

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ  
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №5

Выполнил  
В.Д.Головкина,  
студент группы АС66  
Проверил  
А. А. Крощенко,  
доц. кафедры ИИТ,  
« \_\_ » \_\_\_\_\_ 2025 г.

Цель работы: изучить и выполнить моделирование прогнозирующей нелинейной ИНС.

Выполнить моделирование прогнозирующей нелинейной ИНС. Для генерации обучающих и тестовых данных использовать функцию. Для прогнозирования использовать многослойную ИНС с одним скрытым слоем. В качестве функций активации для скрытого слоя использовать сигмоидную функцию, для выходного - линейную.

$$y = a \cos(bx) + c \sin(dx) .$$

Варианты заданий приведены в следующей таблице:

| № варианта | a   | b   | c    | d   | Кол-во входов ИНС | Кол-во НЭ в скрытом слое |
|------------|-----|-----|------|-----|-------------------|--------------------------|
| 1          | 0.1 | 0.1 | 0.05 | 0.1 | 6                 | 2                        |

```
import torch
import torch.nn as nn
import torch.optim as optim
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

a, b, c, d = 0.1, 0.1, 0.05, 0.1
n_inputs = 6
n_hidden = 2
n_samples = 200

def target_function(X):
    x1, x2, x3, x4, x5, x6 = X[:, 0], X[:, 1], X[:, 2], X[:, 3], X[:, 4], X[:, 5]
    return a * np.cos(b * x1) + c * np.sin(d * x2) + x3 * x4 - x5 + 0.5 * x6

np.random.seed(42)
X = np.random.uniform(-1, 1, (n_samples, n_inputs))
y = target_function(X)

split = int(0.8 * n_samples)
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]

X_train_tensor = torch.tensor(X_train, dtype=torch.float32) #тензоры
y_train_tensor = torch.tensor(y_train, dtype=torch.float32).view(-1, 1)
X_test_tensor = torch.tensor(X_test, dtype=torch.float32)
y_test_tensor = torch.tensor(y_test, dtype=torch.float32).view(-1, 1)

class MLP(nn.Module):
    def __init__(self):
        super(MLP, self).__init__()
        self.hidden = nn.Sequential(
            nn.Linear(n_inputs, n_hidden),
            nn.Sigmoid()
        )
```

```

        self.output = nn.Linear(n_hidden, 1) # лин выход

    def forward(self, x):
        x = self.hidden(x)
        return self.output(x)

model = MLP()
criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters(), lr=0.01)

n_epochs = 200 # обуч
losses = []

for epoch in range(n_epochs):
    model.train()
    y_pred = model(X_train_tensor)
    loss = criterion(y_pred, y_train_tensor)
    losses.append(loss.item())

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

plt.figure()
plt.plot(losses)
plt.xlabel("Эпоха")
plt.ylabel("Ошибка (MSE)")
plt.title("График изменения ошибки")
plt.grid(True)
plt.show()

model.eval()#прогноз по обуч выб
with torch.no_grad():
    y_train_pred = model(X_train_tensor).numpy().flatten()
    y_test_pred = model(X_test_tensor).numpy().flatten()

train_df = pd.DataFrame({ #резь
    "Эталонное значение": y_train,
    "Полученное значение": y_train_pred,
    "Отклонение": y_train_pred - y_train
})
train_df.to_csv("train_results.csv", index=False)

test_df = pd.DataFrame({
    "Эталонное значение": y_test,
    "Полученное значение": y_test_pred,
    "Отклонение": y_test_pred - y_test
})
test_df.to_csv("test_results.csv", index=False)

plt.figure()
plt.plot(y_train, label="Эталон")
plt.plot(y_train_pred, label="Прогноз")

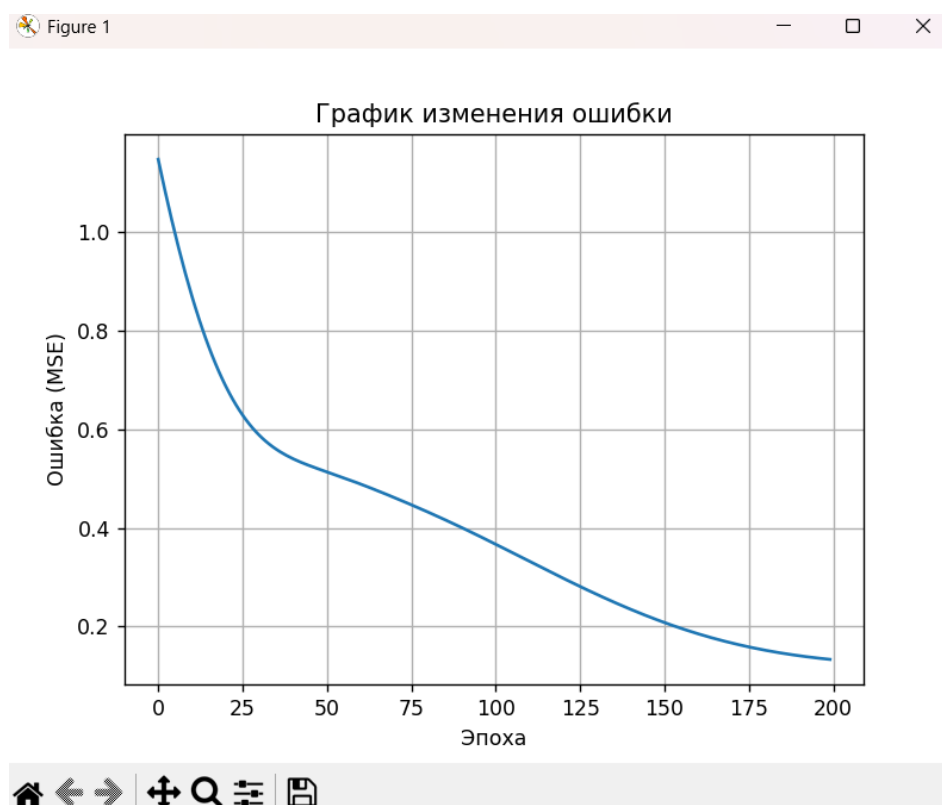
```

```
plt.title("Прогнозируемая функция на обучении")
plt.xlabel("Индекс")
plt.ylabel("Значение")
plt.legend()
plt.grid(True)
plt.show()

print("\n Результаты обучения (первые 5 строк):")
print(train_df.head())

print("\n Результаты прогнозирования (первые 5 строк):")
print(test_df.head())

print("\n Ошибка на последней эпохе:", losses[-1])
```





```
PS D:\y--ba\3 kurs\омо\1> & C:/Users/Пипка/AppData/Local/FontLab5/FontLab5.py"
```

Результаты обучения (первые 5 строк):

|   | Эталонное значение | Полученное значение | Отклонение |
|---|--------------------|---------------------|------------|
| 0 | 0.539979           | 0.477580            | -0.062399  |
| 1 | 1.616166           | 1.323255            | -0.292911  |
| 2 | 0.916107           | 0.563883            | -0.352225  |
| 3 | 0.218682           | 0.476766            | 0.258085   |
| 4 | -0.552632          | -0.605898           | -0.053265  |

Результаты прогнозирования (первые 5 строк):

|   | Эталонное значение | Полученное значение | Отклонение |
|---|--------------------|---------------------|------------|
| 0 | 0.491733           | 0.357707            | -0.134026  |
| 1 | 0.201002           | 0.162587            | -0.038415  |
| 2 | -0.095869          | -0.412553           | -0.316684  |
| 3 | 0.283288           | 0.362159            | 0.078871   |
| 4 | 0.877034           | 0.991010            | 0.113976   |

Ошибка на последней эпохе: 0.10666313022375107

```
PS D:\y--ba\3 kurs\омо\1>
```

Вывод: я изучила и выполнила моделирование прогнозирующей нелинейной ИНС.