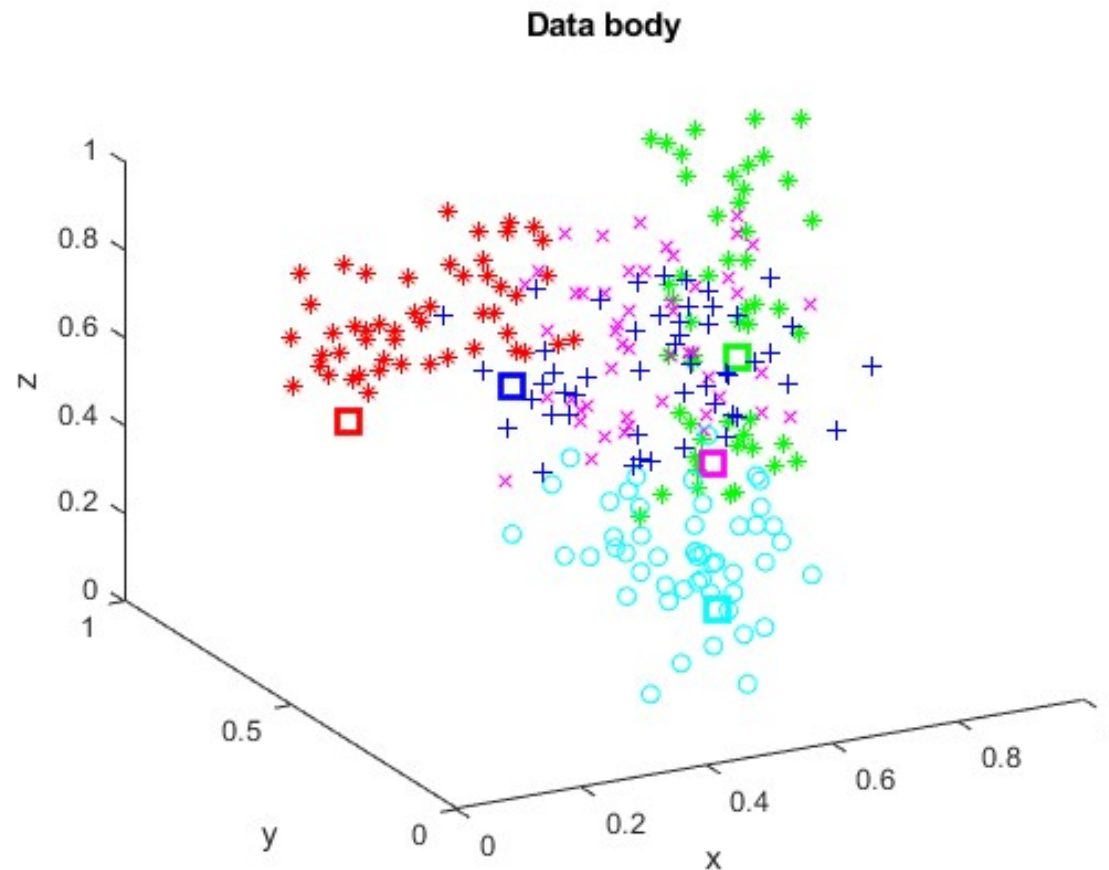


***Viacvrstvové  
perceptrónové siete  
(Multilayer Perceptron networks - MLP)***

***Zadania a postup riešenia  
k úlohám 4, 5, 6.***

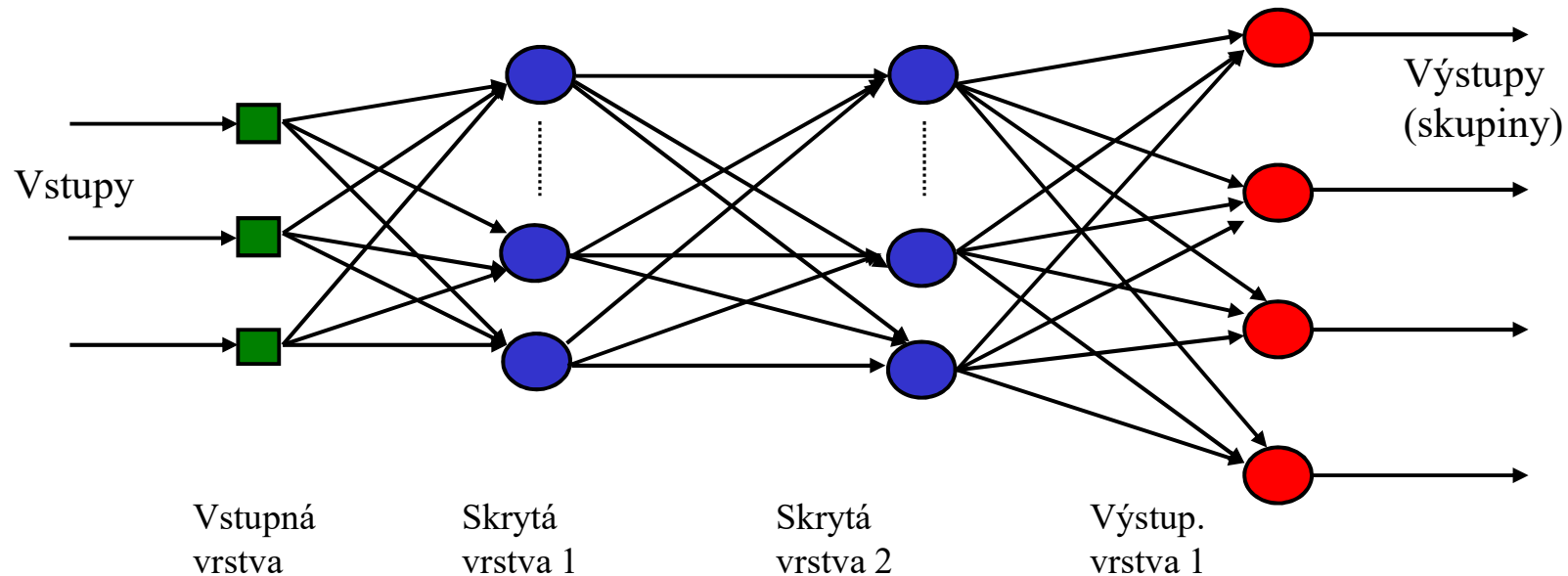
## Úloha 4 - Klasifikácia do skupín pomocou MLP siete.



Na základe rozmiestnenia bodov  $(x, y, z)$  súradníc – vstupných parametrov do MLP siete realizovať klasifikáciu (zatriedenie) bodov do 5 skupín (tried)

# Klasifikácia – MLP sieť

## Štruktúra MLP siete pre klasifikáciu, rozpoznávanie (patternnet)



- Vstupy NS sú parametre na základe, ktorých realizujeme triedenie do skupín
- Výstupy NS prislúchajú k jednotlivým skupinám, t.j. počet výstupov sa rovná počtu skupín, rozsah výstupu je  $<0,1>$  a vyjadruje pravdepodobnosť zatriedenia do skupiny.
- Aktivačné funkcie vo všetkých vrstvách môžu byť „tansig“ alebo „logsig“, vo výstupnej vrstve sa často používa „softmax“
- Počty skrytých vrstiev ako aj počty neurónov v nich sa definuje experimentálne podľa zložitosti klasifikačnej úlohy.
- Chybová funkcia – krížová entropia (cross entropy)

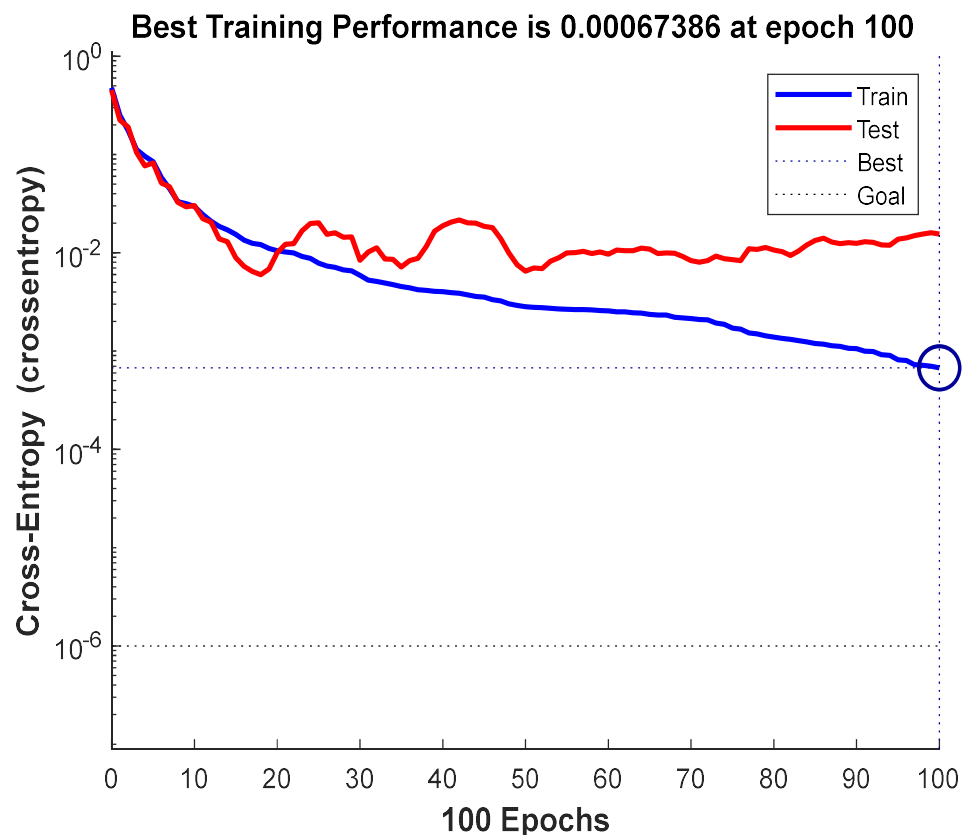
## Ukončenie procesu tréovania

- a) Dosiahnutie predpísanej presnosti modelu (globálnej chyby)
- b) Uskutočnenie predpísaného počtu epoch tréovania
- c) Stagnácia priebehu chyby (gradient chyby dosiahne def. úroveň)
- d) Alebo po úspešnom teste zovšeobecňovacej schopnosti n.s.

Proces tréovania je potrebné včas ukončiť, v opačnom prípade môže nastať stav "pretréovania", keď sa parametre n.s. už prestanú zlepšovať a začínajú na ne mať vplyv rôzne parazitné signály ako šum a pod., ktoré môžu následne zhoršiť vlastnosti siete (kopírovanie šumu ...).

Počas tréovania sa vyhodnocuje chyba na inom balíku dát, ak chyba začne od určitej etapy tréovania narastať – tréovanie sa zastaví

# Proces tréovania – vyhodnotenie úspešnosti klasifikácie (kontingenčná tabuľka)

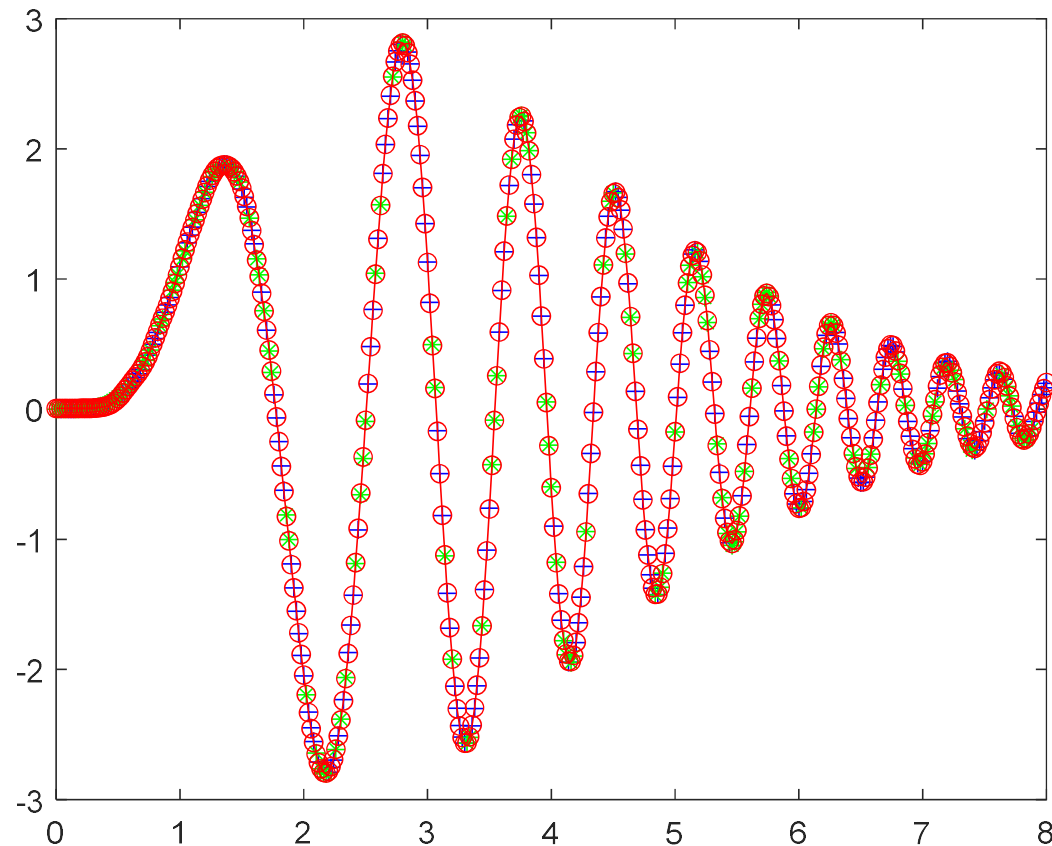


Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	Summary
1	50 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	50 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	50 20.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	50 20.0%	1 0.4%	98.0% 2.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 19.6%	100% 0.0%
Summary	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	98.0% 2.0%	99.6% 0.4%

Nastavenie počtu neurónov a parametrov ukončenia tréovania, tak aby chyba na testovacích dátach klesla čo najnižšie a zastaviť tréovanie pri jej väčšom náraste.

## Úloha 5 - Aproximácia nelineárnej funkcie pomocou MLP siete.



Pomocou MLP siete aproximujeme závislosť výstupu funkcie  $y$  od vstupu  $x$  ,  $y=f(x)$  .

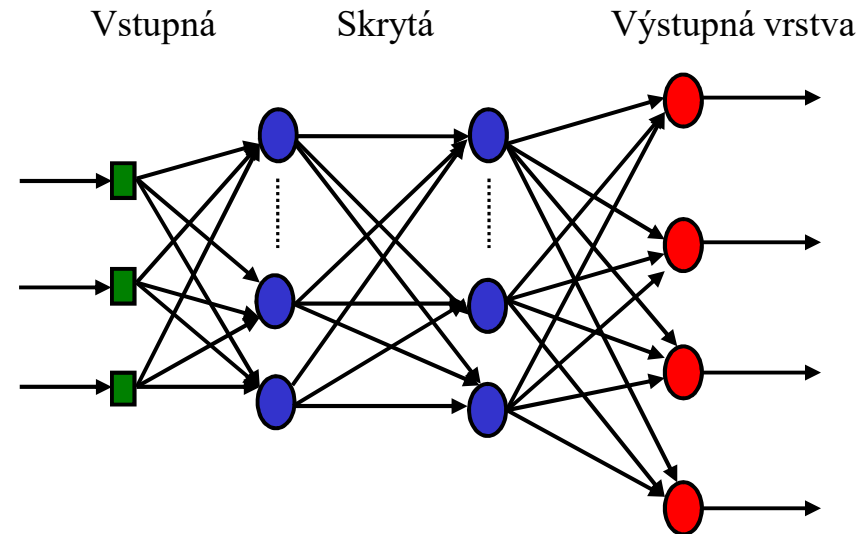
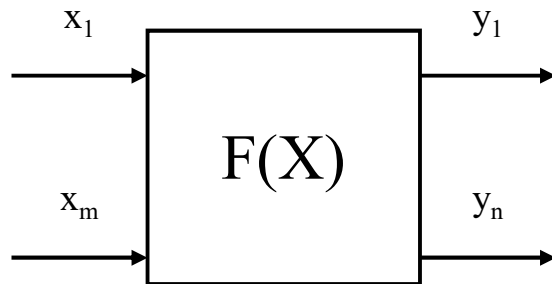
# Aproximácia, modelovanie – MLP sieť

## Štruktúra MLP siete pre aproximáciu, modelovanie (fitnet)

Aproximovaná funkcia, modelovaný systém:

$Y=F(X)$ , Y-výstupy, X-vstupy, F-nelineárna

funkcia,  $X=[x_1, \dots, x_m]$ ,  $Y=[y_1, \dots, y_n]$



- Vstupy NS sú vstupy systému (funkcie), výstupy NS sú výstupy systému (funkcie)
- Aktivačné funkcie v skrytých vrstvách sú „tansig“ a vo výstupnej „purelin“.
- Počty skrytých vrstiev ako aj počty neurónov v nich sa definuje experimentálne podľa zložitosti modelovaného systému.
- Chybová funkcia – Stredná kvadratická odchýlka (MSE – Mean square error)

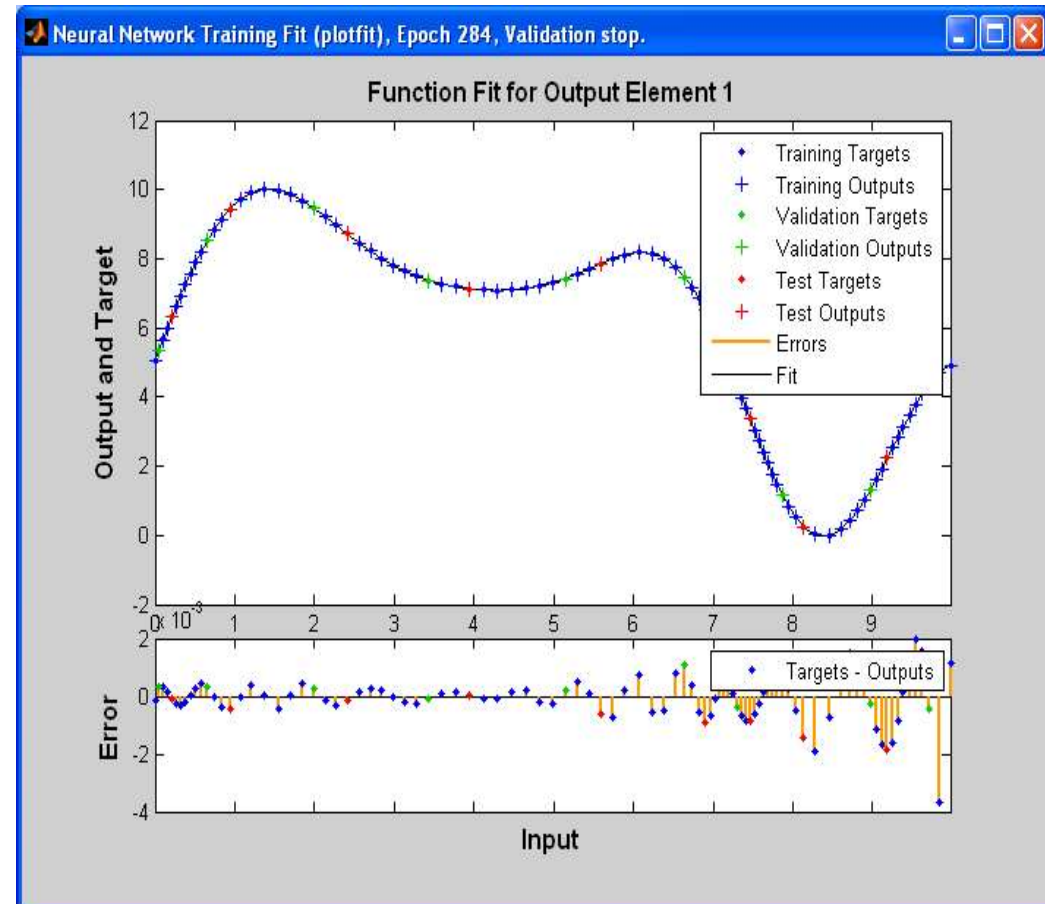
# Aproximácia nelineárnej funkcie $y=f(x)$ z meraných dát pomocou UNS

```
[x,y]=simplefit_dataset;
```

```
net=fitnet(10);
```

```
net.divideFcn='divideint';  
net.divideParam.trainRatio=0.8;  
net.divideParam.valRatio=0.1;  
net.divideParam.testRatio=0.1;
```

```
net.trainParam.goal = 1e-7;  
net.trainParam.show = 5;  
net.trainParam.epochs = 100;  
net=train(net,x,y);
```





## Spôsoby generovania (rozdeľovania) trénovacích, validačných a testovacích dát

Typ funkcie na generovanie dát

**net.divideFcn**

```
net.divideFcn='dividerand';    % náhodné rozdelenie  
net.divideFcn='divideblock';   % rozdelenie po blokoch dát za sebou  
net.divideFcn='divideint';     % je použitá každá n-tá vzorka  
net.divideFcn='dividettrain';  % všetky dáta sú iba trénovacie
```

% parametre rozdelenia dát

```
net.divideParam.trainRatio=0.8;  
net.divideParam.valRatio=0.1;  
net.divideParam.testRatio=0.1;
```

## Spôsoby generovania (rozdeľovania) trénovacích, validačných a testovacích dát

Typ funkcie na generovanie dát

**net.divideFcn**

```
net.divideFcn='divideind';    % indexové rozdelenie
```

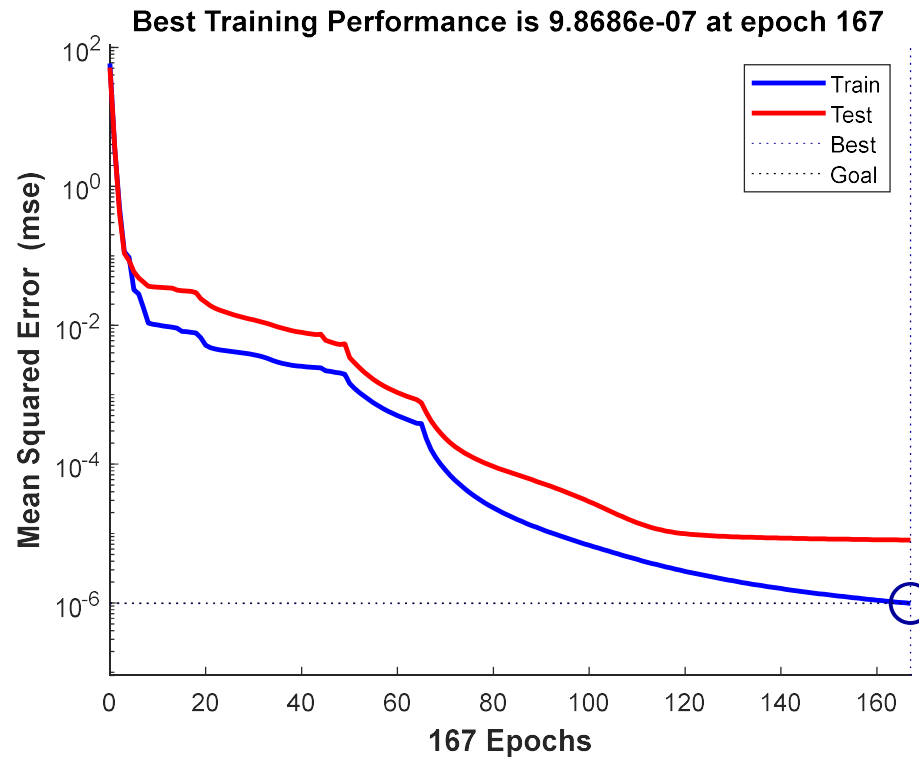
% parametre rozdelenia dát

```
net.divideParam.trainInd=1:2:n;
```

```
net.divideParam.valInd=2:2:n2;
```

```
net.divideParam.testInd=n2+1:2:n;
```

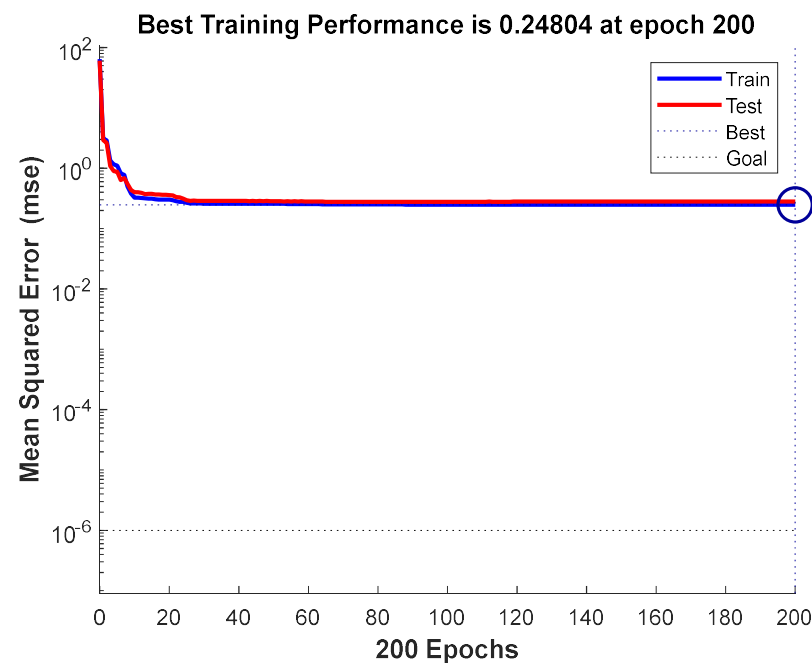
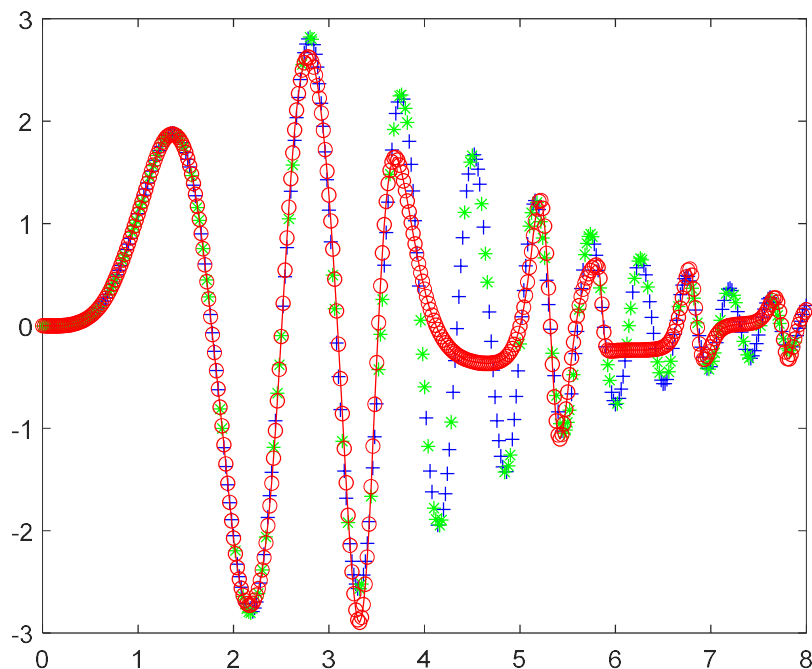
# Proces tréovania – vyhodnotenie kvality aproximácie (chyby SSE, MSE, MAE)



Train: SSE1 = 1.1842e-04, MSE1 = 9.8686e-07, MAE1 = 0.0025

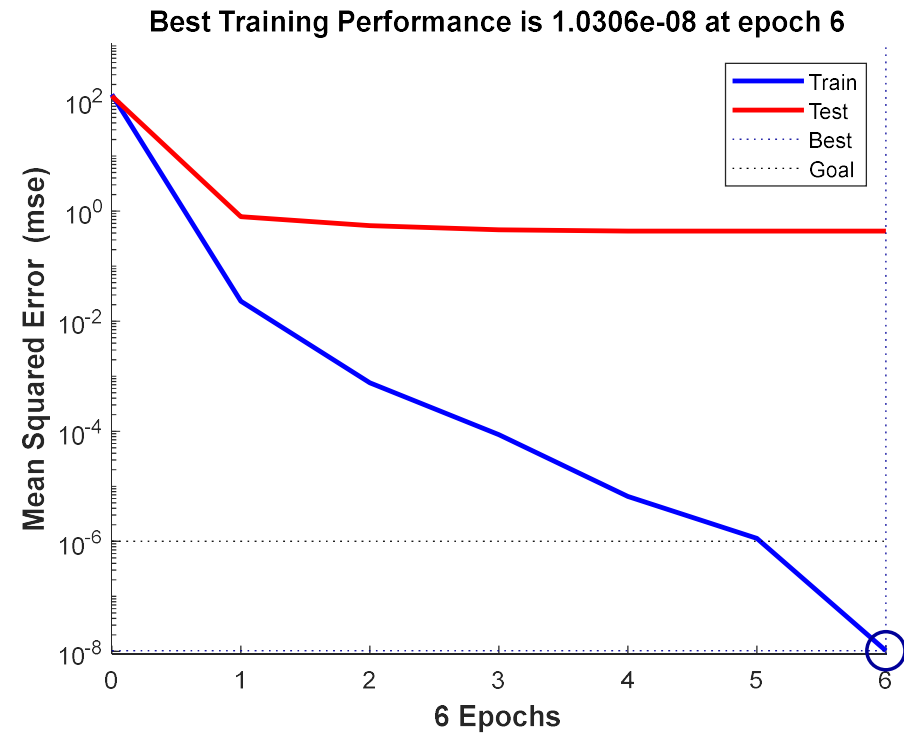
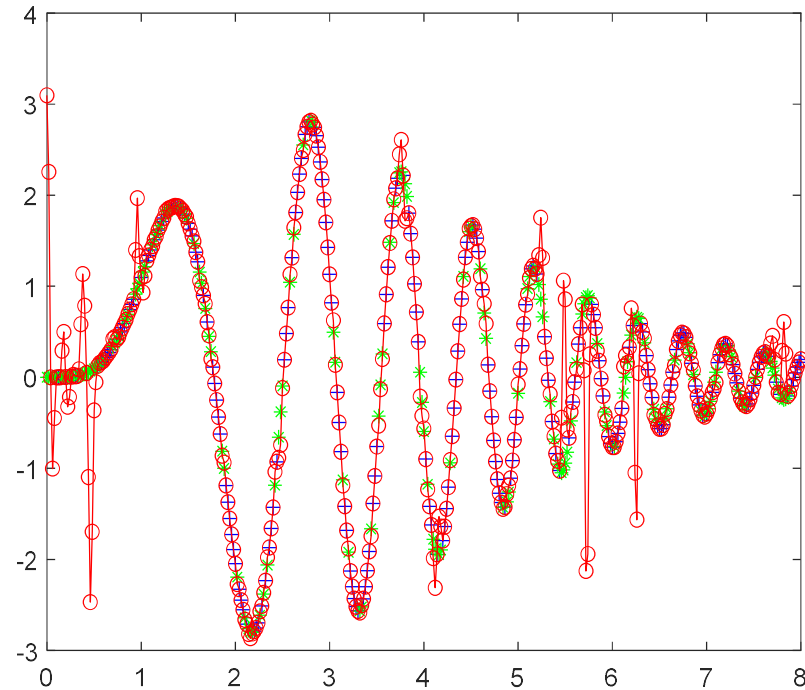
Test: SSE2 = 6.5177e-04, MSE2 = 8.0466e-06, MAE2 = 0.0183

## *Proces tréovania – malý počet neurónov*



Odchýlka (chyba) na tréovacích aj testovacích datach je vysoká

## *Proces tréovania – veľký počet neurónov*



Odchýlka (chyba) na tréovacích datach je nízka (NS sa zafixuje na tréovacie data), ale na testovacích datach je chyba vysoká

## Úloha 6 – Rozpoznávanie (diagnostika) ochorenia pomocou MLP siete.

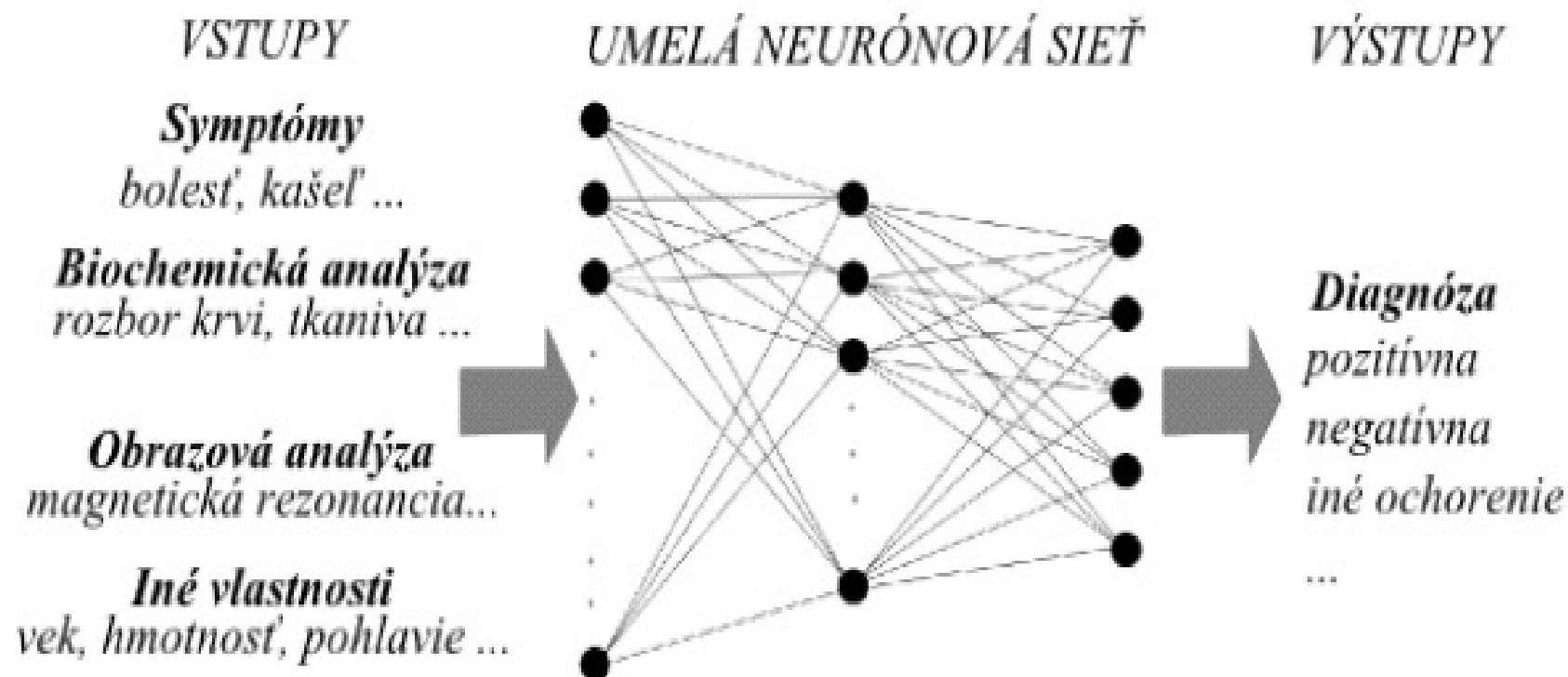
Na základe nameraných príznakov z CTG vyšetrenia, MLP sieť hodnotí stav bábätka do troch skupín.

Výpočet úspešnosti klasifikácie, senzitivity a špecificity.

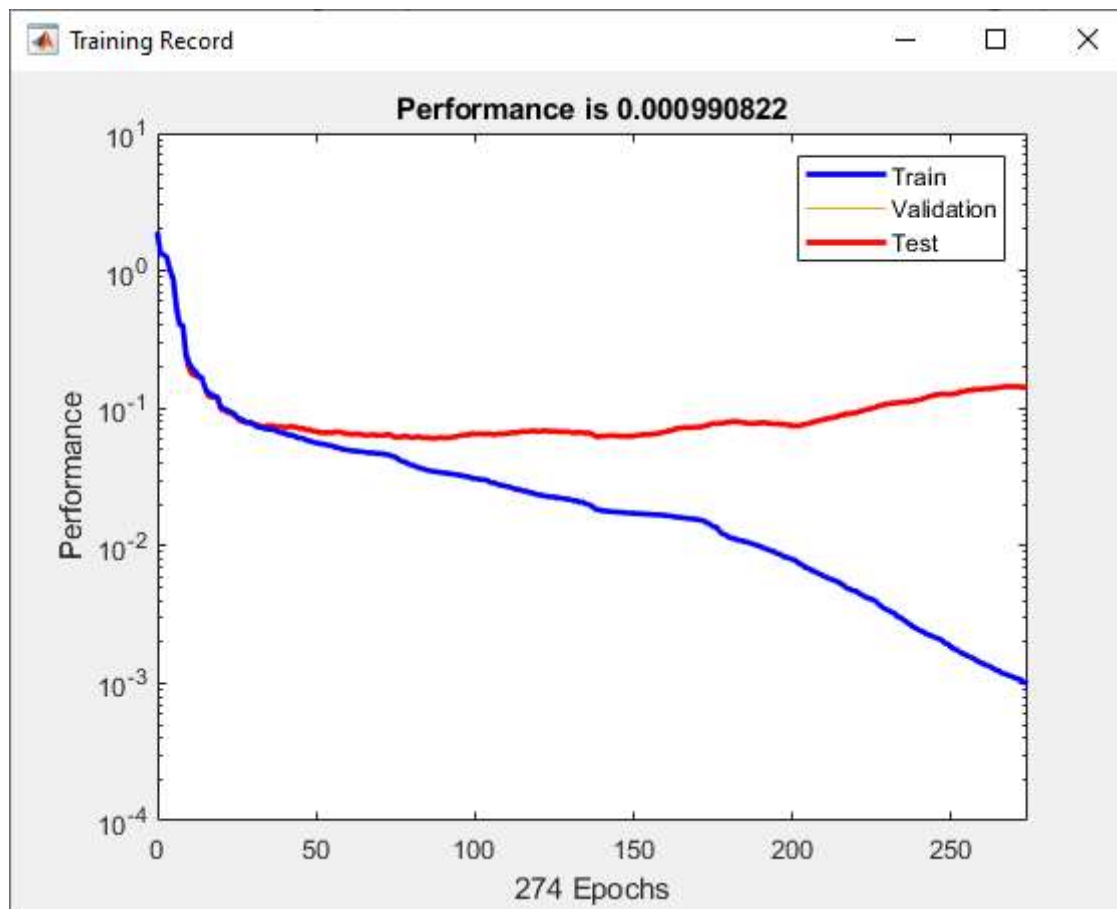
Štruktúra MLP siete – rovnaká ako pri klasifikácii

- Vstupy - príznaky získané z kardiotokografického záznamu
- Výstupy – 3 výstupy (triedy) – stav (**1-normálny, 2-podozrivý, 3-patologický**)
- Štruktúra siete **patternet** s jednou (alebo dvoma) skrytými vrstvami
- Rozdelenie dát – 60% trénovanie, 40% testovanie
- Nastavenie počtu neurónov a ukončovacích podmienok trénovania

# Diagnostika ochorení pomocou NS



# Proces tréovania – vyhodnotenie úspešnosti klasifikácie



Nastavenie počtu neurónov a parametrov ukončenia tréovania, tak aby chyba na testovacích dátach klesla čo najnižšie a zastaviť tréovanie pri jej väčšom náraste.



# Vyhodnotenie úspešnosti klasifikácie

## Kontingenčná tabuľka

Postupovať podľa zadania:

- 10 spustení tréovania
- urobiť štatistiku úspešnosti
- podrobnejšie sa venovať najlepšej NS
- na porovnanie 2 iné štruktúry siete

Kontingencna - testovacie data

Output Class	1	2	3	
	645 75.9%	24 2.8%	4 0.5%	95.8% 4.2%
	22 2.6%	89 10.5%	2 0.2%	78.8% 21.2%
	0 0.0%	5 0.6%	59 6.9%	92.2% 7.8%
				Target Class
				1 96.7% 3.3%
				2 75.4% 24.6%
				3 90.8% 9.2%
				93.3% 6.7%

Naj.Úspešnosť klasifikácie (celková, tréovanie, testovanie): 97.2248 99.8433 93.2941 %

Úspešnosť celková klasifikácie (min, max, mean): 96.6604 97.2248 96.9755 %

Úspešnosť tréovacie klasifikácie (min, max, mean): 99.7649 100.0000 99.8824 %

Úspešnosť testovacie klasifikácie (min, max, mean): 91.8824 93.2941 92.6118 %

# Úspešnosť klasifikácie, senzitivita (citlivosť), špecificita

**sensitivity, recall, hit rate, or true positive rate (TPR)**

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

**specificity, selectivity or true negative rate (TNR)**

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

**precision or positive predictive value (PPV)**

$$PPV = \frac{TP}{TP + FP} = 1 - FDR$$

**negative predictive value (NPV)**

$$NPV = \frac{TN}{TN + FN} = 1 - FOR$$

**accuracy (ACC)**

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

**balanced accuracy (BA)**

$$BA = \frac{TPR + TNR}{2}$$

**F1 score**

is the **harmonic mean** of **precision** and **sensitivity**

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$$

Confusion Matrix			
On	<div>TN</div> <div>19932</div> <div>50.6%</div>	<div>FP</div> <div>852</div> <div>2.2%</div>	<div>TNR</div> <div>95.9%</div> <div>4.1%</div>
	<div>FN</div> <div>700</div> <div>1.8%</div>	<div>TP</div> <div>17926</div> <div>45.5%</div>	<div>TPR</div> <div>96.2%</div> <div>3.8%</div>
	<div>NPV</div> <div>96.6%</div> <div>3.4%</div>	<div>PPV</div> <div>95.5%</div> <div>4.5%</div>	<div>ACC</div> <div>96.1%</div> <div>3.9%</div>
	On	Off	Úspešnosť
	Target Class		

