Assignment9_ErenAkgunduz

March 31, 2024

1 Assignment 9

1.1 Eren Akgunduz

1.1.1 Deep Learning — 31 March 2024

1.1.2 Link to notebook

[1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import to_categorical
from sklearn.model_selection import train_test_split
```

1.1.3 Helper functions

[3]:
```
def plot_curve(accuracy, loss, split: str):
    epochs=np.arange(loss.shape[0])
    plt.subplot(1,2,1)
    plt.plot(epochs, accuracy) # plt.axis([-1,2,-1,2])
    plt.xlabel('Epoch#')
    plt.ylabel('Accuracy')
    plt.title(f'{split} Accuracy')
    plt.subplot(1,2,2)
    plt.plot(epochs, loss)
    plt.xlabel('Epoch#')
    plt.ylabel('Binary crossentropy loss')
    plt.title(f'{split} loss')
    plt.show()
```

[4]:
```
def plot_display_metrics(model, model_history, split: str):
    match split:
        case "Training":
            accuracy = np.array(model_history.history["accuracy"])
            loss = np.array(model_history.history["loss"])
        case "Validation":
            accuracy = np.array(model_history.history["val_accuracy"])
            loss = np.array(model_history.history["val_loss"])
        case _:
            raise ValueError("Not a valid set for displaying metrics, try␣
 ↪'Training' or 'Validation'")

    plot_curve(accuracy, loss, split)

    score = [accuracy[-1], loss[-1]]
    print(f"\n{split} accuracy: {round(score[0], 4) * 100}%")
    print(f"{split} loss: {round(score[1], 3)}")
```

[5]:
```
def features_labels_split(data, label_column):
    features = data.copy()
    columns_titles = list(data. columns)
    features = features.reindex(columns=columns_titles)
    labels = features.pop(label_column)
    feature_names = list(features.columns)
    features = np.array(features)
    labels = np.array(labels)
    labels = labels.reshape((features.shape[0],1))
    return features, labels, feature_names

def features_labels_split_test(data, label_column, feature_names):
    features = data.copy()
    columns_titles = list(data.columns)
    features = features.reindex(columns=columns_titles)
    labels = features.pop(label_column)
    features_subset = np.zeros((labels.shape[0], 0))
    for column_i in feature_names:
        feature_i = np.expand_dims(features.pop(column_i), axis=1)
        features_subset = np.append(features_subset, feature_i, axis=1)
    labels = np.array(labels)
    labels = labels.reshape((features.shape[0],1))
    return features_subset, labels
```

1.1.4 Loading in data

[6]:
```
df = pd.read_csv("/content/drive/MyDrive/spotify_preprocessed.csv", sep=",")
df.head()
```

[6]:
```
   danceability    energy       key  loudness  mode  speechiness  \
0      0.738790  0.626533  0.090909  0.899432   0.0     0.070809
1      0.418807  0.247058  0.454545  0.687954   0.0     0.012962
2      0.530910  0.415269  0.818182  0.862211   0.0     0.031601
3      0.478668  0.648560  0.000000  0.880682   0.0     0.032351
4      0.810623  0.887860  0.090909  0.919516   1.0     0.270487

   acousticness  instrumentalness  liveness   valence     tempo  duration_ms  \
0      0.020080           0.00000  0.068476  0.723361  0.400098     0.093080
1      0.874498           0.81809  0.080700  0.256148  0.676658     0.086266
2      0.161647           0.00000  0.094582  0.280738  0.773251     0.103036
3      0.005151           0.00000  0.194033  0.298156  0.305743     0.095749
4      0.003825           0.00000  0.387755  0.799180  0.705958     0.067117

   time_signature  chorus_hit  sections  target
0             0.8    0.193225  0.093023     1.0
1             0.6    0.155665  0.081395     0.0
2             0.8    0.210605  0.081395     1.0
3             0.8    0.138515  0.058140     0.0
4             0.8    0.117248  0.069767     1.0
```

[7]:
```
data, labels, feature_names = features_labels_split(df, "target")
```

[8]:
```
data
```

[8]:
```
array([[0.73878973, 0.62653279, 0.09090909, …, 0.8       , 0.1932247 ,
        0.09302326],
       [0.41880714, 0.24705807, 0.45454545, …, 0.6       , 0.15566527,
        0.08139535],
       [0.53090988, 0.4152685 , 0.81818182, …, 0.8       , 0.21060483,
        0.08139535],
       …,
       [0.71484545, 0.80475575, 0.90909091, …, 0.8       , 0.09727058,
        0.05813953],
       [0.58532869, 0.17697039, 0.63636364, …, 0.8       , 0.10158341,
        0.13953488],
       [0.06399652, 0.12290275, 0.36363636, …, 0.8       , 0.33334162,
        0.15116279]])
```

[9]:
```
labels
```

[9]:
```
array([[1.],
       [0.],
```

```
        [1.],
        …,
        [1.],
        [0.],
        [0.]])
```

[10]:
```
feature_names
```

[10]:
```
['danceability',
 'energy',
 'key',
 'loudness',
 'mode',
 'speechiness',
 'acousticness',
 'instrumentalness',
 'liveness',
 'valence',
 'tempo',
 'duration_ms',
 'time_signature',
 'chorus_hit',
 'sections']
```

1.1.5 Splitting data

[11]:
```
X, X_test, y, y_test = train_test_split(data, labels, test_size=0.1,␣
↪shuffle=True)
```

[12]:
```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,␣
↪shuffle=True)
```

[13]:
```
X_train.shape
```

[13]:
```
(4606, 15)
```

[14]:
```
X_val.shape
```

[14]:
```
(1152, 15)
```

[15]:
```
X_test.shape
```

[15]:
```
(640, 15)
```

[16]:
```
y_train.shape
```

[16]:
```
(4606, 1)
```

[17]:
```
y_val.shape
```

[17]:
```
(1152, 1)
```

[18]:
```
y_test.shape
```

[18]:
```
(640, 1)
```

[19]:
```
data.shape
```

[19]:
```
(6398, 15)
```

[20]:
```
y_train.shape[0] + y_val.shape[0] + y_test.shape[0] == data.shape[0]
```

[20]:
```
True
```

1.1.6 Build the model

[21]:
```
model = Sequential()
model.add(Dense(input_dim=data.shape[1], units=32, activation="tanh"))
model.add(Dense(units=32, activation="tanh"))
model.add(Dense(units=1, activation="sigmoid"))
model.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 32)                512

 dense_1 (Dense)             (None, 32)                1056

 dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 1601 (6.25 KB)
Trainable params: 1601 (6.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[22]:
```
opt = tf.keras.optimizers.SGD(learning_rate=0.01)
model.compile(optimizer=opt,
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

1.1.7 Train the model

[23]:
```
history = model.fit(X_train, y_train, batch_size=16, epochs=50,␣
↪validation_data=(X_val, y_val))
```

```
Epoch 1/50
288/288 [==============================] - 1s 3ms/step - loss: 0.6649 -
accuracy: 0.6405 - val_loss: 0.6243 - val_accuracy: 0.7283
Epoch 2/50
288/288 [==============================] - 1s 2ms/step - loss: 0.6035 -
accuracy: 0.7284 - val_loss: 0.5682 - val_accuracy: 0.7500
Epoch 3/50
288/288 [==============================] - 1s 2ms/step - loss: 0.5557 -
accuracy: 0.7449 - val_loss: 0.5281 - val_accuracy: 0.7578
Epoch 4/50
288/288 [==============================] - 1s 2ms/step - loss: 0.5242 -
accuracy: 0.7607 - val_loss: 0.5040 - val_accuracy: 0.7587
Epoch 5/50
288/288 [==============================] - 1s 2ms/step - loss: 0.5045 -
accuracy: 0.7701 - val_loss: 0.4900 - val_accuracy: 0.7717
Epoch 6/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4913 -
accuracy: 0.7712 - val_loss: 0.4780 - val_accuracy: 0.7656
Epoch 7/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4839 -
accuracy: 0.7712 - val_loss: 0.4746 - val_accuracy: 0.7682
Epoch 8/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4777 -
accuracy: 0.7753 - val_loss: 0.4666 - val_accuracy: 0.7700
Epoch 9/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4739 -
accuracy: 0.7796 - val_loss: 0.4657 - val_accuracy: 0.7717
Epoch 10/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4696 -
accuracy: 0.7822 - val_loss: 0.4603 - val_accuracy: 0.7752
Epoch 11/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4672 -
accuracy: 0.7831 - val_loss: 0.4624 - val_accuracy: 0.7595
Epoch 12/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4662 -
accuracy: 0.7827 - val_loss: 0.4563 - val_accuracy: 0.7821
Epoch 13/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4626 -
accuracy: 0.7914 - val_loss: 0.4584 - val_accuracy: 0.7622
Epoch 14/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4612 -
accuracy: 0.7831 - val_loss: 0.4525 - val_accuracy: 0.7726
Epoch 15/50
```

```
288/288 [==============================] - 1s 2ms/step - loss: 0.4602 -
accuracy: 0.7846 - val_loss: 0.4578 - val_accuracy: 0.7734
Epoch 16/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4589 -
accuracy: 0.7872 - val_loss: 0.4495 - val_accuracy: 0.7778
Epoch 17/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4560 -
accuracy: 0.7894 - val_loss: 0.4536 - val_accuracy: 0.7795
Epoch 18/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4572 -
accuracy: 0.7890 - val_loss: 0.4470 - val_accuracy: 0.7786
Epoch 19/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4549 -
accuracy: 0.7898 - val_loss: 0.4465 - val_accuracy: 0.7752
Epoch 20/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4550 -
accuracy: 0.7922 - val_loss: 0.4466 - val_accuracy: 0.7752
Epoch 21/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4532 -
accuracy: 0.7911 - val_loss: 0.4448 - val_accuracy: 0.7847
Epoch 22/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4525 -
accuracy: 0.7935 - val_loss: 0.4432 - val_accuracy: 0.7786
Epoch 23/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4519 -
accuracy: 0.7966 - val_loss: 0.4425 - val_accuracy: 0.7847
Epoch 24/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4495 -
accuracy: 0.7948 - val_loss: 0.4409 - val_accuracy: 0.7830
Epoch 25/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4482 -
accuracy: 0.7940 - val_loss: 0.4463 - val_accuracy: 0.7812
Epoch 26/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4488 -
accuracy: 0.7903 - val_loss: 0.4439 - val_accuracy: 0.7812
Epoch 27/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4470 -
accuracy: 0.7979 - val_loss: 0.4412 - val_accuracy: 0.7882
Epoch 28/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4472 -
accuracy: 0.8009 - val_loss: 0.4489 - val_accuracy: 0.7769
Epoch 29/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4482 -
accuracy: 0.7972 - val_loss: 0.4390 - val_accuracy: 0.7839
Epoch 30/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4453 -
accuracy: 0.7957 - val_loss: 0.4375 - val_accuracy: 0.7882
Epoch 31/50
```

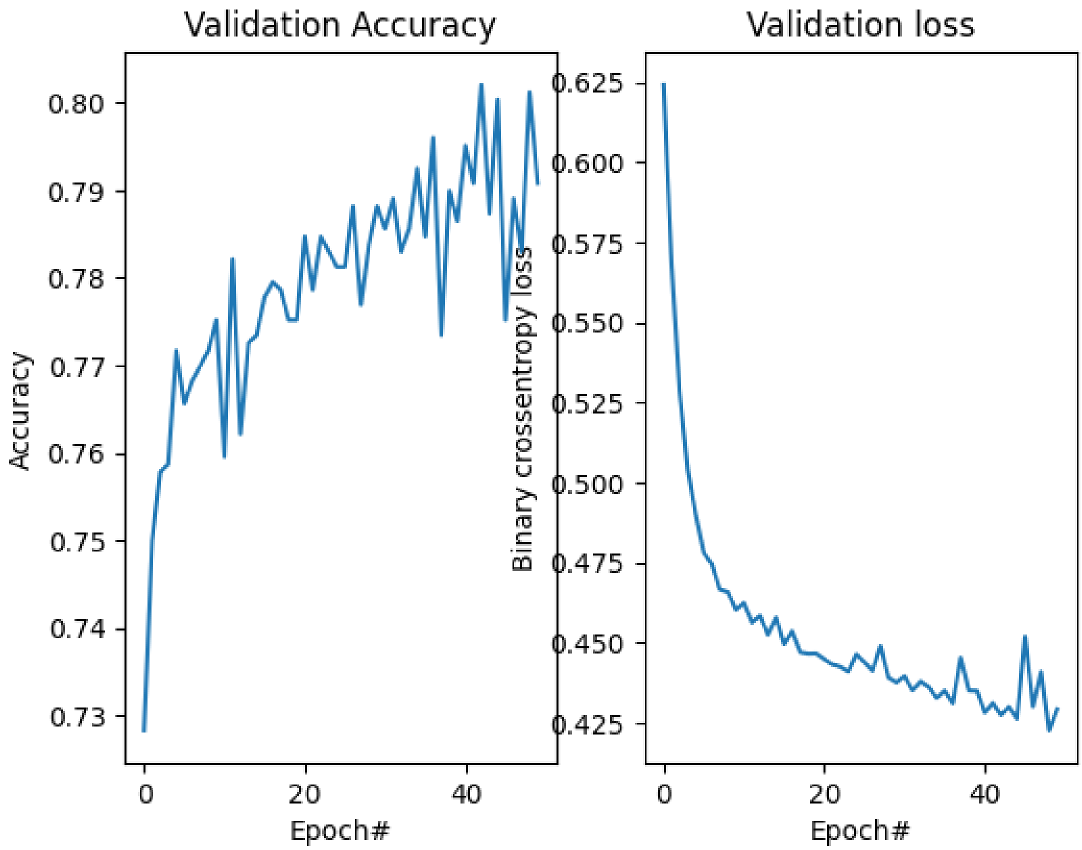
```
288/288 [==============================] - 1s 3ms/step - loss: 0.4458 -
accuracy: 0.8000 - val_loss: 0.4395 - val_accuracy: 0.7856
Epoch 32/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4452 -
accuracy: 0.7937 - val_loss: 0.4351 - val_accuracy: 0.7891
Epoch 33/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4434 -
accuracy: 0.7964 - val_loss: 0.4379 - val_accuracy: 0.7830
Epoch 34/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4447 -
accuracy: 0.7957 - val_loss: 0.4362 - val_accuracy: 0.7856
Epoch 35/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4421 -
accuracy: 0.7966 - val_loss: 0.4326 - val_accuracy: 0.7925
Epoch 36/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4424 -
accuracy: 0.8046 - val_loss: 0.4351 - val_accuracy: 0.7847
Epoch 37/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4416 -
accuracy: 0.8003 - val_loss: 0.4310 - val_accuracy: 0.7960
Epoch 38/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4387 -
accuracy: 0.8022 - val_loss: 0.4453 - val_accuracy: 0.7734
Epoch 39/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4404 -
accuracy: 0.8026 - val_loss: 0.4351 - val_accuracy: 0.7899
Epoch 40/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4411 -
accuracy: 0.8000 - val_loss: 0.4349 - val_accuracy: 0.7865
Epoch 41/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4392 -
accuracy: 0.7977 - val_loss: 0.4281 - val_accuracy: 0.7951
Epoch 42/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4375 -
accuracy: 0.8035 - val_loss: 0.4312 - val_accuracy: 0.7908
Epoch 43/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4381 -
accuracy: 0.8055 - val_loss: 0.4274 - val_accuracy: 0.8021
Epoch 44/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4357 -
accuracy: 0.8063 - val_loss: 0.4299 - val_accuracy: 0.7873
Epoch 45/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4365 -
accuracy: 0.8070 - val_loss: 0.4262 - val_accuracy: 0.8003
Epoch 46/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4354 -
accuracy: 0.8070 - val_loss: 0.4519 - val_accuracy: 0.7752
Epoch 47/50
```

```
288/288 [==============================] - 1s 2ms/step - loss: 0.4359 -
accuracy: 0.8013 - val_loss: 0.4300 - val_accuracy: 0.7891
Epoch 48/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4333 -
accuracy: 0.8059 - val_loss: 0.4410 - val_accuracy: 0.7830
Epoch 49/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4339 -
accuracy: 0.8057 - val_loss: 0.4226 - val_accuracy: 0.8012
Epoch 50/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4342 -
accuracy: 0.8042 - val_loss: 0.4292 - val_accuracy: 0.7908
```

[24]:
```python
plot_display_metrics(model, history, "Training")
```


```
Training accuracy: 80.42%
Training loss: 0.434
```

[25]:
```python
plot_display_metrics(model, history, "Validation")
```




```
Validation accuracy: 79.08%
Validation loss: 0.429
```

[26]:
```
train_score = model.evaluate(X_train, y_train)
print(f"\nTraining accuracy: {round(train_score[1], 4) * 100}%")
print(f"Training loss: {round(train_score[0], 3)}")
```

```
144/144 [==============================] - 0s 2ms/step - loss: 0.4312 -
accuracy: 0.8009

Training accuracy: 80.08999999999999%
Training loss: 0.431
```

[27]:
```
val_score = model.evaluate(X_val, y_val)
print(f"\nValidation accuracy: {round(val_score[1], 4) * 100}%")
print(f"Validation loss: {round(val_score[0], 3)}")
```

```
36/36 [==============================] - 0s 2ms/step - loss: 0.4292 - accuracy:
0.7908
```

```
Validation accuracy: 79.08%
Validation loss: 0.429
```

1.1.8 Trying out modifications to model

Take 1: Substitute SGD with Adam

[29]:
```
model_1 = Sequential()
model_1.add(Dense(input_dim=data.shape[1], units=32, activation="tanh"))
model_1.add(Dense(units=32, activation="tanh"))
model_1.add(Dense(units=1, activation="sigmoid"))
model_1.summary()
```

```
Model: "sequential_1"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 32)                512

 dense_4 (Dense)             (None, 32)                1056

 dense_5 (Dense)             (None, 1)                 33

=================================================================
Total params: 1601 (6.25 KB)
Trainable params: 1601 (6.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[30]:
```
opt = tf.keras.optimizers.Adam(learning_rate=0.0001)
model_1.compile(optimizer=opt,
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

[31]:
```
history = model_1.fit(X_train, y_train, batch_size=16, epochs=50,␣
 ↪validation_data=(X_val, y_val))
```

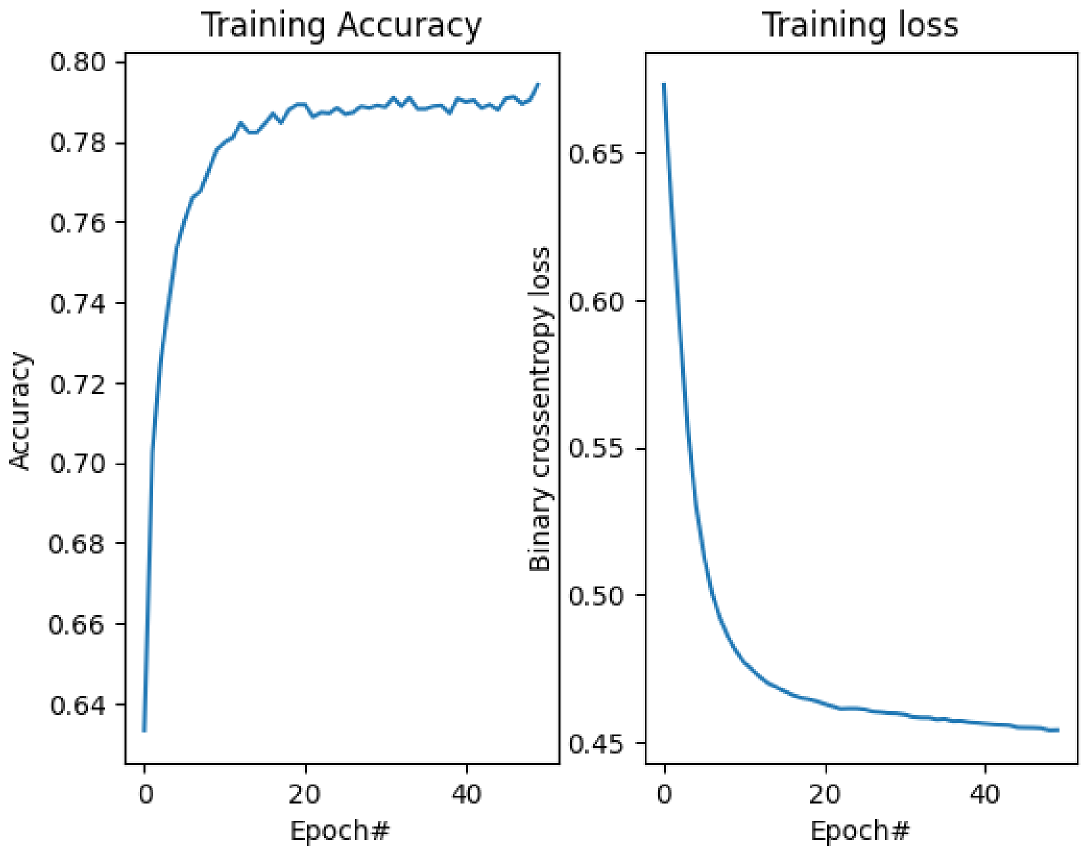
```
Epoch 1/50
288/288 [==============================] - 2s 3ms/step - loss: 0.6729 -
accuracy: 0.6333 - val_loss: 0.6482 - val_accuracy: 0.6927
Epoch 2/50
288/288 [==============================] - 1s 3ms/step - loss: 0.6304 -
accuracy: 0.7028 - val_loss: 0.6042 - val_accuracy: 0.7049
Epoch 3/50
288/288 [==============================] - 1s 3ms/step - loss: 0.5898 -
accuracy: 0.7247 - val_loss: 0.5644 - val_accuracy: 0.7405
Epoch 4/50
288/288 [==============================] - 1s 3ms/step - loss: 0.5555 -
accuracy: 0.7393 - val_loss: 0.5352 - val_accuracy: 0.7326
Epoch 5/50
```

```
288/288 [==============================] - 1s 3ms/step - loss: 0.5306 -
accuracy: 0.7534 - val_loss: 0.5136 - val_accuracy: 0.7465
Epoch 6/50
288/288 [==============================] - 1s 4ms/step - loss: 0.5131 -
accuracy: 0.7603 - val_loss: 0.4999 - val_accuracy: 0.7535
Epoch 7/50
288/288 [==============================] - 1s 2ms/step - loss: 0.5006 -
accuracy: 0.7660 - val_loss: 0.4902 - val_accuracy: 0.7613
Epoch 8/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4921 -
accuracy: 0.7677 - val_loss: 0.4847 - val_accuracy: 0.7587
Epoch 9/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4857 -
accuracy: 0.7727 - val_loss: 0.4789 - val_accuracy: 0.7726
Epoch 10/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4808 -
accuracy: 0.7779 - val_loss: 0.4760 - val_accuracy: 0.7648
Epoch 11/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4771 -
accuracy: 0.7799 - val_loss: 0.4743 - val_accuracy: 0.7665
Epoch 12/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4746 -
accuracy: 0.7809 - val_loss: 0.4706 - val_accuracy: 0.7700
Epoch 13/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4721 -
accuracy: 0.7846 - val_loss: 0.4690 - val_accuracy: 0.7700
Epoch 14/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4701 -
accuracy: 0.7822 - val_loss: 0.4672 - val_accuracy: 0.7717
Epoch 15/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4687 -
accuracy: 0.7822 - val_loss: 0.4666 - val_accuracy: 0.7734
Epoch 16/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4674 -
accuracy: 0.7844 - val_loss: 0.4658 - val_accuracy: 0.7717
Epoch 17/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4661 -
accuracy: 0.7870 - val_loss: 0.4662 - val_accuracy: 0.7726
Epoch 18/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4651 -
accuracy: 0.7846 - val_loss: 0.4634 - val_accuracy: 0.7726
Epoch 19/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4647 -
accuracy: 0.7879 - val_loss: 0.4632 - val_accuracy: 0.7743
Epoch 20/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4640 -
accuracy: 0.7892 - val_loss: 0.4624 - val_accuracy: 0.7726
Epoch 21/50
```

```
288/288 [==============================] - 1s 3ms/step - loss: 0.4630 -
accuracy: 0.7892 - val_loss: 0.4617 - val_accuracy: 0.7691
Epoch 22/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4622 -
accuracy: 0.7861 - val_loss: 0.4620 - val_accuracy: 0.7656
Epoch 23/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4614 -
accuracy: 0.7872 - val_loss: 0.4608 - val_accuracy: 0.7743
Epoch 24/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4615 -
accuracy: 0.7870 - val_loss: 0.4613 - val_accuracy: 0.7743
Epoch 25/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4615 -
accuracy: 0.7883 - val_loss: 0.4605 - val_accuracy: 0.7769
Epoch 26/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4612 -
accuracy: 0.7868 - val_loss: 0.4597 - val_accuracy: 0.7734
Epoch 27/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4605 -
accuracy: 0.7872 - val_loss: 0.4626 - val_accuracy: 0.7700
Epoch 28/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4603 -
accuracy: 0.7888 - val_loss: 0.4604 - val_accuracy: 0.7769
Epoch 29/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4599 -
accuracy: 0.7883 - val_loss: 0.4590 - val_accuracy: 0.7752
Epoch 30/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4598 -
accuracy: 0.7890 - val_loss: 0.4589 - val_accuracy: 0.7786
Epoch 31/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4594 -
accuracy: 0.7885 - val_loss: 0.4582 - val_accuracy: 0.7743
Epoch 32/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4586 -
accuracy: 0.7909 - val_loss: 0.4579 - val_accuracy: 0.7734
Epoch 33/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4584 -
accuracy: 0.7888 - val_loss: 0.4575 - val_accuracy: 0.7769
Epoch 34/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4584 -
accuracy: 0.7909 - val_loss: 0.4578 - val_accuracy: 0.7786
Epoch 35/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4578 -
accuracy: 0.7881 - val_loss: 0.4587 - val_accuracy: 0.7760
Epoch 36/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4580 -
accuracy: 0.7881 - val_loss: 0.4566 - val_accuracy: 0.7734
Epoch 37/50
```

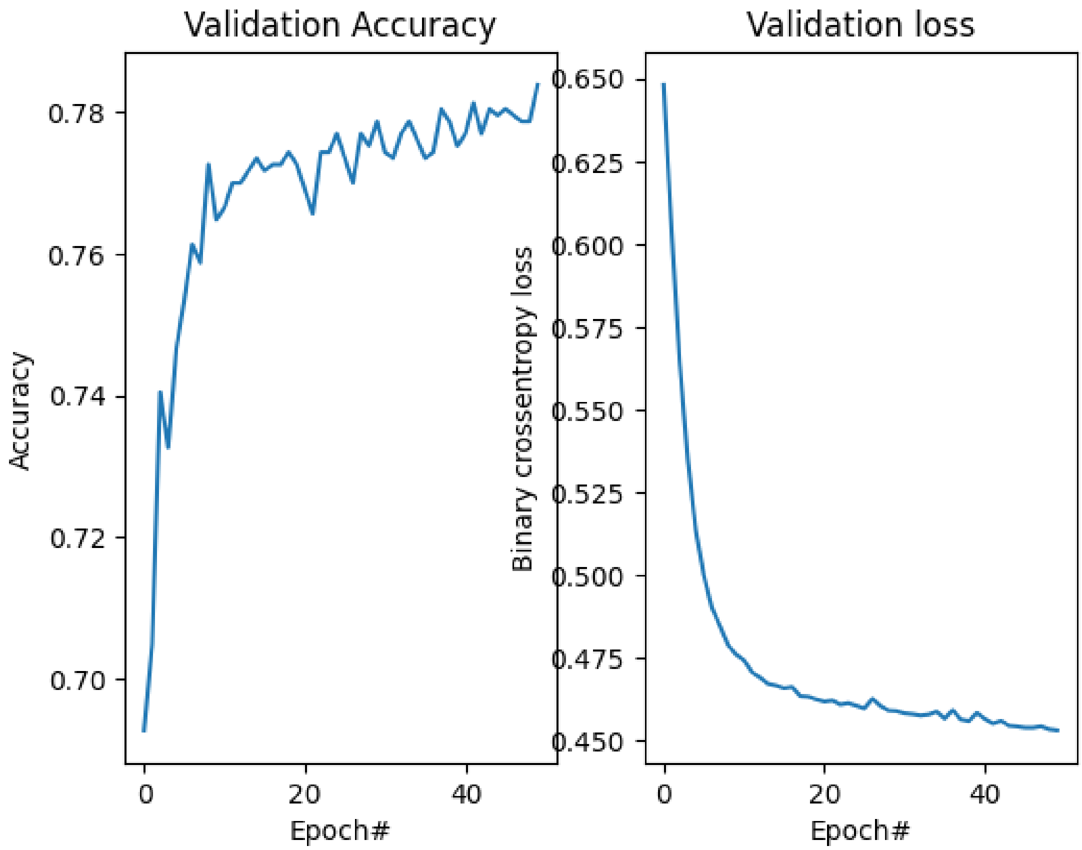
```
288/288 [==============================] - 1s 2ms/step - loss: 0.4572 -
accuracy: 0.7888 - val_loss: 0.4591 - val_accuracy: 0.7743
Epoch 38/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4573 -
accuracy: 0.7890 - val_loss: 0.4563 - val_accuracy: 0.7804
Epoch 39/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4568 -
accuracy: 0.7870 - val_loss: 0.4558 - val_accuracy: 0.7786
Epoch 40/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4566 -
accuracy: 0.7907 - val_loss: 0.4583 - val_accuracy: 0.7752
Epoch 41/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4564 -
accuracy: 0.7898 - val_loss: 0.4565 - val_accuracy: 0.7769
Epoch 42/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4561 -
accuracy: 0.7903 - val_loss: 0.4550 - val_accuracy: 0.7812
Epoch 43/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4560 -
accuracy: 0.7883 - val_loss: 0.4559 - val_accuracy: 0.7769
Epoch 44/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4558 -
accuracy: 0.7892 - val_loss: 0.4544 - val_accuracy: 0.7804
Epoch 45/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4551 -
accuracy: 0.7879 - val_loss: 0.4542 - val_accuracy: 0.7795
Epoch 46/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4550 -
accuracy: 0.7907 - val_loss: 0.4538 - val_accuracy: 0.7804
Epoch 47/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4550 -
accuracy: 0.7911 - val_loss: 0.4538 - val_accuracy: 0.7795
Epoch 48/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4548 -
accuracy: 0.7894 - val_loss: 0.4543 - val_accuracy: 0.7786
Epoch 49/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4541 -
accuracy: 0.7903 - val_loss: 0.4533 - val_accuracy: 0.7786
Epoch 50/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4542 -
accuracy: 0.7942 - val_loss: 0.4530 - val_accuracy: 0.7839
```

[32]:
```
plot_display_metrics(model_1, history, "Training")
```




```
Training accuracy: 79.42%
Training loss: 0.454
```

[33]:
```
plot_display_metrics(model_1, history, "Validation")
```




```
Validation accuracy: 78.39%
Validation loss: 0.453
```

Take 2: Double the units of first layer

[34]:
```python
model_2 = Sequential()
model_2.add(Dense(input_dim=data.shape[1], units=64, activation="tanh"))
model_2.add(Dense(units=32, activation="tanh"))
model_2.add(Dense(units=1, activation="sigmoid"))
model_2.summary()
```

```
Model: "sequential_2"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_6 (Dense)             (None, 64)                1024

 dense_7 (Dense)             (None, 32)                2080

 dense_8 (Dense)             (None, 1)                 33
```

```
=================================================================
Total params: 3137 (12.25 KB)
Trainable params: 3137 (12.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[35]:
```
opt = tf.keras.optimizers.SGD(learning_rate=0.01)
model_2.compile(optimizer=opt,
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

[36]:
```
history = model_2.fit(X_train, y_train, batch_size=16, epochs=50,␣
 ↪validation_data=(X_val, y_val))
```

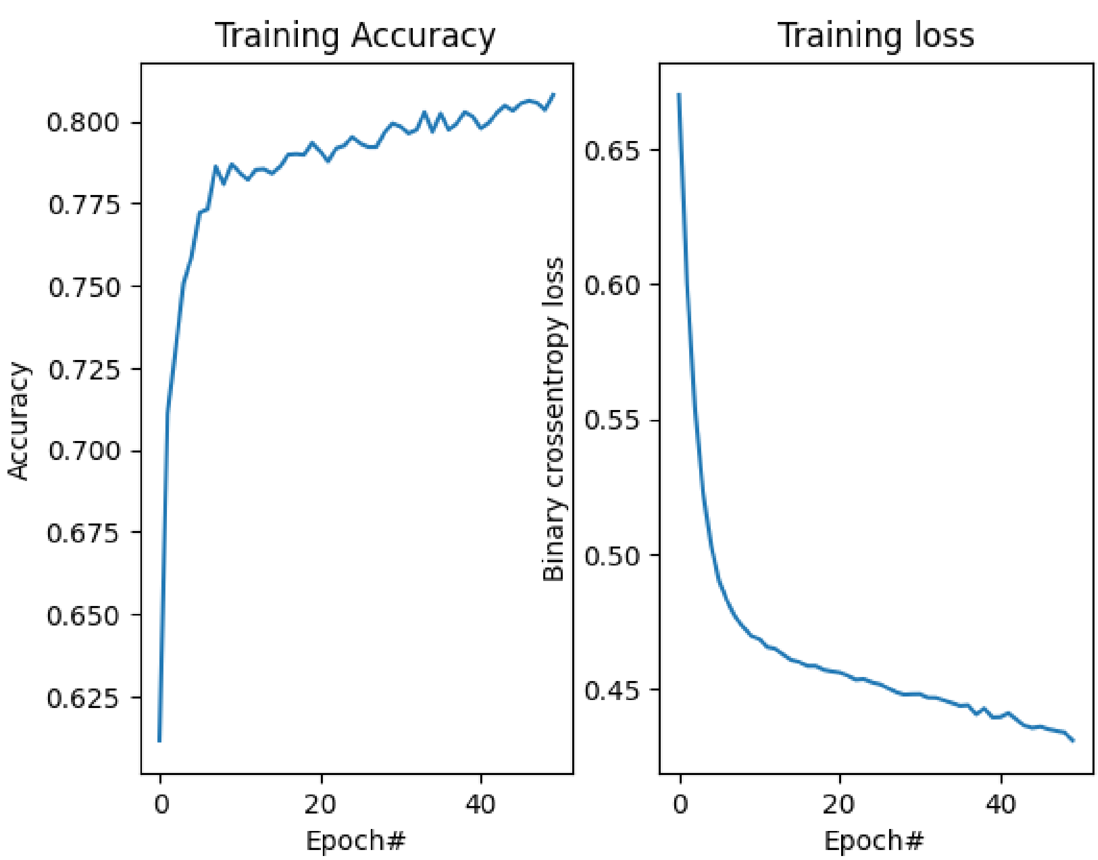
```
Epoch 1/50
288/288 [==============================] - 1s 3ms/step - loss: 0.6702 -
accuracy: 0.6116 - val_loss: 0.6287 - val_accuracy: 0.6892
Epoch 2/50
288/288 [==============================] - 1s 2ms/step - loss: 0.6006 -
accuracy: 0.7110 - val_loss: 0.5699 - val_accuracy: 0.7005
Epoch 3/50
288/288 [==============================] - 1s 2ms/step - loss: 0.5538 -
accuracy: 0.7304 - val_loss: 0.5270 - val_accuracy: 0.7352
Epoch 4/50
288/288 [==============================] - 1s 2ms/step - loss: 0.5227 -
accuracy: 0.7508 - val_loss: 0.5040 - val_accuracy: 0.7448
Epoch 5/50
288/288 [==============================] - 1s 2ms/step - loss: 0.5033 -
accuracy: 0.7588 - val_loss: 0.4884 - val_accuracy: 0.7656
Epoch 6/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4900 -
accuracy: 0.7723 - val_loss: 0.4813 - val_accuracy: 0.7622
Epoch 7/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4826 -
accuracy: 0.7733 - val_loss: 0.4843 - val_accuracy: 0.7561
Epoch 8/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4767 -
accuracy: 0.7864 - val_loss: 0.4714 - val_accuracy: 0.7682
Epoch 9/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4730 -
accuracy: 0.7809 - val_loss: 0.4677 - val_accuracy: 0.7587
Epoch 10/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4697 -
accuracy: 0.7870 - val_loss: 0.4640 - val_accuracy: 0.7786
Epoch 11/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4685 -
accuracy: 0.7844 - val_loss: 0.4644 - val_accuracy: 0.7726
```

```
Epoch 12/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4656 -
accuracy: 0.7822 - val_loss: 0.4600 - val_accuracy: 0.7804
Epoch 13/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4649 -
accuracy: 0.7853 - val_loss: 0.4609 - val_accuracy: 0.7752
Epoch 14/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4627 -
accuracy: 0.7855 - val_loss: 0.4560 - val_accuracy: 0.7743
Epoch 15/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4608 -
accuracy: 0.7842 - val_loss: 0.4665 - val_accuracy: 0.7691
Epoch 16/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4600 -
accuracy: 0.7861 - val_loss: 0.4531 - val_accuracy: 0.7778
Epoch 17/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4586 -
accuracy: 0.7898 - val_loss: 0.4520 - val_accuracy: 0.7752
Epoch 18/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4585 -
accuracy: 0.7901 - val_loss: 0.4516 - val_accuracy: 0.7656
Epoch 19/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4572 -
accuracy: 0.7898 - val_loss: 0.4515 - val_accuracy: 0.7804
Epoch 20/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4566 -
accuracy: 0.7935 - val_loss: 0.4507 - val_accuracy: 0.7821
Epoch 21/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4562 -
accuracy: 0.7909 - val_loss: 0.4521 - val_accuracy: 0.7821
Epoch 22/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4550 -
accuracy: 0.7879 - val_loss: 0.4493 - val_accuracy: 0.7778
Epoch 23/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4535 -
accuracy: 0.7918 - val_loss: 0.4588 - val_accuracy: 0.7561
Epoch 24/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4537 -
accuracy: 0.7927 - val_loss: 0.4522 - val_accuracy: 0.7839
Epoch 25/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4525 -
accuracy: 0.7953 - val_loss: 0.4450 - val_accuracy: 0.7786
Epoch 26/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4518 -
accuracy: 0.7933 - val_loss: 0.4432 - val_accuracy: 0.7769
Epoch 27/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4504 -
accuracy: 0.7922 - val_loss: 0.4534 - val_accuracy: 0.7778
```

```
Epoch 28/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4489 -
accuracy: 0.7922 - val_loss: 0.4456 - val_accuracy: 0.7769
Epoch 29/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4480 -
accuracy: 0.7966 - val_loss: 0.4427 - val_accuracy: 0.7847
Epoch 30/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4481 -
accuracy: 0.7994 - val_loss: 0.4434 - val_accuracy: 0.7830
Epoch 31/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4482 -
accuracy: 0.7985 - val_loss: 0.4386 - val_accuracy: 0.7856
Epoch 32/50
288/288 [==============================] - 2s 8ms/step - loss: 0.4468 -
accuracy: 0.7964 - val_loss: 0.4425 - val_accuracy: 0.7769
Epoch 33/50
288/288 [==============================] - 3s 9ms/step - loss: 0.4468 -
accuracy: 0.7974 - val_loss: 0.4421 - val_accuracy: 0.7795
Epoch 34/50
288/288 [==============================] - 2s 5ms/step - loss: 0.4458 -
accuracy: 0.8029 - val_loss: 0.4366 - val_accuracy: 0.7908
Epoch 35/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4448 -
accuracy: 0.7968 - val_loss: 0.4386 - val_accuracy: 0.7830
Epoch 36/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4438 -
accuracy: 0.8024 - val_loss: 0.4356 - val_accuracy: 0.7908
Epoch 37/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4440 -
accuracy: 0.7974 - val_loss: 0.4381 - val_accuracy: 0.7847
Epoch 38/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4406 -
accuracy: 0.7994 - val_loss: 0.4375 - val_accuracy: 0.7873
Epoch 39/50
288/288 [==============================] - 2s 5ms/step - loss: 0.4429 -
accuracy: 0.8029 - val_loss: 0.4368 - val_accuracy: 0.7847
Epoch 40/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4395 -
accuracy: 0.8016 - val_loss: 0.4412 - val_accuracy: 0.7821
Epoch 41/50
288/288 [==============================] - 1s 5ms/step - loss: 0.4396 -
accuracy: 0.7979 - val_loss: 0.4323 - val_accuracy: 0.7934
Epoch 42/50
288/288 [==============================] - 3s 12ms/step - loss: 0.4411 -
accuracy: 0.7996 - val_loss: 0.4303 - val_accuracy: 0.7951
Epoch 43/50
288/288 [==============================] - 2s 7ms/step - loss: 0.4387 -
accuracy: 0.8026 - val_loss: 0.4331 - val_accuracy: 0.7873
```

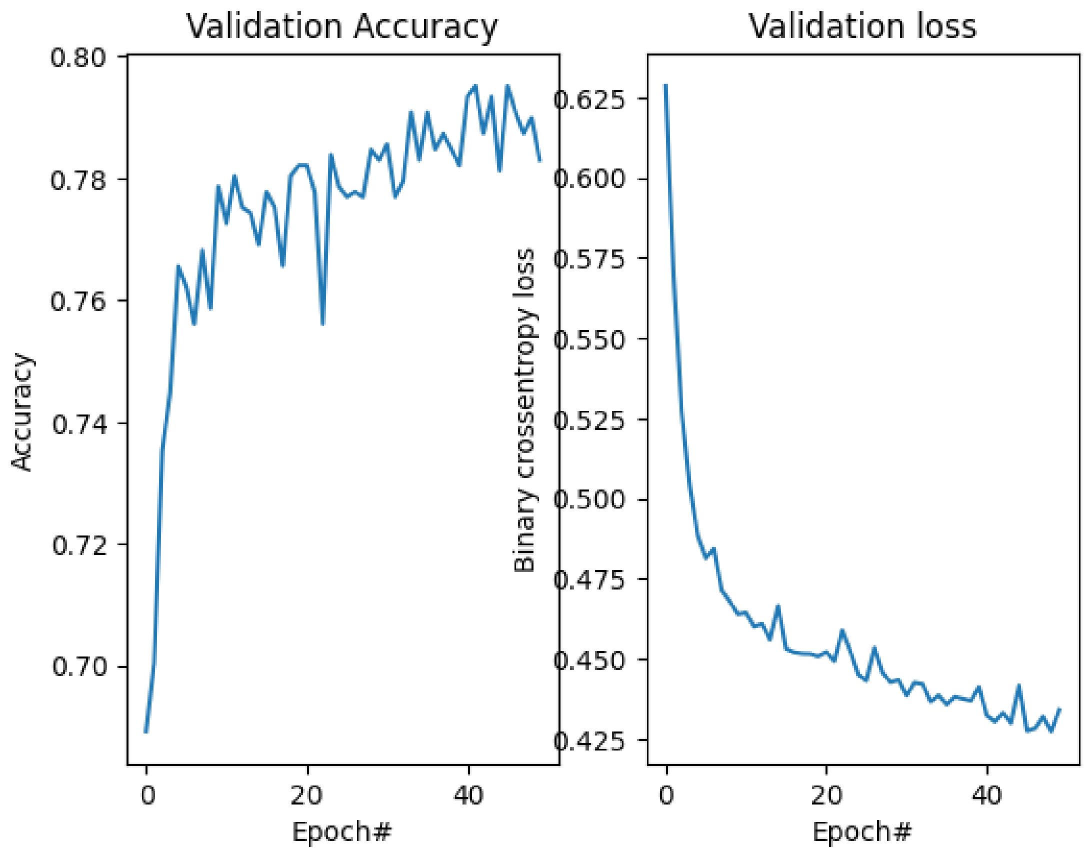
```
Epoch 44/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4365 -
accuracy: 0.8048 - val_loss: 0.4299 - val_accuracy: 0.7934
Epoch 45/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4356 -
accuracy: 0.8033 - val_loss: 0.4417 - val_accuracy: 0.7812
Epoch 46/50
288/288 [==============================] - 1s 5ms/step - loss: 0.4361 -
accuracy: 0.8055 - val_loss: 0.4275 - val_accuracy: 0.7951
Epoch 47/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4351 -
accuracy: 0.8063 - val_loss: 0.4282 - val_accuracy: 0.7908
Epoch 48/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4345 -
accuracy: 0.8057 - val_loss: 0.4320 - val_accuracy: 0.7873
Epoch 49/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4339 -
accuracy: 0.8035 - val_loss: 0.4273 - val_accuracy: 0.7899
Epoch 50/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4310 -
accuracy: 0.8081 - val_loss: 0.4340 - val_accuracy: 0.7830
```

[37]:
```
plot_display_metrics(model_2, history, "Training")
```



```
Training accuracy: 80.81%
Training loss: 0.431
```

[38]:
```python
plot_display_metrics(model_2, history, "Validation")
```




```
Validation accuracy: 78.3%
Validation loss: 0.434
```

Take 3: Combine the two approaches above

[39]:
```python
model_3 = Sequential()
model_3.add(Dense(input_dim=data.shape[1], units=64, activation="tanh"))
model_3.add(Dense(units=32, activation="tanh"))
model_3.add(Dense(units=1, activation="sigmoid"))
model_3.summary()
```

```
Model: "sequential_3"
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_9 (Dense)             (None, 64)                1024

 dense_10 (Dense)            (None, 32)                2080

 dense_11 (Dense)            (None, 1)                 33

=================================================================
Total params: 3137 (12.25 KB)
Trainable params: 3137 (12.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[40]:
```python
opt = tf.keras.optimizers.Adam(learning_rate=0.0001)
model_3.compile(optimizer=opt,
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

[42]:
```python
history = model_3.fit(X_train, y_train, batch_size=16, epochs=50,␣
 ↪validation_data=(X_val, y_val))
```

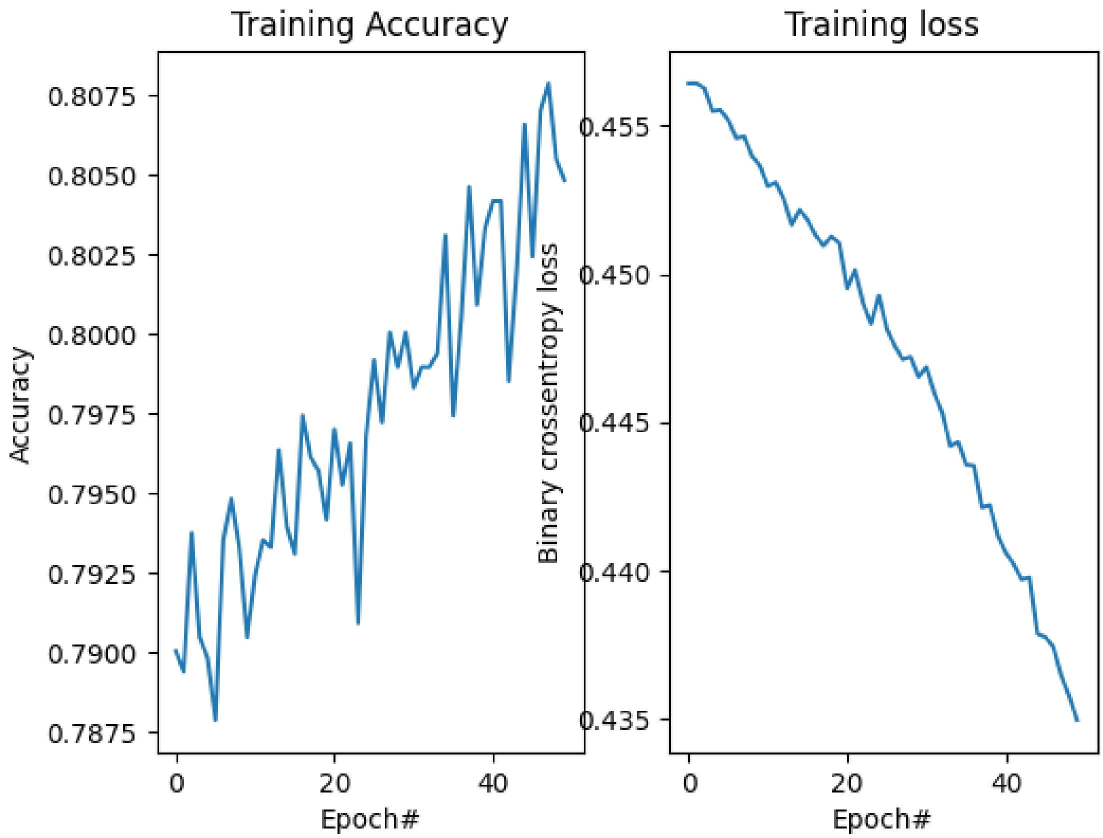
```
Epoch 1/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4564 -
accuracy: 0.7901 - val_loss: 0.4540 - val_accuracy: 0.7769
Epoch 2/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4564 -
accuracy: 0.7894 - val_loss: 0.4552 - val_accuracy: 0.7778
Epoch 3/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4563 -
accuracy: 0.7937 - val_loss: 0.4527 - val_accuracy: 0.7769
Epoch 4/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4555 -
accuracy: 0.7905 - val_loss: 0.4522 - val_accuracy: 0.7804
Epoch 5/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4555 -
accuracy: 0.7898 - val_loss: 0.4526 - val_accuracy: 0.7778
Epoch 6/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4552 -
accuracy: 0.7879 - val_loss: 0.4524 - val_accuracy: 0.7795
Epoch 7/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4546 -
accuracy: 0.7935 - val_loss: 0.4532 - val_accuracy: 0.7795
Epoch 8/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4546 -
accuracy: 0.7948 - val_loss: 0.4512 - val_accuracy: 0.7847
Epoch 9/50
```

```
288/288 [==============================] - 1s 3ms/step - loss: 0.4540 -
accuracy: 0.7933 - val_loss: 0.4510 - val_accuracy: 0.7821
Epoch 10/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4536 -
accuracy: 0.7905 - val_loss: 0.4503 - val_accuracy: 0.7804
Epoch 11/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4530 -
accuracy: 0.7924 - val_loss: 0.4538 - val_accuracy: 0.7795
Epoch 12/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4531 -
accuracy: 0.7935 - val_loss: 0.4495 - val_accuracy: 0.7821
Epoch 13/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4525 -
accuracy: 0.7933 - val_loss: 0.4492 - val_accuracy: 0.7839
Epoch 14/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4517 -
accuracy: 0.7964 - val_loss: 0.4487 - val_accuracy: 0.7847
Epoch 15/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4522 -
accuracy: 0.7940 - val_loss: 0.4487 - val_accuracy: 0.7839
Epoch 16/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4518 -
accuracy: 0.7931 - val_loss: 0.4485 - val_accuracy: 0.7839
Epoch 17/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4513 -
accuracy: 0.7974 - val_loss: 0.4477 - val_accuracy: 0.7839
Epoch 18/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4510 -
accuracy: 0.7961 - val_loss: 0.4473 - val_accuracy: 0.7847
Epoch 19/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4513 -
accuracy: 0.7957 - val_loss: 0.4526 - val_accuracy: 0.7795
Epoch 20/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4510 -
accuracy: 0.7942 - val_loss: 0.4508 - val_accuracy: 0.7830
Epoch 21/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4495 -
accuracy: 0.7970 - val_loss: 0.4465 - val_accuracy: 0.7856
Epoch 22/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4501 -
accuracy: 0.7953 - val_loss: 0.4460 - val_accuracy: 0.7856
Epoch 23/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4490 -
accuracy: 0.7966 - val_loss: 0.4456 - val_accuracy: 0.7873
Epoch 24/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4483 -
accuracy: 0.7909 - val_loss: 0.4566 - val_accuracy: 0.7708
Epoch 25/50
```

```
288/288 [==============================] - 1s 3ms/step - loss: 0.4493 -
accuracy: 0.7968 - val_loss: 0.4449 - val_accuracy: 0.7856
Epoch 26/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4481 -
accuracy: 0.7992 - val_loss: 0.4463 - val_accuracy: 0.7839
Epoch 27/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4476 -
accuracy: 0.7972 - val_loss: 0.4446 - val_accuracy: 0.7865
Epoch 28/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4471 -
accuracy: 0.8000 - val_loss: 0.4440 - val_accuracy: 0.7847
Epoch 29/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4472 -
accuracy: 0.7990 - val_loss: 0.4432 - val_accuracy: 0.7847
Epoch 30/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4465 -
accuracy: 0.8000 - val_loss: 0.4435 - val_accuracy: 0.7873
Epoch 31/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4469 -
accuracy: 0.7983 - val_loss: 0.4428 - val_accuracy: 0.7882
Epoch 32/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4460 -
accuracy: 0.7990 - val_loss: 0.4422 - val_accuracy: 0.7856
Epoch 33/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4453 -
accuracy: 0.7990 - val_loss: 0.4411 - val_accuracy: 0.7856
Epoch 34/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4442 -
accuracy: 0.7994 - val_loss: 0.4411 - val_accuracy: 0.7873
Epoch 35/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4443 -
accuracy: 0.8031 - val_loss: 0.4401 - val_accuracy: 0.7882
Epoch 36/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4436 -
accuracy: 0.7974 - val_loss: 0.4396 - val_accuracy: 0.7882
Epoch 37/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4435 -
accuracy: 0.8005 - val_loss: 0.4391 - val_accuracy: 0.7891
Epoch 38/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4421 -
accuracy: 0.8046 - val_loss: 0.4424 - val_accuracy: 0.7882
Epoch 39/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4422 -
accuracy: 0.8009 - val_loss: 0.4380 - val_accuracy: 0.7899
Epoch 40/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4412 -
accuracy: 0.8033 - val_loss: 0.4374 - val_accuracy: 0.7882
Epoch 41/50
```

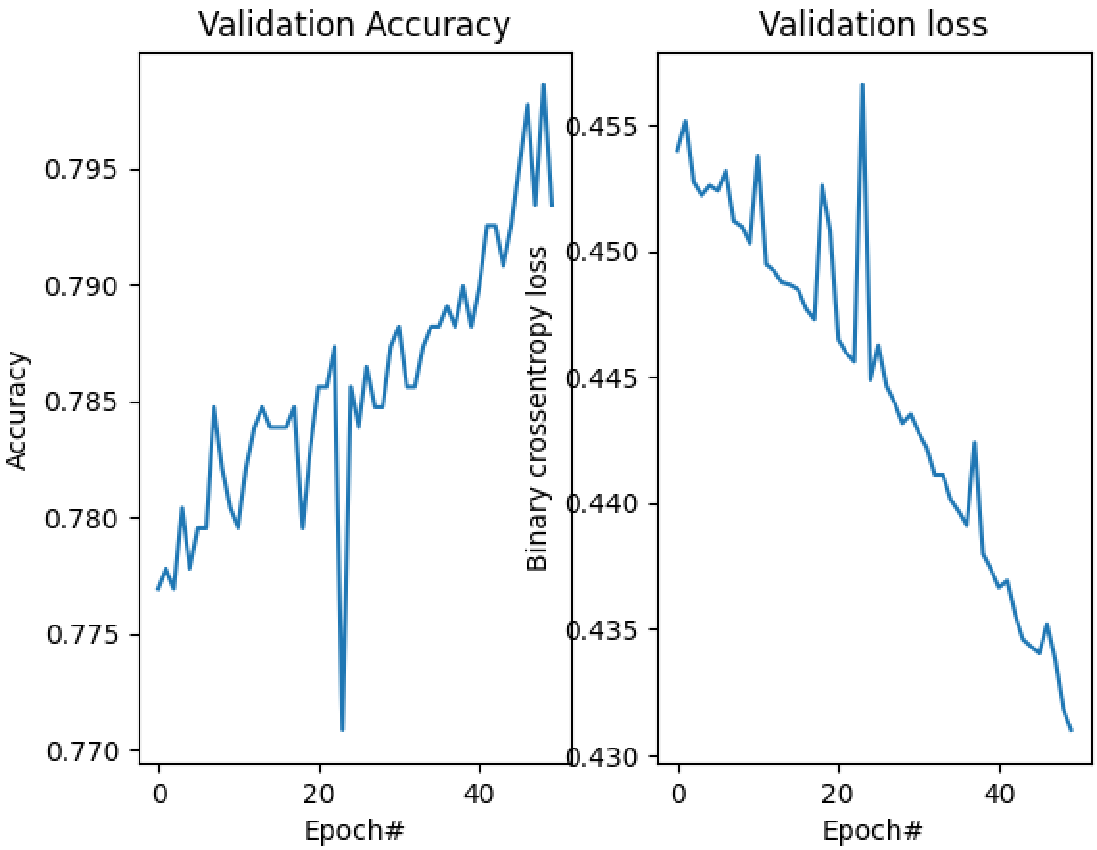
```
288/288 [==============================] - 1s 2ms/step - loss: 0.4406 -
accuracy: 0.8042 - val_loss: 0.4366 - val_accuracy: 0.7899
Epoch 42/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4402 -
accuracy: 0.8042 - val_loss: 0.4369 - val_accuracy: 0.7925
Epoch 43/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4397 -
accuracy: 0.7985 - val_loss: 0.4356 - val_accuracy: 0.7925
Epoch 44/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4398 -
accuracy: 0.8020 - val_loss: 0.4346 - val_accuracy: 0.7908
Epoch 45/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4379 -
accuracy: 0.8066 - val_loss: 0.4343 - val_accuracy: 0.7925
Epoch 46/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4377 -
accuracy: 0.8024 - val_loss: 0.4340 - val_accuracy: 0.7951
Epoch 47/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4374 -
accuracy: 0.8070 - val_loss: 0.4352 - val_accuracy: 0.7977
Epoch 48/50
288/288 [==============================] - 1s 4ms/step - loss: 0.4364 -
accuracy: 0.8079 - val_loss: 0.4338 - val_accuracy: 0.7934
Epoch 49/50
288/288 [==============================] - 1s 3ms/step - loss: 0.4357 -
accuracy: 0.8055 - val_loss: 0.4318 - val_accuracy: 0.7986
Epoch 50/50
288/288 [==============================] - 1s 2ms/step - loss: 0.4349 -
accuracy: 0.8048 - val_loss: 0.4310 - val_accuracy: 0.7934
```

[43]:
```
plot_display_metrics(model_3, history, "Training")
```




```
Training accuracy: 80.47999999999999%
Training loss: 0.435
```

[44]:
```python
plot_display_metrics(model_3, history, "Validation")
```




```
Validation accuracy: 79.34%
Validation loss: 0.431
```

1.1.9 Evaluating selected model on test set

I have selected Model 2 (units of first layer doubled), as both this model and Model 1 exhibited nearly the same validation accuracy, but Model 2 saw a slightly higher training accuracy — meanwhile, Model 3 seemed to have a far more volatile training and validation trajectory based on the plots.

[45]:
```
test_score = model_2.evaluate(X_test, y_test)
print(f"\nTesting accuracy: {round(test_score[1], 4) * 100}%")
print(f"Testing loss: {round(test_score[0], 3)}")
```

```
20/20 [==============================] - 1s 6ms/step - loss: 0.4414 - accuracy:
0.7844

Testing accuracy: 78.44%
Testing loss: 0.441
```