Error	100 hidden layer		200 hidden layer	
	7d	14d	7d	14d
MSE	51478.591	32516.112	52408.673	30828.780
RMSE	226.889	180.322	228.929	175.581
MAPE	3.525	3.266	3.665	2.902
MPE	-0.336	3.266	-1.099	- 0.641

14day: จะเห็นได้ว่า error hidden layer 200 units น่านำไปใช้ที่สุด

7day: จะเห็นได้ว่า error hidden layer 100 units น่านำไปใช้ที่สุด

Code: Set day

```
    m = 7  # number of lags, pred at 8th day
    per = (1736 - m)/1826  # percentage of data used for training
3   print(per)
    size = int(len(data) * per)
    d_train, d_test = data[0:size], data[size:len(data)]
    mean_train = np.mean(d_train)
    sd_train = np.std(d_train)
    d_train = (d_train-mean_train)/sd_train
9   d_test = (d_test-mean_train)/sd_train

0.9468784227820373
```

Code: Set Hidden Layer

```python
import scipy.stats as stats
input_size = X_train.shape[1]
hidden_size = 200  # no. of hidden neurons
mu, sigma = 0, 1
w_lo = -1
w_hi = 1
b_lo = -1
b_hi = 1
# initialising input weights and biases randomly drawn from a
input_weights = stats.truncnorm.rvs((w_lo - mu) / sigma, (w_hi
biases = stats.truncnorm.rvs((b_lo - mu) / sigma, (b_hi - mu)


def relu(x):  # hidden layer activation function
    return np.maximum(x, 0, x)
```

[1290] ✓ 0.0s

Code: Print MSE

```python
    correct = 0
    total = X_test.shape[0]
    y_test = (y_test*sd_train) + mean_train
    prediction = (prediction*sd_train) + mean_train
    # evaluate forecasts
    rmse = math.sqrt(mean_squared_error(y_test, prediction))
    mse = mean_squared_error(y_test, prediction)
    print('Test MSE: %.3f' % mse)
    print('Test RMSE: %.3f' % rmse)
10  mape_sum = 0
    for i, j in zip(y_test, prediction):
      mape_sum = mape_sum + (abs((i-j)/i))
    mape = (mape_sum/total)*100
    mpe_sum = 0
    for i, j in zip(y_test, prediction):
      mpe_sum = mpe_sum + ((i-j)/i)
    mpe = (mpe_sum/total)*100
    print('Test MAPE: %.3f' % mape)
19  print('Test MPE: %.3f' % mpe)
```

✓ 0.0s

Test MSE: 52408.673
Test RMSE: 228.929
Test MAPE: 3.665
Test MPE: -1.099