RNN

October 30, 2023

[5]:

```
###Inspecting the data of the Jena weather dataset
import os
data_dir = 'C:\\Users\\zhong\\Desktop\\FA23_DL\\hw3\\jena_climate'
fname = os.path.join(data_dir, 'jena_climate_2009_2016.csv')

with open(fname) as f:
    data = f.read()

lines = data.split("\n")
header = lines[0].split(",")
lines = lines[1:]
print(header)
print(len(lines))
```

```
['"Date Time"', '"p (mbar)"', '"T (degC)"', '"Tpot (K)"', '"Tdew (degC)"', '"rh
(%)"', '"VPmax (mbar)"', '"VPact (mbar)"', '"VPdef (mbar)"', '"sh (g/kg)"',
'"H2OC (mmol/mol)"', '"rho (g/m**3)"', '"wv (m/s)"', '"max. wv (m/s)"', '"wd
(deg)"']
420451
```

[6]:

```
###Parsing the data
import numpy as np
temperature = np.zeros((len(lines),))
raw_data = np.zeros((len(lines), len(header) - 1))
for i, line in enumerate(lines):
    values = [float(x) for x in line.split(",")[1:]]
    temperature[i] = values[1]
    raw_data[i, :] = values[:]
```

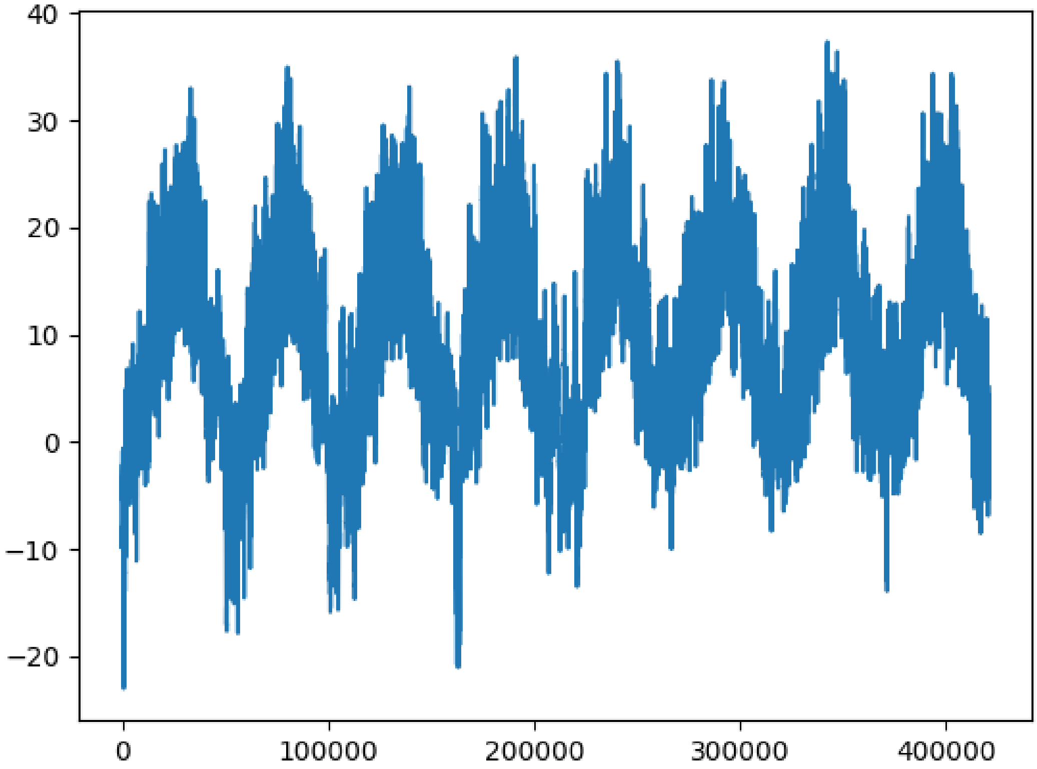
[7]:

```
###Plotting the temperature timeseries

from matplotlib import pyplot as plt
plt.plot(range(len(temperature)), temperature)
```

[7]:

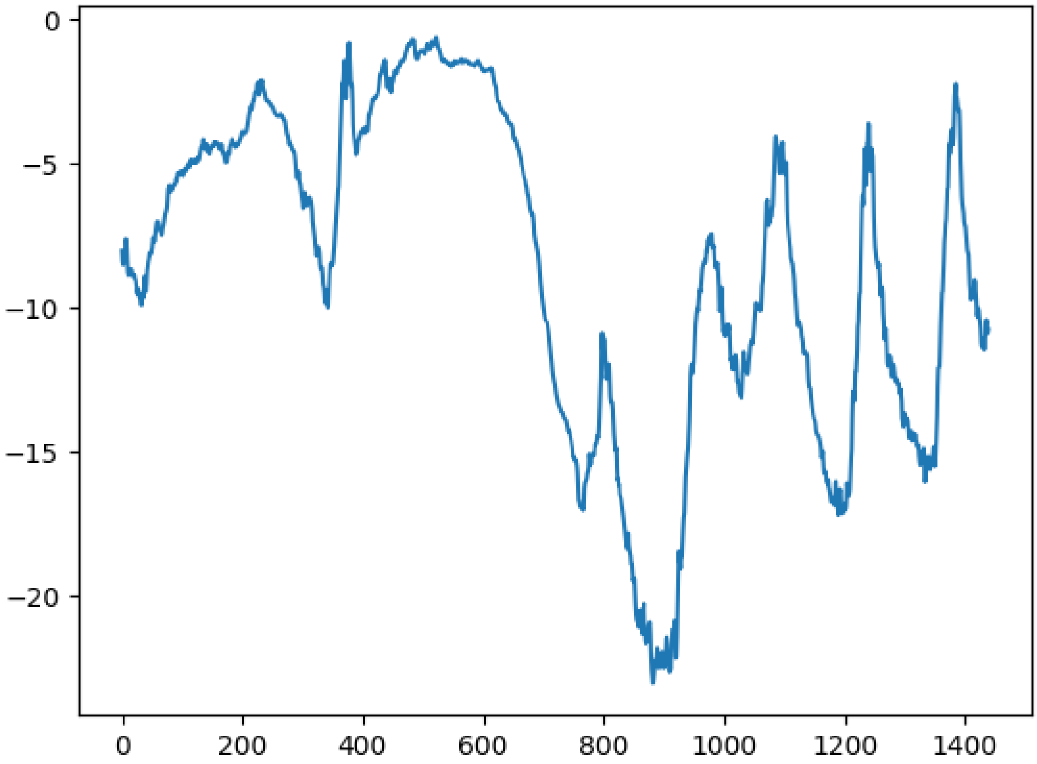
```
[<matplotlib.lines.Line2D at 0x2974f13aca0>]
```



```
[8]: """ Plotting the first 10 days of the temperature timeseries """

     plt.plot(range(1440), temperature[:1440])
```

```
[8]: [<matplotlib.lines.Line2D at 0x2974f8e9610>]
```



[9]:

```
""" Computing the number of samples we'll use for each data split """

num_train_samples = int(0.5 * len(raw_data))
num_val_samples = int(0.25 * len(raw_data))
num_test_samples = len(raw_data) - num_train_samples - num_val_samples
print("num_train_samples:", num_train_samples)
print("num_val_samples:", num_val_samples)
print("num_test_samples:", num_test_samples)
```

```
num_train_samples: 210225
num_val_samples: 105112
num_test_samples: 105114
```

[10]:

```
""" Preparing the data """
```

[10]:

```
' Preparing the data '
```

[11]:

```
""" Normalizing the data """

mean = raw_data[:num_train_samples].mean(axis=0)
raw_data -= mean
std = raw_data[:num_train_samples].std(axis=0)
```

```
raw_data /= std
import numpy as np
from tensorflow import keras
int_sequence = np.arange(10)
dummy_dataset = keras.utils.timeseries_dataset_from_array(
    data=int_sequence[:-3],
    targets=int_sequence[3:],
    sequence_length=3,
    batch_size=2,
)

for inputs, targets in dummy_dataset:
    for i in range(inputs.shape[0]):
        print([int(x) for x in inputs[i]], int(targets[i]))
```

```
[0, 1, 2] 3
[1, 2, 3] 4
[2, 3, 4] 5
[3, 4, 5] 6
[4, 5, 6] 7
```

[12]:

```
""" Instantiating datasets for training, validation, and testing """

sampling_rate = 6
sequence_length = 120
delay = sampling_rate * (sequence_length + 24 - 1)
batch_size = 256

train_dataset = keras.utils.timeseries_dataset_from_array(
    raw_data[:-delay],
    targets=temperature[delay:],
    sampling_rate=sampling_rate,
    sequence_length=sequence_length,
    shuffle=True,
    batch_size=batch_size,
    start_index=0,
    end_index=num_train_samples)

val_dataset = keras.utils.timeseries_dataset_from_array(
    raw_data[:-delay],
    targets=temperature[delay:],
    sampling_rate=sampling_rate,
    sequence_length=sequence_length,
    shuffle=True,
    batch_size=batch_size,
    start_index=num_train_samples,
    end_index=num_train_samples + num_val_samples)
```

```
test_dataset = keras.utils.timeseries_dataset_from_array(
    raw_data[:-delay],
    targets=temperature[delay:],
    sampling_rate=sampling_rate,
    sequence_length=sequence_length,
    shuffle=True,
    batch_size=batch_size,
    start_index=num_train_samples + num_val_samples)
```

[13]:

```
""" Inspecting the output of one of our datasets """

for samples, targets in train_dataset:
    print("samples shape:", samples.shape)
    print("targets shape:", targets.shape)
    break
```

```
samples shape: (256, 120, 14)
targets shape: (256,)
```

[14]:

```
""" A common-sense, non-machine-learning baseline """
```

[14]:

```
' A common-sense, non-machine-learning baseline '
```

[15]:

```
""" Computing the common-sense baseline MAE """
#########################################################################

def evaluate_naive_method(dataset):
    total_abs_err = 0.
    samples_seen = 0
    for samples, targets in dataset:
        preds = samples[:, -1, 1] * std[1] + mean[1]
        total_abs_err += np.sum(np.abs(preds - targets))
        samples_seen += samples.shape[0]
    return total_abs_err / samples_seen

print(f"Validation MAE: {evaluate_naive_method(val_dataset):.2f}")
print(f"Test MAE: {evaluate_naive_method(test_dataset):.2f}")
```

```
Validation MAE: 2.44
Test MAE: 2.62
```

[16]:

```
""" Let's try a basic machine-learning model
Training and evaluating a densely connected model """

from tensorflow import keras
from tensorflow.keras import layers
```

```python
inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x = layers.Flatten()(inputs)
x = layers.Dense(16, activation="relu")(x)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_dense.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=10,
                    validation_data=val_dataset,
                    callbacks=callbacks)

model = keras.models.load_model("jena_dense.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
```

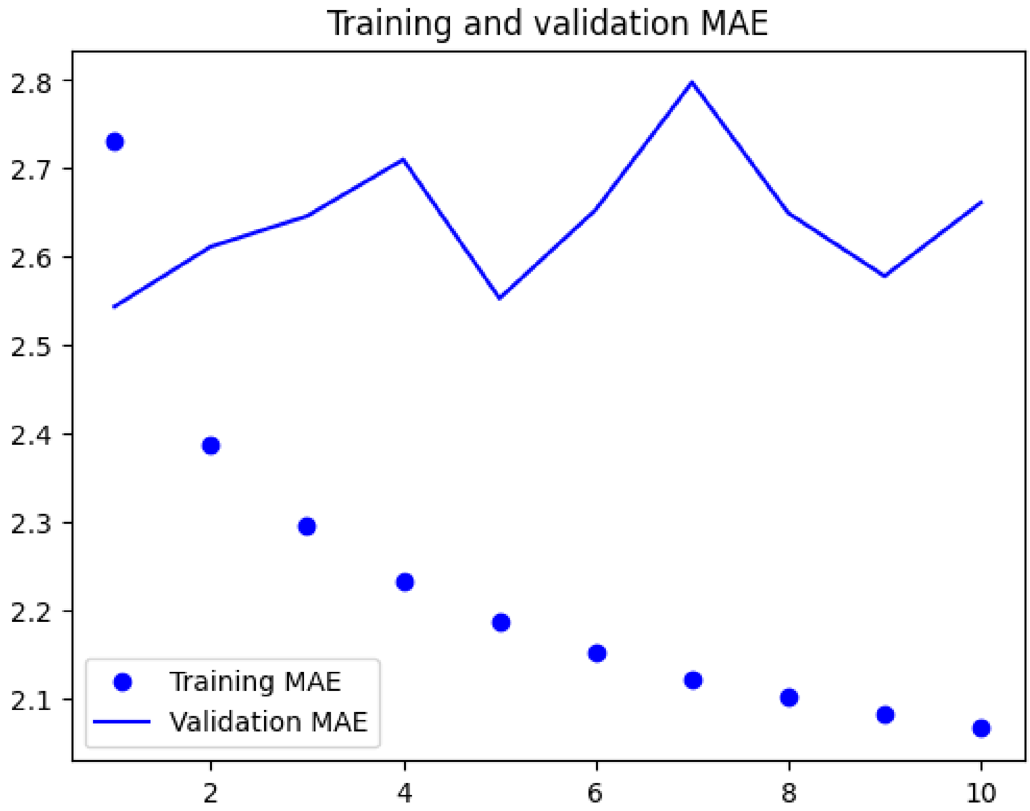
```
Epoch 1/10
819/819 [==============================] - 58s 38ms/step - loss: 12.4537 - mae:
2.7315 - val_loss: 10.4041 - val_mae: 2.5438
Epoch 2/10
819/819 [==============================] - 7s 8ms/step - loss: 9.2183 - mae:
2.3869 - val_loss: 10.9592 - val_mae: 2.6115
Epoch 3/10
819/819 [==============================] - 7s 8ms/step - loss: 8.4990 - mae:
2.2966 - val_loss: 11.3068 - val_mae: 2.6461
Epoch 4/10
819/819 [==============================] - 7s 8ms/step - loss: 8.0135 - mae:
2.2326 - val_loss: 11.7190 - val_mae: 2.7101
Epoch 5/10
819/819 [==============================] - 7s 9ms/step - loss: 7.6987 - mae:
2.1883 - val_loss: 10.4503 - val_mae: 2.5531
Epoch 6/10
819/819 [==============================] - 7s 9ms/step - loss: 7.4472 - mae:
2.1518 - val_loss: 11.2203 - val_mae: 2.6534
Epoch 7/10
819/819 [==============================] - 7s 9ms/step - loss: 7.2421 - mae:
2.1232 - val_loss: 12.6793 - val_mae: 2.7976
Epoch 8/10
819/819 [==============================] - 7s 8ms/step - loss: 7.0966 - mae:
2.1021 - val_loss: 11.2464 - val_mae: 2.6496
Epoch 9/10
819/819 [==============================] - 8s 9ms/step - loss: 6.9647 - mae:
2.0829 - val_loss: 10.6269 - val_mae: 2.5783
Epoch 10/10
819/819 [==============================] - 7s 8ms/step - loss: 6.8566 - mae:
```

```
2.0668 - val_loss: 11.3331 - val_mae: 2.6614
405/405 [==============================] - 2s 6ms/step - loss: 11.7956 - mae:
2.6973
Test MAE: 2.70
```

[17]:

```
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE")
plt.legend()
plt.show()
```



[18]:

```
""" Let's try a 1D convolutional model """

inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
```

```
x = layers.Conv1D(8, 24, activation="relu")(inputs)
x = layers.MaxPooling1D(2)(x)
x = layers.Conv1D(8, 12, activation="relu")(x)
x = layers.MaxPooling1D(2)(x)
x = layers.Conv1D(8, 6, activation="relu")(x)
x = layers.GlobalAveragePooling1D()(x)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_conv.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=10,
                    validation_data=val_dataset,
                    callbacks=callbacks)

model = keras.models.load_model("jena_conv.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
```

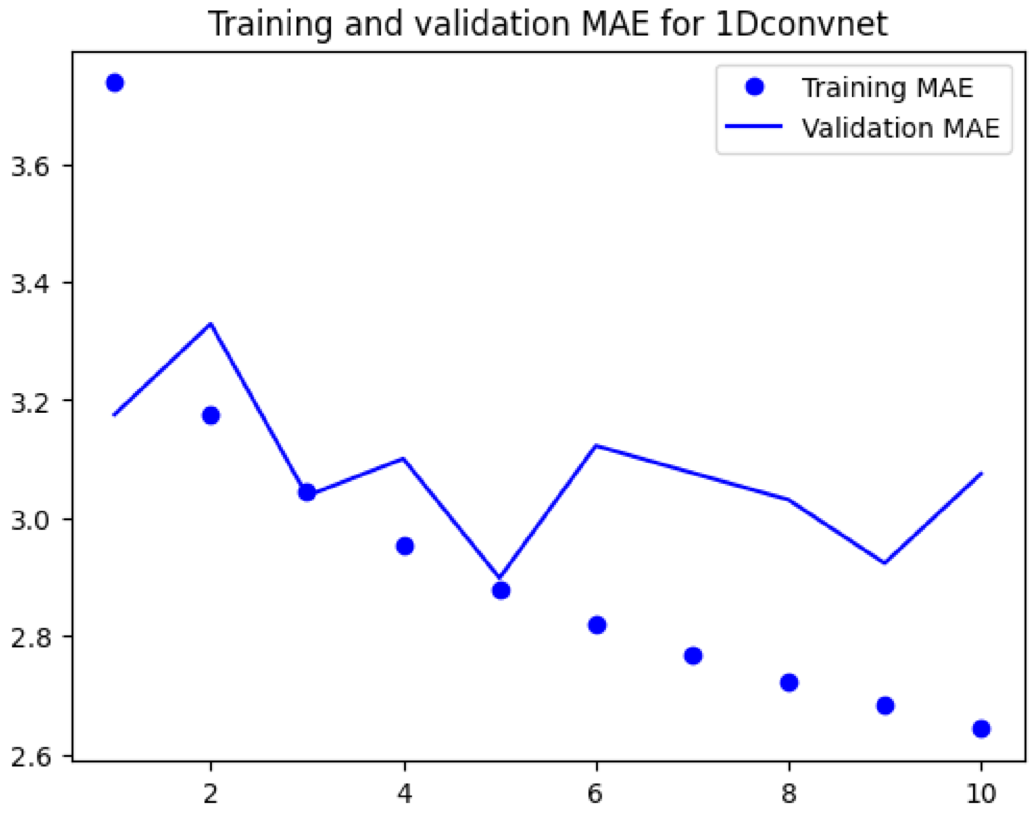
```
Epoch 1/10
819/819 [==============================] - 22s 10ms/step - loss: 22.7434 - mae:
3.7376 - val_loss: 16.1472 - val_mae: 3.1753
Epoch 2/10
819/819 [==============================] - 8s 9ms/step - loss: 15.9333 - mae:
3.1744 - val_loss: 17.8546 - val_mae: 3.3290
Epoch 3/10
819/819 [==============================] - 7s 9ms/step - loss: 14.6887 - mae:
3.0451 - val_loss: 14.8057 - val_mae: 3.0377
Epoch 4/10
819/819 [==============================] - 7s 9ms/step - loss: 13.8552 - mae:
2.9535 - val_loss: 15.2517 - val_mae: 3.1006
Epoch 5/10
819/819 [==============================] - 7s 9ms/step - loss: 13.2215 - mae:
2.8781 - val_loss: 13.4629 - val_mae: 2.8983
Epoch 6/10
819/819 [==============================] - 7s 9ms/step - loss: 12.7066 - mae:
2.8206 - val_loss: 15.6948 - val_mae: 3.1225
Epoch 7/10
819/819 [==============================] - 7s 9ms/step - loss: 12.2412 - mae:
2.7669 - val_loss: 14.9241 - val_mae: 3.0766
Epoch 8/10
819/819 [==============================] - 7s 9ms/step - loss: 11.8817 - mae:
2.7233 - val_loss: 14.5689 - val_mae: 3.0312
Epoch 9/10
819/819 [==============================] - 8s 9ms/step - loss: 11.5368 - mae:
```

```
2.6832 - val_loss: 13.8512 - val_mae: 2.9235
Epoch 10/10
819/819 [==============================] - 8s 10ms/step - loss: 11.1915 - mae:
2.6429 - val_loss: 14.9504 - val_mae: 3.0749
405/405 [==============================] - 3s 6ms/step - loss: 15.7211 - mae:
3.1576
Test MAE: 3.16
```

[19]:

```python
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE for 1Dconvnet")
plt.legend()
plt.show()
```



[20]:

```python
""" A simple LSTM-based model """

inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x = layers.LSTM(16)(inputs)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=10,
                    validation_data=val_dataset,
                    callbacks=callbacks)

model = keras.models.load_model("jena_lstm.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
```

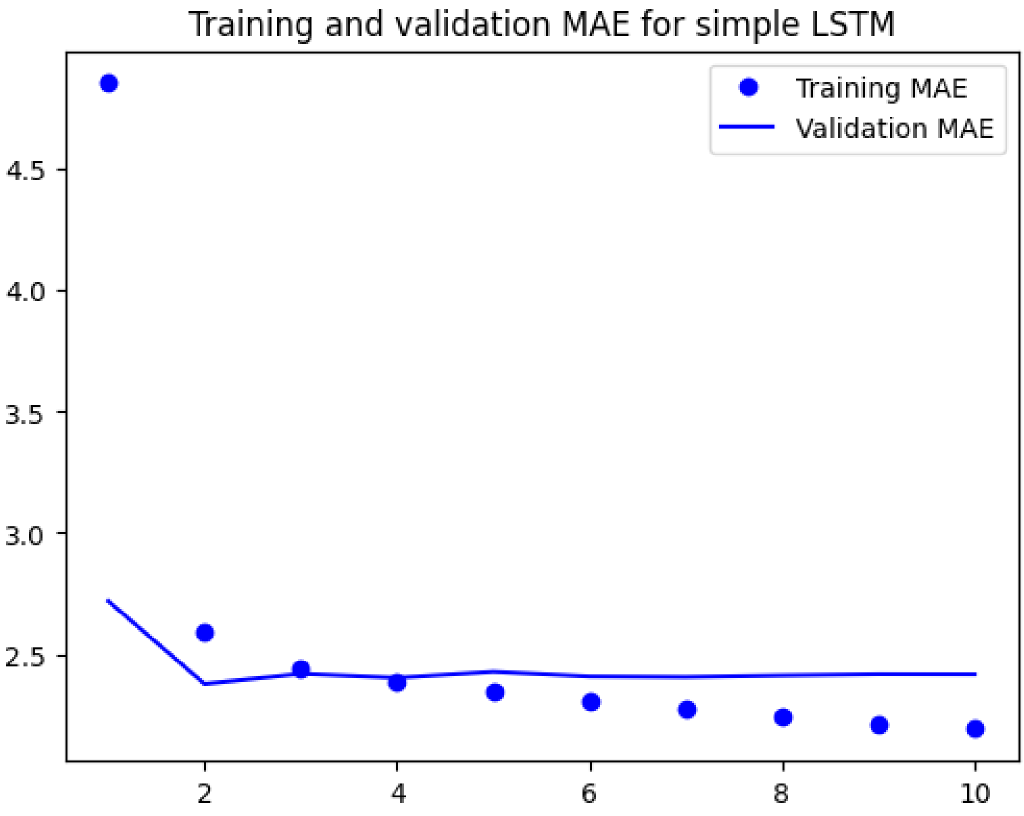
```
Epoch 1/10
819/819 [==============================] - 13s 14ms/step - loss: 43.9837 - mae:
4.8550 - val_loss: 12.8036 - val_mae: 2.7188
Epoch 2/10
819/819 [==============================] - 11s 13ms/step - loss: 11.2612 - mae:
2.5938 - val_loss: 9.3156 - val_mae: 2.3776
Epoch 3/10
819/819 [==============================] - 11s 13ms/step - loss: 9.7796 - mae:
2.4398 - val_loss: 9.6852 - val_mae: 2.4194
Epoch 4/10
819/819 [==============================] - 11s 13ms/step - loss: 9.3493 - mae:
2.3866 - val_loss: 9.6090 - val_mae: 2.4028
Epoch 5/10
819/819 [==============================] - 11s 13ms/step - loss: 9.0621 - mae:
2.3459 - val_loss: 9.8540 - val_mae: 2.4275
Epoch 6/10
819/819 [==============================] - 11s 13ms/step - loss: 8.7639 - mae:
2.3053 - val_loss: 9.6259 - val_mae: 2.4083
Epoch 7/10
819/819 [==============================] - 11s 13ms/step - loss: 8.5195 - mae:
2.2732 - val_loss: 9.6449 - val_mae: 2.4060
Epoch 8/10
819/819 [==============================] - 11s 13ms/step - loss: 8.2977 - mae:
2.2438 - val_loss: 9.7725 - val_mae: 2.4124
Epoch 9/10
819/819 [==============================] - 11s 13ms/step - loss: 8.0745 - mae:
2.2124 - val_loss: 9.7378 - val_mae: 2.4175
Epoch 10/10
```

```
819/819 [==============================] - 11s 13ms/step - loss: 7.9061 - mae:
2.1915 - val_loss: 9.7400 - val_mae: 2.4168
405/405 [==============================] - 3s 7ms/step - loss: 10.4429 - mae:
2.5237
Test MAE: 2.52
```

[21]:

```
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE for simple LSTM")
plt.legend()
plt.show()
```



[22]:

```
""" A simple GRU-based model """
```

```python
inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x = layers.GRU(16)(inputs)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=10,
                    validation_data=val_dataset,
                    callbacks=callbacks)

model = keras.models.load_model("jena_lstm.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
```

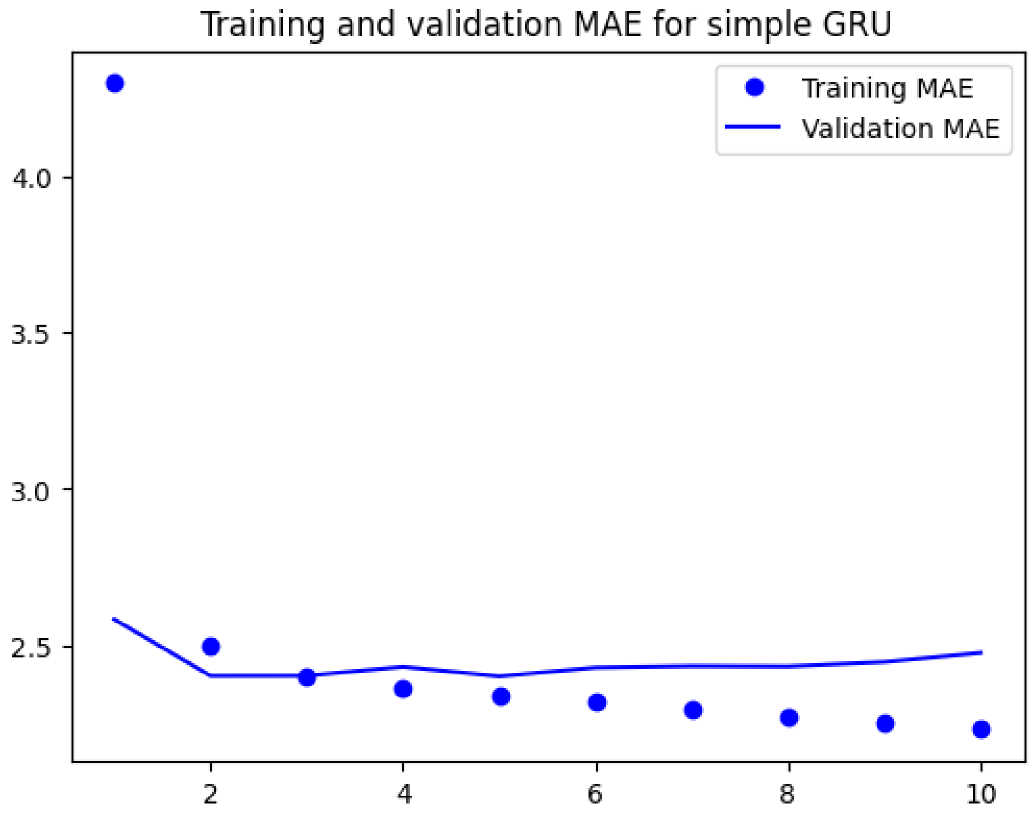
```
Epoch 1/10
819/819 [==============================] - 12s 14ms/step - loss: 35.0911 - mae:
4.2999 - val_loss: 11.6922 - val_mae: 2.5815
Epoch 2/10
819/819 [==============================] - 11s 13ms/step - loss: 10.2532 - mae:
2.4973 - val_loss: 9.8439 - val_mae: 2.4008
Epoch 3/10
819/819 [==============================] - 10s 13ms/step - loss: 9.4385 - mae:
2.3966 - val_loss: 9.9940 - val_mae: 2.4010
Epoch 4/10
819/819 [==============================] - 10s 13ms/step - loss: 9.1661 - mae:
2.3624 - val_loss: 10.3045 - val_mae: 2.4295
Epoch 5/10
819/819 [==============================] - 10s 13ms/step - loss: 8.9626 - mae:
2.3392 - val_loss: 9.7981 - val_mae: 2.3990
Epoch 6/10
819/819 [==============================] - 10s 13ms/step - loss: 8.7712 - mae:
2.3161 - val_loss: 10.0851 - val_mae: 2.4270
Epoch 7/10
819/819 [==============================] - 11s 13ms/step - loss: 8.5789 - mae:
2.2930 - val_loss: 9.9521 - val_mae: 2.4319
Epoch 8/10
819/819 [==============================] - 11s 13ms/step - loss: 8.3962 - mae:
2.2706 - val_loss: 10.0373 - val_mae: 2.4308
Epoch 9/10
819/819 [==============================] - 11s 14ms/step - loss: 8.2348 - mae:
2.2496 - val_loss: 10.1578 - val_mae: 2.4452
Epoch 10/10
819/819 [==============================] - 11s 14ms/step - loss: 8.0807 - mae:
2.2290 - val_loss: 10.5648 - val_mae: 2.4742
```

```
405/405 [==============================] - 4s 8ms/step - loss: 9.7997 - mae:
2.4705
Test MAE: 2.47
```

[23]:

```
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE for simple GRU")
plt.legend()
plt.show()
```



[26]:

```
""" Training and evaluating a dropout-regularized LSTM """

inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x = layers.LSTM(32, recurrent_dropout=0.25)(inputs)
```

```python
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm_dropout.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=40,
                    validation_data=val_dataset,
                    callbacks=callbacks)
```

```
WARNING:tensorflow:Layer lstm_3 will not use cuDNN kernels since it doesn't meet
the criteria. It will use a generic GPU kernel as fallback when running on GPU.
Epoch 1/40
819/819 [==============================] - 425s 517ms/step - loss: 28.3720 -
mae: 3.9362 - val_loss: 9.5627 - val_mae: 2.4039
Epoch 2/40
819/819 [==============================] - 399s 487ms/step - loss: 14.9874 -
mae: 3.0028 - val_loss: 9.3852 - val_mae: 2.3753
Epoch 3/40
819/819 [==============================] - 430s 525ms/step - loss: 14.1287 -
mae: 2.9104 - val_loss: 9.4640 - val_mae: 2.3710
Epoch 4/40
819/819 [==============================] - 428s 522ms/step - loss: 13.4689 -
mae: 2.8454 - val_loss: 10.1273 - val_mae: 2.4538
Epoch 5/40
819/819 [==============================] - 424s 518ms/step - loss: 13.0634 -
mae: 2.7983 - val_loss: 9.8079 - val_mae: 2.4173
Epoch 6/40
819/819 [==============================] - 427s 521ms/step - loss: 12.6926 -
mae: 2.7560 - val_loss: 9.6589 - val_mae: 2.4086
Epoch 7/40
819/819 [==============================] - 429s 524ms/step - loss: 12.4013 -
mae: 2.7236 - val_loss: 9.5989 - val_mae: 2.3910
Epoch 8/40
819/819 [==============================] - 402s 491ms/step - loss: 12.0327 -
mae: 2.6820 - val_loss: 9.7502 - val_mae: 2.4249
Epoch 9/40
819/819 [==============================] - 430s 525ms/step - loss: 11.7390 -
mae: 2.6521 - val_loss: 9.4987 - val_mae: 2.4006
Epoch 10/40
819/819 [==============================] - 431s 527ms/step - loss: 11.5549 -
mae: 2.6293 - val_loss: 9.6822 - val_mae: 2.4222
Epoch 11/40
819/819 [==============================] - 296s 361ms/step - loss: 11.3485 -
```

```
mae: 2.6054 - val_loss: 9.6835 - val_mae: 2.4215
Epoch 12/40
819/819 [==============================] - 241s 294ms/step - loss: 11.1299 -
mae: 2.5809 - val_loss: 9.8307 - val_mae: 2.4374
Epoch 13/40
819/819 [==============================] - 241s 294ms/step - loss: 10.9280 -
mae: 2.5575 - val_loss: 9.8694 - val_mae: 2.4480
Epoch 14/40
819/819 [==============================] - 241s 295ms/step - loss: 10.7960 -
mae: 2.5422 - val_loss: 10.0878 - val_mae: 2.4672
Epoch 15/40
819/819 [==============================] - 241s 294ms/step - loss: 10.6308 -
mae: 2.5209 - val_loss: 9.9794 - val_mae: 2.4592
Epoch 16/40
819/819 [==============================] - 242s 295ms/step - loss: 10.5205 -
mae: 2.5104 - val_loss: 10.2776 - val_mae: 2.4853
Epoch 17/40
819/819 [==============================] - 241s 295ms/step - loss: 10.3764 -
mae: 2.4927 - val_loss: 10.4206 - val_mae: 2.5020
Epoch 18/40
819/819 [==============================] - 241s 295ms/step - loss: 10.2853 -
mae: 2.4847 - val_loss: 10.4216 - val_mae: 2.5104
Epoch 19/40
819/819 [==============================] - 241s 295ms/step - loss: 10.1560 -
mae: 2.4714 - val_loss: 10.4683 - val_mae: 2.5116
Epoch 20/40
819/819 [==============================] - 240s 293ms/step - loss: 10.0791 -
mae: 2.4567 - val_loss: 11.2759 - val_mae: 2.5906
Epoch 21/40
819/819 [==============================] - 240s 293ms/step - loss: 10.0179 -
mae: 2.4502 - val_loss: 10.5400 - val_mae: 2.5217
Epoch 22/40
819/819 [==============================] - 242s 295ms/step - loss: 9.9581 - mae:
2.4447 - val_loss: 10.8098 - val_mae: 2.5577
Epoch 23/40
819/819 [==============================] - 366s 447ms/step - loss: 9.8392 - mae:
2.4315 - val_loss: 10.6564 - val_mae: 2.5501
Epoch 24/40
819/819 [==============================] - 414s 506ms/step - loss: 9.7599 - mae:
2.4207 - val_loss: 10.6537 - val_mae: 2.5544
Epoch 25/40
819/819 [==============================] - 427s 521ms/step - loss: 9.6698 - mae:
2.4080 - val_loss: 10.6119 - val_mae: 2.5377
Epoch 26/40
819/819 [==============================] - 426s 521ms/step - loss: 9.6524 - mae:
2.4074 - val_loss: 10.8765 - val_mae: 2.5730
Epoch 27/40
819/819 [==============================] - 427s 522ms/step - loss: 9.5844 - mae:
```

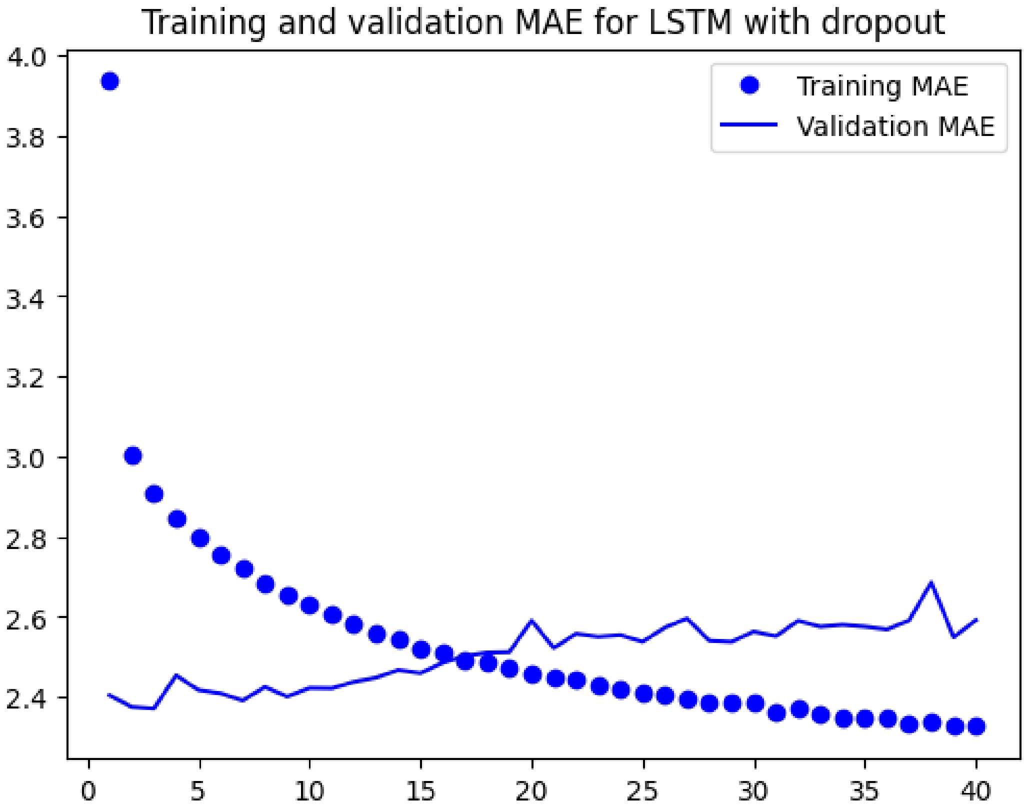
```
2.3949 - val_loss: 11.0066 - val_mae: 2.5956
Epoch 28/40
819/819 [==============================] - 434s 529ms/step - loss: 9.4918 - mae:
2.3849 - val_loss: 10.5632 - val_mae: 2.5403
Epoch 29/40
819/819 [==============================] - 427s 522ms/step - loss: 9.4833 - mae:
2.3847 - val_loss: 10.5279 - val_mae: 2.5376
Epoch 30/40
819/819 [==============================] - 430s 524ms/step - loss: 9.5301 - mae:
2.3868 - val_loss: 10.8731 - val_mae: 2.5630
Epoch 31/40
819/819 [==============================] - 427s 522ms/step - loss: 9.3121 - mae:
2.3640 - val_loss: 10.7090 - val_mae: 2.5514
Epoch 32/40
819/819 [==============================] - 428s 522ms/step - loss: 9.3716 - mae:
2.3717 - val_loss: 11.0940 - val_mae: 2.5899
Epoch 33/40
819/819 [==============================] - 426s 520ms/step - loss: 9.2738 - mae:
2.3577 - val_loss: 10.8852 - val_mae: 2.5752
Epoch 34/40
819/819 [==============================] - 426s 520ms/step - loss: 9.2054 - mae:
2.3487 - val_loss: 10.9430 - val_mae: 2.5799
Epoch 35/40
819/819 [==============================] - 429s 524ms/step - loss: 9.2215 - mae:
2.3501 - val_loss: 10.9811 - val_mae: 2.5754
Epoch 36/40
819/819 [==============================] - 429s 524ms/step - loss: 9.1520 - mae:
2.3456 - val_loss: 10.8816 - val_mae: 2.5682
Epoch 37/40
819/819 [==============================] - 421s 513ms/step - loss: 9.0651 - mae:
2.3343 - val_loss: 10.9759 - val_mae: 2.5909
Epoch 38/40
819/819 [==============================] - 424s 518ms/step - loss: 9.1079 - mae:
2.3362 - val_loss: 12.1554 - val_mae: 2.6859
Epoch 39/40
819/819 [==============================] - 432s 527ms/step - loss: 9.0244 - mae:
2.3297 - val_loss: 10.8103 - val_mae: 2.5488
Epoch 40/40
819/819 [==============================] - 431s 526ms/step - loss: 9.0110 - mae:
2.3263 - val_loss: 11.0704 - val_mae: 2.5913
```

[27]:

```python
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
```

```
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE for LSTM with dropout")
plt.legend()
plt.show()
```



[29]:

```
""" Training and evaluating a dropout-regularized GRU """

inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x = layers.GRU(32, recurrent_dropout=0.25)(inputs)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm_dropout.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
```

```
history = model.fit(train_dataset,
                    epochs=40,
                    validation_data=val_dataset,
                    callbacks=callbacks)
```

```
WARNING:tensorflow:Layer gru_1 will not use cuDNN kernels since it doesn't meet
the criteria. It will use a generic GPU kernel as fallback when running on GPU.
Epoch 1/40
819/819 [==============================] - 231s 279ms/step - loss: 27.0269 -
mae: 3.8608 - val_loss: 9.3896 - val_mae: 2.3660
Epoch 2/40
819/819 [==============================] - 228s 279ms/step - loss: 14.4872 -
mae: 2.9573 - val_loss: 9.4374 - val_mae: 2.3797
Epoch 3/40
819/819 [==============================] - 223s 273ms/step - loss: 13.7832 -
mae: 2.8821 - val_loss: 9.6698 - val_mae: 2.4150
Epoch 4/40
819/819 [==============================] - 222s 271ms/step - loss: 13.2080 -
mae: 2.8198 - val_loss: 8.8987 - val_mae: 2.3175
Epoch 5/40
819/819 [==============================] - 222s 271ms/step - loss: 12.7813 -
mae: 2.7745 - val_loss: 8.8456 - val_mae: 2.3114
Epoch 6/40
819/819 [==============================] - 221s 270ms/step - loss: 12.5174 -
mae: 2.7441 - val_loss: 8.8085 - val_mae: 2.3052
Epoch 7/40
819/819 [==============================] - 221s 270ms/step - loss: 12.2477 -
mae: 2.7128 - val_loss: 8.8715 - val_mae: 2.3092
Epoch 8/40
819/819 [==============================] - 346s 423ms/step - loss: 12.0430 -
mae: 2.6902 - val_loss: 8.8190 - val_mae: 2.3131
Epoch 9/40
819/819 [==============================] - 390s 476ms/step - loss: 11.7900 -
mae: 2.6658 - val_loss: 8.9659 - val_mae: 2.3301
Epoch 10/40
819/819 [==============================] - 389s 475ms/step - loss: 11.6207 -
mae: 2.6491 - val_loss: 8.8623 - val_mae: 2.3129
Epoch 11/40
819/819 [==============================] - 386s 471ms/step - loss: 11.4274 -
mae: 2.6271 - val_loss: 9.3258 - val_mae: 2.3732
Epoch 12/40
819/819 [==============================] - 394s 481ms/step - loss: 11.2872 -
mae: 2.6122 - val_loss: 8.8724 - val_mae: 2.3251
Epoch 13/40
819/819 [==============================] - 391s 477ms/step - loss: 11.1236 -
mae: 2.5949 - val_loss: 9.4404 - val_mae: 2.3833
Epoch 14/40
819/819 [==============================] - 391s 477ms/step - loss: 11.0037 -
```

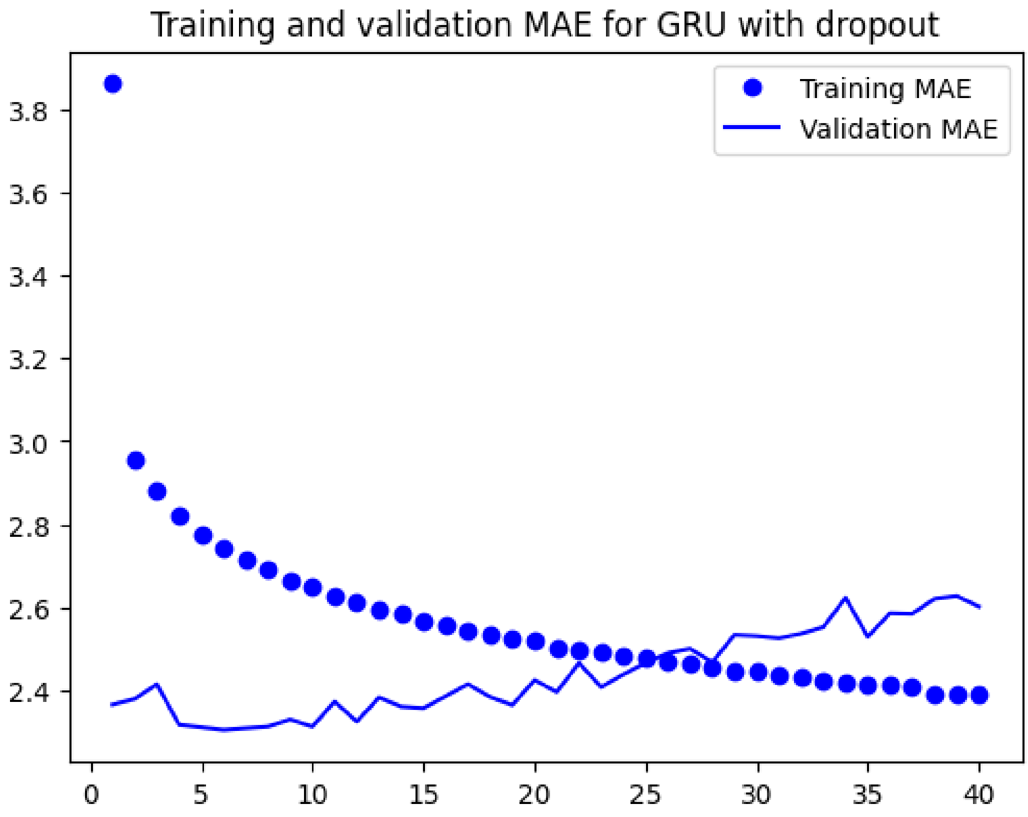
```
mae: 2.5856 - val_loss: 9.1387 - val_mae: 2.3610
Epoch 15/40
819/819 [==============================] - 397s 484ms/step - loss: 10.8539 -
mae: 2.5644 - val_loss: 9.2011 - val_mae: 2.3572
Epoch 16/40
819/819 [==============================] - 373s 456ms/step - loss: 10.7773 -
mae: 2.5558 - val_loss: 9.4228 - val_mae: 2.3863
Epoch 17/40
819/819 [==============================] - 361s 440ms/step - loss: 10.6879 -
mae: 2.5446 - val_loss: 9.6109 - val_mae: 2.4151
Epoch 18/40
819/819 [==============================] - 269s 328ms/step - loss: 10.5889 -
mae: 2.5356 - val_loss: 9.4127 - val_mae: 2.3844
Epoch 19/40
819/819 [==============================] - 216s 263ms/step - loss: 10.4890 -
mae: 2.5245 - val_loss: 9.2529 - val_mae: 2.3649
Epoch 20/40
819/819 [==============================] - 217s 265ms/step - loss: 10.4525 -
mae: 2.5193 - val_loss: 9.6428 - val_mae: 2.4251
Epoch 21/40
819/819 [==============================] - 220s 268ms/step - loss: 10.3100 -
mae: 2.5021 - val_loss: 9.4531 - val_mae: 2.3962
Epoch 22/40
819/819 [==============================] - 219s 267ms/step - loss: 10.2419 -
mae: 2.4979 - val_loss: 9.9848 - val_mae: 2.4662
Epoch 23/40
819/819 [==============================] - 271s 331ms/step - loss: 10.1899 -
mae: 2.4906 - val_loss: 9.5902 - val_mae: 2.4082
Epoch 24/40
819/819 [==============================] - 397s 485ms/step - loss: 10.1205 -
mae: 2.4837 - val_loss: 9.7960 - val_mae: 2.4387
Epoch 25/40
819/819 [==============================] - 393s 479ms/step - loss: 10.0579 -
mae: 2.4769 - val_loss: 9.9807 - val_mae: 2.4662
Epoch 26/40
819/819 [==============================] - 389s 475ms/step - loss: 10.0088 -
mae: 2.4688 - val_loss: 10.1646 - val_mae: 2.4901
Epoch 27/40
819/819 [==============================] - 388s 474ms/step - loss: 9.9817 - mae:
2.4642 - val_loss: 10.2550 - val_mae: 2.5007
Epoch 28/40
819/819 [==============================] - 389s 475ms/step - loss: 9.9384 - mae:
2.4577 - val_loss: 10.1221 - val_mae: 2.4681
Epoch 29/40
819/819 [==============================] - 397s 484ms/step - loss: 9.8148 - mae:
2.4456 - val_loss: 10.4482 - val_mae: 2.5337
Epoch 30/40
819/819 [==============================] - 387s 472ms/step - loss: 9.8330 - mae:
```

```
2.4471 - val_loss: 10.4118 - val_mae: 2.5311
Epoch 31/40
819/819 [==============================] - 384s 469ms/step - loss: 9.7384 - mae:
2.4359 - val_loss: 10.4476 - val_mae: 2.5258
Epoch 32/40
819/819 [==============================] - 387s 472ms/step - loss: 9.7022 - mae:
2.4317 - val_loss: 10.4980 - val_mae: 2.5366
Epoch 33/40
819/819 [==============================] - 379s 463ms/step - loss: 9.6412 - mae:
2.4235 - val_loss: 10.7193 - val_mae: 2.5525
Epoch 34/40
819/819 [==============================] - 394s 481ms/step - loss: 9.5988 - mae:
2.4204 - val_loss: 11.1245 - val_mae: 2.6232
Epoch 35/40
819/819 [==============================] - 394s 481ms/step - loss: 9.5849 - mae:
2.4137 - val_loss: 10.4550 - val_mae: 2.5292
Epoch 36/40
819/819 [==============================] - 386s 471ms/step - loss: 9.5670 - mae:
2.4159 - val_loss: 10.8837 - val_mae: 2.5861
Epoch 37/40
819/819 [==============================] - 390s 477ms/step - loss: 9.5213 - mae:
2.4076 - val_loss: 10.8181 - val_mae: 2.5847
Epoch 38/40
819/819 [==============================] - 385s 470ms/step - loss: 9.3962 - mae:
2.3925 - val_loss: 11.1505 - val_mae: 2.6206
Epoch 39/40
819/819 [==============================] - 357s 436ms/step - loss: 9.3970 - mae:
2.3920 - val_loss: 11.2715 - val_mae: 2.6275
Epoch 40/40
819/819 [==============================] - 226s 276ms/step - loss: 9.3911 - mae:
2.3914 - val_loss: 10.9976 - val_mae: 2.6020
```

[31]:

```python
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE for GRU with dropout")
plt.legend()
plt.show()
```



[32]:

```
""" Training and evaluating a dropout-regularized, stacked GRU model """

inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
x = layers.GRU(32, recurrent_dropout=0.5, return_sequences=True)(inputs)
x = layers.GRU(32, recurrent_dropout=0.5)(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_stacked_gru_dropout.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=30,
                    validation_data=val_dataset,
                    callbacks=callbacks)
model = keras.models.load_model("jena_stacked_gru_dropout.keras")
print(f"Test MAE: {model.evaluate(test_dataset)[1]:.2f}")
```

```
WARNING:tensorflow:Layer gru_2 will not use cuDNN kernels since it doesn't meet
the criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer gru_3 will not use cuDNN kernels since it doesn't meet
the criteria. It will use a generic GPU kernel as fallback when running on GPU.
Epoch 1/30
819/819 [==============================] - 471s 572ms/step - loss: 24.8814 -
mae: 3.6886 - val_loss: 9.6731 - val_mae: 2.4093
Epoch 2/30
819/819 [==============================] - 471s 575ms/step - loss: 14.0324 -
mae: 2.9016 - val_loss: 8.9929 - val_mae: 2.3295
Epoch 3/30
819/819 [==============================] - 486s 594ms/step - loss: 13.2505 -
mae: 2.8226 - val_loss: 8.7357 - val_mae: 2.2919
Epoch 4/30
819/819 [==============================] - 462s 564ms/step - loss: 12.7158 -
mae: 2.7635 - val_loss: 8.9419 - val_mae: 2.3215
Epoch 5/30
819/819 [==============================] - 457s 558ms/step - loss: 12.2671 -
mae: 2.7143 - val_loss: 8.7233 - val_mae: 2.2870
Epoch 6/30
819/819 [==============================] - 469s 572ms/step - loss: 11.7947 -
mae: 2.6657 - val_loss: 8.9836 - val_mae: 2.3171
Epoch 7/30
819/819 [==============================] - 523s 638ms/step - loss: 11.4104 -
mae: 2.6252 - val_loss: 8.8971 - val_mae: 2.3083
Epoch 8/30
819/819 [==============================] - 523s 638ms/step - loss: 11.0211 -
mae: 2.5781 - val_loss: 9.3330 - val_mae: 2.3614
Epoch 9/30
819/819 [==============================] - 482s 588ms/step - loss: 10.6505 -
mae: 2.5360 - val_loss: 9.6855 - val_mae: 2.4148
Epoch 10/30
819/819 [==============================] - 478s 583ms/step - loss: 10.4170 -
mae: 2.5056 - val_loss: 9.2283 - val_mae: 2.3555
Epoch 11/30
819/819 [==============================] - 484s 590ms/step - loss: 10.0882 -
mae: 2.4689 - val_loss: 9.5193 - val_mae: 2.3946
Epoch 12/30
819/819 [==============================] - 480s 586ms/step - loss: 9.8373 - mae:
2.4377 - val_loss: 9.6191 - val_mae: 2.4122
Epoch 13/30
819/819 [==============================] - 483s 590ms/step - loss: 9.6205 - mae:
2.4085 - val_loss: 9.9737 - val_mae: 2.4641
Epoch 14/30
819/819 [==============================] - 482s 588ms/step - loss: 9.4262 - mae:
2.3837 - val_loss: 10.1354 - val_mae: 2.4786
Epoch 15/30
819/819 [==============================] - 478s 584ms/step - loss: 9.1830 - mae:
```

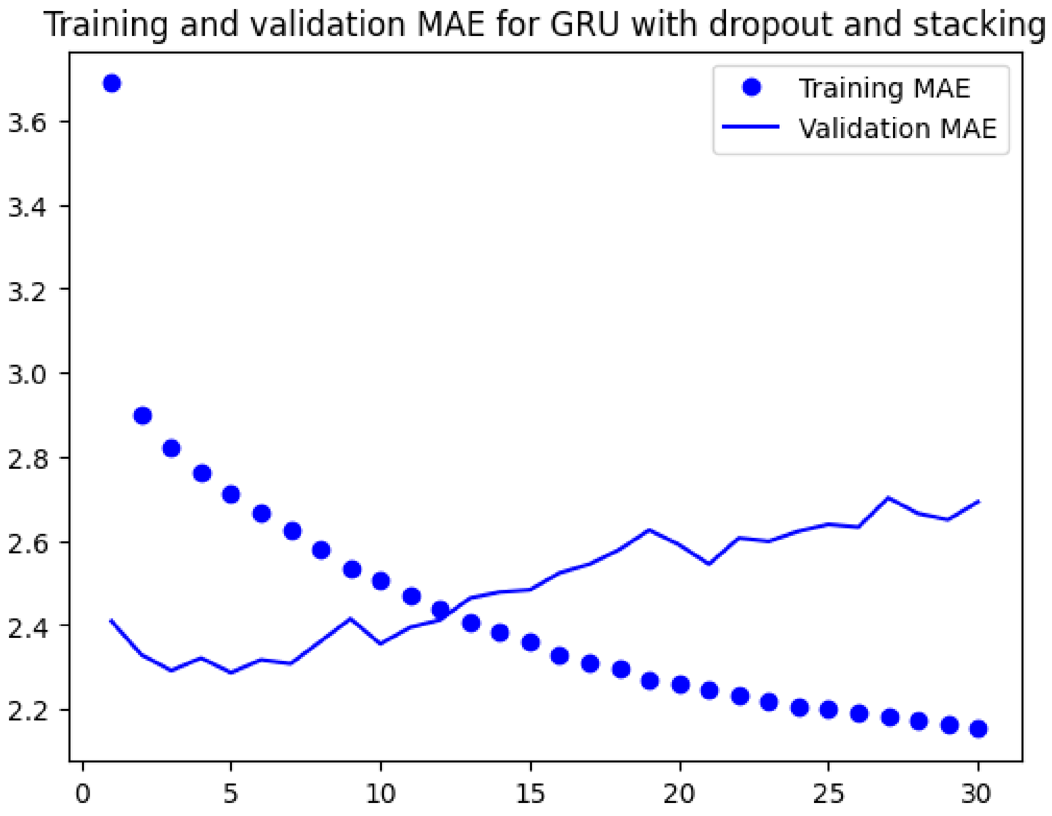
```
2.3592 - val_loss: 10.2359 - val_mae: 2.4837
Epoch 16/30
819/819 [==============================] - 483s 590ms/step - loss: 9.0052 - mae:
2.3307 - val_loss: 10.5599 - val_mae: 2.5239
Epoch 17/30
819/819 [==============================] - 483s 589ms/step - loss: 8.8498 - mae:
2.3124 - val_loss: 10.6341 - val_mae: 2.5449
Epoch 18/30
819/819 [==============================] - 482s 589ms/step - loss: 8.7102 - mae:
2.2954 - val_loss: 11.0961 - val_mae: 2.5794
Epoch 19/30
819/819 [==============================] - 482s 589ms/step - loss: 8.5497 - mae:
2.2692 - val_loss: 11.2775 - val_mae: 2.6259
Epoch 20/30
819/819 [==============================] - 478s 583ms/step - loss: 8.4650 - mae:
2.2607 - val_loss: 11.0376 - val_mae: 2.5911
Epoch 21/30
819/819 [==============================] - 483s 589ms/step - loss: 8.3542 - mae:
2.2472 - val_loss: 10.7542 - val_mae: 2.5451
Epoch 22/30
819/819 [==============================] - 481s 588ms/step - loss: 8.2468 - mae:
2.2318 - val_loss: 11.1758 - val_mae: 2.6069
Epoch 23/30
819/819 [==============================] - 486s 593ms/step - loss: 8.1296 - mae:
2.2187 - val_loss: 11.3402 - val_mae: 2.5990
Epoch 24/30
819/819 [==============================] - 483s 589ms/step - loss: 8.0633 - mae:
2.2077 - val_loss: 11.5350 - val_mae: 2.6232
Epoch 25/30
819/819 [==============================] - 480s 587ms/step - loss: 8.0453 - mae:
2.2032 - val_loss: 11.5807 - val_mae: 2.6396
Epoch 26/30
819/819 [==============================] - 479s 585ms/step - loss: 7.9752 - mae:
2.1927 - val_loss: 11.4600 - val_mae: 2.6330
Epoch 27/30
819/819 [==============================] - 482s 589ms/step - loss: 7.8775 - mae:
2.1814 - val_loss: 11.9908 - val_mae: 2.7024
Epoch 28/30
819/819 [==============================] - 482s 589ms/step - loss: 7.8095 - mae:
2.1722 - val_loss: 11.9296 - val_mae: 2.6653
Epoch 29/30
819/819 [==============================] - 487s 595ms/step - loss: 7.7438 - mae:
2.1627 - val_loss: 11.5044 - val_mae: 2.6505
Epoch 30/30
819/819 [==============================] - 482s 588ms/step - loss: 7.6828 - mae:
2.1531 - val_loss: 12.0060 - val_mae: 2.6928
WARNING:tensorflow:Layer gru_2 will not use cuDNN kernels since it doesn't meet
the criteria. It will use a generic GPU kernel as fallback when running on GPU.
```

```
WARNING:tensorflow:Layer gru_3 will not use cuDNN kernels since it doesn't meet
the criteria. It will use a generic GPU kernel as fallback when running on GPU.
405/405 [==============================] - 29s 72ms/step - loss: 9.6111 - mae:
2.4283
Test MAE: 2.43
```

[33]:

```python
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE for GRU with dropout and stacking")
plt.legend()
plt.show()
```



[35]:

```python
#
inputs = keras.Input(shape=(sequence_length, raw_data.shape[-1]))
```

```
#
x = layers.Conv1D(8, 24, activation="relu")(inputs)
x = layers.MaxPooling1D(2)(x)
x = layers.Conv1D(8, 12, activation="relu")(x)
x = layers.MaxPooling1D(2)(x)
x = layers.Conv1D(8, 6, activation="relu")(x)

#      GRU
x = layers.GRU(32, recurrent_dropout=0.25, return_sequences=True)(x)

#
x = layers.GlobalAveragePooling1D()(x)

#
outputs = layers.Dense(1)(x)

#
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint("jena_lstm_dropout.keras",
                                    save_best_only=True)
]
model.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
history = model.fit(train_dataset,
                    epochs=30,
                    validation_data=val_dataset,
                    callbacks=callbacks)
```

```
WARNING:tensorflow:Layer gru_5 will not use cuDNN kernels since it doesn't meet
the criteria. It will use a generic GPU kernel as fallback when running on GPU.
Epoch 1/30
819/819 [==============================] - 38s 44ms/step - loss: 27.3607 - mae:
3.9523 - val_loss: 14.7524 - val_mae: 2.9991
Epoch 2/30
819/819 [==============================] - 37s 45ms/step - loss: 12.9019 - mae:
2.8470 - val_loss: 15.8174 - val_mae: 3.1046
Epoch 3/30
819/819 [==============================] - 35s 43ms/step - loss: 11.4599 - mae:
2.6755 - val_loss: 13.9399 - val_mae: 2.9414
Epoch 4/30
819/819 [==============================] - 36s 43ms/step - loss: 10.4308 - mae:
2.5471 - val_loss: 16.9300 - val_mae: 3.2012
Epoch 5/30
819/819 [==============================] - 35s 43ms/step - loss: 9.6771 - mae:
2.4528 - val_loss: 15.5660 - val_mae: 3.0970
Epoch 6/30
```

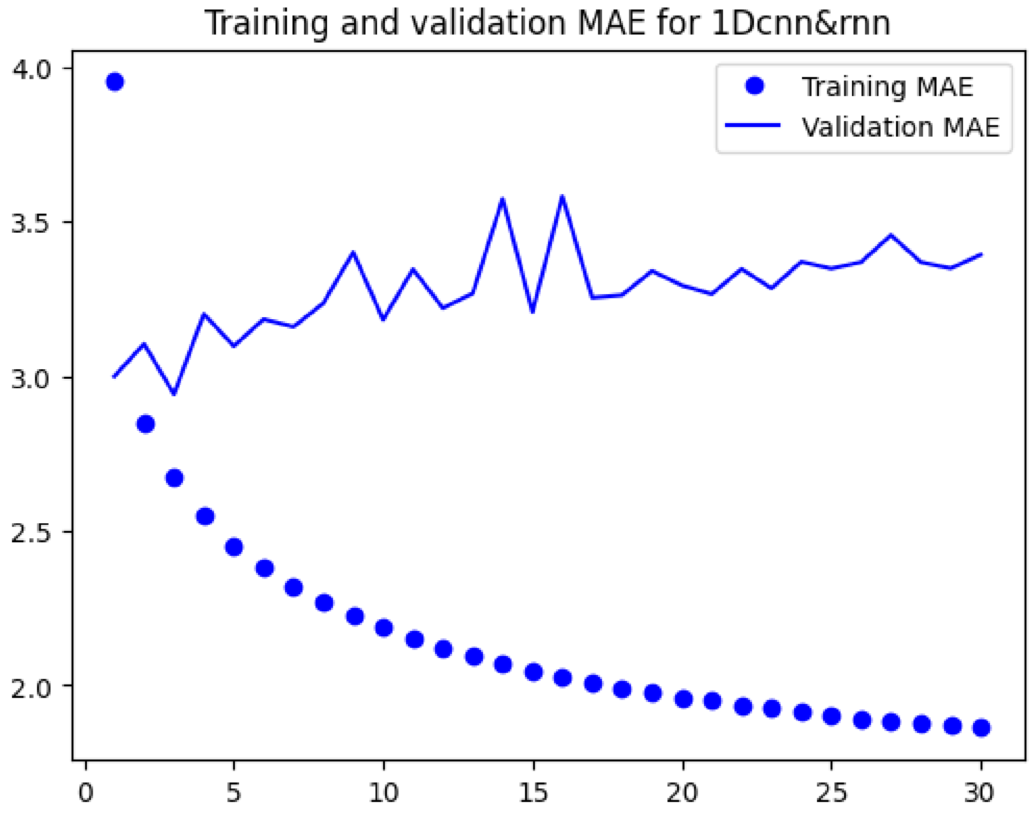
```
819/819 [==============================] - 35s 43ms/step - loss: 9.1625 - mae:
2.3825 - val_loss: 16.3681 - val_mae: 3.1839
Epoch 7/30
819/819 [==============================] - 35s 43ms/step - loss: 8.7075 - mae:
2.3202 - val_loss: 16.0830 - val_mae: 3.1595
Epoch 8/30
819/819 [==============================] - 36s 44ms/step - loss: 8.3371 - mae:
2.2680 - val_loss: 17.2523 - val_mae: 3.2358
Epoch 9/30
819/819 [==============================] - 35s 43ms/step - loss: 8.0167 - mae:
2.2244 - val_loss: 18.8692 - val_mae: 3.4009
Epoch 10/30
819/819 [==============================] - 35s 43ms/step - loss: 7.7516 - mae:
2.1869 - val_loss: 16.4811 - val_mae: 3.1823
Epoch 11/30
819/819 [==============================] - 36s 44ms/step - loss: 7.4950 - mae:
2.1499 - val_loss: 18.4007 - val_mae: 3.3460
Epoch 12/30
819/819 [==============================] - 36s 44ms/step - loss: 7.3006 - mae:
2.1217 - val_loss: 16.9697 - val_mae: 3.2202
Epoch 13/30
819/819 [==============================] - 36s 44ms/step - loss: 7.1069 - mae:
2.0938 - val_loss: 17.3617 - val_mae: 3.2676
Epoch 14/30
819/819 [==============================] - 37s 45ms/step - loss: 6.9346 - mae:
2.0689 - val_loss: 20.6704 - val_mae: 3.5733
Epoch 15/30
819/819 [==============================] - 35s 43ms/step - loss: 6.7910 - mae:
2.0462 - val_loss: 16.6892 - val_mae: 3.2078
Epoch 16/30
819/819 [==============================] - 35s 43ms/step - loss: 6.6589 - mae:
2.0260 - val_loss: 20.6466 - val_mae: 3.5813
Epoch 17/30
819/819 [==============================] - 36s 43ms/step - loss: 6.5379 - mae:
2.0077 - val_loss: 17.2323 - val_mae: 3.2536
Epoch 18/30
819/819 [==============================] - 35s 43ms/step - loss: 6.4159 - mae:
1.9898 - val_loss: 17.3701 - val_mae: 3.2622
Epoch 19/30
819/819 [==============================] - 35s 43ms/step - loss: 6.3437 - mae:
1.9778 - val_loss: 18.2291 - val_mae: 3.3402
Epoch 20/30
819/819 [==============================] - 39s 48ms/step - loss: 6.2321 - mae:
1.9602 - val_loss: 17.6734 - val_mae: 3.2936
Epoch 21/30
819/819 [==============================] - 36s 44ms/step - loss: 6.1718 - mae:
1.9510 - val_loss: 17.3840 - val_mae: 3.2655
Epoch 22/30
```

```
819/819 [==============================] - 36s 44ms/step - loss: 6.0743 - mae:
1.9366 - val_loss: 18.2140 - val_mae: 3.3466
Epoch 23/30
819/819 [==============================] - 36s 44ms/step - loss: 6.0217 - mae:
1.9273 - val_loss: 17.6646 - val_mae: 3.2842
Epoch 24/30
819/819 [==============================] - 36s 44ms/step - loss: 5.9359 - mae:
1.9151 - val_loss: 18.6234 - val_mae: 3.3700
Epoch 25/30
819/819 [==============================] - 35s 43ms/step - loss: 5.8879 - mae:
1.9049 - val_loss: 18.2151 - val_mae: 3.3477
Epoch 26/30
819/819 [==============================] - 35s 43ms/step - loss: 5.8110 - mae:
1.8932 - val_loss: 18.7006 - val_mae: 3.3686
Epoch 27/30
819/819 [==============================] - 35s 43ms/step - loss: 5.7623 - mae:
1.8858 - val_loss: 19.6707 - val_mae: 3.4564
Epoch 28/30
819/819 [==============================] - 35s 43ms/step - loss: 5.7132 - mae:
1.8772 - val_loss: 18.5733 - val_mae: 3.3685
Epoch 29/30
819/819 [==============================] - 35s 43ms/step - loss: 5.6717 - mae:
1.8700 - val_loss: 18.1998 - val_mae: 3.3486
Epoch 30/30
819/819 [==============================] - 35s 43ms/step - loss: 5.6247 - mae:
1.8625 - val_loss: 18.8962 - val_mae: 3.3926
```

[36]:

```python
""" Plotting results """

import matplotlib.pyplot as plt
loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure()
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE for 1Dcnn&rnn")
plt.legend()
plt.show()
```


Training and validation MAE for 1Dcnn&rnn
Training MAE
Validation MAE
4.0
3.5
3.0
2.5
2.0
0
5
10
15
20
25
30