] - 0s 16ms/step - loss: 0.2310 - accuracy:
0.9115 - val_loss: 0.2760 - val_accuracy: 0.8891
Epoch 4/20
30/30 [==============================] - 0s 15ms/step - loss: 0.1783 - accuracy:
0.9369 - val_loss: 0.4063 - val_accuracy: 0.8433
Epoch 5/20
30/30 [==============================] - 0s 13ms/step - loss: 0.1525 - accuracy:
0.9439 - val_loss: 0.2916 - val_accuracy: 0.8833
Epoch 6/20
30/30 [==============================] - 0s 13ms/step - loss: 0.1214 - accuracy:
0.9569 - val_loss: 0.3743 - val_accuracy: 0.8629
Epoch 7/20
30/30 [==============================] - 0s 13ms/step - loss: 0.1061 - accuracy:
0.9607 - val_loss: 0.3623 - val_accuracy: 0.8721
Epoch 8/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0953 - accuracy:
0.9661 - val_loss: 0.3509 - val_accuracy: 0.8791
Epoch 9/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0674 - accuracy:
0.9794 - val_loss: 0.3670 - val_accuracy: 0.8780
Epoch 10/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0560 - accuracy:
0.9845 - val_loss: 0.3925 - val_accuracy: 0.8786
Epoch 11/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0517 - accuracy:
0.9855 - val_loss: 0.4186 - val_accuracy: 0.8755
Epoch 12/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0431 - accuracy:
0.9867 - val_loss: 0.4306 - val_accuracy: 0.8780
Epoch 13/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0151 - accuracy:
0.9987 - val_loss: 0.4787 - val_accuracy: 0.8755
Epoch 14/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0383 - accuracy:
0.9889 - val_loss: 0.4914 - val_accuracy: 0.8745
Epoch 15/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0366 - accuracy:
0.9895 - val_loss: 0.4995 - val_accuracy: 0.8746
```

```
Epoch 16/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0067 - accuracy:
0.9997 - val_loss: 0.5371 - val_accuracy: 0.8746
Epoch 17/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0371 - accuracy:
0.9890 - val_loss: 0.5495 - val_accuracy: 0.8732
Epoch 18/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0042 - accuracy:
0.9999 - val_loss: 0.5839 - val_accuracy: 0.8722
Epoch 19/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0069 - accuracy:
0.9986 - val_loss: 0.9168 - val_accuracy: 0.8340
Epoch 20/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0090 - accuracy:
0.9975 - val_loss: 0.6154 - val_accuracy: 0.8733
```

[31]:
```python
import matplotlib.pyplot as plt
history_dict = history.history
loss_values = history_dict["loss"]
val_loss_values = history_dict["val_loss"]
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, "bo", label="Training loss")
plt.plot(epochs, val_loss_values, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



[32]:
```
plt.clf()
acc = history_dict["accuracy"]
val_acc = history_dict["val_accuracy"]
plt.plot(epochs, acc, "bo", label="Training acc")
plt.plot(epochs, val_acc, "b", label="Validation acc")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```



[33]:
```
# the third one:
#three layers
            #16,16,1 nodes
                        #mse loss function is deployed instead of␣
 ↪binary_crossentropy
model = keras.Sequential([
    layers.Dense(16, activation="relu"),
    layers.Dense(16, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="rmsprop",
              loss="mse",
              metrics=["accuracy"])
```

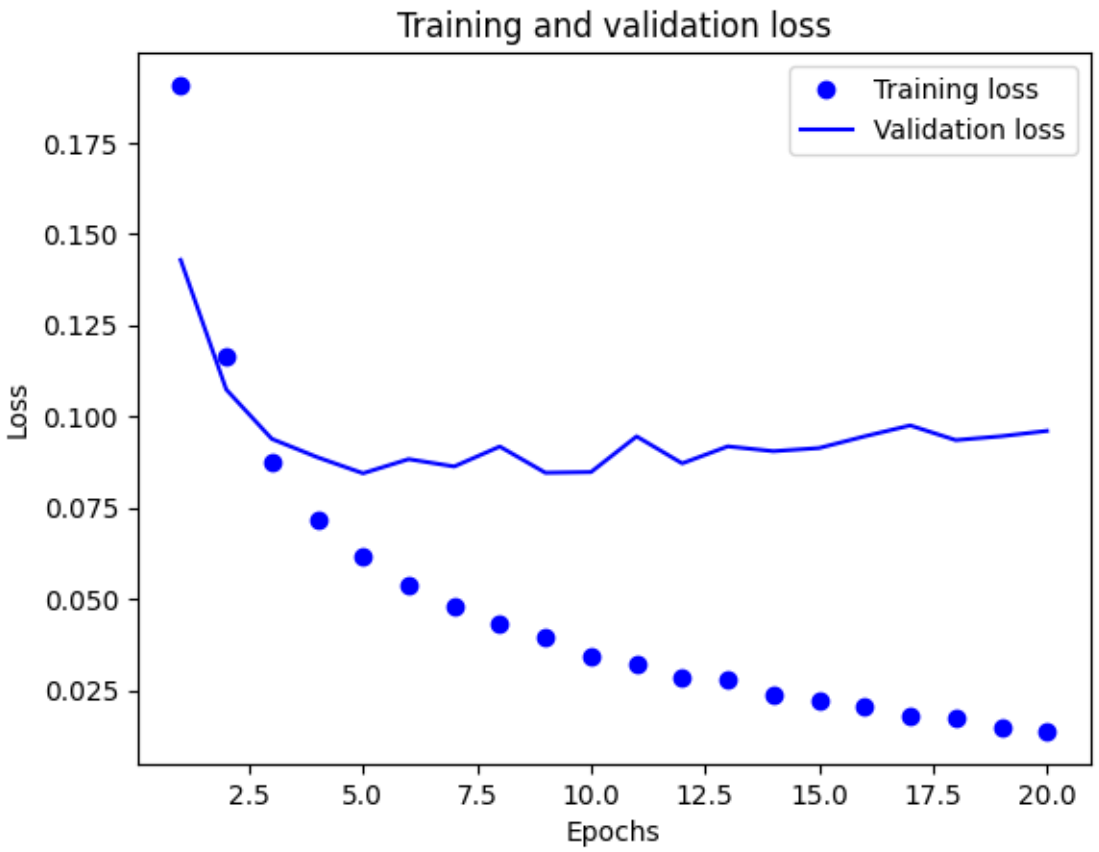
[34]:
```
x_val = x_train[:10000]
partial_x_train = x_train[10000:]
y_val = y_train[:10000]
partial_y_train = y_train[10000:]
```

[35]:
```
history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(x_val, y_val))
```

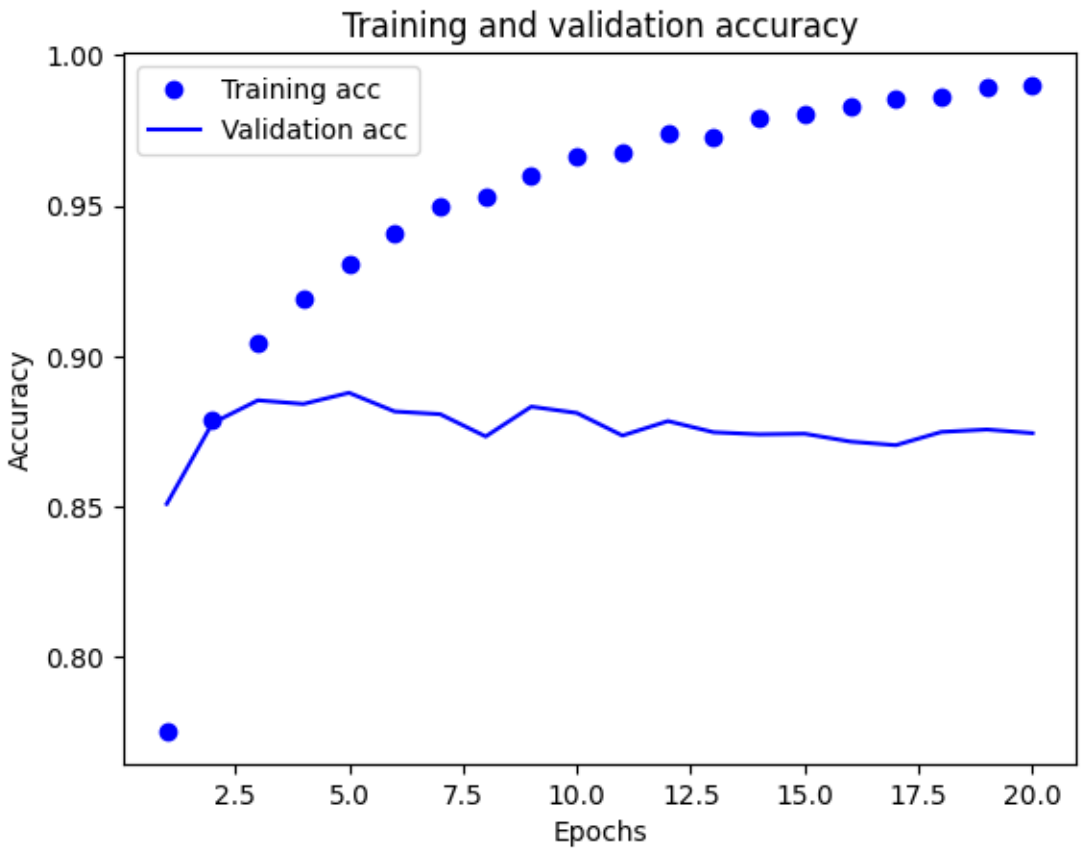
```
Epoch 1/20
30/30 [==============================] - 1s 23ms/step - loss: 0.1907 - accuracy:
0.7754 - val_loss: 0.1429 - val_accuracy: 0.8509
Epoch 2/20
30/30 [==============================] - 0s 14ms/step - loss: 0.1164 - accuracy:
0.8787 - val_loss: 0.1074 - val_accuracy: 0.8777
Epoch 3/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0876 - accuracy:
0.9045 - val_loss: 0.0939 - val_accuracy: 0.8853
Epoch 4/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0716 - accuracy:
0.9193 - val_loss: 0.0888 - val_accuracy: 0.8841
Epoch 5/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0615 - accuracy:
0.9303 - val_loss: 0.0844 - val_accuracy: 0.8878
Epoch 6/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0538 - accuracy:
0.9405 - val_loss: 0.0883 - val_accuracy: 0.8816
Epoch 7/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0478 - accuracy:
0.9500 - val_loss: 0.0863 - val_accuracy: 0.8807
Epoch 8/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0433 - accuracy:
0.9530 - val_loss: 0.0918 - val_accuracy: 0.8733
Epoch 9/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0394 - accuracy:
0.9601 - val_loss: 0.0846 - val_accuracy: 0.8832
Epoch 10/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0345 - accuracy:
0.9664 - val_loss: 0.0848 - val_accuracy: 0.8811
Epoch 11/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0323 - accuracy:
0.9675 - val_loss: 0.0945 - val_accuracy: 0.8736
Epoch 12/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0285 - accuracy:
0.9739 - val_loss: 0.0872 - val_accuracy: 0.8784
Epoch 13/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0278 - accuracy:
0.9729 - val_loss: 0.0918 - val_accuracy: 0.8747
Epoch 14/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0238 - accuracy:
0.9794 - val_loss: 0.0905 - val_accuracy: 0.8740
```

```
Epoch 15/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0223 - accuracy:
0.9802 - val_loss: 0.0914 - val_accuracy: 0.8742
Epoch 16/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0209 - accuracy:
0.9831 - val_loss: 0.0946 - val_accuracy: 0.8716
Epoch 17/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0181 - accuracy:
0.9855 - val_loss: 0.0976 - val_accuracy: 0.8704
Epoch 18/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0175 - accuracy:
0.9862 - val_loss: 0.0935 - val_accuracy: 0.8748
Epoch 19/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0149 - accuracy:
0.9890 - val_loss: 0.0946 - val_accuracy: 0.8756
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0138 - accuracy:
0.9900 - val_loss: 0.0961 - val_accuracy: 0.8744
```

[36]:
```python
import matplotlib.pyplot as plt
history_dict = history.history
loss_values = history_dict["loss"]
val_loss_values = history_dict["val_loss"]
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, "bo", label="Training loss")
plt.plot(epochs, val_loss_values, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



[37]:
```
plt.clf()
acc = history_dict["accuracy"]
val_acc = history_dict["val_accuracy"]
plt.plot(epochs, acc, "bo", label="Training acc")
plt.plot(epochs, val_acc, "b", label="Validation acc")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```



[38]:
```
# the fourth one:
#three layers
                # 64,64,1 nodes
                            #tanh as the replacement of relu
model = keras.Sequential([
    layers.Dense(64, activation="tanh"),
    layers.Dense(64, activation="tanh"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

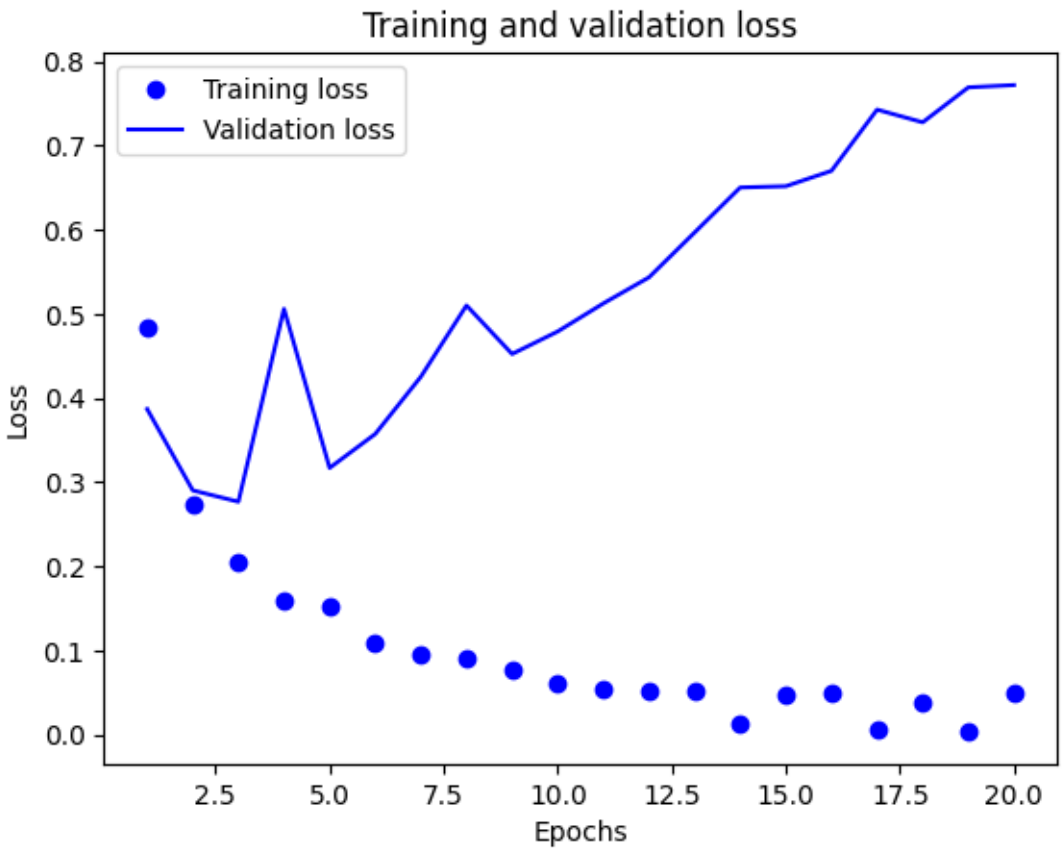
[39]:
```
x_val = x_train[:10000]
partial_x_train = x_train[10000:]
y_val = y_train[:10000]
partial_y_train = y_train[10000:]
```

[40]:
```
history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(x_val, y_val))
```

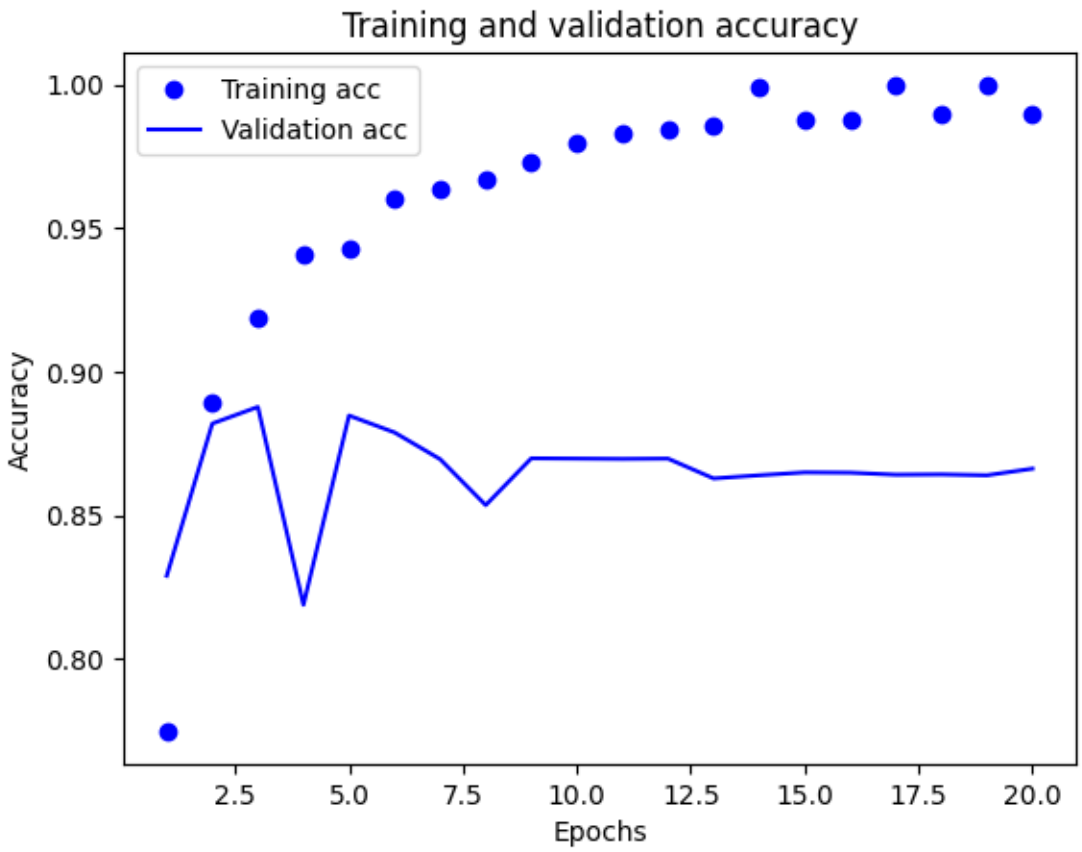
```
Epoch 1/20
30/30 [==============================] - 1s 21ms/step - loss: 0.4835 - accuracy:
0.7748 - val_loss: 0.3868 - val_accuracy: 0.8289
Epoch 2/20
30/30 [==============================] - 0s 13ms/step - loss: 0.2737 - accuracy:
0.8893 - val_loss: 0.2902 - val_accuracy: 0.8819
Epoch 3/20
30/30 [==============================] - 0s 13ms/step - loss: 0.2044 - accuracy:
0.9186 - val_loss: 0.2766 - val_accuracy: 0.8877
Epoch 4/20
30/30 [==============================] - 0s 13ms/step - loss: 0.1588 - accuracy:
0.9411 - val_loss: 0.5059 - val_accuracy: 0.8188
Epoch 5/20
30/30 [==============================] - 0s 13ms/step - loss: 0.1511 - accuracy:
0.9431 - val_loss: 0.3168 - val_accuracy: 0.8847
Epoch 6/20
30/30 [==============================] - 0s 13ms/step - loss: 0.1083 - accuracy:
0.9600 - val_loss: 0.3572 - val_accuracy: 0.8788
Epoch 7/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0959 - accuracy:
0.9633 - val_loss: 0.4252 - val_accuracy: 0.8695
Epoch 8/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0898 - accuracy:
0.9670 - val_loss: 0.5099 - val_accuracy: 0.8535
Epoch 9/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0775 - accuracy:
0.9730 - val_loss: 0.4522 - val_accuracy: 0.8698
Epoch 10/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0608 - accuracy:
0.9795 - val_loss: 0.4790 - val_accuracy: 0.8697
Epoch 11/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0530 - accuracy:
0.9832 - val_loss: 0.5123 - val_accuracy: 0.8696
Epoch 12/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0526 - accuracy:
0.9845 - val_loss: 0.5433 - val_accuracy: 0.8697
Epoch 13/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0510 - accuracy:
0.9858 - val_loss: 0.5966 - val_accuracy: 0.8628
Epoch 14/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0122 - accuracy:
0.9991 - val_loss: 0.6501 - val_accuracy: 0.8639
```

```
Epoch 15/20
30/30 [==============================] - 0s 15ms/step - loss: 0.0464 - accuracy:
0.9877 - val_loss: 0.6516 - val_accuracy: 0.8649
Epoch 16/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0499 - accuracy:
0.9879 - val_loss: 0.6699 - val_accuracy: 0.8648
Epoch 17/20
30/30 [==============================] - 0s 15ms/step - loss: 0.0059 - accuracy:
0.9997 - val_loss: 0.7426 - val_accuracy: 0.8641
Epoch 18/20
30/30 [==============================] - 0s 15ms/step - loss: 0.0386 - accuracy:
0.9899 - val_loss: 0.7275 - val_accuracy: 0.8642
Epoch 19/20
30/30 [==============================] - 0s 15ms/step - loss: 0.0039 - accuracy:
0.9997 - val_loss: 0.7691 - val_accuracy: 0.8639
Epoch 20/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0491 - accuracy:
0.9893 - val_loss: 0.7718 - val_accuracy: 0.8662
```

[41]:
```python
import matplotlib.pyplot as plt
history_dict = history.history
loss_values = history_dict["loss"]
val_loss_values = history_dict["val_loss"]
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, "bo", label="Training loss")
plt.plot(epochs, val_loss_values, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



[42]:
```
plt.clf()
acc = history_dict["accuracy"]
val_acc = history_dict["val_accuracy"]
plt.plot(epochs, acc, "bo", label="Training acc")
plt.plot(epochs, val_acc, "b", label="Validation acc")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```



[ ]:

[44]:
```
#model wuth dropout
from tensorflow.keras.datasets import imdb
(train_data, train_labels), _ = imdb.load_data(num_words=10000)

def vectorize_sequences(sequences, dimension=10000):
    results = np.zeros((len(sequences), dimension))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1.
    return results
train_data = vectorize_sequences(train_data)

model = keras.Sequential([
    layers.Dense(16, activation="relu"),
    layers.Dropout(0.5),
    layers.Dense(16, activation="relu"),
    layers.Dropout(0.5),
    layers.Dense(1, activation="sigmoid")
])
```

```
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
history_dropout = model.fit(
    train_data, train_labels,
    epochs=20, batch_size=512, validation_split=0.4)
```

```
Epoch 1/20
30/30 [==============================] - 1s 22ms/step - loss: 0.6486 - accuracy:
0.6207 - val_loss: 0.5468 - val_accuracy: 0.8367
Epoch 2/20
30/30 [==============================] - 0s 10ms/step - loss: 0.5330 - accuracy:
0.7527 - val_loss: 0.4238 - val_accuracy: 0.8657
Epoch 3/20
30/30 [==============================] - 0s 10ms/step - loss: 0.4449 - accuracy:
0.8182 - val_loss: 0.3536 - val_accuracy: 0.8810
Epoch 4/20
30/30 [==============================] - 0s 11ms/step - loss: 0.3787 - accuracy:
0.8521 - val_loss: 0.3158 - val_accuracy: 0.8823
Epoch 5/20
30/30 [==============================] - 0s 10ms/step - loss: 0.3244 - accuracy:
0.8773 - val_loss: 0.2850 - val_accuracy: 0.8892
Epoch 6/20
30/30 [==============================] - 0s 11ms/step - loss: 0.2843 - accuracy:
0.8997 - val_loss: 0.2758 - val_accuracy: 0.8903
Epoch 7/20
30/30 [==============================] - 0s 10ms/step - loss: 0.2501 - accuracy:
0.9135 - val_loss: 0.2757 - val_accuracy: 0.8915
Epoch 8/20
30/30 [==============================] - 0s 10ms/step - loss: 0.2255 - accuracy:
0.9235 - val_loss: 0.2862 - val_accuracy: 0.8856
Epoch 9/20
30/30 [==============================] - 0s 9ms/step - loss: 0.2005 - accuracy:
0.9333 - val_loss: 0.3037 - val_accuracy: 0.8894
Epoch 10/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1796 - accuracy:
0.9410 - val_loss: 0.3059 - val_accuracy: 0.8869
Epoch 11/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1675 - accuracy:
0.9461 - val_loss: 0.3341 - val_accuracy: 0.8868
Epoch 12/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1422 - accuracy:
0.9534 - val_loss: 0.3326 - val_accuracy: 0.8887
Epoch 13/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1359 - accuracy:
0.9537 - val_loss: 0.3706 - val_accuracy: 0.8863
Epoch 14/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1206 - accuracy:
```

```
0.9621 - val_loss: 0.3803 - val_accuracy: 0.8868
Epoch 15/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1131 - accuracy:
0.9640 - val_loss: 0.4075 - val_accuracy: 0.8845
Epoch 16/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1037 - accuracy:
0.9664 - val_loss: 0.4222 - val_accuracy: 0.8858
Epoch 17/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0960 - accuracy:
0.9701 - val_loss: 0.4329 - val_accuracy: 0.8834
Epoch 18/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0871 - accuracy:
0.9710 - val_loss: 0.4815 - val_accuracy: 0.8832
Epoch 19/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0832 - accuracy:
0.9715 - val_loss: 0.5092 - val_accuracy: 0.8840
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0816 - accuracy:
0.9730 - val_loss: 0.5392 - val_accuracy: 0.8830
```

[ ]:
```
#I dont know how to plot so I plot using the above values in excel
```