



Neurális Hálózatok I.

Intelligens Rendszerek
Gyakorlat



1. feladat: másodfokú függvény

Készítsünk egy olyan neurális hálózatot, amely az alábbi átviteli függvényt valósítja meg a

$-10 \leq x \leq 10$ intervallumban:

$$Y = 2x^2 + 5x - 10$$



- Matlabban készítsük el az x intervallumot:

$x = -10:10$

- Melynek eredménye a következő sorozat:

$x =$

-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	
0	1	2	3	4	5	6	7	8	9	10



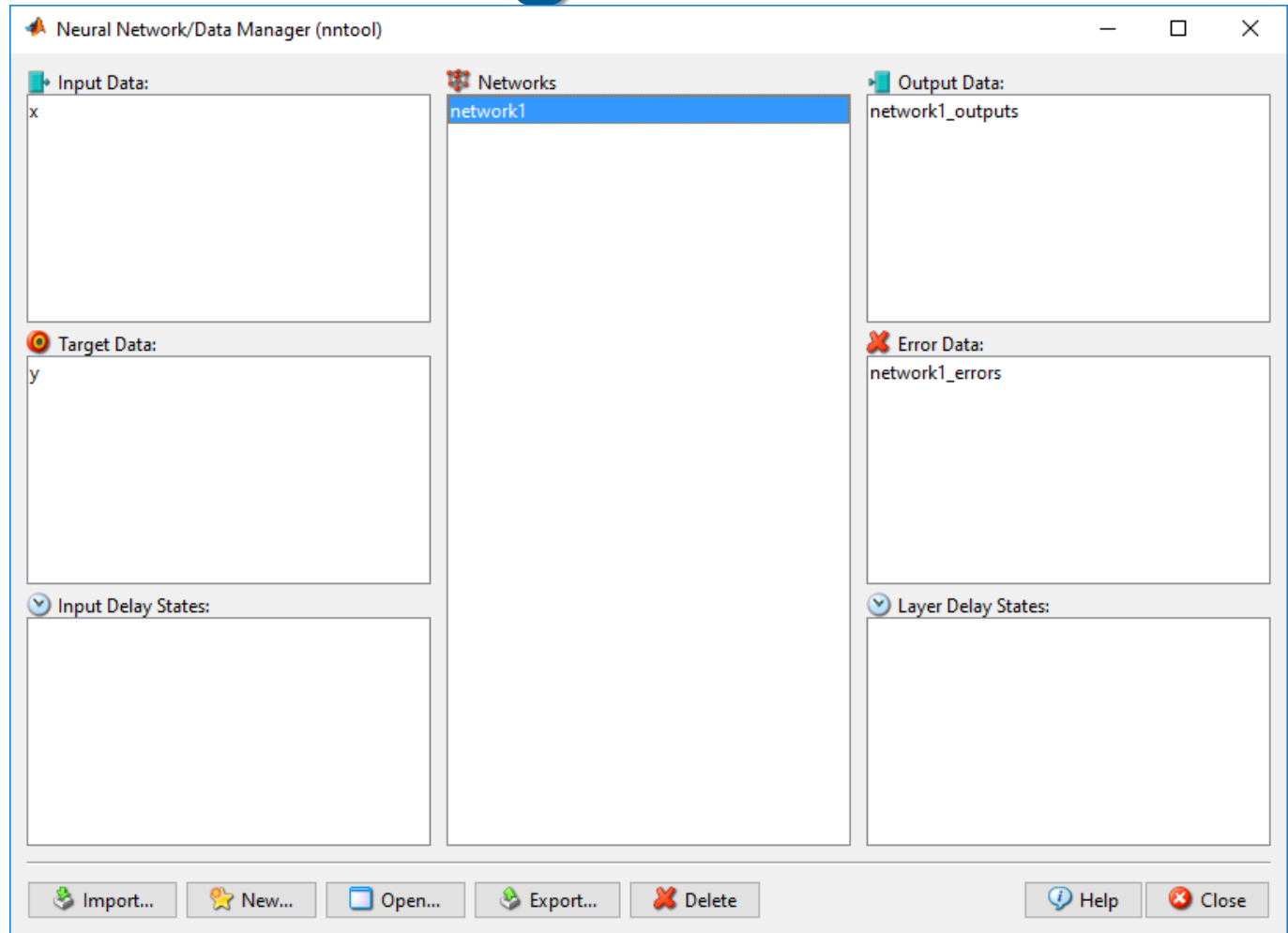
- Készítsük el a tanító mintákat is:

$$y=2 \cdot x \cdot x+5 \cdot x-10$$



Neural Network/Data Manager

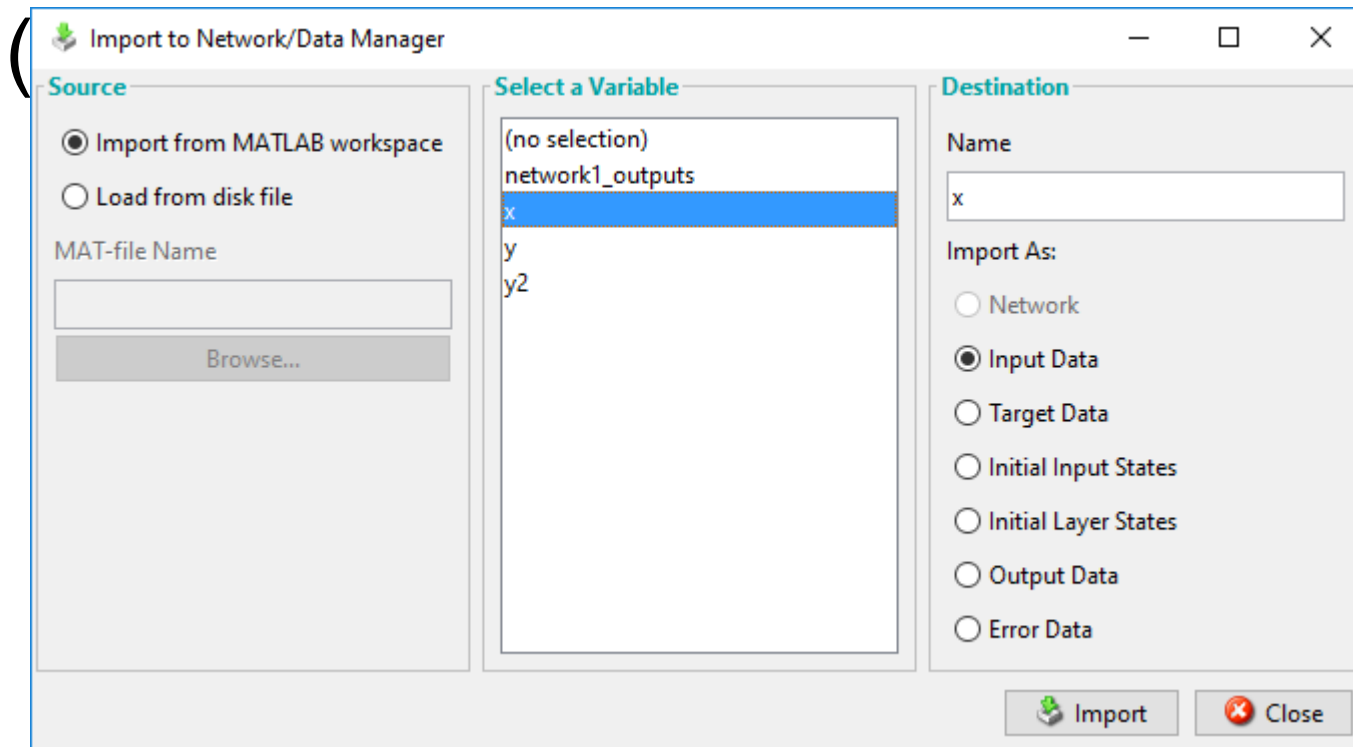
- nntool





Neural Network/Data Manager

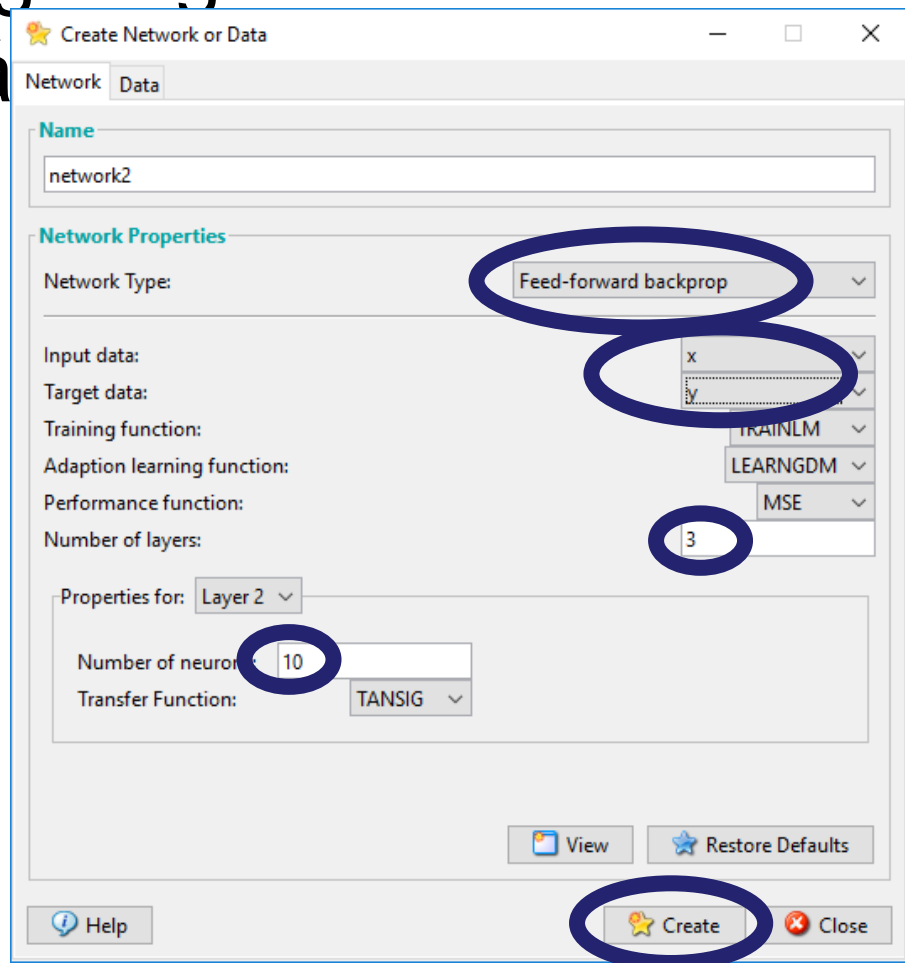
- Az „Import” gombbal töltjük be x-et bemeneti adatsorként, y-t célértékként





Neural Network/Data Manager

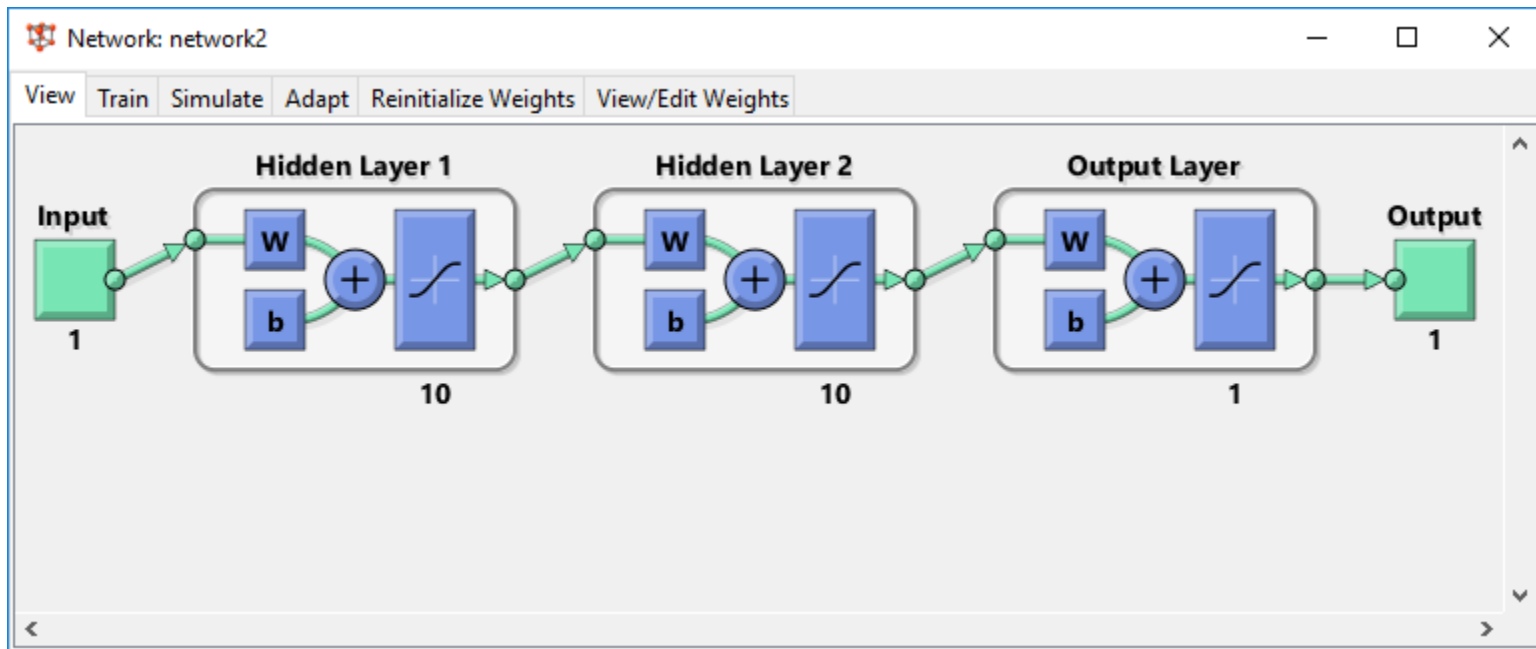
- A „new” gomb segítségével készítsünk egy új neurális hálózatot
- Lehetőség van többek között a tanítás módjának, illetve a rétegek és neuronok számának módosítására valamint a hálózat topológiájának változtatására





Tanítás

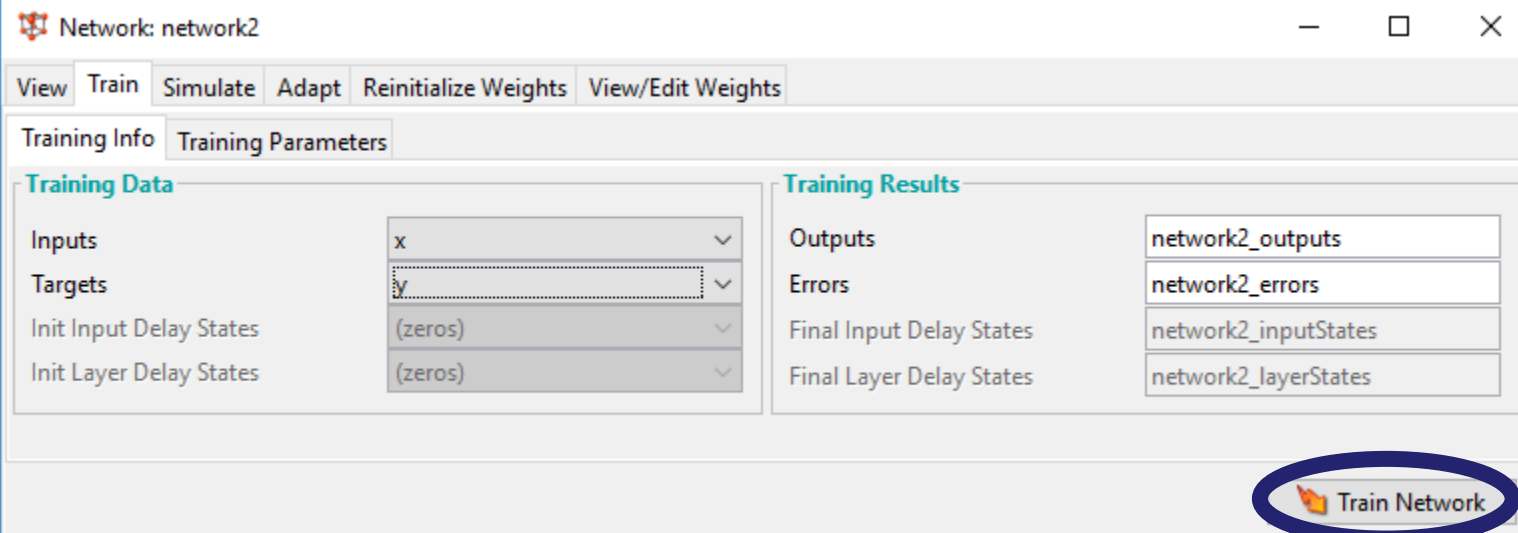
- Kattintsunk a létrehozott új hálózatra





Tanítás

- A tanítás fülön válasszuk ki a bemeneti adatsort és a tanítómintát (x és y).
- networkN_outouts és networkN_errors lesz a hálózat kimenete és hibája

- 

Network: network2

View Train Simulate Adapt Reinitialize Weights View/Edit Weights

Training Info Training Parameters

Training Data

Inputs	x
Targets	y
Init Input Delay States	(zeros)
Init Layer Delay States	(zeros)

Training Results

Outputs	network2_outouts
Errors	network2_errors
Final Input Delay States	network2_inputStates
Final Layer Delay States	network2_layerStates

Train Network



- A tanítást követően láthatjuk annak részleteit, idejét, jóságát

Neural Network Training (nntraintool)

Neural Network

Input: 1

Hidden Layer 1: W, b, 10

Hidden Layer 2: W, b, 10

Output Layer: W, b, 1

Output: 1

Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	10 iterations	1000
Time:		0:00:00	
Performance:	1.29e+04	0.00544	0.00
Gradient:	1.11e+04	6.74	1.00e-07
Mu:	0.00100	0.0100	1.00e+10
Validation Checks:	0	6	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

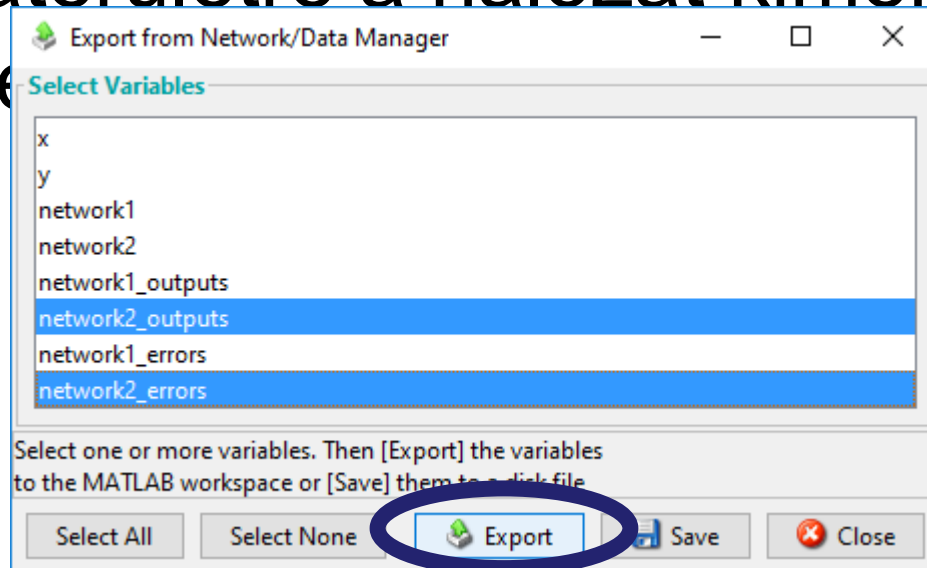
✓ Validation stop.

Stop Training Cancel



Hálózat kimenetének ábrázolása

- Térjünk vissza a „Neural Network/Data Manager” ablakba
- Az „export” funkcióval exportáljuk a munkaterületre a hálózat kimenetét és kimenetét



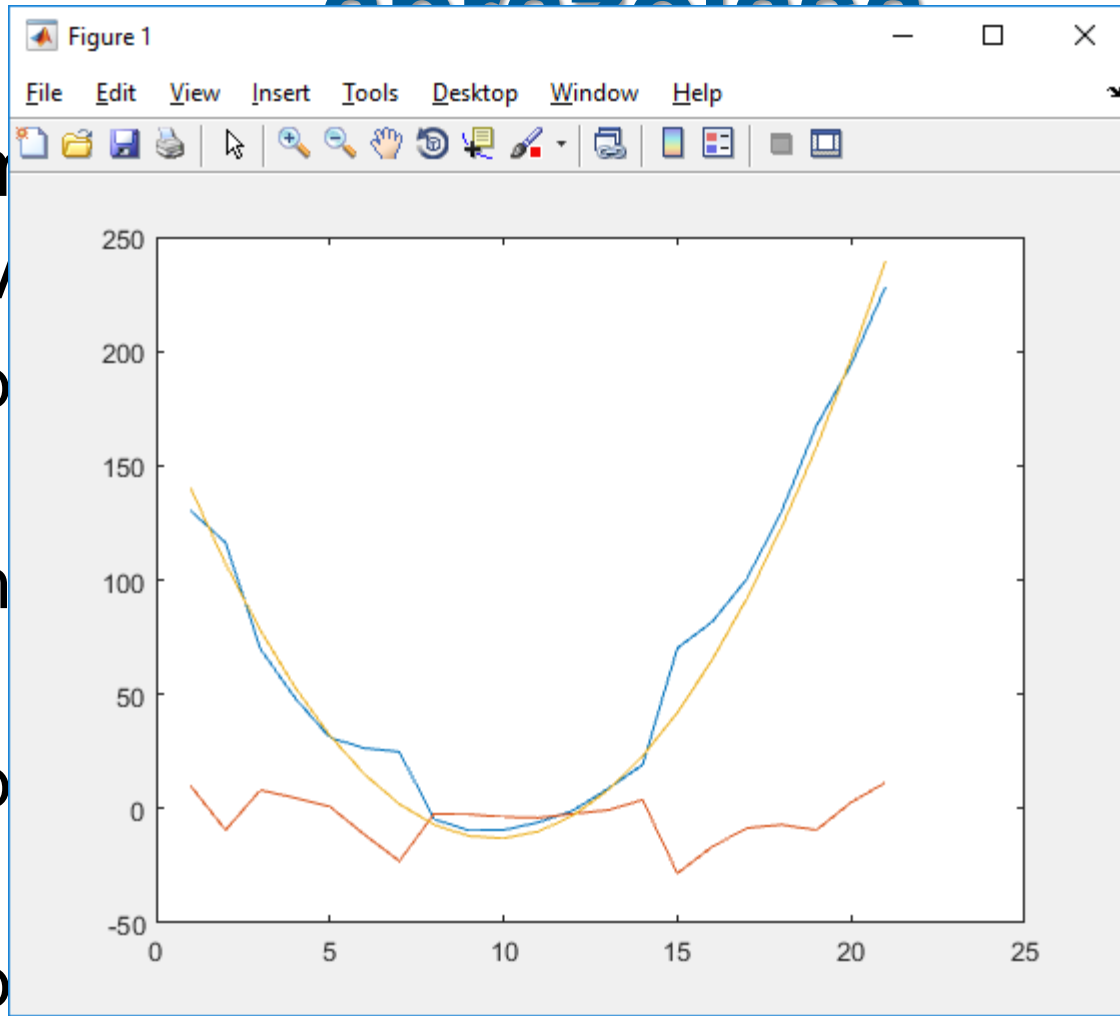


Hálózat kimenetének ábrázolása

- A matlab főablakba gépeljük a következőket:
 - `plot(network2_outputs)`
 - % hálózat kimenete
 - `hold on`
 - % ábra megtartása
 - `plot(network2_errors)`
 - % hálózat hibája (tanítómintától való eltérése)
 - `plot(network2_outputs+network2_errors)`
 - % kimenet+hiba=tanítóminta



Hálózat kimenetének ábrázolása



- A n
- köv
- p
- h
- p
- p

Itérése)
rrors)

• % kimenet+hiba=tanítóminta



Feladat

- A korábbi órákról megismert feladatok megoldása Matlab segítségével:
- Szabadesés: ejtési magasság 0:20m
- Ferdehajítás: dobás 30° -ban felfelé, kezdősebesség 0:30 m/s
- *Figyelem! A bemutatott módon csak azonos dimenziójú ki és bemenetet tudunk kezelni a neurális hálózattal!*