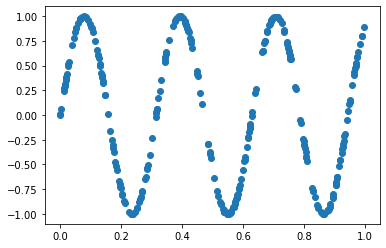
Atanas Delevski

ECE 559 Homework #3 Report

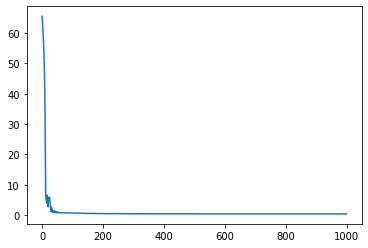
10/25/2020

**Question 1:**  


*Part 3:*

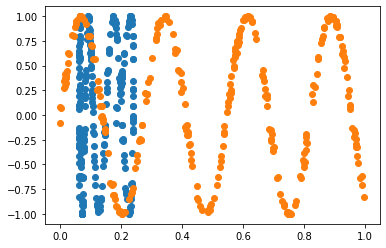
*Part 4:*

I couldn’t get my algorithm to avoid local minima. I also think I did a poor job with the matrix multiplication because it seems to train very slowly at only around 100 epochs per second.

MSE vs Epochs (100)

Part 5:

Like I mentioned in part 4, my algorithm kept converging on local minima and therefore never got a good fit. Here is an example of my algorithm after 1000 epochs.



Code:

from matplotlib import pyplot as plt

import numpy as np

np.random.seed(1)

n = 300 ## number of inputs

N = 24 ## number of neurons

eta = 0.01 ## learning rate

num\_epochs = 1000 ## number of epochs

x = np.random.uniform(0,1,n)

y = np.random.uniform(0,1,n)

v = np.random.uniform(-.1,.1,n)

d = [np.sin(20\*x[i]+3\*x[i]+v[i]) for i in range(n)]

darr = np.array(d)

v2\_list = np.ones(n)

v1\_list = np.ones((n, N))

w = np.random.normal(-1,1,3\*N+1)

plt.scatter(x, d, marker='o')

def tanh(x): ## tanh function

return np.tanh(x)

def tanh\_prime(x):

return 1 - np.square(np.tanh(x)) ## tanh derivative function

def forward(xarr, warr):

mse = 0

y2 = np.zeros(n)

for i in range(n):

v1 = warr[0:2\*N].reshape(N,2)@np.array([[1], [xarr[i]]]) ## forward pass of algorithm. Each line follow the matrix formulation of the algo.

v1\_list[i] = v1.reshape(N)

y1 = tanh\_prime(v1)

temp = np.ones((N+1, 1))

temp[1:] = y1

v2 = (warr[2\*N:3\*N+1].reshape(1,N+1)@temp).reshape(1)

v2\_list[i] = v2

y2[i] = v2

mse += np.square(darr[i]-y2[i])

mse = mse/n

return np.array(v1\_list), np.array(v2\_list), temp, y2, mse

def backprop(darr, temp, y2, v1\_list, v2\_list, w):

for i in range(n):

d2 = (darr[i]-y2[i])\*tanh\_prime(v2\_list[i]) ## the backwards pass of the algorithm

d1 = (w[2\*N+1:3\*N+1].reshape(N, 1)\*d2)\*(tanh\_prime(v1\_list[i])).reshape(N, 1)

w[0:2\*N] = (w[0:2\*N].reshape(N, 2) + eta\*(d1@np.array([[1, x[1]]]))).reshape(2\*N)

w[2\*N:3\*N+1] = w[2\*N:3\*N+1] + eta\*((d2\*temp).reshape(N+1))

return w

v1\_list, v2\_list, temp, y2, mse = forward(x, w) ## initial pass

mse\_list = []

epochs = 0

for epoch in range(num\_epochs): ## training phase, prints ever 100 epochs, has dynamic learning rate

v1\_list, v2\_list, temp, y2, mse\_check = forward(x, w)

w = backprop(darr, temp, y2, v1\_list, v2\_list, w)

mse\_list.append(mse\_check)

if mse\_check > mse:

eta = .9\*eta

mse = mse\_check

if epoch % 100 == 0:

print(f"Epoch: {epoch}, MSE: {mse}")

eps = np.arange(0, num\_epochs)

plt.plot(eps, mse\_list)

plt.scatter(y2, darr)

plt.scatter(x, darr)