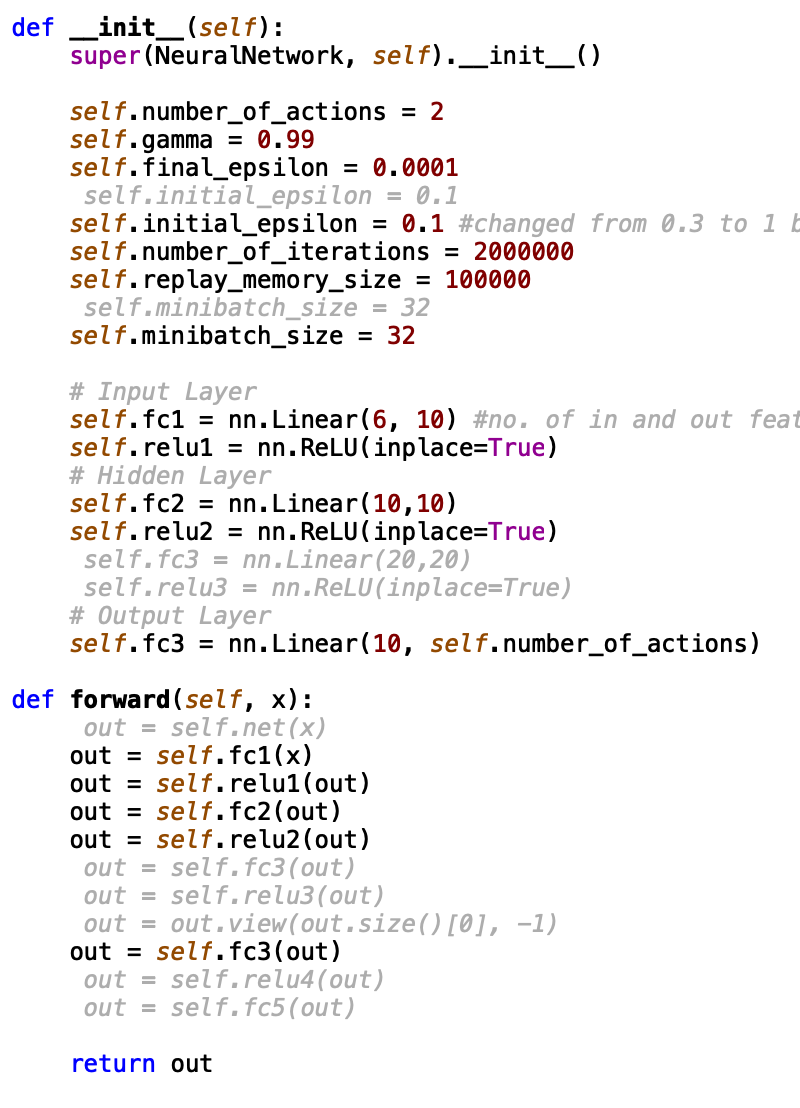
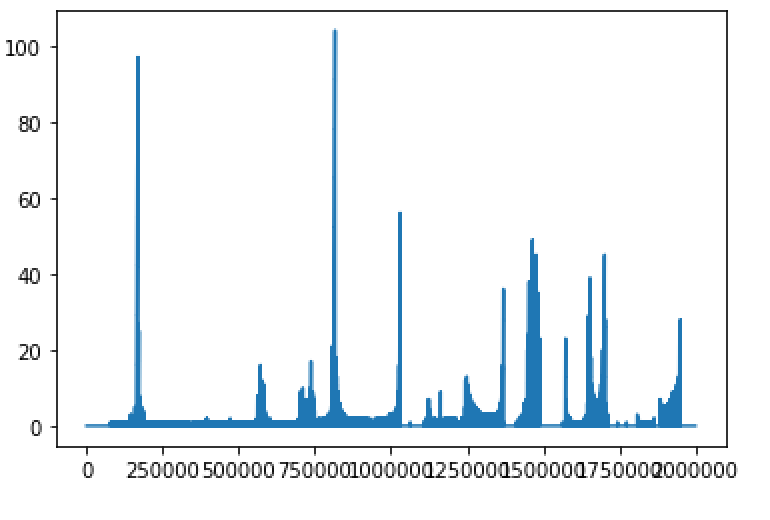
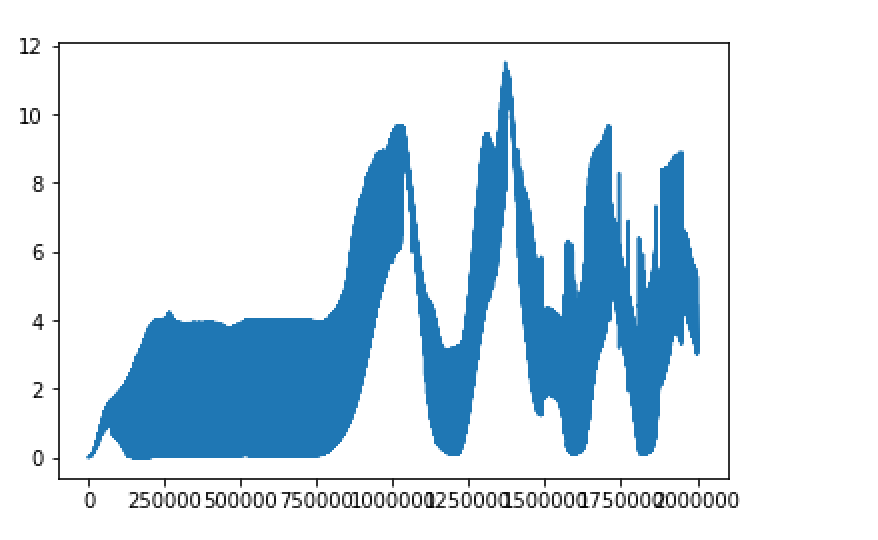
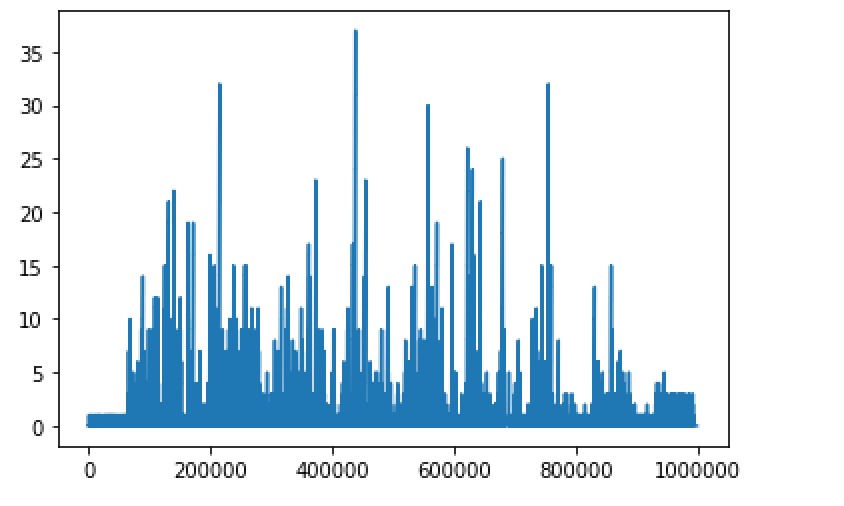
neural network

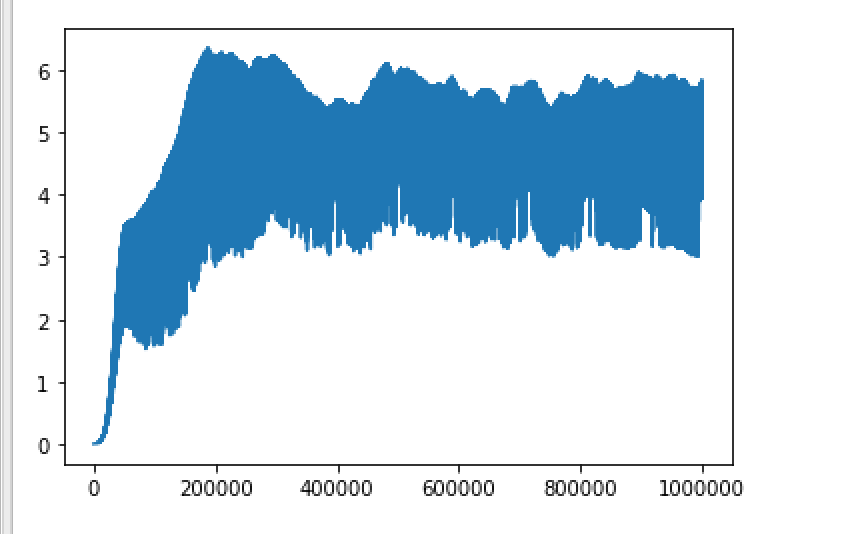












class NeuralNetwork(nn.Module):

def \_\_init\_\_(self):

super(NeuralNetwork, self).\_\_init\_\_()

self.number\_of\_actions = 2

self.gamma = 0.99

self.final\_epsilon = 0.0001

# self.initial\_epsilon = 0.1

self.initial\_epsilon = 0.1 #changed from 0.3 to 1 by Lalita

self.number\_of\_iterations = 1500000

self.replay\_memory\_size = 300000

# self.minibatch\_size = 32

self.minibatch\_size = 32

# Input Layer

self.fc1 = nn.Linear(6, 10) #no. of in and out features

self.relu1 = nn.ReLU(inplace=True)

# Hidden Layer

self.fc2 = nn.Linear(10,10)

self.relu2 = nn.ReLU(inplace=True)

self.fc3 = nn.Linear(10,10)

self.relu3 = nn.ReLU(inplace=True)

# Output Layer

self.fc4 = nn.Linear(10, self.number\_of\_actions)

def forward(self, x):

# out = self.net(x)

out = self.fc1(x)

out = self.relu1(out)

out = self.fc2(out)

out = self.relu2(out)

out = self.fc3(out)

out = self.relu3(out)

# out = out.view(out.size()[0], -1)

out = self.fc4(out)

# out = self.relu4(out)

# out = self.fc5(out)

return out

