Лабораторная работа № 2

Линейная нейронная сеть. Правило обучения Уидроу-Хоффа

Цель работы: исследование свойств линейной нейронной сети и алгоритмов ее обучения, применение сети в задачах аппроксимации и фильтрации

Студент	Попов И.П.
Группа	М8О-406Б-20
Вариант	17

```
import numpy as np
import torch
import torch.nn as nn
import matplotlib.pyplot as plt
```

Аппроксимация функции

Создается временная последовательность X с использованием функции x(t), а затем формируются последовательности длиной D из временного ряда и соответствующие им последующие дискреты для обучения модели с заданным числом задержек D.

```
D = 5

def x(t):
    return np.sin(np.sin(t) * t**2 + 3*t - 10)

t = np.arange(2.5, 5, 0.01)
X = x(t).tolist()


sequences = [X[i:i+D] for i in range(0, len(X) - D)]
upcoming_points = [X[i] for i in range(D, len(X))]

assert len(sequences) == len(upcoming_points)
```

Определяется функция sync_shuffle, которая перемешивает два массива a и b с использованием одной и той же случайной перестановки, обеспечивая синхронное перемешивание элементов обоих массивов.

```
def sync_shuffle(a, b):
    assert len(a) == len(b)
    p = np.random.permutation(len(a))
    return a[p], b[p]
```

Обучающие данные sequences и upcoming_points синхронно перемешиваются с использованием функции sync_shuffle.

```
x_train, y_train = sync_shuffle(np.array(sequences),
np.array(upcoming_points))

x_train = torch.FloatTensor(x_train)
y_train = torch.FloatTensor(y_train)
```

Создание модели AdaptiveLinearNetwork, повторяет Perceptron из ЛР 1

```
class AdaptiveLinearNetwork(nn.Module):
    def __init__(self, in_features: int, out_features: int, bias: bool
= True):
        super().__init__()
        self.weights = nn.Parameter(torch.randn(in_features,
out_features))
        self.bias = bias
        if bias:
            self.bias_term = nn.Parameter(torch.randn(out_features))

    def forward(self, x):
        x = x @ self.weights
        if self.bias:
            x += self.bias_term
        return x
```

Создается экземпляр модели с числом входных признаков D и одним выходным признаком. Для обучения используется среднеквадратичная ошибка в качестве функции потерь, и оптимизатор стохастического градиентного спуска (SGD) с заданной скоростью обучения 0.05.

```
adaptiveLinearNetwork = AdaptiveLinearNetwork(D, 1)
loss_function = nn.MSELoss()
optimizer = torch.optim.SGD(adaptiveLinearNetwork.parameters(),
lr=0.05)
```

Определена функция fit, которая выполняет обучение модели. В каждой эпохе производится прямой проход, вычисляется и обновляется функция потерь, итерации отображаются с использованием tqdm. Возвращается список значений функции потерь на каждой эпохе.

```
def fit(model, x_train, y_train, criterion, optimizer, epochs):
    losses = []
    log_template = "\nEpoch {ep:03d} train_loss: {t_loss:0.4f}"
    for epoch in range(epochs):
        optimizer.zero_grad()
        outp = model(x_train)
```

```
        loss = criterion(outp.view(-1), y_train)

        loss.backward()
        losses.append(loss.detach().flatten()[0])
        optimizer.step()

        print(log_template.format(ep=epoch+1, t_loss=loss))
    return losses
```

Определена функция predict, которая использует обученную модель для выполнения предсказания на входных данных x_test. Модель переводится в режим оценки (eval), и с предсказаниями возвращается результат.

```
def predict(model, x_test):
    with torch.no_grad():
        model.eval()
        outp = model(x_test)
    return outp
```

Модель обучается на обучающих данных

```
losses = fit(adaptiveLinearNetwork, x_train, y_train, loss_function,
optimizer, 50)
```

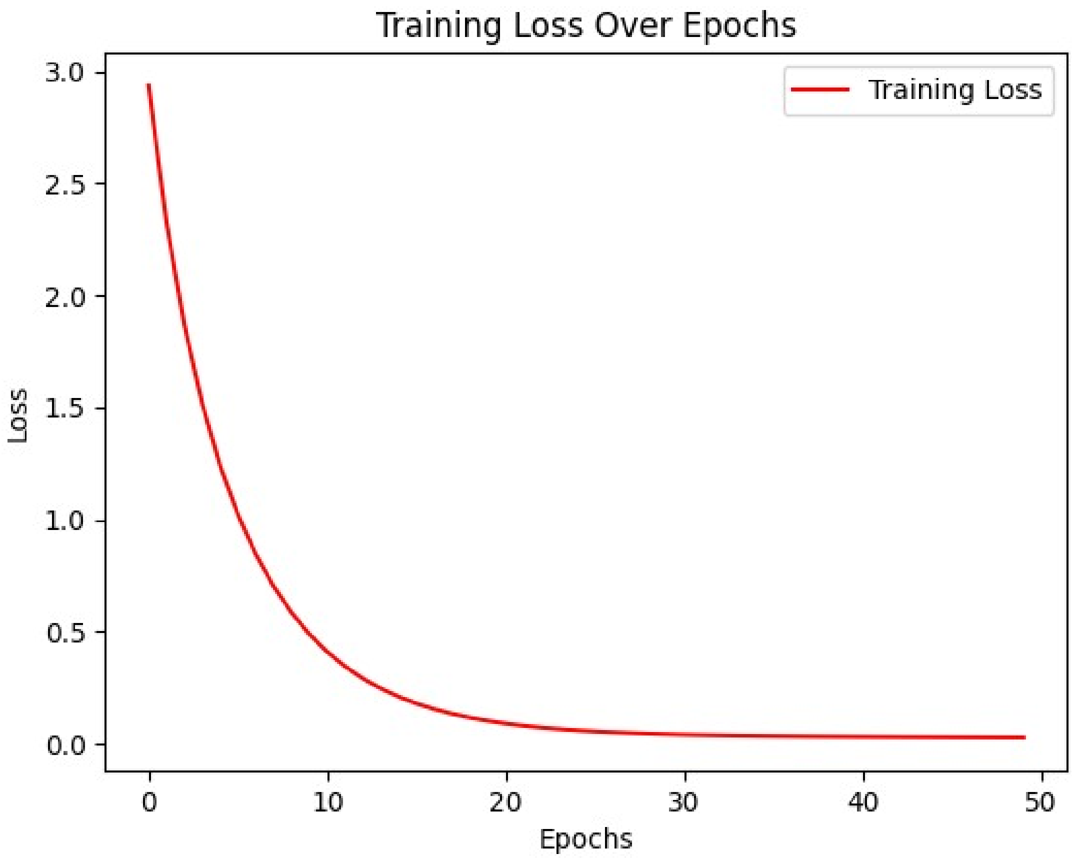
```
Epoch 001 train_loss: 2.9362

Epoch 002 train_loss: 2.3250

Epoch 003 train_loss: 1.8679

Epoch 004 train_loss: 1.5167

Epoch 005 train_loss: 1.2413

Epoch 006 train_loss: 1.0219

Epoch 007 train_loss: 0.8452

Epoch 008 train_loss: 0.7016

Epoch 009 train_loss: 0.5843

Epoch 010 train_loss: 0.4881

Epoch 011 train_loss: 0.4090

Epoch 012 train_loss: 0.3438

Epoch 013 train_loss: 0.2900
```

```
Epoch 014 train_loss: 0.2456
Epoch 015 train_loss: 0.2088
Epoch 016 train_loss: 0.1784
Epoch 017 train_loss: 0.1533
Epoch 018 train_loss: 0.1324
Epoch 019 train_loss: 0.1152
Epoch 020 train_loss: 0.1009
Epoch 021 train_loss: 0.0890
Epoch 022 train_loss: 0.0791
Epoch 023 train_loss: 0.0709
Epoch 024 train_loss: 0.0641
Epoch 025 train_loss: 0.0584
Epoch 026 train_loss: 0.0537
Epoch 027 train_loss: 0.0497
Epoch 028 train_loss: 0.0464
Epoch 029 train_loss: 0.0436
Epoch 030 train_loss: 0.0412
Epoch 031 train_loss: 0.0392
Epoch 032 train_loss: 0.0376
Epoch 033 train_loss: 0.0361
Epoch 034 train_loss: 0.0349
Epoch 035 train_loss: 0.0338
Epoch 036 train_loss: 0.0329
Epoch 037 train_loss: 0.0321
Epoch 038 train_loss: 0.0314
Epoch 039 train_loss: 0.0308
```

```
Epoch 040 train_loss: 0.0302

Epoch 041 train_loss: 0.0297

Epoch 042 train_loss: 0.0293

Epoch 043 train_loss: 0.0289

Epoch 044 train_loss: 0.0285

Epoch 045 train_loss: 0.0281

Epoch 046 train_loss: 0.0278

Epoch 047 train_loss: 0.0275

Epoch 048 train_loss: 0.0272

Epoch 049 train_loss: 0.0269

Epoch 050 train_loss: 0.0266
```

График функции потерь (ошибка - MSE)

```
plt.plot(losses, color='red', label='Training Loss')
plt.legend()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training Loss Over Epochs')
plt.show()
```



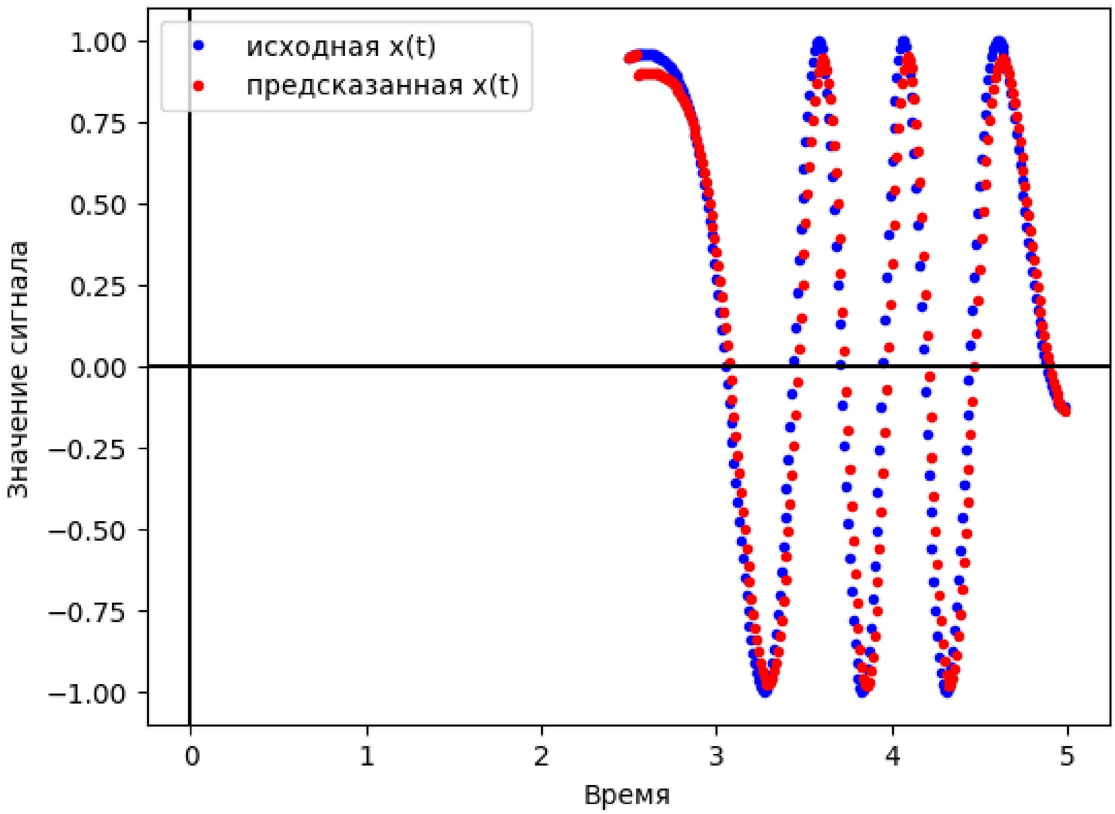
Производится прогнозирование значений моделью для последующих точек временной последовательности. Предсказанные значения добавляются к списку X_PRED, а ошибки предсказания записываются в список errors.

```
X_PRED = X[:D]
errors = []

for i in range(0, len(upcoming_points)):
    x_test = torch.FloatTensor(np.array(sequences[i]))
    upcoming_point_pred = predict(adaptiveLinearNetwork, x_test)
    X_PRED += upcoming_point_pred.numpy().tolist()
    errors += (upcoming_point_pred -
upcoming_points[i]).numpy().tolist()
```

На графике синим цветом обозначены исходные значения временной последовательности x(t), а красным - предсказанные значения моделью AdaptiveLinearNetwork.

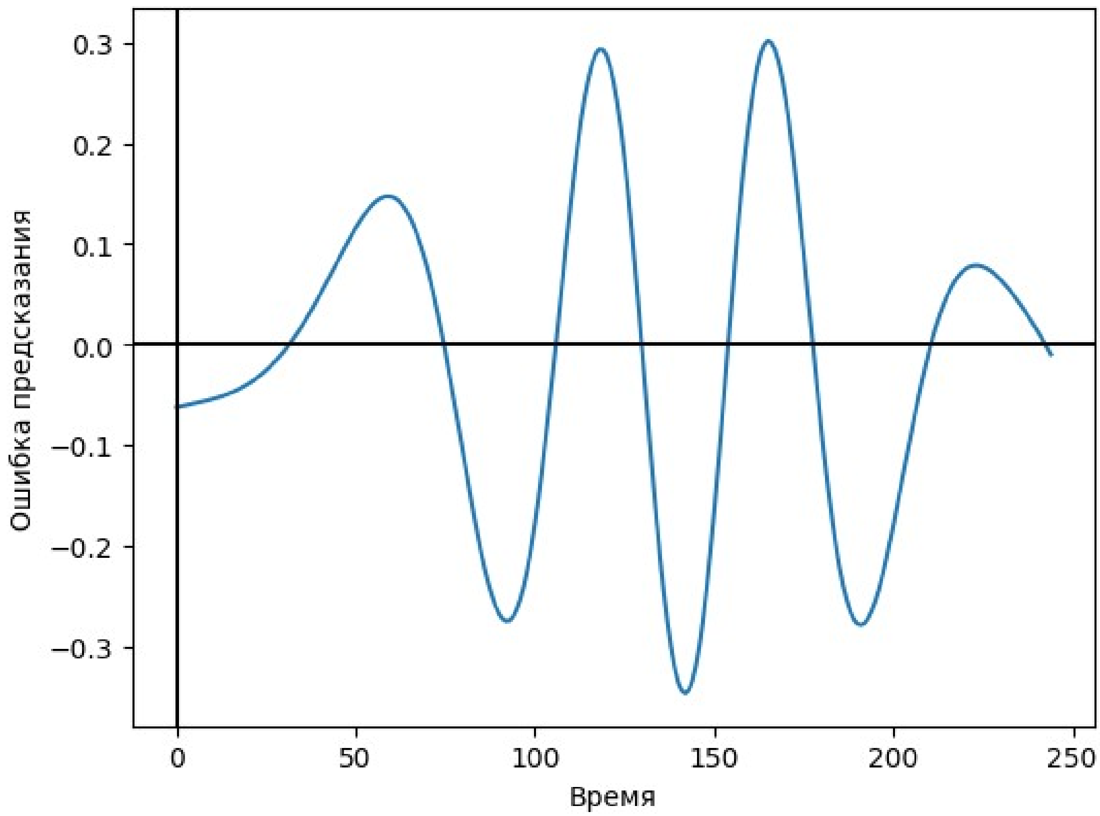
```
plt.plot(t, X, '.', color="blue", label='исходная x(t)')
plt.plot(t, X_PRED, '.', color="red", label='предсказанная x(t)')

plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')
```

```
plt.xlabel('Время')
plt.ylabel('Значение сигнала')
plt.legend()
plt.show()
```



На этом графике отображены ошибки предсказания для каждого момента времени.

```
plt.plot(errors)

plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')

plt.xlabel('Время')
plt.ylabel('Ошибка предсказания')
plt.show()
```



Подавление помех

Создаются истинный сигнал X и зашумленный сигнал Y с использованием функций true_signal и noized_signal. Затем формируются последовательности из зашумленного сигнала Y длиной D и соответствующие им последующие дискреты истинного сигнала X для задачи подавления помех.

```
D = 4


def true_signal(t):
    return np.cos(np.cos(t) * t**2 + 5*t)


def noized_signal(t):
    return (1 / 5) * np.cos(np.cos(t) * t**2 + 5 * t + 4)


t = np.arange(0, 3.5, 0.01)
X = true_signal(t).tolist()
Y = noized_signal(t).tolist()


noized_sequences = [Y[i:i+D] for i in range(0, len(Y) - D)]
upcoming_points_true = [X[i] for i in range(D, len(X))]
```

```
assert len(noized_sequences) == len(upcoming_points_true)
```

Обучающие данные noized_sequences и upcoming_points_true синхронно перемешиваются с использованием функции sync_shuffle

```
x_train, y_train = sync_shuffle(np.array(noized_sequences),
np.array(upcoming_points_true))

x_train = torch.FloatTensor(x_train)
y_train = torch.FloatTensor(y_train)

adaptiveLinearNetwork2 = AdaptiveLinearNetwork(D, 1)
loss_function = nn.MSELoss()
optimizer = torch.optim.SGD(adaptiveLinearNetwork2.parameters(),
lr=0.05)

losses2 = fit(adaptiveLinearNetwork2, x_train, y_train, loss_function,
optimizer, 600)
```

```
Epoch 001 train_loss: 6.1653

Epoch 002 train_loss: 5.0690

Epoch 003 train_loss: 4.1804

Epoch 004 train_loss: 3.4600

Epoch 005 train_loss: 2.8759

Epoch 006 train_loss: 2.4022

Epoch 007 train_loss: 2.0178

Epoch 008 train_loss: 1.7058

Epoch 009 train_loss: 1.4523

Epoch 010 train_loss: 1.2464

Epoch 011 train_loss: 1.0789

Epoch 012 train_loss: 0.9426

Epoch 013 train_loss: 0.8315

Epoch 014 train_loss: 0.7409

Epoch 015 train_loss: 0.6668
```

```
Epoch 016 train_loss: 0.6061

Epoch 017 train_loss: 0.5564

Epoch 018 train_loss: 0.5154

Epoch 019 train_loss: 0.4816

Epoch 020 train_loss: 0.4537

Epoch 021 train_loss: 0.4304

Epoch 022 train_loss: 0.4109

Epoch 023 train_loss: 0.3946

Epoch 024 train_loss: 0.3808

Epoch 025 train_loss: 0.3690

Epoch 026 train_loss: 0.3589

Epoch 027 train_loss: 0.3502

Epoch 028 train_loss: 0.3426

Epoch 029 train_loss: 0.3359

Epoch 030 train_loss: 0.3299

Epoch 031 train_loss: 0.3246

Epoch 032 train_loss: 0.3198

Epoch 033 train_loss: 0.3154

Epoch 034 train_loss: 0.3113

Epoch 035 train_loss: 0.3075

Epoch 036 train_loss: 0.3040

Epoch 037 train_loss: 0.3006

Epoch 038 train_loss: 0.2975

Epoch 039 train_loss: 0.2945

Epoch 040 train_loss: 0.2916

Epoch 041 train_loss: 0.2888
```

```
Epoch 042 train_loss: 0.2862

Epoch 043 train_loss: 0.2836

Epoch 044 train_loss: 0.2811

Epoch 045 train_loss: 0.2786

Epoch 046 train_loss: 0.2762

Epoch 047 train_loss: 0.2739

Epoch 048 train_loss: 0.2716

Epoch 049 train_loss: 0.2694

Epoch 050 train_loss: 0.2673

Epoch 051 train_loss: 0.2651

Epoch 052 train_loss: 0.2630

Epoch 053 train_loss: 0.2610

Epoch 054 train_loss: 0.2590

Epoch 055 train_loss: 0.2570

Epoch 056 train_loss: 0.2551

Epoch 057 train_loss: 0.2532

Epoch 058 train_loss: 0.2513

Epoch 059 train_loss: 0.2495

Epoch 060 train_loss: 0.2477

Epoch 061 train_loss: 0.2459

Epoch 062 train_loss: 0.2442

Epoch 063 train_loss: 0.2424

Epoch 064 train_loss: 0.2408

Epoch 065 train_loss: 0.2391

Epoch 066 train_loss: 0.2375

Epoch 067 train_loss: 0.2359
```

```
Epoch 068 train_loss: 0.2343
Epoch 069 train_loss: 0.2328
Epoch 070 train_loss: 0.2313
Epoch 071 train_loss: 0.2298
Epoch 072 train_loss: 0.2283
Epoch 073 train_loss: 0.2269
Epoch 074 train_loss: 0.2254
Epoch 075 train_loss: 0.2240
Epoch 076 train_loss: 0.2227
Epoch 077 train_loss: 0.2213
Epoch 078 train_loss: 0.2200
Epoch 079 train_loss: 0.2187
Epoch 080 train_loss: 0.2174
Epoch 081 train_loss: 0.2162
Epoch 082 train_loss: 0.2149
Epoch 083 train_loss: 0.2137
Epoch 084 train_loss: 0.2125
Epoch 085 train_loss: 0.2113
Epoch 086 train_loss: 0.2102
Epoch 087 train_loss: 0.2090
Epoch 088 train_loss: 0.2079
Epoch 089 train_loss: 0.2068
Epoch 090 train_loss: 0.2057
Epoch 091 train_loss: 0.2047
Epoch 092 train_loss: 0.2036
Epoch 093 train_loss: 0.2026
```

```
Epoch 094 train_loss: 0.2016

Epoch 095 train_loss: 0.2006

Epoch 096 train_loss: 0.1996

Epoch 097 train_loss: 0.1986

Epoch 098 train_loss: 0.1977

Epoch 099 train_loss: 0.1968

Epoch 100 train_loss: 0.1959

Epoch 101 train_loss: 0.1950

Epoch 102 train_loss: 0.1941

Epoch 103 train_loss: 0.1932

Epoch 104 train_loss: 0.1924

Epoch 105 train_loss: 0.1915

Epoch 106 train_loss: 0.1907

Epoch 107 train_loss: 0.1899

Epoch 108 train_loss: 0.1891

Epoch 109 train_loss: 0.1883

Epoch 110 train_loss: 0.1875

Epoch 111 train_loss: 0.1868

Epoch 112 train_loss: 0.1860

Epoch 113 train_loss: 0.1853

Epoch 114 train_loss: 0.1846

Epoch 115 train_loss: 0.1839

Epoch 116 train_loss: 0.1832

Epoch 117 train_loss: 0.1825

Epoch 118 train_loss: 0.1818

Epoch 119 train_loss: 0.1811
```

```
Epoch 120 train_loss: 0.1805

Epoch 121 train_loss: 0.1799

Epoch 122 train_loss: 0.1792

Epoch 123 train_loss: 0.1786

Epoch 124 train_loss: 0.1780

Epoch 125 train_loss: 0.1774

Epoch 126 train_loss: 0.1768

Epoch 127 train_loss: 0.1762

Epoch 128 train_loss: 0.1757

Epoch 129 train_loss: 0.1751

Epoch 130 train_loss: 0.1746

Epoch 131 train_loss: 0.1740

Epoch 132 train_loss: 0.1735

Epoch 133 train_loss: 0.1730

Epoch 134 train_loss: 0.1725

Epoch 135 train_loss: 0.1719

Epoch 136 train_loss: 0.1715

Epoch 137 train_loss: 0.1710

Epoch 138 train_loss: 0.1705

Epoch 139 train_loss: 0.1700

Epoch 140 train_loss: 0.1695

Epoch 141 train_loss: 0.1691

Epoch 142 train_loss: 0.1686

Epoch 143 train_loss: 0.1682

Epoch 144 train_loss: 0.1678

Epoch 145 train_loss: 0.1673
```

```
Epoch 146 train_loss: 0.1669

Epoch 147 train_loss: 0.1665

Epoch 148 train_loss: 0.1661

Epoch 149 train_loss: 0.1657

Epoch 150 train_loss: 0.1653

Epoch 151 train_loss: 0.1649

Epoch 152 train_loss: 0.1645

Epoch 153 train_loss: 0.1642

Epoch 154 train_loss: 0.1638

Epoch 155 train_loss: 0.1635

Epoch 156 train_loss: 0.1631

Epoch 157 train_loss: 0.1627

Epoch 158 train_loss: 0.1624

Epoch 159 train_loss: 0.1621

Epoch 160 train_loss: 0.1617

Epoch 161 train_loss: 0.1614

Epoch 162 train_loss: 0.1611

Epoch 163 train_loss: 0.1608

Epoch 164 train_loss: 0.1605

Epoch 165 train_loss: 0.1602

Epoch 166 train_loss: 0.1599

Epoch 167 train_loss: 0.1596

Epoch 168 train_loss: 0.1593

Epoch 169 train_loss: 0.1590

Epoch 170 train_loss: 0.1587

Epoch 171 train_loss: 0.1584
```

```
Epoch 172 train_loss: 0.1582

Epoch 173 train_loss: 0.1579

Epoch 174 train_loss: 0.1576

Epoch 175 train_loss: 0.1574

Epoch 176 train_loss: 0.1571

Epoch 177 train_loss: 0.1569

Epoch 178 train_loss: 0.1566

Epoch 179 train_loss: 0.1564

Epoch 180 train_loss: 0.1562

Epoch 181 train_loss: 0.1559

Epoch 182 train_loss: 0.1557

Epoch 183 train_loss: 0.1555

Epoch 184 train_loss: 0.1553

Epoch 185 train_loss: 0.1550

Epoch 186 train_loss: 0.1548

Epoch 187 train_loss: 0.1546

Epoch 188 train_loss: 0.1544

Epoch 189 train_loss: 0.1542

Epoch 190 train_loss: 0.1540

Epoch 191 train_loss: 0.1538

Epoch 192 train_loss: 0.1536

Epoch 193 train_loss: 0.1534

Epoch 194 train_loss: 0.1532

Epoch 195 train_loss: 0.1531

Epoch 196 train_loss: 0.1529

Epoch 197 train_loss: 0.1527
```

```
Epoch 198 train_loss: 0.1525

Epoch 199 train_loss: 0.1524

Epoch 200 train_loss: 0.1522

Epoch 201 train_loss: 0.1520

Epoch 202 train_loss: 0.1519

Epoch 203 train_loss: 0.1517

Epoch 204 train_loss: 0.1515

Epoch 205 train_loss: 0.1514

Epoch 206 train_loss: 0.1512

Epoch 207 train_loss: 0.1511

Epoch 208 train_loss: 0.1509

Epoch 209 train_loss: 0.1508

Epoch 210 train_loss: 0.1507

Epoch 211 train_loss: 0.1505

Epoch 212 train_loss: 0.1504

Epoch 213 train_loss: 0.1502

Epoch 214 train_loss: 0.1501

Epoch 215 train_loss: 0.1500

Epoch 216 train_loss: 0.1498

Epoch 217 train_loss: 0.1497

Epoch 218 train_loss: 0.1496

Epoch 219 train_loss: 0.1495

Epoch 220 train_loss: 0.1494

Epoch 221 train_loss: 0.1492

Epoch 222 train_loss: 0.1491

Epoch 223 train_loss: 0.1490
```

```
Epoch 224 train_loss: 0.1489

Epoch 225 train_loss: 0.1488

Epoch 226 train_loss: 0.1487

Epoch 227 train_loss: 0.1486

Epoch 228 train_loss: 0.1485

Epoch 229 train_loss: 0.1484

Epoch 230 train_loss: 0.1483

Epoch 231 train_loss: 0.1482

Epoch 232 train_loss: 0.1481

Epoch 233 train_loss: 0.1480

Epoch 234 train_loss: 0.1479

Epoch 235 train_loss: 0.1478

Epoch 236 train_loss: 0.1477

Epoch 237 train_loss: 0.1476

Epoch 238 train_loss: 0.1475

Epoch 239 train_loss: 0.1474

Epoch 240 train_loss: 0.1473

Epoch 241 train_loss: 0.1472

Epoch 242 train_loss: 0.1472

Epoch 243 train_loss: 0.1471

Epoch 244 train_loss: 0.1470

Epoch 245 train_loss: 0.1469

Epoch 246 train_loss: 0.1468

Epoch 247 train_loss: 0.1468

Epoch 248 train_loss: 0.1467

Epoch 249 train_loss: 0.1466
```

```
Epoch 250 train_loss: 0.1465

Epoch 251 train_loss: 0.1465

Epoch 252 train_loss: 0.1464

Epoch 253 train_loss: 0.1463

Epoch 254 train_loss: 0.1463

Epoch 255 train_loss: 0.1462

Epoch 256 train_loss: 0.1461

Epoch 257 train_loss: 0.1461

Epoch 258 train_loss: 0.1460

Epoch 259 train_loss: 0.1459

Epoch 260 train_loss: 0.1459

Epoch 261 train_loss: 0.1458

Epoch 262 train_loss: 0.1458

Epoch 263 train_loss: 0.1457

Epoch 264 train_loss: 0.1456

Epoch 265 train_loss: 0.1456

Epoch 266 train_loss: 0.1455

Epoch 267 train_loss: 0.1455

Epoch 268 train_loss: 0.1454

Epoch 269 train_loss: 0.1454

Epoch 270 train_loss: 0.1453

Epoch 271 train_loss: 0.1453

Epoch 272 train_loss: 0.1452

Epoch 273 train_loss: 0.1452

Epoch 274 train_loss: 0.1451

Epoch 275 train_loss: 0.1451
```

```
Epoch 276 train_loss: 0.1450

Epoch 277 train_loss: 0.1450

Epoch 278 train_loss: 0.1449

Epoch 279 train_loss: 0.1449

Epoch 280 train_loss: 0.1448

Epoch 281 train_loss: 0.1448

Epoch 282 train_loss: 0.1447

Epoch 283 train_loss: 0.1447

Epoch 284 train_loss: 0.1447

Epoch 285 train_loss: 0.1446

Epoch 286 train_loss: 0.1446

Epoch 287 train_loss: 0.1445

Epoch 288 train_loss: 0.1445

Epoch 289 train_loss: 0.1445

Epoch 290 train_loss: 0.1444

Epoch 291 train_loss: 0.1444

Epoch 292 train_loss: 0.1444

Epoch 293 train_loss: 0.1443

Epoch 294 train_loss: 0.1443

Epoch 295 train_loss: 0.1443

Epoch 296 train_loss: 0.1442

Epoch 297 train_loss: 0.1442

Epoch 298 train_loss: 0.1442

Epoch 299 train_loss: 0.1441

Epoch 300 train_loss: 0.1441

Epoch 301 train_loss: 0.1441
```

```
Epoch 302 train_loss: 0.1440

Epoch 303 train_loss: 0.1440

Epoch 304 train_loss: 0.1440

Epoch 305 train_loss: 0.1439

Epoch 306 train_loss: 0.1439

Epoch 307 train_loss: 0.1439

Epoch 308 train_loss: 0.1438

Epoch 309 train_loss: 0.1438

Epoch 310 train_loss: 0.1438

Epoch 311 train_loss: 0.1438

Epoch 312 train_loss: 0.1437

Epoch 313 train_loss: 0.1437

Epoch 314 train_loss: 0.1437

Epoch 315 train_loss: 0.1437

Epoch 316 train_loss: 0.1436

Epoch 317 train_loss: 0.1436

Epoch 318 train_loss: 0.1436

Epoch 319 train_loss: 0.1436

Epoch 320 train_loss: 0.1435

Epoch 321 train_loss: 0.1435

Epoch 322 train_loss: 0.1435

Epoch 323 train_loss: 0.1435

Epoch 324 train_loss: 0.1435

Epoch 325 train_loss: 0.1434

Epoch 326 train_loss: 0.1434

Epoch 327 train_loss: 0.1434
```

```
Epoch 328 train_loss: 0.1434

Epoch 329 train_loss: 0.1434

Epoch 330 train_loss: 0.1433

Epoch 331 train_loss: 0.1433

Epoch 332 train_loss: 0.1433

Epoch 333 train_loss: 0.1433

Epoch 334 train_loss: 0.1433

Epoch 335 train_loss: 0.1432

Epoch 336 train_loss: 0.1432

Epoch 337 train_loss: 0.1432

Epoch 338 train_loss: 0.1432

Epoch 339 train_loss: 0.1432

Epoch 340 train_loss: 0.1432

Epoch 341 train_loss: 0.1431

Epoch 342 train_loss: 0.1431

Epoch 343 train_loss: 0.1431

Epoch 344 train_loss: 0.1431

Epoch 345 train_loss: 0.1431

Epoch 346 train_loss: 0.1431

Epoch 347 train_loss: 0.1430

Epoch 348 train_loss: 0.1430

Epoch 349 train_loss: 0.1430

Epoch 350 train_loss: 0.1430

Epoch 351 train_loss: 0.1430

Epoch 352 train_loss: 0.1430

Epoch 353 train_loss: 0.1430
```

```
Epoch 354 train_loss: 0.1429

Epoch 355 train_loss: 0.1429

Epoch 356 train_loss: 0.1429

Epoch 357 train_loss: 0.1429

Epoch 358 train_loss: 0.1429

Epoch 359 train_loss: 0.1429

Epoch 360 train_loss: 0.1429

Epoch 361 train_loss: 0.1429

Epoch 362 train_loss: 0.1428

Epoch 363 train_loss: 0.1428

Epoch 364 train_loss: 0.1428

Epoch 365 train_loss: 0.1428

Epoch 366 train_loss: 0.1428

Epoch 367 train_loss: 0.1428

Epoch 368 train_loss: 0.1428

Epoch 369 train_loss: 0.1428

Epoch 370 train_loss: 0.1428

Epoch 371 train_loss: 0.1427

Epoch 372 train_loss: 0.1427

Epoch 373 train_loss: 0.1427

Epoch 374 train_loss: 0.1427

Epoch 375 train_loss: 0.1427

Epoch 376 train_loss: 0.1427

Epoch 377 train_loss: 0.1427

Epoch 378 train_loss: 0.1427

Epoch 379 train_loss: 0.1427
```

```
Epoch 380 train_loss: 0.1427

Epoch 381 train_loss: 0.1427

Epoch 382 train_loss: 0.1426

Epoch 383 train_loss: 0.1426

Epoch 384 train_loss: 0.1426

Epoch 385 train_loss: 0.1426

Epoch 386 train_loss: 0.1426

Epoch 387 train_loss: 0.1426

Epoch 388 train_loss: 0.1426

Epoch 389 train_loss: 0.1426

Epoch 390 train_loss: 0.1426

Epoch 391 train_loss: 0.1426

Epoch 392 train_loss: 0.1426

Epoch 393 train_loss: 0.1426

Epoch 394 train_loss: 0.1426

Epoch 395 train_loss: 0.1425

Epoch 396 train_loss: 0.1425

Epoch 397 train_loss: 0.1425

Epoch 398 train_loss: 0.1425

Epoch 399 train_loss: 0.1425

Epoch 400 train_loss: 0.1425

Epoch 401 train_loss: 0.1425

Epoch 402 train_loss: 0.1425

Epoch 403 train_loss: 0.1425

Epoch 404 train_loss: 0.1425

Epoch 405 train_loss: 0.1425
```

```
Epoch 406 train_loss: 0.1425

Epoch 407 train_loss: 0.1425

Epoch 408 train_loss: 0.1425

Epoch 409 train_loss: 0.1425

Epoch 410 train_loss: 0.1424

Epoch 411 train_loss: 0.1424

Epoch 412 train_loss: 0.1424

Epoch 413 train_loss: 0.1424

Epoch 414 train_loss: 0.1424

Epoch 415 train_loss: 0.1424

Epoch 416 train_loss: 0.1424

Epoch 417 train_loss: 0.1424

Epoch 418 train_loss: 0.1424

Epoch 419 train_loss: 0.1424

Epoch 420 train_loss: 0.1424

Epoch 421 train_loss: 0.1424

Epoch 422 train_loss: 0.1424

Epoch 423 train_loss: 0.1424

Epoch 424 train_loss: 0.1424

Epoch 425 train_loss: 0.1424

Epoch 426 train_loss: 0.1424

Epoch 427 train_loss: 0.1424

Epoch 428 train_loss: 0.1424

Epoch 429 train_loss: 0.1423

Epoch 430 train_loss: 0.1423

Epoch 431 train_loss: 0.1423
```

```
Epoch 432 train_loss: 0.1423

Epoch 433 train_loss: 0.1423

Epoch 434 train_loss: 0.1423

Epoch 435 train_loss: 0.1423

Epoch 436 train_loss: 0.1423

Epoch 437 train_loss: 0.1423

Epoch 438 train_loss: 0.1423

Epoch 439 train_loss: 0.1423

Epoch 440 train_loss: 0.1423

Epoch 441 train_loss: 0.1423

Epoch 442 train_loss: 0.1423

Epoch 443 train_loss: 0.1423

Epoch 444 train_loss: 0.1423

Epoch 445 train_loss: 0.1423

Epoch 446 train_loss: 0.1423

Epoch 447 train_loss: 0.1423

Epoch 448 train_loss: 0.1423

Epoch 449 train_loss: 0.1423

Epoch 450 train_loss: 0.1423

Epoch 451 train_loss: 0.1423

Epoch 452 train_loss: 0.1423

Epoch 453 train_loss: 0.1423

Epoch 454 train_loss: 0.1423

Epoch 455 train_loss: 0.1422

Epoch 456 train_loss: 0.1422

Epoch 457 train_loss: 0.1422
```

```
Epoch 458 train_loss: 0.1422

Epoch 459 train_loss: 0.1422

Epoch 460 train_loss: 0.1422

Epoch 461 train_loss: 0.1422

Epoch 462 train_loss: 0.1422

Epoch 463 train_loss: 0.1422

Epoch 464 train_loss: 0.1422

Epoch 465 train_loss: 0.1422

Epoch 466 train_loss: 0.1422

Epoch 467 train_loss: 0.1422

Epoch 468 train_loss: 0.1422

Epoch 469 train_loss: 0.1422

Epoch 470 train_loss: 0.1422

Epoch 471 train_loss: 0.1422

Epoch 472 train_loss: 0.1422

Epoch 473 train_loss: 0.1422

Epoch 474 train_loss: 0.1422

Epoch 475 train_loss: 0.1422

Epoch 476 train_loss: 0.1422

Epoch 477 train_loss: 0.1422

Epoch 478 train_loss: 0.1422

Epoch 479 train_loss: 0.1422

Epoch 480 train_loss: 0.1422

Epoch 481 train_loss: 0.1422

Epoch 482 train_loss: 0.1422

Epoch 483 train_loss: 0.1422
```

```
Epoch 484 train_loss: 0.1422

Epoch 485 train_loss: 0.1422

Epoch 486 train_loss: 0.1422

Epoch 487 train_loss: 0.1422

Epoch 488 train_loss: 0.1422

Epoch 489 train_loss: 0.1422

Epoch 490 train_loss: 0.1422

Epoch 491 train_loss: 0.1421

Epoch 492 train_loss: 0.1421

Epoch 493 train_loss: 0.1421

Epoch 494 train_loss: 0.1421

Epoch 495 train_loss: 0.1421

Epoch 496 train_loss: 0.1421

Epoch 497 train_loss: 0.1421

Epoch 498 train_loss: 0.1421

Epoch 499 train_loss: 0.1421

Epoch 500 train_loss: 0.1421

Epoch 501 train_loss: 0.1421

Epoch 502 train_loss: 0.1421

Epoch 503 train_loss: 0.1421

Epoch 504 train_loss: 0.1421

Epoch 505 train_loss: 0.1421

Epoch 506 train_loss: 0.1421

Epoch 507 train_loss: 0.1421

Epoch 508 train_loss: 0.1421

Epoch 509 train_loss: 0.1421
```

```
Epoch 510 train_loss: 0.1421

Epoch 511 train_loss: 0.1421

Epoch 512 train_loss: 0.1421

Epoch 513 train_loss: 0.1421

Epoch 514 train_loss: 0.1421

Epoch 515 train_loss: 0.1421

Epoch 516 train_loss: 0.1421

Epoch 517 train_loss: 0.1421

Epoch 518 train_loss: 0.1421

Epoch 519 train_loss: 0.1421

Epoch 520 train_loss: 0.1421

Epoch 521 train_loss: 0.1421

Epoch 522 train_loss: 0.1421

Epoch 523 train_loss: 0.1421

Epoch 524 train_loss: 0.1421

Epoch 525 train_loss: 0.1421

Epoch 526 train_loss: 0.1421

Epoch 527 train_loss: 0.1421

Epoch 528 train_loss: 0.1421

Epoch 529 train_loss: 0.1421

Epoch 530 train_loss: 0.1421

Epoch 531 train_loss: 0.1421

Epoch 532 train_loss: 0.1421

Epoch 533 train_loss: 0.1421

Epoch 534 train_loss: 0.1421

Epoch 535 train_loss: 0.1421
```

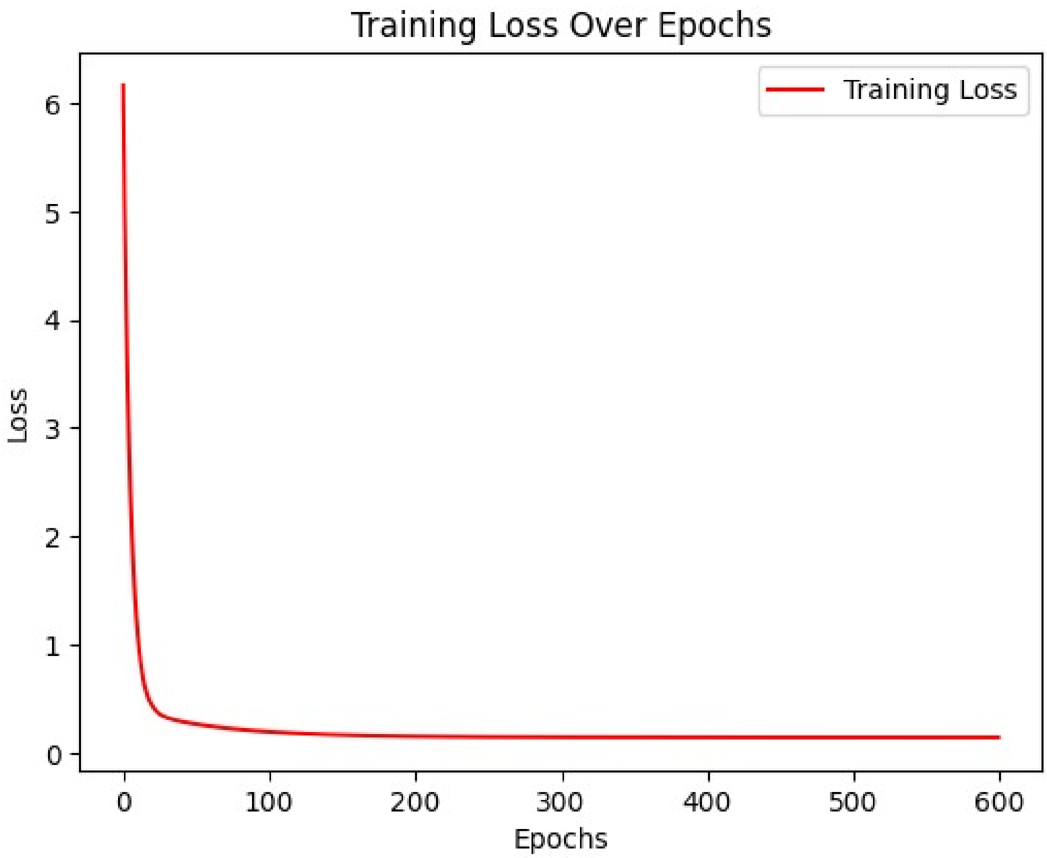
```
Epoch 536 train_loss: 0.1421

Epoch 537 train_loss: 0.1421

Epoch 538 train_loss: 0.1421

Epoch 539 train_loss: 0.1421

Epoch 540 train_loss: 0.1421

Epoch 541 train_loss: 0.1421

Epoch 542 train_loss: 0.1421

Epoch 543 train_loss: 0.1421

Epoch 544 train_loss: 0.1421

Epoch 545 train_loss: 0.1420

Epoch 546 train_loss: 0.1420

Epoch 547 train_loss: 0.1420

Epoch 548 train_loss: 0.1420

Epoch 549 train_loss: 0.1420

Epoch 550 train_loss: 0.1420

Epoch 551 train_loss: 0.1420

Epoch 552 train_loss: 0.1420

Epoch 553 train_loss: 0.1420

Epoch 554 train_loss: 0.1420

Epoch 555 train_loss: 0.1420

Epoch 556 train_loss: 0.1420

Epoch 557 train_loss: 0.1420

Epoch 558 train_loss: 0.1420

Epoch 559 train_loss: 0.1420

Epoch 560 train_loss: 0.1420

Epoch 561 train_loss: 0.1420
```

```
Epoch 562 train_loss: 0.1420

Epoch 563 train_loss: 0.1420

Epoch 564 train_loss: 0.1420

Epoch 565 train_loss: 0.1420

Epoch 566 train_loss: 0.1420

Epoch 567 train_loss: 0.1420

Epoch 568 train_loss: 0.1420

Epoch 569 train_loss: 0.1420

Epoch 570 train_loss: 0.1420

Epoch 571 train_loss: 0.1420

Epoch 572 train_loss: 0.1420

Epoch 573 train_loss: 0.1420

Epoch 574 train_loss: 0.1420

Epoch 575 train_loss: 0.1420

Epoch 576 train_loss: 0.1420

Epoch 577 train_loss: 0.1420

Epoch 578 train_loss: 0.1420

Epoch 579 train_loss: 0.1420

Epoch 580 train_loss: 0.1420

Epoch 581 train_loss: 0.1420

Epoch 582 train_loss: 0.1420

Epoch 583 train_loss: 0.1420

Epoch 584 train_loss: 0.1420

Epoch 585 train_loss: 0.1420

Epoch 586 train_loss: 0.1420

Epoch 587 train_loss: 0.1420
```

```
Epoch 588 train_loss: 0.1420

Epoch 589 train_loss: 0.1420

Epoch 590 train_loss: 0.1420

Epoch 591 train_loss: 0.1420

Epoch 592 train_loss: 0.1420

Epoch 593 train_loss: 0.1420

Epoch 594 train_loss: 0.1420

Epoch 595 train_loss: 0.1420

Epoch 596 train_loss: 0.1420

Epoch 597 train_loss: 0.1420

Epoch 598 train_loss: 0.1420

Epoch 599 train_loss: 0.1420

Epoch 600 train_loss: 0.1420
```

График функции потерь (ошибка - MSE)

```
plt.plot(losses2, color='red', label='Training Loss')
plt.legend()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training Loss Over Epochs')
plt.show()
```



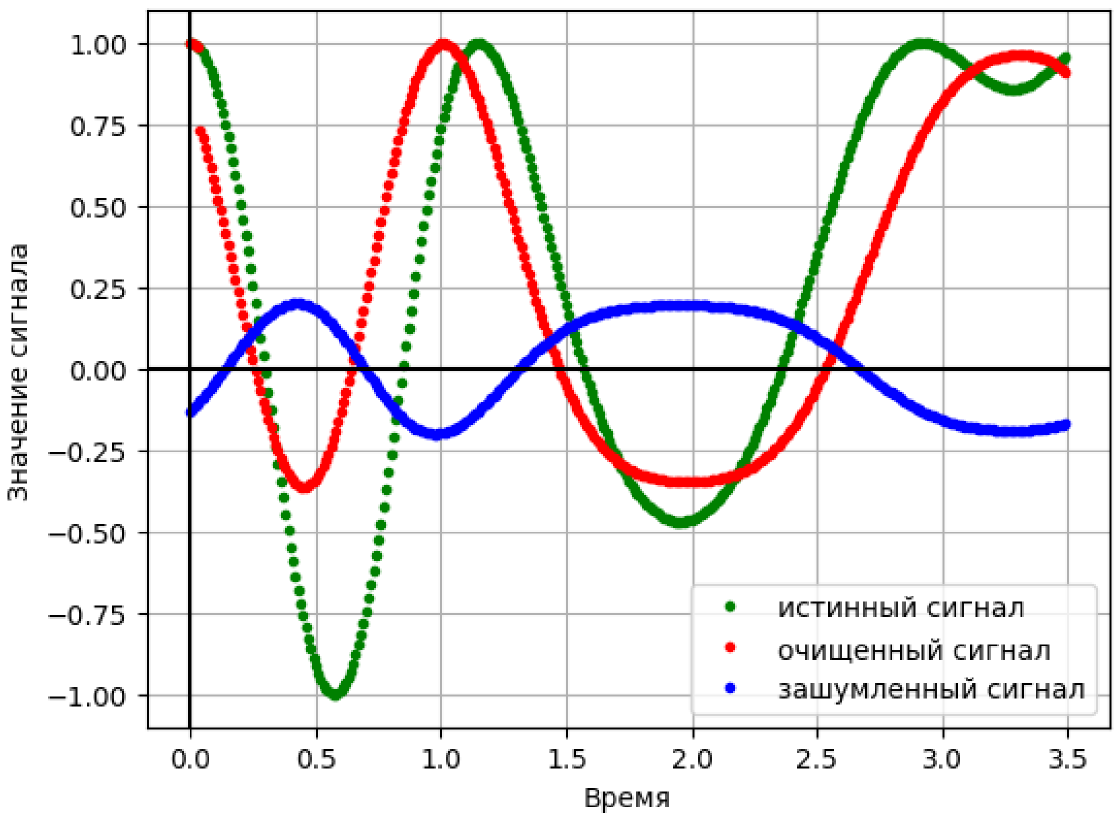
Выполняется предсказание на зашумленных данных noized_sequences. Предсказанные значения добавляются к списку X_PRED, и ошибки предсказания записываются в список errors2.

```
x_test = torch.FloatTensor(np.array(noized_sequences))
upcoming_points_pred = predict(adaptiveLinearNetwork2,
x_test).numpy().flatten().tolist()

X_PRED = X[:D] + upcoming_points_pred

errors2 = [pred - true for pred, true in zip(upcoming_points_pred,
upcoming_points_true)]
```

На графике зеленым цветом обозначен истинный сигнал, красным - очищенный сигнал, предсказанный моделью, и синим - зашумленный сигнал.

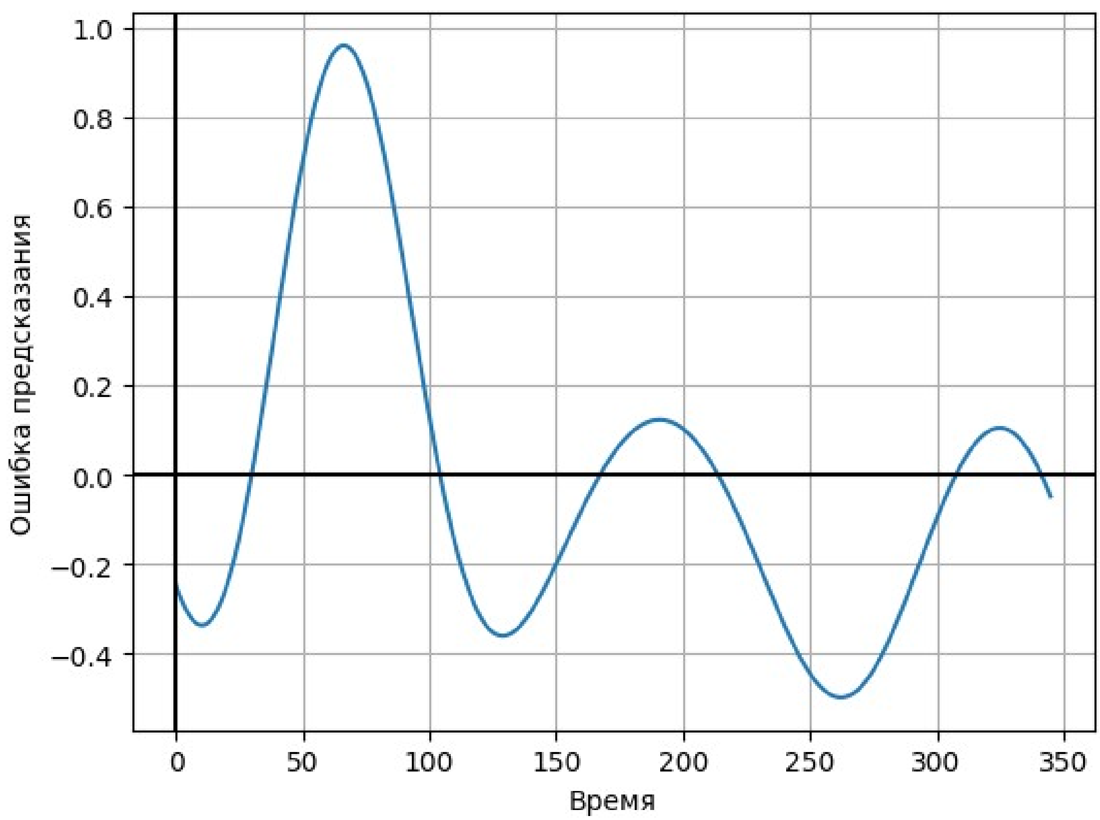
```
plt.plot(t, X, '.', color="green", label='истинный сигнал')
plt.plot(t, X_PRED, '.', color="red", label='очищенный сигнал')
plt.plot(t, Y, '.', color="blue", label='зашумленный сигнал')

plt.grid(True, which='both')
plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')
```

```
plt.xlabel('Время')
plt.ylabel('Значение сигнала')
plt.legend()
plt.show()
```



На графике отображены ошибки предсказания для каждого момента времени при решении задачи подавления помех.

```
plt.plot(errors2)

plt.grid(True, which='both')
plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')

plt.xlabel('Время')
plt.ylabel('Ошибка предсказания')
plt.show()
```



Выводы: В ходе выполнения лабораторной работы была успешно построена и обучена линейная нейросетевая модель для двух задач: аппроксимации функции и подавления помех в сигнале. После проведения обучения на 50 и 600 эпохах для каждой задачи соответственно, модель продемонстрировала хорошие результаты, что подтверждается анализом графиков и значений функции потерь.

Что я усвоил и из чего состояли задачи:

- Работа с временными последовательностями и задачами прогнозирования.
- Освоение процесса синхронного перемешивания данных для обучения модели.
- Изучение влияния числа эпох обучения на результаты модели в контексте двух различных задач: аппроксимации функции и подавления помех в сигнале.