# Лабораторная работа № 8

## Динамические сети

Цель работы: исследование свойств некоторых динамических нейронных сетей, алгоритмов обучения, а также применение сетей в задачах аппроксимации функций и распознавания динамических образов.

Студент Мариничев И.А.

Группа М8О-408Б-19

Вариант 5

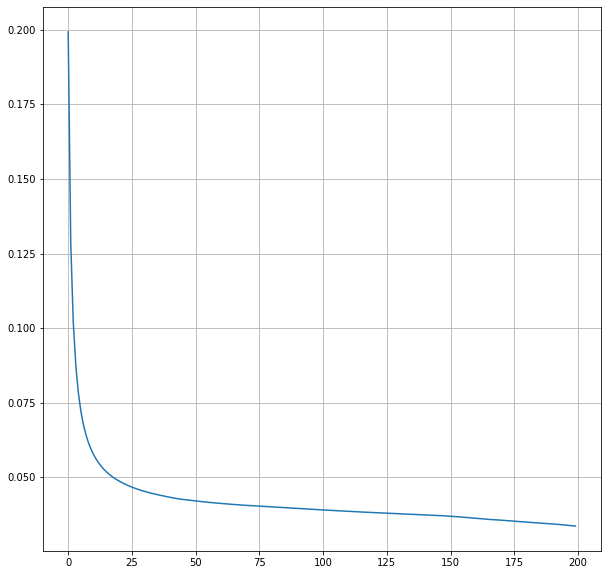
Зададим набор данных для обучения:

N, w = 600, 5  
t = np.linspace(0, 5, N)  
  
# управляющий сигнал  
uk = np.sin(t\*\*2 - 2\*t + 5)   
  
# истинный сигнал  
yk = [0]  
for i in range(N-1):  
 yk += [yk[-1] / (1 + yk[-1]\*\*2) + pow(uk[i], 3)]

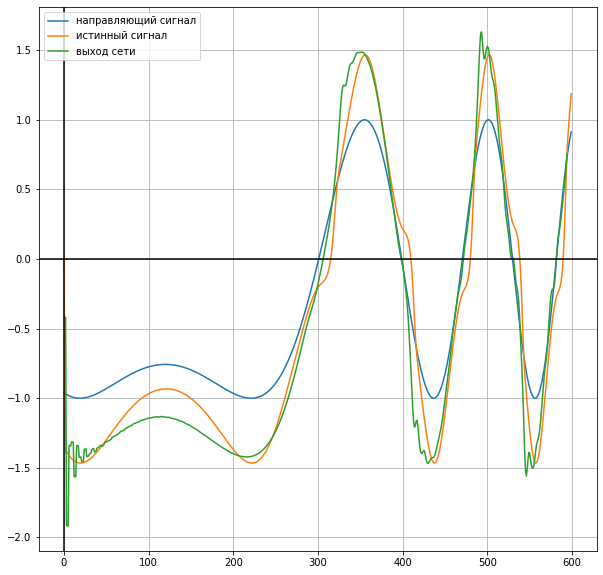
Построим нелинейную авторегрессионную сеть с внешними входами (Non-linearAutoRegressive network with eXogeneous inputs, NARX):

# Nonlinear AutoRegressive eXogenous model  
class NARX(nn.Module):  
 def \_\_init\_\_(self, in\_features, hidden\_features, out\_features, dealay1, delay2):  
 super(NARX, self).\_\_init\_\_()  
  
 self.in\_features = in\_features  
 self.hidden\_features = hidden\_features  
 self.out\_features = out\_features  
  
 self.line1 = TDL(in\_features, dealay1)  
 self.line2 = TDL(out\_features, delay2)  
  
 self.fc1 = nn.Linear(in\_features, hidden\_features)  
 self.fc2 = nn.Linear(hidden\_features, out\_features)  
 self.fc3 = nn.Linear(out\_features, hidden\_features, bias=False)  
  
 def clear(self):  
 self.line1.clear()  
 self.line2.clear()  
  
 def forward(self, inputs):  
 out1 = torch.tanh(self.fc1(self.line1()) + self.fc3(self.line2()))  
 out2 = self.fc2(out1)  
  
 self.line1.push(inputs.clone().detach())  
 self.line2.push(out2.clone().detach())  
  
 return out2

Определим функцию обучения. А также определим функцию предсказания. Обучим модель. Посмотрим на график функции потерь, вычисляющей MSE между исходными и полученными данными:



И посмотрим на наши результаты:



Выводы: в ходе данной работы была построена и обучена нелинейная авторегрессионная сеть с внешними входами, которая была использована в задаче аппроксимации траектории динамической системы, кроме того при помощи нее был выполнен многошаговый прогноз значений системы. После 200 эпох обучения были получены корректные результаты.