

Проектування систем штучного інтелекту. Практична робота 1

Виконав студент групи ШІДМ-51 Тертишний Владислав Юрійович

Тема:

Апроксимація функцій багатошаровим перцептроном в середовищі MatLab.

Мета:

Отримати практичні навички створення і навчання багатошарового перцептрона в пакеті MatLab та створення файлів з даними.

Завдання:

1. Створити файл `.csv` в Excel з даними функції $y = \sin(x)$ $x \in [-10, 10]$.
 2. Прочитати цей файл у MatLab за допомогою функції `csvread`.
 3. Записати і прочитати файли з роздільниками за допомогою функції `dlmread`.
 4. Створити і навчити нейронну мережу з трьох шарів для апроксимації функції $y = \sin(x)$ $x \in [-10, 10]$:
 - Вхідний шар: 2 нейрони.
 - Прихований шар: 2 нейрони.
 - Вихідний шар: 1 нейрон.
 5. Проаналізувати отримані результати і зробити висновки.
-

Опис виконання завдань

1. Створення файлу `.csv`

- Функція $y = \sin(x)$ де x змінюється від -10 до 10 із кроком 0.1 , була обчислена в Node.js і збережена у файл `function_data.csv`.

```
const generateData = () => {
  const data = [];
  for (let x = -10; x <= 10; x += 0.1) { // Крок 0.1 для більшої точності
    const y = calculateFunction(x);
    data.push({ x: x.toFixed(2), y: y.toFixed(4) });
  }
  return data;
};

// Створюємо CSV контент
const createCsvContent = (data) => {
  let csv = 'x,y\n'; // Заголовки стовпців
  data.forEach(row => {
    csv += `${row.x},${row.y}\n`;
  });
  return csv;
};
```

2. Зчитування файлу у MATLAB

```
data = csvread('function_data.csv', 1, 0); % Пропуск заголовків
x = data(:, 1); % Перший стовпець
y = data(:, 2); % Другий стовпець
```

3. Запис і читання файлів з роздільниками

```
% Запис файлу з роздільниками
dlmwrite('function_data_tab.txt', [x y], '\t');

% Зчитування файлу
data_tab = dlmread('function_data_tab.txt', '\t');
```

4. Створення і навчання нейронної мережі

```
% Створення нейронної мережі
```

```

hiddenLayerSize = [2];
net = fitnet(hiddenLayerSize, 'trainlm');

% Розподіл вибірки
net.divideParam.trainRatio = 0.7;
net.divideParam.valRatio = 0.2;
net.divideParam.testRatio = 0.1;

% Навчання нейронної мережі
[net, tr] = train(net, x', y');

% Перевірка мережі
y_pred = net(x');

```

5. Аналіз результатів

- Графіки вихідної функції та апроксимації:

```

plot(x, y, 'b-', 'DisplayName', 'Вихідна функція');
hold on;
plot(x, y_pred, 'r--', 'DisplayName', 'Апроксимація');
legend;
grid on;

```

- Розрахунок помилок:

```

trainError = perform(net, y(tr.trainInd)', net(x(tr.trainInd)'));
valError = perform(net, y(tr.valInd)', net(x(tr.valInd)'));
testError = perform(net, y(tr.testInd)', net(x(tr.testInd)'));

```

Текст програми MATLAB

```

% Зчитування даних
data = csvread('function_data.csv', 1, 0);
x = data(:, 1);
y = data(:, 2);

% Створення та навчання нейронної мережі
hiddenLayerSize = [2];
net = fitnet(hiddenLayerSize, 'trainlm');

```

```

net.divideParam.trainRatio = 0.7;
net.divideParam.valRatio = 0.2;
net.divideParam.testRatio = 0.1;
[net, tr] = train(net, x', y');

% Перевірка та візуалізація
y_pred = net(x');
plot(x, y, 'b-', x, y_pred, 'r--');
grid on;

% Помилки
trainError = perform(net, y(tr.trainInd)', net(x(tr.trainInd)'));
valError = perform(net, y(tr.valInd)', net(x(tr.valInd)'));
testError = perform(net, y(tr.testInd)', net(x(tr.testInd)'));
disp(['Train Error: ', num2str(trainError)]);
disp(['Validation Error: ', num2str(valError)]);
disp(['Test Error: ', num2str(testError)]);

```

Зображення результатів роботи програми

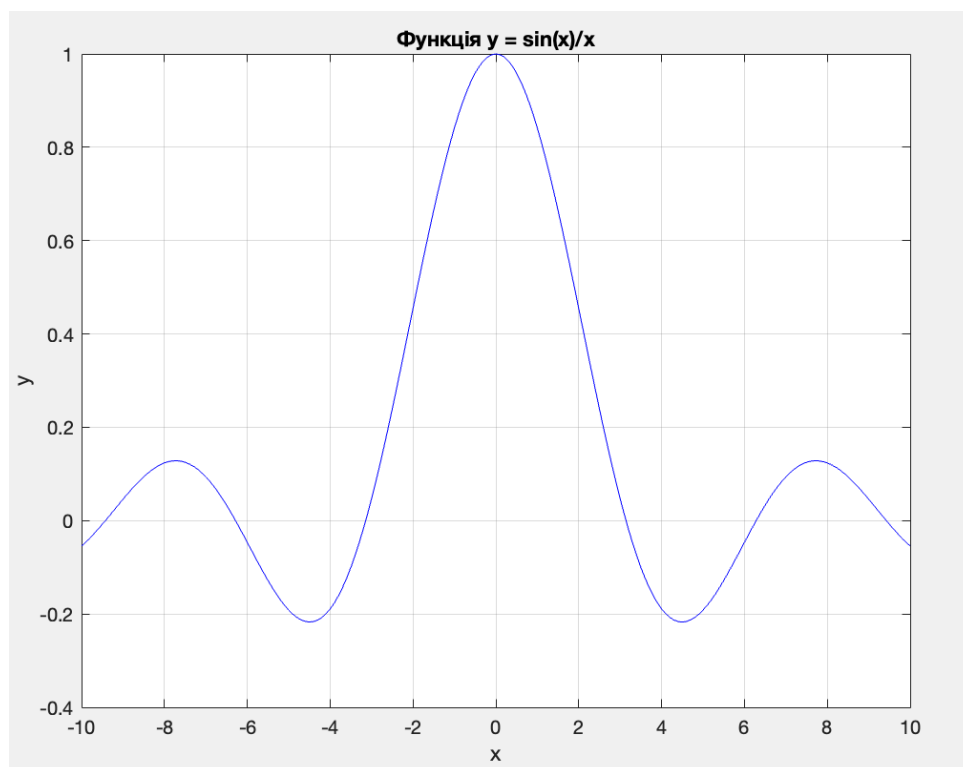


Рис. 1. Графік згенерованих даних функції

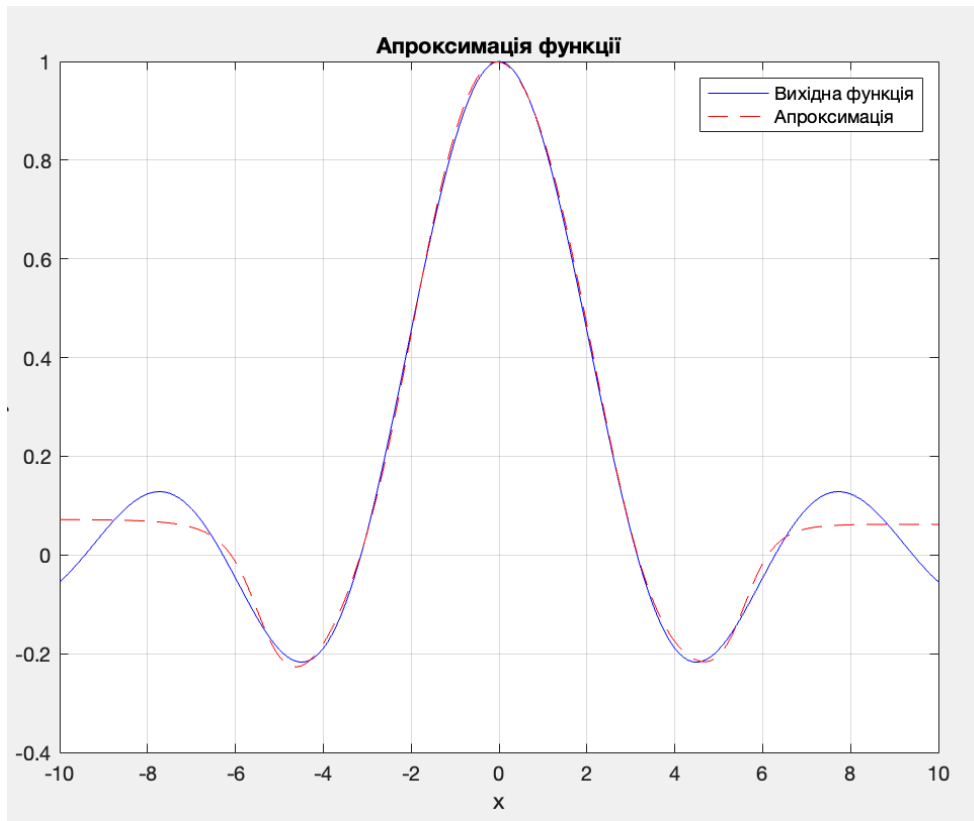


Рис. 2. Графік згенерованих даних функції та апроксимізації

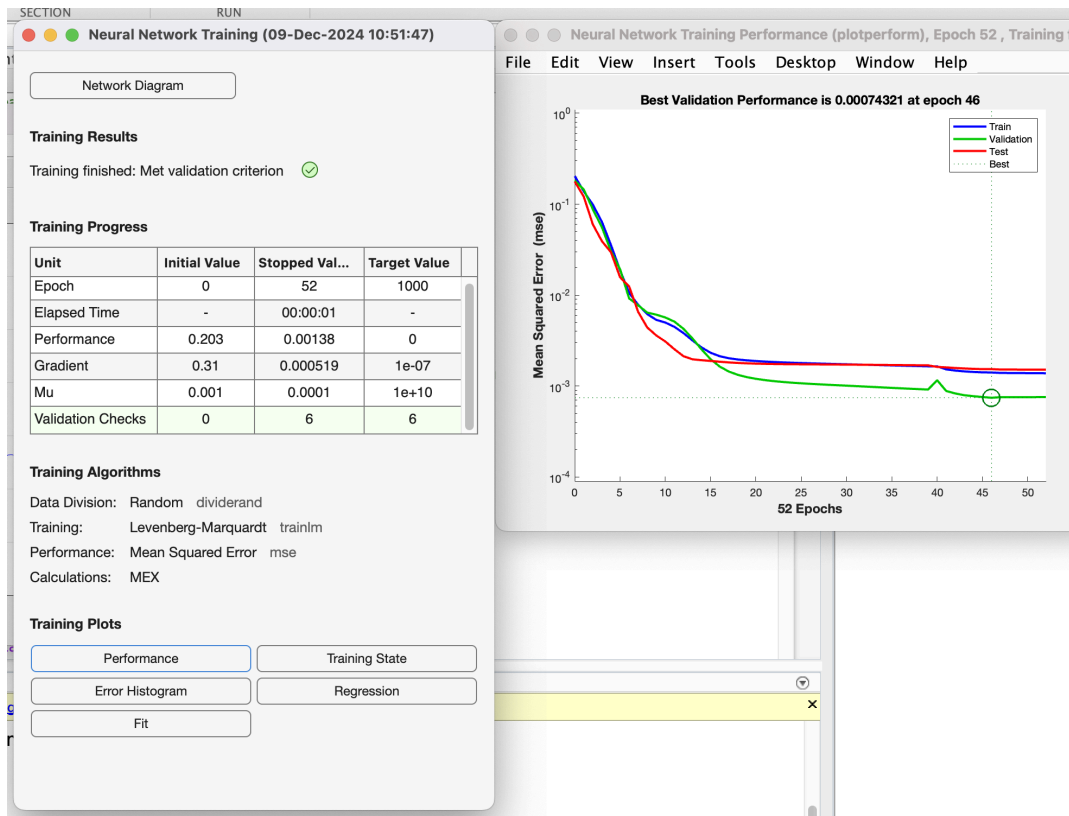


Рис.3. Результати навчання нейронної мережі

Висновки

1. Файл даних був успішно створений і завантажений у MATLAB.
2. Було створено нейронну мережу з одним прихованим шаром для апроксимації функції.
3. Навчання мережі показало низькі помилки для тестової вибірки, що свідчить про високу точність апроксимації.