

МГТУ им. Н. Э. Баумана  
Факультет ФН «Фундаментальные Науки»  
Кафедра ФН-12 «Математическое моделирование»

Отчет по домашнему заданию №5  
По дисциплине "Численные методы решения задач теории управления"

Студент: Петров М.И.  
Преподаватель: Тверская Е. С.  
Группа: ФН12-61Б

Москва 2024

# Задание 1

1. Постановка задачи. Обучить построенную нейронную сеть на наборе данных, соответствующему 15 варианту

## 2. Задание

Построить многослойную нейронную сеть (рекомендации: входной слой, выходной слой и скрытый слой). Разработать алгоритм обучения данной нейронной сети, основанный на методе стохастического градиентного спуска. При разработке допускается использование библиотек numpy, matplotlib и pytorch.

Рассматриваемые выборка с целевой функцией:

```
X = torch.rand(1500, 2)
Y = (((X[:, 1]+0.1)**2)-(2*X[:, 0]-0.9)**2 < 0.4).float()
Y = Y.view(-1, 1)
```

Создание трёхслойной нейронной сети:

```
class ThreeLayersNet(nn.Module):
    def __init__(self, nX, nH, nY, nZ):
        super(ThreeLayersNet, self).__init__()

        self.fc1 = nn.Linear(nX, nH)
        self.fc2 = nn.Linear(nH, nY)
        self.fc3 = nn.Linear(nY, nZ)

    def forward(self, x):
        x = self.fc1(x)
        x = nn.Sigmoid() (x)
        x = self.fc2(x)
        x = nn.Sigmoid() (x)
        x = self.fc3(x)
        x = nn.Sigmoid() (x)
        return x
```

Создание алгоритма обучения на основе метода стохастического градиентного спуска:

```
def fit(model, X,Y, batch_size=1500, train=True):
    model.train(train)
    sumL, sumA, numB = 0, 0, int( len(X)/batch_size )

    for i in range(0, numB*batch_size, batch_size):
        xb = X[i: i+batch_size]
        yb = Y[i: i+batch_size]

        y = model(xb)
        L = loss(y, yb)

        if train:
            optimizer.zero_grad()
            L.backward()
            optimizer.step()

        sumL += L.item()
```

```
sumA += (y.round() == yb).float().mean()  
return sumL/numB, sumA/numB
```

Итоговый результат обучения нейронной сети:

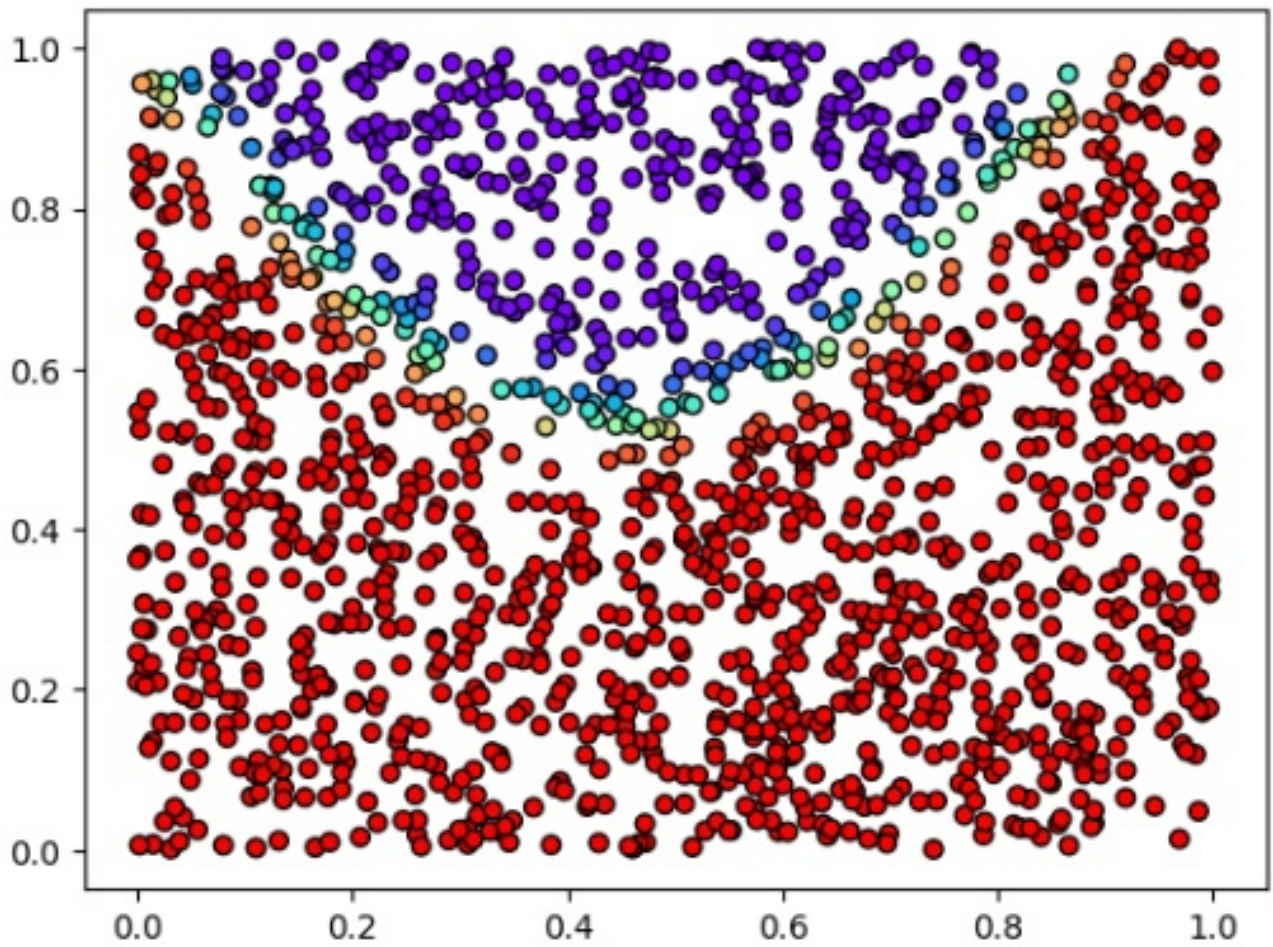


Рис. 1: Полученный результат  
Значения качества обучения на соответствующих эпохах обучения:

```
before:      loss: 0.8402 accuracy: 0.2547
epoch:      0 loss: 0.8402 accuracy: 0.2547
epoch:    100 loss: 0.5626 accuracy: 0.7453
epoch:    200 loss: 0.3576 accuracy: 0.8327
epoch:    300 loss: 0.2659 accuracy: 0.8667
epoch:    400 loss: 0.2631 accuracy: 0.8667
epoch:    500 loss: 0.2539 accuracy: 0.8720
epoch:    600 loss: 0.2097 accuracy: 0.9013
epoch:    700 loss: 0.1056 accuracy: 0.9667
epoch:    800 loss: 0.0627 accuracy: 0.9913
epoch:    900 loss: 0.0478 accuracy: 0.9940
epoch:    999 loss: 0.0400 accuracy: 0.9947
```

Рис. 2: Функция потерь и оценка качества