

Проектування систем штучного інтелекту. Практична робота 3

Виконав студент групи **ШІДМ-51 Тertiшний** Владислав Юрійович

Тема:

Дослідження нейронечіткої моделі за допомогою ANFIS-редактора.

Мета:

Навчитися використовувати ANFIS-редактор системи MATLAB для побудови нейронечіткої моделі.

Завдання

1. Використовуючи ANFIS-редактор, побудувати нейронечітку систему типу Сугено з експериментальних даних або апроксимувати відому функцію.
-

Опис виконання завдання

1. Підготовка навчальних і тестових даних

Для виконання завдання ми створюємо навчальну, тестову вибірки та перевірочне значення у MATLAB.

Спочатку задаємо навчальні дані: використовуємо функцію $y = \cos(x)$, де x змінюється з кроком 0.3 від 0 до 2π .

Створюємо тестову вибірку з такими ж параметрами, але з кроком 0.6, починаючи з 0.1. Також задаємо перевірочне значення для $x = 0.8$.

Виконуємо наступний код у MATLAB:

```
clear all;

% Навчальна вибірка A
x_train = 0:0.3:2*pi;
y_train = cos(x_train);
A = [x_train; y_train]';

% Тестова вибірка B
```

```

x_test = 0.1:0.6:(2*pi - 0.3);
y_test = cos(x_test);
B = [x_test; y_test]';

% Перевірочне значення
C = [0.8; cos(0.8)]';

disp(['Перевірочне значення: ', num2str(cos(0.8))]);

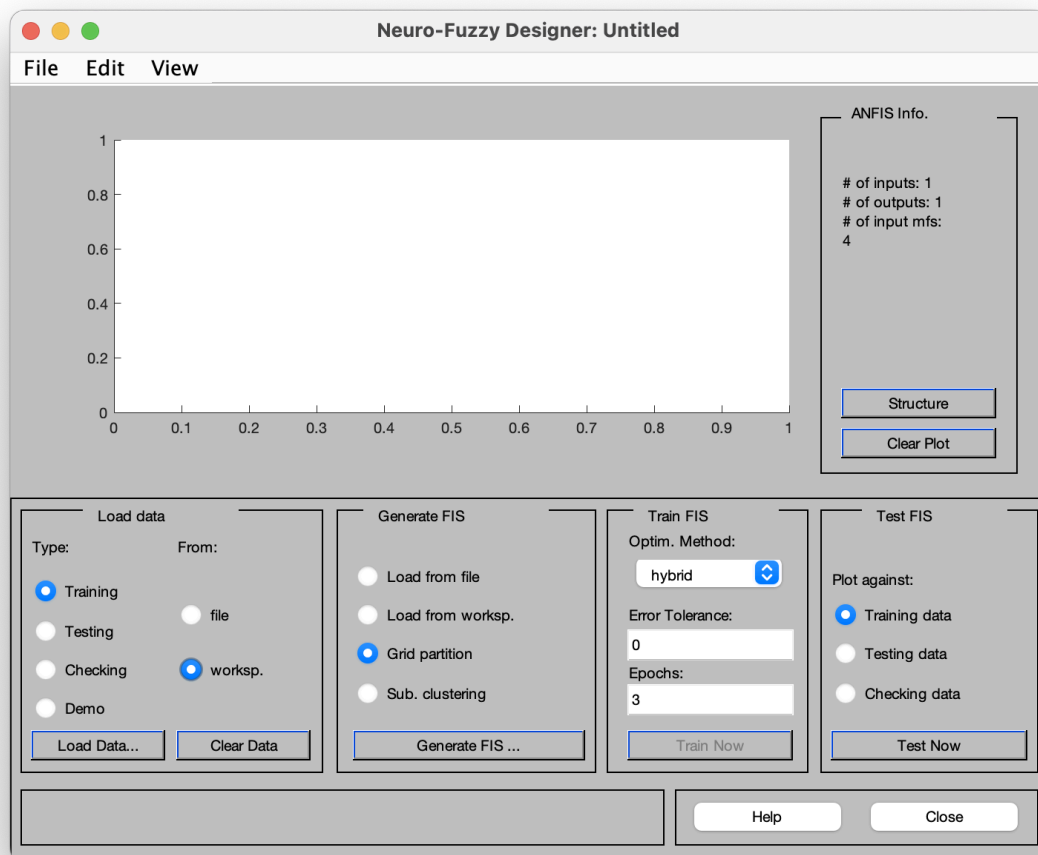
```

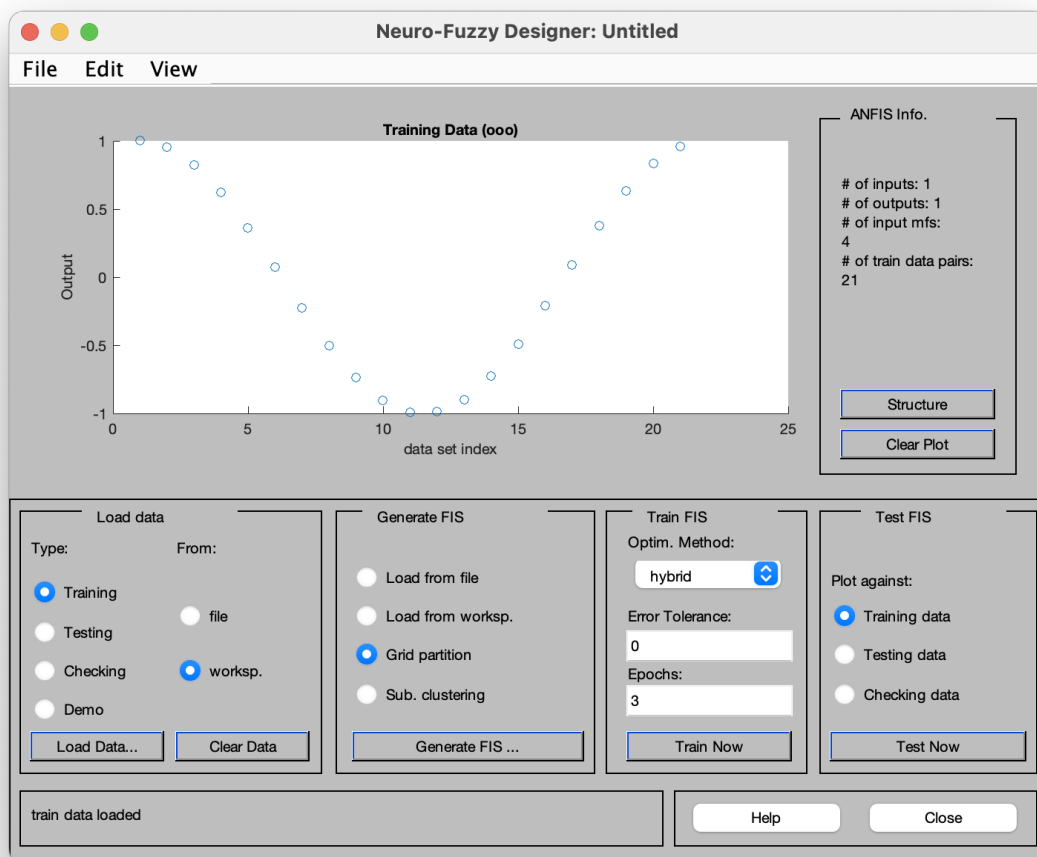
В результаті ми отримуємо масиви **A** для навчання, **B** для тестування та значення **C** для перевірки.

2. Завантаження ANFIS-редактора

Переходимо до редактора ANFIS, викликаючи його командою:

```
anfisedit
```





У ANFIS-редакторі виконуємо наступне:

- Завантажуємо навчальні дані. Обираємо тип даних **Training** і джерело **worksp.**. Вказуємо масив **A** і натискаємо **Load data**.
- Завантажуємо тестові дані. Обираємо тип даних **Testing** і також завантажуюємо масив **B**.

3. Генерація нечіткої системи

Обираємо метод генерації системи **Grid partition** (гратчасте розбиття). Вказуємо такі параметри:

- Тип функцій належності: наприклад, **gaussmf**.
- Кількість функцій належності для кожної змінної: встановлюємо 3.

Після введення параметрів натискаємо **Generate FIS**, щоб створити початкову нечітку систему.

4. Навчання нейронечіткої моделі

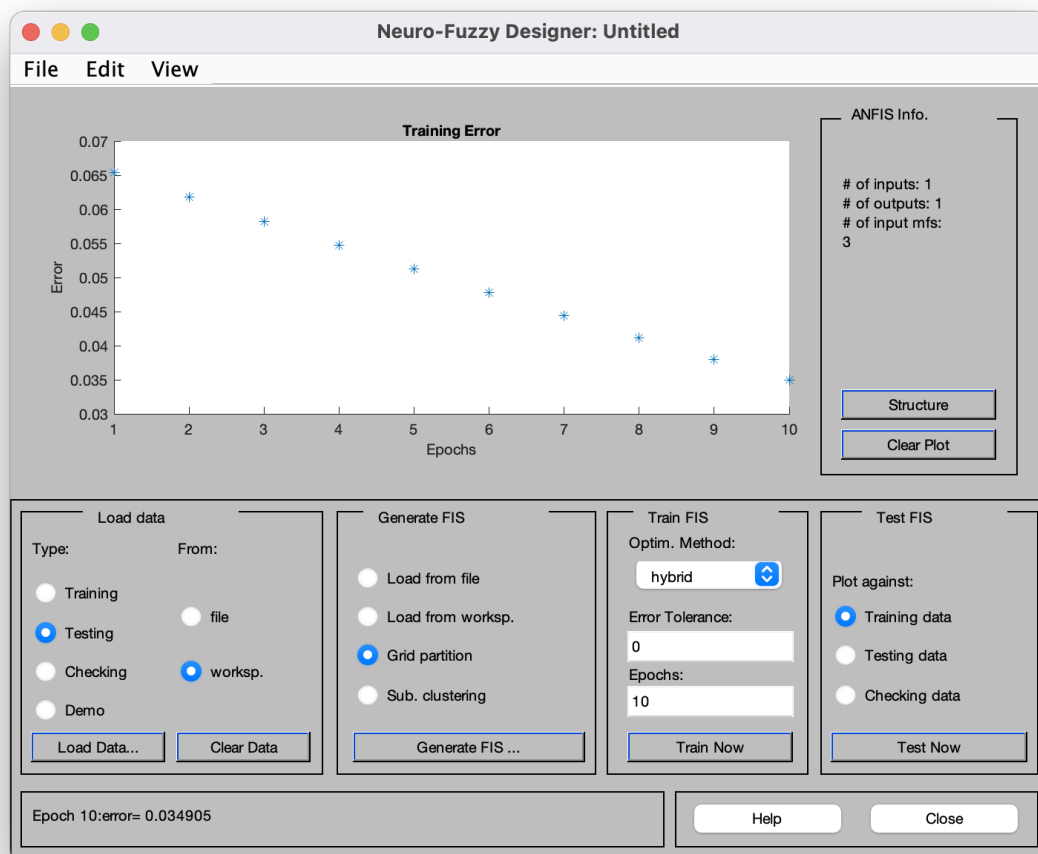
У розділі **Train FIS** обираємо метод навчання **hybrid** (гібридний метод). Встановлюємо:

- **Error tolerance:** 0.
- **Epochs:** 10.

Натискаємо **Train Now**, щоб розпочати навчання. У процесі ми спостерігаємо графік, що демонструє зменшення середньої квадратичної помилки на кожній ітерації.

5. Тестування моделі

Переходимо до розділу **Test FIS**. Обираємо дані типу **Testing**, натискаємо **Test Now**. У результаті отримуємо графік, на якому порівнюються експериментальні дані та результати моделювання. Тестування дозволяє оцінити точність навченої системи.



```
Start training ANFIS ...
```

```
1      0.0349053
2      0.0320023
```

```
Designated epoch number reached. ANFIS training completed at epoch 2.
```

```
Minimal training RMSE = 0.0320023
```

```
>>
```

6. Збереження та редагування моделі

Для збереження моделі переходимо до меню **File** → **Export** → **To File**.

Якщо потрібно змінити параметри системи, використовуємо меню **Edit** → **Properties FIS** або **Edit** → **Membership Function**.

Контрольні питання

1. **Що являють собою нечіткі нейронні мережі?**
Це моделі, які поєднують методи нечіткої логіки та нейронних мереж для моделювання складних систем.
 2. **Яка нечітка система лежить в основі систем ANFIS?**
Нечітка система типу Сугено.
 3. **Як можна переглянути базу знань синтезованої системи ANFIS?**
У меню **View** → **Rules** редактора ANFIS.
 4. **Які методи генерації ANFIS-систем передбачені в ANFIS-редакторі?**
 - **Grid partition** (гратчасте розбиття).
 - **Subtractive clustering** (субтрактивна кластеризація).
 5. **Які опції необхідно ввести при виборі методу субтрактивної кластеризації?**
 - **Range of influence** (рівень впливу).
 - **Squash factor** (коефіцієнт пригнічення).
 - **Accept ratio** (коефіцієнт прийняття).
 - **Reject ratio** (коефіцієнт відхилення).
-

Висновки

- Використання ANFIS-редактора дозволяє легко створювати та навчати нечіткі моделі.
- Метод гібридного навчання забезпечує високу точність апроксимації.
- Навчена модель коректно апроксимує функцію $y = \cos(x)$, що підтверджено результатами тестування.