Model

July 29, 2024

[1]:

```
import numpy as np
import tensorflow as tf
```

[2]:

```
npz_file = np.load('Audiobooks_data_train.npz')
train_inputs = npz_file['inputs'].astype(float)
train_targets = npz_file['targets'].astype(int)

npz_file = np.load('Audiobooks_data_validation.npz')
validation_inputs = npz_file['inputs'].astype(float)
validation_targets = npz_file['targets'].astype(int)

npz_file = np.load('Audiobooks_data_test.npz')
test_inputs = npz_file['inputs'].astype(float)
test_targets = npz_file['targets'].astype(int)
```

0.0.1 Building and Training Model

[3]:

```
input_size = 10
output_size = 2
hidden_layer_size = 50
batch_size = 100
max_epochs = 100

early_stopping = tf.keras.callbacks.EarlyStopping(patience=2) #Prevent␣
 ↪Overfitting

model = tf.keras.Sequential([
    tf.keras.layers.Dense(hidden_layer_size, activation='relu'),
    tf.keras.layers.Dense(hidden_layer_size, activation='relu'),
    tf.keras.layers.Dense(output_size, activation='softmax')
])

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',␣
 ↪metrics=['accuracy'])

model.fit(train_inputs, train_targets,
          batch_size=batch_size,
          epochs=max_epochs,
          callbacks=[early_stopping],
          validation_data=(validation_inputs,validation_targets),
          verbose=2)
```

```
Epoch 1/100
36/36 - 1s - loss: 0.5790 - accuracy: 0.6806 - val_loss: 0.5090 - val_accuracy:
0.7651 - 1s/epoch - 29ms/step
Epoch 2/100
36/36 - 0s - loss: 0.4611 - accuracy: 0.7714 - val_loss: 0.4434 - val_accuracy:
0.7830 - 70ms/epoch - 2ms/step
Epoch 3/100
36/36 - 0s - loss: 0.4102 - accuracy: 0.7835 - val_loss: 0.4064 - val_accuracy:
0.7740 - 78ms/epoch - 2ms/step
Epoch 4/100
36/36 - 0s - loss: 0.3844 - accuracy: 0.7974 - val_loss: 0.3927 - val_accuracy:
0.7740 - 91ms/epoch - 3ms/step
Epoch 5/100
36/36 - 0s - loss: 0.3697 - accuracy: 0.7944 - val_loss: 0.3827 - val_accuracy:
0.7919 - 83ms/epoch - 2ms/step
Epoch 6/100
36/36 - 0s - loss: 0.3567 - accuracy: 0.8117 - val_loss: 0.3780 - val_accuracy:
0.7919 - 80ms/epoch - 2ms/step
Epoch 7/100
36/36 - 0s - loss: 0.3498 - accuracy: 0.8167 - val_loss: 0.3718 - val_accuracy:
0.7875 - 79ms/epoch - 2ms/step
Epoch 8/100
36/36 - 0s - loss: 0.3433 - accuracy: 0.8131 - val_loss: 0.3638 - val_accuracy:
0.7942 - 75ms/epoch - 2ms/step
Epoch 9/100
36/36 - 0s - loss: 0.3378 - accuracy: 0.8181 - val_loss: 0.3638 - val_accuracy:
0.8098 - 74ms/epoch - 2ms/step
Epoch 10/100
36/36 - 0s - loss: 0.3359 - accuracy: 0.8181 - val_loss: 0.3621 - val_accuracy:
0.8009 - 74ms/epoch - 2ms/step
Epoch 11/100
36/36 - 0s - loss: 0.3332 - accuracy: 0.8170 - val_loss: 0.3583 - val_accuracy:
0.8031 - 70ms/epoch - 2ms/step
Epoch 12/100
36/36 - 0s - loss: 0.3301 - accuracy: 0.8220 - val_loss: 0.3588 - val_accuracy:
0.8076 - 77ms/epoch - 2ms/step
Epoch 13/100
36/36 - 0s - loss: 0.3270 - accuracy: 0.8248 - val_loss: 0.3589 - val_accuracy:
0.8166 - 78ms/epoch - 2ms/step
```

```
[3]: <keras.callbacks.History at 0x1f0e3cd2560>
```

0.0.2 Testing

[4]:
```
test_loss, test_accuracy = model.evaluate(test_inputs, test_targets)
```

```
14/14 [==============================] - 0s 1ms/step - loss: 0.3275 - accuracy:
0.8281
```

[13]:
```
test_inputs[0], test_inputs[10]
```

[13]:
```
(array([ 0.21053387, -0.18888517,  0.34457208,  0.09587919,  2.10780037,
        -2.60893407,  0.51682305,  0.63990752, -0.20183481, -0.62711546]),
 array([ 0.21053387, -0.18888517, -0.08811106, -0.25963932, -0.47442823,
         0.00142655, -0.42647785, -0.41569922, -0.20183481, -0.80255852]))
```

[12]:
```
test_targets[0], test_targets[10]
```

[12]:
```
(0, 1)
```

[8]:
```
y_hat = model.predict(test_inputs)
```

```
14/14 [==============================] - 0s 924us/step
```

[14]:
```
np.argmax(y_hat[0]), np.argmax(y_hat[10])
```

[14]:
```
(0, 1)
```

[ ]:
```

```