MLP for XDot21

November 24, 2024

[5]:

```
# first neural network with keras tutorial
import tensorflow as tf
from tensorflow import keras
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
#from keras.wrappers.scikit_learn import KerasRegressor
from scikeras.wrappers import KerasClassifier, KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.datasets import make_regression
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error
from numpy import asarray
from numpy import unique
from numpy import argmax
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
import numpy as np
from tensorflow.keras.utils import plot_model
from sklearn.preprocessing import StandardScaler
```

[7]:

```
# load the dataset
dataset = loadtxt('Data/Dataset_xdot21.csv', delimiter=',')
```

[9]:

```
# #ESSAI 3  -----OK
X = dataset[:,0:6]
y = dataset[:,6]
scalarX, scalarY = MinMaxScaler(feature_range=(0,1)),␣
  ↪MinMaxScaler(feature_range=(0,0.75))
scalarX.fit(X)
scalarY.fit(y.reshape(133,1))
X = scalarX.transform(X)
y=np.array(y).reshape(133,1)
y = scalarY.transform(y)
```

[11]:

```
print(y)
```

```
[[0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
```

```
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.        ]
[0.00143569]
[0.01666002]
[0.0468037 ]
[0.07236852]
[0.126294  ]
[0.18900111]
[0.26477931]
[0.35459409]
[0.35459409]
[0.48576325]
[0.48576325]
[0.48576325]
[0.48576325]
[0.75      ]
[0.75      ]
[0.75      ]
[0.75      ]
[0.75      ]
```

```
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]
 [0.75      ]]
```

[13]:

```python
# define the keras model
model = Sequential()
model.add(Dense(20, input_dim=6, kernel_initializer='normal', activation='relu')) #kernel_initializer='normal'
model.add(Dense(1, kernel_initializer='normal', activation='linear')) #linear
print(model.summary())
```

```
H:\Anaconda\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do
not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in the model
```

```
instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
Model: "sequential"


 Layer (type)                     Output Shape                  Param #

 dense (Dense)                    (None, 20)                        140

 dense_1 (Dense)                  (None, 1)                          21


 Total params: 161 (644.00 B)

 Trainable params: 161 (644.00 B)

 Non-trainable params: 0 (0.00 B)

None
```

[15]:

```python
model.compile(optimizer='adam', loss='mean_absolute_error')
```

[17]:

```python
# fit the keras model on the dataset
history = model.fit(X, y, epochs=100, batch_size=8, verbose=2,␣
 ↪validation_split=0.25)
```

```
Epoch 1/100
13/13 - 1s - 41ms/step - loss: 0.0978 - val_loss: 0.7501
Epoch 2/100
13/13 - 0s - 3ms/step - loss: 0.0953 - val_loss: 0.7495
Epoch 3/100
13/13 - 0s - 3ms/step - loss: 0.0951 - val_loss: 0.7501
Epoch 4/100
13/13 - 0s - 3ms/step - loss: 0.0949 - val_loss: 0.7496
Epoch 5/100
13/13 - 0s - 3ms/step - loss: 0.0949 - val_loss: 0.7496
Epoch 6/100
13/13 - 0s - 3ms/step - loss: 0.0949 - val_loss: 0.7493
Epoch 7/100
13/13 - 0s - 3ms/step - loss: 0.0949 - val_loss: 0.7492
Epoch 8/100
13/13 - 0s - 3ms/step - loss: 0.0948 - val_loss: 0.7489
Epoch 9/100
13/13 - 0s - 3ms/step - loss: 0.0947 - val_loss: 0.7469
Epoch 10/100
```

```
13/13 - 0s - 3ms/step - loss: 0.0947 - val_loss: 0.7467
Epoch 11/100
13/13 - 0s - 3ms/step - loss: 0.0945 - val_loss: 0.7477
Epoch 12/100
13/13 - 0s - 3ms/step - loss: 0.0945 - val_loss: 0.7465
Epoch 13/100
13/13 - 0s - 3ms/step - loss: 0.0945 - val_loss: 0.7453
Epoch 14/100
13/13 - 0s - 3ms/step - loss: 0.0944 - val_loss: 0.7462
Epoch 15/100
13/13 - 0s - 3ms/step - loss: 0.0947 - val_loss: 0.7463
Epoch 16/100
13/13 - 0s - 3ms/step - loss: 0.0948 - val_loss: 0.7439
Epoch 17/100
13/13 - 0s - 3ms/step - loss: 0.0947 - val_loss: 0.7420
Epoch 18/100
13/13 - 0s - 3ms/step - loss: 0.0942 - val_loss: 0.7410
Epoch 19/100
13/13 - 0s - 3ms/step - loss: 0.0942 - val_loss: 0.7402
Epoch 20/100
13/13 - 0s - 3ms/step - loss: 0.0941 - val_loss: 0.7403
Epoch 21/100
13/13 - 0s - 3ms/step - loss: 0.0939 - val_loss: 0.7394
Epoch 22/100
13/13 - 0s - 3ms/step - loss: 0.0937 - val_loss: 0.7380
Epoch 23/100
13/13 - 0s - 3ms/step - loss: 0.0935 - val_loss: 0.7328
Epoch 24/100
13/13 - 0s - 3ms/step - loss: 0.0934 - val_loss: 0.7342
Epoch 25/100
13/13 - 0s - 3ms/step - loss: 0.0932 - val_loss: 0.7299
Epoch 26/100
13/13 - 0s - 3ms/step - loss: 0.0931 - val_loss: 0.7321
Epoch 27/100
13/13 - 0s - 3ms/step - loss: 0.0926 - val_loss: 0.7283
Epoch 28/100
13/13 - 0s - 3ms/step - loss: 0.0924 - val_loss: 0.7248
Epoch 29/100
13/13 - 0s - 3ms/step - loss: 0.0924 - val_loss: 0.7163
Epoch 30/100
13/13 - 0s - 3ms/step - loss: 0.0916 - val_loss: 0.7244
Epoch 31/100
13/13 - 0s - 3ms/step - loss: 0.0918 - val_loss: 0.7176
Epoch 32/100
13/13 - 0s - 3ms/step - loss: 0.0914 - val_loss: 0.7107
Epoch 33/100
13/13 - 0s - 3ms/step - loss: 0.0914 - val_loss: 0.7172
Epoch 34/100
```

```
13/13 - 0s - 3ms/step - loss: 0.0905 - val_loss: 0.7048
Epoch 35/100
13/13 - 0s - 3ms/step - loss: 0.0903 - val_loss: 0.7021
Epoch 36/100
13/13 - 0s - 3ms/step - loss: 0.0894 - val_loss: 0.6997
Epoch 37/100
13/13 - 0s - 3ms/step - loss: 0.0890 - val_loss: 0.6915
Epoch 38/100
13/13 - 0s - 3ms/step - loss: 0.0884 - val_loss: 0.6881
Epoch 39/100
13/13 - 0s - 3ms/step - loss: 0.0877 - val_loss: 0.6805
Epoch 40/100
13/13 - 0s - 3ms/step - loss: 0.0870 - val_loss: 0.6799
Epoch 41/100
13/13 - 0s - 3ms/step - loss: 0.0861 - val_loss: 0.6683
Epoch 42/100
13/13 - 0s - 3ms/step - loss: 0.0859 - val_loss: 0.6601
Epoch 43/100
13/13 - 0s - 3ms/step - loss: 0.0852 - val_loss: 0.6502
Epoch 44/100
13/13 - 0s - 3ms/step - loss: 0.0841 - val_loss: 0.6478
Epoch 45/100
13/13 - 0s - 3ms/step - loss: 0.0834 - val_loss: 0.6484
Epoch 46/100
13/13 - 0s - 3ms/step - loss: 0.0829 - val_loss: 0.6340
Epoch 47/100
13/13 - 0s - 3ms/step - loss: 0.0815 - val_loss: 0.6313
Epoch 48/100
13/13 - 0s - 3ms/step - loss: 0.0817 - val_loss: 0.6266
Epoch 49/100
13/13 - 0s - 3ms/step - loss: 0.0806 - val_loss: 0.6197
Epoch 50/100
13/13 - 0s - 3ms/step - loss: 0.0792 - val_loss: 0.5962
Epoch 51/100
13/13 - 0s - 3ms/step - loss: 0.0784 - val_loss: 0.6094
Epoch 52/100
13/13 - 0s - 3ms/step - loss: 0.0775 - val_loss: 0.5947
Epoch 53/100
13/13 - 0s - 3ms/step - loss: 0.0760 - val_loss: 0.5865
Epoch 54/100
13/13 - 0s - 3ms/step - loss: 0.0753 - val_loss: 0.5810
Epoch 55/100
13/13 - 0s - 3ms/step - loss: 0.0746 - val_loss: 0.5677
Epoch 56/100
13/13 - 0s - 3ms/step - loss: 0.0743 - val_loss: 0.5686
Epoch 57/100
13/13 - 0s - 3ms/step - loss: 0.0730 - val_loss: 0.5545
Epoch 58/100
```

```
13/13 - 0s - 3ms/step - loss: 0.0719 - val_loss: 0.5451
Epoch 59/100
13/13 - 0s - 3ms/step - loss: 0.0714 - val_loss: 0.5372
Epoch 60/100
13/13 - 0s - 3ms/step - loss: 0.0702 - val_loss: 0.5320
Epoch 61/100
13/13 - 0s - 3ms/step - loss: 0.0699 - val_loss: 0.5103
Epoch 62/100
13/13 - 0s - 3ms/step - loss: 0.0681 - val_loss: 0.5164
Epoch 63/100
13/13 - 0s - 3ms/step - loss: 0.0674 - val_loss: 0.4971
Epoch 64/100
13/13 - 0s - 3ms/step - loss: 0.0671 - val_loss: 0.5109
Epoch 65/100
13/13 - 0s - 3ms/step - loss: 0.0670 - val_loss: 0.4662
Epoch 66/100
13/13 - 0s - 3ms/step - loss: 0.0651 - val_loss: 0.4892
Epoch 67/100
13/13 - 0s - 3ms/step - loss: 0.0619 - val_loss: 0.4523
Epoch 68/100
13/13 - 0s - 3ms/step - loss: 0.0657 - val_loss: 0.4408
Epoch 69/100
13/13 - 0s - 3ms/step - loss: 0.0612 - val_loss: 0.4812
Epoch 70/100
13/13 - 0s - 3ms/step - loss: 0.0608 - val_loss: 0.4492
Epoch 71/100
13/13 - 0s - 3ms/step - loss: 0.0590 - val_loss: 0.4235
Epoch 72/100
13/13 - 0s - 3ms/step - loss: 0.0573 - val_loss: 0.4387
Epoch 73/100
13/13 - 0s - 3ms/step - loss: 0.0565 - val_loss: 0.4187
Epoch 74/100
13/13 - 0s - 3ms/step - loss: 0.0561 - val_loss: 0.4117
Epoch 75/100
13/13 - 0s - 3ms/step - loss: 0.0549 - val_loss: 0.4174
Epoch 76/100
13/13 - 0s - 3ms/step - loss: 0.0542 - val_loss: 0.3982
Epoch 77/100
13/13 - 0s - 3ms/step - loss: 0.0532 - val_loss: 0.3945
Epoch 78/100
13/13 - 0s - 3ms/step - loss: 0.0527 - val_loss: 0.3723
Epoch 79/100
13/13 - 0s - 3ms/step - loss: 0.0515 - val_loss: 0.3942
Epoch 80/100
13/13 - 0s - 3ms/step - loss: 0.0518 - val_loss: 0.3813
Epoch 81/100
13/13 - 0s - 3ms/step - loss: 0.0503 - val_loss: 0.3585
Epoch 82/100
```

```
13/13 - 0s - 3ms/step - loss: 0.0510 - val_loss: 0.3426
Epoch 83/100
13/13 - 0s - 3ms/step - loss: 0.0510 - val_loss: 0.3936
Epoch 84/100
13/13 - 0s - 3ms/step - loss: 0.0493 - val_loss: 0.3625
Epoch 85/100
13/13 - 0s - 3ms/step - loss: 0.0470 - val_loss: 0.3095
Epoch 86/100
13/13 - 0s - 3ms/step - loss: 0.0481 - val_loss: 0.3306
Epoch 87/100
13/13 - 0s - 3ms/step - loss: 0.0460 - val_loss: 0.3412
Epoch 88/100
13/13 - 0s - 3ms/step - loss: 0.0454 - val_loss: 0.3315
Epoch 89/100
13/13 - 0s - 3ms/step - loss: 0.0445 - val_loss: 0.3332
Epoch 90/100
13/13 - 0s - 5ms/step - loss: 0.0440 - val_loss: 0.3183
Epoch 91/100
13/13 - 0s - 4ms/step - loss: 0.0438 - val_loss: 0.3062
Epoch 92/100
13/13 - 0s - 3ms/step - loss: 0.0435 - val_loss: 0.3020
Epoch 93/100
13/13 - 0s - 3ms/step - loss: 0.0441 - val_loss: 0.3070
Epoch 94/100
13/13 - 0s - 3ms/step - loss: 0.0423 - val_loss: 0.2956
Epoch 95/100
13/13 - 0s - 3ms/step - loss: 0.0425 - val_loss: 0.2973
Epoch 96/100
13/13 - 0s - 3ms/step - loss: 0.0415 - val_loss: 0.2798
Epoch 97/100
13/13 - 0s - 3ms/step - loss: 0.0408 - val_loss: 0.2772
Epoch 98/100
13/13 - 0s - 3ms/step - loss: 0.0402 - val_loss: 0.3009
Epoch 99/100
13/13 - 0s - 3ms/step - loss: 0.0411 - val_loss: 0.3176
Epoch 100/100
13/13 - 0s - 3ms/step - loss: 0.0416 - val_loss: 0.2685
```

[18]:

```python
# evaluate on test set
yhat = model.predict(X)
error = mean_absolute_error(y, yhat)
print('MAE: %.5f' % error)
```

```
5/5                0s 5ms/step
MAE: 0.09767
```

[21]:

```python
print(yhat)
```

```
[[-2.78289081e-04]
```

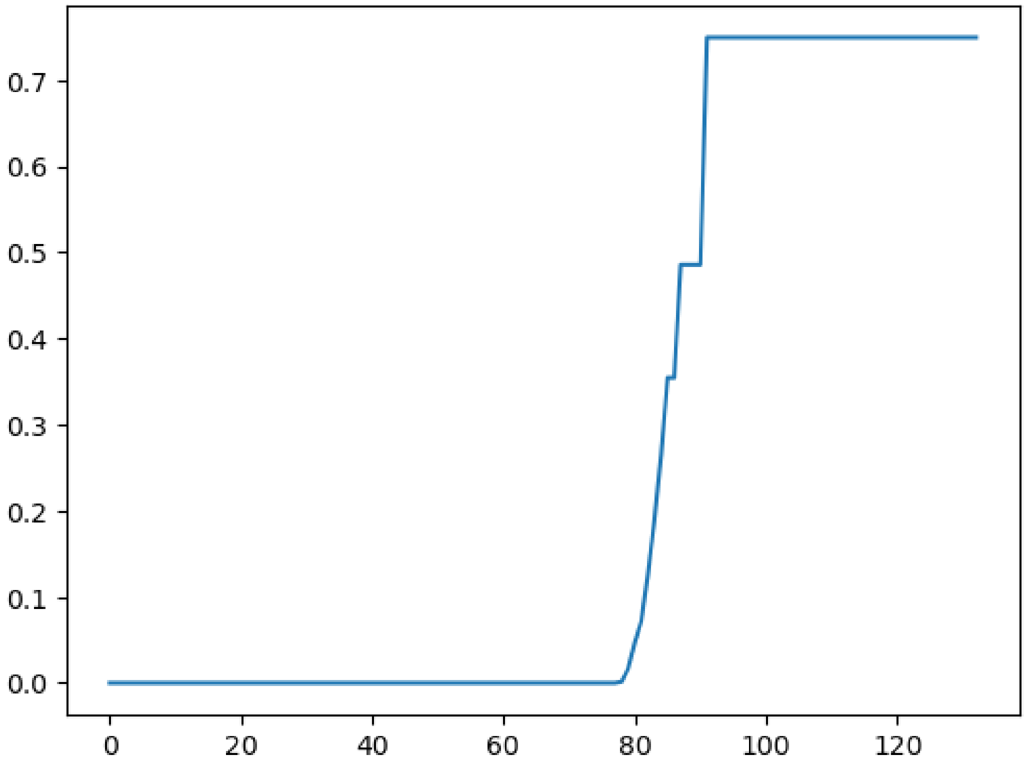
```
[-3.98990000e-04]
[-3.37593956e-04]
[-4.05335799e-04]
[-2.96597602e-04]
[-3.62993916e-04]
[-2.86926515e-04]
[-6.06518355e-04]
[-6.06104732e-04]
[-4.59668227e-04]
[-5.97294187e-04]
[-4.57932008e-04]
[-7.48066348e-04]
[-6.63720304e-04]
[-4.67212521e-04]
[-5.07488498e-04]
[-4.46535298e-04]
[-1.19354809e-04]
[-1.15707400e-04]
[-1.61209493e-04]
[-8.43416550e-04]
[-5.25716692e-04]
[-2.89229676e-04]
[-8.44891183e-05]
[-5.32681588e-04]
[ 1.69139355e-04]
[ 1.31362118e-04]
[ 4.68385406e-06]
[ 1.27462903e-04]
[ 2.32188031e-05]
[-3.58478690e-04]
[-1.88919366e-04]
[-2.31887214e-04]
[ 1.61579112e-04]
[-1.09067187e-05]
[-1.72194792e-04]
[-1.70679297e-04]
[-6.65832194e-05]
[-2.68367119e-04]
[-2.54441053e-04]
[ 5.56441955e-05]
[ 1.32537098e-04]
[ 1.14697730e-04]
[-1.55188609e-04]
[-1.28399581e-04]
[-1.07251690e-04]
[-3.14894482e-04]
[-1.19261793e-04]
[-1.05537474e-04]
```

```
[-3.09258234e-04]
[-2.41101487e-04]
[-1.73029257e-05]
[ 1.03782713e-01]
[ 2.88738403e-04]
[-2.63463589e-04]
[ 3.99169289e-02]
[-6.56398479e-05]
[-4.76766145e-05]
[ 4.96660359e-05]
[ 2.02226415e-01]
[ 3.20652639e-03]
[-2.96172220e-06]
[ 1.11814530e-03]
[ 2.07318425e-01]
[ 4.84937988e-02]
[ 1.06840558e-01]
[ 1.99537550e-04]
[ 5.24555966e-02]
[ 3.91239300e-05]
[-1.17820688e-04]
[ 1.37573807e-05]
[-2.61047448e-04]
[ 8.57539475e-02]
[ 1.60566401e-02]
[-9.20100138e-06]
[ 1.13928996e-01]
[-1.66893413e-04]
[ 1.03208993e-04]
[-1.39015226e-03]
[ 8.76784697e-03]
[ 1.46812931e-01]
[ 1.99252635e-01]
[ 2.13367984e-01]
[ 1.81033105e-01]
[ 3.91103774e-01]
[ 3.45948875e-01]
[ 3.62819254e-01]
[ 4.94096100e-01]
[ 4.84270751e-01]
[ 5.90294719e-01]
[ 4.30817842e-01]
[ 2.45215371e-01]
[ 1.46851867e-01]
[ 2.07783908e-01]
[ 6.63169622e-01]
[ 6.48764491e-01]
[ 7.09936857e-01]
```

```
 [ 5.67083657e-01]
 [ 5.92549026e-01]
 [ 5.92538655e-01]
 [ 7.16810048e-01]
 [ 5.30878603e-01]
 [ 6.28787220e-01]
 [ 5.99280357e-01]
 [ 4.32558060e-01]
 [ 2.33480290e-01]
 [ 4.55289930e-02]
 [ 5.75717032e-01]
 [ 4.55345005e-01]
 [ 4.27424103e-01]
 [ 5.45565307e-01]
 [ 5.39102912e-01]
 [ 4.60255057e-01]
 [ 4.33492303e-01]
 [ 5.64291000e-01]
 [ 5.10596752e-01]
 [ 2.72761416e-02]
 [ 5.94664991e-01]
 [ 4.90116417e-01]
 [ 6.86705112e-01]
 [ 6.76551163e-01]
 [ 5.81535459e-01]
 [ 6.83884084e-01]
 [ 2.50886213e-02]
 [ 5.84972560e-01]
 [ 7.60855913e-01]
 [ 7.08570957e-01]
 [ 5.51136196e-01]
 [ 5.99369526e-01]
 [ 8.83534372e-01]
 [ 3.58196795e-02]
 [ 6.06380031e-02]
 [ 1.08212614e+00]]
```

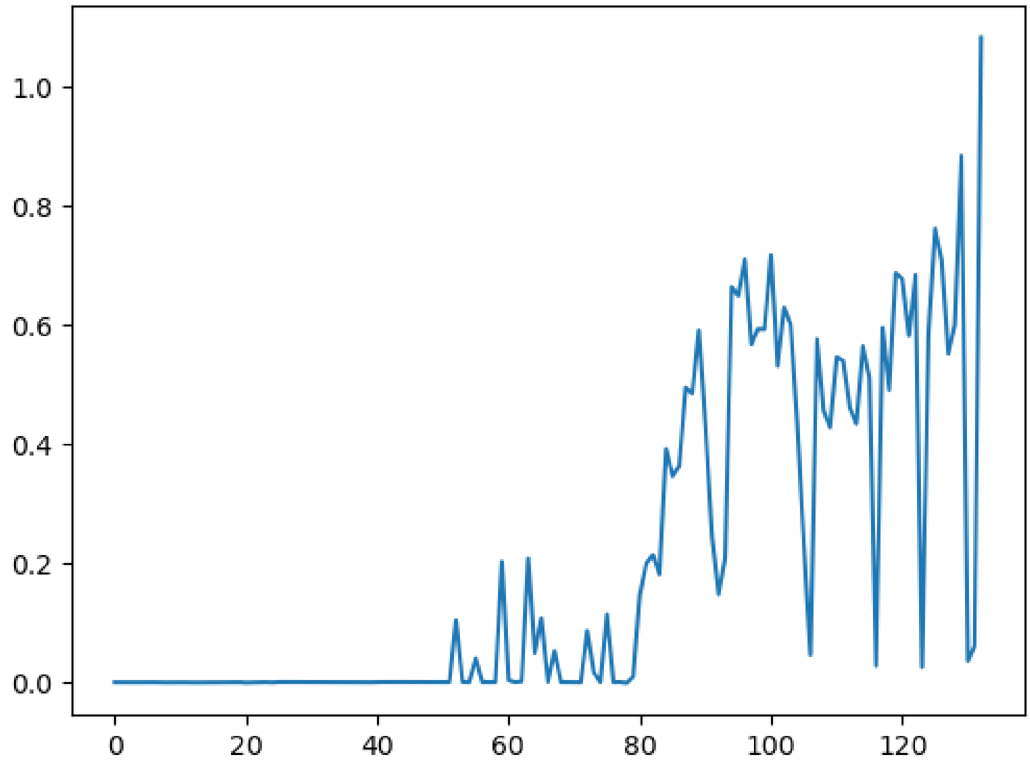
```
[23]: plt.plot(y)
```

```
[23]: [<matplotlib.lines.Line2D at 0x283e8bacec0>]
```



[25]:
```
plt.plot(yhat)
```

[25]:
```
[<matplotlib.lines.Line2D at 0x283eae32270>]
```



```
[27]: print(yhat-y)
```

```
[[-2.78289081e-04]
 [-3.98990000e-04]
 [-3.37593956e-04]
 [-4.05335799e-04]
 [-2.96597602e-04]
 [-3.62993916e-04]
 [-2.86926515e-04]
 [-6.06518355e-04]
 [-6.06104732e-04]
 [-4.59668227e-04]
 [-5.97294187e-04]
 [-4.57932008e-04]
 [-7.48066348e-04]
 [-6.63720304e-04]
 [-4.67212521e-04]
 [-5.07488498e-04]
 [-4.46535298e-04]
 [-1.19354809e-04]
 [-1.15707400e-04]
 [-1.61209493e-04]
```

```
[-8.43416550e-04]
[-5.25716692e-04]
[-2.89229676e-04]
[-8.44891183e-05]
[-5.32681588e-04]
[ 1.69139355e-04]
[ 1.31362118e-04]
[ 4.68385406e-06]
[ 1.27462903e-04]
[ 2.32188031e-05]
[-3.58478690e-04]
[-1.88919366e-04]
[-2.31887214e-04]
[ 1.61579112e-04]
[-1.09067187e-05]
[-1.72194792e-04]
[-1.70679297e-04]
[-6.65832194e-05]
[-2.68367119e-04]
[-2.54441053e-04]
[ 5.56441955e-05]
[ 1.32537098e-04]
[ 1.14697730e-04]
[-1.55188609e-04]
[-1.28399581e-04]
[-1.07251690e-04]
[-3.14894482e-04]
[-1.19261793e-04]
[-1.05537474e-04]
[-3.09258234e-04]
[-2.41101487e-04]
[-1.73029257e-05]
[ 1.03782713e-01]
[ 2.88738403e-04]
[-2.63463589e-04]
[ 3.99169289e-02]
[-6.56398479e-05]
[-4.76766145e-05]
[ 4.96660359e-05]
[ 2.02226415e-01]
[ 3.20652639e-03]
[-2.96172220e-06]
[ 1.11814530e-03]
[ 2.07318425e-01]
[ 4.84937988e-02]
[ 1.06840558e-01]
[ 1.99537550e-04]
[ 5.24555966e-02]
[ 3.91239300e-05]
[-1.17820688e-04]
[ 1.37573807e-05]
[-2.61047448e-04]
[ 8.57539475e-02]
[ 1.60566401e-02]
[-9.20100138e-06]
[ 1.13928996e-01]
[-1.66893413e-04]
[ 1.03208993e-04]
[-2.82583843e-03]
[-7.89217798e-03]
[ 1.00009227e-01]
[ 1.26884114e-01]
[ 8.70739844e-02]
[-7.96800731e-03]
[ 1.26324463e-01]
[-8.64521680e-03]
[ 8.22516265e-03]
[ 8.33284832e-03]
[-1.49250053e-03]
[ 1.04531467e-01]
[-5.49454095e-02]
[-5.04784629e-01]
[-6.03148133e-01]
[-5.42216092e-01]
[-8.68303776e-02]
[-1.01235509e-01]
[-4.00631428e-02]
[-1.82916343e-01]
[-1.57450974e-01]
[-1.57461345e-01]
[-3.31899524e-02]
[-2.19121397e-01]
[-1.21212780e-01]
[-1.50719643e-01]
[-3.17441940e-01]
[-5.16519710e-01]
[-7.04471007e-01]
[-1.74282968e-01]
[-2.94654995e-01]
[-3.22575897e-01]
[-2.04434693e-01]
[-2.10897088e-01]
[-2.89744943e-01]
[-3.16507697e-01]
[-1.85709000e-01]
[-2.39403248e-01]
```

```
 [-7.22723858e-01]
 [-1.55335009e-01]
 [-2.59883583e-01]
 [-6.32948875e-02]
 [-7.34488368e-02]
 [-1.68464541e-01]
 [-6.61159158e-02]
 [-7.24911379e-01]
 [-1.65027440e-01]
 [ 1.08559132e-02]
 [-4.14290428e-02]
 [-1.98863804e-01]
 [-1.50630474e-01]
 [ 1.33534372e-01]
 [-7.14180321e-01]
 [-6.89361997e-01]
 [ 3.32126141e-01]]
```

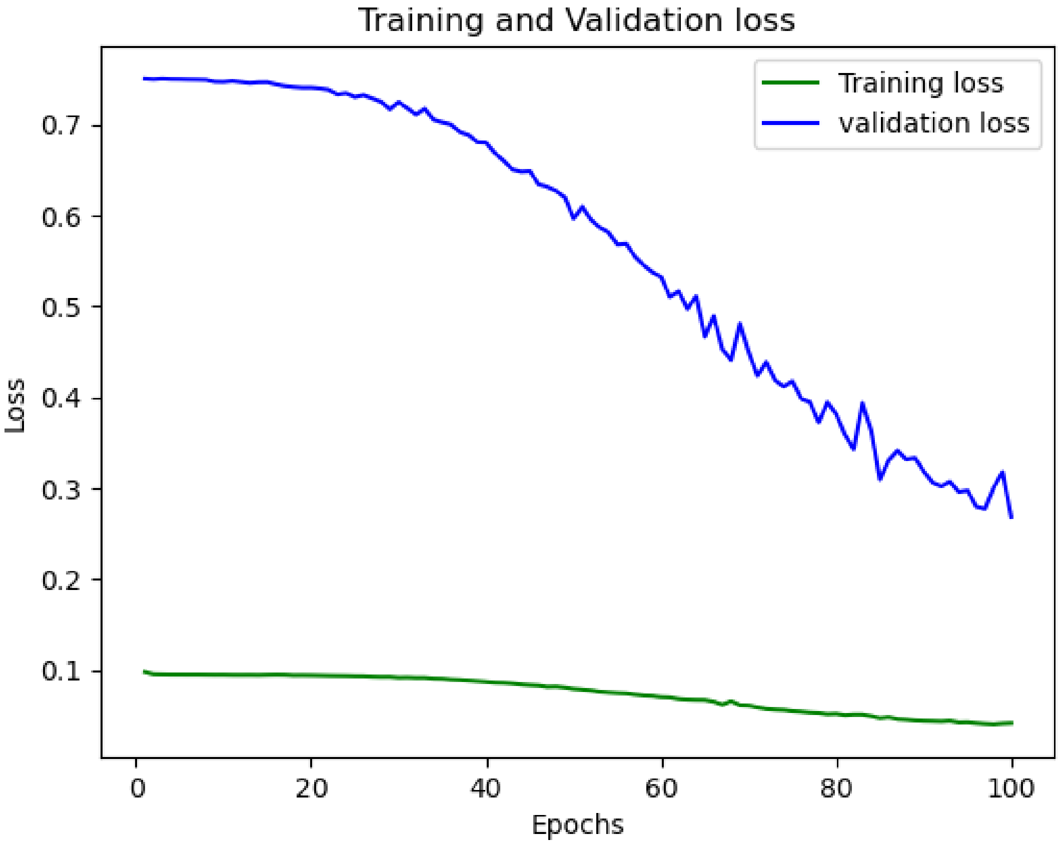
[29]:

```python
print(history.history.keys())
```

```
dict_keys(['loss', 'val_loss'])
```

[31]:

```python
loss_train = history.history['loss']
loss_val = history.history['val_loss']
epochs = range(1,101)
plt.plot(epochs, loss_train, 'g', label='Training loss')
plt.plot(epochs, loss_val, 'b', label='validation loss')
plt.title('Training and Validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```


Training and Validation loss
Training loss
validation loss
Loss
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Epochs
0
20
40
60
80
100

[ ]: