

Exercise 2.2:Complex Machine Learning Models and Keras Part 1

I chose RNN model instead of CNN because I believe it is a better suit since it has the ability to process temporal information and this is a key concept for this analysis.

RNN

```
epochs = 8
batch_size = 16
n_hidden = 32

timesteps = len(X_train[0])
input_dim = len(X_train[0][0])
n_classes = len(y_train[0])

model = Sequential()
model.add(Conv1D(n_hidden, kernel_size=2, activation='relu', input_shape=(timesteps, input_dim)))
model.add(MaxPooling1D())
model.add(LSTM(n_hidden, input_shape=(timesteps, input_dim)))
model.add(Dropout(0.5))
model.add(Dense(n_classes, activation='tanh'))
```

```
model.fit(X_train,
          y_train,
          batch_size=batch_size,
          validation_data=(X_test, y_test),
          epochs=epochs)
```

```
Epoch 1/8
1076/1076 ————— 6s 4ms/step - accuracy: 0.0373 - loss: 201099984.0000 - val_accuracy: 0.0000e+00 - val_loss: 313068448.0000
Epoch 2/8
1076/1076 ————— 4s 4ms/step - accuracy: 0.3006 - loss: 297441344.0000 - val_accuracy: 0.2954 - val_loss: 314228064.0000
Epoch 3/8
1076/1076 ————— 4s 4ms/step - accuracy: 0.1789 - loss: 299953984.0000 - val_accuracy: 0.0000e+00 - val_loss: 320931840.0000
Epoch 4/8
1076/1076 ————— 4s 4ms/step - accuracy: 0.1534 - loss: 306399008.0000 - val_accuracy: 0.6642 - val_loss: 313001536.0000
Epoch 5/8
1076/1076 ————— 4s 4ms/step - accuracy: 0.4702 - loss: 313470208.0000 - val_accuracy: 0.0000e+00 - val_loss: 311902432.0000
Epoch 6/8
1076/1076 ————— 5s 5ms/step - accuracy: 0.1023 - loss: 310622016.0000 - val_accuracy: 0.0000e+00 - val_loss: 320931744.0000
Epoch 7/8
1076/1076 ————— 5s 4ms/step - accuracy: 0.0371 - loss: 316207712.0000 - val_accuracy: 0.0000e+00 - val_loss: 320931744.0000
Epoch 8/8
1076/1076 ————— 4s 4ms/step - accuracy: 0.0078 - loss: 319110688.0000 - val_accuracy: 0.0000e+00 - val_loss: 320931744.0000
<keras.src.callbacks.history.History at 0x195a053f0e0>
```

```
print(confusion_matrix(y_test, model.predict(X_test)))
```

```
180/180 ————— 0s 1ms/step
Pred  DUSSELDORF  MAASTRICHT  MADRID  SONNBLICK  STOCKHOLM  VALENTIA
True
BASEL      1834          124         20         2066         1620          74
```

The accuracy does not increase and only has significant values in 2 epochs.

The model only recognises 6 stations.

While the RNN model seemed like a better fit, it is not working successfully.

CNN

```
epochs = 12
batch_size = 32
n_hidden = 256
timesteps = X_train.shape[1]
input_dim = X_train.shape[2]
n_classes = y_train.shape[1]

model = Sequential()
model.add(Conv1D(n_hidden, kernel_size=2, activation='relu', input_shape=(timesteps, input_dim)))
model.add(Dense(16, activation='relu'))
model.add(MaxPooling1D())
model.add(Flatten())
model.add(Dense(n_classes, activation='softmax')) # Options: sigmoid, tanh, softmax, relu
```

```
model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs, verbose=2)
```

```
Epoch 1/12
538/538 - 2s - 5ms/step - accuracy: 0.1302 - loss: 5375.4297
Epoch 2/12
538/538 - 1s - 2ms/step - accuracy: 0.1179 - loss: 56786.4414
Epoch 3/12
538/538 - 1s - 2ms/step - accuracy: 0.1201 - loss: 184703.7031
Epoch 4/12
538/538 - 1s - 2ms/step - accuracy: 0.1188 - loss: 400643.8750
Epoch 5/12
538/538 - 1s - 3ms/step - accuracy: 0.1195 - loss: 742940.8750
Epoch 6/12
538/538 - 1s - 2ms/step - accuracy: 0.1248 - loss: 1152820.2500
Epoch 7/12
538/538 - 1s - 2ms/step - accuracy: 0.1196 - loss: 1701139.2500
Epoch 8/12
538/538 - 1s - 2ms/step - accuracy: 0.1258 - loss: 2373463.2500
Epoch 9/12
538/538 - 1s - 2ms/step - accuracy: 0.1250 - loss: 3108797.2500
Epoch 10/12
538/538 - 1s - 2ms/step - accuracy: 0.1216 - loss: 4083777.7500
Epoch 11/12
538/538 - 1s - 2ms/step - accuracy: 0.1231 - loss: 5179535.5000
Epoch 12/12
538/538 - 1s - 2ms/step - accuracy: 0.1275 - loss: 6404762.0000
<keras.src.callbacks.history.History at 0x1f955fd9070>
```

The model is not converging, the accuracy tends to lower and the loss increases.

Second try

```
epochs = 16
batch_size = 12
n_hidden = 120
timesteps = X_train.shape[1]
input_dim = X_train.shape[2]
n_classes = y_train.shape[1]

model = Sequential()
model.add(Conv1D(n_hidden, kernel_size=2, activation='relu', input_shape=(timesteps, input_dim)))
model.add(Dense(16, activation='relu'))
model.add(MaxPooling1D())
model.add(Flatten())
model.add(Dense(n_classes, activation='softmax')) # Options: sigmoid, tanh, softmax, relu
```

```
: model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs, verbose=2)
```

```
Epoch 1/16
1435/1435 - 3s - 2ms/step - accuracy: 0.1042 - loss: 41085.8672
Epoch 2/16
1435/1435 - 3s - 2ms/step - accuracy: 0.1156 - loss: 474297.5312
Epoch 3/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1227 - loss: 1542147.8750
Epoch 4/16
1435/1435 - 2s - 1ms/step - accuracy: 0.1211 - loss: 3343080.0000
Epoch 5/16
1435/1435 - 3s - 2ms/step - accuracy: 0.1239 - loss: 5980917.0000
Epoch 6/16
1435/1435 - 3s - 2ms/step - accuracy: 0.1260 - loss: 9549740.0000
Epoch 7/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1260 - loss: 14166143.0000
Epoch 8/16
1435/1435 - 2s - 1ms/step - accuracy: 0.1271 - loss: 19728270.0000
Epoch 9/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1251 - loss: 26930220.0000
Epoch 10/16
1435/1435 - 2s - 1ms/step - accuracy: 0.1196 - loss: 35530988.0000
Epoch 11/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1230 - loss: 45351028.0000
Epoch 12/16
1435/1435 - 3s - 2ms/step - accuracy: 0.1241 - loss: 56538432.0000
Epoch 13/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1287 - loss: 69959456.0000
Epoch 14/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1268 - loss: 85156408.0000
Epoch 15/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1219 - loss: 101752096.0000
Epoch 16/16
1435/1435 - 2s - 2ms/step - accuracy: 0.1264 - loss: 120339184.0000
: <keras.src.callbacks.history.History at 0x1f95565e5d0>
```

With these parameters the accuracy increases.

CONFUSION MATRIX

180/180		1s 3ms/step						
Pred		BASEL	BELGRADE	BUDAPEST	DEBILT	DUSSELDORF	HEATHROW	KASSEL \
True								
BASEL		2	68	864	78	248	61	200
BELGRADE		0	89	77	4	78	1	6
BUDAPEST		0	15	9	3	21	0	0
DEBILT		0	2	0	0	14	0	0
DUSSELDORF		0	0	1	0	5	0	0
HEATHROW		0	6	2	3	13	0	1
KASSEL		0	1	0	0	1	0	0
LJUBLJANA		0	4	0	1	8	0	0
MAASTRICHT		0	0	0	2	2	0	1
MADRID		0	28	50	10	35	4	4
MUNCHENB		0	1	0	1	1	0	0
OSLO		0	2	0	0	0	0	0
STOCKHOLM		0	2	0	0	1	0	0
VALENTIA		0	0	0	0	0	0	0
Pred		LJUBLJANA	MAASTRICHT	MADRID	MUNCHENB	OSLO	SONNBLICK \	
True								
BASEL		1183	1	303	413	115	6	
BELGRADE		642	0	20	142	9	0	
BUDAPEST		120	0	9	18	3	0	
DEBILT		49	0	2	11	0	0	
DUSSELDORF		16	0	1	4	2	0	
HEATHROW		36	0	8	9	1	0	
KASSEL		7	0	0	0	1	0	
LJUBLJANA		34	0	8	1	2	0	
MAASTRICHT		2	0	0	2	0	0	
MADRID		134	0	139	28	11	0	
MUNCHENB		3	0	0	1	1	0	
OSLO		2	0	0	1	0	0	
STOCKHOLM		0	0	0	0	0	0	
VALENTIA		0	0	0	0	0	0	
Pred		STOCKHOLM	VALENTIA					
True								
BASEL		103	37					
BELGRADE		24	0					
BUDAPEST		16	0					
DEBILT		4	0					
DUSSELDORF		0	0					
HEATHROW		3	0					
KASSEL		1	0					
LJUBLJANA		3	0					
MAASTRICHT		0	0					

The model does recognise all the 15 stations.
It has an accuracy of 12,28%