

## Úloha 5a

---

◆ **1. Stručne vlastnými slovami, čo bolo úlohou (max. 3 vety):**

Úlohou bolo natrénovať viacvrstvovú perceptrónovú (MLP) neurónovú sieť, ktorá správne rozpozná a klasifikuje 3D body do piatich skupín. Cieľom bolo nájsť najjednoduchšiu možnú sieť (s čo najmenším počtom neurónov), ktorá zvládne klasifikovať minimálne 248 z 250 bodov. Na záver bolo úlohou otestovať 5 nových bodov a pomocou siete určiť, do ktorej skupiny patria.

---

◆ **2. Popis vstupných a výstupných dát, ako ste ich rozdelili na trénovacie, testovacie alebo aj validačné dátá:**

- **Vstupné dátá:** 3D súradnice bodov piatich skupín, každá skupina obsahovala 50 bodov.
- **Výstupné dátá:** čísla tried 0–4 reprezentujúce 5 skupín, ktoré sú zakódované pomocou one-hot reprezentácie.
- **Rozdelenie dát:**
  - Trénovacie dátá: 80%
  - Testovacie dátá: 20%
  - Validačné dátá sa nepoužili (`valInd = 0`)
  - Rozdelenie prebehlo náhodne pomocou dividerand.

◆ **3. Popis štruktúry MLP siete (vstupy, výstupy, typy neurónov v jednotlivých vrstvách, ich počet):**

- Vstupná vrstva: 3 neuróny (x, y, z)
  - Skrytá vrstva: 11 neurónov, aktivačná funkcia tansig
  - Výstupná vrstva: 5 neurónov (pre 5 tried), aktivačná funkcia softmax
  - Použitá sieť: patternnet(11)
- 

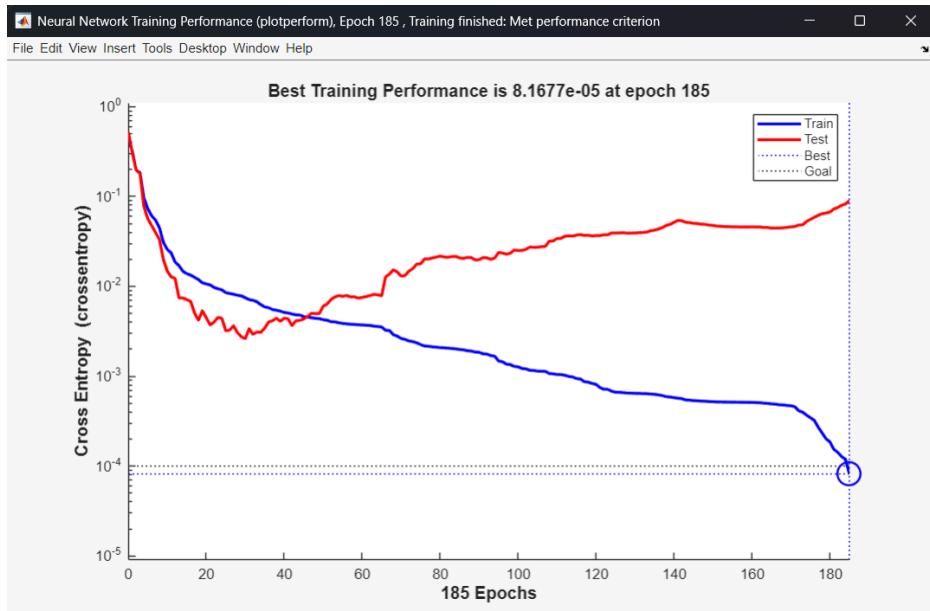
◆ **4. Nastavené parametre trénoania (ukončovacie podmienky, použitá kriteriálna funkcia):**

- **Ukončovacia podmienka:** cieľová chyba (`goal`) =  $1e-4$
  - **Maximálny počet epoch:** 200
  - **Frekvencia výpisu:** každých 20 epoch
  - **Maximálny počet neúspešných pokusov o zlepšenie:** `max_fail` = 12
  - Použitá kriteriálna funkcia: **Cross-Entropy**
- 

◆ **45 Stačí jedna čo najjednoduchšia štruktúra NS, čo spĺňala zadanie úlohy:**

- Skrytá vrstva mala **11 neurónov**
  - Pri týchto parametroch sieť správne klasifikovala **248/250 bodov**
  - Sieť bola trénovaná 185 epoch a dosiahla cieľovú chybu
- 

◆ **6. Graf priebehu procesu trénoania aj s komentárom:**



### Komentár:

- Graf zobrazuje **chybu (cross entropy)** počas učenia.
- **Modrá krivka (Train)** – chyba na trénovacích dátach – klesá pravidelne, čo naznačuje, že sieť sa efektívne učí.
- **Červená krivka (Test)** – chyba na testovacích dátach – najskôr klesá, no potom jemne stúpa, čo môže naznačovať **začínajúce pretrénovanie** po ~100. epoche.
- Najlepšia výkonnosť bola dosiahnutá pri **185. epoche**, kde sieť splnila požiadavku na presnosť.

### ◆ 7. Kontingenčná matica (plotconfusion) aj s komentárom:



### Komentár:

Sieť dosiahla vysokú presnosť na všetkých dátach. Trénovacia aj testovacia matica ukazujú minimálne chyby, väčšina vzoriek bola správne klasifikovaná. Celková presnosť je nad 99 %, čo potvrzuje správnu funkčnosť siete.

#### ◆ 8. Postup testovania vybraných 5 bodov:

5 nových bodov bolo definovaných ako:

newPoints = [0.2 0.3 0.7;

0.8 0.2 0.4;

0.5 0.5 0.5;

0.1 0.9 0.1;

0.6 0.6 0.6]';

Boli vložené do siete pomocou funkcie sim a výstupy boli klasifikované podľa triedy s najvyššou hodnotou (softmax).

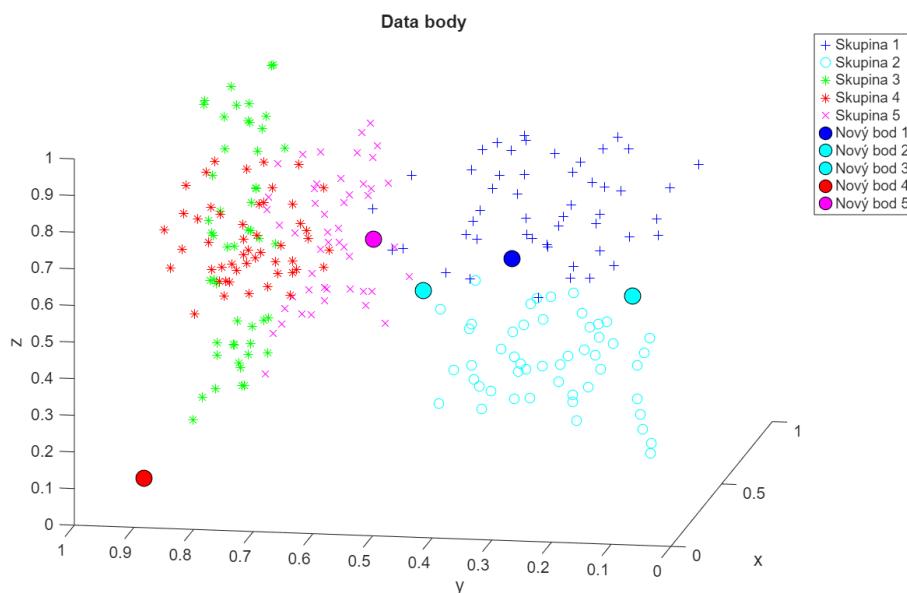
Výsledok bol:

#### Command Window

Triedy novych bodov:

1 2 2 4 5

Tieto body boli následne znázornené v 3D grafe ako farebné bodky, kde farba zodpovedala klasifikovanej triede.



## Úloha 5b

---

### ◆ 1. Stručne vlastnými slovami, čo bolo úlohou (max. 3 vety)

Cieľom úlohy bolo navrhnúť, natreňovať a otestovať viacvrstvovú perceptrónovú neurónovú sieť (MLP), ktorá aproximuje nelineárnu funkciu s jedným vstupom a výstupom. Úlohou bolo nájsť čo najjednoduchšiu sieť (s čo najmenším počtom neurónov), ktorá dosiahne dostatočnú presnosť – chyba na testovacích dátach menšia ako 1e-4. Výsledky siete boli vizuálne porovnané s reálnymi dátami a vyhodnotené pomocou chýb SSE, MSE a MAE.

---

### ◆ 2. Popis vstupných a výstupných dát, ako ste ich rozdelili na trénovacie, testovacie data:

- **Vstupné dátá:** premenná x – jednorozmerné hodnoty vstupnej funkcie.
- **Výstupné dátá:** premenná y – výstupy meranej nelineárnej funkcie  $f(x)$ .
- **Rozdelenie:**
  - Trénovacie dátá: podľa indexov `indx_train` (60 % vzoriek)
  - Testovacie dátá: podľa indexov `indx_test` (40 % vzoriek)
  - Rozdelenie bolo **indexové** (`divideind`), podľa pripravených indexov v súbore `datafun.mat`.

### ◆ 3. Popis štruktúry MLP siete (vstupy, výstupy, typy neurónov v jednotlivých vrstvách, ich počet):

- Vstupná vrstva: **1 neurón** (vstupná hodnota x)
- Skrytá vrstva: **25 neurónov** s aktivačnou funkciou **tansig** (implicitne)
- Výstupná vrstva: **1 neurón** s aktivačnou funkciou **purelin**
- Trénovacia funkcia: **Levenberg-Marquardt (trainlm)**
- Vytvorená pomocou: `fitnet(25)`

### ◆ 4. Nastavené parametre trénovania (ukončovacie podmienky, použitá kriteriálna funkcia):

- Ukončovacia podmienka: chyba (goal) = **1e-4**
- Maximálny počet epoch: **250**
- Výpis chyby každých **5 epoch**

### ◆ 5. Najprv nastaviť počet neurónov v skrytej vrstve NS, čo spĺňa kvalitu aproximácie funkcie. Potom dať podstatne menší a väčší počet neurónov a zdokumentovať výsledky:

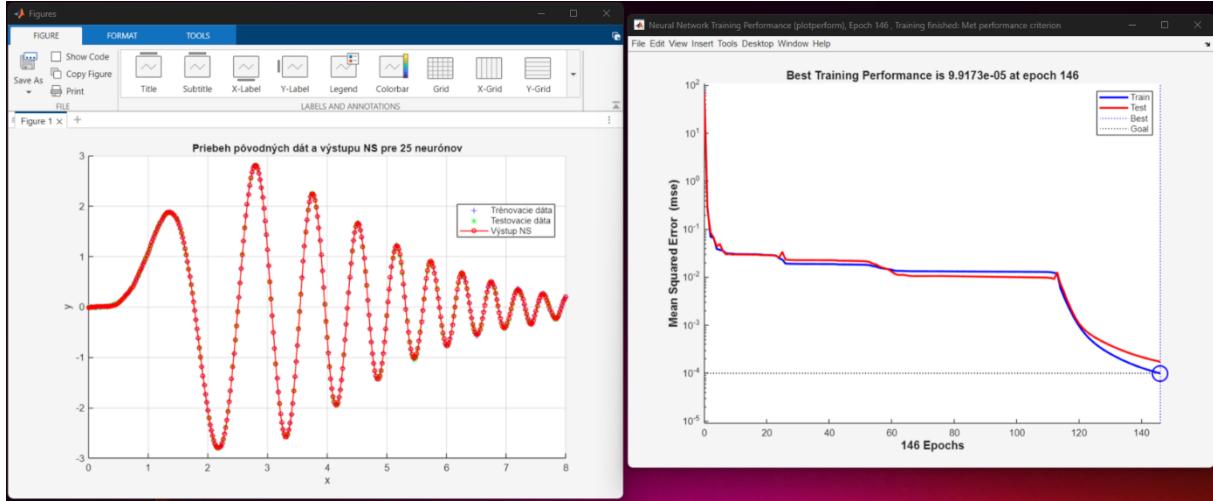
**Počet neurónov: 25** – bol náhodne zvolený a splnil podmienku chyby na testovacích dátach

- Výsledky boli porovnané aj s:
  - **menším počtom neurónov** (napr. 5) – nedostatočná aproximácia, vyššia chyba
  - **väčším počtom neurónov** (napr. 150) – funkcia bola preučená, vznikli lokálne výkyvy

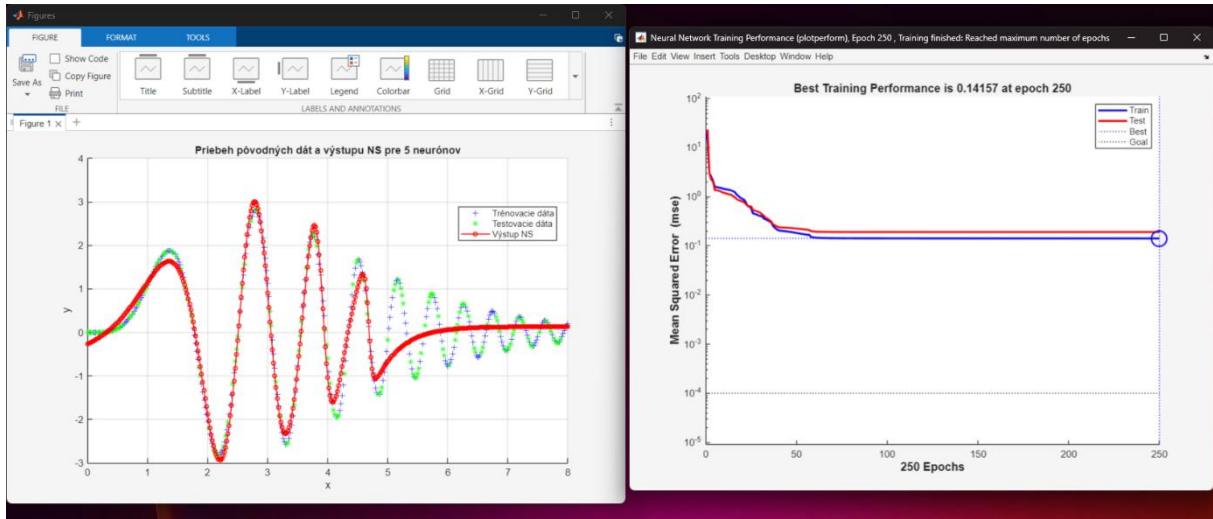
### ◆ 6. Grafy priebehu procesu trénovania aj s komentárom pre 3 nastavenia neurónov:

Sieť s 5 neurónmi nedokázala dosťatočne aproximovať cieľovú funkciu – tréning trval 250 epoch a chyba zostala vysoká. Naopak, 25 neurónov poskytlo výborný kompromis medzi jednoduchosťou a presnosťou. Sieť so 150 neurónmi spôsobila výkyvy v testovacej chybe v dôsledku pretrénovania. Najlepšie výsledky preto dosiahla sieť s 25 neurónmi, ktorá bola efektívna a presná zároveň.

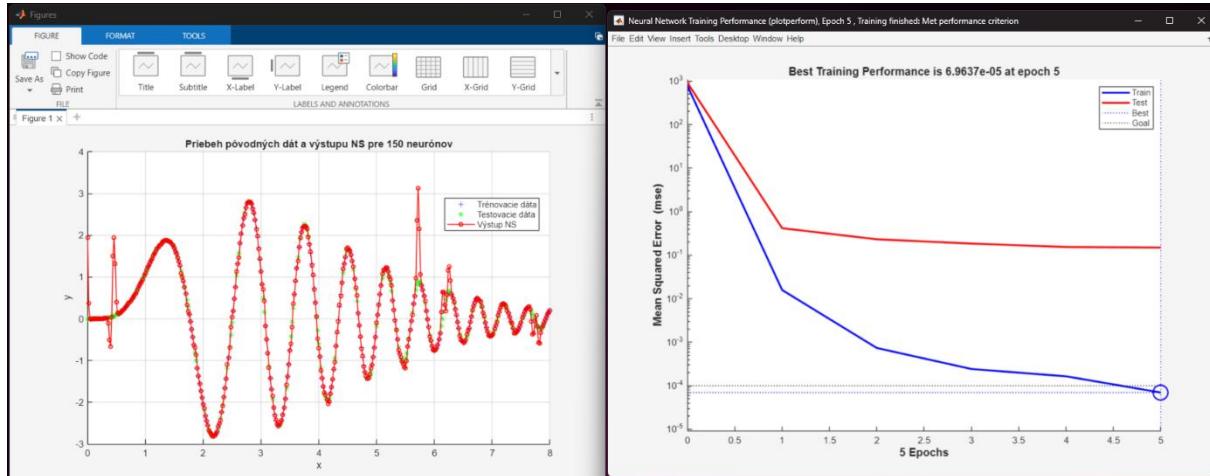
25



5



150



- ◆ 7. Porovnajte priebehy výstupov funkcie a NS a vyčísliť chyby SSE, MSE, MAE pre 3 nastavenia neurónov:

25

Chyby na tréningových dátach:  
SSE = 0.023802, MSE = 0.000099, MAE = 0.033346  
Chyby na testovacích dátach:  
SSE = 0.027980, MSE = 0.000174, MAE = 0.035525

5

Chyby na tréningových dátach:  
SSE = 33.976889, MSE = 0.141570, MAE = 1.639611  
Chyby na testovacích dátach:  
SSE = 30.802317, MSE = 0.191319, MAE = 1.595087

150

Chyby na tréningových dátach:  
SSE = 0.016713, MSE = 0.000070, MAE = 0.043051  
Chyby na testovacích dátach:  
SSE = 24.049747, MSE = 0.149377, MAE = 2.259114

## Úloha 6

---

◆ **1. Stručne vlastnými slovami, čo bolo úlohou (max. 3 vety):**

Cieľom úlohy bolo vytvoriť a natrénovať viacvrstvovú perceptrónovú (MLP) neurónovú sieť na klasifikáciu zdravotného stavu bábätka na základe CTG dát. Dátová sada obsahovala 25 vstupných parametrov z vyšetrení a cieľom bolo zaradiť každé meranie do jednej z troch tried (normálny, podozrivý, patologický). Výsledkom mala byť sieť s presnosťou klasifikácie na testovacích dátach vyššou ako 92 %.

---

◆ **2. Popis vstupných a výstupných dát, ako ste ich rozdelili na trénovacie, testovacie alebo aj validačné data:**

- **Vstupy:** 25-parametrové vektory z CTG vyšetrenia uložené v NDATA.
  - **Výstupy:** Triedy ochorenia (1 = normálny, 2 = podozrivý, 3 = patologický), konvertované na one-hot vektory.
  - **Rozdelenie dát:** Dáta boli rozdelené blokovo – vytvorilo sa 55 blokov pre každú triedu a na každý beh siete sa vybralo náhodne 60 % blokov pre trénovanie a 40 % pre testovanie
- 

◆ **3. Popis štruktúry MLP siete (vstupy, výstupy, typy neurónov v jednotlivých vrstvách, ich počet):**

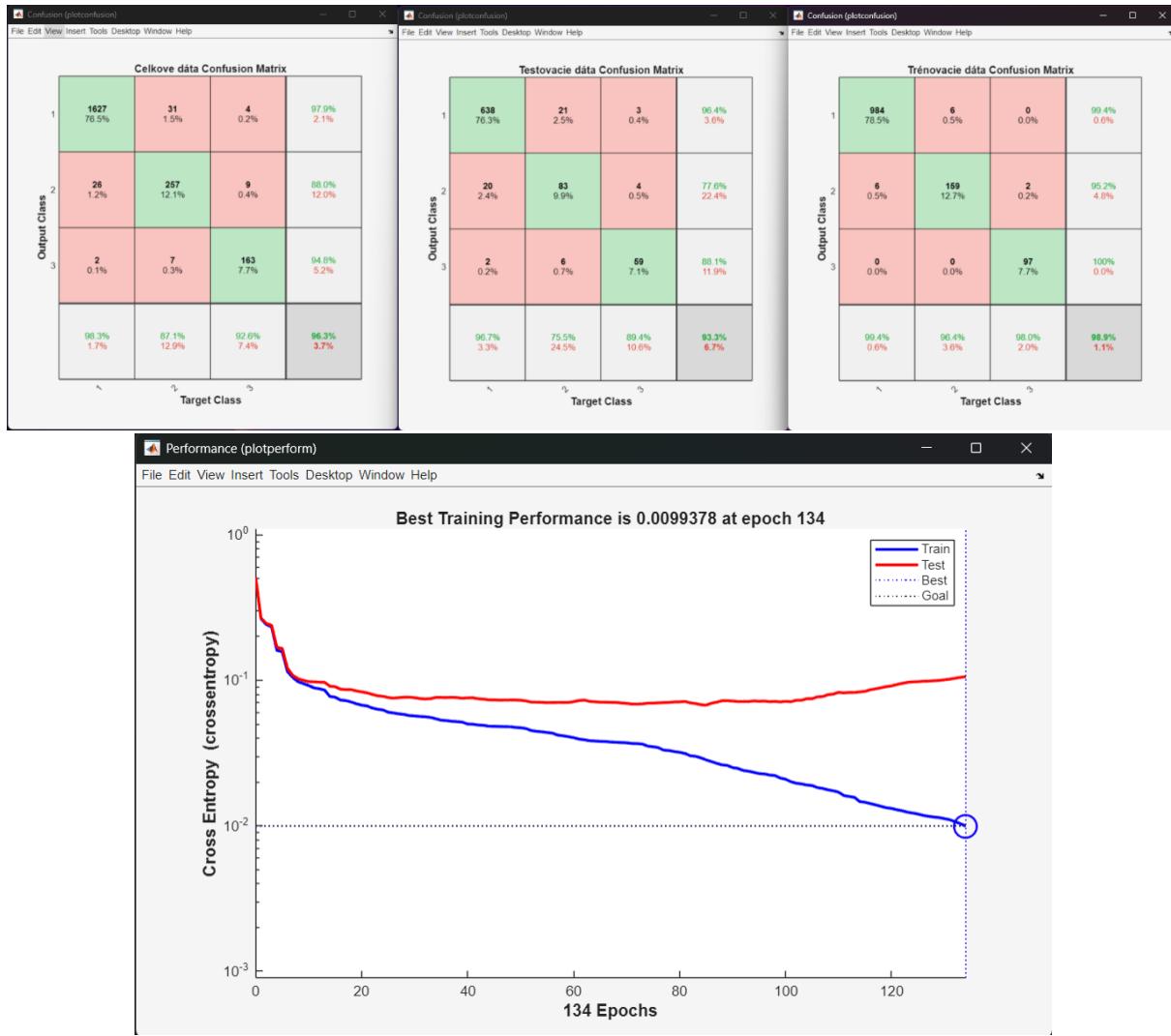
- **Počet vstupov:** 25 neurónov (každý zodpovedá jednému príznaku CTG).
  - **Skrytá vrstva:** 1 vrstva so **30 neurónmi** s aktivačnou funkciou tansig.
  - **Výstupná vrstva:** 3 neuróny pre 3 triedy, s aktivačnou funkciou softmax (pravdepodobnosť klasifikácia).
  - Použitá bola funkcia patternnet na klasifikáciu.
- 

◆ **4. Nastavené parametre trénovania (ukončovacie podmienky, použitá kriteriálna funkcia):**

- Ukončovacia podmienka: chyba goal = 1e-2
  - Maximálny počet epoch: 200
  - Kritériálna funkcia: **krížová entropia (crossentropy)**
  - Delenie dát: divideind (na základe indexov z blokového výberu)
- 

◆ **5. Najprv nastaviť počet neurónov v skrytej vrstve NS, čo splňa požiadavky klasifikácie. Zdokumentovať výsledky pre najlepšie trénovanie. (proces trénovania, kontingenčná matica)**

- Najlepšie výsledky boli dosiahnuté s **30 neurónmi**.
- Výsledné kontingenčné matice ukazujú vysokú presnosť, napr.:
  - **Trénovacia presnosť:** 98.9 %
  - **Testovacia presnosť:** 93.3 %
  - **Celková presnosť:** 96.3 %



- ◆ 6. Pri tomto nastavení parametrov urobiť krížovú validáciu alebo 5 krát spustiť trénovanie s náhodným rozdelením dát a vyčísiť min, max, priemernú úspešnosť klasifikácie:

- Výsledky testovacej presnosti po 5 opakovaniach:

--- Presnosti ---

Testovacia presnosť: MIN 91.51 % | MAX 93.30 % | PRIEMER 92.39 %

Trénovacia presnosť: MIN 98.88 % | MAX 99.60 % | PRIEMER 99.23 %

Celkova presnosť: MIN 95.91 % | MAX 96.28 % | PRIEMER 96.15 %

- ◆ 7. Potom ukázať dve iné nastavenie parametrov NS a zdokumentovať výsledky (proces trénoania, kontingenčná matica) (stačí jedno trénovanie)

- 5 neurónov – výstup neboli dostatočne presné, siet' sa nenaučila dobre triediť podozrivé prípady.



```
== Výsledky po 5 opakovaniach ==
Najlepšia sieť bola v behu č. 5 (počet neurónov: 5)
```

```
-- Presnosti --
```

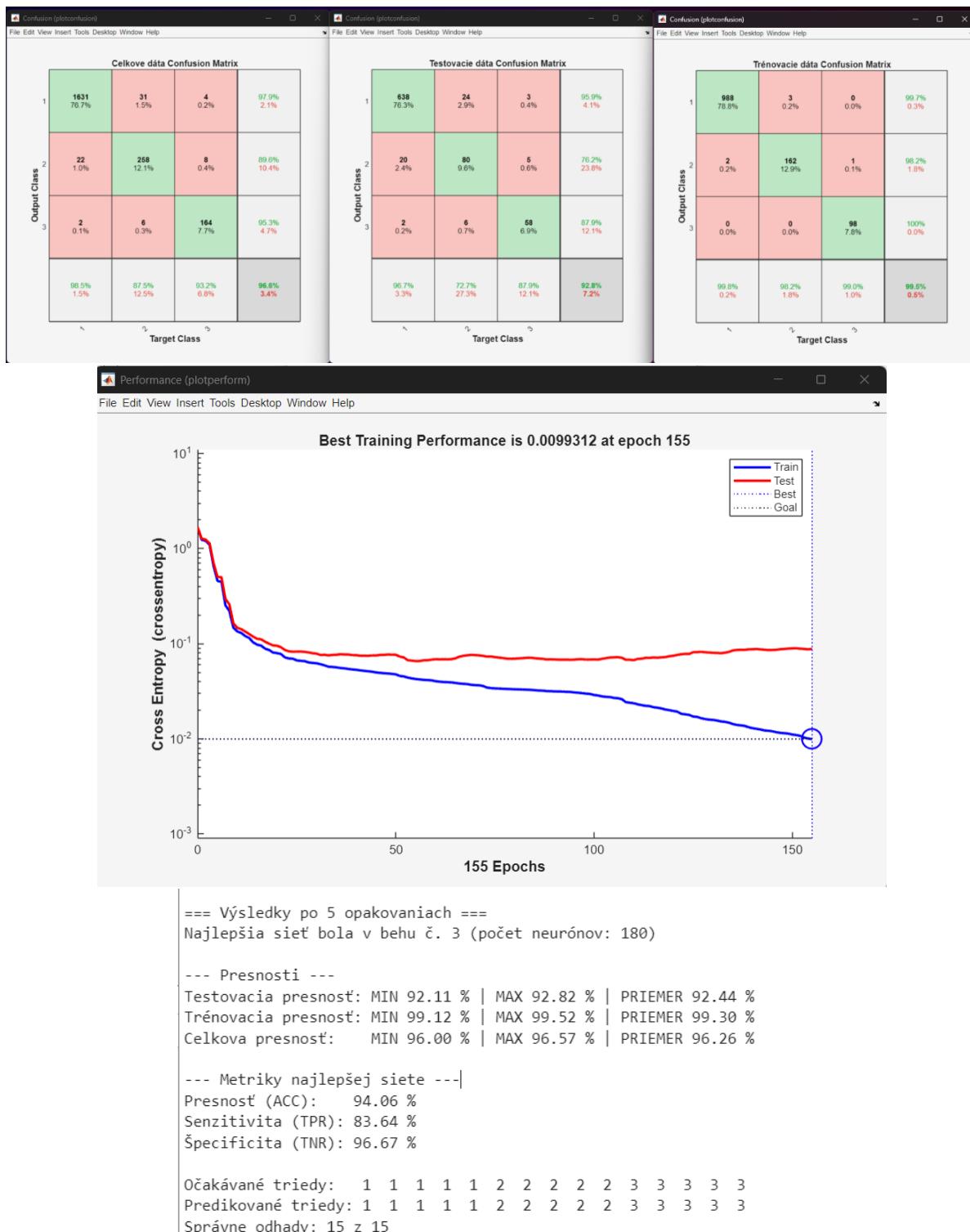
```
Testovacia presnosť: MIN 91.27 % | MAX 93.18 % | PRIEMER 92.08 %
Trénovacia presnosť: MIN 93.14 % | MAX 95.37 % | PRIEMER 94.39 %
Celkova presnosť: MIN 92.71 % | MAX 93.84 % | PRIEMER 93.25 %
```

```
-- Metriky najlepšej siete --
```

```
Presnosť (ACC): 94.88 %
Senzitivita (TPR): 86.96 %
Špecificka (TNR): 96.82 %
```

```
Očakávané triedy: 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Predikované triedy: 1 2 1 1 1 1 2 2 2 1 3 3 3 3 3
Správne odhady: 12 z 15
```

2. **180 neurónov** – veľmi vysoká presnosť, ale pomalšie trénovanie a tendencia k pretrénovaniu.



#### ◆ 8. Postup testovania vybraných vzoriek:

- Bolo vybraných 5 vzoriek z každej triedy.
- Pre tieto vzorky sa zistovali očakávané a predikované triedy pomocou sim(net, testSamples) a porovnávali sa.
- Výsledok (napr.):

Očakávané triedy: 1 1 1 1 1 2 2 2 2 3 3 3 3  
Predikované triedy: 1 1 1 1 1 2 2 2 2 3 3 3 3 2  
Správne odhady: 14 z 15  
..

---

◆ 9. Pre najlepšie natrénovanú sieť, vypočítať úspešnosť klasifikácie, senzitivitu a špecifiku. (1. skupina – negatívny, 2. a 3. skupina – pozitívny)

--- Metriky najlepšej siete ---

Presnosť (ACC): 94.43 %

Senzitivita (TPR): 85.54 %

Špecificka (TNR): 96.67 %

## Úloha 7

### 1. Stručne vlastnými slovami, čo bolo úlohou (max. 3 vety):

Úlohou bolo natrénovať MLP sieť na rozpoznávanie rukou písaných číslic (0–9) na základe dát MNIST. Každý obrázok bol reprezentovaný ako vektor s 784 vstupmi (28×28 pixelov). Cieľom bolo dosiahnuť klasifikačnú presnosť väčšiu ako 95 %.

---

### 2. Popis vstupných a výstupných dát, ako ste ich rozdelili na trénovacie, testovacie alebo aj validačné dátá:

- **Vstupné dátá:** Obrázky číslic vo formáte 28×28 pixelov rozložené do vektora veľkosti 784 prvkov (XDataall)
- **Výstupné dátá:** Triedy číslic (0–9) zakódované pomocou one-hot reprezentácie (ind2vec)
- **Rozdelenie dát:**
  - Trénovacie dátá: 60 %
  - Testovacie dátá: 40 %
  - Validačné dátá sa nepoužívali (valRatio = 0)

---

### 3. Popis štruktúry MLP siete (vstupy, výstupy, typy neurónov v jednotlivých vrstvách, ich počet):

- **Vstupná vrstva:** 784 vstupov (jeden pre každý pixel)
- **Skrytá vrstva:** 40 neurónov, aktivačná funkcia tansig (štandardne pri patternnet)
- **Výstupná vrstva:** 10 výstupov (pre čísllice 0–9), aktivačná funkcia softmax

---

### 4. Nastavené parametre trénovalia (ukončovacie podmienky, použitá kriteriálna funkcia):

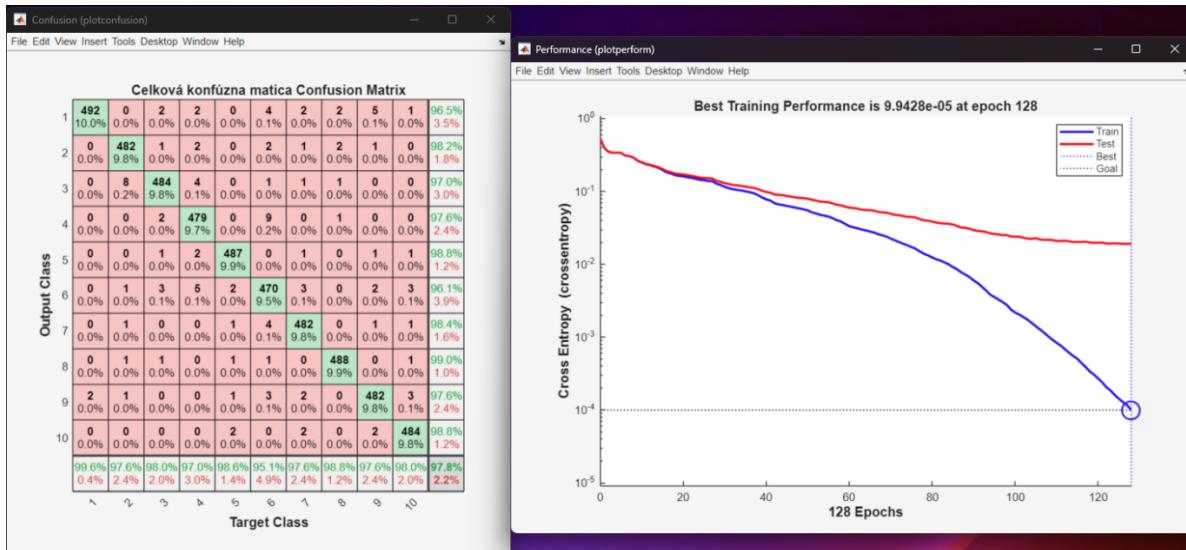
- Trénovacia metóda: Levenberg-Marquardt (trainlm)
- **Počet epoch:** 300
- **Cielová chyba (goal):** 1e-4
- **Kriteriálna funkcia:** krížová entropia (v patternnet predvolené)

---

### 5. Najprv nastaviť počet neurónov v skrytej vrstve NS, čo spĺňa požiadavky klasifikácie.

Následne ladiť parametre trénovalia. Zdokumentovať výsledky pre najlepšie trénovalie (proces trénovalia, kontingenčná matica):

- Nastavený počet neurónov: **40**
- Tréning prebiehal 5-krát s náhodným rozdelením dát.
- Výber najlepšej siete bol na základe najvyššej testovacej presnosti.
- Výstupy siete boli vyhodnotené pomocou plotconfusion a plotperform.



## 6. Pri tomto nastavení parametrov urobiť krížovú validáciu alebo 5 krát spustiť trénovanie s náhodným rozdelením dát a vyčísliť min, max, priemernú úspešnosť klasifikácie:

- Bolo realizovaných 5 behov tréningu.
- Výsledky:

Výsledky po 5 behoch

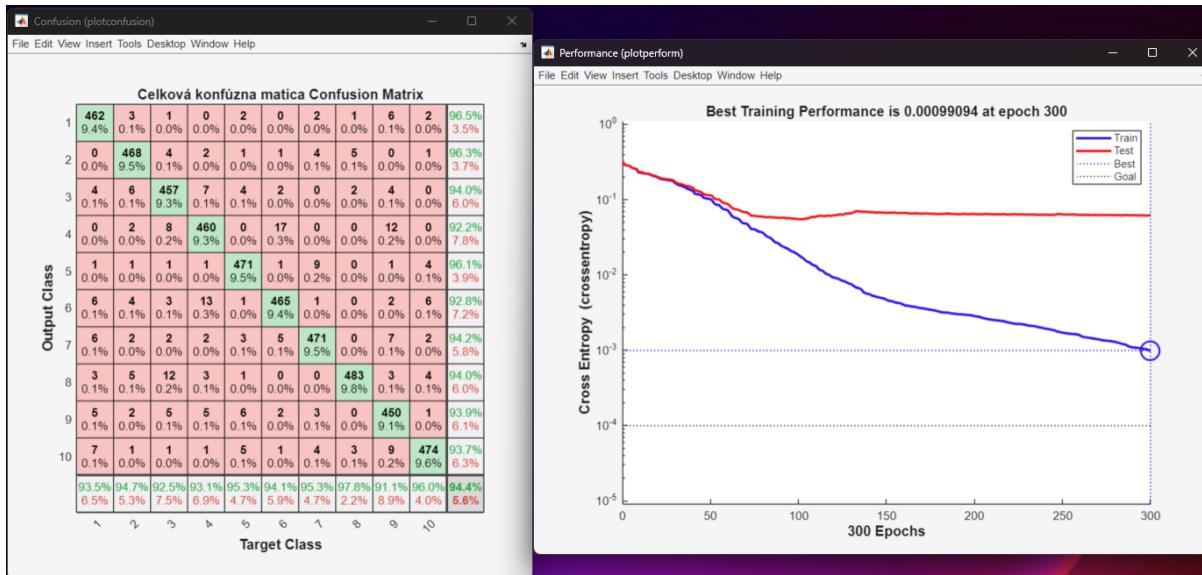
Testovacia presnosť: MIN 93.42 % | MAX 94.43 % | PRIEMER 93.94 %

Trénovacia presnosť: MIN 100.00 % | MAX 100.00 % | PRIEMER 100.00 %

Najlepšia sieť (beh č. 4)

## 7. Vedieť ukázať aj iné nastavenie parametrov NS:

- 10 neurónov



#### Výsledky po 5 behoch

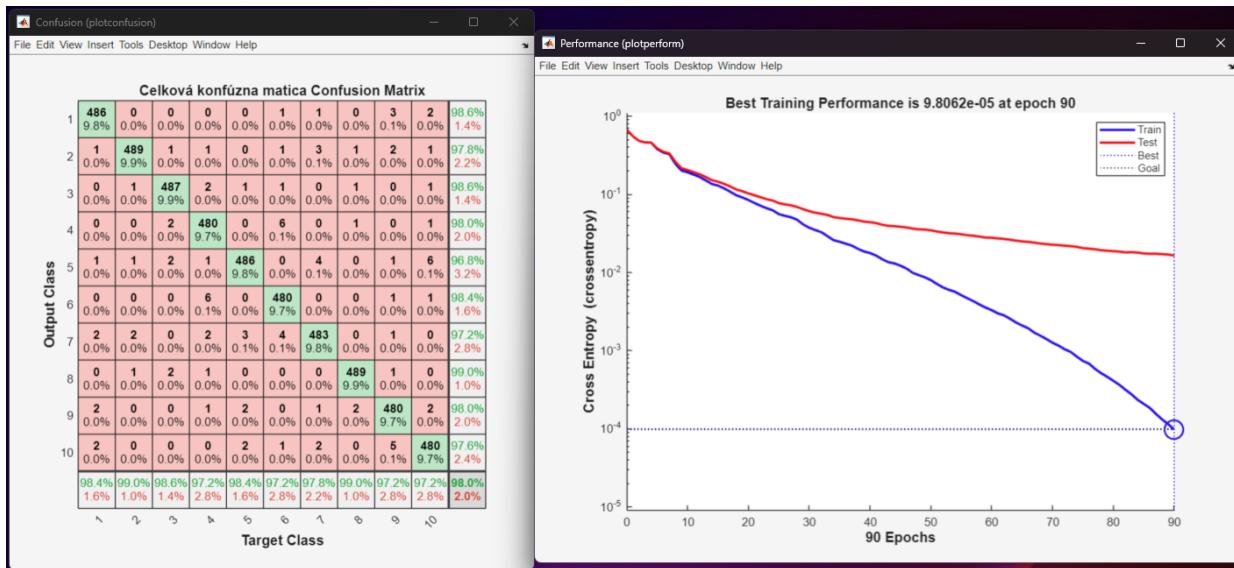
Testovacia presnosť: MIN 71.15 % | MAX 86.03 % | PRIEMER 82.01 %  
Trénovacia presnosť: MIN 95.28 % | MAX 99.97 % | PRIEMER 98.99 %

Najlepšia sieť (beh č. 2)

#### Testovanie vzoriek

Vzorka 1 -> Výstup siete: [1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 0  
Vzorka 2 -> Výstup siete: [0.00 0.99 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00] -> Predikcia: 1  
Vzorka 3 -> Výstup siete: [0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 2  
Vzorka 4 -> Výstup siete: [0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.94 0.03 0.00] -> Predikcia: 7  
Vzorka 5 -> Výstup siete: [0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 4  
Vzorka 6 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00] -> Predikcia: 5  
Vzorka 7 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00] -> Predikcia: 6  
Vzorka 8 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00] -> Predikcia: 7  
Vzorka 9 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00] -> Predikcia: 8  
Vzorka 10 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00] -> Predikcia: 9  
Skutočné čísla: 0 1 2 3 4 5 6 7 8 9  
Predikované čísla: 0 1 2 7 4 5 6 7 8 9  
Správne klasifikovaných: 9 z 10

### • 130 neurónov



#### Výsledky po 5 behoch

Testovacia presnosť: MIN 93.67 % | MAX 94.94 % | PRIEMER 94.29 %  
Trénovacia presnosť: MIN 100.00 % | MAX 100.00 % | PRIEMER 100.00 %

Najlepšia sieť (beh č. 5)

#### Testovanie vzoriek

Vzorka 1 -> Výstup siete: [1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 0  
Vzorka 2 -> Výstup siete: [0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 1  
Vzorka 3 -> Výstup siete: [0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 2  
Vzorka 4 -> Výstup siete: [0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 3  
Vzorka 5 -> Výstup siete: [0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00] -> Predikcia: 4  
Vzorka 6 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00] -> Predikcia: 5  
Vzorka 7 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00] -> Predikcia: 6  
Vzorka 8 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00] -> Predikcia: 7  
Vzorka 9 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00] -> Predikcia: 8  
Vzorka 10 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00] -> Predikcia: 9  
Skutočné čísla: 0 1 2 3 4 5 6 7 8 9  
Predikované čísla: 0 1 2 3 4 5 6 7 8 9  
Správne klasifikovaných: 10 z 10

## 8. Postup testovania vybraných vzoriek:

- Bolo vybraných 10 vzoriek (jedna pre každú číslicu 0–9).
- Vzorky boli prehnané najlepšou natrénovanou sieťou.
- Pre každú vzorku sa vypísal výstup siete (pravdepodobnosti) a výsledná predikcia čísla.
- Vyhodnotila sa správnosť klasifikácie (porovnanie očakávaného a predikovaného čísla).

Testovanie vzoriek

Vzorka 1 -> Výstup siete: [1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ] -> Predikcia: 0  
Vzorka 2 -> Výstup siete: [0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ] -> Predikcia: 1  
Vzorka 3 -> Výstup siete: [0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ] -> Predikcia: 2  
Vzorka 4 -> Výstup siete: [0.00 0.00 0.00 0.99 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ] -> Predikcia: 3  
Vzorka 5 -> Výstup siete: [0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 ] -> Predikcia: 4  
Vzorka 6 -> Výstup siete: [0.00 0.00 0.01 0.12 0.00 0.87 0.00 0.00 0.00 0.00 0.00 ] -> Predikcia: 5  
Vzorka 7 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 ] -> Predikcia: 6  
Vzorka 8 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 ] -> Predikcia: 7  
Vzorka 9 -> Výstup siete: [0.03 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.97 0.00 0.00 ] -> Predikcia: 8  
Vzorka 10 -> Výstup siete: [0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 ] -> Predikcia: 9  
Skutočné čísla: 0 1 2 3 4 5 6 7 8 9  
Predikované čísla: 0 1 2 3 4 5 6 7 8 9  
Správne klasifikovaných: 10 z 10

## Úloha 8

### 1. Stručne vlastnými slovami, čo bolo úlohou (max. 3 vety):

Úlohou bolo vytvoriť a natrénovať konvolučnú neurónovú sieť (CNN) na rozpoznávanie rukou písaných číslíc (0–9) z obrázkov MNIST. Cieľom bolo dosiahnuť vysokú klasifikačnú úspešnosť na testovacích dátach a porovnať výsledky CNN so sieťou MLP. Tréning mal byť opakovaný minimálne 5-krát pre overenie stability výsledkov.

---

### 2. Popis vstupných a výstupných dát, ako ste ich rozdelili na trénovacie, testovacie alebo aj validačné data:

- **Vstupné dátá:** Obrázky číslíc vo formáte  $28 \times 28$  pixelov, vstup CNN mal tvar  $28 \times 28 \times 1$  (Ximgs)
- **Výstupné dátá:** Triedy číslic 0–9 (kategorizované pomocou categorical)
- **Rozdelenie dát:**
  - Trénovacie dátá: 60 %
  - Testovacie dátá: 40 %
  - Validačné dátá sa nepoužívali

---

### 3. Popis štruktúry CNN siete (vstupy, výstupy, jednotlivé vrstvy):

- **Vstupná vrstva:** imageInputLayer([28 28 1])
- **Konvolučné vrstvy:**
  - Prvá konvolučná vrstva: 6 filtrov veľkosti  $5 \times 5$
  - Dávková normalizácia a ReLU
  - Druhá konvolučná vrstva: 12 filtrov veľkosti  $5 \times 5$
  - Dávková normalizácia, ReLU a Max pooling ( $2 \times 2$ , krok 2)
- **Plne prepojené vrstvy:**
  - fullyConnectedLayer(32), následne dropout(0.5)
  - fullyConnectedLayer(10)
- **Výstupné vrstvy:** softmaxLayer, classificationLayer

---

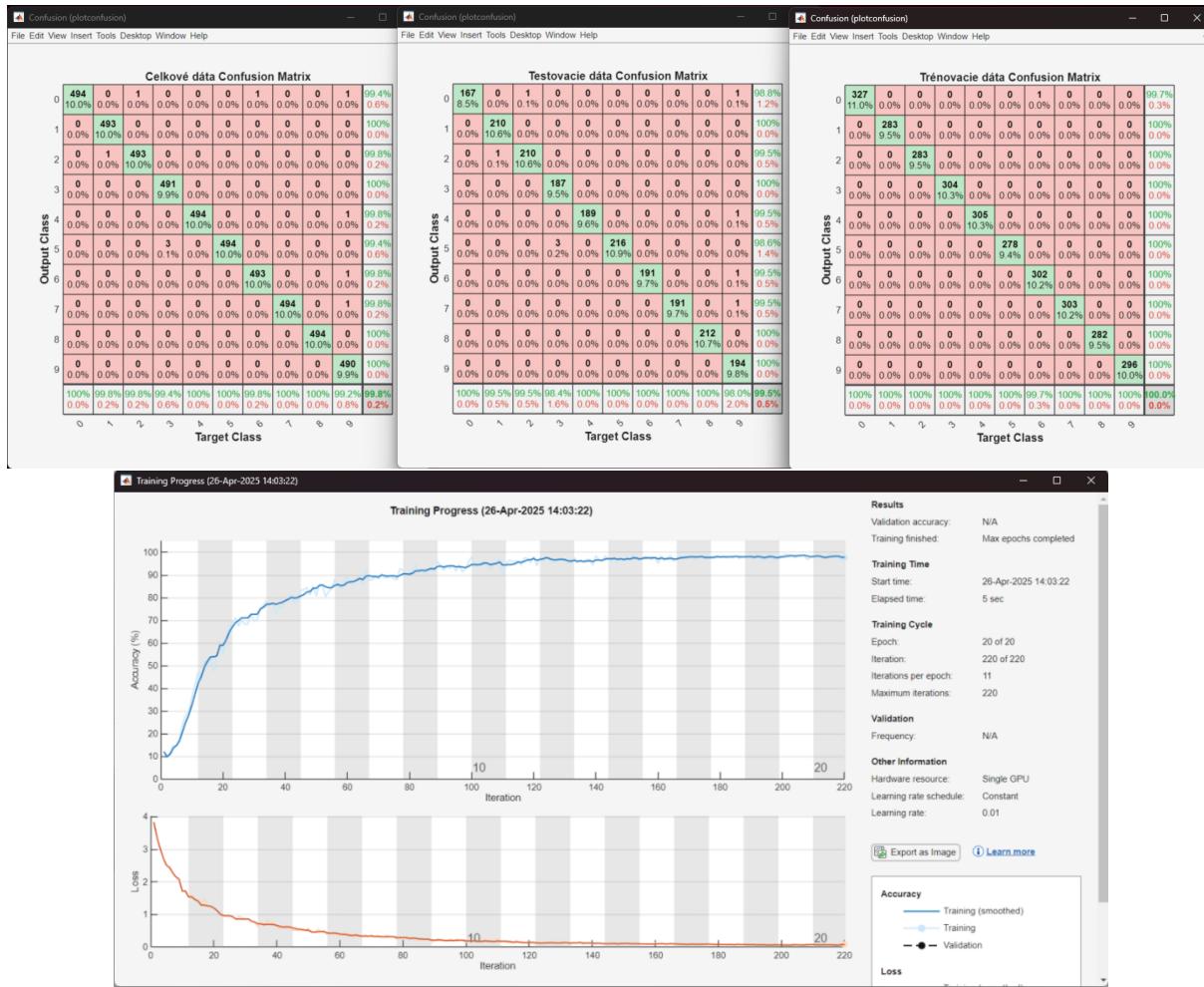
### 4. Nastavené parametre trénovalia (ukončovacie podmienky, použitá kriteriálna funkcia, krok učenia, veľkosť dávky, ...):

- **Optimalizačný algoritmus:** sgdm (Stochastic Gradient Descent with Momentum)
- **Počet epoch:** 20
- **Počiatočná rýchlosť učenia (InitialLearnRate):** 0.01
- **Veľkosť dávky (MiniBatchSize):** 256
- **Miešanie dát (Shuffle): raz pred začiatkom trénovalia** (once)
- **Kriteriálna funkcia:** Kategorizovaná krízová entropia (implicitne v classificationLayer)

---

### 5. Najprv zvoliť architektúru CNN (2 alebo 3 konvolučné vrstvy), čo spĺňa požiadavky klasifikácie. Následne ladiť parametre trénovalia. Zdokumentovať výsledky pre najlepšie trénovalie (proces trénovalia, kontingenčná matica, úspešnosť klasifikácie trénovacie a testovacie data):

- Použité boli **2 konvolučné vrstvy**. (vid. kód na konci strany)
- Po 5 behoch bola vybraná najlepšia sieť na základe najvyššej testovacej presnosti.
- Výsledky boli zdokumentované pomocou plotconfusion (pre trénovacie, testovacie a celkové dátá) a priebehu učenia (training-progress).



**6. Pri tomto nastavení parametrov minimálne 3 krát (ideálne 5 krát) spustiť trénovanie s náhodným rozdelením dát a vyčísiť min, max, priemernú úspešnosť klasifikácie:**

- Bolo realizovaných 5 behov tréningu.
  - Výsledky (*vid. kód na konci strany*):

--- Výsledky po 5 behoch ---  
Presnosť: MIN 99.14 % | MAX 99.54 % | PRIEMER 99.34 %  
Najlepšia siet' bola natrénovaná v behu č. 2

#### **7. Vedieť ukázať aj iné nastavenie parametrov CNN:**

## Príklad – úprava CNN architektúry:

- Zväčšenie počtu filtrov na 8 a 16
  - Pridanie tretej konvolučnej vrstvy (napr. 24 filtrov)

```
layers = [  
    imageInputLayer([28 28 1])
```

```
convolution2dLayer(5, 8)      % 8 filtrov, 5x5  
batchNormalizationLayer  
reluLayer()
```

```
convolution2dLayer(5, 16)    % 16 filtrov, 5x5
batchNormalizationLayer
reluLayer()

convolution2dLayer(3, 24)    % Pridaná 3. konvolučná vrstva: 24 filtrov, 3x3
batchNormalizationLayer
reluLayer()
maxPooling2dLayer(2, 'Stride', 2)
fullyConnectedLayer(32)
dropoutLayer(0.5)
fullyConnectedLayer(10)
softmaxLayer()
classificationLayer()

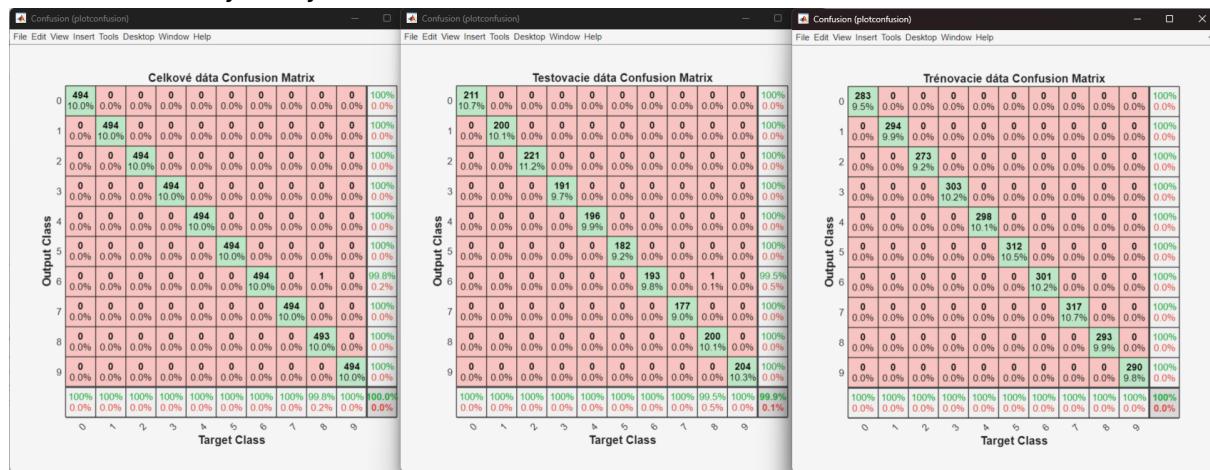
];
```

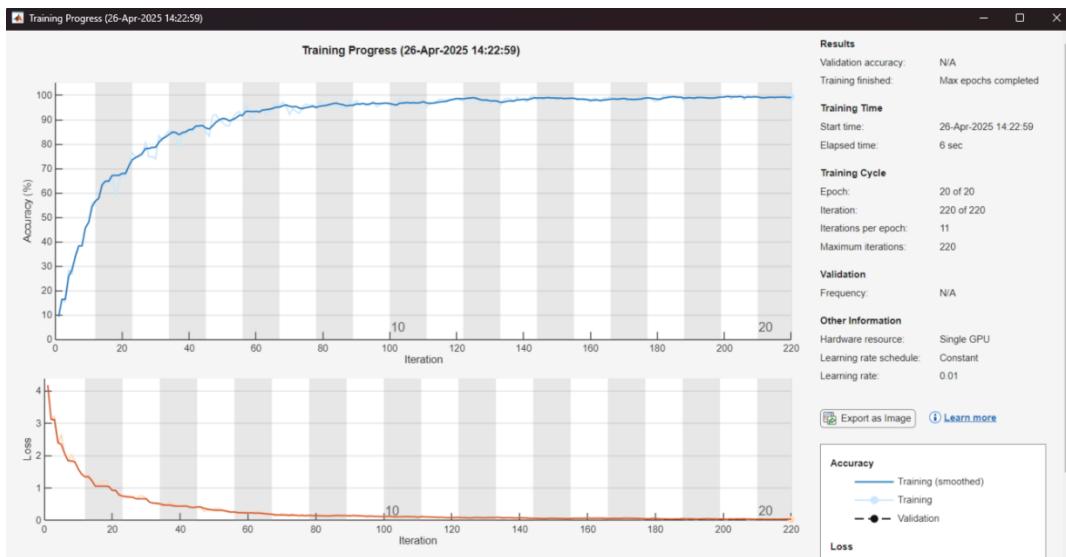
## Dopad tejto zmeny:

- Väčší počet filtrov a pridaná tretia konvolučná vrstva umožnia sieti zachytiť viac detailov a komplexnejších vzorov na obrázkoch.
  - Väčšinou vedie k zvýšeniu presnosti na trénovacích aj testovacích dátach, hlavne pri zložitejších úlohách.
  - Nevýhoda: vyšia výpočtová náročnosť (dlhší tréning), a ak je sieť príliš veľká na jednoduché dátá, hrozí pretrénovanie.

**8. Porovnať dve štruktúry CNN (proces trénovania, kontingenčná matica, úspešnosť klasifikácie trénovacie a testovacie data):**

- 2 konvolučné vrstvy (Výsledky vyššie a zhodnotenie klasifikácie v otázke 10)
  - 3 konvolučné vrstvy
    - Výsledky:



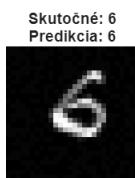
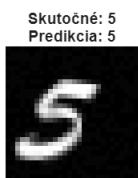
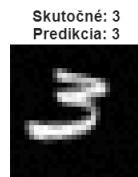
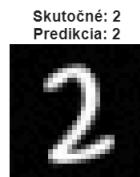
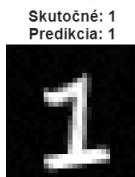


== Výsledky po 5 behoch ==

Presnosť: MIN 99.65 % | MAX 99.95 % | PRIEMER 99.81 %

Najlepšia sieť bola natrénovaná v behu č. 2

--- Testovanie vzoriek ---

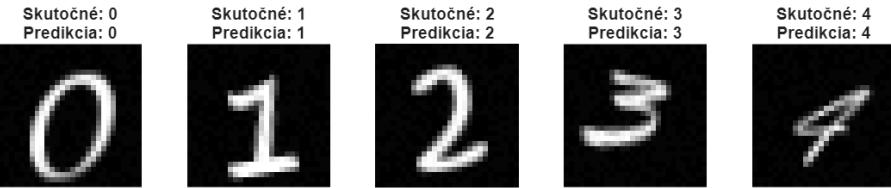


## 9. Porovnať MLP a CNN sieť (proces trénovalia, kontingenčná matica, úspešnosť klasifikácie trénovalie a testovacie data):

- MLP sieť (Úloha 7): dosahovala presnosť okolo 95–97 %, závislá od počtu neurónov.
- CNN sieť (Úloha 8): dosiahla vyššiu presnosť (nad 99 %), lepšie zvláda priestorové vlastnosti obrázkov a generalizuje lepšie na nové dátu.

## 10. Postup testovania vybraných vzoriek:

- Z každej triedy (0–9) bola vybraná jedna vzorka.
- Obrázok bol zobrazený (imshow) a sieťou klasifikovaný (classify).
- Výsledok obsahoval správne číslo aj predikované číslo. (vid. kód na konci strany)



**\*kód z Matlabu:**

```
%Definícia CNN štruktúry
layers = [
    imageInputLayer([28 28 1]) % vstupná vrstva – obraz 28x28x1
    convolution2dLayer(5, 6) % 2D konvolúcia – 6 filtrov, rozmer 5x5
    batchNormalizationLayer % dávková normalizácia
    reluLayer() % ReLU funkcia

    convolution2dLayer(5, 12) % 2D konvolúcia – 12 filtrov, rozmer 5x5
    batchNormalizationLayer % dávková normalizácia
    reluLayer() % ReLU funkcia
    maxPooling2dLayer(2, 'Stride', 2) % max pooling – 2x2, krok 2
    fullyConnectedLayer(32) % plne prepojená vrstva – 32 neurónov
    dropoutLayer(0.5) % dropout vrstva
    fullyConnectedLayer(10) % plne prepojená vrstva – 10 neurónov
    softmaxLayer() % softmax aktivačná funkcia
    classificationLayer() % klasifikačná vrstva – 10 tried
];
```