

## Вступ

В останні роки спостерігається підвищений інтерес до нейронних мереж, які знайшли застосування в різних областях людської діяльності - бізнесі, медицині, техніці. Нейронні мережі використовуються при вирішенні завдань прогнозування, класифікації, управління. Такий вражаючий успіх визначається декількома причинами:

- Нейронні мережі - це виключно потужний метод імітації процесів і явищ, що дозволяє відтворювати надзвичайно складні залежності. Нейронні мережі по своїй природі є нелінійними, в той час як протягом багатьох років для побудови моделей використовувався лінійний підхід. Крім того, у багатьох випадках нейронні мережі дозволяють подолати "прокляття розмірності", обумовлене тим, що моделювання нелінійних явищ в разі великого числа змінних вимагає величезної кількості обчислювальних ресурсів.

- Інша особливість нейронних мереж пов'язана з тим, що вони використовують механізм навчання. Користувач нейронної мережі підбирає представницькі дані, а потім запускає алгоритм навчання, який автоматично налаштовує параметри мережі. При цьому від користувача, звичайно, потрібно якийсь набір евристичних знань про те, як слід відбирати і готувати дані, вибирати потрібну архітектуру мережі та інтерпретувати результати, проте рівень знань, необхідний для успішного застосування нейронних мереж, набагато скромніше, ніж, наприклад, при використанні традиційних методів.

Штучні нейронні мережі базуються на біологічній моделі нервової системи. Нервова система людини, побудована з елементів, що називаються нейронами. Близько  $10^{11}$  нейронів беруть участь у приблизно в  $10^{15}$  передавальних зв'язках, що мають довжину метр і більше. Кожен нейрон має багато якостей, спільних з іншими елементами організму людини, але його унікальною здатністю є прийом, обробка і передача електрохімічних сигналів по нервових шляхах, які утворюють комунікаційну систему мозку.

Інтенсивність сигналу, одержуваного нейроном (отже, і можливість його активації), сильно залежить від активності лінії зв'язку. Кожна лінія зв'язку має певну протяжність, і спеціальні хімічні речовини передають сигнал вздовж неї. Передбачається, що навчання полягає в першу чергу в зміні інтенсивності зв'язків. Наприклад, в класичному досвіді Павлова кожен раз безпосередньо перед годуванням собаки дзвонив дзвоник і собака швидко навчилася пов'язувати дзвінок дзвоника з їжею.

Мозок, побудований з дуже великого числа нейронів (кожен з яких приймає зважену суму вхідних сигналів і при певних умовах має можливість передавати сигнал далі), здатний вирішувати надзвичайно складні завдання. Таку ж модель реалізують і штучні нейронні мережі.

Теорія нейронних мереж виникла з досліджень в області штучного інтелекту, а саме зі спроб відтворити здатність нервових біологічних систем навчатися і виправляти помилки, моделюючи низькорівневу структуру мозку. Щоб створити штучний інтелект, необхідно побудувати систему з архітектурою схожою на будову мозку.

Нейронну мережу можна навчати для виконання конкретної функції, регулюючи значення коефіцієнтів (ваг) зв'язку. Зазвичай штучні нейронні мережі налаштовуються або навчаються так, щоб конкретні входи перетворювалися в заданий цільовий вихід. Мережа налаштовується (навчається), ґрунтуючись на порівнянні сигналів виходу і цілі до тих пір, поки вихід мережі не буде відповідати меті. Щоб навчити мережу при такому керованому навчанні, як правило, використовується багато пар значень сигналів вхід / мета.

Нейронні мережі навчаються виконувати складні функції в різних областях, включаючи розпізнавання образів, ідентифікацію, класифікацію об'єктів, розпізнавання мови, системи технічного зору та системи управління. У цих випадках застосовуються досить прості нейронні мережі, але навіть вони виявляються потужним інструментом в арсеналі фахівця.

Сьогодні можуть бути побудовані штучні нейронні мережі для вирішення таких завдань, які є важкими як для комп'ютерів, побудованих за традиційною схемою, так і для людини.

Теорія нейронних мереж розвивається протягом останніх шести десятиліть, але знайшла широке застосування тільки в останні 25 років. Тому суворі термінологія в області нейронних мереж ще не встановилася. Ця теорія нейронних мереж відрізняється від теорії управління або оптимізації, де термінологія, основні математичні методи і процедури проектування вивірені при вирішенні практичних завдань протягом багатьох років.

**ІІІІ (пакет прикладних програм) Neural Network Toolbox** служить засобом, який допомагає користувачам розвивати методи проектування і розширювати сферу застосування нейронних мереж.

Оскільки дана область досліджень характеризується значною новизною, в межах даного курсу ми розглянемо лише основні положення теорії нейронних мереж і застосування їх на деяких прикладах.

Нейронні мережі знайшли застосування в багатьох областях техніки, де вони використовуються для вирішення численних прикладних задач:

- **в космонавтиці і аеронавтиці** - для побудови систем автоматичного пілотування літаків, імітації траєкторій польоту, розробки перспективних конструкцій автопілотів, моделювання і виявлення несправностей агрегатів літальних апаратів, управління повітряним рухом, обробки аерокосмічної інформації;

- **в автомобілебудуванні** - для проектування технічних засобів АСУ ТП;

- **в банківській справі** - для автоматичного читання документів і їх контролю, оцінки ефективності кредитних вкладень;

- **у військовій справі** - для управління зброєю, стеження за цілями, виділення і розпізнавання об'єктів, побудови нових типів датчиків, обробки звукових, радіолокаційних і телевізійних зображень, постановки і придушення перешкод, ідентифікації сигналів і зображень;

- **в електроніці** - для управління процесом проектування і розміщення мікросхем на платі, нелінійного моделювання та аналізу відмови мікросхем, для побудови систем машинного зору і синтезу мови;

- **в фінансовому справі** - для аналізу кредитних потоків, оцінки нерухомості, загального фінансового аналізу, прогнозування вартості валюти;

- **в страховій справі** - для визначення оптимальної стратегії страхування;

- **в промисловому виробництві** - для управління виробничими процесами, аналізу продукції, діагностики роботи машин, контролю якості, тестування продукції, аналізу якості зварювальних робіт, шліфувальних операцій, аналізу і синтезу хімічних речовин, прогнозування ціни продукту;

- **в медицині** - для аналізу ракових клітин, діагностики захворювань, конструювання протезів, оптимізації часу трансплантації, планування витрат лікарні, консультацій під час відсутності фахівців;

- **в нафтогазовій промисловості** - для розвідки родовищ;

- **в робототехніці** - для керування роботами, побудови контролерів роботів і маніпуляторів і систем технічного зору;

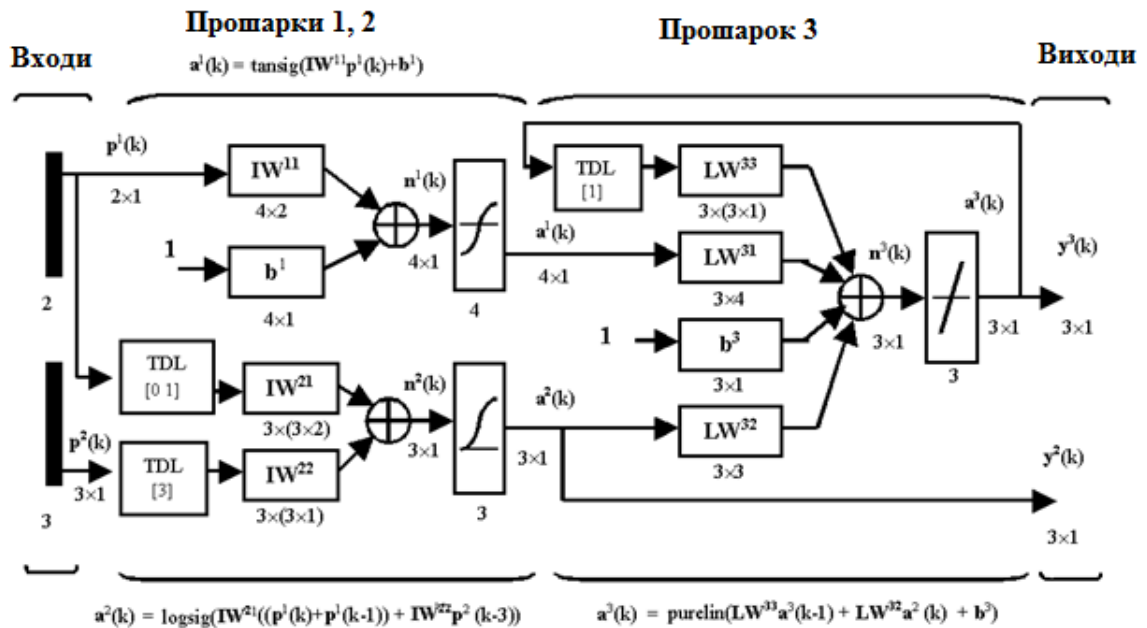
- **при передачі даних** - для стиснення і відображення даних, розпізнавання мови, класифікації голосних звуків, перетворення тексту в мову, для синхронного перекладу.

Цей перелік можна продовжити і далі. Широке застосування нейронних мереж, значні ресурси, що вкладаються в створення програмного забезпечення та апаратури для реалізації нейронних мереж, показують, що є велика зацікавленість у розробці штучних нейронних мереж.

**Позначення для прошарків нейронної мережі:**

- верхній індекс з одного символу застосовується для того, щоб вказати належність деякого елемента прошарку. Наприклад, вектор входу прошарку 3 позначається як  $p^3$ ;
- верхній індекс з двох символів застосовується для того, щоб вказати джерело сигналу ( $l$ ) і його пункт призначення ( $k$ ); він використовується для позначення матриць ваг входу  $IW^{k,l}$  і матриць ваг прошарку  $LW^{k,l}$ . Наприклад, матриця ваг від прошарку 2 до прошарку 4 буде позначатися як  $LW^{4,2}$ .

Приклад структурної схеми нейронної мережі:



На малюнку наведені позначення, що використовуються при зображенні структурних схем нейронних мереж.

При переході від математичних позначень до кодів системи MATLAB слід керуватися наступними правилами:

- Верхні індекси відповідають індексам масиву комірок:  $p^1 \rightarrow p\{1\}$ ;
- Нижні індекси відповідають індексам масиву чисел:  $p_2 \rightarrow p(2)$ ,  $p_2^1 \rightarrow p\{1\}(2)$ ;
- Індекс в круглих дужках відповідає другому індексу масиву комірок:  $p^1(k) \rightarrow p\{1, k\}$ .

## ППП Neural Network Toolbox. GUI-інтерфейс для ППП NNT

До складу пакету нейронних мереж ПППNNT входить інструментальний засіб NNTool. Цей графічний інтерфейс дозволяє, не звертаючись до командного вікна MATLAB, виконувати створення, навчання, моделювання, а також імпорт та експорт нейронних мереж і даних, використовуючи тільки інструментальні можливості GUI-інтерфейсу. Звичайно, такі інструменти найефективніші лише на початковій стадії роботи з пакетом, оскільки мають певні обмеження, але при цьому користувач виграє у часі та ефективності освоєння нових об'єктів.

Виклик GUI-інтерфейсу NNTool можливий або командою `nntool` з командного рядка, або з вікна запуску програм LaunchPad за допомогою опції NNTool з розділу Neural Network Toolbox. Після виклику на екрані терміналу з'являється вікно Network/Data Manager (Управління мережею/даними) (рис. 1)

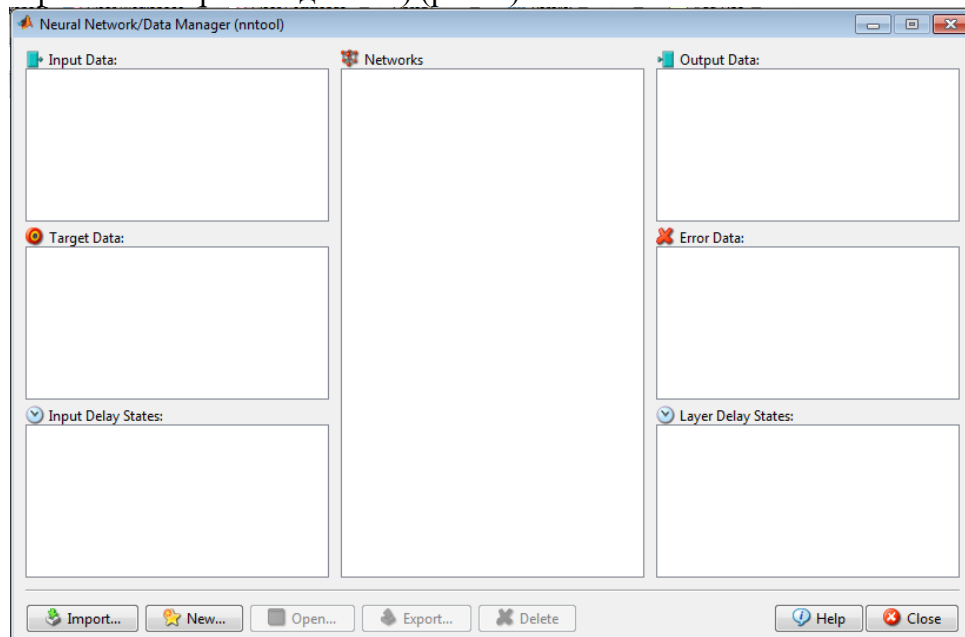


Рис. 1

Тут:

Help – кнопка виклику вікна підказки Network/Data Manager Help (рис. 2);

New Data... – кнопка виклику вікна формування даних Create New Data (рис. 3);

New Network... – кнопка виклику вікна створення нової нейронної мережі Create New Network (рис. 4);

Import... – кнопка виклику вікна для імпорту або завантаження даних Import or Load to Network/Data Manager (рис. 1.17);

Export... – кнопка виклику вікна для експорту або запису даних у файл Export or Save from Network/Data Manager (рис. 1.18).

Кнопки View, Delete стають активними лише після створення та активізації даних, що стосуються послідовностей входу, мети, виходу чи помилок мережі. Кнопка View дозволяє переглянути, а кнопка Delete видалити активізовані дані.

Кнопки View, Delete, Initialize..., Simulate..., Train..., Adapt... стають активними після створення та активізації самої нейронної мережі. Вони дозволяють переглянути, видалити, ініціалізувати, промодельовувати, навчити або адаптувати нейронну мережу та будуть описані нижче.

Насамперед розглянемо призначення та способи роботи з переліченими вище вікнами.

Вікно Network/Data Manager Help. Це вікно підказки показано на рис. 2 та описує правила роботи з диспетчером Network/Data Manager при створенні нейронної мережі.

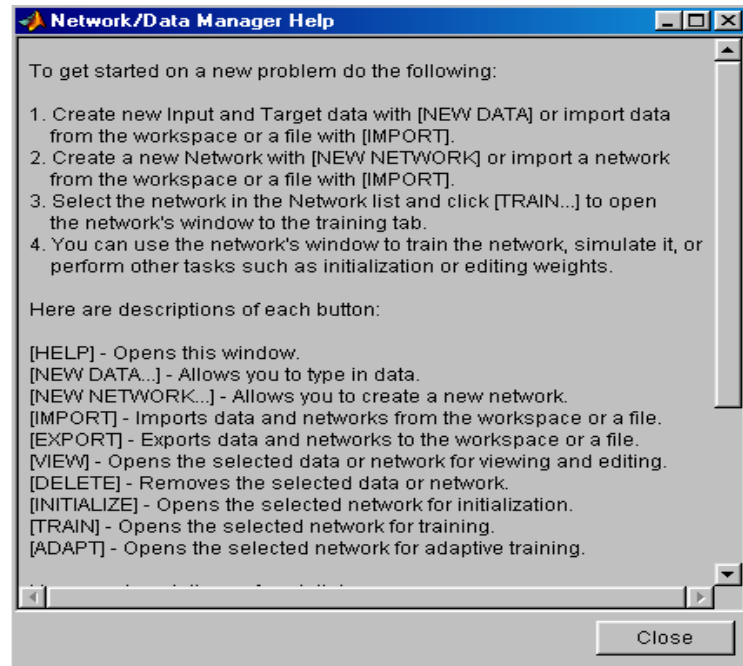


Рис. 2

Щоб створити нейронну мережу, необхідно виконати такі операції:

- Сформувати послідовності входів та цілей (кнопка New Data), або завантажити їх із робочої області системи MATLAB чи з файлу (кнопка Import).
- Створити нову нейронну мережу (кнопка New Network) або завантажити її з робочої області системи MATLAB або з файлу (кнопка Import).
- Вибрати тип нейронної мережі та натиснути кнопку Train..., щоб відкрити вікно для встановлення параметрів процедури навчання.
- Відкрити вікно Network для перегляду, ініціалізації, моделювання, навчання та адаптації мережі.

Вікно Create New Data. Це вікно показано на рис. 3 і включає 2 області редагування тексту для запису імені даних (область Name) і введення самих даних (область Value), а також 6 кнопок для вказівки типу даних, що вводяться.

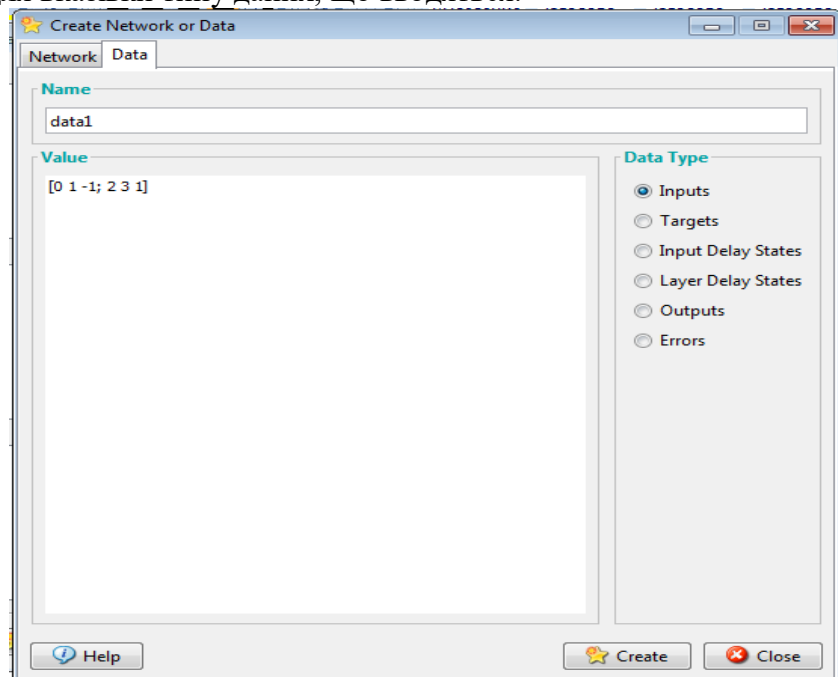


Рис. 3

Розрізняють такі типи даних:

- Inputs(Входи) – послідовність значень входів;
- Targets (Цілі) - послідовність значень цілей;
- Input Delay States (Стани лінійної затримки входу) – початкові умови лінії затримки на вході;
- Layer Delay States (Стан ЛЗ прошарку) - початкові умови лінії затримки в прошарку;
- Outputs (Виходи) – послідовність значень виходів мережі;
- Errors (Помилки) – різниця значень цілей та виходів.

Як правило, користувач задає лише послідовності входу та цілей, тобто типи даних Inputs і Targets.

Вікно Create New Network. Це вікно показано на рис. 4 і включає поля для завдання параметрів мережі, що створюється. Залежно від типу мережі кількість полів та їх назви змінюються.

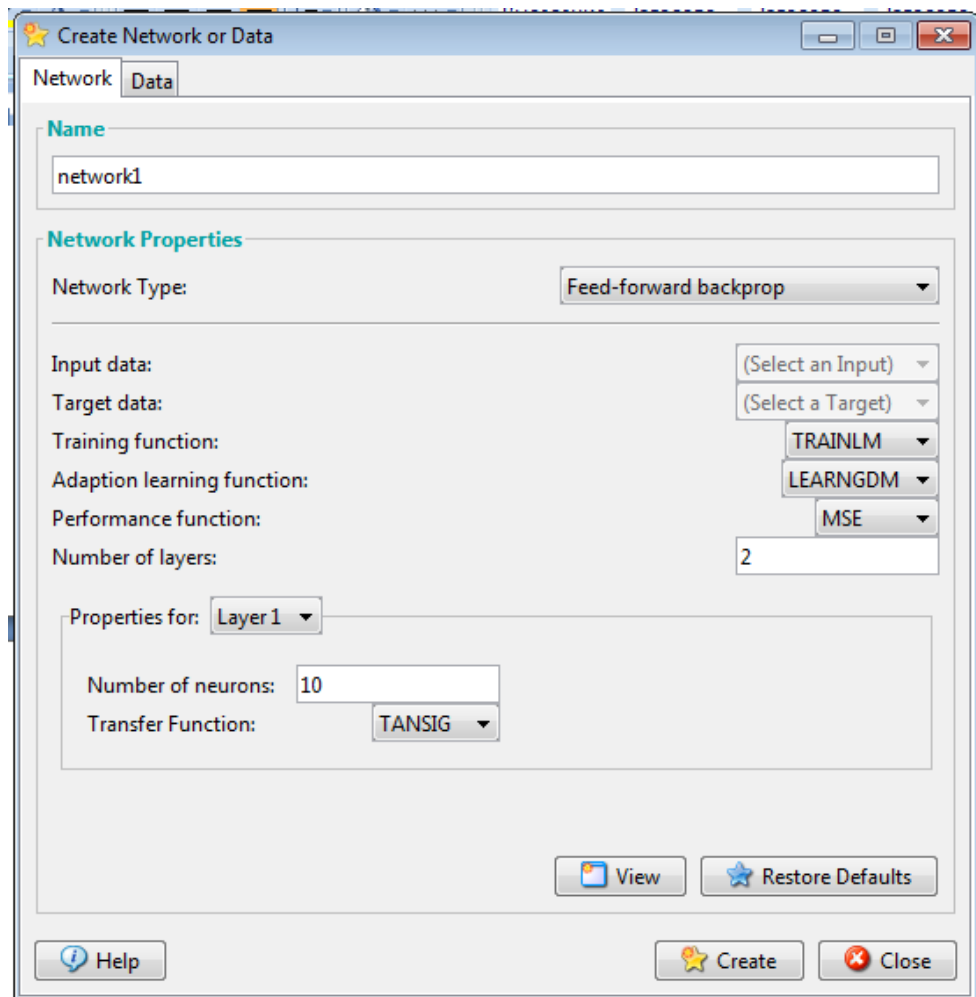


Рис. 4

Звернемося до опису полів.

Network Name (Ім'я мережі) – стандартне ім'я мережі, яке присвоюється GUI-інтерфейсом NNTool; у процесі створення нових мереж порядковий номер змінюватиметься автоматично.

Network Type(Тип мережі) – список мереж, доступних для роботи з інтерфейсом NNTool. Для зручності цей список повторений у таблиці нижче.

| № | Name        | Назва             | Кількість прошарків | Параметри, що навчаються |
|---|-------------|-------------------|---------------------|--------------------------|
| 1 | Competitive | Конкуруюча мережа | 1                   | $IW\{1, 1\}$ , $b\{1\}$  |

|    |                              |  |   |  |
|----|------------------------------|--|---|--|
| 2  | Cascade-forward backprop     | Каскадна мережа з прямим поширенням сигналу і зворотним поширенням помилки | 2 | IW{1, 1}, b{1}, LW{2, 1}, IW{2, 1}, b{2} |
| 3  | Elman backprop               | Мережа Елмана зі зворотним поширенням помилки                              | 2 | IW{1, 1}, b{1}, LW{2, 1}, b{2}, LW{1, 1} |
| 4  | Feed-forward backprop        | Мережа з прямим поширенням сигналу і зворотним поширенням помилки          | 2 | IW{1, 1}, b{1}, LW{2, 1}, b{2}           |
| 5  | Feed-forward time delay      | Мережа із запізненням  | 2 | IW{1, 1}, b{1}, LW{2, 1}, b{2}           |
| 6  | Generalized regression       | Узагальнена регресійна мережа  | 2 | IW{1, 1}, b{1}, LW{2, 1}                 |
| 7  | Hopfield                     | Мережа Хопфілда  | 1 | LW{1, 1}, b{1}                           |
| 8  | Linear layer (design)        | Лінійний шар (створення)   | 1 | IW{1, 1}, b{1}                           |
| 9  | Linear layer (train)         | Лінійний шар (навчання)  | 1 | IW{1, 1}, b{1}                           |
| 10 | LVQ                          | Мережа для класифікації вхідних векторів                                   | 2 | IW{1, 1}, LW{2, 1}                       |
| 11 | Perceptron                   | Персептрон   | 1 | IW{1, 1}, b{1}                           |
| 12 | Probabalistic                | Ймовірнісна мережа   | 2 | IW{1, 1}, b{1}, LW{2, 1}                 |
| 13 | Radial basis (exact fit)     | Радіально-базисна мережа з нульовою помилкою                               | 2 | IW{1, 1}, b{1}, LW{2, 1}                 |
| 14 | Radial basis (fewer neurons) | Радіально-базисна мережа з мінімальним числом нейронів (ітеративна)        | 2 | IW{1, 1}, b{1}, LW{2, 1}, b{2}           |
| 15 | Self-organizing map          | Самоорганізована карта Кохонена  | 1 | IW{1, 1}                                 |

*Зауваження. В новіших версіях MATLAB уже доступно значно більше архітектур для нейронних мереж.*

Продовжимо опис полів Рис. 4.

Input ranges(Діапазони входу) – допустимі межі входів, які або призначаються користувачем, або визначаються автоматично на ім'я вхідної послідовності, що вибирається зі списку Get from Inp... .

Training function (Функція навчання) – список навчальних функцій.

Adaption learning function (Функції налаштування режиму адаптації) – список функцій налаштувань.

Performance function (Функція якості навчання) – перелік функцій оцінки якості навчання.

Number of layers (Кількість прошарків) – кількість шарів нейронної мережі.

Properties for (Властивості) – список шарів: Layer1 (Прошарок 1), Layer2 (Прошарок 2).

Number of neurons (Кількість нейронів) – кількість нейронів у прошарку.

Transfer function(Функція активації) – функція активації прошарку.

Вікно Import or Load to Network/Data Manager. Це вікно показано на рис. 5 та включає 3 поля.

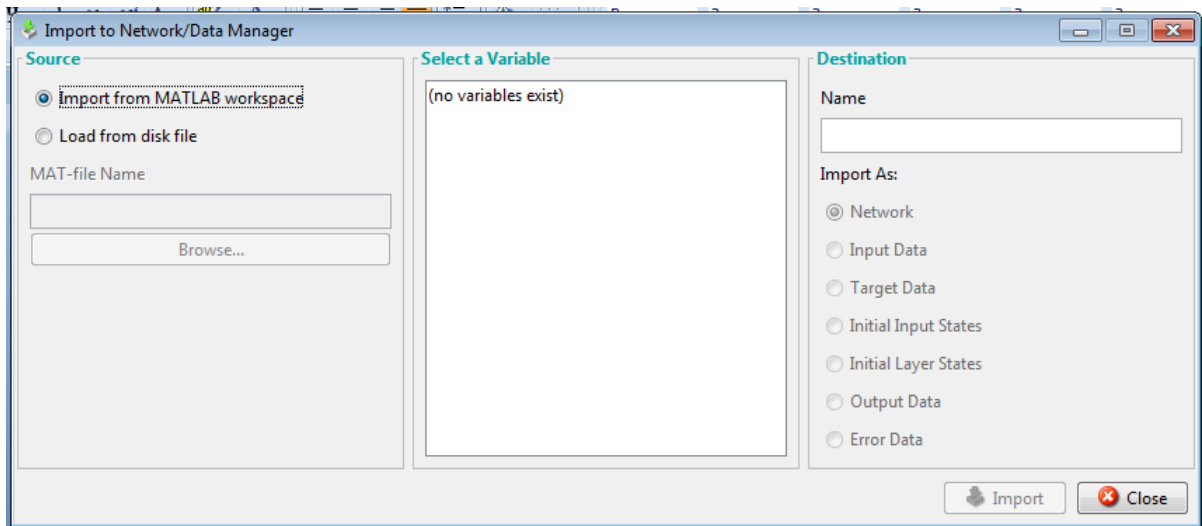


Рис. 5

Source (Джерело) – поле для вибору джерела даних. Це або робоча область системи MATLAB (кнопка вибору Import from MATLAB Workspace), або файл (кнопка вибору Load from disk file).

Якщо вибрано першу кнопку, то в полі Select a Variable ви можете бачити всі змінні робочої області і, вибравши одну з них, можете визначити її в полі Destination (Призначення) як послідовність входу Inputs (Входи).

Якщо вибирається кнопка Load from disk file, то активізується поле MAT-file Name і кнопка Browse, що дозволяє розпочати пошук та завантаження файлу з файлової системи.

Вікно Export or Save from Network/Data Manager. Це вікно показано на рис. 6 і дозволяє передати дані з робочої області GUI-інтерфейсу NNTool в робочу область системи MATLAB або записати їх у вигляді файлу на диску.

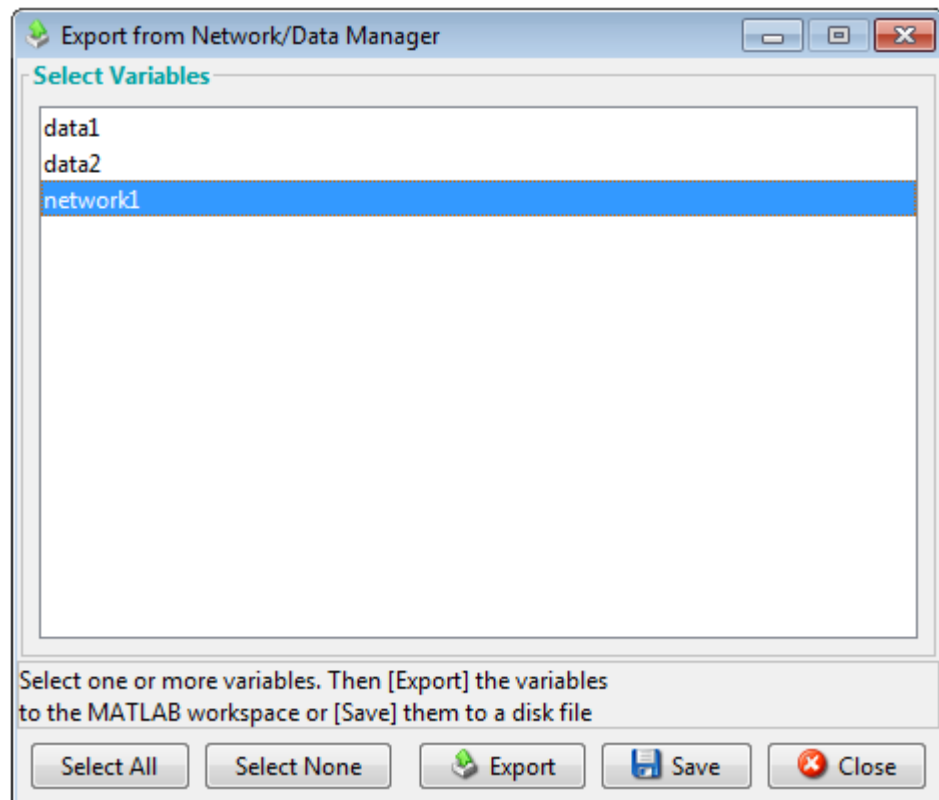


Рис. 6

У даному випадку нами обрана змінна network1, яка належить до класу networkobject і описує нейронну мережу. Після того, як ця змінна експортована в робочу область,



можна, наприклад, побудувати модель нейронної мережі в системі Simulink за допомогою оператора gensim.

Діалогова панель Network показана на рис. 7.

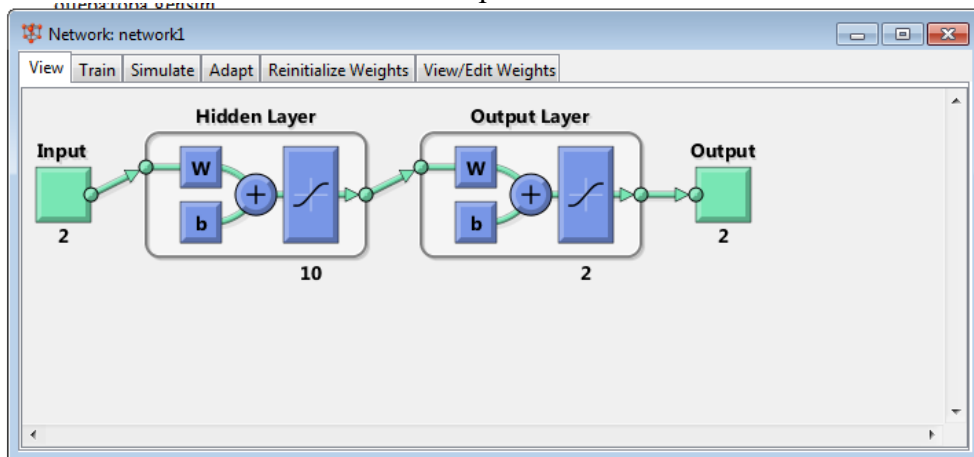


Рис. 7

Вона відкривається тільки в тому випадку, коли у вікні Network/Data Manager виділена створена мережа і стають активними кнопки View, Initialize, Simulate, Train, Adapt.

Панель має вкладки:

- View (Переглянути) – структура мережі;
- Train (Навчання) – навчання мережі;
- Simulate (Моделювання) - моделювання мережі;
- Adapt(Адаптація) – адаптація та налаштування параметрів мережі;

**Зауваження.** Не у всіх версіях MATLAB ця функція працює коректно. Тому надалі не розглядатимемо навчання мереж, використовуючи **Adapt**.

- Reinitialize/View/Edit Weights (Ваги) – повторна ініціалізація, перегляд, редагування встановлених ваг і зсувів.

Особливості роботи з відповідними вікнами будуть розглянуті на наведених нижче прикладах створення конкретних нейронних мереж.

**Приклад.** Нейронна мережа із прямою передачею сигналу.

Завдання: Створити та навчити нейронну мережу виконання операції  $y=x_1^2+x_2$ , якщо задані послідовності входу  $P= [1 \ 0.5 \ 0 \ 1; -2 \ 0 \ 0.5 \ 1]$  і цілі  $T = [-1 \ 0.25 \ 0.5 \ 2]$ .

Сформуємо послідовності входів та цілей у робочій області GUI-інтерфейсу NNTool, використовуючи вікно Create New Data.

Виберемо нейронну мережу типу Feed-forward backprop з прямою передачею сигналу і зі зворотним поширенням помилки.

Потім виконується навчання мережі, для чого вибирається закладка Train та відкривається діалогова панель, показана на рис. 7.

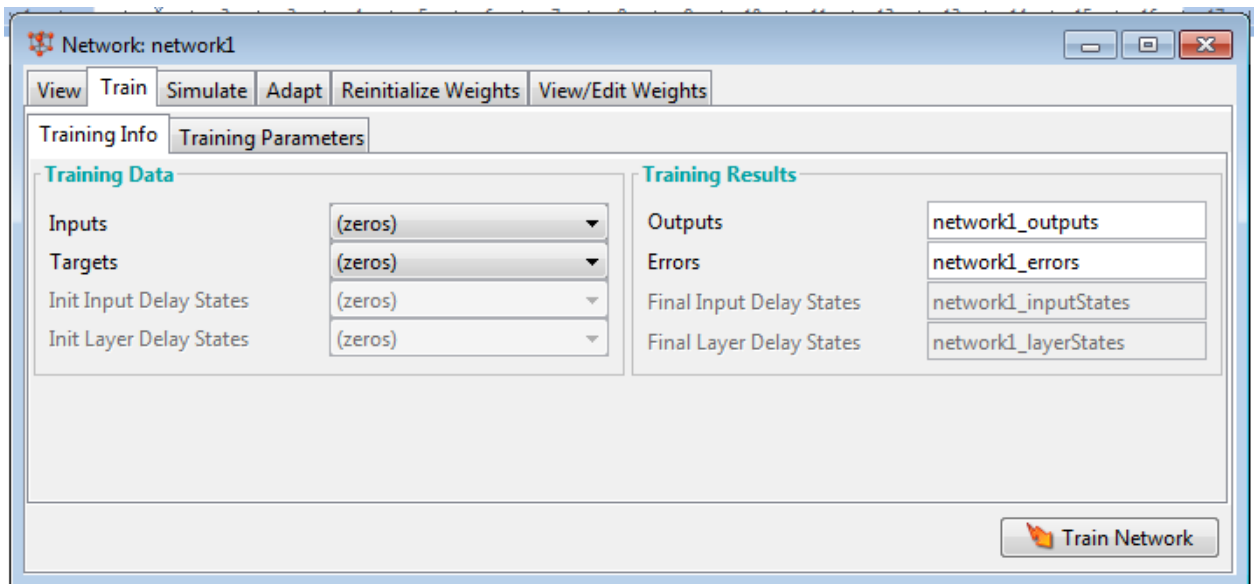


Рис. 7

Панель має дві закладки:

Training Info (Інформація про навчальні послідовності) (рис. 7);

Training Parameters (Параметри навчання) (рис. 8);

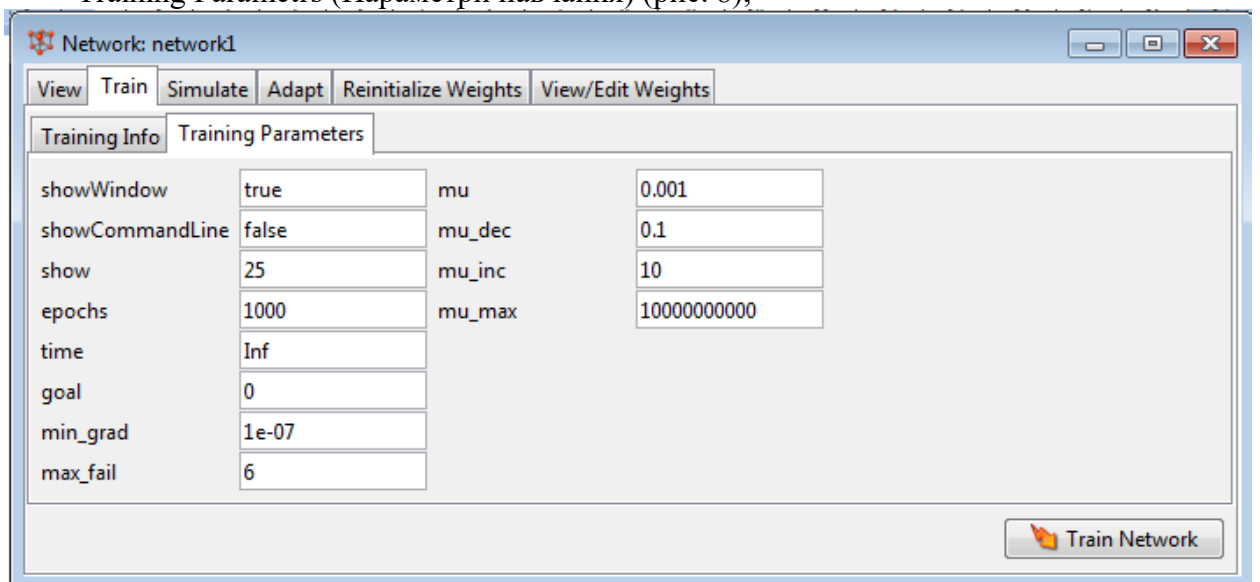


Рис. 8

Тепер можна розпочати навчання мережі (кнопка Train Network).

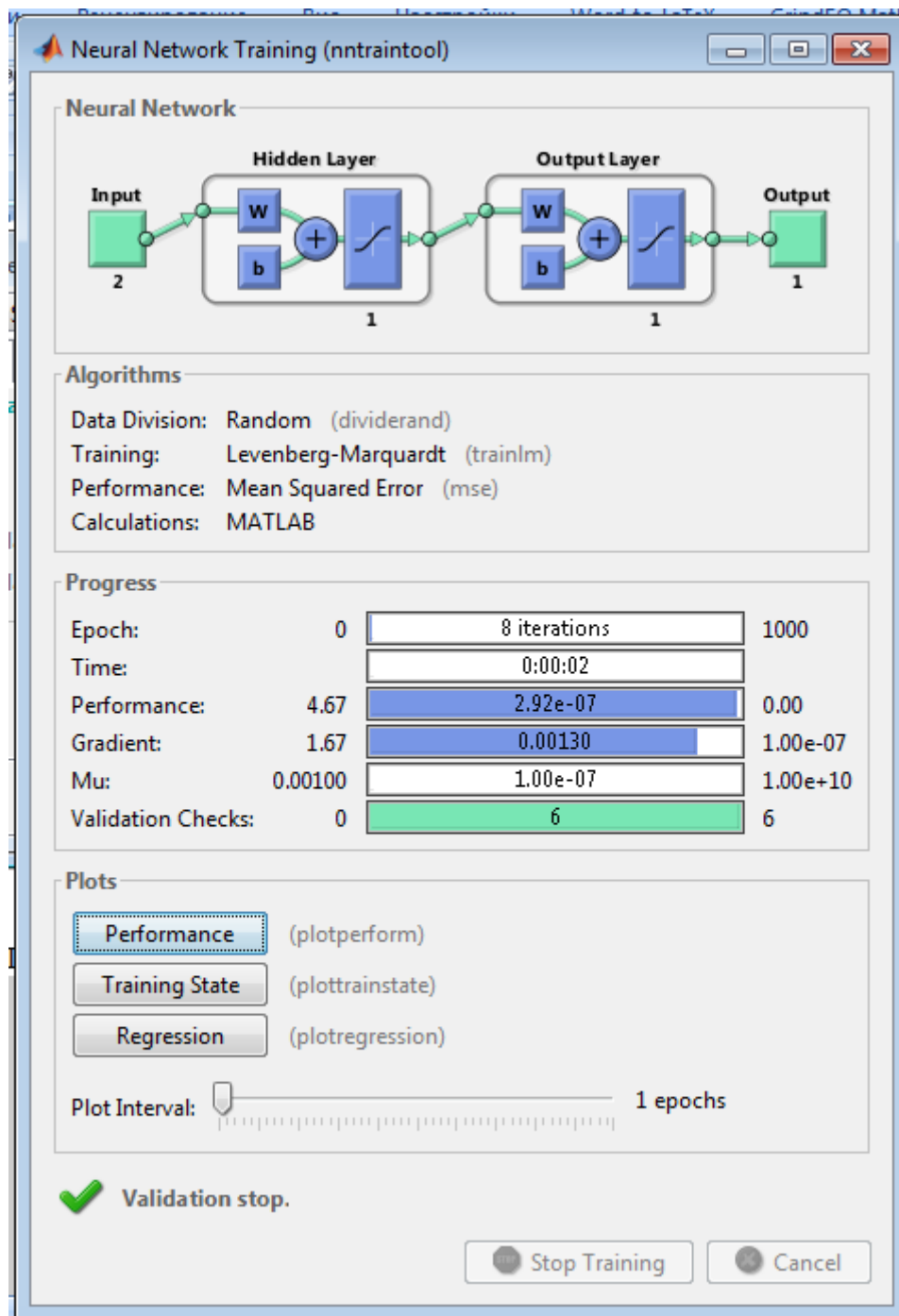


Рис. 9

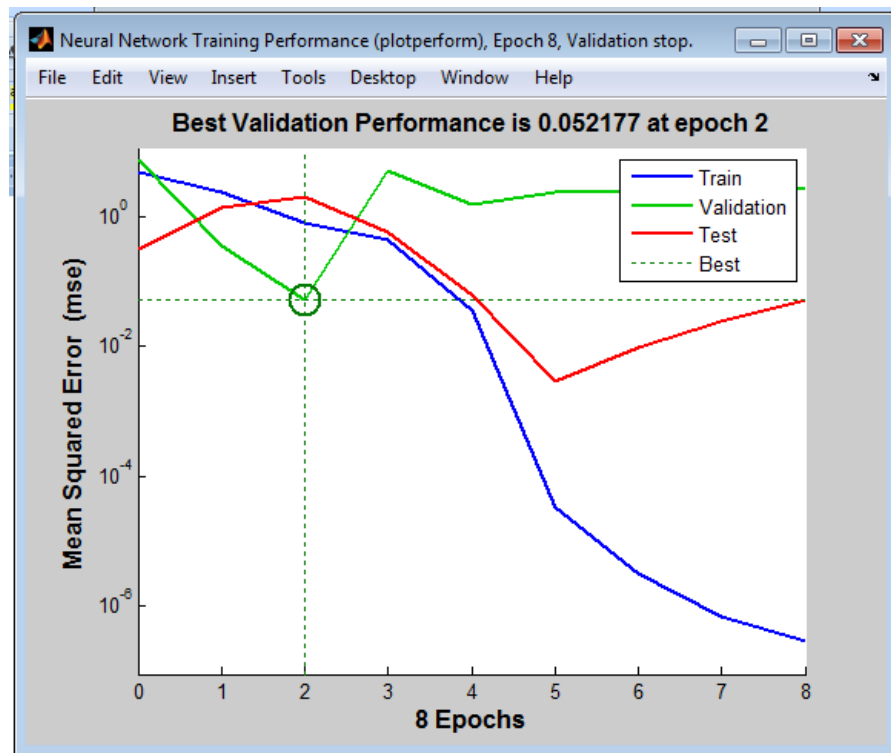


Рис. 10

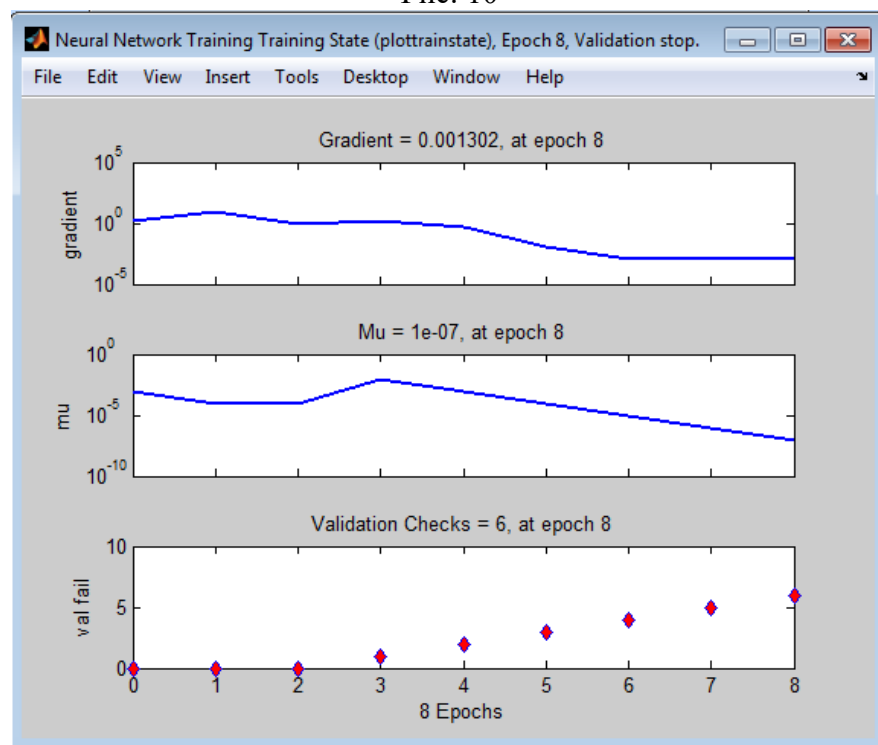


Рис. 11

Якість навчання мережі з прямою передачею сигналу на обраній навчальній послідовності показана на рис. 10 і 11. Майже нульова точність досягається за 8 циклів навчання.

Відповідні ваги та зміщення можна побачити, якщо вибрати закладку Weights (рис. 12).

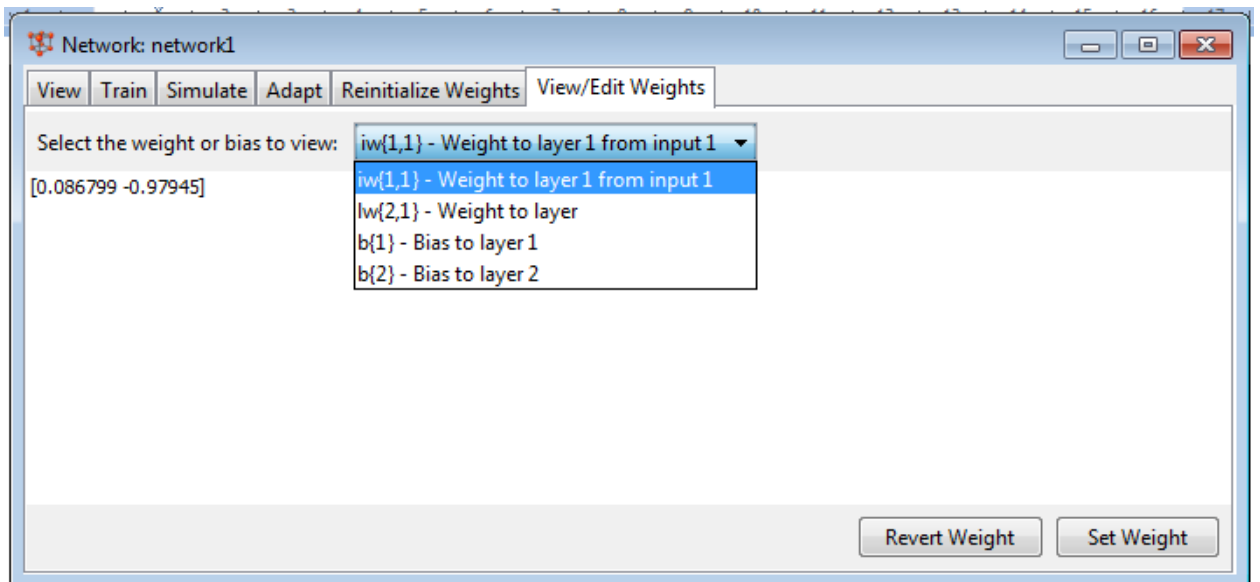


Рис. 12

Для зручності роботи можна експортувати створену нейронну мережу в робочу область системи MATLAB та отримати інформацію про ваги та зміщення безпосередньо у робочому вікні системи.

Результати навчання можна переглянути у вікні Network/Data Manager. Тепер, активізуючи імена послідовностей виходу або помилок network1\_outputs і network1\_errors, можна переглянути результати, використовуючи кнопку View (Рис. 13).

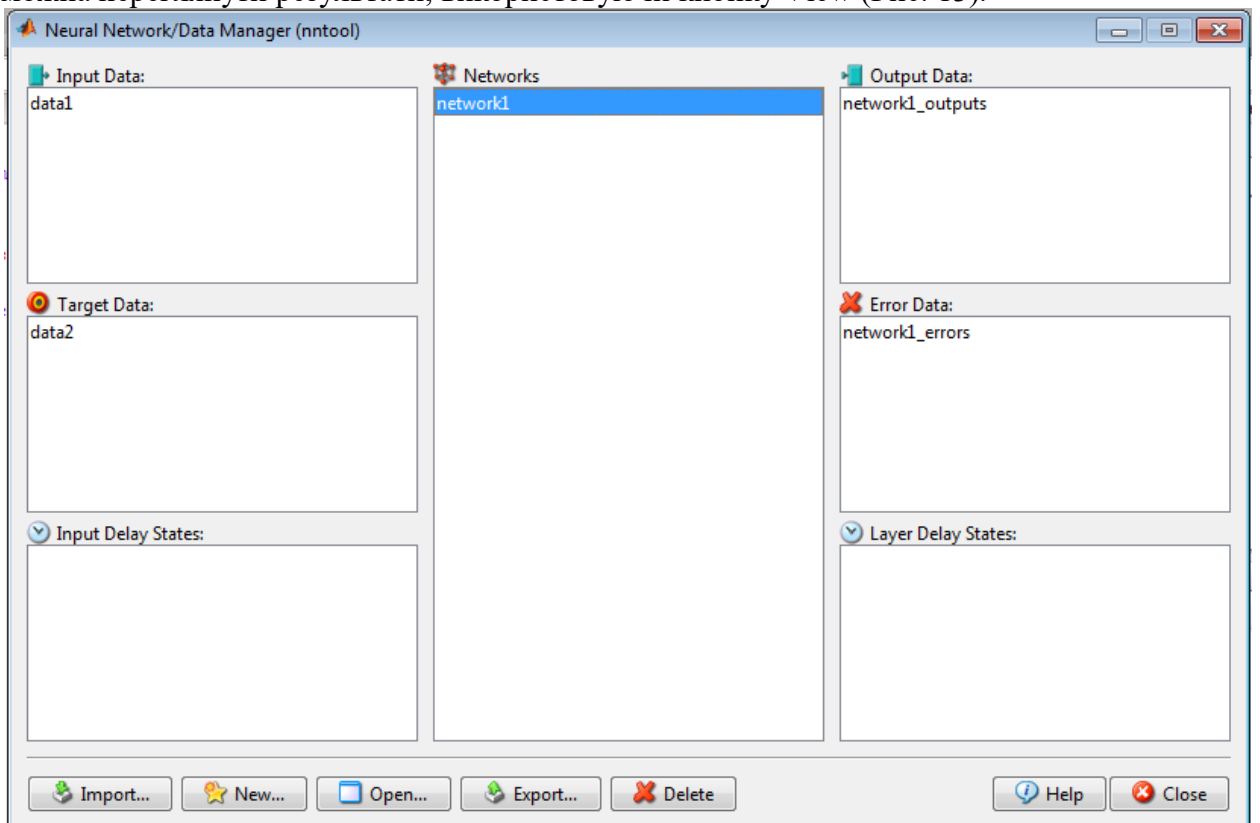


Рис. 13

#### Приклад. Класифікація входних векторів.

Завдання: Створити нейронну мережу у вигляді персептрону, яка поділяє вектори входу на два класи. Позначимо ці класи як 0 і 1. Навчальну послідовність сформуємо як два масиви комірок: масив входів  $P = \{[2; 2] [1; 2] [-2; 2] [-1; 1] [1; -2]\}$  та масив цілей  $T = \{0 0 1 1 1\}$ , який задає належність кожного вектору входу до певного класу.

Задамо персептрон з одним нейроном, функцією активації HARDLIM і правилом налаштування LEARNP. Для цього зі списку нейронних мереж виберемо тип мережі Perceptron (рис. 14) і поставимо необхідні параметри.

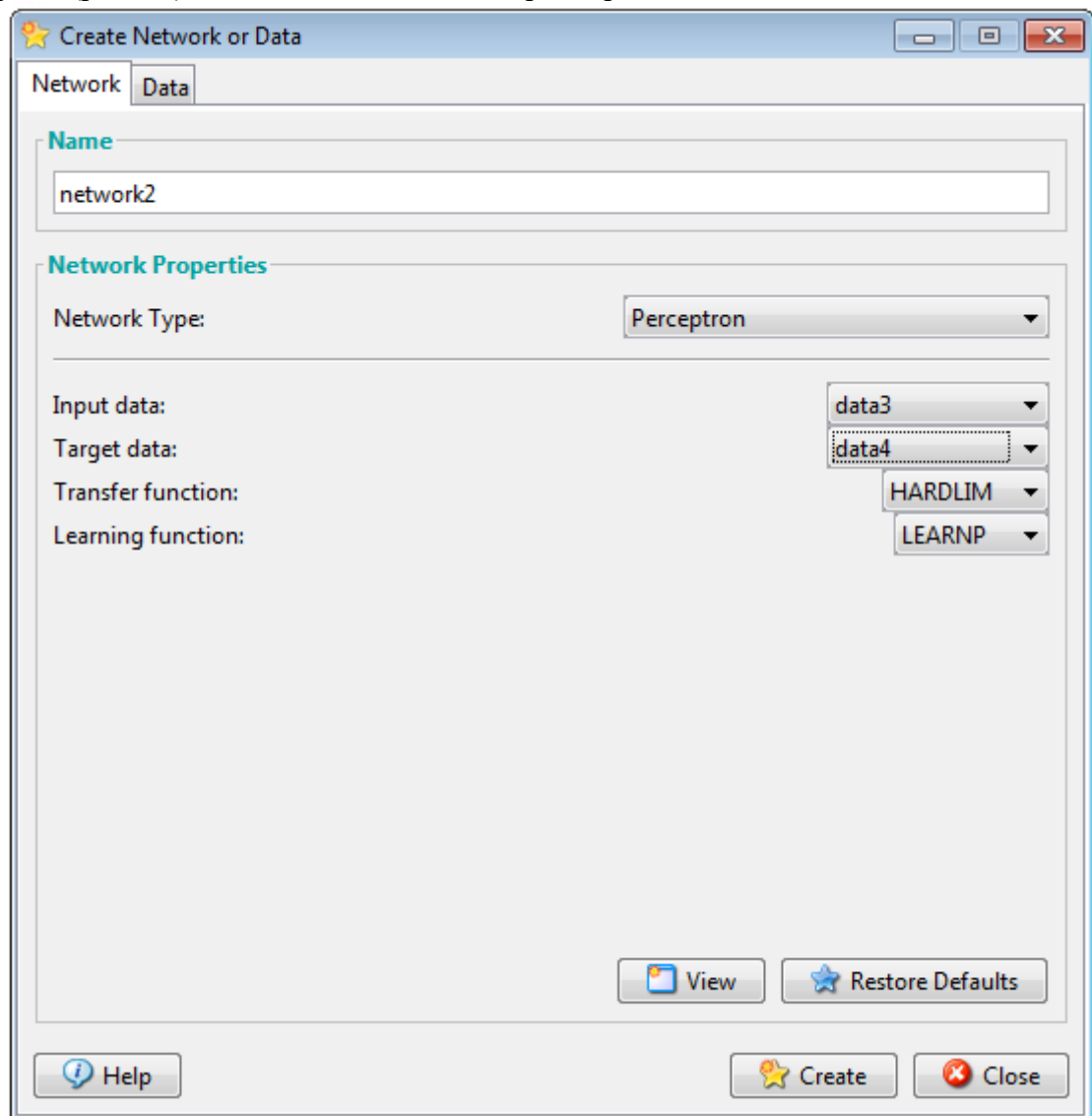


Рис. 14

Щоб побачити структурну схему мережі, скористаємося кнопкою View (рис. 15).

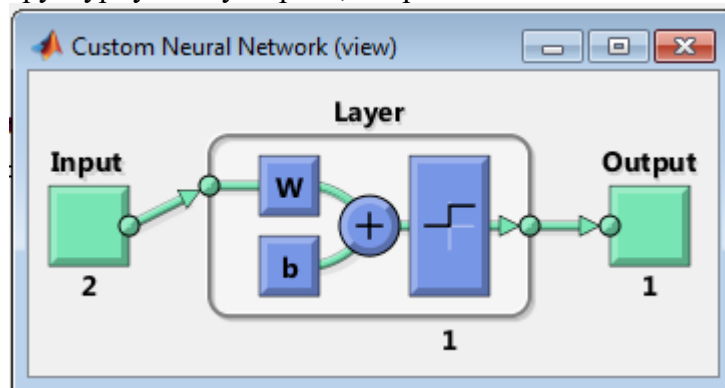


Рис. 15

Тепер виконаємо навчання та налаштування параметрів мережі, використовуючи закладку Train вікна Network. Для цього слід вказати імена входів та цілей, а також інші параметри навчання (за потреби) та натиснути кнопку Train Network.

В результаті навчання будуть встановлені наступні значення ваг та зміщень, які можна побачити, вибравши закладку Weights. Для цієї мережі вектор ваг дорівнює  $IW \{1, 1\} = [-3 \ -2]$ , а зміщення  $b \{1\} = 1$ . Таким чином, отримали пряму, що розділяє площину на дві області:

$$L: -3p_1 - 2p_2 + 1 = 0.$$

Перейшовши у вікно Network/Data Manager, можна переглянути значення сигналів на виході та помилку мережі (рис. 16,а та б відповідно). Неважко переконатися, що на навчальній послідовності мережа навчена точно класифікувати вхідні вектори.

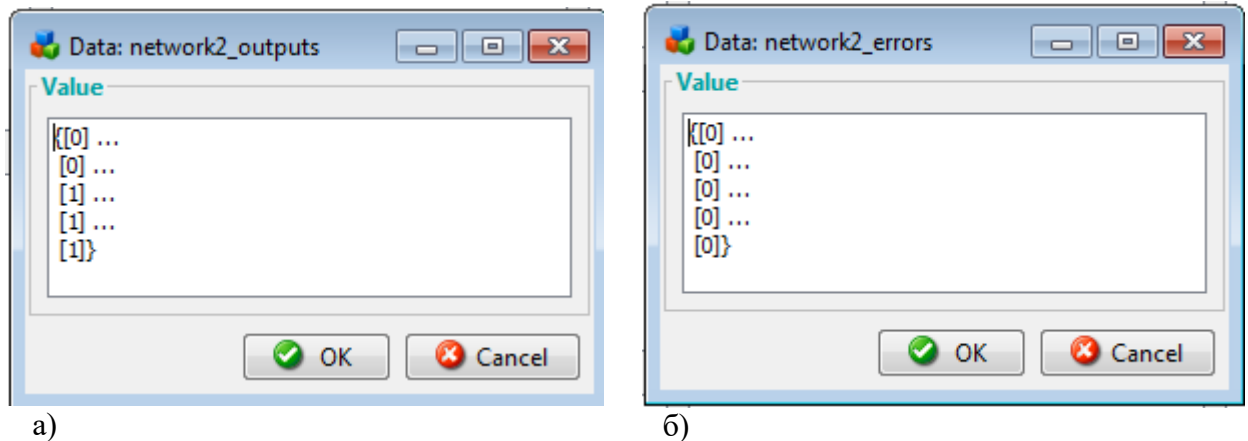


Рис. 16

Знайомство з можливостями графічного інтерфейсу NNTool дозволяє зробити висновок, що цей засіб може бути дуже корисним на початковій стадії вивчення та порівняльного аналізу нейронних мереж.

#### Демонстраційні приклади ППП NNT

Перелік всіх демонстраційних прикладів, включених до ППП NNT, можна отримати за командою `help nndemos`.

|                |  |
|----------------|--|
| Персептрони    |  |
| demop1         | Класифікація з використанням персептрону з двома входами |
| demop4         | Формування вхідних векторів зовнішнього шару             |
| demop5         | Навчання з використанням нормованої функції налаштування |
| demop6         | Приклад лінійно нероздільних векторів                    |
| Лінійні мережі |  |
| demolin1       | Приклад функціонування лінійної мережі                   |
| demolin2       | Навчання лінійного нейрона                               |
| demolin4       | Завдання лінійної апроксимації                           |
| demolin5       | Завдання з неповними даними                              |
| demolin6       | Завдання з лінійно залежними даними                      |
| demolin7       | Оцінка впливу параметра швидкості налаштування           |
| demolin8       | Адаптований лінійний шар                                 |

|   |   |
|---|---|
| Радіально-базисні мережі                                    |   |
| demorb1   | Радіально-базисні мережі                          |
| demorb3   | Приклад функцій активації, що не перекриваються   |
| demorb4   | Приклад функцій активації, що перекриваються      |
| demogrn1  | Мережа GRNN та апроксимація функцій               |
| demopnn1  | Мережа PNN та класифікація векторів               |
| Мережі кластеризації і класифікації даних                   |   |
| Самоорганізаційні мережі                                    |   |
| democ1  | Налаштування прошарку Кохонена                    |
| demosm1   | Одновимірна карта Кохонена                        |
| demosm2   | Двовимірна карта Кохонена                         |
| LVQ-мережі  |   |
| demolvq1  | Класифікація векторів                             |
| Рекурентні мережі   |   |
| Мережі Елмана   |   |
| appelm1   | Мережа Елмана                                     |
| Мережі Хопфілда   |   |
| demohop1  | Приклад двовимірної модифікованої мережі Хопфілда |
| demohop2  | Приклад нестійкої точки рівноваги                 |
| demohop3  | Приклад тривимірної модифікованої мережі Хопфілда |
| demohop4  | Приклад стійких паразитних точок рівноваги        |
| Застосування нейронних мереж                                |   |
| applin1   | Передбачення стаціонарного сигналу                |
| applin2   | Передбачення нестаціонарного сигналу              |
| appelm1   | Детектування амплітуди за допомогою мережі Елмана |
| appcr1  | Розпізнавання символів                            |
| Нейронні мережі та системи управління (середовище Simulink) |   |
| predcstr  | Управління каталітичним реактором                 |



|              |                               |
|--------------|-------------------------------|
| narmamaglev  | Управління магнітною подушкою |
| mrefrobotarm | Управління ланкою робота      |