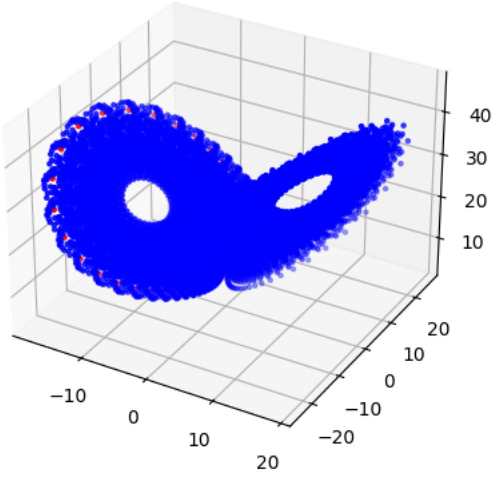
```
import numpy as np
import pandas as pd
from sklearn.linear_model import Ridge
import matplotlib.pyplot as plt
```

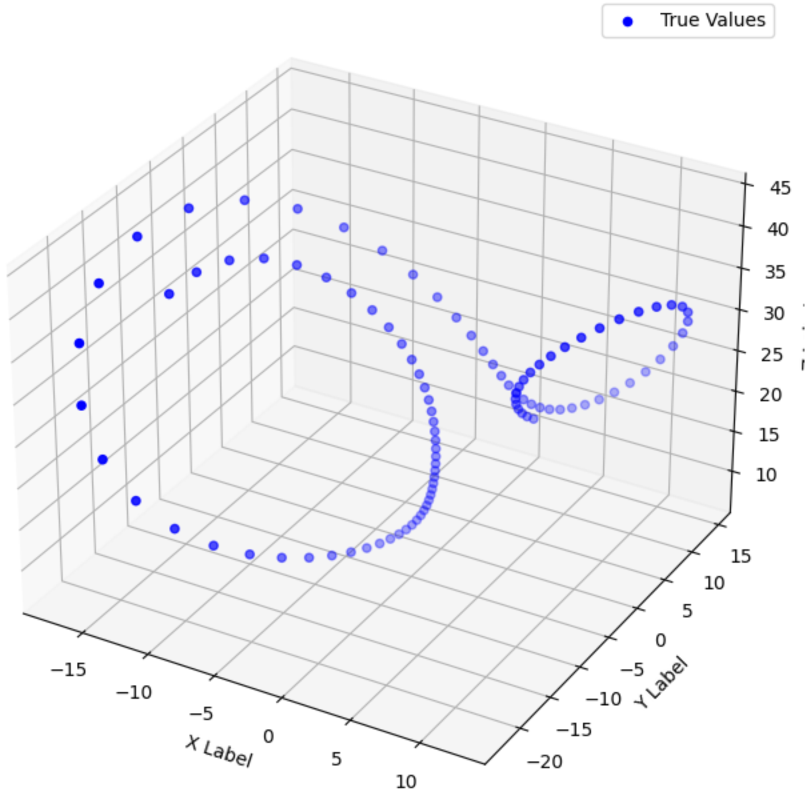
```
train = pd.read_csv('train.csv',header=None)
test = pd.read_csv('testset.csv',header=None)

x = train.iloc[0,:].values
y = train.iloc[1,:].values
z = train.iloc[2,:].values

x1 = test.iloc[0,:].values
y1 = test.iloc[1,:].values
z1 = test.iloc[2,:].values

fig=plt.figure()
ax = fig.add_subplot(111,projection = '3d')


ax.scatter(x1,y1,z1, c = 'r', marker = '.')
ax.scatter(x,y,z,c = 'b', marker = '.')
plt.show()
```



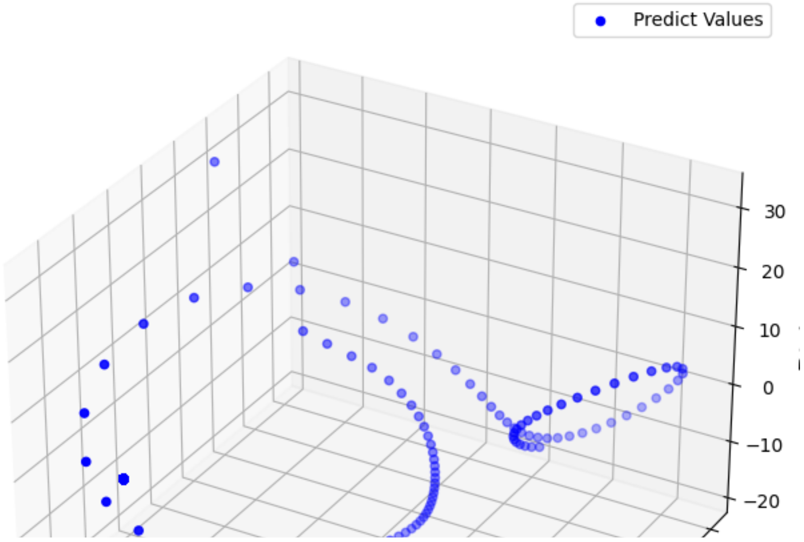
```
dt = 0.02
nInput = 3
nReservior =500
k = 0.01
nStep = 500

#initialize weights
wI = np.random.randn(nReservior,nInput) * np.sqrt(0.02)
wR = np.random.randn(nReservior,nReservior) * np.sqrt(2/500)
np.fill_diagonal(wR, 0)

reserviors = np.zeros((train.shape[1],nReservior))
reservior = np.zeros(nReservior)


for t in range(1,train.shape[1]):
    input = train.iloc[:,t-1].values
    tmp_Reservior = np.tanh(np.dot(wI,input)+np.dot(wR,reservior))
    reservior = tmp_Reservior
    reserviors[t] = tmp_Reservior


target = train.iloc[:,1:].values.T
clf = Ridge(alpha=k)
clf.fit(reserviors[:-1], target)

wO = clf.coef_
wO
```

```
array([[ 0.28005685,  0.03690323,  0.31457783, ..., -0.05947474,
        -0.08395412,  0.01950987],
       [ 1.46855091, -0.11498987,  0.23300178, ...,  0.22095465,
        -0.45103521,  0.88446405],
       [ 0.05641527,  0.37542359,  1.17707542, ..., -0.27481676,
         0.27865345,  0.75291738]])
```

```
test_reserviors = np.zeros((test.shape[1]+nStep, nReservior))
output_predictions = np.zeros((test.shape[1] + nStep, nInput))
for t in range(1, test.shape[1]):
    input = test.iloc[:, t-1].values
    tmp_Reservior = np.tanh(np.dot(wI, input) + np.dot(wR, test_reserviors[t - 1]))
    test_reserviors[t] = tmp_Reservior
    output_predictions[t] = np.dot(wO, tmp_Reservior)



testt = test.values.T
x, y, z = testt[:, 0], testt[:, 1], testt[:, 2]
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, c='b', marker='o', label='True Values', s=20)
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
ax.legend()


plt.show()
```



```
x, y, z = output_predictions[:, 0], output_predictions[:, 1], output_predictions[:, 2]
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, c='b', marker='o', label='Predict Values', s=20)
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
ax.legend()

plt.show()
```



```
input = test.iloc[:, -1].values

for t in range(test.shape[1], test.shape[1] + nStep):
    tmp_Reservior = np.tanh(np.dot(wI, input) + np.dot(wR, test_reserviors[t - 1]))
    test_reserviors[t] = tmp_Reservior
    output_predictions[t] = np.dot(wO, tmp_Reservior)
    input = output_predictions[t]

output = output_predictions[-500:]
```

Created in Deepnote