Assignment 5

# Tau = 4

Chart, line chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

# Tau = 6

Chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

# Tau = 8

Chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

# Tau = 10

Chart, line chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

# Tau = 12

Chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

# Tau = 14

Chart, line chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

# Tau = 16

Chart, line chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Histogram

Description automatically generated

# Tau = 18

Chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

# Tau = 20

Chart, line chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

# Chart, histogram Description automatically generated

# MSE Values

Text

Description automatically generatedAs you can observe in the figure, **a tau value of 10 gives the smallest MSE.**

# Code

# DL12C.py CS5173/6073 cheng 2023  
# autoregression on hospitalization  
# using RNN  
# using MSELoss and Adam  
# with random sample of training data  
# Usage: python DL12C.py  
  
import numpy as np  
import random  
import torch  
import matplotlib.pyplot as plt  
  
x = torch.tensor(np.genfromtxt('hamiltonCountyHospitalization.txt'), dtype=torch.float32) / 500.0  
T = len(x)  
num\_train = T // 2  
input\_size = 1  
hidden\_size = 10  
output\_size = 1  
batch\_size = 32  
  
for tau in range(4, 22, 2):  
  
 features = [x[i: T-tau+i] for i in range(tau)]  
 X = torch.stack(features, 1)  
 y = x[tau:].reshape((-1, 1))  
 Xtrain = X[:num\_train]  
 ytrain = y[:num\_train]  
  
 class RNN2(torch.nn.Module):  
 def \_\_init\_\_(self):  
 super(RNN2, self).\_\_init\_\_()  
 self.rnn = torch.nn.RNN(input\_size, hidden\_size, 1)  
 self.linear = torch.nn.Linear(hidden\_size, output\_size)  
  
 def forward(self, x):  
 X2 = torch.reshape(x.T, (tau, len(x), input\_size))  
 h0 = torch.randn(1, len(x), hidden\_size)  
 \_, hn = self.rnn(X2, h0)  
 return self.linear(hn[0])  
  
 model = RNN2()  
 y2 = model(X)  
 plt.plot(y)  
 plt.plot(y2.detach().numpy())  
 plt.show() # Figure 1  
  
 loss\_fun = torch.nn.MSELoss()  
 optimizer = torch.optim.Adam(model.parameters())  
 rounds = 1000  
 losses = np.zeros(rounds)  
 indices = list(range(num\_train))  
 for i in range(rounds):  
 random.shuffle(indices)  
 batch\_indices = torch.tensor(indices[:batch\_size])  
 y\_pred = model(X[batch\_indices])  
 loss = loss\_fun(y\_pred, y[batch\_indices])  
 losses[i] = loss.item()  
 optimizer.zero\_grad()  
 loss.backward()  
 optimizer.step()  
  
 y2 = model(X)  
 plt.plot(y) # Figure 2  
 plt.plot(y2.detach().numpy())  
 plt.show()  
  
 print(f'Tau = {tau}; MSE = {losses[rounds - 1]}')  
 plt.plot(losses)  
 plt.show() # Figure 3