

```
In [1]: 1 import pandas as pd
2 from keras.models import Sequential
3 from keras.layers import Dense, PReLU, Dropout
4 from sklearn.model_selection import train_test_split
5 from tensorflow import keras
6 import matplotlib.pyplot as plt
7 #from sklearn.model_selection import train_test_split
8 from sklearn.metrics import mean_squared_error
9 import tensorflow as tf
10 import numpy as np
```

```
In [2]: 1 df = pd.read_csv("row_size25_vector_size8000.csv")
```

```
In [3]: 1 X = df.iloc[:,7:157]
2 #X = df[['accelerometer_reading_x_0', 'accelerometer_reading_y_0', 'accelerometer_reading_z_0', 'gyro
3 #t = df[['linear_position_x', 'linear_position_y', 'linear_position_z', 'angular_position_phi', 'angul
4 #t = df[['phi_change', 'theta_change', 'psi_change']]
5 t = df[['phi_change']]
```

```
In [4]: 1 t.shape
```

```
Out[4]: (8000, 1)
```

```
In [5]: 1 in_dim = X.shape[1]
2 out_dim = t.shape[1]
```

```
In [6]: 1 in_dim
```

```
Out[6]: 150
```

```
In [7]: 1 out_dim
```

```
Out[7]: 1
```

```
In [8]: 1 X_train, X_test, t_train, t_test = train_test_split(X, t, test_size=0.2)
```

```
In [9]: 1 t_train.shape
```

```
Out[9]: (6400, 1)
```

```
In [10]: 1 model = Sequential()
2 model.add(Dense(256, input_dim=in_dim, activation="sigmoid"))
3 model.add(PReLU())
4 model.add(Dropout(.4))
5 model.add(Dense(128, activation="sigmoid"))
6 model.add(Dense(64, activation="sigmoid"))
7 model.add(Dense(32, activation="sigmoid"))
8 model.add(Dense(out_dim, activation="sigmoid"))
9 model.compile(loss="mse", optimizer="sgd")
```

```
In [11]: 1 model.fit(X_train, t_train, epochs=30, batch_size=12, verbose=2)
```

```
Epoch 1/30
534/534 - 0s - loss: 0.0305
Epoch 2/30
534/534 - 0s - loss: 0.0034
Epoch 3/30
534/534 - 0s - loss: 0.0018
Epoch 4/30
534/534 - 0s - loss: 0.0012
Epoch 5/30
534/534 - 0s - loss: 9.2987e-04
Epoch 6/30
534/534 - 0s - loss: 7.4437e-04
Epoch 7/30
534/534 - 0s - loss: 6.1938e-04
Epoch 8/30
```

```

epoch 8/30
534/534 - 0s - loss: 5.2940e-04
Epoch 9/30
534/534 - 0s - loss: 4.6220e-04
Epoch 10/30
534/534 - 0s - loss: 4.0979e-04
Epoch 11/30
534/534 - 0s - loss: 3.6742e-04
Epoch 12/30
534/534 - 0s - loss: 3.3316e-04
Epoch 13/30
534/534 - 0s - loss: 3.0468e-04
Epoch 14/30
534/534 - 0s - loss: 2.8056e-04
Epoch 15/30
534/534 - 0s - loss: 2.5969e-04
Epoch 16/30
534/534 - 0s - loss: 2.4182e-04
Epoch 17/30
534/534 - 0s - loss: 2.2617e-04
Epoch 18/30
534/534 - 0s - loss: 2.1247e-04
Epoch 19/30
534/534 - 0s - loss: 2.0025e-04
Epoch 20/30
534/534 - 0s - loss: 1.8932e-04
Epoch 21/30
534/534 - 0s - loss: 1.7958e-04
Epoch 22/30
534/534 - 0s - loss: 1.7067e-04
Epoch 23/30
534/534 - 0s - loss: 1.6255e-04
Epoch 24/30
534/534 - 0s - loss: 1.5524e-04
Epoch 25/30
534/534 - 0s - loss: 1.4852e-04
Epoch 26/30
534/534 - 1s - loss: 1.4248e-04
Epoch 27/30
534/534 - 0s - loss: 1.3672e-04
Epoch 28/30
534/534 - 0s - loss: 1.3150e-04
Epoch 29/30
534/534 - 0s - loss: 1.2651e-04
Epoch 30/30
534/534 - 0s - loss: 1.2199e-04

```

Out[11]: <tensorflow.python.keras.callbacks.History at 0x1e0dfdf9040>

```

In [12]: 1 ypred = model.predict(X_test)
          2 print("y1 MSE: ", mean_squared_error(t_test.iloc[:, 0], ypred[:,0]))
          3 #print("y2 MSE: ", mean_squared_error(t_test.iloc[:, 1], ypred[:,1]))
          4 #print("y3 MSE: ", mean_squared_error(t_test.iloc[:, 2], ypred[:,2]))
          5

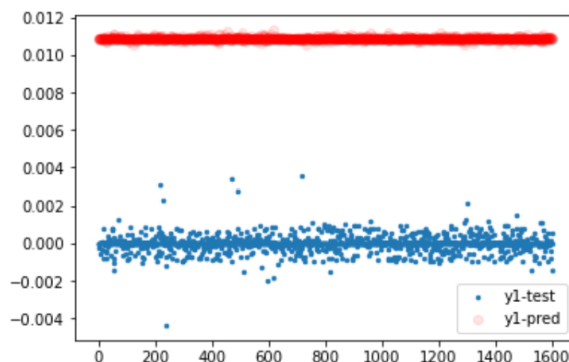
```

y1 MSE: 0.00011865859691458849

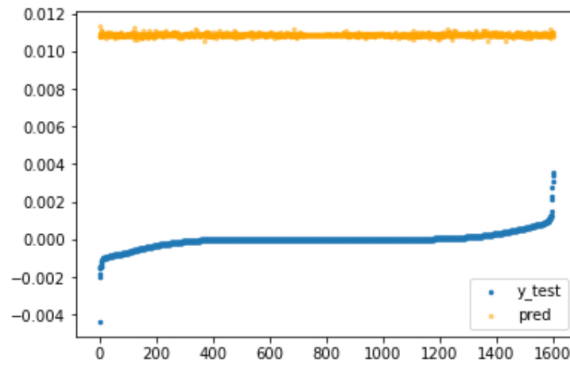
```

In [13]: 1 x_ax = range(len(X_test))
          2
          3 plt.scatter(x_ax, t_test.iloc[:, 0], s=6, label="y1-test")
          4 plt.scatter(x_ax, ypred[:,0], label="y1-pred",c="red",alpha = 0.1)
          5
          6 plt.legend()
          7 plt.show()

```



```
In [14]: 1 x_ax = range(len(X_test))
2
3 y_test_index = np.argsort(t_test.iloc[:, 0], axis=0).to_numpy()
4
5 f = plt.figure()
6 plt.scatter(x_ax, t_test.iloc[y_test_index], s=6, label="y_test")
7 plt.scatter(x_ax, ypred[y_test_index], s=6, label="pred", c="orange", alpha=0.5)
8 #plt.ylim(t_test.iloc[y_test_index[0]].to_numpy()[0])
9 plt.legend()
10 plt.show()
11
12 f.savefig("foo.pdf", bbox_inches='tight')
```



```
In [ ]: 1
```