Assignment 5

April 16, 2023

Gabriel Avinaz

Week 5

4/16/23

0.0.1 Assignment 5.1

Section 3.4

[ ]:
```
from keras.datasets import imdb

(train_data, train_labels), (test_data, test_labels) = imdb.
 ↪load_data(num_words=10000)

print(train_data[0])
print(train_labels[0])
```

```
[1, 14, 22, 16, 43, 530, 973, 1622, 1385, 65, 458, 4468, 66, 3941, 4, 173, 36,
256, 5, 25, 100, 43, 838, 112, 50, 670, 2, 9, 35, 480, 284, 5, 150, 4, 172, 112,
167, 2, 336, 385, 39, 4, 172, 4536, 1111, 17, 546, 38, 13, 447, 4, 192, 50, 16,
6, 147, 2025, 19, 14, 22, 4, 1920, 4613, 469, 4, 22, 71, 87, 12, 16, 43, 530,
38, 76, 15, 13, 1247, 4, 22, 17, 515, 17, 12, 16, 626, 18, 2, 5, 62, 386, 12, 8,
316, 8, 106, 5, 4, 2223, 5244, 16, 480, 66, 3785, 33, 4, 130, 12, 16, 38, 619,
5, 25, 124, 51, 36, 135, 48, 25, 1415, 33, 6, 22, 12, 215, 28, 77, 52, 5, 14,
407, 16, 82, 2, 8, 4, 107, 117, 5952, 15, 256, 4, 2, 7, 3766, 5, 723, 36, 71,
43, 530, 476, 26, 400, 317, 46, 7, 4, 2, 1029, 13, 104, 88, 4, 381, 15, 297, 98,
32, 2071, 56, 26, 141, 6, 194, 7486, 18, 4, 226, 22, 21, 134, 476, 26, 480, 5,
144, 30, 5535, 18, 51, 36, 28, 224, 92, 25, 104, 4, 226, 65, 16, 38, 1334, 88,
12, 16, 283, 5, 16, 4472, 113, 103, 32, 15, 16, 5345, 19, 178, 32]
1
```

[ ]:
```
max([max(sequence) for sequence in train_data])
```

[ ]: 9999

[ ]:
```
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]])
decoded_review
```

[ ]: "? this film was just brilliant casting location scenery story direction everyone's really suited the part they played and you could just imagine being there robert ? is an amazing actor and now the same being director ? father came from the same scottish island as myself so i loved the fact there was a real connection with this film the witty remarks throughout the film were great it was just brilliant so much that i bought the film as soon as it was released for ? and would recommend it to everyone to watch and the fly fishing was amazing really cried at the end it was so sad and you know what they say if you cry at a film it must have been good and this definitely was also ? to the two little boy's that played the ? of norman and paul they were just brilliant children are often left out of the ? list i think because the stars that play them all grown up are such a big profile for the whole film but these children are amazing and should be praised for what they have done don't you think the whole story was so lovely because it was true and was someone's life after all that was shared with us all"

[ ]:
```
import numpy as np

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

x_train = vectorize_sequences(train_data)
x_test = vectorize_sequences(test_data)

x_train[0]
```

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

[ ]:
```
y_train = np.array(train_labels).astype('float32')
y_test = np.array(test_labels).astype('float32')
```

[ ]:
```
from keras import models, layers

model = models.Sequential()
model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model.add(layers.Dense(16,activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer='rmsprop', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])
```

[ ]:
```
from keras import optimizers

model.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
 ↪loss='binary_crossentropy', metrics=['accuracy'])
```

[ ]:
```
from keras import losses, metrics

model.compile(optimizer=optimizers.RMSprop(learning_rate=0.001), loss=losses.
 ↪binary_crossentropy, metrics=[metrics.binary_accuracy])
```

[ ]:
```
x_val = x_train[:10000]
partial_x_train = x_train[10000:]

y_val = y_train[:10000]
partial_y_train = y_train[10000:]
```

[ ]:
```
model.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
 ↪loss='binary_crossentropy', metrics=['acc'])

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 25ms/step - loss: 0.5059 - acc:
0.7801 - val_loss: 0.4060 - val_acc: 0.8346
Epoch 2/20
30/30 [==============================] - 0s 8ms/step - loss: 0.2931 - acc:
0.9004 - val_loss: 0.2938 - val_acc: 0.8898
Epoch 3/20
30/30 [==============================] - 0s 8ms/step - loss: 0.2100 - acc:
0.9312 - val_loss: 0.2782 - val_acc: 0.8903
Epoch 4/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1711 - acc:
0.9435 - val_loss: 0.2757 - val_acc: 0.8907
Epoch 5/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1372 - acc:
0.9559 - val_loss: 0.2874 - val_acc: 0.8853
Epoch 6/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1121 - acc:
0.9660 - val_loss: 0.3008 - val_acc: 0.8836
Epoch 7/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0950 - acc:
0.9721 - val_loss: 0.3478 - val_acc: 0.8774
Epoch 8/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0781 - acc:
0.9782 - val_loss: 0.3413 - val_acc: 0.8811
Epoch 9/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0638 - acc:
```

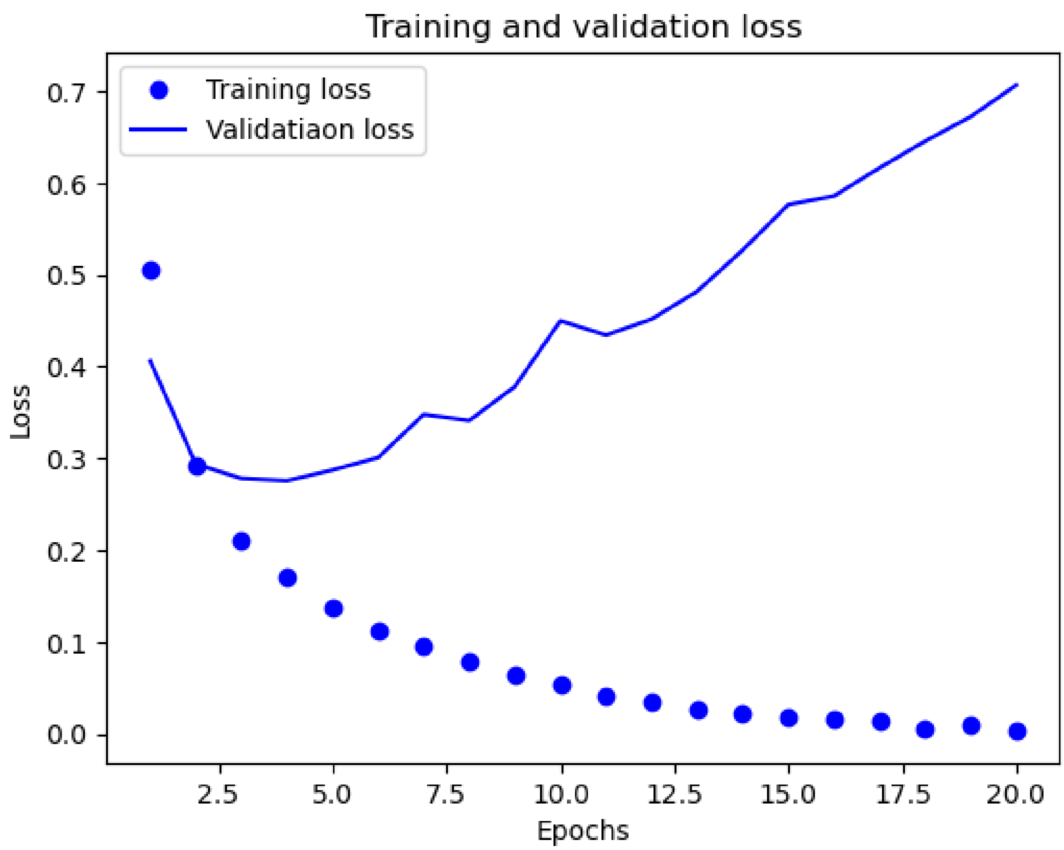
```
0.9821 - val_loss: 0.3782 - val_acc: 0.8748
Epoch 10/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0529 - acc:
0.9867 - val_loss: 0.4498 - val_acc: 0.8604
Epoch 11/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0421 - acc:
0.9903 - val_loss: 0.4342 - val_acc: 0.8742
Epoch 12/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0348 - acc:
0.9924 - val_loss: 0.4516 - val_acc: 0.8723
Epoch 13/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0268 - acc:
0.9954 - val_loss: 0.4818 - val_acc: 0.8723
Epoch 14/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0223 - acc:
0.9957 - val_loss: 0.5269 - val_acc: 0.8684
Epoch 15/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0188 - acc:
0.9971 - val_loss: 0.5763 - val_acc: 0.8624
Epoch 16/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0163 - acc:
0.9970 - val_loss: 0.5856 - val_acc: 0.8664
Epoch 17/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0135 - acc:
0.9971 - val_loss: 0.6163 - val_acc: 0.8656
Epoch 18/20
30/30 [==============================] - 0s 7ms/step - loss: 0.0064 - acc:
0.9996 - val_loss: 0.6453 - val_acc: 0.8666
Epoch 19/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0109 - acc:
0.9973 - val_loss: 0.6720 - val_acc: 0.8641
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0039 - acc:
0.9999 - val_loss: 0.7064 - val_acc: 0.8635
```

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

[ ]:
```
dict_keys(['loss', 'acc', 'val_loss', 'val_acc'])
```

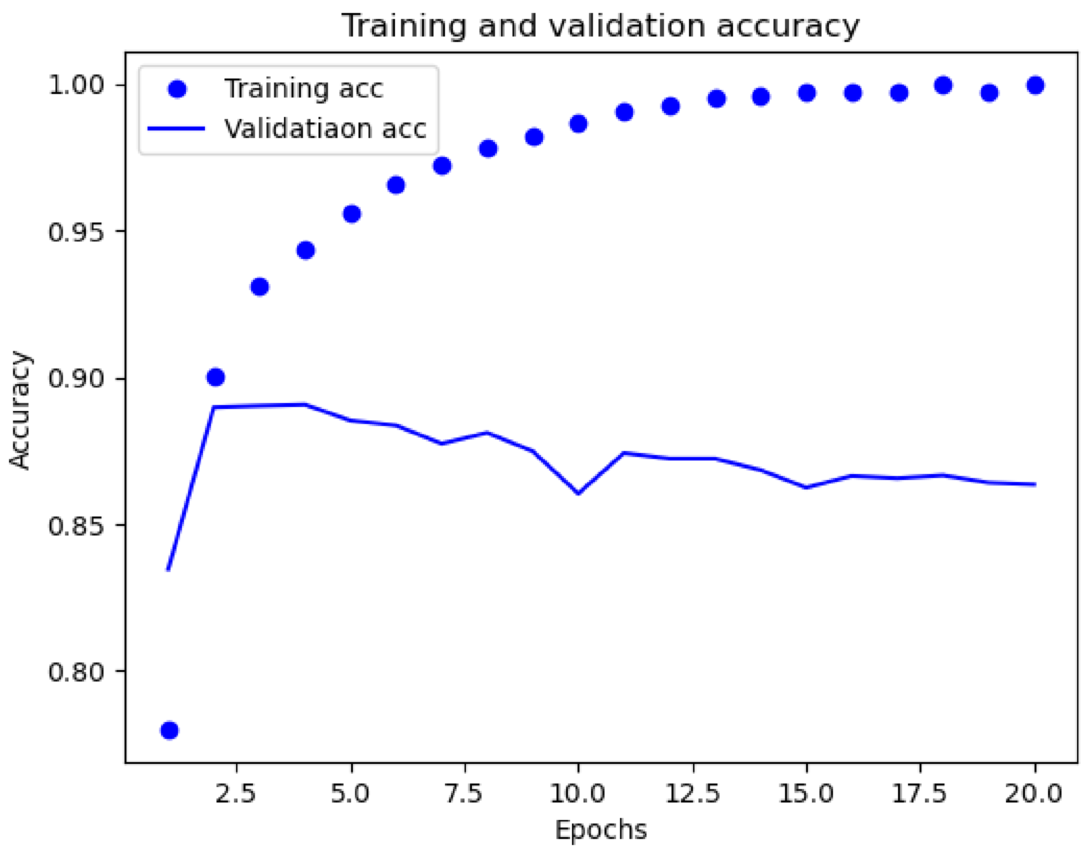
[ ]:
```
import matplotlib.pyplot as plt

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="Validatiaon loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

acc = history_dict['acc']
val_acc = history_dict['val_acc']

plt.plot(epochs, acc, 'bo', label="Training acc")
plt.plot(epochs, val_acc, 'b', label="Validatiaon acc")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
```

```
plt.show()
```



[ ]:
```
model = models.Sequential()
model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model.add(layers.Dense(16,activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
↪loss='binary_crossentropy', metrics=['accuracy'])

model.fit(x_train, y_train, epochs=4, batch_size=512)
results = model.evaluate(x_test, y_test)
```

```
Epoch 1/4
49/49 [==============================] - 1s 4ms/step - loss: 0.4761 - accuracy:
0.8186
Epoch 2/4
49/49 [==============================] - 0s 5ms/step - loss: 0.2718 - accuracy:
0.9066
Epoch 3/4
```

```
49/49 [==============================] - 0s 4ms/step - loss: 0.2069 - accuracy:
0.9255
Epoch 4/4
49/49 [==============================] - 0s 4ms/step - loss: 0.1716 - accuracy:
0.9408
782/782 [==============================] - 13s 17ms/step - loss: 0.2972 -
accuracy: 0.8820
```

[ ]:
```
results
```

[ ]:
```
[0.2972390949726105, 0.8819599747657776]
```

[ ]:
```
model.predict(x_test)
```

```
782/782 [==============================] - 23s 29ms/step
```

[ ]:
```
array([[0.14347336],
       [0.9999418 ],
       [0.831859  ],
       …,
       [0.10431504],
       [0.06340731],
       [0.41998455]], dtype=float32)
```

[ ]:
```
model_1 = models.Sequential()
model_1.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model_1.add(layers.Dense(1, activation='sigmoid'))

model_1.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
 ↪loss='binary_crossentropy', metrics=['acc'])

history_1 = model_1.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))

model_2 = models.Sequential()
model_2.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model_2.add(layers.Dense(16,activation='relu'))
model_2.add(layers.Dense(1, activation='sigmoid'))

model_2.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
 ↪loss='binary_crossentropy', metrics=['acc'])

history_2 = model_2.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))

model_3 = models.Sequential()
model_3.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model_3.add(layers.Dense(16,activation='relu'))
```

```
model_3.add(layers.Dense(16,activation='relu'))
model_3.add(layers.Dense(1, activation='sigmoid'))

model_3.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
↪loss='binary_crossentropy', metrics=['acc'])

history_3 = model_3.fit(partial_x_train, partial_y_train, epochs=20,␣
↪batch_size=512, validation_data=(x_val, y_val))


history_dict_1 = history_1.history
history_dict_2 = history_2.history
history_dict_3 = history_3.history
```

```
Epoch 1/20
30/30 [==============================] - 1s 23ms/step - loss: 0.4982 - acc:
0.7992 - val_loss: 0.3863 - val_acc: 0.8670
Epoch 2/20
30/30 [==============================] - 0s 8ms/step - loss: 0.3117 - acc:
0.9046 - val_loss: 0.3310 - val_acc: 0.8749
Epoch 3/20
30/30 [==============================] - 0s 9ms/step - loss: 0.2413 - acc:
0.9259 - val_loss: 0.2946 - val_acc: 0.8863
Epoch 4/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1987 - acc:
0.9387 - val_loss: 0.2797 - val_acc: 0.8884
Epoch 5/20
30/30 [==============================] - 0s 10ms/step - loss: 0.1677 - acc:
0.9507 - val_loss: 0.2744 - val_acc: 0.8907
Epoch 6/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1465 - acc:
0.9571 - val_loss: 0.2778 - val_acc: 0.8879
Epoch 7/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1279 - acc:
0.9637 - val_loss: 0.2870 - val_acc: 0.8850
Epoch 8/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1119 - acc:
0.9695 - val_loss: 0.2868 - val_acc: 0.8862
Epoch 9/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0990 - acc:
0.9739 - val_loss: 0.2961 - val_acc: 0.8855
Epoch 10/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0872 - acc:
0.9779 - val_loss: 0.3209 - val_acc: 0.8784
Epoch 11/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0779 - acc:
0.9812 - val_loss: 0.3211 - val_acc: 0.8834
```

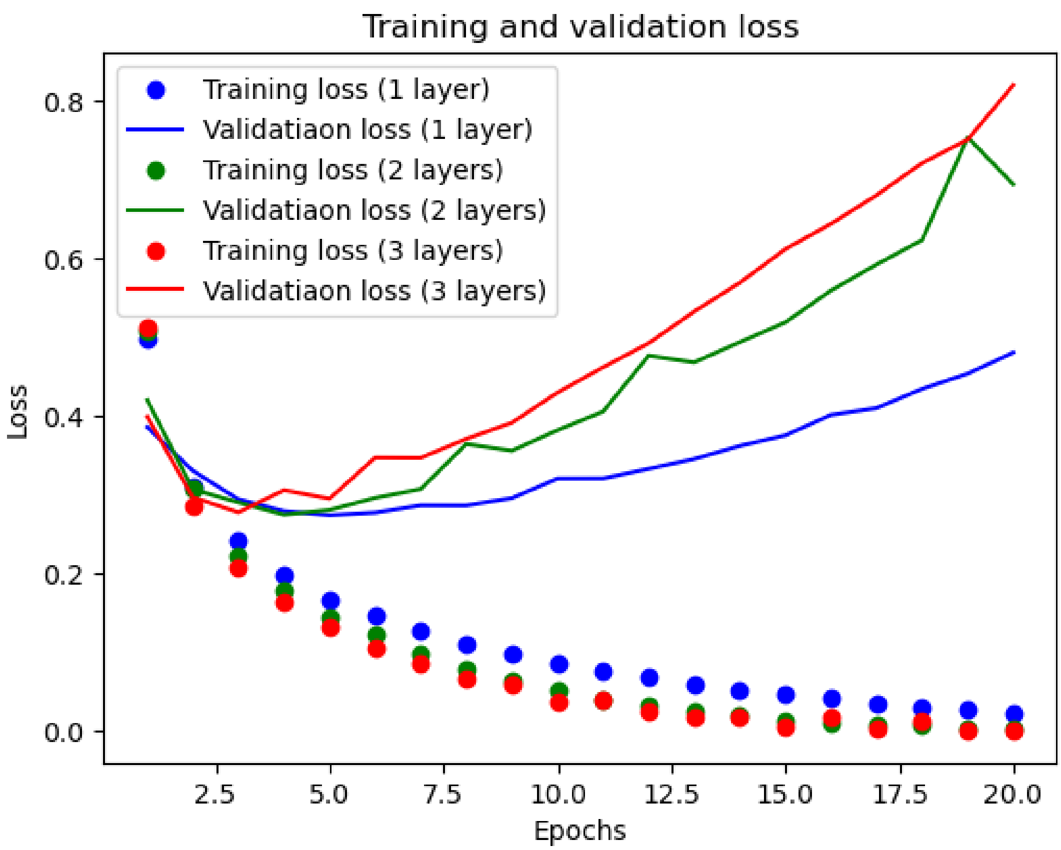
```
Epoch 12/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0688 - acc:
0.9840 - val_loss: 0.3333 - val_acc: 0.8800
Epoch 13/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0609 - acc:
0.9873 - val_loss: 0.3458 - val_acc: 0.8806
Epoch 14/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0531 - acc:
0.9898 - val_loss: 0.3624 - val_acc: 0.8772
Epoch 15/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0469 - acc:
0.9906 - val_loss: 0.3758 - val_acc: 0.8775
Epoch 16/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0418 - acc:
0.9923 - val_loss: 0.4018 - val_acc: 0.8776
Epoch 17/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0362 - acc:
0.9941 - val_loss: 0.4104 - val_acc: 0.8755
Epoch 18/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0313 - acc:
0.9955 - val_loss: 0.4348 - val_acc: 0.8759
Epoch 19/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0282 - acc:
0.9963 - val_loss: 0.4541 - val_acc: 0.8682
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0237 - acc:
0.9974 - val_loss: 0.4809 - val_acc: 0.8660
Epoch 1/20
30/30 [==============================] - 1s 21ms/step - loss: 0.5076 - acc:
0.7899 - val_loss: 0.4205 - val_acc: 0.8364
Epoch 2/20
30/30 [==============================] - 0s 13ms/step - loss: 0.3086 - acc:
0.9053 - val_loss: 0.3073 - val_acc: 0.8865
Epoch 3/20
30/30 [==============================] - 0s 9ms/step - loss: 0.2239 - acc:
0.9297 - val_loss: 0.2907 - val_acc: 0.8836
Epoch 4/20
30/30 [==============================] - 0s 12ms/step - loss: 0.1794 - acc:
0.9413 - val_loss: 0.2752 - val_acc: 0.8899
Epoch 5/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1457 - acc:
0.9541 - val_loss: 0.2812 - val_acc: 0.8854
Epoch 6/20
30/30 [==============================] - 0s 11ms/step - loss: 0.1222 - acc:
0.9620 - val_loss: 0.2964 - val_acc: 0.8858
Epoch 7/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0980 - acc:
0.9723 - val_loss: 0.3076 - val_acc: 0.8833
```

```
Epoch 8/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0790 - acc:
0.9797 - val_loss: 0.3648 - val_acc: 0.8725
Epoch 9/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0652 - acc:
0.9833 - val_loss: 0.3563 - val_acc: 0.8743
Epoch 10/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0536 - acc:
0.9874 - val_loss: 0.3824 - val_acc: 0.8755
Epoch 11/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0399 - acc:
0.9925 - val_loss: 0.4060 - val_acc: 0.8753
Epoch 12/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0328 - acc:
0.9942 - val_loss: 0.4770 - val_acc: 0.8675
Epoch 13/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0266 - acc:
0.9951 - val_loss: 0.4685 - val_acc: 0.8727
Epoch 14/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0214 - acc:
0.9960 - val_loss: 0.4941 - val_acc: 0.8730
Epoch 15/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0143 - acc:
0.9985 - val_loss: 0.5193 - val_acc: 0.8739
Epoch 16/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0104 - acc:
0.9994 - val_loss: 0.5596 - val_acc: 0.8710
Epoch 17/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0094 - acc:
0.9990 - val_loss: 0.5927 - val_acc: 0.8700
Epoch 18/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0080 - acc:
0.9984 - val_loss: 0.6236 - val_acc: 0.8688
Epoch 19/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0039 - acc:
0.9999 - val_loss: 0.7541 - val_acc: 0.8523
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0037 - acc:
0.9998 - val_loss: 0.6944 - val_acc: 0.8665
Epoch 1/20
30/30 [==============================] - 1s 21ms/step - loss: 0.5138 - acc:
0.7727 - val_loss: 0.3995 - val_acc: 0.8425
Epoch 2/20
30/30 [==============================] - 0s 9ms/step - loss: 0.2874 - acc:
0.9047 - val_loss: 0.2972 - val_acc: 0.8850
Epoch 3/20
30/30 [==============================] - 0s 8ms/step - loss: 0.2091 - acc:
0.9299 - val_loss: 0.2782 - val_acc: 0.8874
```

```
Epoch 4/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1639 - acc:
0.9435 - val_loss: 0.3063 - val_acc: 0.8803
Epoch 5/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1328 - acc:
0.9549 - val_loss: 0.2955 - val_acc: 0.8856
Epoch 6/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1061 - acc:
0.9647 - val_loss: 0.3476 - val_acc: 0.8760
Epoch 7/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0856 - acc:
0.9721 - val_loss: 0.3473 - val_acc: 0.8772
Epoch 8/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0660 - acc:
0.9817 - val_loss: 0.3713 - val_acc: 0.8747
Epoch 9/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0602 - acc:
0.9827 - val_loss: 0.3919 - val_acc: 0.8730
Epoch 10/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0379 - acc:
0.9913 - val_loss: 0.4301 - val_acc: 0.8778
Epoch 11/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0414 - acc:
0.9884 - val_loss: 0.4622 - val_acc: 0.8716
Epoch 12/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0247 - acc:
0.9949 - val_loss: 0.4927 - val_acc: 0.8737
Epoch 13/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0189 - acc:
0.9961 - val_loss: 0.5330 - val_acc: 0.8731
Epoch 14/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0185 - acc:
0.9957 - val_loss: 0.5695 - val_acc: 0.8722
Epoch 15/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0069 - acc:
0.9996 - val_loss: 0.6122 - val_acc: 0.8703
Epoch 16/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0188 - acc:
0.9937 - val_loss: 0.6443 - val_acc: 0.8695
Epoch 17/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0034 - acc:
0.9998 - val_loss: 0.6806 - val_acc: 0.8680
Epoch 18/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0135 - acc:
0.9959 - val_loss: 0.7214 - val_acc: 0.8669
Epoch 19/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0018 - acc:
0.9999 - val_loss: 0.7512 - val_acc: 0.8665
```

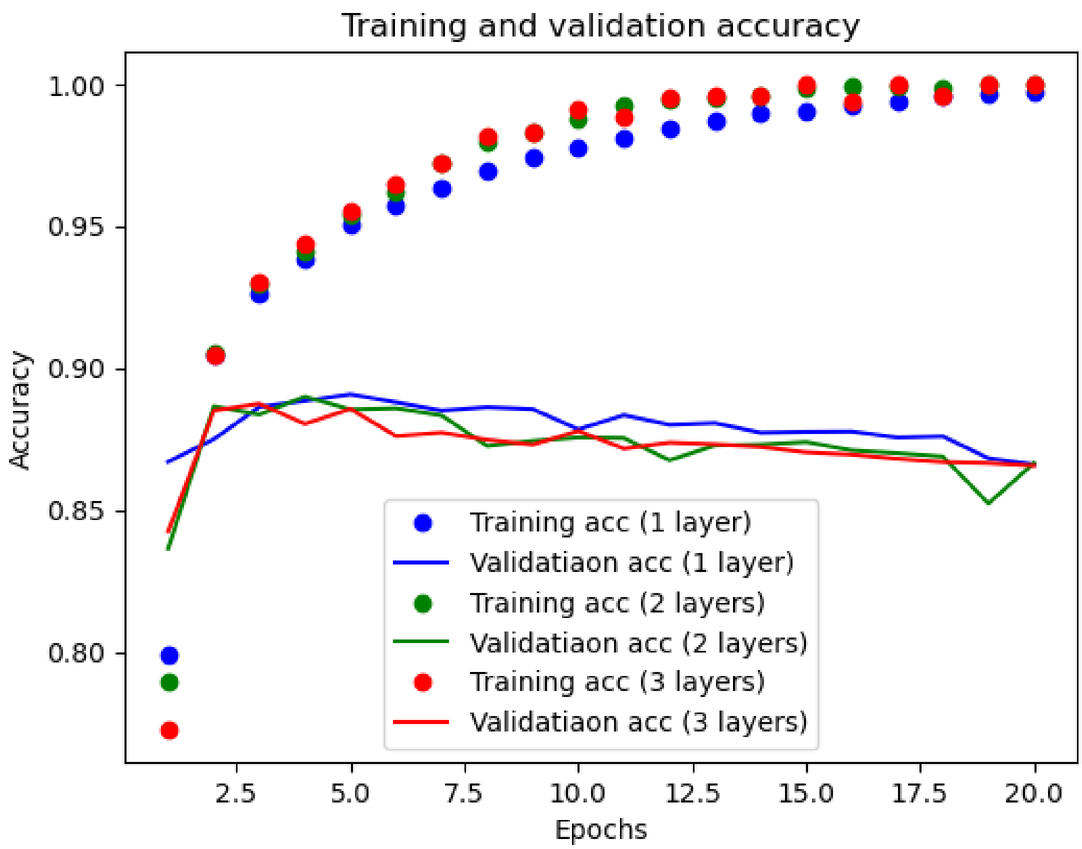
```
Epoch 20/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0014 - acc:
0.9999 - val_loss: 0.8203 - val_acc: 0.8656
```

[ ]:
```
loss_values_1 = history_dict_1['loss']
val_loss_values_1 = history_dict_1['val_loss']

loss_values_2 = history_dict_2['loss']
val_loss_values_2 = history_dict_2['val_loss']

loss_values_3 = history_dict_3['loss']
val_loss_values_3 = history_dict_3['val_loss']

epochs = range(1, len(loss_values_1) + 1)

plt.plot(epochs,loss_values_1, 'bo', label="Training loss (1 layer)")
plt.plot(epochs,val_loss_values_1, 'b', label="Validatiaon loss (1 layer)")
plt.plot(epochs,loss_values_2, 'go', label="Training loss (2 layers)")
plt.plot(epochs,val_loss_values_2, 'g', label="Validatiaon loss (2 layers)")
plt.plot(epochs,loss_values_3, 'ro', label="Training loss (3 layers)")
plt.plot(epochs,val_loss_values_3, 'r', label="Validatiaon loss (3 layers)")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

acc_1 = history_dict_1['acc']
val_acc_1 = history_dict_1['val_acc']
acc_2 = history_dict_2['acc']
val_acc_2 = history_dict_2['val_acc']
acc_3 = history_dict_3['acc']
val_acc_3 = history_dict_3['val_acc']

plt.plot(epochs, acc_1, 'bo', label="Training acc (1 layer)")
plt.plot(epochs, val_acc_1, 'b', label="Validatiaon acc (1 layer)")
plt.plot(epochs, acc_2, 'go', label="Training acc (2 layers)")
plt.plot(epochs, val_acc_2, 'g', label="Validatiaon acc (2 layers)")
plt.plot(epochs, acc_3, 'ro', label="Training acc (3 layers)")
plt.plot(epochs, val_acc_3, 'r', label="Validatiaon acc (3 layers)")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
```

```
plt.show()
```



[ ]:
```
model_16 = models.Sequential()
model_16.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model_16.add(layers.Dense(16,activation='relu'))
model_16.add(layers.Dense(1, activation='sigmoid'))

model_16.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
↪loss='binary_crossentropy', metrics=['acc'])

history_16 = model_16.fit(partial_x_train, partial_y_train, epochs=20,␣
↪batch_size=512, validation_data=(x_val, y_val))

model_32 = models.Sequential()
model_32.add(layers.Dense(32, activation='relu', input_shape=(10000,)))
model_32.add(layers.Dense(32,activation='relu'))
model_32.add(layers.Dense(1, activation='sigmoid'))
```

```python
model_32.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
 ↪loss='binary_crossentropy', metrics=['acc'])

history_32 = model_32.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))

model_64 = models.Sequential()
model_64.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
model_64.add(layers.Dense(64,activation='relu'))
model_64.add(layers.Dense(1, activation='sigmoid'))

model_64.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
 ↪loss='binary_crossentropy', metrics=['acc'])

history_64 = model_64.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))


history_dict_16 = history_16.history
history_dict_32 = history_32.history
history_dict_64 = history_64.history
```

```
Epoch 1/20
30/30 [==============================] - 1s 23ms/step - loss: 0.5080 - acc:
0.7830 - val_loss: 0.3770 - val_acc: 0.8749
Epoch 2/20
30/30 [==============================] - 0s 8ms/step - loss: 0.2946 - acc:
0.9065 - val_loss: 0.3050 - val_acc: 0.8842
Epoch 3/20
30/30 [==============================] - 0s 8ms/step - loss: 0.2168 - acc:
0.9287 - val_loss: 0.2970 - val_acc: 0.8820
Epoch 4/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1695 - acc:
0.9469 - val_loss: 0.2983 - val_acc: 0.8816
Epoch 5/20
30/30 [==============================] - 0s 11ms/step - loss: 0.1398 - acc:
0.9544 - val_loss: 0.3147 - val_acc: 0.8768
Epoch 6/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1183 - acc:
0.9631 - val_loss: 0.2979 - val_acc: 0.8867
Epoch 7/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0956 - acc:
0.9721 - val_loss: 0.3242 - val_acc: 0.8800
Epoch 8/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0804 - acc:
0.9772 - val_loss: 0.3326 - val_acc: 0.8812
Epoch 9/20
```

```
30/30 [==============================] - 0s 8ms/step - loss: 0.0654 - acc:
0.9830 - val_loss: 0.3675 - val_acc: 0.8797
Epoch 10/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0543 - acc:
0.9861 - val_loss: 0.3829 - val_acc: 0.8758
Epoch 11/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0427 - acc:
0.9906 - val_loss: 0.4250 - val_acc: 0.8745
Epoch 12/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0340 - acc:
0.9923 - val_loss: 0.5166 - val_acc: 0.8568
Epoch 13/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0297 - acc:
0.9943 - val_loss: 0.4668 - val_acc: 0.8736
Epoch 14/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0225 - acc:
0.9957 - val_loss: 0.5526 - val_acc: 0.8604
Epoch 15/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0169 - acc:
0.9982 - val_loss: 0.5332 - val_acc: 0.8719
Epoch 16/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0156 - acc:
0.9972 - val_loss: 0.5681 - val_acc: 0.8694
Epoch 17/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0087 - acc:
0.9995 - val_loss: 0.6150 - val_acc: 0.8642
Epoch 18/20
30/30 [==============================] - 0s 15ms/step - loss: 0.0089 - acc:
0.9991 - val_loss: 0.6401 - val_acc: 0.8671
Epoch 19/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0080 - acc:
0.9986 - val_loss: 0.6718 - val_acc: 0.8668
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0038 - acc:
0.9999 - val_loss: 0.7107 - val_acc: 0.8648
Epoch 1/20
30/30 [==============================] - 2s 25ms/step - loss: 0.4862 - acc:
0.7860 - val_loss: 0.3670 - val_acc: 0.8567
Epoch 2/20
30/30 [==============================] - 0s 10ms/step - loss: 0.2715 - acc:
0.9062 - val_loss: 0.3129 - val_acc: 0.8723
Epoch 3/20
30/30 [==============================] - 0s 13ms/step - loss: 0.1982 - acc:
0.9309 - val_loss: 0.2736 - val_acc: 0.8914
Epoch 4/20
30/30 [==============================] - 0s 10ms/step - loss: 0.1526 - acc:
0.9485 - val_loss: 0.2822 - val_acc: 0.8884
Epoch 5/20
```

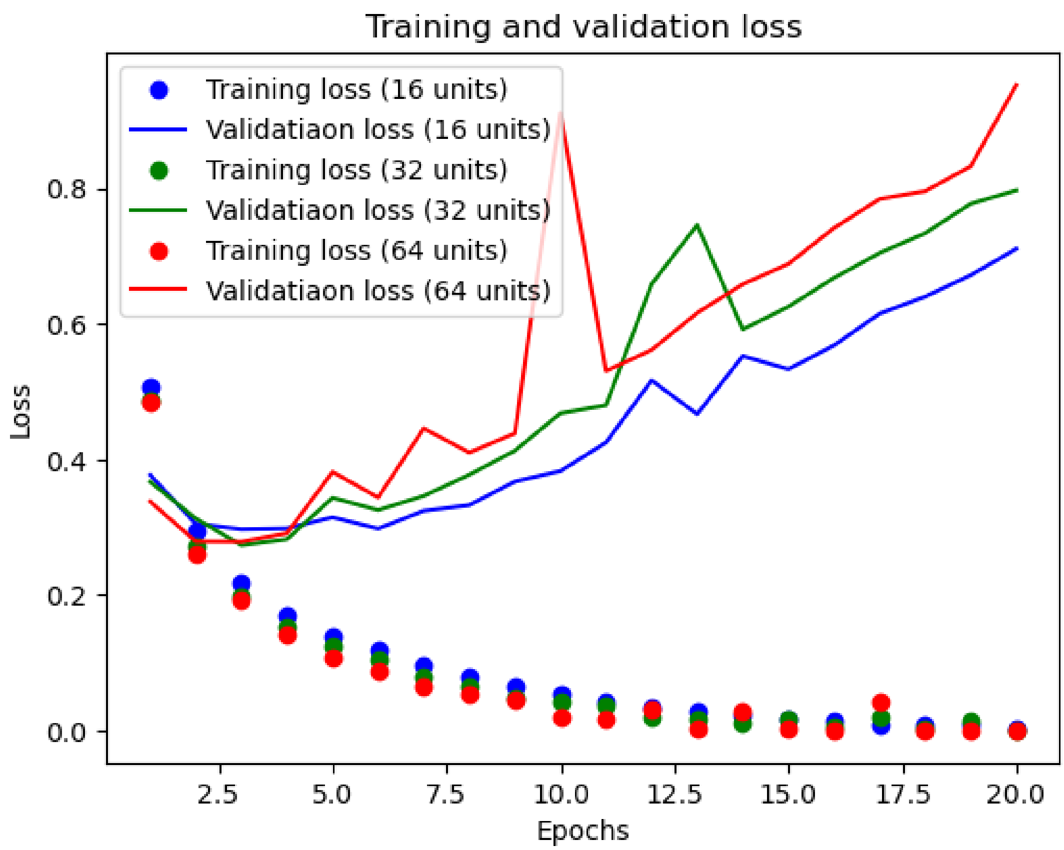
```
30/30 [==============================] - 0s 11ms/step - loss: 0.1250 - acc:
0.9576 - val_loss: 0.3434 - val_acc: 0.8697
Epoch 6/20
30/30 [==============================] - 0s 11ms/step - loss: 0.1050 - acc:
0.9643 - val_loss: 0.3254 - val_acc: 0.8817
Epoch 7/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0783 - acc:
0.9758 - val_loss: 0.3462 - val_acc: 0.8786
Epoch 8/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0641 - acc:
0.9794 - val_loss: 0.3774 - val_acc: 0.8758
Epoch 9/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0477 - acc:
0.9871 - val_loss: 0.4129 - val_acc: 0.8767
Epoch 10/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0423 - acc:
0.9877 - val_loss: 0.4682 - val_acc: 0.8673
Epoch 11/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0367 - acc:
0.9895 - val_loss: 0.4800 - val_acc: 0.8724
Epoch 12/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0188 - acc:
0.9971 - val_loss: 0.6586 - val_acc: 0.8551
Epoch 13/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0181 - acc:
0.9962 - val_loss: 0.7460 - val_acc: 0.8380
Epoch 14/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0117 - acc:
0.9981 - val_loss: 0.5919 - val_acc: 0.8685
Epoch 15/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0183 - acc:
0.9951 - val_loss: 0.6253 - val_acc: 0.8681
Epoch 16/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0045 - acc:
0.9999 - val_loss: 0.6677 - val_acc: 0.8672
Epoch 17/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0191 - acc:
0.9948 - val_loss: 0.7044 - val_acc: 0.8668
Epoch 18/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0023 - acc:
0.9999 - val_loss: 0.7337 - val_acc: 0.8656
Epoch 19/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0154 - acc:
0.9958 - val_loss: 0.7772 - val_acc: 0.8654
Epoch 20/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0014 - acc:
0.9999 - val_loss: 0.7969 - val_acc: 0.8661
Epoch 1/20
```

```
30/30 [==============================] - 1s 27ms/step - loss: 0.4832 - acc:
0.7721 - val_loss: 0.3378 - val_acc: 0.8664
Epoch 2/20
30/30 [==============================] - 0s 12ms/step - loss: 0.2599 - acc:
0.9047 - val_loss: 0.2791 - val_acc: 0.8881
Epoch 3/20
30/30 [==============================] - 0s 14ms/step - loss: 0.1939 - acc:
0.9287 - val_loss: 0.2786 - val_acc: 0.8908
Epoch 4/20
30/30 [==============================] - 0s 15ms/step - loss: 0.1425 - acc:
0.9481 - val_loss: 0.2912 - val_acc: 0.8897
Epoch 5/20
30/30 [==============================] - 0s 12ms/step - loss: 0.1073 - acc:
0.9627 - val_loss: 0.3817 - val_acc: 0.8690
Epoch 6/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0886 - acc:
0.9689 - val_loss: 0.3439 - val_acc: 0.8841
Epoch 7/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0657 - acc:
0.9801 - val_loss: 0.4458 - val_acc: 0.8648
Epoch 8/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0533 - acc:
0.9835 - val_loss: 0.4101 - val_acc: 0.8794
Epoch 9/20
30/30 [==============================] - 0s 14ms/step - loss: 0.0452 - acc:
0.9857 - val_loss: 0.4385 - val_acc: 0.8785
Epoch 10/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0205 - acc:
0.9961 - val_loss: 0.9109 - val_acc: 0.8091
Epoch 11/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0168 - acc:
0.9963 - val_loss: 0.5305 - val_acc: 0.8757
Epoch 12/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0303 - acc:
0.9910 - val_loss: 0.5615 - val_acc: 0.8765
Epoch 13/20
30/30 [==============================] - 0s 15ms/step - loss: 0.0036 - acc:
0.9999 - val_loss: 0.6162 - val_acc: 0.8746
Epoch 14/20
30/30 [==============================] - 0s 15ms/step - loss: 0.0287 - acc:
0.9917 - val_loss: 0.6585 - val_acc: 0.8725
Epoch 15/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0016 - acc:
0.9999 - val_loss: 0.6884 - val_acc: 0.8727
Epoch 16/20
30/30 [==============================] - 0s 13ms/step - loss: 0.0011 - acc:
0.9999 - val_loss: 0.7414 - val_acc: 0.8702
Epoch 17/20
```

```
30/30 [==============================] - 0s 15ms/step - loss: 0.0413 - acc:
0.9914 - val_loss: 0.7842 - val_acc: 0.8666
Epoch 18/20
30/30 [==============================] - 0s 16ms/step - loss: 6.7327e-04 - acc:
1.0000 - val_loss: 0.7956 - val_acc: 0.8694
Epoch 19/20
30/30 [==============================] - 0s 13ms/step - loss: 3.9858e-04 - acc:
1.0000 - val_loss: 0.8324 - val_acc: 0.8695
Epoch 20/20
30/30 [==============================] - 0s 13ms/step - loss: 2.7332e-04 - acc:
1.0000 - val_loss: 0.9522 - val_acc: 0.8640
```

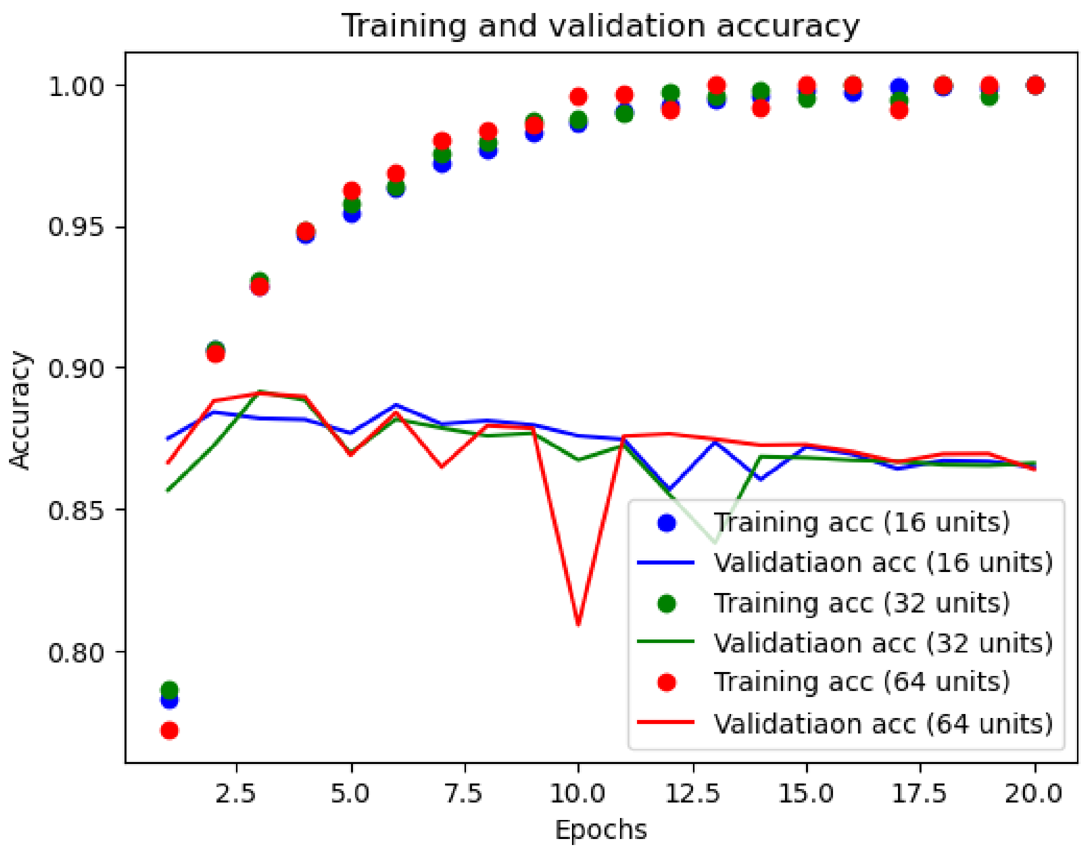
[ ]:

```
loss_values_16 = history_dict_16['loss']
val_loss_values_16 = history_dict_16['val_loss']

loss_values_32 = history_dict_32['loss']
val_loss_values_32 = history_dict_32['val_loss']

loss_values_64 = history_dict_64['loss']
val_loss_values_64 = history_dict_64['val_loss']

epochs = range(1, len(loss_values_64) + 1)

plt.plot(epochs,loss_values_16, 'bo', label="Training loss (16 units)")
plt.plot(epochs,val_loss_values_16, 'b', label="Validatiaon loss (16 units)")
plt.plot(epochs,loss_values_32, 'go', label="Training loss (32 units)")
plt.plot(epochs,val_loss_values_32, 'g', label="Validatiaon loss (32 units)")
plt.plot(epochs,loss_values_64, 'ro', label="Training loss (64 units)")
plt.plot(epochs,val_loss_values_64, 'r', label="Validatiaon loss (64 units)")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

acc_16 = history_dict_16['acc']
val_acc_16 = history_dict_16['val_acc']
acc_32 = history_dict_32['acc']
val_acc_32 = history_dict_32['val_acc']
acc_64 = history_dict_64['acc']
val_acc_64 = history_dict_64['val_acc']

plt.plot(epochs, acc_16, 'bo', label="Training acc (16 units)")
plt.plot(epochs, val_acc_16, 'b', label="Validatiaon acc (16 units)")
plt.plot(epochs, acc_32, 'go', label="Training acc (32 units)")
plt.plot(epochs, val_acc_32, 'g', label="Validatiaon acc (32 units)")
plt.plot(epochs, acc_64, 'ro', label="Training acc (64 units)")
plt.plot(epochs, val_acc_64, 'r', label="Validatiaon acc (64 units)")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
```

```
plt.show()
```




```
[ ]: model_bin = models.Sequential()
model_bin.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model_bin.add(layers.Dense(16,activation='relu'))
model_bin.add(layers.Dense(1, activation='sigmoid'))

model_bin.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
↪loss='binary_crossentropy', metrics=['acc'])

history_bin = model_bin.fit(partial_x_train, partial_y_train, epochs=20,␣
↪batch_size=512, validation_data=(x_val, y_val))

history_dict_bin = history_bin.history

model_mse = models.Sequential()
model_mse.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model_mse.add(layers.Dense(16,activation='relu'))
model_mse.add(layers.Dense(1, activation='sigmoid'))
```

```python
model_mse.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
 ↪loss='mse', metrics=['acc'])

history_mse = model_mse.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))

history_dict_mse = history_mse.history
```

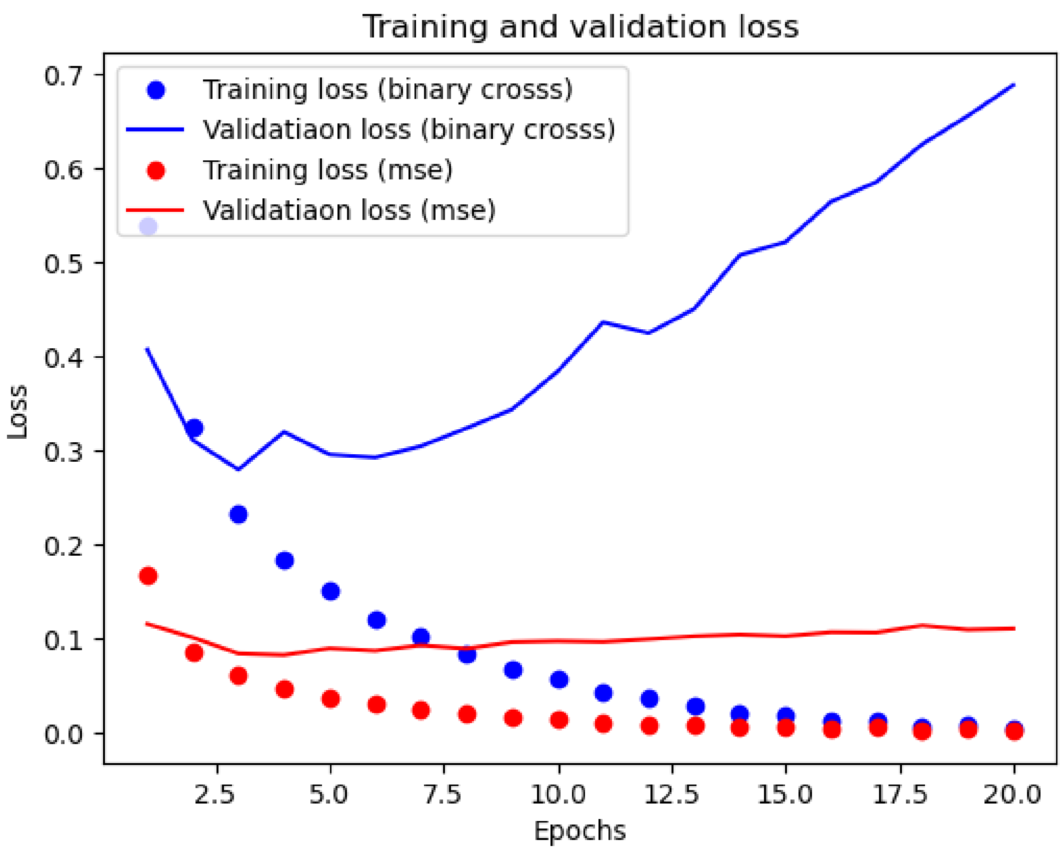
```
Epoch 1/20
30/30 [==============================] - 1s 28ms/step - loss: 0.5380 - acc:
0.7799 - val_loss: 0.4071 - val_acc: 0.8757
Epoch 2/20
30/30 [==============================] - 0s 8ms/step - loss: 0.3256 - acc:
0.9006 - val_loss: 0.3109 - val_acc: 0.8905
Epoch 3/20
30/30 [==============================] - 0s 10ms/step - loss: 0.2335 - acc:
0.9253 - val_loss: 0.2797 - val_acc: 0.8911
Epoch 4/20
30/30 [==============================] - 0s 11ms/step - loss: 0.1841 - acc:
0.9409 - val_loss: 0.3197 - val_acc: 0.8700
Epoch 5/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1513 - acc:
0.9515 - val_loss: 0.2958 - val_acc: 0.8812
Epoch 6/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1217 - acc:
0.9635 - val_loss: 0.2926 - val_acc: 0.8858
Epoch 7/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1017 - acc:
0.9693 - val_loss: 0.3045 - val_acc: 0.8844
Epoch 8/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0847 - acc:
0.9767 - val_loss: 0.3236 - val_acc: 0.8821
Epoch 9/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0686 - acc:
0.9827 - val_loss: 0.3435 - val_acc: 0.8784
Epoch 10/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0570 - acc:
0.9864 - val_loss: 0.3838 - val_acc: 0.8727
Epoch 11/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0439 - acc:
0.9899 - val_loss: 0.4362 - val_acc: 0.8690
Epoch 12/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0376 - acc:
0.9918 - val_loss: 0.4247 - val_acc: 0.8744
Epoch 13/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0295 - acc:
```

```
0.9941 - val_loss: 0.4503 - val_acc: 0.8742
Epoch 14/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0213 - acc:
0.9968 - val_loss: 0.5075 - val_acc: 0.8663
Epoch 15/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0180 - acc:
0.9969 - val_loss: 0.5214 - val_acc: 0.8711
Epoch 16/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0133 - acc:
0.9989 - val_loss: 0.5642 - val_acc: 0.8682
Epoch 17/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0120 - acc:
0.9982 - val_loss: 0.5852 - val_acc: 0.8696
Epoch 18/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0063 - acc:
0.9997 - val_loss: 0.6252 - val_acc: 0.8680
Epoch 19/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0076 - acc:
0.9985 - val_loss: 0.6553 - val_acc: 0.8677
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0053 - acc:
0.9991 - val_loss: 0.6881 - val_acc: 0.8659
Epoch 1/20
30/30 [==============================] - 2s 39ms/step - loss: 0.1675 - acc:
0.7774 - val_loss: 0.1157 - val_acc: 0.8703
Epoch 2/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0867 - acc:
0.9052 - val_loss: 0.1013 - val_acc: 0.8668
Epoch 3/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0614 - acc:
0.9342 - val_loss: 0.0843 - val_acc: 0.8894
Epoch 4/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0479 - acc:
0.9481 - val_loss: 0.0828 - val_acc: 0.8863
Epoch 5/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0380 - acc:
0.9607 - val_loss: 0.0899 - val_acc: 0.8794
Epoch 6/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0316 - acc:
0.9693 - val_loss: 0.0873 - val_acc: 0.8834
Epoch 7/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0258 - acc:
0.9750 - val_loss: 0.0931 - val_acc: 0.8731
Epoch 8/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0211 - acc:
0.9799 - val_loss: 0.0895 - val_acc: 0.8793
Epoch 9/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0168 - acc:
```

```
0.9863 - val_loss: 0.0965 - val_acc: 0.8732
Epoch 10/20
30/30 [==============================] - 0s 12ms/step - loss: 0.0142 - acc:
0.9884 - val_loss: 0.0975 - val_acc: 0.8722
Epoch 11/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0117 - acc:
0.9906 - val_loss: 0.0967 - val_acc: 0.8741
Epoch 12/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0094 - acc:
0.9929 - val_loss: 0.0996 - val_acc: 0.8753
Epoch 13/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0086 - acc:
0.9931 - val_loss: 0.1027 - val_acc: 0.8726
Epoch 14/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0073 - acc:
0.9939 - val_loss: 0.1045 - val_acc: 0.8690
Epoch 15/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0063 - acc:
0.9948 - val_loss: 0.1028 - val_acc: 0.8711
Epoch 16/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0039 - acc:
0.9971 - val_loss: 0.1070 - val_acc: 0.8679
Epoch 17/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0060 - acc:
0.9941 - val_loss: 0.1066 - val_acc: 0.8674
Epoch 18/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0032 - acc:
0.9973 - val_loss: 0.1141 - val_acc: 0.8609
Epoch 19/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0048 - acc:
0.9958 - val_loss: 0.1096 - val_acc: 0.8652
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0027 - acc:
0.9976 - val_loss: 0.1108 - val_acc: 0.8649
```

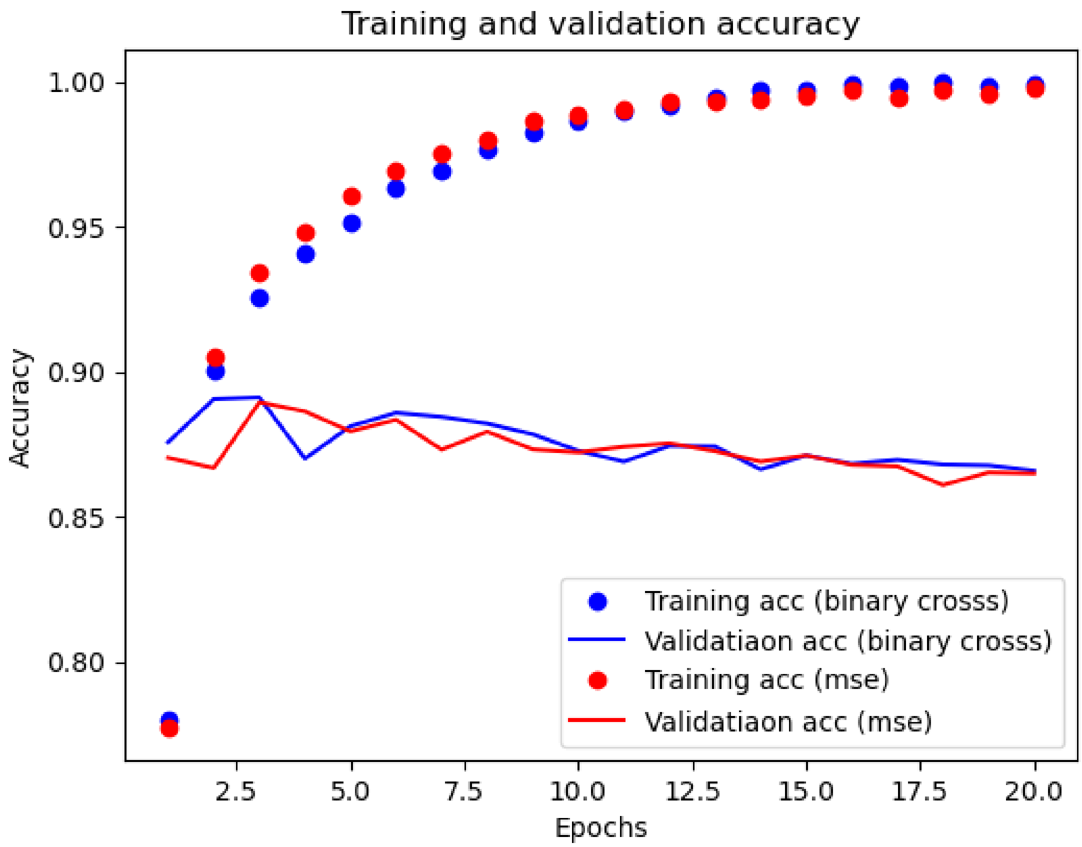
[ ]:
```
loss_values_bin = history_dict_bin['loss']
val_loss_values_bin = history_dict_bin['val_loss']

loss_values_mse = history_dict_mse['loss']
val_loss_values_mse = history_dict_mse['val_loss']

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

plt.plot(epochs, loss_values_bin, 'bo', label="Training loss (binary crosss)")
plt.plot(epochs, val_loss_values_bin, 'b', label="Validatiaon loss (binary␣
↪crosss)")
plt.plot(epochs, loss_values_mse, 'ro', label="Training loss (mse)")
```

```
plt.plot(epochs, val_loss_values_mse, 'r', label="Validatiaon loss (mse)")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

acc_bin = history_dict_bin['acc']
val_acc_bin = history_dict_bin['val_acc']
acc_mse = history_dict_mse['acc']
val_acc_mse = history_dict_mse['val_acc']

plt.plot(epochs, acc_bin, 'bo', label="Training acc (binary crosss)")
plt.plot(epochs, val_acc_bin, 'b', label="Validatiaon acc (binary crosss)")
plt.plot(epochs, acc_mse, 'ro', label="Training acc (mse)")
plt.plot(epochs, val_acc_mse, 'r', label="Validatiaon acc (mse)")
plt.title("Training and validation accuracy")
```

```
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()

plt.show()
```



[ ]:
```
model_relu = models.Sequential()
model_relu.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model_relu.add(layers.Dense(16,activation='relu'))
model_relu.add(layers.Dense(1, activation='sigmoid'))

model_relu.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),␣
↪loss='binary_crossentropy', metrics=['acc'])

history_relu = model_relu.fit(partial_x_train, partial_y_train, epochs=20,␣
↪batch_size=512, validation_data=(x_val, y_val))

history_dict_relu = history_relu.history
```

```python
model_tanh = models.Sequential()
model_tanh.add(layers.Dense(16, activation='tanh', input_shape=(10000,)))
model_tanh.add(layers.Dense(16,activation='tanh'))
model_tanh.add(layers.Dense(1, activation='sigmoid'))

model_tanh.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),
 ↪loss='binary_crossentropy', metrics=['acc'])

history_tanh = model_tanh.fit(partial_x_train, partial_y_train, epochs=20,
 ↪batch_size=512, validation_data=(x_val, y_val))

history_dict_tanh = history_tanh.history
```

```
Epoch 1/20
30/30 [==============================] - 1s 20ms/step - loss: 0.5102 - acc:
0.7929 - val_loss: 0.3876 - val_acc: 0.8711
Epoch 2/20
30/30 [==============================] - 0s 10ms/step - loss: 0.3110 - acc:
0.9021 - val_loss: 0.3095 - val_acc: 0.8888
Epoch 3/20
30/30 [==============================] - 0s 9ms/step - loss: 0.2270 - acc:
0.9295 - val_loss: 0.3152 - val_acc: 0.8727
Epoch 4/20
30/30 [==============================] - 0s 9ms/step - loss: 0.1779 - acc:
0.9441 - val_loss: 0.2955 - val_acc: 0.8804
Epoch 5/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1448 - acc:
0.9549 - val_loss: 0.2784 - val_acc: 0.8876
Epoch 6/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1150 - acc:
0.9647 - val_loss: 0.2928 - val_acc: 0.8849
Epoch 7/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0968 - acc:
0.9702 - val_loss: 0.3082 - val_acc: 0.8830
Epoch 8/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0773 - acc:
0.9789 - val_loss: 0.3364 - val_acc: 0.8781
Epoch 9/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0608 - acc:
0.9844 - val_loss: 0.3577 - val_acc: 0.8774
Epoch 10/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0496 - acc:
0.9876 - val_loss: 0.3787 - val_acc: 0.8786
Epoch 11/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0375 - acc:
0.9925 - val_loss: 0.4115 - val_acc: 0.8751
Epoch 12/20
```

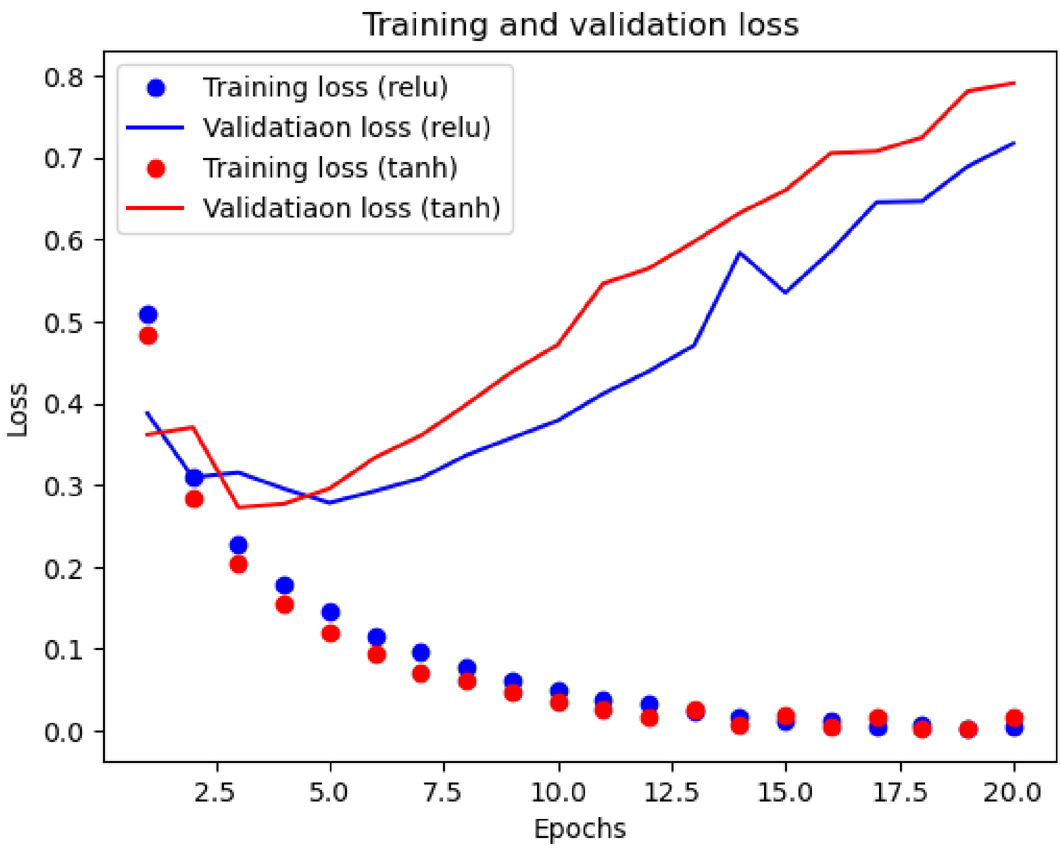
```
30/30 [==============================] - 0s 8ms/step - loss: 0.0323 - acc:
0.9930 - val_loss: 0.4389 - val_acc: 0.8760
Epoch 13/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0232 - acc:
0.9959 - val_loss: 0.4708 - val_acc: 0.8748
Epoch 14/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0162 - acc:
0.9980 - val_loss: 0.5839 - val_acc: 0.8562
Epoch 15/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0128 - acc:
0.9988 - val_loss: 0.5350 - val_acc: 0.8725
Epoch 16/20
30/30 [==============================] - 0s 11ms/step - loss: 0.0114 - acc:
0.9982 - val_loss: 0.5857 - val_acc: 0.8694
Epoch 17/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0057 - acc:
0.9997 - val_loss: 0.6456 - val_acc: 0.8630
Epoch 18/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0067 - acc:
0.9991 - val_loss: 0.6469 - val_acc: 0.8702
Epoch 19/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0030 - acc:
0.9999 - val_loss: 0.6892 - val_acc: 0.8696
Epoch 20/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0049 - acc:
0.9991 - val_loss: 0.7173 - val_acc: 0.8667
Epoch 1/20
30/30 [==============================] - 2s 37ms/step - loss: 0.4826 - acc:
0.8022 - val_loss: 0.3616 - val_acc: 0.8775
Epoch 2/20
30/30 [==============================] - 0s 8ms/step - loss: 0.2846 - acc:
0.9068 - val_loss: 0.3704 - val_acc: 0.8385
Epoch 3/20
30/30 [==============================] - 0s 8ms/step - loss: 0.2049 - acc:
0.9315 - val_loss: 0.2727 - val_acc: 0.8883
Epoch 4/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1546 - acc:
0.9497 - val_loss: 0.2772 - val_acc: 0.8871
Epoch 5/20
30/30 [==============================] - 0s 8ms/step - loss: 0.1193 - acc:
0.9619 - val_loss: 0.2958 - val_acc: 0.8861
Epoch 6/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0947 - acc:
0.9723 - val_loss: 0.3337 - val_acc: 0.8801
Epoch 7/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0717 - acc:
0.9784 - val_loss: 0.3604 - val_acc: 0.8774
Epoch 8/20
```

```
30/30 [==============================] - 0s 9ms/step - loss: 0.0615 - acc:
0.9819 - val_loss: 0.3983 - val_acc: 0.8768
Epoch 9/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0465 - acc:
0.9877 - val_loss: 0.4384 - val_acc: 0.8726
Epoch 10/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0350 - acc:
0.9917 - val_loss: 0.4712 - val_acc: 0.8727
Epoch 11/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0252 - acc:
0.9938 - val_loss: 0.5461 - val_acc: 0.8588
Epoch 12/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0173 - acc:
0.9969 - val_loss: 0.5647 - val_acc: 0.8674
Epoch 13/20
30/30 [==============================] - 0s 10ms/step - loss: 0.0258 - acc:
0.9928 - val_loss: 0.5973 - val_acc: 0.8671
Epoch 14/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0081 - acc:
0.9991 - val_loss: 0.6325 - val_acc: 0.8652
Epoch 15/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0196 - acc:
0.9941 - val_loss: 0.6600 - val_acc: 0.8656
Epoch 16/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0051 - acc:
0.9995 - val_loss: 0.7055 - val_acc: 0.8601
Epoch 17/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0177 - acc:
0.9954 - val_loss: 0.7079 - val_acc: 0.8641
Epoch 18/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0035 - acc:
0.9997 - val_loss: 0.7247 - val_acc: 0.8635
Epoch 19/20
30/30 [==============================] - 0s 8ms/step - loss: 0.0028 - acc:
0.9997 - val_loss: 0.7812 - val_acc: 0.8601
Epoch 20/20
30/30 [==============================] - 0s 9ms/step - loss: 0.0163 - acc:
0.9953 - val_loss: 0.7910 - val_acc: 0.8601
```

[ ]:

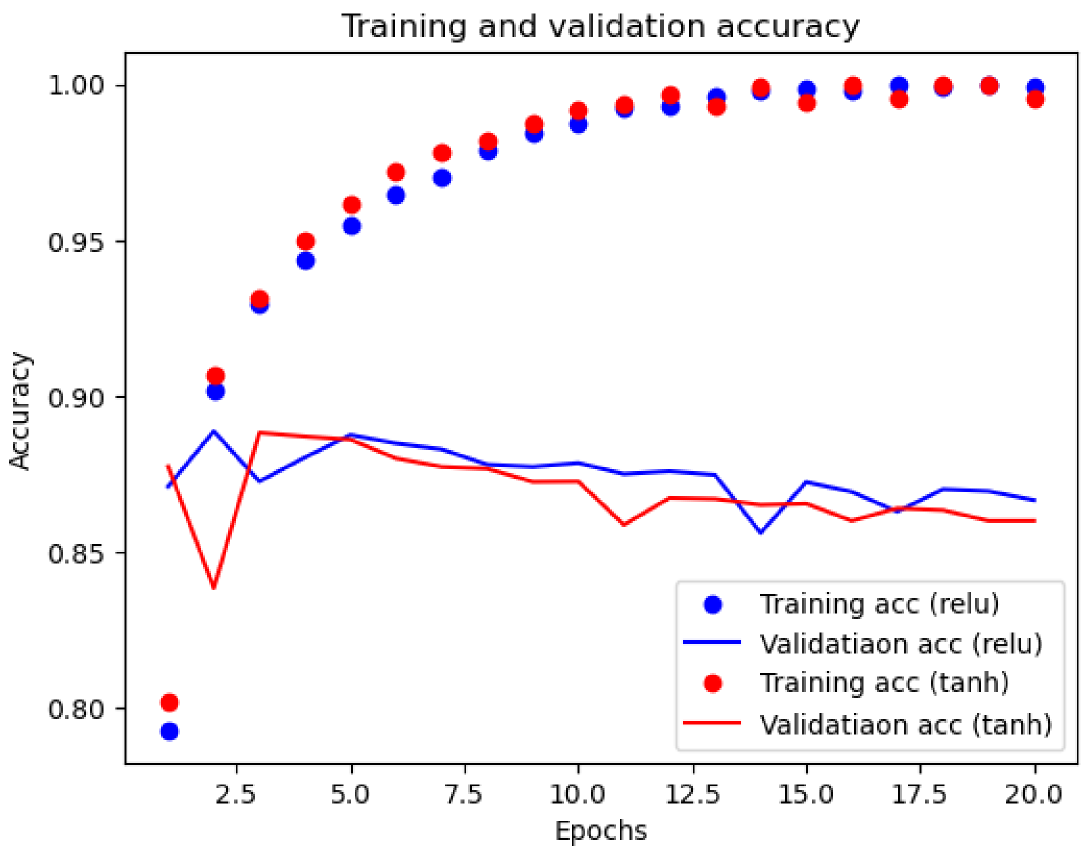
```
loss_values_relu = history_dict_relu['loss']
val_loss_values_relu = history_dict_relu['val_loss']

loss_values_tanh = history_dict_tanh['loss']
val_loss_values_tanh = history_dict_tanh['val_loss']

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

```
plt.plot(epochs, loss_values_relu, 'bo', label="Training loss (relu)")
plt.plot(epochs, val_loss_values_relu, 'b', label="Validatiaon loss (relu)")
plt.plot(epochs, loss_values_tanh, 'ro', label="Training loss (tanh)")
plt.plot(epochs, val_loss_values_tanh, 'r', label="Validatiaon loss (tanh)")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

acc_relu = history_dict_relu['acc']
val_acc_relu = history_dict_relu['val_acc']
acc_tanh = history_dict_tanh['acc']
val_acc_tanh = history_dict_tanh['val_acc']

plt.plot(epochs, acc_relu, 'bo', label="Training acc (relu)")
plt.plot(epochs, val_acc_relu, 'b', label="Validatiaon acc (relu)")
```

```
plt.plot(epochs, acc_tanh, 'ro', label="Training acc (tanh)")
plt.plot(epochs, val_acc_tanh, 'r', label="Validatiaon acc (tanh)")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()

plt.show()
```



0.0.2 Assignment 5.2

Section 3.5

[ ]:
```
from keras.datasets import reuters

(train_data, train_labels), (test_data, test_labels) = reuters.
  ↪load_data(num_words=10000)

print(len(train_data))
print(len(test_data))
```

8982
2246

[ ]:
```
print(train_data[10])
```

[1, 245, 273, 207, 156, 53, 74, 160, 26, 14, 46, 296, 26, 39, 74, 2979, 3554,
14, 46, 4689, 4329, 86, 61, 3499, 4795, 14, 61, 451, 4329, 17, 12]

[ ]:
```
word_index = reuters.get_word_index()
reverse_word_index = dict([(value, key) for (key, value) in word_index.items()])

decoded_newswire = ' '.join([reverse_word_index.get(i - 3, '?') for i in␣
↪train_data[0]])

train_labels[10]
```

[ ]: 3

[ ]:
```
x_train = vectorize_sequences(train_data)
x_test = vectorize_sequences(test_data)

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

one_hot_train_labels = to_one_hot(train_labels)
one_hot_test_labels = to_one_hot(test_labels)
```

[ ]:
```
from keras.utils.np_utils import to_categorical

one_hot_train_labels = to_categorical(train_labels)
one_hot_test_labels = to_categorical(test_labels)
```

[ ]:
```
model = models.Sequential()
model.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(46, activation='softmax'))

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

[ ]:
```
x_val = x_train[:1000]
partial_x_train = x_train[1000:]

y_val = one_hot_train_labels[:1000]
partial_y_train = one_hot_train_labels[1000:]
```

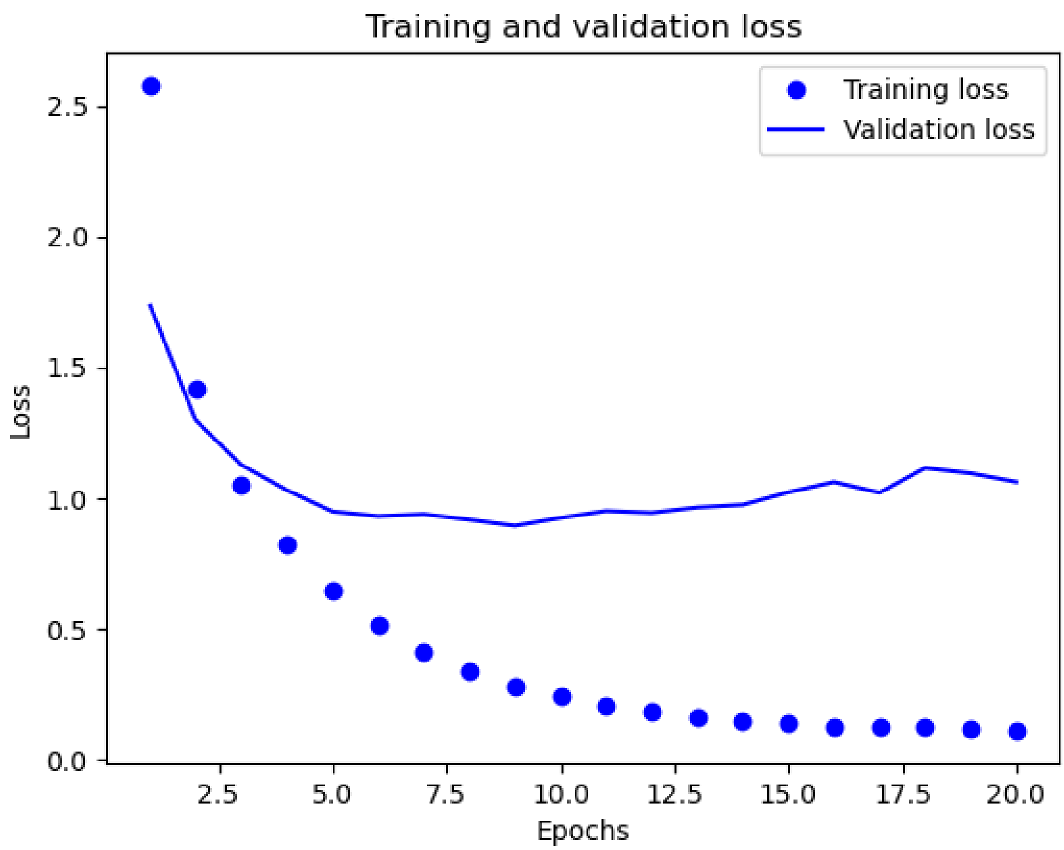
```python
history = model.fit(partial_x_train, partial_y_train, epochs=20,␣
↪batch_size=512, validation_data=(x_val, y_val))
```

```
Epoch 1/20
16/16 [==============================] - 1s 25ms/step - loss: 2.5777 - acc:
0.5279 - val_loss: 1.7344 - val_acc: 0.6430
Epoch 2/20
16/16 [==============================] - 0s 11ms/step - loss: 1.4191 - acc:
0.7096 - val_loss: 1.2960 - val_acc: 0.7150
Epoch 3/20
16/16 [==============================] - 0s 9ms/step - loss: 1.0512 - acc:
0.7829 - val_loss: 1.1259 - val_acc: 0.7650
Epoch 4/20
16/16 [==============================] - 0s 11ms/step - loss: 0.8199 - acc:
0.8326 - val_loss: 1.0295 - val_acc: 0.7830
Epoch 5/20
16/16 [==============================] - 0s 9ms/step - loss: 0.6500 - acc:
0.8637 - val_loss: 0.9482 - val_acc: 0.8000
Epoch 6/20
16/16 [==============================] - 0s 9ms/step - loss: 0.5155 - acc:
0.8928 - val_loss: 0.9303 - val_acc: 0.7930
Epoch 7/20
16/16 [==============================] - 0s 9ms/step - loss: 0.4141 - acc:
0.9143 - val_loss: 0.9377 - val_acc: 0.7900
Epoch 8/20
16/16 [==============================] - 0s 13ms/step - loss: 0.3388 - acc:
0.9290 - val_loss: 0.9169 - val_acc: 0.8040
Epoch 9/20
16/16 [==============================] - 0s 10ms/step - loss: 0.2817 - acc:
0.9371 - val_loss: 0.8940 - val_acc: 0.8110
Epoch 10/20
16/16 [==============================] - 0s 10ms/step - loss: 0.2404 - acc:
0.9461 - val_loss: 0.9237 - val_acc: 0.8160
Epoch 11/20
16/16 [==============================] - 0s 9ms/step - loss: 0.2035 - acc:
0.9486 - val_loss: 0.9503 - val_acc: 0.8020
Epoch 12/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1817 - acc:
0.9546 - val_loss: 0.9434 - val_acc: 0.8070
Epoch 13/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1632 - acc:
0.9540 - val_loss: 0.9639 - val_acc: 0.8080
Epoch 14/20
16/16 [==============================] - 0s 10ms/step - loss: 0.1509 - acc:
0.9535 - val_loss: 0.9740 - val_acc: 0.8050
Epoch 15/20
16/16 [==============================] - 0s 13ms/step - loss: 0.1394 - acc:
```

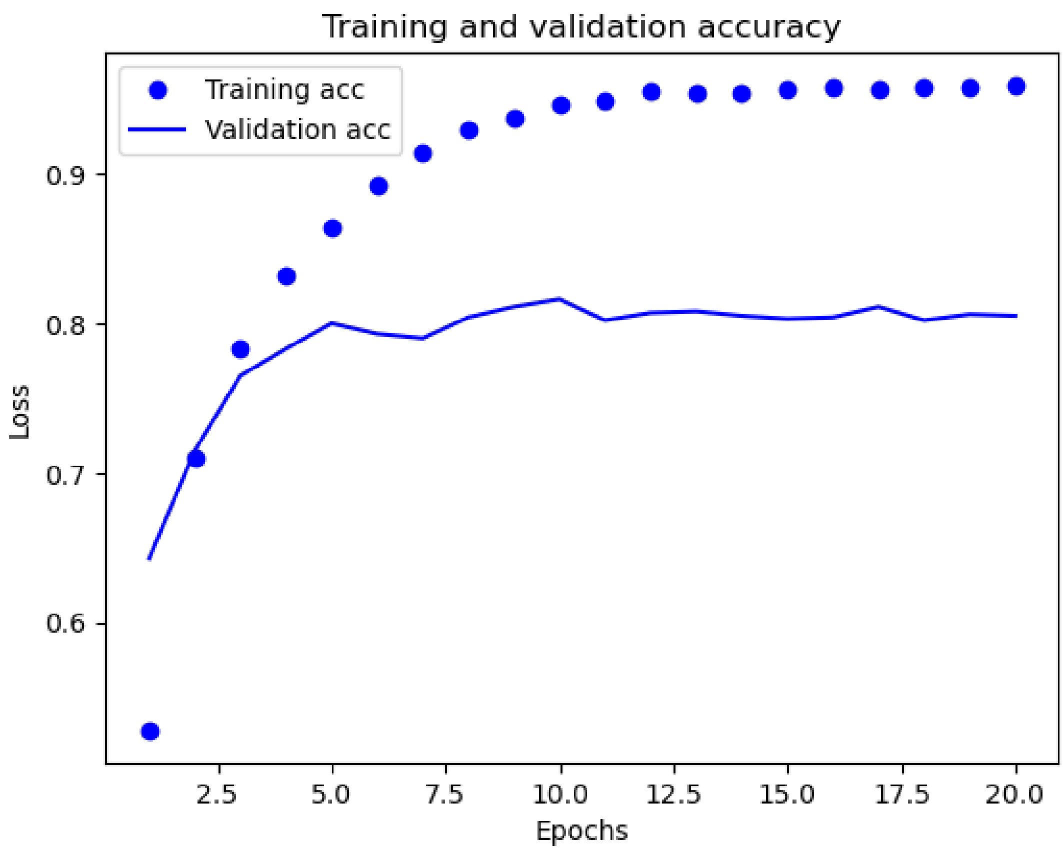
```
0.9570 - val_loss: 1.0218 - val_acc: 0.8030
Epoch 16/20
16/16 [==============================] - 0s 10ms/step - loss: 0.1291 - acc:
0.9575 - val_loss: 1.0607 - val_acc: 0.8040
Epoch 17/20
16/16 [==============================] - 0s 11ms/step - loss: 0.1229 - acc:
0.9560 - val_loss: 1.0201 - val_acc: 0.8110
Epoch 18/20
16/16 [==============================] - 0s 10ms/step - loss: 0.1229 - acc:
0.9580 - val_loss: 1.1142 - val_acc: 0.8020
Epoch 19/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1165 - acc:
0.9572 - val_loss: 1.0941 - val_acc: 0.8060
Epoch 20/20
16/16 [==============================] - 0s 8ms/step - loss: 0.1115 - acc:
0.9592 - val_loss: 1.0612 - val_acc: 0.8050
```

[ ]:

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

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

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

plt.show()
```



[ ]:
```
plt.clf()

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

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

plt.show()
```



[ ]:
```
model = models.Sequential()
model.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(46, activation='softmax'))

model.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
  ↪metrics=['acc'])
model.fit(partial_x_train, partial_y_train, epochs=8, batch_size=512,␣
  ↪validation_data=(x_val, y_val))
results = model.evaluate(x_test, one_hot_test_labels)
results
```

```
Epoch 1/8
16/16 [==============================] - 1s 24ms/step - loss: 2.7395 - acc:
0.5322 - val_loss: 1.7931 - val_acc: 0.6450
Epoch 2/8
16/16 [==============================] - 0s 11ms/step - loss: 1.4607 - acc:
0.7031 - val_loss: 1.3090 - val_acc: 0.7180
Epoch 3/8
16/16 [==============================] - 0s 14ms/step - loss: 1.0547 - acc:
```

```
0.7752 - val_loss: 1.1179 - val_acc: 0.7540
Epoch 4/8
16/16 [==============================] - 0s 10ms/step - loss: 0.8190 - acc:
0.8257 - val_loss: 1.0161 - val_acc: 0.7860
Epoch 5/8
16/16 [==============================] - 0s 10ms/step - loss: 0.6521 - acc:
0.8604 - val_loss: 0.9635 - val_acc: 0.7900
Epoch 6/8
16/16 [==============================] - 0s 10ms/step - loss: 0.5201 - acc:
0.8931 - val_loss: 0.9542 - val_acc: 0.7830
Epoch 7/8
16/16 [==============================] - 0s 10ms/step - loss: 0.4209 - acc:
0.9108 - val_loss: 0.9073 - val_acc: 0.8080
Epoch 8/8
16/16 [==============================] - 0s 9ms/step - loss: 0.3408 - acc:
0.9271 - val_loss: 0.9281 - val_acc: 0.8070
71/71 [==============================] - 1s 20ms/step - loss: 0.9947 - acc:
0.7827
```

[ ]: [0.9946668148040771, 0.7827248573303223]

[ ]:
```
import copy

test_labels_copy = copy.copy(test_labels)
np.random.shuffle(test_labels_copy)
float(np.sum(np.array(test_labels) == np.array(test_labels_copy))) /␣
 ↪len(test_labels)
```

[ ]: 0.18165627782724844

[ ]:
```
predictions = model.predict(x_test)
predictions[0].shape
```

```
71/71 [==============================] - 2s 33ms/step
```

[ ]: (46,)

[ ]:
```
np.sum(predictions[0])
```

[ ]: 1.0000001

[ ]:
```
np.argmax(predictions[0])
```

[ ]: 3

[ ]:
```
y_train = np.array(train_labels)
y_test = np.array(test_labels)
```

```
model.compile(optimizer='rmsprop', loss='sparse_categorical_crossentropy',␣
 ↪metrics=['acc'])

model = models.Sequential()
model.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
model.add(layers.Dense(4, activation='relu'))
model.add(layers.Dense(46, activation='softmax'))

model.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
model.fit(partial_x_train, partial_y_train, epochs=20, batch_size=128,␣
 ↪validation_data=(x_val, y_val))
```

```
Epoch 1/20
63/63 [==============================] - 1s 8ms/step - loss: 2.8850 - accuracy:
0.3760 - val_loss: 2.0033 - val_accuracy: 0.5820
Epoch 2/20
63/63 [==============================] - 0s 5ms/step - loss: 1.6445 - accuracy:
0.6255 - val_loss: 1.4635 - val_accuracy: 0.6400
Epoch 3/20
63/63 [==============================] - 0s 5ms/step - loss: 1.2869 - accuracy:
0.6740 - val_loss: 1.3605 - val_accuracy: 0.6580
Epoch 4/20
63/63 [==============================] - 0s 5ms/step - loss: 1.1184 - accuracy:
0.7170 - val_loss: 1.3060 - val_accuracy: 0.6850
Epoch 5/20
63/63 [==============================] - 0s 4ms/step - loss: 0.9973 - accuracy:
0.7423 - val_loss: 1.2793 - val_accuracy: 0.6940
Epoch 6/20
63/63 [==============================] - 0s 4ms/step - loss: 0.9037 - accuracy:
0.7662 - val_loss: 1.2910 - val_accuracy: 0.7020
Epoch 7/20
63/63 [==============================] - 0s 4ms/step - loss: 0.8226 - accuracy:
0.7849 - val_loss: 1.2974 - val_accuracy: 0.6990
Epoch 8/20
63/63 [==============================] - 0s 5ms/step - loss: 0.7556 - accuracy:
0.8001 - val_loss: 1.3531 - val_accuracy: 0.7020
Epoch 9/20
63/63 [==============================] - 0s 5ms/step - loss: 0.7013 - accuracy:
0.8107 - val_loss: 1.3574 - val_accuracy: 0.7080
Epoch 10/20
63/63 [==============================] - 0s 4ms/step - loss: 0.6506 - accuracy:
0.8259 - val_loss: 1.3887 - val_accuracy: 0.7070
Epoch 11/20
63/63 [==============================] - 0s 4ms/step - loss: 0.6094 - accuracy:
0.8428 - val_loss: 1.4368 - val_accuracy: 0.7140
Epoch 12/20
```

```
63/63 [==============================] - 0s 4ms/step - loss: 0.5715 - accuracy:
0.8545 - val_loss: 1.4797 - val_accuracy: 0.7110
Epoch 13/20
63/63 [==============================] - 0s 4ms/step - loss: 0.5387 - accuracy:
0.8593 - val_loss: 1.5517 - val_accuracy: 0.7050
Epoch 14/20
63/63 [==============================] - 0s 5ms/step - loss: 0.5080 - accuracy:
0.8644 - val_loss: 1.6403 - val_accuracy: 0.7050
Epoch 15/20
63/63 [==============================] - 0s 5ms/step - loss: 0.4822 - accuracy:
0.8715 - val_loss: 1.6493 - val_accuracy: 0.7030
Epoch 16/20
63/63 [==============================] - 0s 5ms/step - loss: 0.4590 - accuracy:
0.8756 - val_loss: 1.7236 - val_accuracy: 0.7040
Epoch 17/20
63/63 [==============================] - 0s 4ms/step - loss: 0.4363 - accuracy:
0.8801 - val_loss: 1.7677 - val_accuracy: 0.7030
Epoch 18/20
63/63 [==============================] - 0s 5ms/step - loss: 0.4217 - accuracy:
0.8811 - val_loss: 1.8639 - val_accuracy: 0.7010
Epoch 19/20
63/63 [==============================] - 0s 4ms/step - loss: 0.4048 - accuracy:
0.8889 - val_loss: 1.9237 - val_accuracy: 0.7030
Epoch 20/20
63/63 [==============================] - 0s 5ms/step - loss: 0.3926 - accuracy:
0.8871 - val_loss: 1.9612 - val_accuracy: 0.7010
```

[ ]: <keras.callbacks.History at 0x236a2b1ac10>

[ ]:

```
model_32 = models.Sequential()
model_32.add(layers.Dense(32, activation='relu', input_shape=(10000,)))
model_32.add(layers.Dense(32, activation='relu'))
model_32.add(layers.Dense(46, activation='softmax'))

model_32.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
 ↪metrics=['acc'])
history_32 = model_32.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))

model_64 = models.Sequential()
model_64.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
model_64.add(layers.Dense(64, activation='relu'))
model_64.add(layers.Dense(46, activation='softmax'))

model_64.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
 ↪metrics=['acc'])
```

```python
history_64 = model_64.fit(partial_x_train, partial_y_train, epochs=20,␣
↪batch_size=512, validation_data=(x_val, y_val))

model_128 = models.Sequential()
model_128.add(layers.Dense(128, activation='relu', input_shape=(10000,)))
model_128.add(layers.Dense(128, activation='relu'))
model_128.add(layers.Dense(46, activation='softmax'))

model_128.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
↪metrics=['acc'])
history_128 = model_128.fit(partial_x_train, partial_y_train, epochs=20,␣
↪batch_size=512, validation_data=(x_val, y_val))
```

```
Epoch 1/20
16/16 [==============================] - 1s 22ms/step - loss: 2.9694 - acc:
0.4996 - val_loss: 2.2671 - val_acc: 0.6080
Epoch 2/20
16/16 [==============================] - 0s 8ms/step - loss: 1.8988 - acc:
0.6689 - val_loss: 1.6634 - val_acc: 0.6600
Epoch 3/20
16/16 [==============================] - 0s 8ms/step - loss: 1.4275 - acc:
0.7093 - val_loss: 1.3851 - val_acc: 0.6980
Epoch 4/20
16/16 [==============================] - 0s 9ms/step - loss: 1.1665 - acc:
0.7464 - val_loss: 1.2350 - val_acc: 0.7240
Epoch 5/20
16/16 [==============================] - 0s 8ms/step - loss: 0.9875 - acc:
0.7871 - val_loss: 1.1529 - val_acc: 0.7430
Epoch 6/20
16/16 [==============================] - 0s 13ms/step - loss: 0.8462 - acc:
0.8151 - val_loss: 1.0762 - val_acc: 0.7660
Epoch 7/20
16/16 [==============================] - 0s 8ms/step - loss: 0.7302 - acc:
0.8405 - val_loss: 1.0372 - val_acc: 0.7680
Epoch 8/20
16/16 [==============================] - 0s 8ms/step - loss: 0.6306 - acc:
0.8626 - val_loss: 0.9975 - val_acc: 0.7810
Epoch 9/20
16/16 [==============================] - 0s 8ms/step - loss: 0.5459 - acc:
0.8835 - val_loss: 0.9610 - val_acc: 0.8010
Epoch 10/20
16/16 [==============================] - 0s 7ms/step - loss: 0.4713 - acc:
0.8954 - val_loss: 0.9560 - val_acc: 0.7950
Epoch 11/20
16/16 [==============================] - 0s 7ms/step - loss: 0.4098 - acc:
0.9085 - val_loss: 0.9387 - val_acc: 0.8020
Epoch 12/20
```

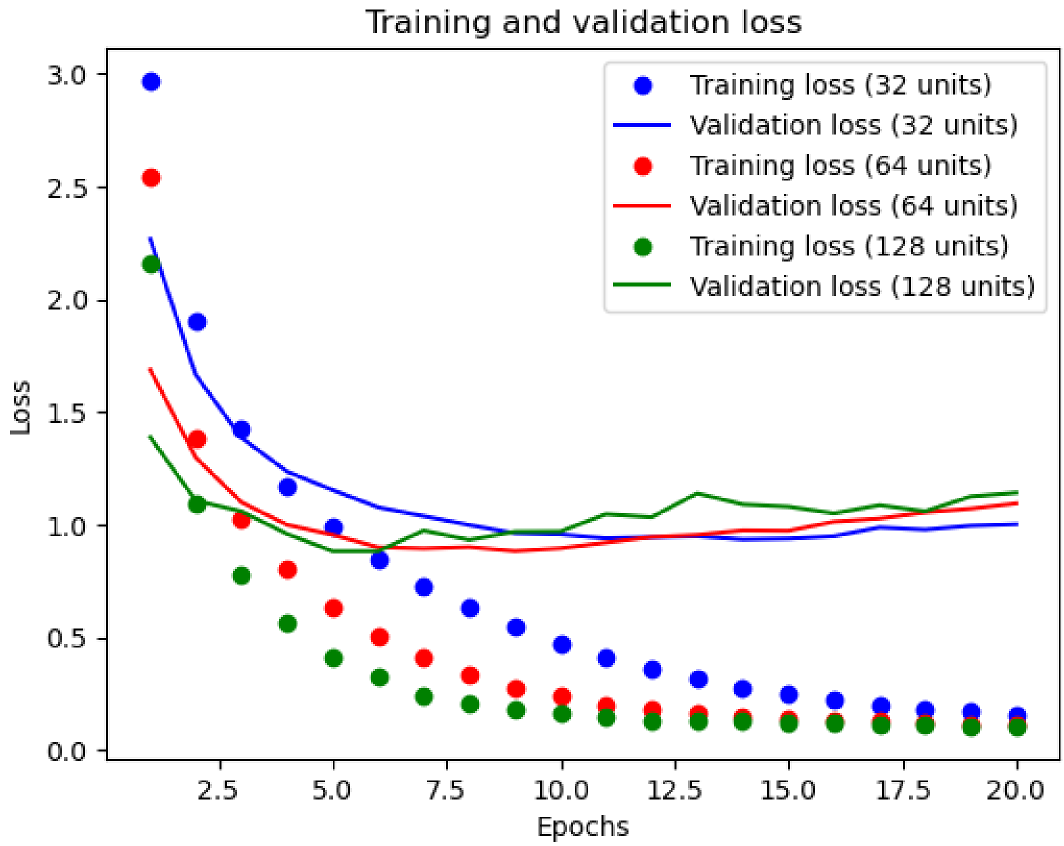
```
16/16 [==============================] - 0s 7ms/step - loss: 0.3572 - acc:
0.9194 - val_loss: 0.9439 - val_acc: 0.8020
Epoch 13/20
16/16 [==============================] - 0s 8ms/step - loss: 0.3158 - acc:
0.9270 - val_loss: 0.9482 - val_acc: 0.8090
Epoch 14/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2767 - acc:
0.9364 - val_loss: 0.9334 - val_acc: 0.8090
Epoch 15/20
16/16 [==============================] - 0s 12ms/step - loss: 0.2469 - acc:
0.9405 - val_loss: 0.9369 - val_acc: 0.8150
Epoch 16/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2223 - acc:
0.9460 - val_loss: 0.9482 - val_acc: 0.8120
Epoch 17/20
16/16 [==============================] - 0s 7ms/step - loss: 0.2005 - acc:
0.9491 - val_loss: 0.9868 - val_acc: 0.7970
Epoch 18/20
16/16 [==============================] - 0s 7ms/step - loss: 0.1830 - acc:
0.9506 - val_loss: 0.9772 - val_acc: 0.8070
Epoch 19/20
16/16 [==============================] - 0s 8ms/step - loss: 0.1690 - acc:
0.9538 - val_loss: 0.9945 - val_acc: 0.8060
Epoch 20/20
16/16 [==============================] - 0s 7ms/step - loss: 0.1557 - acc:
0.9549 - val_loss: 1.0011 - val_acc: 0.8010
Epoch 1/20
16/16 [==============================] - 1s 25ms/step - loss: 2.5398 - acc:
0.5296 - val_loss: 1.6864 - val_acc: 0.6350
Epoch 2/20
16/16 [==============================] - 0s 10ms/step - loss: 1.3850 - acc:
0.7097 - val_loss: 1.2965 - val_acc: 0.7040
Epoch 3/20
16/16 [==============================] - 0s 13ms/step - loss: 1.0226 - acc:
0.7809 - val_loss: 1.0988 - val_acc: 0.7790
Epoch 4/20
16/16 [==============================] - 0s 9ms/step - loss: 0.8002 - acc:
0.8327 - val_loss: 0.9985 - val_acc: 0.8010
Epoch 5/20
16/16 [==============================] - 0s 9ms/step - loss: 0.6317 - acc:
0.8690 - val_loss: 0.9543 - val_acc: 0.7970
Epoch 6/20
16/16 [==============================] - 0s 11ms/step - loss: 0.5084 - acc:
0.8948 - val_loss: 0.8985 - val_acc: 0.8120
Epoch 7/20
16/16 [==============================] - 0s 10ms/step - loss: 0.4096 - acc:
0.9123 - val_loss: 0.8933 - val_acc: 0.8080
Epoch 8/20
```

```
16/16 [==============================] - 0s 10ms/step - loss: 0.3358 - acc:
0.9282 - val_loss: 0.8995 - val_acc: 0.8010
Epoch 9/20
16/16 [==============================] - 0s 9ms/step - loss: 0.2773 - acc:
0.9406 - val_loss: 0.8821 - val_acc: 0.8180
Epoch 10/20
16/16 [==============================] - 0s 9ms/step - loss: 0.2416 - acc:
0.9440 - val_loss: 0.8942 - val_acc: 0.8130
Epoch 11/20
16/16 [==============================] - 0s 13ms/step - loss: 0.2010 - acc:
0.9498 - val_loss: 0.9194 - val_acc: 0.8200
Epoch 12/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1780 - acc:
0.9516 - val_loss: 0.9446 - val_acc: 0.8120
Epoch 13/20
16/16 [==============================] - 0s 10ms/step - loss: 0.1663 - acc:
0.9519 - val_loss: 0.9536 - val_acc: 0.8190
Epoch 14/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1476 - acc:
0.9557 - val_loss: 0.9736 - val_acc: 0.8130
Epoch 15/20
16/16 [==============================] - 0s 10ms/step - loss: 0.1401 - acc:
0.9560 - val_loss: 0.9720 - val_acc: 0.8020
Epoch 16/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1304 - acc:
0.9579 - val_loss: 1.0120 - val_acc: 0.8060
Epoch 17/20
16/16 [==============================] - 0s 10ms/step - loss: 0.1251 - acc:
0.9583 - val_loss: 1.0266 - val_acc: 0.8040
Epoch 18/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1210 - acc:
0.9568 - val_loss: 1.0537 - val_acc: 0.7990
Epoch 19/20
16/16 [==============================] - 0s 13ms/step - loss: 0.1148 - acc:
0.9575 - val_loss: 1.0701 - val_acc: 0.8040
Epoch 20/20
16/16 [==============================] - 0s 8ms/step - loss: 0.1129 - acc:
0.9580 - val_loss: 1.0935 - val_acc: 0.7990
Epoch 1/20
16/16 [==============================] - 1s 28ms/step - loss: 2.1568 - acc:
0.5660 - val_loss: 1.3875 - val_acc: 0.7030
Epoch 2/20
16/16 [==============================] - 0s 14ms/step - loss: 1.0943 - acc:
0.7615 - val_loss: 1.1057 - val_acc: 0.7590
Epoch 3/20
16/16 [==============================] - 0s 14ms/step - loss: 0.7801 - acc:
0.8338 - val_loss: 1.0581 - val_acc: 0.7620
Epoch 4/20
```

```
16/16 [==============================] - 0s 14ms/step - loss: 0.5632 - acc:
0.8820 - val_loss: 0.9580 - val_acc: 0.7850
Epoch 5/20
16/16 [==============================] - 0s 16ms/step - loss: 0.4147 - acc:
0.9143 - val_loss: 0.8815 - val_acc: 0.8190
Epoch 6/20
16/16 [==============================] - 0s 13ms/step - loss: 0.3223 - acc:
0.9300 - val_loss: 0.8816 - val_acc: 0.8170
Epoch 7/20
16/16 [==============================] - 0s 13ms/step - loss: 0.2415 - acc:
0.9432 - val_loss: 0.9728 - val_acc: 0.8120
Epoch 8/20
16/16 [==============================] - 0s 14ms/step - loss: 0.2030 - acc:
0.9485 - val_loss: 0.9316 - val_acc: 0.8190
Epoch 9/20
16/16 [==============================] - 0s 14ms/step - loss: 0.1780 - acc:
0.9536 - val_loss: 0.9679 - val_acc: 0.8080
Epoch 10/20
16/16 [==============================] - 0s 13ms/step - loss: 0.1604 - acc:
0.9548 - val_loss: 0.9689 - val_acc: 0.8100
Epoch 11/20
16/16 [==============================] - 0s 15ms/step - loss: 0.1460 - acc:
0.9558 - val_loss: 1.0462 - val_acc: 0.8090
Epoch 12/20
16/16 [==============================] - 0s 14ms/step - loss: 0.1322 - acc:
0.9564 - val_loss: 1.0324 - val_acc: 0.8100
Epoch 13/20
16/16 [==============================] - 0s 13ms/step - loss: 0.1286 - acc:
0.9551 - val_loss: 1.1380 - val_acc: 0.7930
Epoch 14/20
16/16 [==============================] - 0s 13ms/step - loss: 0.1305 - acc:
0.9562 - val_loss: 1.0893 - val_acc: 0.8020
Epoch 15/20
16/16 [==============================] - 0s 13ms/step - loss: 0.1180 - acc:
0.9569 - val_loss: 1.0791 - val_acc: 0.8030
Epoch 16/20
16/16 [==============================] - 0s 14ms/step - loss: 0.1167 - acc:
0.9579 - val_loss: 1.0494 - val_acc: 0.8130
Epoch 17/20
16/16 [==============================] - 0s 17ms/step - loss: 0.1100 - acc:
0.9573 - val_loss: 1.0856 - val_acc: 0.8090
Epoch 18/20
16/16 [==============================] - 0s 17ms/step - loss: 0.1109 - acc:
0.9580 - val_loss: 1.0585 - val_acc: 0.8080
Epoch 19/20
16/16 [==============================] - 0s 14ms/step - loss: 0.1063 - acc:
0.9590 - val_loss: 1.1239 - val_acc: 0.8070
Epoch 20/20
```

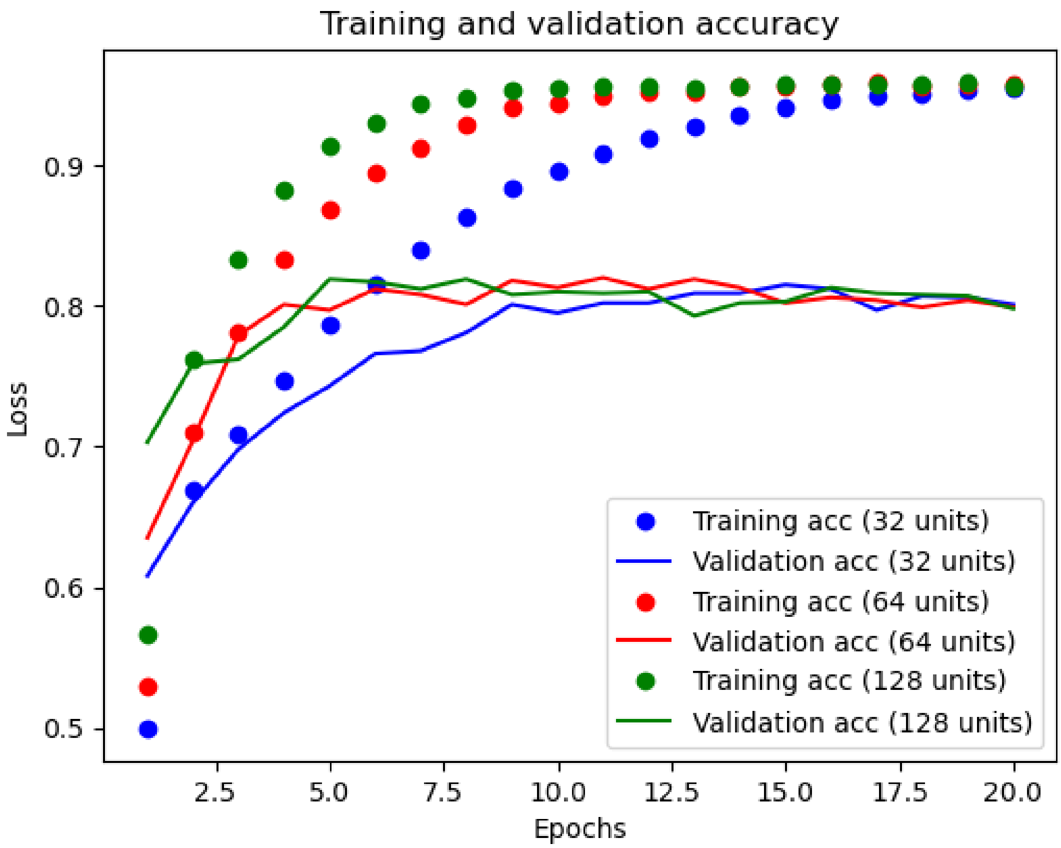
```
16/16 [==============================] - 0s 12ms/step - loss: 0.1042 - acc:
0.9563 - val_loss: 1.1408 - val_acc: 0.7980
```

[ ]:
```
loss_32 = history_32.history['loss']
val_loss_32 = history_32.history['val_loss']

loss_64 = history_64.history['loss']
val_loss_64 = history_64.history['val_loss']

loss_128 = history_128.history['loss']
val_loss_128 = history_128.history['val_loss']

epochs = range(1, len(loss_32) + 1)

plt.plot(epochs, loss_32, 'bo', label='Training loss (32 units)')
plt.plot(epochs, val_loss_32, 'b', label='Validation loss (32 units)')
plt.plot(epochs, loss_64, 'ro', label='Training loss (64 units)')
plt.plot(epochs, val_loss_64, 'r', label='Validation loss (64 units)')
plt.plot(epochs, loss_128, 'go', label='Training loss (128 units)')
plt.plot(epochs, val_loss_128, 'g', label='Validation loss (128 units)')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

acc_32 = history_32.history['acc']
val_acc_32 = history_32.history['val_acc']

acc_64 = history_64.history['acc']
val_acc_64 = history_64.history['val_acc']

acc_128 = history_128.history['acc']
val_acc_128 = history_128.history['val_acc']

plt.plot(epochs, acc_32, 'bo', label='Training acc (32 units)')
plt.plot(epochs, val_acc_32, 'b', label='Validation acc (32 units)')
plt.plot(epochs, acc_64, 'ro', label='Training acc (64 units)')
plt.plot(epochs, val_acc_64, 'r', label='Validation acc (64 units)')
plt.plot(epochs, acc_128, 'go', label='Training acc (128 units)')
plt.plot(epochs, val_acc_128, 'g', label='Validation acc (128 units)')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Loss')
```

```
plt.legend()

plt.show()
```




```
[ ]: model_1 = models.Sequential()
model_1.add(layers.Dense(32, activation='relu', input_shape=(10000,)))
model_1.add(layers.Dense(46, activation='softmax'))

model_1.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
 ↪metrics=['acc'])
history_1 = model_1.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))


model_2 = models.Sequential()
model_2.add(layers.Dense(32, activation='relu', input_shape=(10000,)))
model_2.add(layers.Dense(32, activation='relu'))
model_2.add(layers.Dense(46, activation='softmax'))
```

```python
model_2.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
 ↪metrics=['acc'])
history_2 = model_2.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))


model_3 = models.Sequential()
model_3.add(layers.Dense(32, activation='relu', input_shape=(10000,)))
model_3.add(layers.Dense(32, activation='relu'))
model_3.add(layers.Dense(32, activation='relu'))
model_3.add(layers.Dense(46, activation='softmax'))

model_3.compile(optimizer='rmsprop', loss='categorical_crossentropy',␣
 ↪metrics=['acc'])
history_3 = model_3.fit(partial_x_train, partial_y_train, epochs=20,␣
 ↪batch_size=512, validation_data=(x_val, y_val))
```

```
Epoch 1/20
16/16 [==============================] - 1s 21ms/step - loss: 3.2847 - acc:
0.5055 - val_loss: 2.7193 - val_acc: 0.6290
Epoch 2/20
16/16 [==============================] - 0s 9ms/step - loss: 2.2841 - acc:
0.6721 - val_loss: 1.9871 - val_acc: 0.6830
Epoch 3/20
16/16 [==============================] - 0s 7ms/step - loss: 1.6616 - acc:
0.7187 - val_loss: 1.5624 - val_acc: 0.7140
Epoch 4/20
16/16 [==============================] - 0s 7ms/step - loss: 1.2881 - acc:
0.7625 - val_loss: 1.3153 - val_acc: 0.7380
Epoch 5/20
16/16 [==============================] - 0s 7ms/step - loss: 1.0535 - acc:
0.7912 - val_loss: 1.1691 - val_acc: 0.7550
Epoch 6/20
16/16 [==============================] - 0s 8ms/step - loss: 0.8889 - acc:
0.8197 - val_loss: 1.0735 - val_acc: 0.7840
Epoch 7/20
16/16 [==============================] - 0s 8ms/step - loss: 0.7637 - acc:
0.8454 - val_loss: 1.0082 - val_acc: 0.7910
Epoch 8/20
16/16 [==============================] - 0s 10ms/step - loss: 0.6627 - acc:
0.8677 - val_loss: 0.9616 - val_acc: 0.7950
Epoch 9/20
16/16 [==============================] - 0s 7ms/step - loss: 0.5797 - acc:
0.8851 - val_loss: 0.9242 - val_acc: 0.8070
Epoch 10/20
16/16 [==============================] - 0s 7ms/step - loss: 0.5089 - acc:
0.8999 - val_loss: 0.8993 - val_acc: 0.8150
```

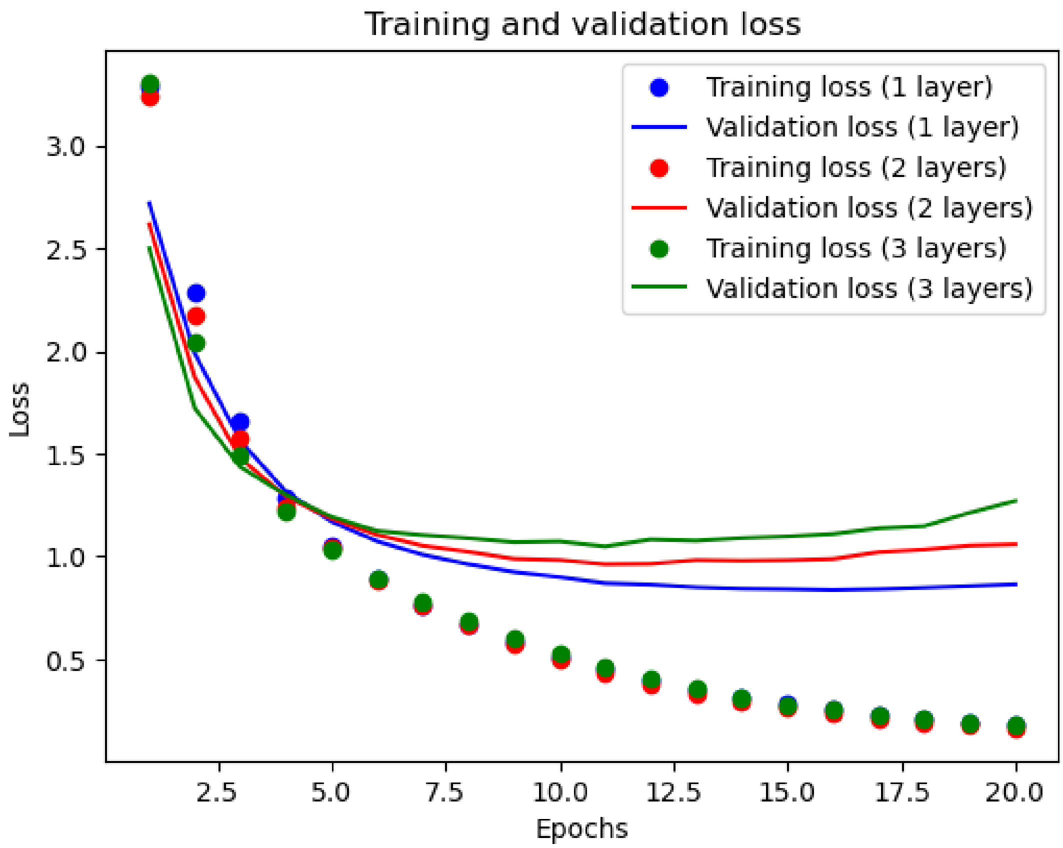
```
Epoch 11/20
16/16 [==============================] - 0s 8ms/step - loss: 0.4495 - acc:
0.9110 - val_loss: 0.8697 - val_acc: 0.8180
Epoch 12/20
16/16 [==============================] - 0s 7ms/step - loss: 0.3971 - acc:
0.9217 - val_loss: 0.8627 - val_acc: 0.8160
Epoch 13/20
16/16 [==============================] - 0s 7ms/step - loss: 0.3529 - acc:
0.9287 - val_loss: 0.8493 - val_acc: 0.8200
Epoch 14/20
16/16 [==============================] - 0s 7ms/step - loss: 0.3157 - acc:
0.9335 - val_loss: 0.8427 - val_acc: 0.8210
Epoch 15/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2826 - acc:
0.9409 - val_loss: 0.8401 - val_acc: 0.8170
Epoch 16/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2543 - acc:
0.9435 - val_loss: 0.8358 - val_acc: 0.8160
Epoch 17/20
16/16 [==============================] - 0s 7ms/step - loss: 0.2303 - acc:
0.9478 - val_loss: 0.8402 - val_acc: 0.8170
Epoch 18/20
16/16 [==============================] - 0s 13ms/step - loss: 0.2108 - acc:
0.9504 - val_loss: 0.8475 - val_acc: 0.8160
Epoch 19/20
16/16 [==============================] - 0s 8ms/step - loss: 0.1928 - acc:
0.9514 - val_loss: 0.8555 - val_acc: 0.8210
Epoch 20/20
16/16 [==============================] - 0s 6ms/step - loss: 0.1783 - acc:
0.9519 - val_loss: 0.8644 - val_acc: 0.8200
Epoch 1/20
16/16 [==============================] - 2s 26ms/step - loss: 3.2385 - acc:
0.3924 - val_loss: 2.6162 - val_acc: 0.5940
Epoch 2/20
16/16 [==============================] - 0s 8ms/step - loss: 2.1704 - acc:
0.6538 - val_loss: 1.8704 - val_acc: 0.6500
Epoch 3/20
16/16 [==============================] - 0s 8ms/step - loss: 1.5740 - acc:
0.7073 - val_loss: 1.4773 - val_acc: 0.6920
Epoch 4/20
16/16 [==============================] - 0s 7ms/step - loss: 1.2426 - acc:
0.7479 - val_loss: 1.2914 - val_acc: 0.7150
Epoch 5/20
16/16 [==============================] - 0s 8ms/step - loss: 1.0373 - acc:
0.7811 - val_loss: 1.1826 - val_acc: 0.7460
Epoch 6/20
16/16 [==============================] - 0s 8ms/step - loss: 0.8864 - acc:
0.8122 - val_loss: 1.1047 - val_acc: 0.7640
```

```
Epoch 7/20
16/16 [==============================] - 0s 8ms/step - loss: 0.7646 - acc:
0.8383 - val_loss: 1.0519 - val_acc: 0.7760
Epoch 8/20
16/16 [==============================] - 0s 8ms/step - loss: 0.6632 - acc:
0.8594 - val_loss: 1.0225 - val_acc: 0.7860
Epoch 9/20
16/16 [==============================] - 0s 9ms/step - loss: 0.5754 - acc:
0.8804 - val_loss: 0.9883 - val_acc: 0.8000
Epoch 10/20
16/16 [==============================] - 0s 8ms/step - loss: 0.5007 - acc:
0.8975 - val_loss: 0.9822 - val_acc: 0.8040
Epoch 11/20
16/16 [==============================] - 0s 9ms/step - loss: 0.4342 - acc:
0.9122 - val_loss: 0.9614 - val_acc: 0.8020
Epoch 12/20
16/16 [==============================] - 0s 7ms/step - loss: 0.3804 - acc:
0.9206 - val_loss: 0.9635 - val_acc: 0.8110
Epoch 13/20
16/16 [==============================] - 0s 8ms/step - loss: 0.3347 - acc:
0.9278 - val_loss: 0.9817 - val_acc: 0.8070
Epoch 14/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2914 - acc:
0.9340 - val_loss: 0.9786 - val_acc: 0.8110
Epoch 15/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2627 - acc:
0.9386 - val_loss: 0.9812 - val_acc: 0.8090
Epoch 16/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2323 - acc:
0.9465 - val_loss: 0.9875 - val_acc: 0.8190
Epoch 17/20
16/16 [==============================] - 0s 7ms/step - loss: 0.2104 - acc:
0.9481 - val_loss: 1.0207 - val_acc: 0.8040
Epoch 18/20
16/16 [==============================] - 0s 8ms/step - loss: 0.1918 - acc:
0.9514 - val_loss: 1.0329 - val_acc: 0.8060
Epoch 19/20
16/16 [==============================] - 0s 8ms/step - loss: 0.1761 - acc:
0.9519 - val_loss: 1.0524 - val_acc: 0.8050
Epoch 20/20
16/16 [==============================] - 0s 11ms/step - loss: 0.1625 - acc:
0.9541 - val_loss: 1.0587 - val_acc: 0.8070
Epoch 1/20
16/16 [==============================] - 1s 22ms/step - loss: 3.3069 - acc:
0.4042 - val_loss: 2.4995 - val_acc: 0.5410
Epoch 2/20
16/16 [==============================] - 0s 8ms/step - loss: 2.0406 - acc:
0.5864 - val_loss: 1.7209 - val_acc: 0.6170
```

```
Epoch 3/20
16/16 [==============================] - 0s 8ms/step - loss: 1.4946 - acc:
0.6642 - val_loss: 1.4352 - val_acc: 0.6730
Epoch 4/20
16/16 [==============================] - 0s 9ms/step - loss: 1.2230 - acc:
0.7225 - val_loss: 1.2986 - val_acc: 0.6940
Epoch 5/20
16/16 [==============================] - 0s 8ms/step - loss: 1.0361 - acc:
0.7656 - val_loss: 1.1926 - val_acc: 0.7320
Epoch 6/20
16/16 [==============================] - 0s 8ms/step - loss: 0.8926 - acc:
0.8036 - val_loss: 1.1240 - val_acc: 0.7540
Epoch 7/20
16/16 [==============================] - 0s 8ms/step - loss: 0.7771 - acc:
0.8271 - val_loss: 1.1035 - val_acc: 0.7570
Epoch 8/20
16/16 [==============================] - 0s 8ms/step - loss: 0.6813 - acc:
0.8449 - val_loss: 1.0886 - val_acc: 0.7620
Epoch 9/20
16/16 [==============================] - 0s 11ms/step - loss: 0.5970 - acc:
0.8589 - val_loss: 1.0689 - val_acc: 0.7600
Epoch 10/20
16/16 [==============================] - 0s 8ms/step - loss: 0.5236 - acc:
0.8760 - val_loss: 1.0727 - val_acc: 0.7730
Epoch 11/20
16/16 [==============================] - 0s 9ms/step - loss: 0.4606 - acc:
0.8871 - val_loss: 1.0483 - val_acc: 0.7780
Epoch 12/20
16/16 [==============================] - 0s 8ms/step - loss: 0.4017 - acc:
0.9038 - val_loss: 1.0820 - val_acc: 0.7650
Epoch 13/20
16/16 [==============================] - 0s 8ms/step - loss: 0.3547 - acc:
0.9164 - val_loss: 1.0760 - val_acc: 0.7730
Epoch 14/20
16/16 [==============================] - 0s 8ms/step - loss: 0.3142 - acc:
0.9262 - val_loss: 1.0894 - val_acc: 0.7690
Epoch 15/20
16/16 [==============================] - 0s 8ms/step - loss: 0.2785 - acc:
0.9336 - val_loss: 1.0968 - val_acc: 0.7830
Epoch 16/20
16/16 [==============================] - 0s 9ms/step - loss: 0.2528 - acc:
0.9415 - val_loss: 1.1090 - val_acc: 0.7820
Epoch 17/20
16/16 [==============================] - 0s 9ms/step - loss: 0.2265 - acc:
0.9424 - val_loss: 1.1367 - val_acc: 0.7850
Epoch 18/20
16/16 [==============================] - 0s 12ms/step - loss: 0.2064 - acc:
0.9485 - val_loss: 1.1473 - val_acc: 0.7770
```

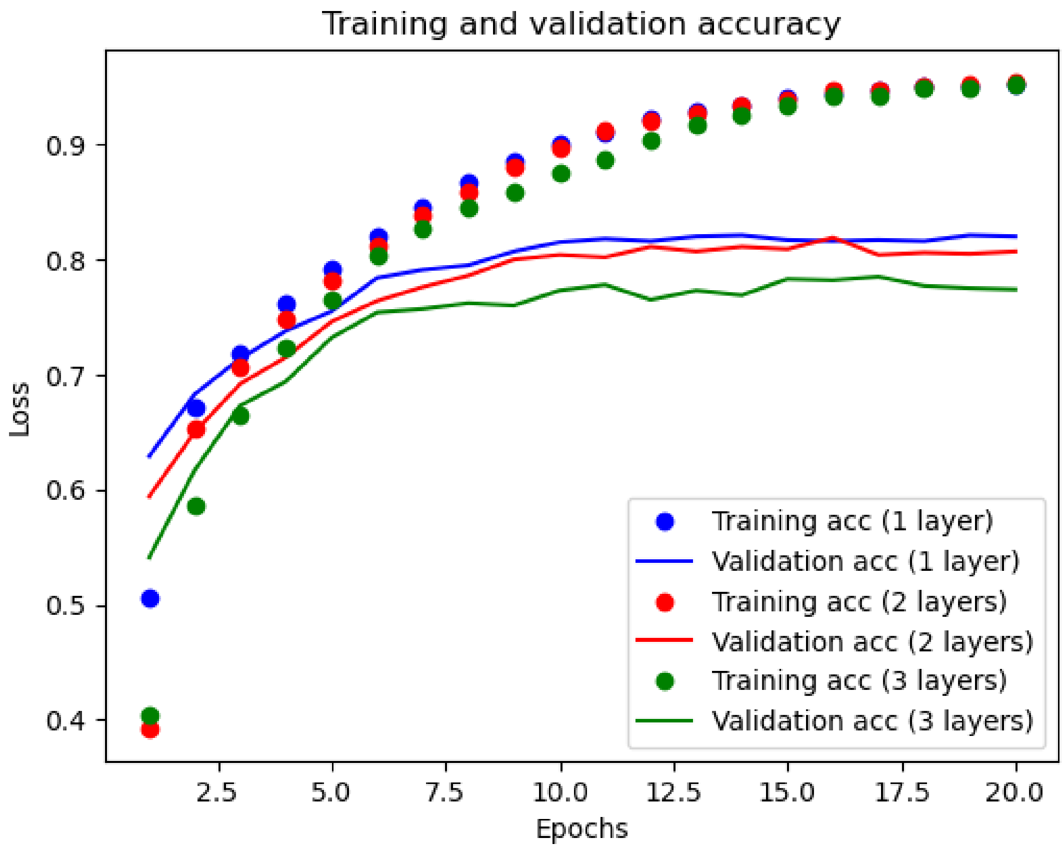
```
Epoch 19/20
16/16 [==============================] - 0s 8ms/step - loss: 0.1923 - acc:
0.9493 - val_loss: 1.2125 - val_acc: 0.7750
Epoch 20/20
16/16 [==============================] - 0s 9ms/step - loss: 0.1788 - acc:
0.9523 - val_loss: 1.2699 - val_acc: 0.7740
```

[ ]:
```python
loss_1 = history_1.history['loss']
val_loss_1 = history_1.history['val_loss']

loss_2 = history_2.history['loss']
val_loss_2 = history_2.history['val_loss']

loss_3 = history_3.history['loss']
val_loss_3 = history_3.history['val_loss']

epochs = range(1, len(loss_32) + 1)

plt.plot(epochs, loss_1, 'bo', label='Training loss (1 layer)')
plt.plot(epochs, val_loss_1, 'b', label='Validation loss (1 layer)')
plt.plot(epochs, loss_2, 'ro', label='Training loss (2 layers)')
plt.plot(epochs, val_loss_2, 'r', label='Validation loss (2 layers)')
plt.plot(epochs, loss_3, 'go', label='Training loss (3 layers)')
plt.plot(epochs, val_loss_3, 'g', label='Validation loss (3 layers)')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



[ ]:
```
plt.clf()

acc_1 = history_1.history['acc']
val_acc_1 = history_1.history['val_acc']

acc_2 = history_2.history['acc']
val_acc_2 = history_2.history['val_acc']

acc_3 = history_3.history['acc']
val_acc_3 = history_3.history['val_acc']

plt.plot(epochs, acc_1, 'bo', label='Training acc (1 layer)')
plt.plot(epochs, val_acc_1, 'b', label='Validation acc (1 layer)')
plt.plot(epochs, acc_2, 'ro', label='Training acc (2 layers)')
plt.plot(epochs, val_acc_2, 'r', label='Validation acc (2 layers)')
plt.plot(epochs, acc_3, 'go', label='Training acc (3 layers)')
plt.plot(epochs, val_acc_3, 'g', label='Validation acc (3 layers)')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Loss')
```

```
plt.legend()

plt.show()
```



0.0.3 Assignment 5.3

Section 3.6

```
[ ]: from keras.datasets import boston_housing

(train_data, train_targets), (test_data, test_targets) =  boston_housing.
 ↪load_data()

print(train_data.shape)
print(test_data.shape)
print(train_targets)
```

```
(404, 13)
(102, 13)
[15.2 42.3 50.  21.1 17.7 18.5 11.3 15.6 15.6 14.4 12.1 17.9 23.1 19.9
 15.7  8.8 50.  22.5 24.1 27.5 10.9 30.8 32.9 24.  18.5 13.3 22.9 34.7
```

```
16.6 17.5 22.3 16.1 14.9 23.1 34.9 25.  13.9 13.1 20.4 20.  15.2 24.7
22.2 16.7 12.7 15.6 18.4 21.  30.1 15.1 18.7  9.6 31.5 24.8 19.1 22.
14.5 11.  32.  29.4 20.3 24.4 14.6 19.5 14.1 14.3 15.6 10.5  6.3 19.3
19.3 13.4 36.4 17.8 13.5 16.5  8.3 14.3 16.  13.4 28.6 43.5 20.2 22.
23.  20.7 12.5 48.5 14.6 13.4 23.7 50.  21.7 39.8 38.7 22.2 34.9 22.5
31.1 28.7 46.  41.7 21.  26.6 15.  24.4 13.3 21.2 11.7 21.7 19.4 50.
22.8 19.7 24.7 36.2 14.2 18.9 18.3 20.6 24.6 18.2  8.7 44.  10.4 13.2
21.2 37.  30.7 22.9 20.  19.3 31.7 32.  23.1 18.8 10.9 50.  19.6  5.
14.4 19.8 13.8 19.6 23.9 24.5 25.  19.9 17.2 24.6 13.5 26.6 21.4 11.9
22.6 19.6  8.5 23.7 23.1 22.4 20.5 23.6 18.4 35.2 23.1 27.9 20.6 23.7
28.  13.6 27.1 23.6 20.6 18.2 21.7 17.1  8.4 25.3 13.8 22.2 18.4 20.7
31.6 30.5 20.3  8.8 19.2 19.4 23.1 23.  14.8 48.8 22.6 33.4 21.1 13.6
32.2 13.1 23.4 18.9 23.9 11.8 23.3 22.8 19.6 16.7 13.4 22.2 20.4 21.8
26.4 14.9 24.1 23.8 12.3 29.1 21.  19.5 23.3 23.8 17.8 11.5 21.7 19.9
25.  33.4 28.5 21.4 24.3 27.5 33.1 16.2 23.3 48.3 22.9 22.8 13.1 12.7
22.6 15.  15.3 10.5 24.  18.5 21.7 19.5 33.2 23.2  5.  19.1 12.7 22.3
10.2 13.9 16.3 17.  20.1 29.9 17.2 37.3 45.4 17.8 23.2 29.  22.  18.
17.4 34.6 20.1 25.  15.6 24.8 28.2 21.2 21.4 23.8 31.  26.2 17.4 37.9
17.5 20.   8.3 23.9  8.4 13.8  7.2 11.7 17.1 21.6 50.  16.1 20.4 20.6
21.4 20.6 36.5  8.5 24.8 10.8 21.9 17.3 18.9 36.2 14.9 18.2 33.3 21.8
19.7 31.6 24.8 19.4 22.8  7.5 44.8 16.8 18.7 50.  50.  19.5 20.1 50.
17.2 20.8 19.3 41.3 20.4 20.5 13.8 16.5 23.9 20.6 31.5 23.3 16.8 14.
33.8 36.1 12.8 18.3 18.7 19.1 29.  30.1 50.  50.  22.  11.9 37.6 50.
22.7 20.8 23.5 27.9 50.  19.3 23.9 22.6 15.2 21.7 19.2 43.8 20.3 33.2
19.9 22.5 32.7 22.  17.1 19.  15.  16.1 25.1 23.7 28.7 37.2 22.6 16.4
25.  29.8 22.1 17.4 18.1 30.3 17.5 24.7 12.6 26.5 28.7 13.3 10.4 24.4
23.  20.  17.8  7.  11.8 24.4 13.8 19.4 25.2 19.4 19.4 29.1]
```

[ ]:
```
mean = train_data.mean(axis=0)
train_data -= mean
std = train_data.std(axis=0)
train_data /= std

test_data -= mean
test_data /= std
```

[ ]:
```
def build_model():
    model = models.Sequential()
    model.add(layers.Dense(64, activation='relu', input_shape=(train_data.
 ↪shape[1],)))
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(1))
    model.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
    return model
```

[ ]:
```
k = 4
num_val_samples = len(train_data)  // k
```

```
num_epochs = 100
all_scores = []
```

[ ]:
```
for i in range(k):
    print('processing fold #', i)
    val_data = train_data[i * num_val_samples: (i + 1) * num_val_samples]
    val_targets = train_targets[i * num_val_samples: (i + 1) * num_val_samples]

    partial_train_data = np.concatenate([train_data[:i * num_val_samples],␣
 ↪train_data[(i + 1) * num_val_samples:]], axis=0)
    partial_train_targets = np.concatenate([train_targets[:i *␣
 ↪num_val_samples], train_targets[(i + 1) * num_val_samples:]], axis=0)

    model = build_model()

    model.fit(partial_train_data, partial_train_targets, epochs=num_epochs,␣
 ↪batch_size=1, verbose=0)

    val_mse, val_mae = model.evaluate(val_data, val_targets, verbose=0)
    all_scores.append(val_mae)
```

```
processing fold # 0
processing fold # 1
processing fold # 2
processing fold # 3
```

[ ]:
```
all_scores
```

[ ]:
```
[2.108128309249878, 2.6189517974853516, 2.4960572719573975, 2.7731285095214844]
```

[ ]:
```
np.mean(all_scores)
```

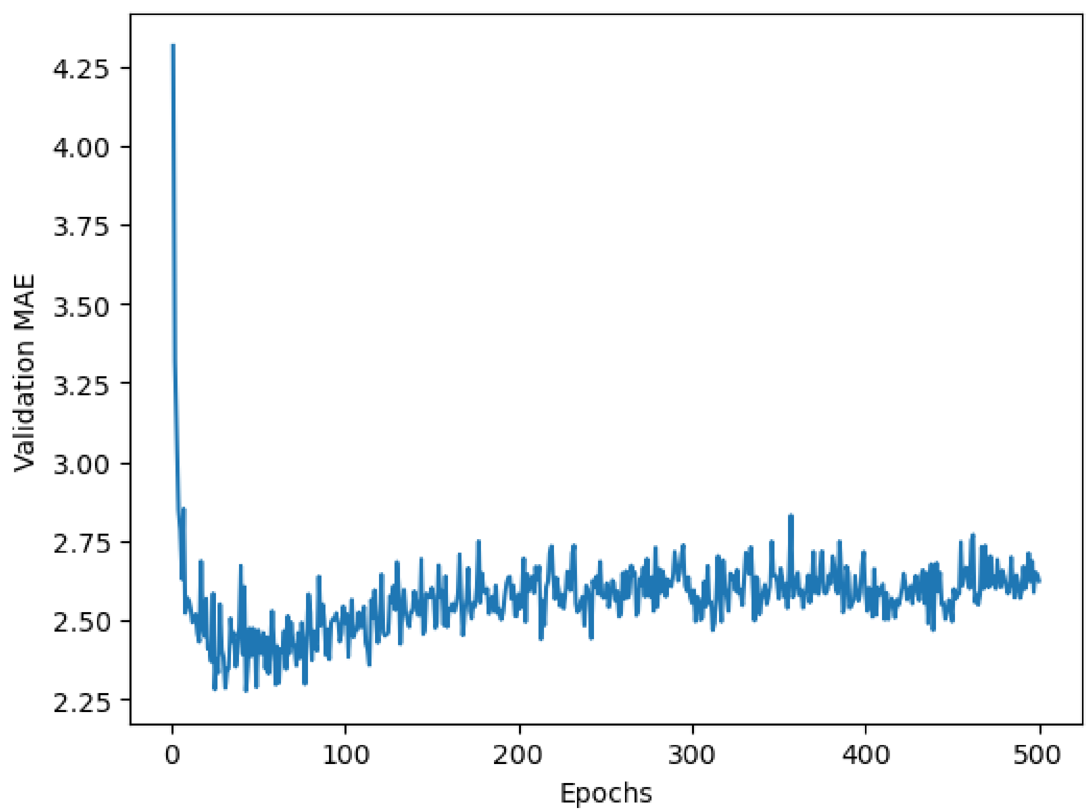
[ ]:
```
2.499066472053528
```

[ ]:
```
num_epochs = 500
all_mae_histories = []
for i in range(k):
    print('processing fold #', i)
    val_data = train_data[i * num_val_samples: (i + 1) * num_val_samples]
    val_targets = train_targets[i * num_val_samples: (i + 1) * num_val_samples]

    partial_train_data = np.concatenate([train_data[:i * num_val_samples],␣
 ↪train_data[(i + 1) * num_val_samples:]], axis=0)
    partial_train_targets = np.concatenate([train_targets[:i *␣
 ↪num_val_samples], train_targets[(i + 1) * num_val_samples:]], axis=0)
    model = build_model()
```

```
    history = model.fit(partial_train_data, partial_train_targets,␣
↪validation_data=(val_data, val_targets), epochs=num_epochs, batch_size=1,␣
↪verbose=0)
    mae_history = history.history['val_mae']
    all_mae_histories.append(mae_history)

    average_mae_history = [np.mean([x[i] for x in all_mae_histories]) for i in␣
↪range(num_epochs)]
```

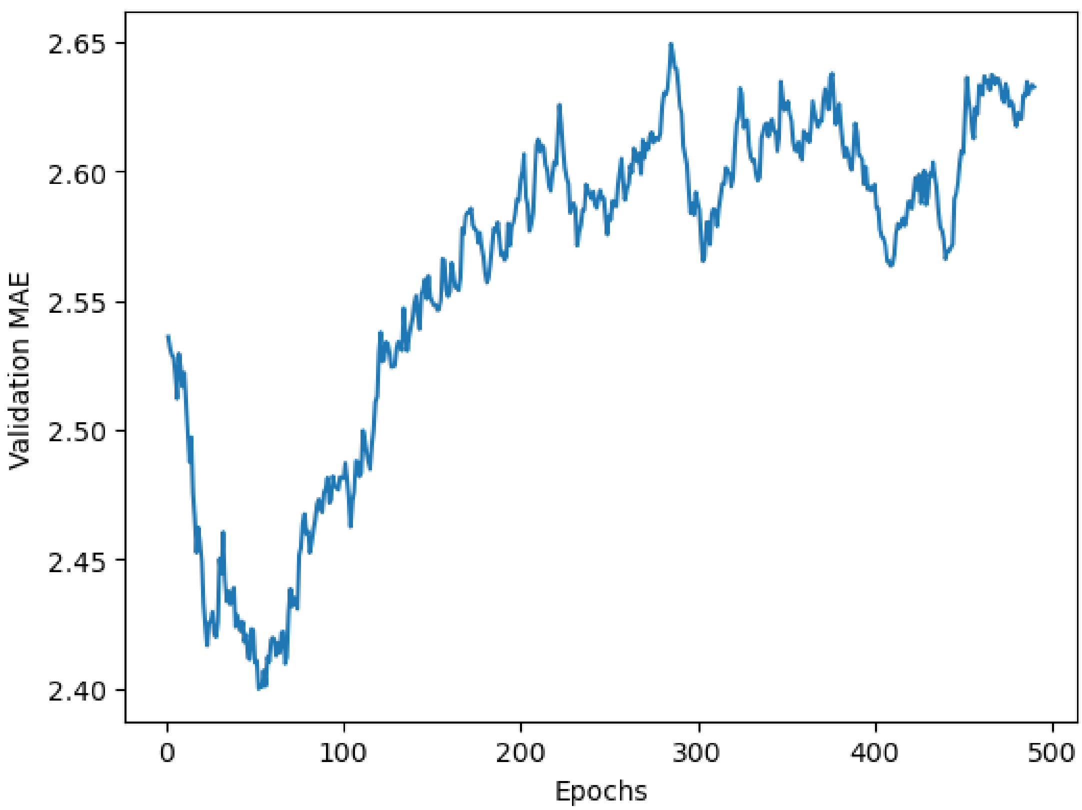
```
processing fold # 0
processing fold # 1
processing fold # 2
processing fold # 3
```

[ ]:
```
plt.plot(range(1, len(average_mae_history) + 1), average_mae_history)
plt.xlabel('Epochs')
plt.ylabel('Validation MAE')
plt.show()
```



[ ]:
```
def smooth_curve(points, factor=0.9):
    smoothed_points = []
```

```
  for point in points:
    if smoothed_points:
      previous = smoothed_points[-1]
      smoothed_points.append(previous * factor + point * (1 - factor))
    else:
      smoothed_points.append(point)
  return smoothed_points

smooth_mae_history = smooth_curve(average_mae_history[10:])

plt.plot(range(1, len(smooth_mae_history) + 1), smooth_mae_history)
plt.xlabel('Epochs')
plt.ylabel('Validation MAE')
plt.show()
```



[ ]:
```
model = build_model()
model.fit(train_data, train_targets, epochs=80, batch_size=16, verbose=0)

test_mae_score, test_mae_score = model.evaluate(test_data, test_targets)
test_mae_score
```

```
4/4 [==============================] - 0s 2ms/step - loss: 16.2238 - mae: 2.5546
```

```
[ ]: 2.5546085834503174
```