Тема 2/Занятие 2/Упражнение

**Изграждане и обучение на изкуствени невронни мрежи в среда на MATLAB.**

MATLAB е програмна среда за автоматизиране на числени пресмятания и самостоятелен програмен език от четвърто поколение. Както подсказва името, системата MATLAB (от MATrix LABoratory“ - на български: „матрична лаборатория“) е специално създадена за извършване на матрични изчисления: решаване на системи линейни уравнения, разлагане на матрици и т.н.

Освен числен анализ софтуерната среда позволява изчертаване графики на функции, представяне на данни, програмна реализация на алгоритми, разработка на човеко-машинни интерфейси и интерфейси с други програмни продукти, написани на различни програмни езици.

**Дефиниране и обучаване на невронна мрежа с използване на вградени функции от Neural Network ToolboxTM.**

Общ изглед на работния екран

Картина, която съдържа текст

Описанието е генерирано автоматично

% Създаваме двуслойна FeedForward Neural Network (по стар стил)

Digits\_FF\_NNet = newff( minmax(M), [ S1 S2 ], { 'logsig' 'logsig' }, 'traingdx' );

%

% NET = NEWFF creates a new network NET with a dialog box.

% Obsoleted in R2010b NNET 7.0.0.

% Last used in R2010a NNET 6.0.4.

% The recommended new function is "feedforwardnet".

%

% NET = newff( PR, [S1 S2...SN ], {TF1 TF2...TFN}, BTF, BLF, PF)

% where the parameters are:

% PR - Rx2 matrix of min and max values for R input elements.

% Si - Size of i-th layer ( where i from 1 of N ).

% TFi - Transfer function of i-th layer, default = 'tansig'

% BTF - Backprop network training function, default = 'trainlm'

% BLF - Backprop weight/bias learning function, default = 'learngdm'

% PF - Performance function, default = 'mse'.

% Possible training functions are:

% 'trainlm' - Levenberg-Marquardt (Default)

% 'trainbr' - Bayesian Regularization

% 'trainbfg' - BFGS Quasi-Newton

% 'trainrp' - Resilient Backpropagation

% 'trainscg' - Scaled Conjugate Gradient

% 'traincgb' - Conjugate Gradient with Powell/Beale Restarts

% 'traincgf' - Fletcher-Powell Conjugate Gradient

% 'traincgp' - Polak-Ribiere Conjugate Gradient

% 'trainoss' - One Step Secant

% 'traingdx' - Variable Learning Rate Gradient Descent

% 'traingdm' - Gradient Descent with Momentum

% 'traingd' - Gradient Descent

%

%

% To create a comon shallow NNet we can use:

% NNet = network( NumInputs, NumLayers, BiasConnect, ...

% InputConnect, LayerConnect, OutputConnect )

% where parameters are:

% NumInputs - Number of inputs, 0

% NumLayers - Number of layers, 0

% BiasConnect - NumLayers-by-1 Boolean vector, zeros

% InputConnect - NumLayers-by-NumInputs Boolean matrix, zeros

% LayerConnect - NumLayers-by-NumLayers Boolean matrix, zeros

% OutputConnect - 1-by-NumLayers Boolean vector, zeros

%

% Example how to create a One-input, Two-layer, FeedForward network:

% net = network( 1, 2, [1; 0], [1; 0], [0 0; 1 0], [0 1] )

%

% Here only Layer 1 has a bias.

% An input weight connects to layer 1 from input 1.

% A layer weight connects to layer 2 from layer 1.

% Layer 2 is a network output and has a target.

%

%

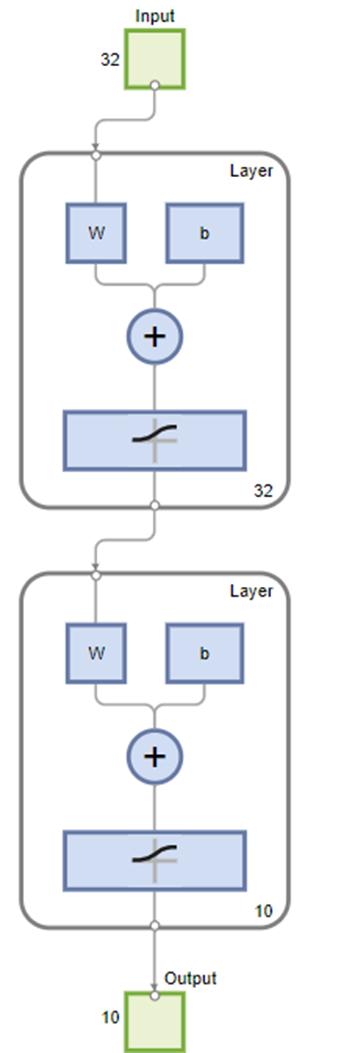
% GUI NNet Interface:

% nnstart (old command was "nntool")

% Можем да визуализираме невроннат мрежа:

view( Digits\_FF\_NNet )

Архитектурата на изградената мрежа има вида:



Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5-03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0-13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531