SHRAVANI

11/25/24

2.2: COMPLEX MACHINE LEARNING MODELS AND KERAS PART 1

Keras Convolutional Neural Network (CNN) model

Starting hyperparameters used:

epochs = 10

Batch_size = 16

n_hidden = 32

kernal_size = 2

Ending hyperparameters used:

5. Build and run CNN keras model

```
# Adjust model hyperparameters
epochs = 50
batch_size = 256
n_hidden = 256
kernel_size = 4 # Adjust the kernel size here
```

Final accuracy and loss:

```
Epoch 1/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - accuracy: 0.1325 - loss: 36.0365
Epoch 2/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1073 - loss: 2167.7600
Epoch 3/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.0997 - loss: 19381.2754
Epoch 4/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.0915 - loss: 32101.5723
Epoch 5/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.0822 - loss: 65302.5781
Epoch 6/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.0891 - loss: 125746.1641
Epoch 7/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1051 - loss: 154311.7188
Epoch 8/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1097 - loss: 168397.9219
Epoch 9/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1011 - loss: 194561.4375
Epoch 10/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - accuracy: 0.1073 - loss: 365984.5938
Epoch 11/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.0893 - loss: 431203.6250
Epoch 12/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1095 - loss: 391106.8438
Epoch 13/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1049 - loss: 367648.2500
Epoch 14/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1218 - loss: 754867.8750
Epoch 15/50
```

```
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1200 - loss: 958719.9375
Epoch 16/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1090 - loss: 911596.0625
Epoch 17/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1204 - loss: 936159.5625
Epoch 18/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1132 - loss: 757484.8750
Epoch 19/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1255 - loss: 927077.4375
Epoch 20/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1034 - loss: 833251.7500
Epoch 21/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1250 - loss: 812757.5625
Epoch 22/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1256 - loss: 1091938.7500
Epoch 23/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1010 - loss: 1191490.8750
Epoch 24/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1199 - loss: 1176658.5000
Epoch 25/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1119 - loss: 1581549.1250
Epoch 26/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.0937 - loss: 1811054.5000
Epoch 27/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - accuracy: 0.1192 - loss: 1604095.5000
Epoch 28/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.1084 - loss: 1486293.2500
Epoch 29/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - accuracy: 0.0919 - loss: 2018593.1250
Epoch 30/50

Epoch 31/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1132 - loss: 1510879.6250
Epoch 32/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1500 - loss: 2106991.2500
Epoch 33/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1217 - loss: 2030779.8750
Epoch 34/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1380 - loss: 1840940.7500
Epoch 35/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1129 - loss: 1931425.7500
Epoch 36/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - accuracy: 0.1255 - loss: 1922366.5000
Epoch 37/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1165 - loss: 2286527.0000
Epoch 38/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - accuracy: 0.1099 - loss: 2708203.7500
Epoch 39/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - accuracy: 0.1345 - loss: 2283637.0000
Epoch 40/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1036 - loss: 2541859.2500
Epoch 41/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - accuracy: 0.1300 - loss: 2356622.0000
Epoch 42/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1179 - loss: 2440108.7500
Epoch 43/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - accuracy: 0.1136 - loss: 2918593.5000
Epoch 44/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1380 - loss: 3524817.7500
Epoch 45/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1115 - loss: 2378272.7500
Epoch 46/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1161 - loss: 2938185.2500
Epoch 47/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1175 - loss: 3230787.7500
Epoch 48/50

72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1120 - loss: 2547187.2500
Epoch 49/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1203 - loss: 3922389.5000
Epoch 50/50
72/72 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - accuracy: 0.1275 - loss: 4377941.0000
<keras.src.callbacks.history.History at 0x2a7680f0590>
```

Partial screenshot of (ending parameters) confusion matrix (easier to see in script):

```
[454]: # Evaluate
       print(confusion_matrix(y_test, model.predict(X_test))) #relatively low accuracy and high loss
```

Pred	BUDAPEST_pleasant_weather	\
True		
BASEL_pleasant_weather	774	
BELGRADE_pleasant_weather	293	
BUDAPEST_pleasant_weather	69	
DEBILT_pleasant_weather	21	
DUSSELDORF_pleasant_weather	5	
HEATHROW_pleasant_weather	19	
KASSEL_pleasant_weather	3	
LJUBLJANA_pleasant_weather	6	
MAASTRICHT_pleasant_weather	2	
MADRID_pleasant_weather	89	
MUNCHENB_pleasant_weather	2	
OSLO_pleasant_weather	2	
STOCKHOLM_pleasant_weather	3	
VALENTIA_pleasant_weather	0	

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 12, 256)	9,472
dense (Dense)	(None, 12, 256)	65,792
max_pooling1d (MaxPooling1D)	(None, 6, 256)	0
flatten (Flatten)	(None, 1536)	0
dense_1 (Dense)	(None, 15)	23,055

Total params: 294,959 (1.13 MB)

Trainable params: 98,319 (384.06 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 196,640 (768.13 KB)

2.2: COMPLEX MACHINE LEARNING MODELS AND KERAS PART 1

Notes:

- The model predicted for all weather stations.
- There is relatively low accuracy and exceptionally high loss for all hyperparameters used in CNN model.
- Accuracy improves slightly then plateaus or decreases while the loss increases with each epoch.
- The model may be too simple to capture the complex data patterns.
- When using binary cross-entropy with sigmoid, similar results occur, however, the accuracy increases while loss converges lower with each epoch to around .08 loss and 0.3 accuracy.

Questions to consider:

- Does the model use an unscaled data approach? (yes)
- Possible preprocessing issues?
- Possible data architectural issues?

Recommendations:

- Implement scaled weather data.
- A better look into data preprocessing and architectural issues.
- Consider more complex layering, a different activation method, or keras model.