Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky
Katedra elektrotechniky a mechatroniky

MODELOVANIE NELINEÁRNEJ FUNKCIE NEURÓNOVOU SIEŤOU

Bc. Andrej Klein 2023/2024

Obsah

Zadanie:

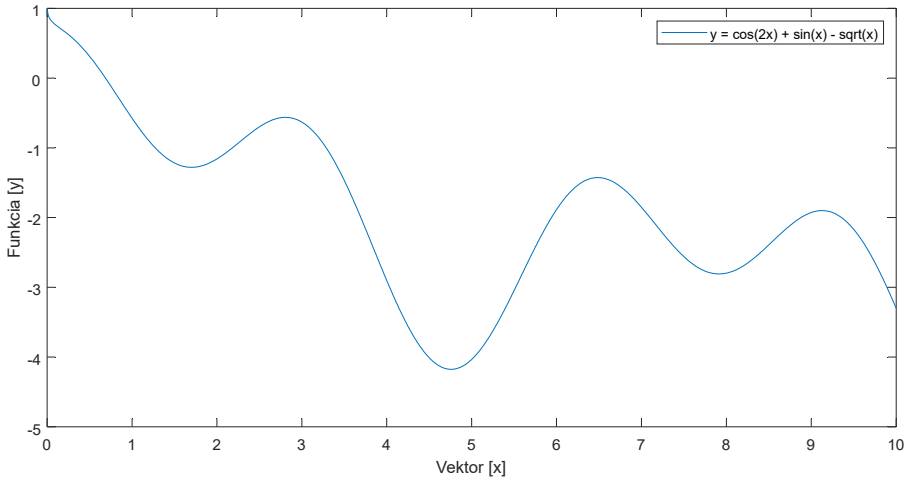
Navrhnite neurónovú sieť na opis vami zvolenej funkcie. Nájdite takú neurónovú sieť, ktorá dokáže opísať danú funkciu s čo najmenším počtom neurónov v skrytej vrstve. Porovnajte výsledky s väčším počtom neurónov v skrytej vrstve.

1. Vytvorenie nelineárnej funkcie

Zvolíme si ľubovoľnú nelineárnu funkciu, ktorá bude mať nasledovný tvar:

[1]

Prvým krokom je zadefinovanie vstupných a výstupných premenných, ktoré sú cieľové. Funkcia bude vykreslená na intervale [0; 10] čo predstavuje vektor x. Krok medzi intervalmi bude 0,01 čo nám zaručí dostatočne hladký priebeh funkcie a dostatok dát pre učenie neurónovej siete. Následne si našu funkciu vykreslíme pomocou programu MATLAB. Na nasledujúcom obrázku môžeme vidieť priebeh našej nelineárnej funkcie.

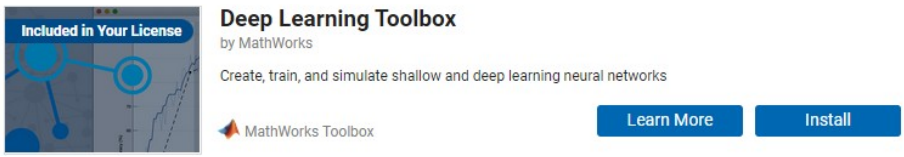


Obr. 1 Želaná spojitá nelineárna funkcia

V ďalšej časti nasleduje krok v ktorom sa určia základne nastavenia neurónovej siete.

2. Vytvorenie neurónovej siete v staršej verzii MATLABu

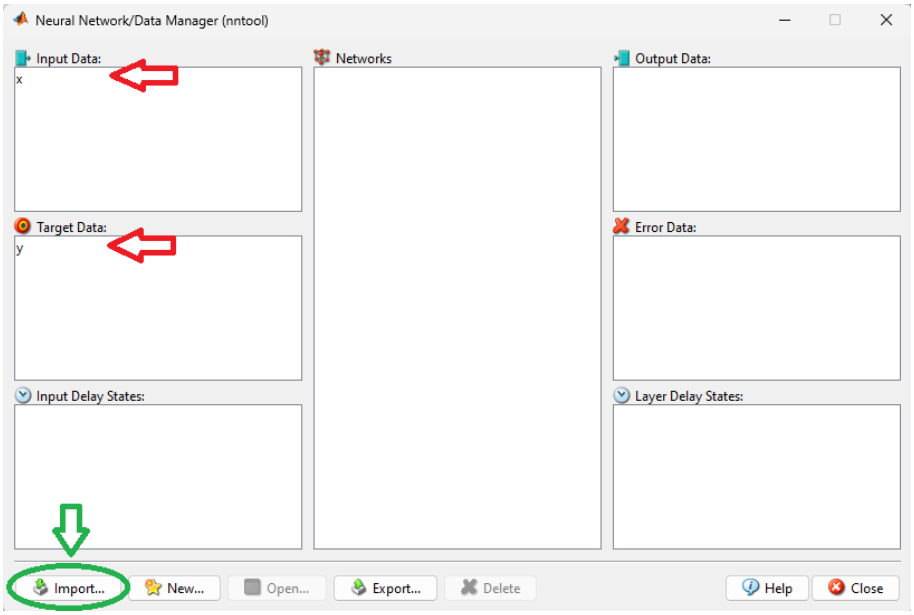
Nastavenia neurónovej siete a celý návrh uskutočnime v staršej verzii MATLABu R2020b. Pre funkčnosť je potrebné doinštalovať prídavný balík „Deep Learning Toolbox“.



Obr. 2 Deep Learning Toolbox

Po nainštalovaní otvoríme rozhranie neurónovej siete príkazom „>> nntool“.
Vytvoríme si neurónovú sieť s jedným neurónom vo výstupnej vrstve a s niekoľkými neurónmi v skrytej vrstve.

Importujeme z Workspace-u v MATLABe vektor x a funkciu y.

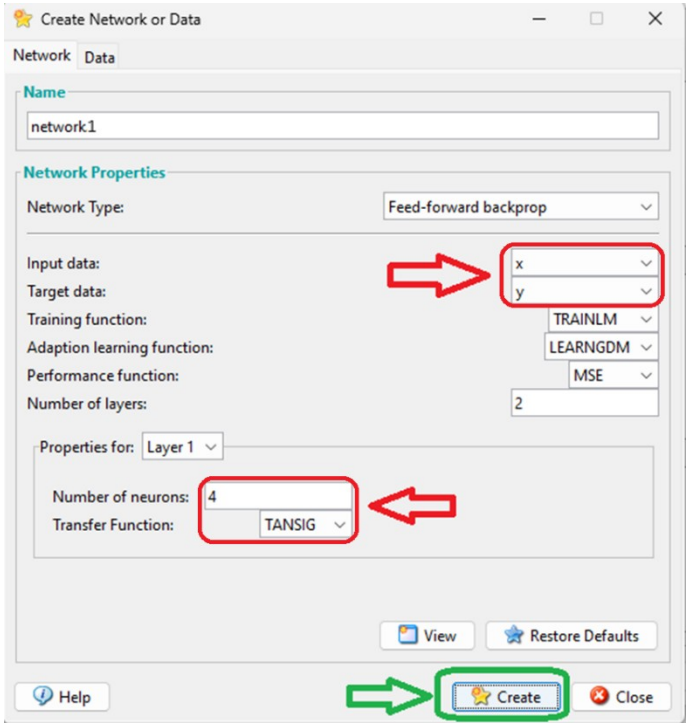


Obr. 3 Import dát z workspace-u

Po úspešnom importovaní vektora „x“ a funkcie „y“ vytvoríme neurónovú sieť, ktorá sa volá network.

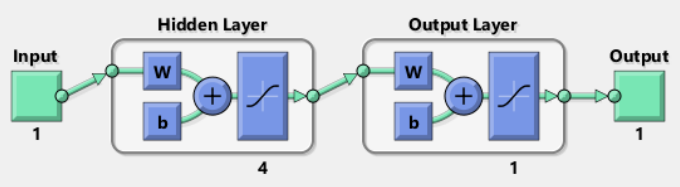
2.1 Neurónová sieť network 1

Zadefinujeme vstupné dáta čo predstavuje náš vektor „x“ a cieľovú funkciu „y“. Ďalej zadefinujeme počet neurónov a typ funkcie. Zvolíme skrytú vrstvu, ktorá bude mať 4 neuróny a bude typu TANSIG. Výstupná vrstva bude tiež typu TANSIG.



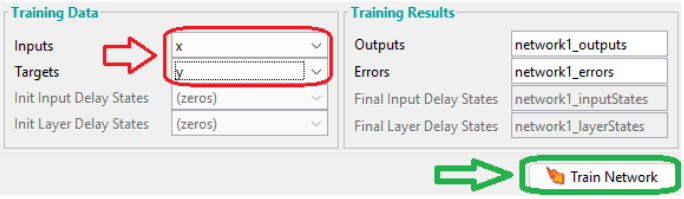
Obr. 4 Vytvorenie neurónovej siete „Network1“

Vygenerovala sa nám štruktúra neurónovej siete, ktorá je zobrazená nižšie. Táto štruktúra ako vidíme obsahuje dve vrstvy skrytú a výstupnú.



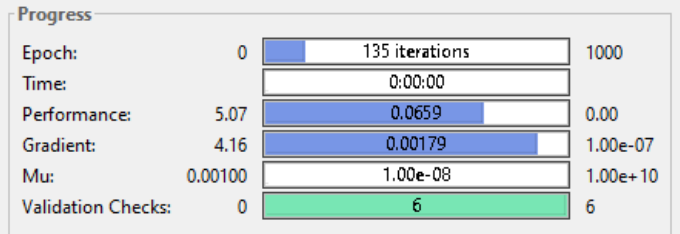
Obr.5 Štruktúra neurónovej siete

Následne zadefinujeme náš vektor „x“ a funkciu „y“ v učiacej časti (Training Data) a dáme Train Network, tým naučíme našu sieť.



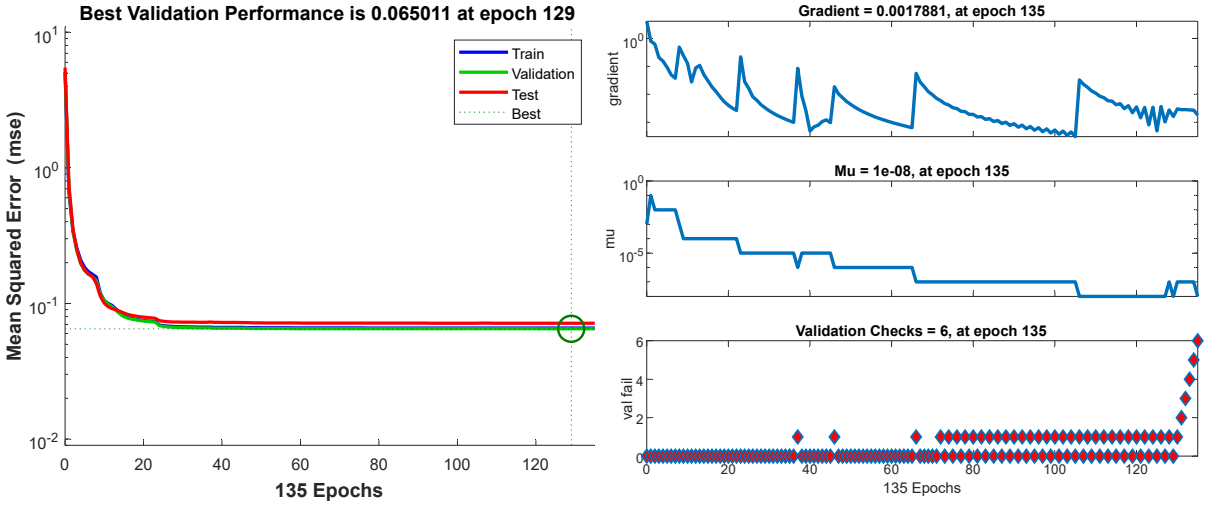
Obr. 6 Učenie dát

V časti „Progress“ môžeme vidieť aktuálny stav učenia, typ trénovanej funkcie a veľkosť chyby.



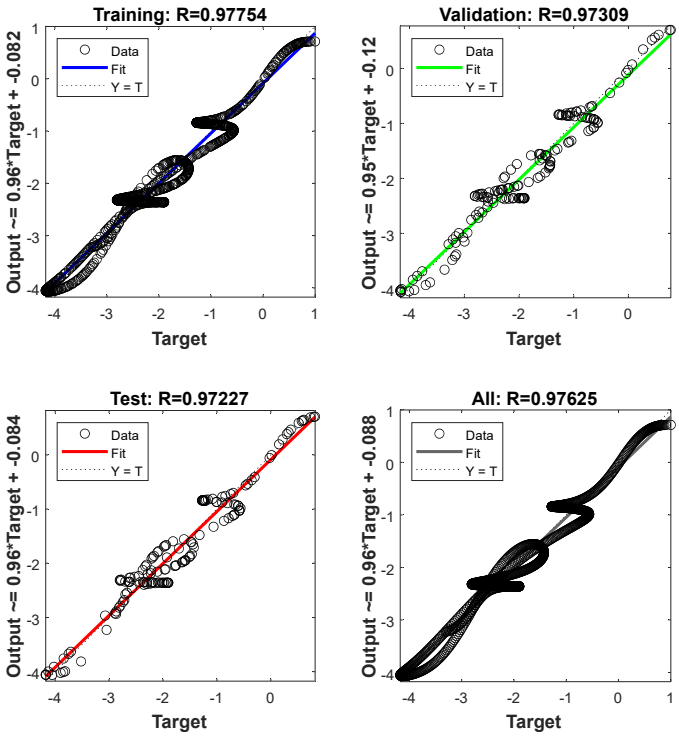
Obr. 7 Proces učenia sa neurónovej siete

Po naučení sa neurónovej sieti vieme zobraziť výsledne charakteristiky našej funkcie, ktoré sú zobrazené nižšie.



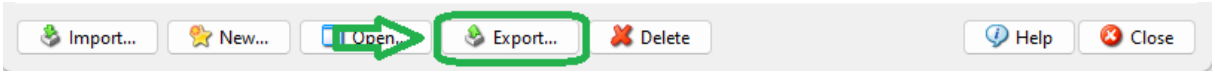
Obr. 8 Výstupné charakteristiky naučenej neurónovej siete

Regresia neurónovej siete po natrénovaní je zobrazená na nasledujúcom obrázku. Tu si môžeme všimnúť, že naša simulácia nie je dostatočne odladená.



Obr. 9 Regresia siete po natrénovaní

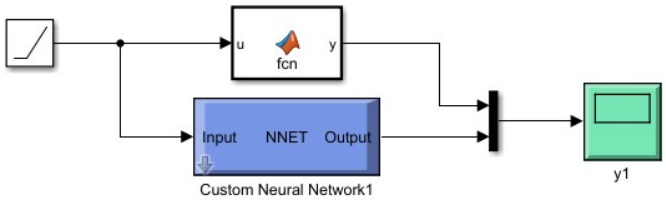
Pre overenie našej simulácie si túto neurónovú sieť exportujeme do workspace-u v MATLABe v nasledujúcom kroku: „export from Network/Data Manager“.



Následne použijeme príkaz >> gensim(network1), tento príkaz nám vytvorí blokovú schému v Simulinku. Túto schému doplníme o funkčný blok, kde zadefinujeme našu funkciu v tvare:

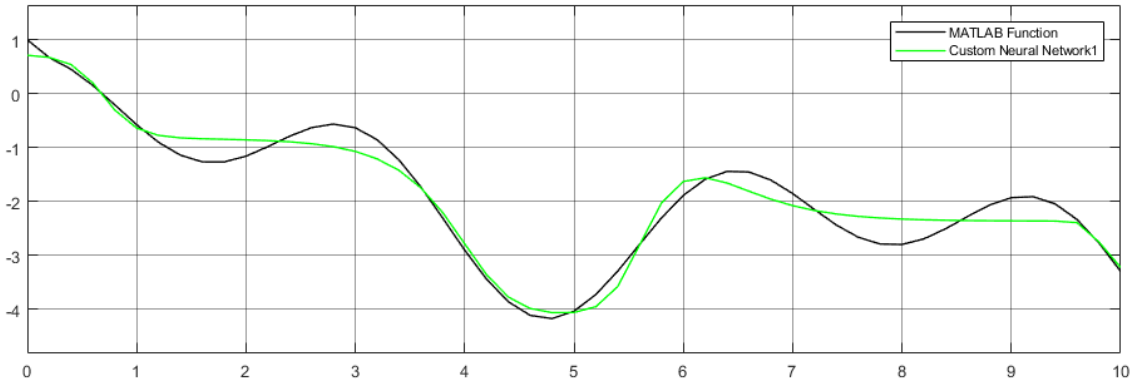
```
y = cos(2*u(1)) + sin(u(1)) - sqrt(u(1));
```

Ďalej zmeníme vstupnú hodnotu na „Ramp“. Teraz vieme odsimulovať našu neurónovú sieť a porovnať ju s konkrétnou nami zvolenou funkciou podľa zadania. Na nasledujúcom obrázku je zobrazená bloková schéma našej simulácie vygenerovaná v programe Simulink.



Obr. 10 Bloková schéma simulácie neurónovej siete

Na nasledujúcom obrázku je znázornený výsledný priebeh nami vytvorenej neurónovej siete.

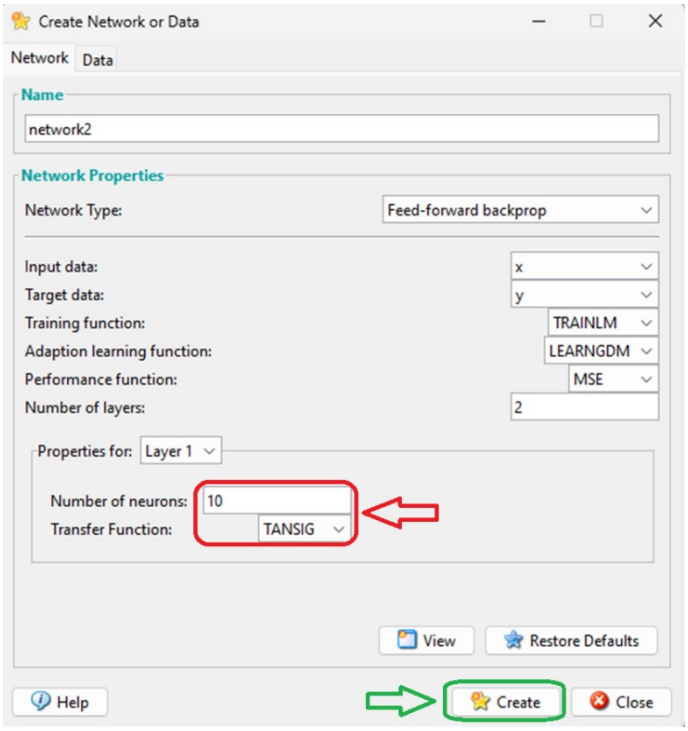


Obr. 11 Výsledný priebeh nami vytvorenej neurónovej siete

Ako môžeme vidieť neurónová sieť vyžaduje viac neurónov, 4 neuróny sú najmenšie množstvo pri ktorej sa vie dotiahnuť, ale priebeh je nedostatočný.

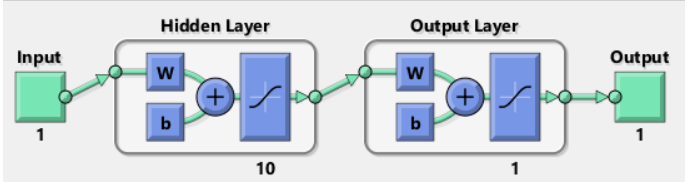
2.2 Neurónová sieť network 2

Pre lepšie vlastnosti upravíme počet neurónov. Táto neurónová sieť už ako vieme pozostáva z dvoch vrstiev, kde skrytá vrstva má už 10 neurónov a je typu TANSIG. Výstupná vrstva ostáva typu TANSIG.



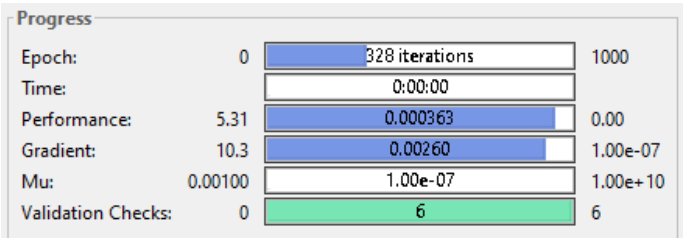
Obr. 12 Vytvorenie neurónovej siete „Network2“

Vygenerovala sa nám štruktúra neurónovej siete, ktorá je zobrazená nižšie. Táto štruktúra ako vidíme má už zmenený počet neurónov na 10.



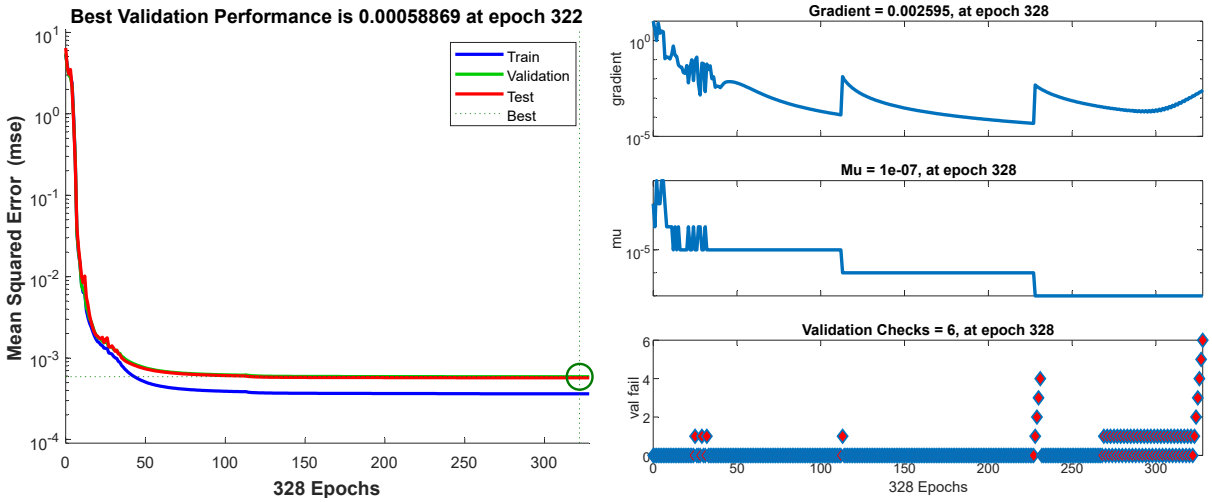
Obr.13 Štruktúra neurónovej siete

Opäť dáme vytrénovať našu sieť ako sme ukázali vyššie v predošlom riešení.
V časti „Progress“ môžeme vidieť aktuálny stav učenia, typ trénovanej funkcie a veľkosť chyby.



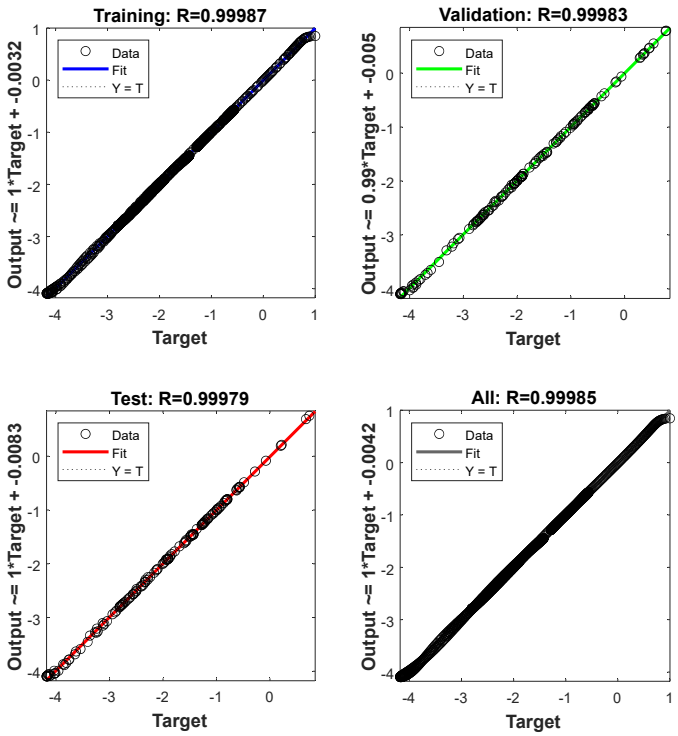
Obr. 14 Proces učenia sa neurónovej siete

Po naučení sa neurónovej sieti opäť zobrazíme výsledné charakteristiky našej funkcie, ktoré sú zobrazené nižšie.



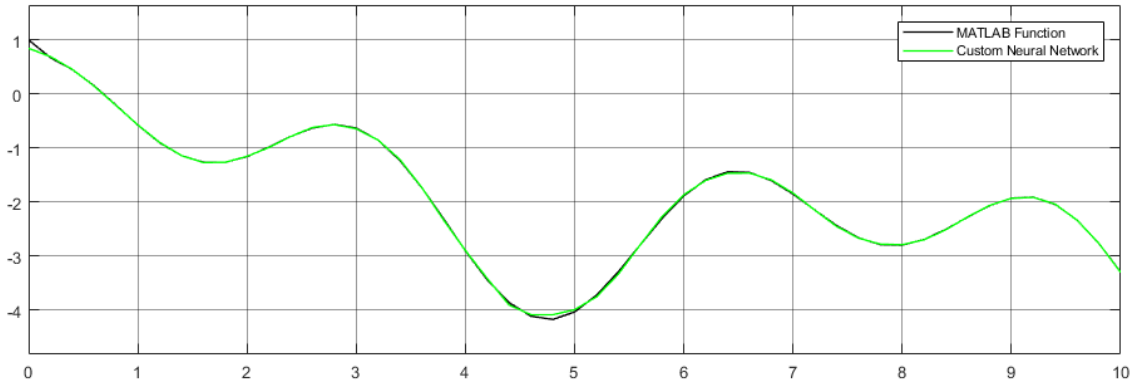
Obr. 15 Výstupné charakteristiky naučenej neurónovej siete

Regresia neurónovej siete po natrénovaní je zobrazená na nasledujúcom obrázku. Tu si môžeme všimnúť, že naša simulácia je už dostatočne odladená.



Obr. 16 Regresia siete po natrénovaní

Opäť vyexportujeme dáta a získame rovnakú blokovú schému, ako je uvedené v predošlom riešení (obr. 10) a danú simuláciu zopakujeme . Na nasledujúcom obrázku je znázornený výsledný priebeh nami vytvorenej neurónovej siete.

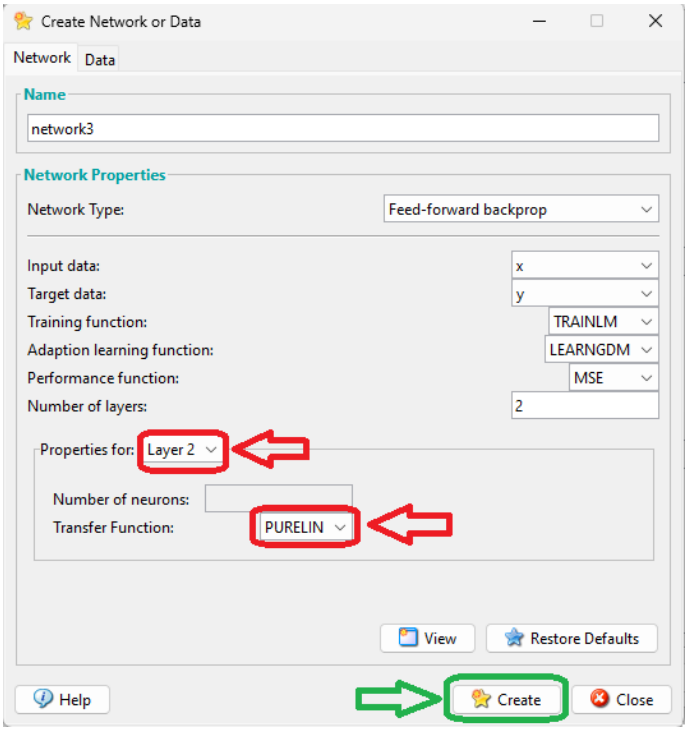


Obr. 17 Výsledný priebeh nami vytvorenej neurónovej siete

Tu vidíme, že počet neurónov je vyhovujúci, ale neurónová sieť vyžaduje doladenie pomocou zmeny funkcie vrstiev.

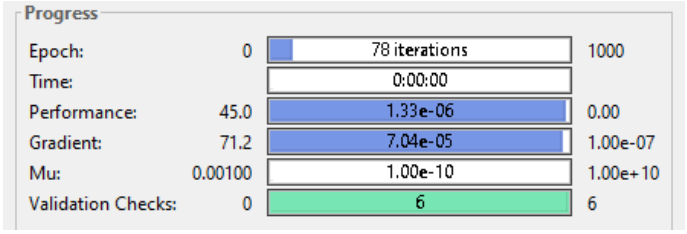
2.3 Neurónová sieť network 3

Výstupnú vrstvu upravíme, tak že zmeníme funkciu na “PURELIN“. Počet neurónov a funkciu skrytej vrstvy nemeníme.



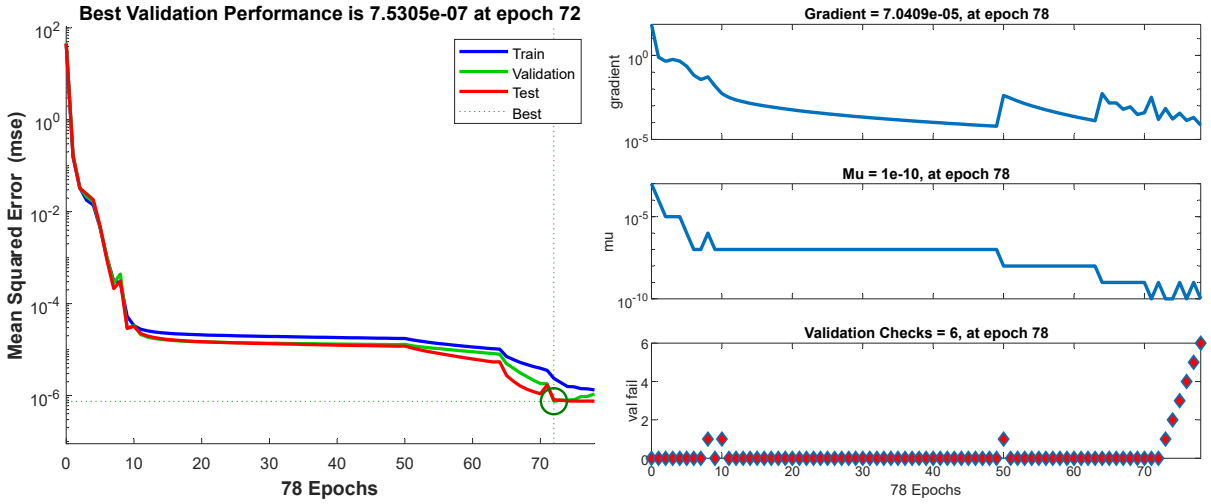
Obr. 18 Vytvorenie neurónovej siete „Network3“

Štruktúra neurónovej siete ostáva bez zmeny, ako je uvedená v predošlom riešení (obr. 13). Opäť dáme vytrénovať našu sieť, ako sme ukázali vyššie v predošlom riešení. V časti „Progress“ môžeme vidieť aktuálny stav učenia, typ trénovanej funkcie a veľkosť chyby.



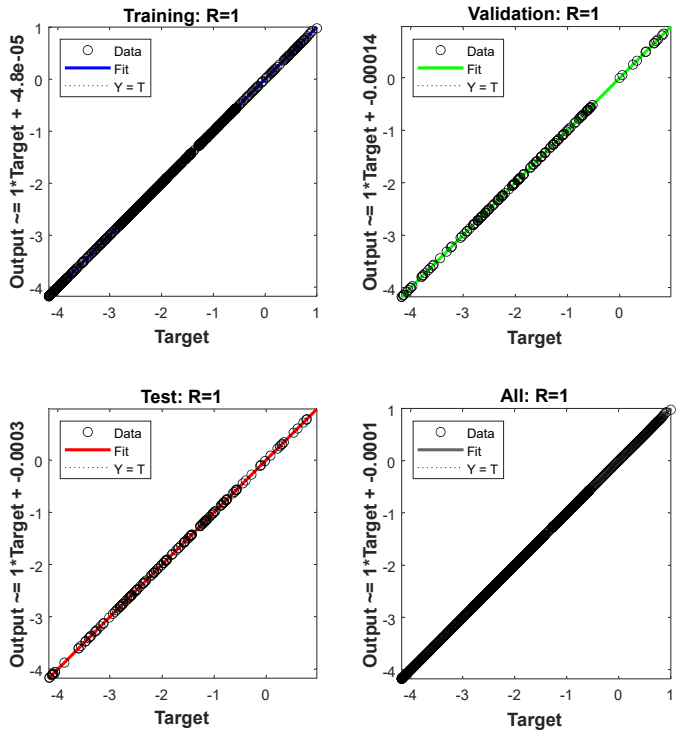
Obr. 19 Proces učenia sa neurónovej siete

Po naučení sa neurónovej sieti opäť zobrazíme výsledné charakteristiky našej funkcie, ktoré sú zobrazené nižšie.



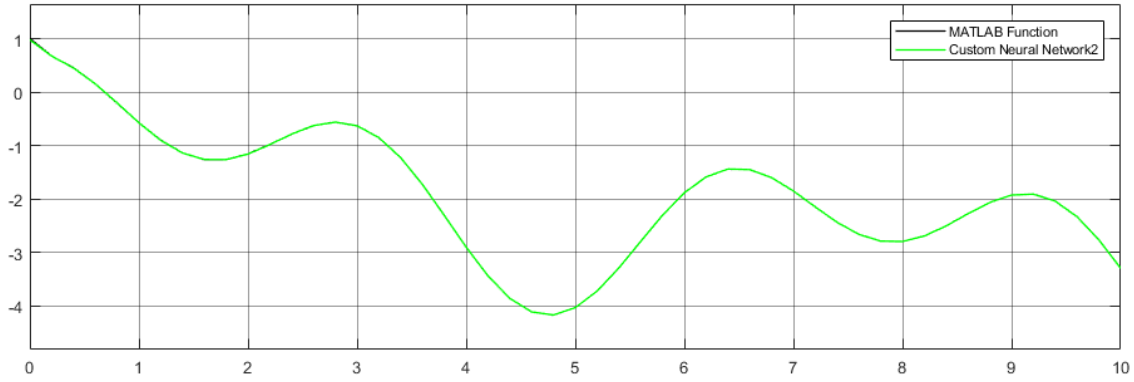
Obr. 20 Výstupné charakteristiky naučenej neurónovej siete

Regresia neurónovej siete po natrénovaní je zobrazená na nasledujúcom obrázku. Tu si môžeme všimnúť, že naša simulácia je už veľmi dobré odladená.



Obr. 21 Regresia siete po natrénovaní

Opäť vyexportujeme dáta a získame rovnakú blokovú schému, ako je uvedené v predošlom riešení (obr. 10) a danú simuláciu zopakujeme . Na nasledujúcom obrázku je znázornený výsledný priebeh nami vytvorenej neurónovej siete.

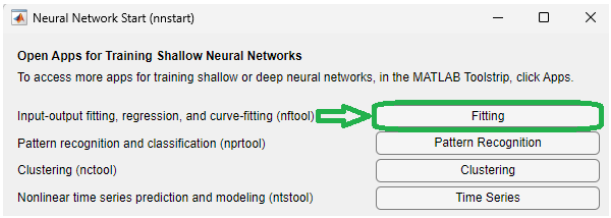


Obr. 22 Výsledný priebeh nami vytvorenej neurónovej siete

Tu vidíme, že počet neurónov aj funkcie vrstiev sú vyhovujúce a neurónová sieť už nevyžaduje ďalšie doladenie.

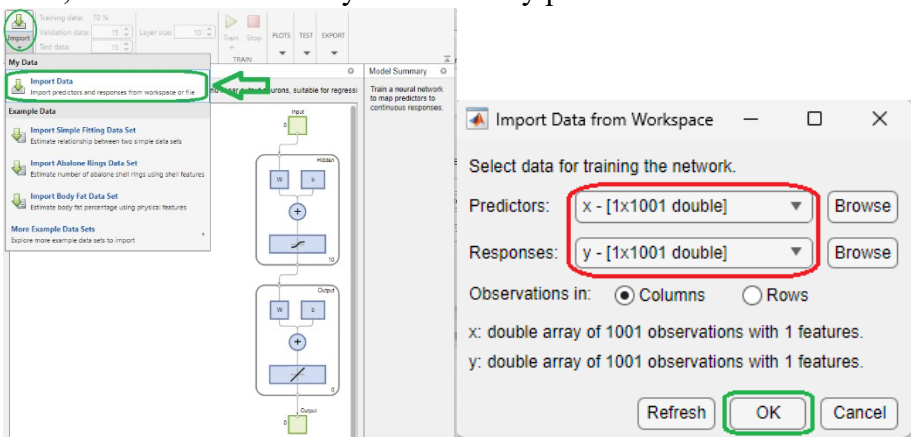
3. Vytvorenie neurónovej siete v novej verzii MATLABu

Keďže príkaz „>> nntool“ sa už nepoužíva a nie je podporovaný, tak budeme sa ďalej venovať návrhom pomocou príkazu „>> nnstart“ (Neural Network Start) v novej verzii MATLABu. Opäť si zadefinujeme náš vektor „x“ a funkciu „y“ do workspace-u v MATLABe, následne otvoríme grafické rozhranie neurónovej siete pomocou príkazu „>> nnstart“.



Obr. 23. Neurónová sieť – grafické rozhranie

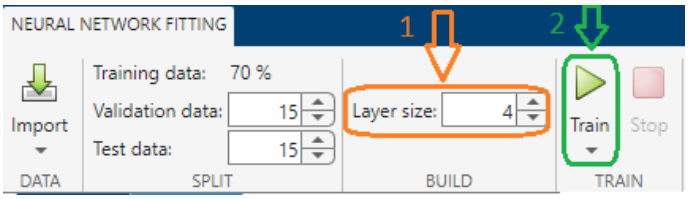
Vytvoríme neurónovú sieť s jedným neurónom vo výstupnej vrstve a z niekoľkými neurónmi v skrytej vrstve. Importujeme z MATLABu vektor „x“ a funkciu „y“, tak ako je nižšie zobrazené. Môžeme si všimnúť, že sa nám automaticky vygenerovala štruktúra neurónovej siete, ktorá má automaticky zadefinovaný počet neurónov.



Obr. 24 Import dát z workspace-u a štruktúra neurónovej siete

3.1 Neurónová sieť so štyrmi neurónmi

Zvolíme si počet neurónov v skrytej vrstve na hodnotu 4, tak ako sme simulovali v staršej verzií a tieto výsledné charakteristiky porovnáme. Na nasledujúcom obrázku je zobrazené ako nastaviť počet neurónov v sieti (krok 1) a následne naučenie pomocou funkcie „Train“ (2).



Obr. 25 Nastavenie počtu neurónov v sieti

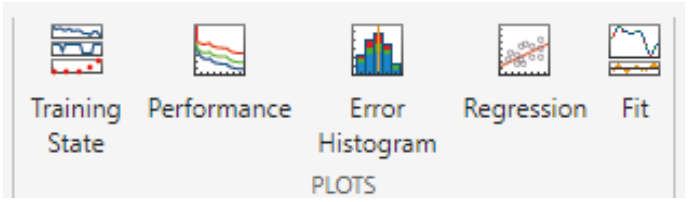
V časti „Training Progress“ môžeme vidieť aktuálny stav učenia, typ trénovanej funkcie a veľkosť chyby.

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	111	1000
Elapsed Time	-	00:00:00	-
Performance	6.56	0.109	0
Gradient	22.2	0.0225	1e-07
Mu	0.001	1e-07	1e+10
Validation Checks	0	6	6

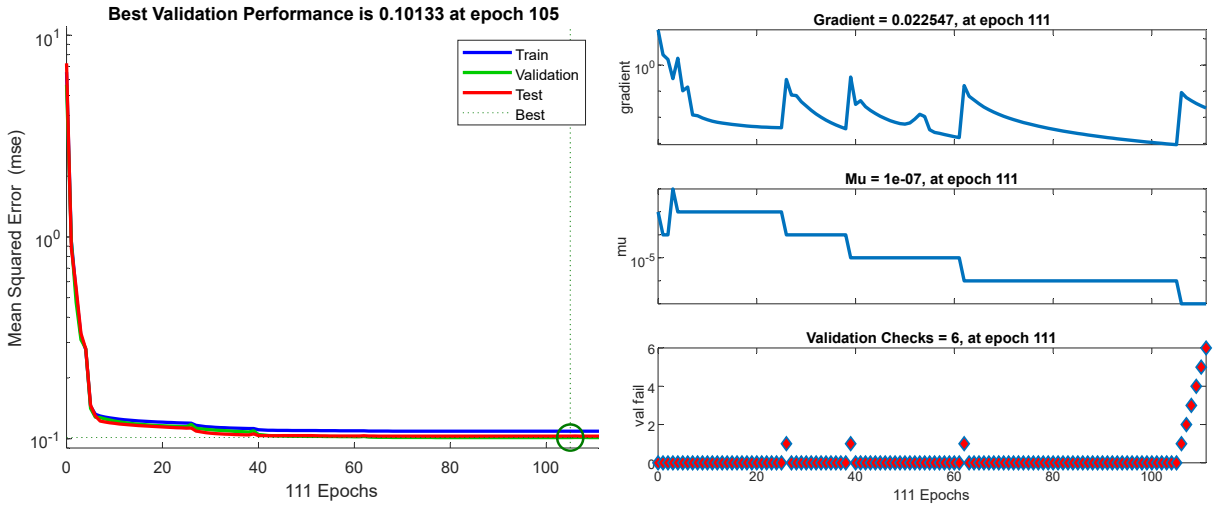
Obr. 26 Proces učenia sa neurónovej siete

Po naučení sa neurónovej sieti vieme zobraziť výsledne charakteristiky našej funkcie, ktoré sú zobrazené nižšie na obrázku 28. Na nasledujúcom obrázku sú zobrazené možnosti grafických charakteristík naučenej neurónovej sieti.



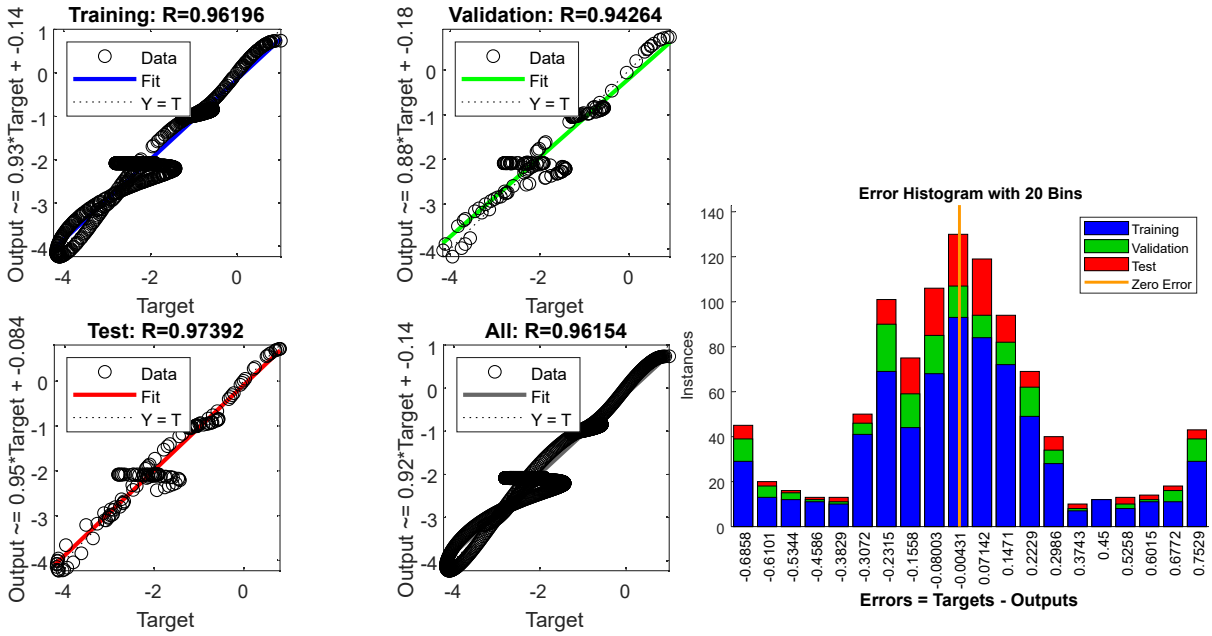
Obr. 27 Možnosti grafických charakteristík naučenej neurónovej sieti

Jednotlivé grafické charakteristiky sú zobrazené nižšie.



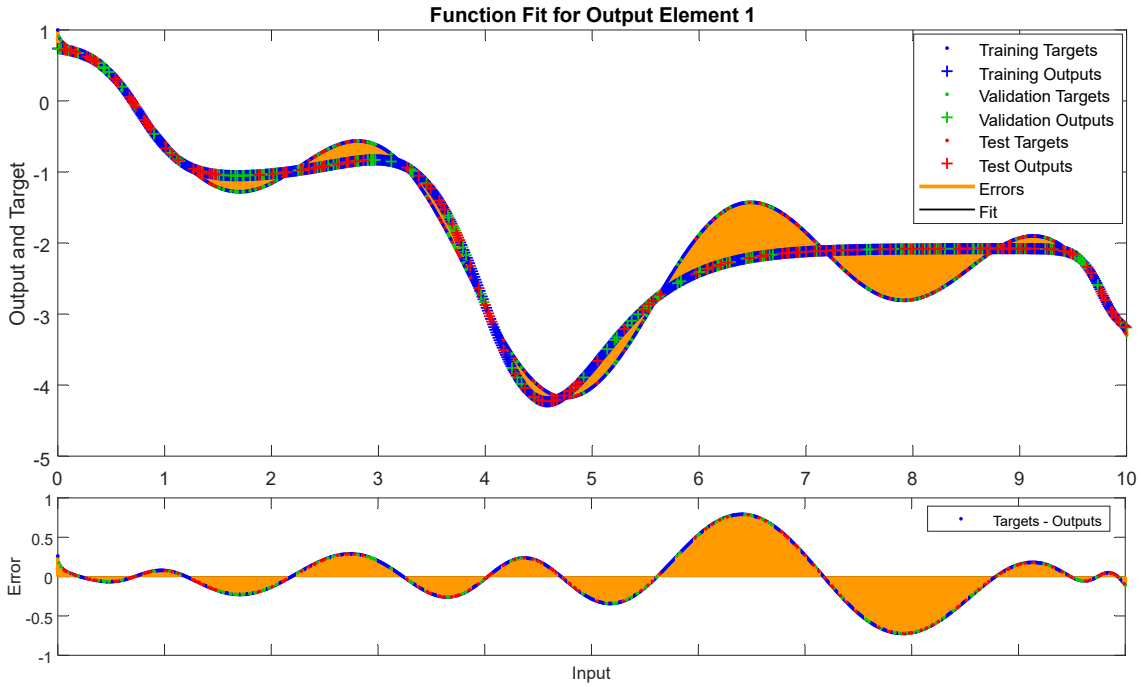
Obr. 28 Výstupné charakteristiky naučenej neurónovej siete

Regresia neurónovej siete po natrénovaní je zobrazená na nasledujúcom obrázku. Tu si môžeme všimnúť, že naša simulácia nie je dostatočne odladená, presne tak ako sme simulovali v staršej verzii MATLABu. Hneď vedľa je nová charakteristika – error histogram.



Obr. 29. Regresia siete po natrénovaní a histogram chýb

Výstupnú charakteristiku počas trénovania môžeme vidieť na nasledujúcom obrázku. Tu je doplnený priebeh s chybami.

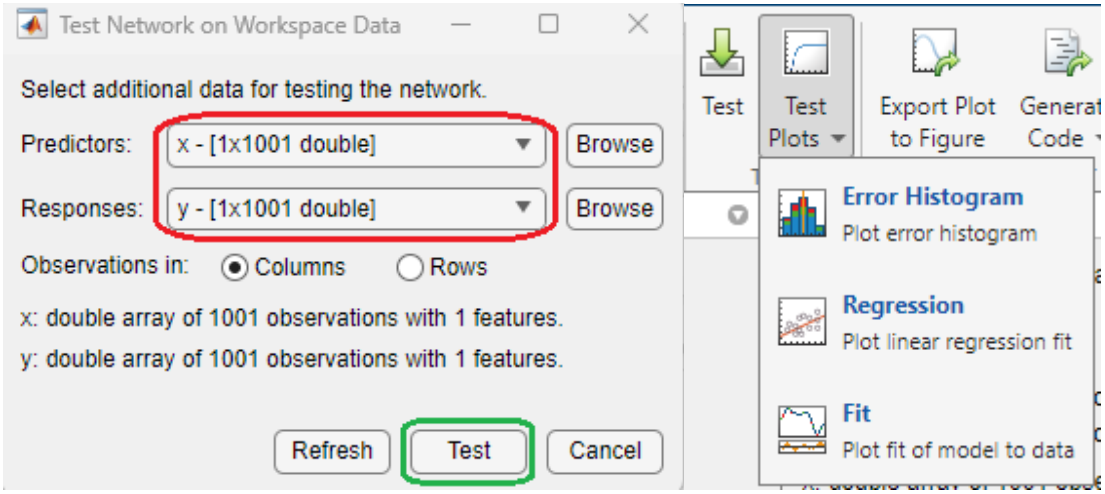


Obr. 30 Priebeh trénovania neurónovej sieti a chýb

V ďalšom kroku využijeme možnosť otestovania našej už naučenej neurónovej sieti.

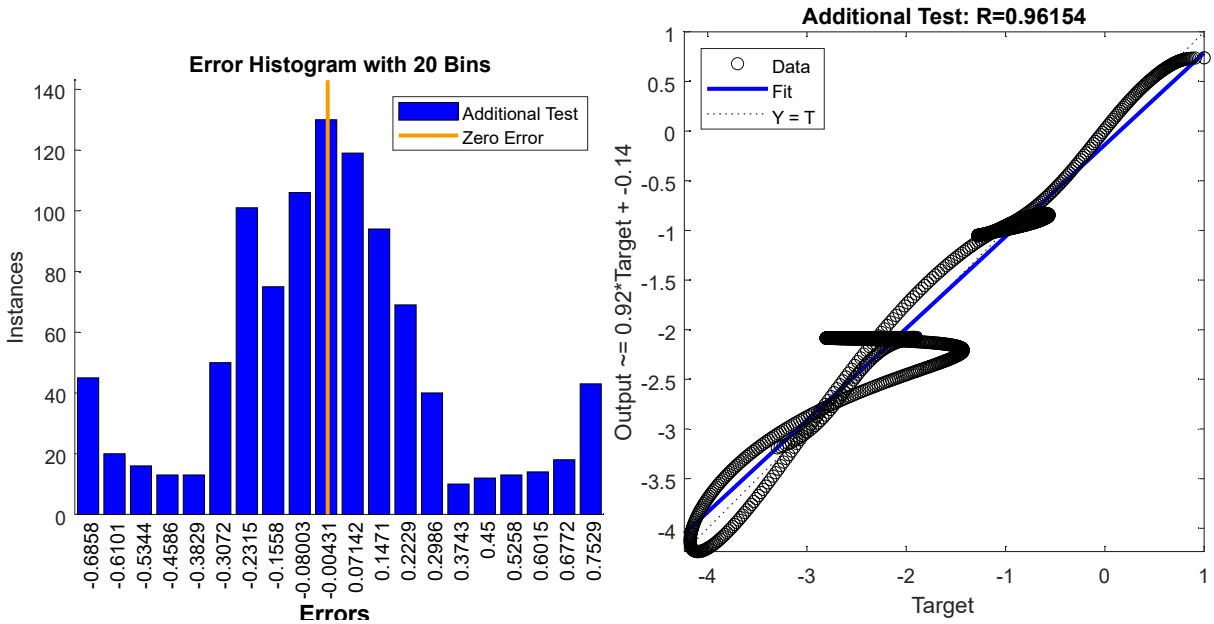
Obr. 31 Test naučenej neurónovej sieti

Opäť zadefinujeme náš vektor „x“ a funkciu „y“ a dáme testovať neurónovú sieť, tak ako je zobrazené nižšie. Po otestovaní máme opäť možnosť zobraziť grafické charakteristiky už otestovanej neurónovej sieti podľa nami zvolenej nelineárnej funkcie.



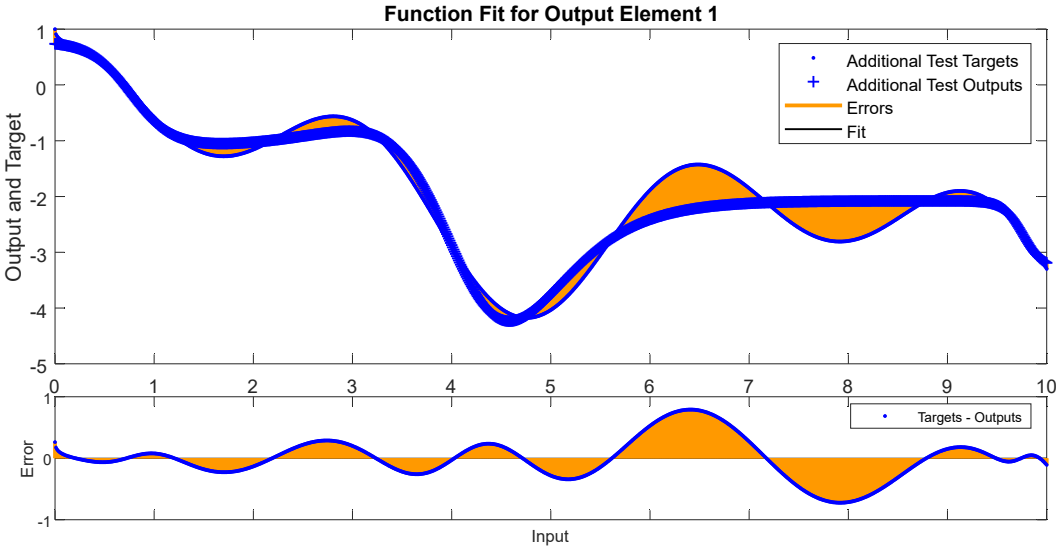
Obr. 32 Test neurónovej siete

Na nasledujúcom obrázku je zobrazený histogram chýb a vedľa vpravo regresia neurónovej siete v porovnaní so želanou nelineárnou funkciou.



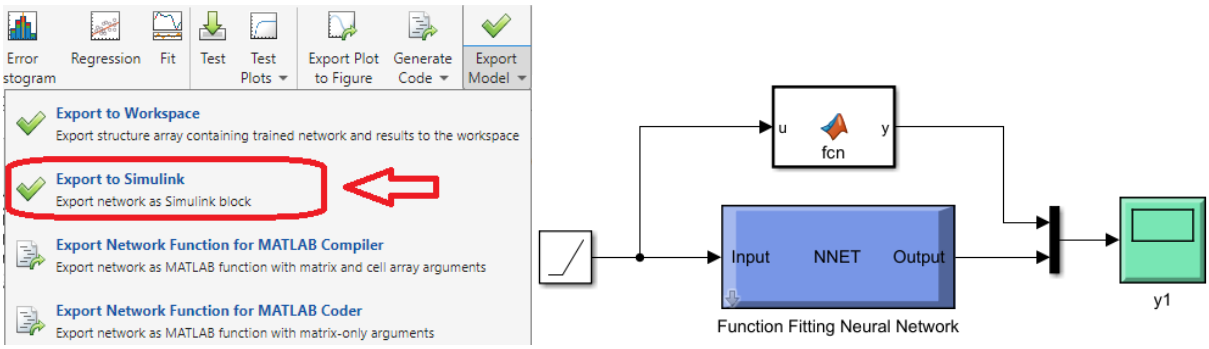
Obr. 33 Histogram chýb a regresia testovanej neurónovej siete

Výstupnú charakteristiku počas testovania môžeme vidieť na nasledujúcom obrázku. Tu je doplnený priebeh s chybami.



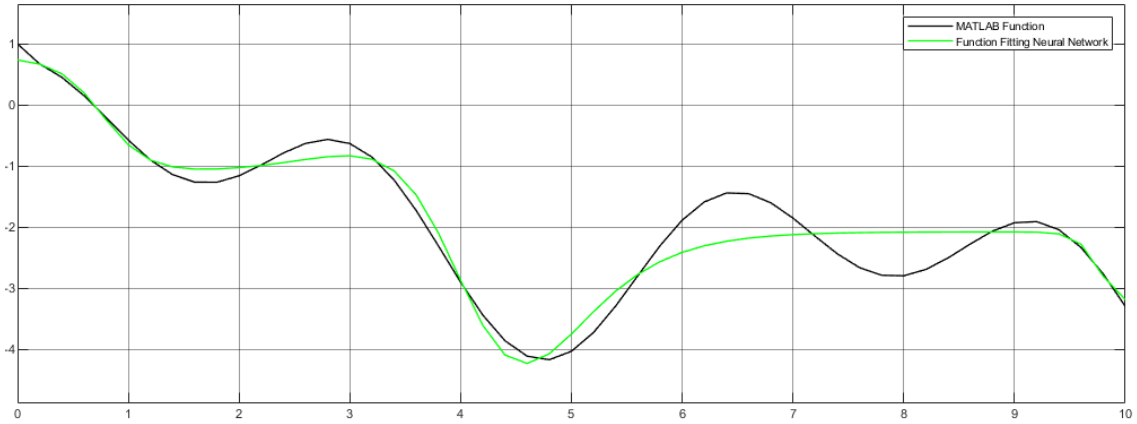
Obr. 34 Priebeh testovania neurónovej sieti a chýb

Pre overenie našej simulácie si túto neurónovú sieť exportujeme do Simulinku, kde nám automaticky vygeneruje blokovú schému. Túto schému doplníme o funkčný blok, kde zadefinujeme našu funkciu v tvare: `y = cos(2*u(1)) + sin(u(1)) - sqrt(u(1));`
Ďalej zmeníme vstupnú hodnotu na „Ramp“.



Obr. 35 Export dát do Simulinku a bloková schéma neurónovej siete v Simulinku

Na nasledujúcom obrázku je znázornený výsledný priebeh nami vytvorenej neurónovej siete.



Obr. 36 Výsledný priebeh nami vytvorenej neurónovej siete

3.2 Neurónová sieť s desiatimi neurónmi

Ako vidíme, tak štyri neuróny v skrytej vrstve nestačia a preto si v ďalšom kroku ukážeme už len výsledné charakteristiky pri desiatich neurónoch v skrytej vrstve. Postup je rovnaký ako v predošlej ukážke. Pre jednoduchosť sa nebudeme zaoberať niektorými charakteristikami, ale len tými najdôležitejšími.

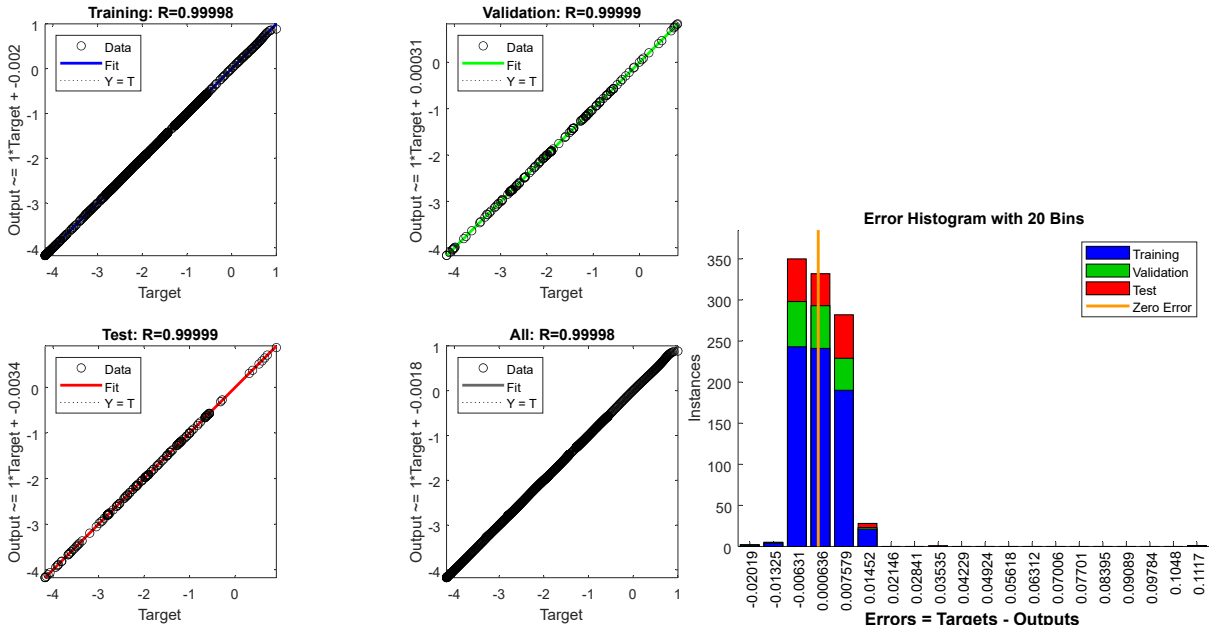
V časti „Training Progress“ môžeme vidieť aktuálny stav učenia, typ trénovanej funkcie a veľkosť chyby.

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	39	1000
Elapsed Time	-	00:00:00	-
Performance	19.8	3.77e-05	0
Gradient	46.4	0.000253	1e-07
Mu	0.001	1e-06	1e+10
Validation Checks	0	6	6

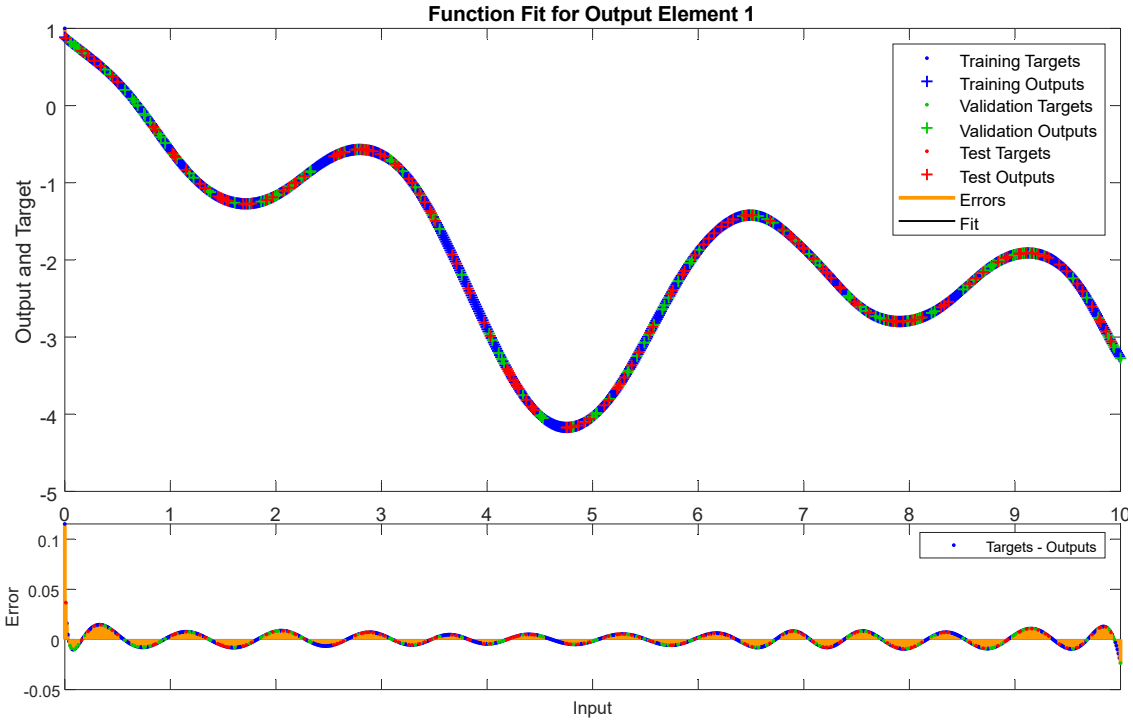
Obr. 37 Proces učenia sa neurónovej siete

Regresia neurónovej siete po natrénovaní je zobrazená na nasledujúcom obrázku. Tu si môžeme všimnúť, že naša simulácia je už veľmi dobré odladená, presne tak ako sme simulovali v staršej verzii MATLABu. Hneď vedľa je nová charakteristika – error histogram.



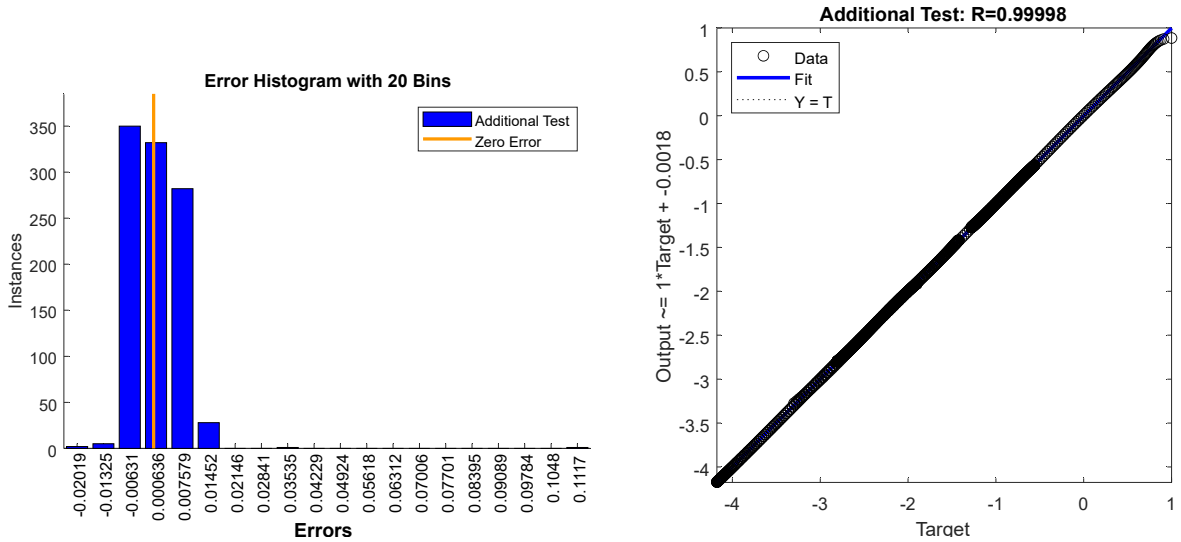
Obr. 38 Regresia siete po natrénovaní a histogram chýb

Výstupnú charakteristiku počas trénovania môžeme vidieť na nasledujúcom obrázku. Tu je doplnený priebeh s chybami.



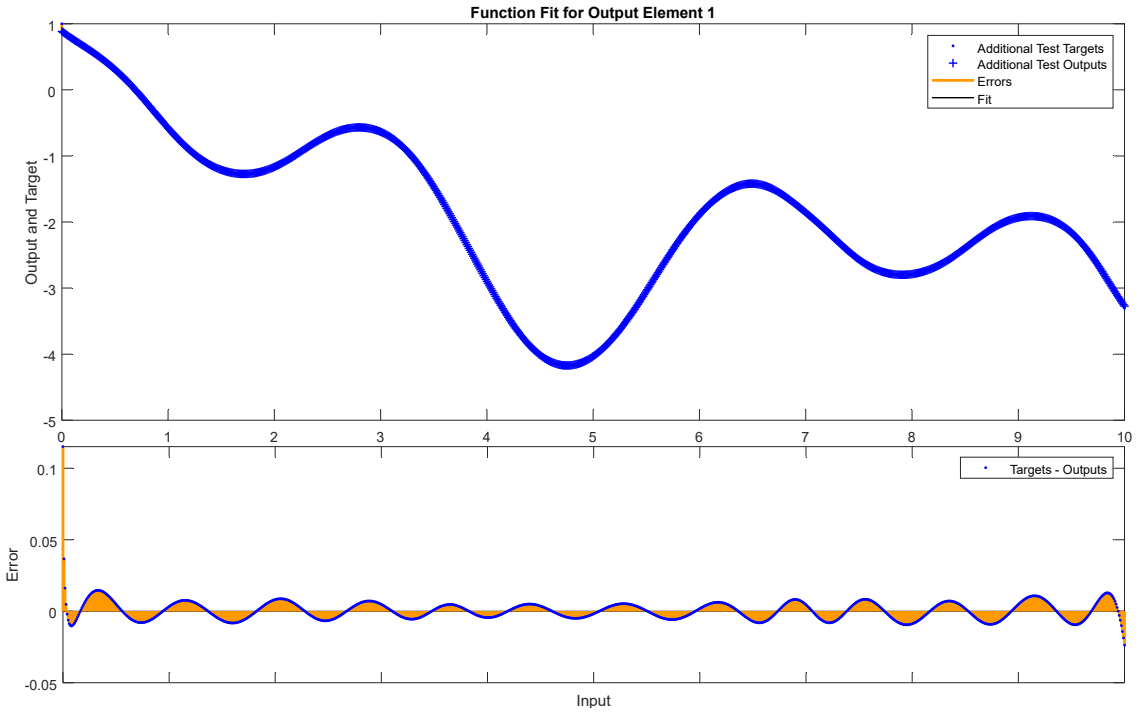
Obr. 39 Priebeh trénovania neurónovej sieti a chýb

V ďalšom kroku využijeme možnosť otestovania našej už naučenej neurónovej sieti. Opäť zadefinujeme náš vektor „x“ a funkciu „y“ a dáme testovať neurónovú sieť, tak ako sme si ukázali vyššie.
Na nasledujúcom obrázku je zobrazený histogram chýb a vedľa vpravo regresia neurónovej siete v porovnaní so želanou nelineárnou funkciou.



Obr. 40 Histogram chýb a regresia testovanej neurónovej siete

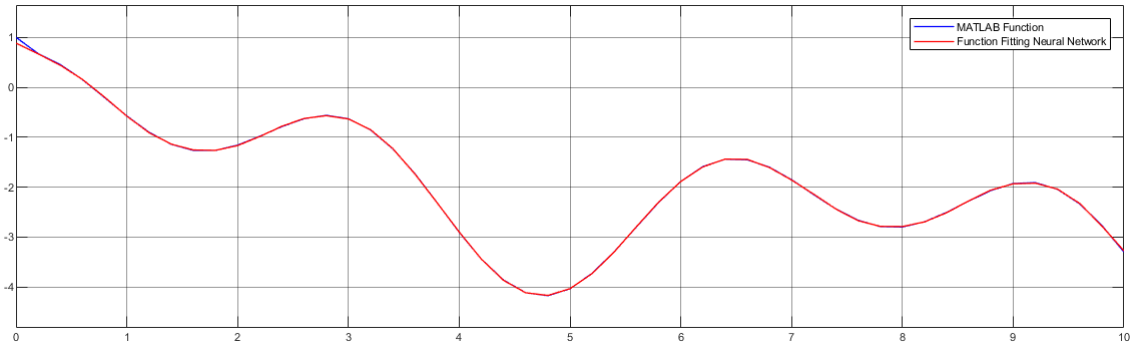
Výstupnú charakteristiku počas testovania môžeme vidieť na nasledujúcom obrázku. Tu je doplnený priebeh s chybami.



Obr. 41 Priebeh testovania neurónovej sieti a chýb

Pre overenie našej simulácie si túto neurónovú sieť opäť exportujeme do Simulinku, kde nám automaticky vygeneruje blokovú schému. Túto schému doplníme o funkčný blok a zmeníme vstupnú hodnotu na „Ramp“, tak ako sme si ukázali vyššie.

Na nasledujúcom obrázku je znázornený výsledný priebeh nami vytvorenej neurónovej siete.



Obr. 42 Výsledný priebeh nami vytvorenej neurónovej siete

ZÁVER

Podľa zadania sme vytvorili nelineárnu funkciu pomocou programu MATLAB, ktorú sme následne odsimulovali. Ďalej sme vytvorili neurónovú sieť v grafickom prostredí, ktoré ponúka „deep learning toolbox“ v MATLABe. Vytvorili sme rôzne simulácie, kde sme si overili správnosť neurónovej siete. Skúšali sme simulácie pri štyroch aj desiatich neurónoch v skrytej vrstve a zmene funkcie. Táto neurónová sieť dokázala opakovať našu nelineárnu funkciu. Simuláciami sme dospeli k záveru, že danú nelineárnu funkciu najlepšie opakuje neurónová sieť pri desiatich neurónoch v skrytej vrstve.

V poslednom rade sme si ukázali, ako tieto simulácie aplikovať v novom grafickom rozhraní novej verzie MATLABu, keďže stará verzia už nie je podporovaná a starší spôsob nie je dostupný v novej verzií. Zistili sme, že nové prostredie je užívateľský prijateľnejšie a jednoduchšie.