**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 dáta:**

* **Vstupné dáta:** 3D súradnice bodov piatich skupín, každá skupina obsahovala 50 bodov.
* **Výstupné dáta:** čísla tried 0–4 reprezentujúce 5 skupín, ktoré sú zakódované pomocou one-hot reprezentácie.
* **Rozdelenie dát:**
  + Trénovacie dáta: 80%
  + Testovacie dáta: 20%
  + Validačné dáta 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 trénovania (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 trénovania aj s komentárom:**

Obrázok, na ktorom je text, snímka obrazovky, vývoj, rad

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

**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:**

Obrázok, na ktorom je text, diagram, snímka obrazovky, plán

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

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 potvrdzuje 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:

*Obrázok, na ktorom je text, písmo, snímka obrazovky, rad

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.*

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

*Obrázok, na ktorom je text, snímka obrazovky, diagram

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.*

**Úloha 5b**

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

Cieľom úlohy bolo navrhnúť, natrénovať 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 dáta**: premenná x – jednorozmerné hodnoty vstupnej funkcie.
* **Výstupné dáta**: premenná y – výstupy meranej nelineárnej funkcie f(x).
* **Rozdelenie**:
  + Trénovacie dáta: podľa indexov indx\_train (60 % vzoriek)
  + Testovacie dáta: 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:**

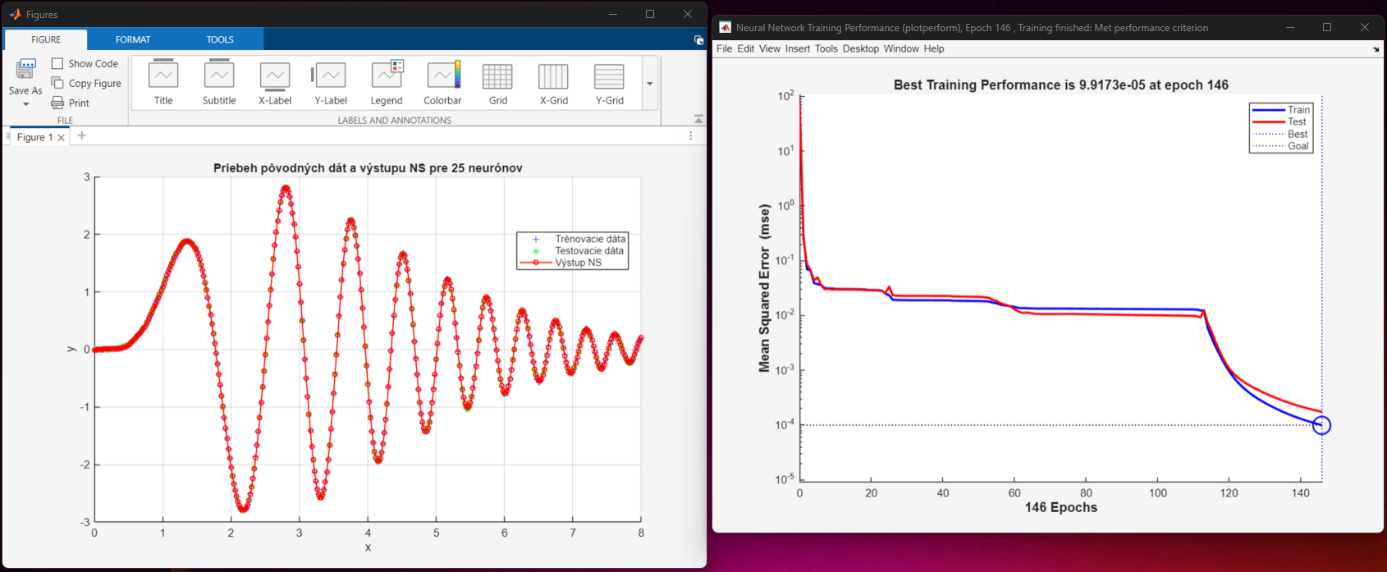
**Počet neurónov: 25** – bol náhodne zvolený a splnil podmienku chyby na testovacích dátach

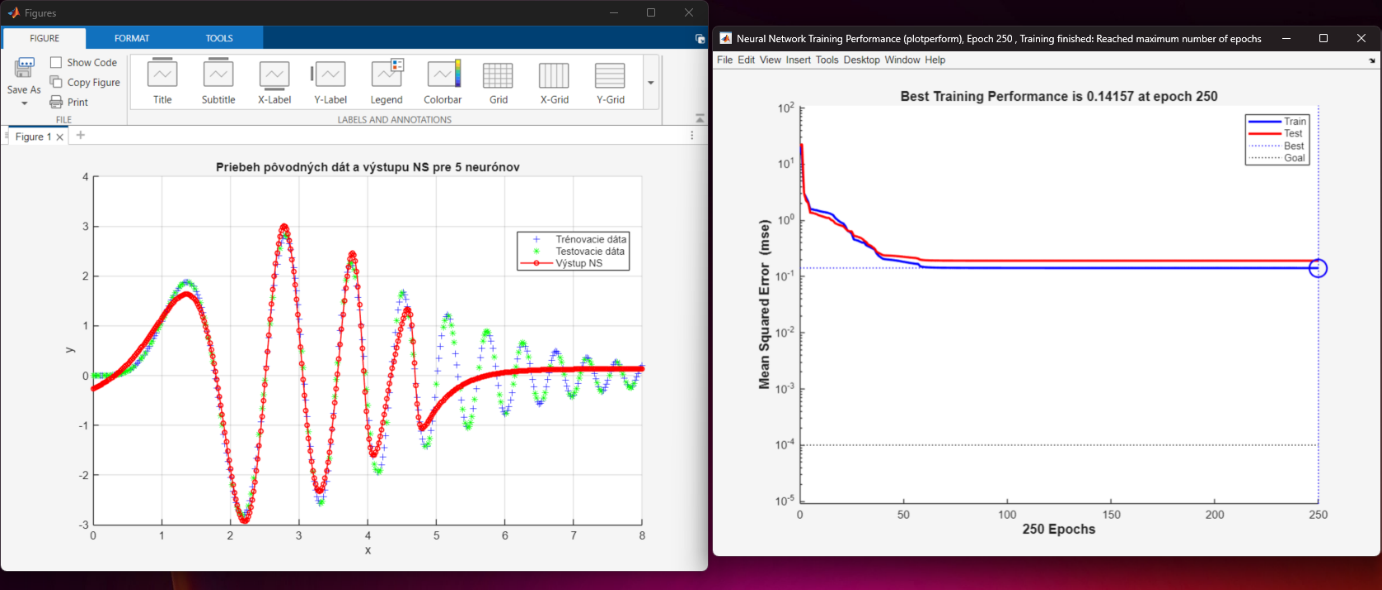
* Výsledky boli porovnané aj s:
  + **menším počtom neurónov** (napr. 5) – nedostatočná aproximácia, vyššia chyba
  + **väčším počtom neurónov** (napr. 150) – funkcia bola preučená, vznikli lokálne výkyvy

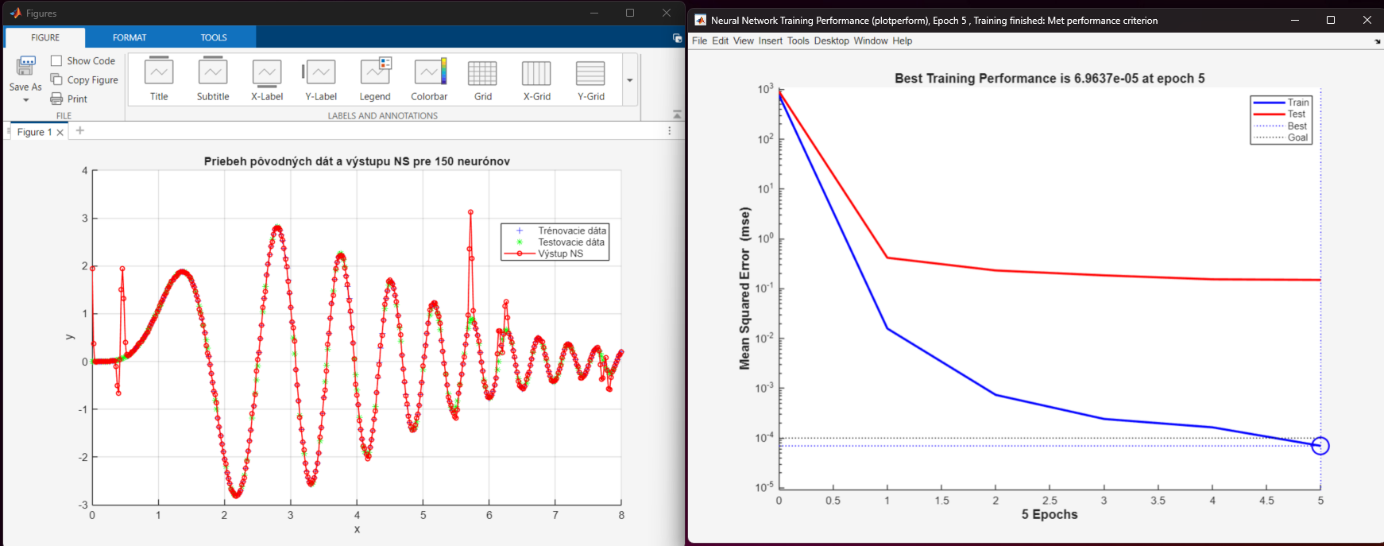
**🔹 6. Grafy priebehu procesu trénovania aj s komentárom pre 3 nastavenia neurónov:**

Sieť s 5 neurónmi nedokázala dostatočne 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

Obrázok, na ktorom je text, písmo, snímka obrazovky, rad

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

5

Obrázok, na ktorom je text, písmo, snímka obrazovky, rad

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

150

Obrázok, na ktorom je text, písmo, snímka obrazovky, rad

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

**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):**

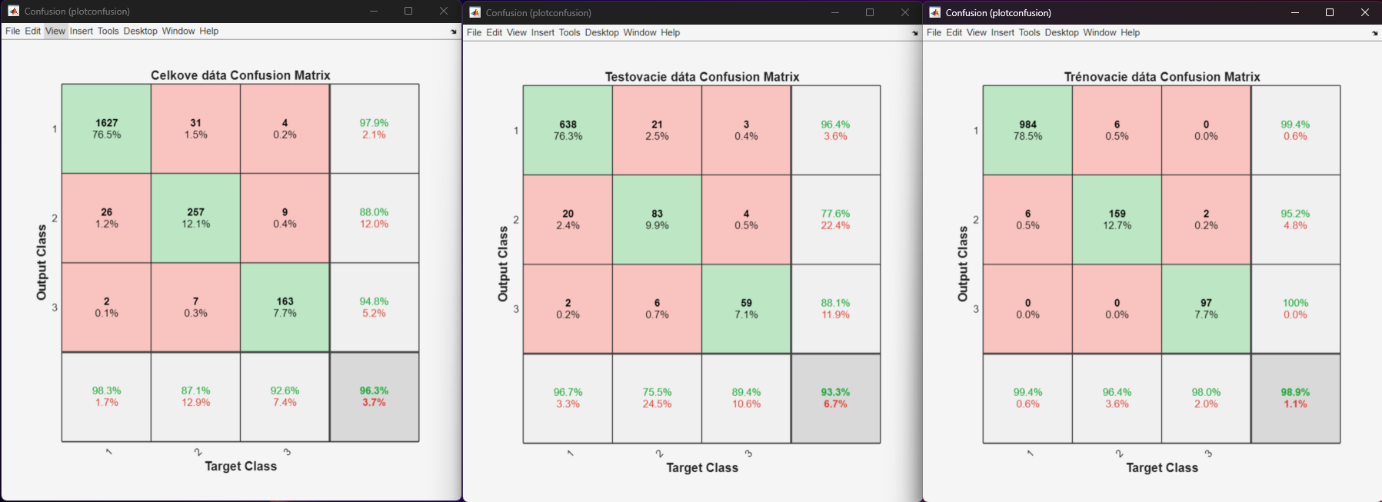
* **Počet vstupov**: 25 neurónov (každý zodpovedá jednému príznaku CTG).
* **Skrytá vrstva**: 1 vrstva so **30 neurónmi** s aktivačnou funkciou tansig.
* **Výstupná vrstva**: 3 neuróny pre 3 triedy, s aktivačnou funkciou softmax (pravdepodobnostná klasifikácia).
* Použitá bola funkcia patternnet na klasifikáciu.

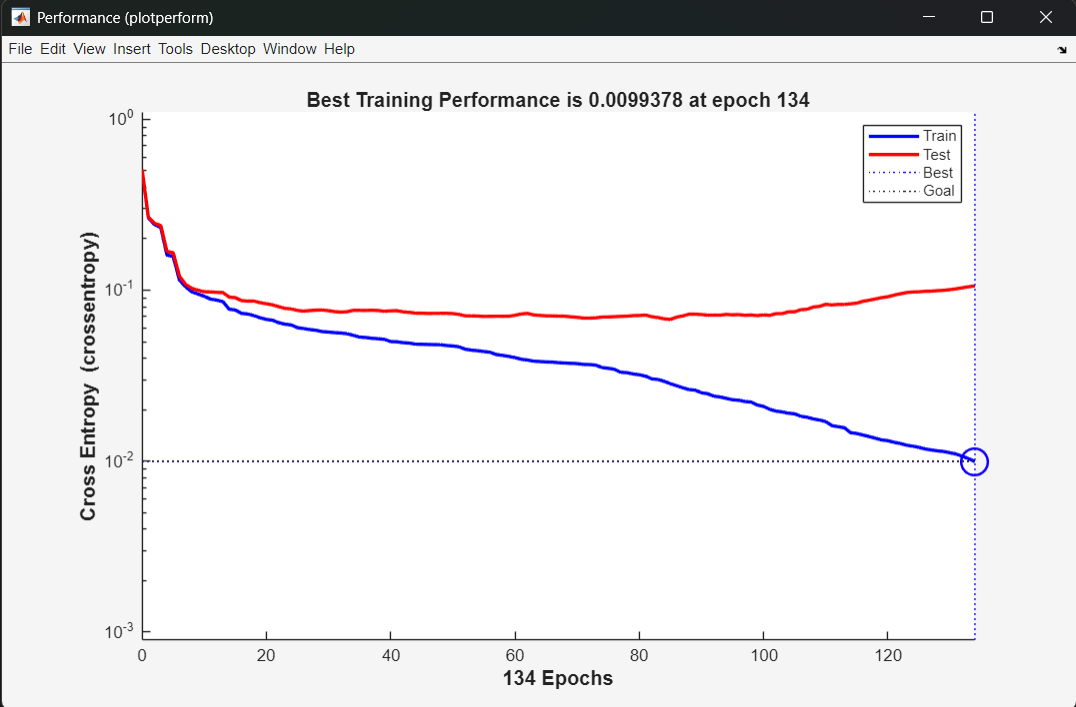
🔹 **4. Nastavené parametre trénovania (ukončovacie podmienky, použitá kriteriálna funkcia):**

* 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 spĺňa požiadavky klasifikácie. Zdokumentovať výsledky pre najlepšie trénovanie. (proces trénovania, kontingenčná matica)**

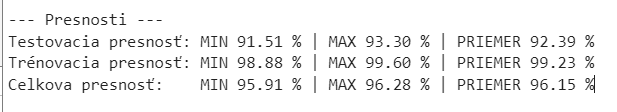
* Najlepšie výsledky boli dosiahnuté s **30 neurónmi**.
* Výsledné kontingenčné matice ukazujú vysokú presnosť, napr.:
  + **Trénovacia presnosť**: 98.9 %
  + **Testovacia presnosť**: 93.3 %
  + **Celková presnosť**: 96.3 %

****



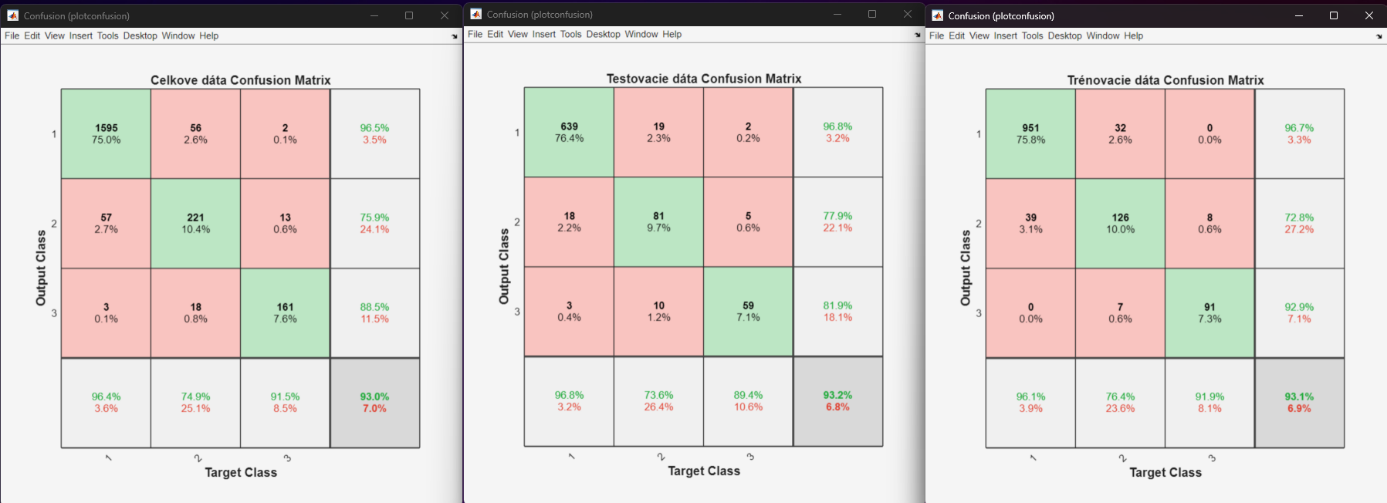
🔹 **6. Pri tomto nastavení parametrov urobiť krížovú validáciu alebo 5 krát spustiť trénovanie s náhodným rozdelením dát a vyčísliť min, max, priemernú úspešnosť klasifikácie:**

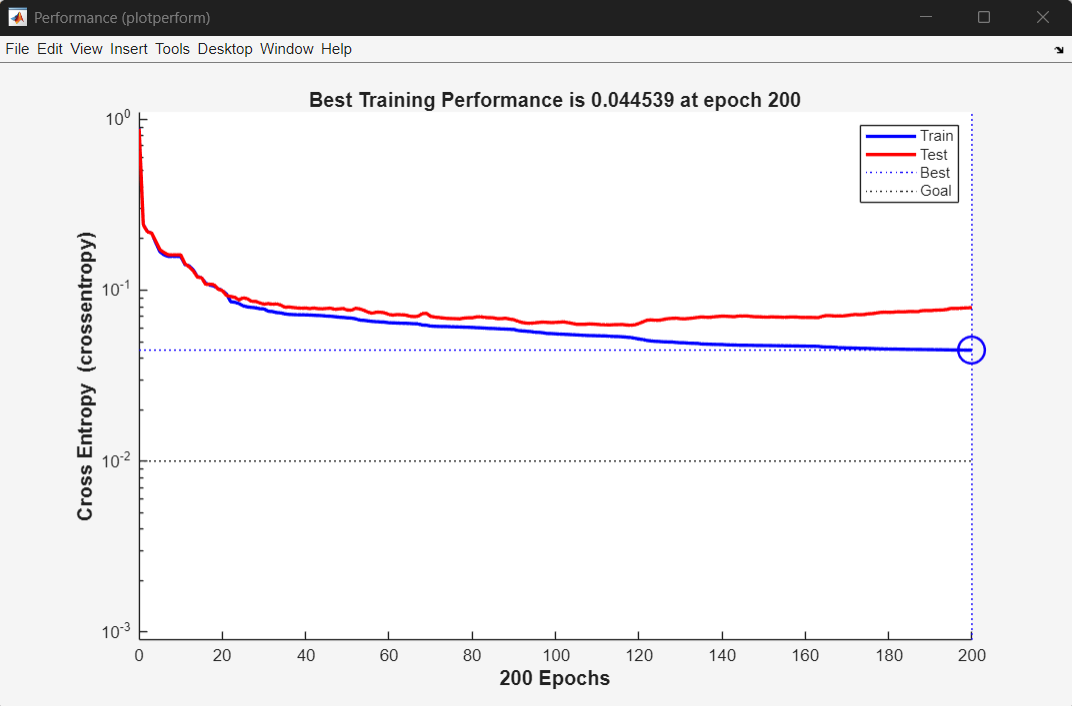
* Výsledky testovacej presnosti po 5 opakovaniach:

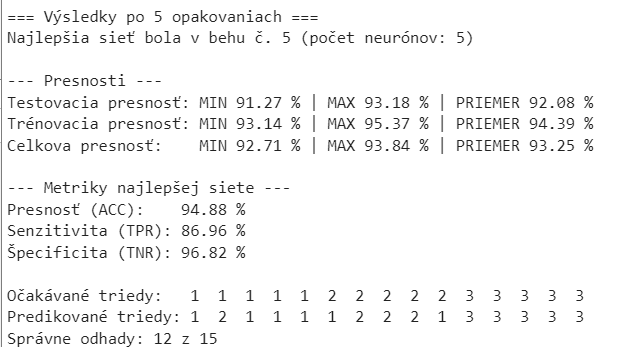


🔹 **7. Potom ukázať dve iné nastavenie parametrov NS a zdokumentovať výsledky (proces trénovania, kontingenčná matica) (stačí jedno trénovanie)**

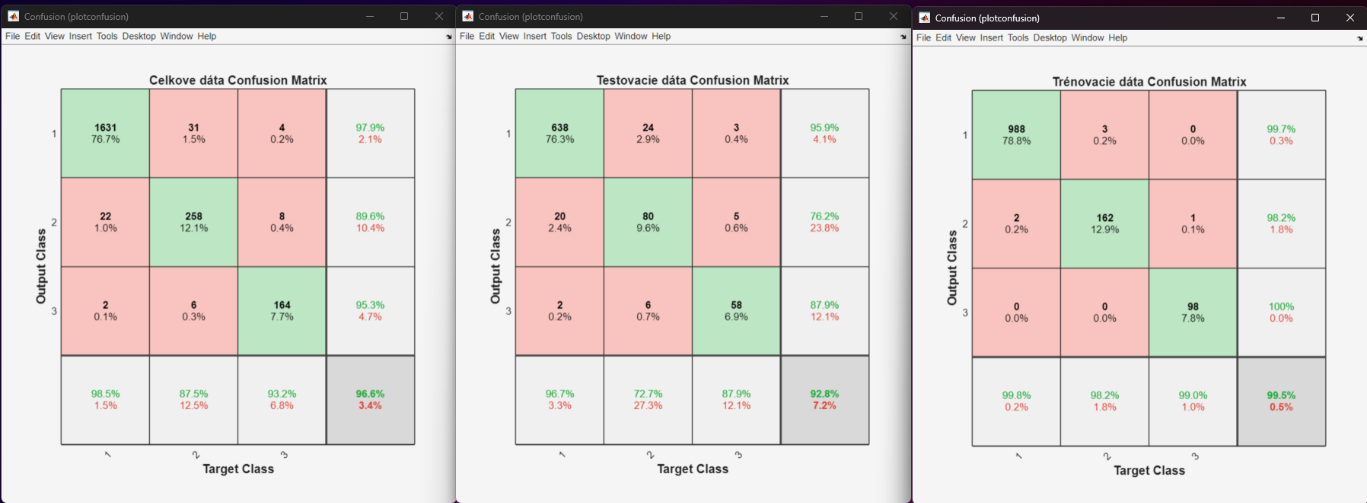
1. **5 neurónov** – výstup nebol dostatočne presný, sieť sa nenaučila dobre triediť podozrivé prípady.

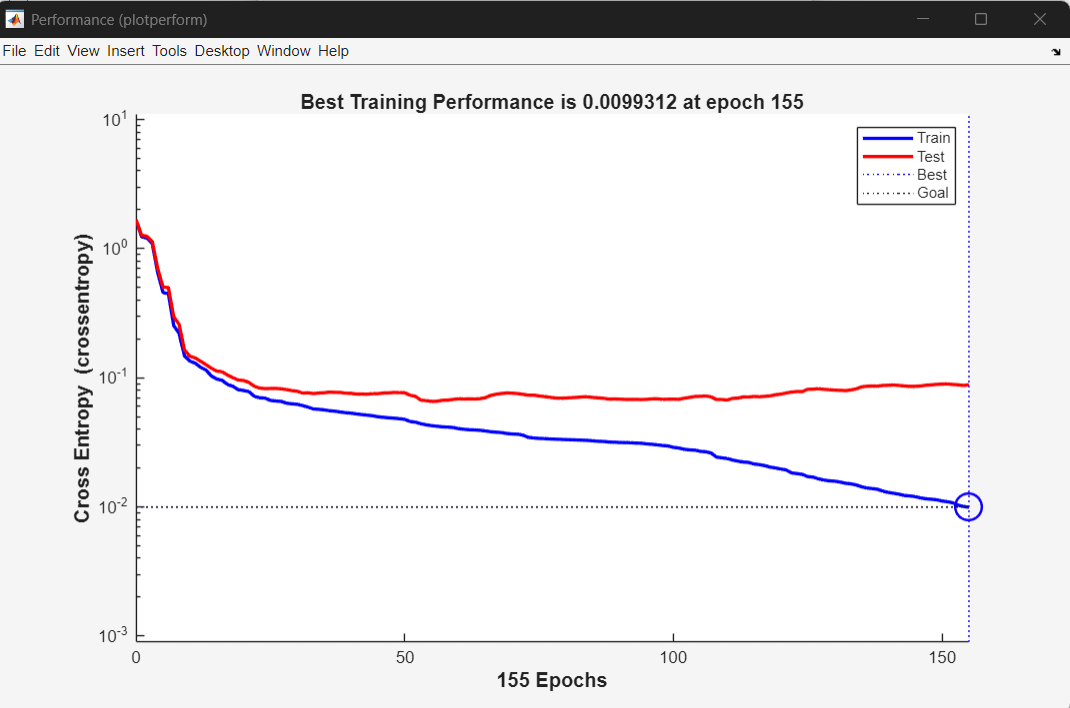
****

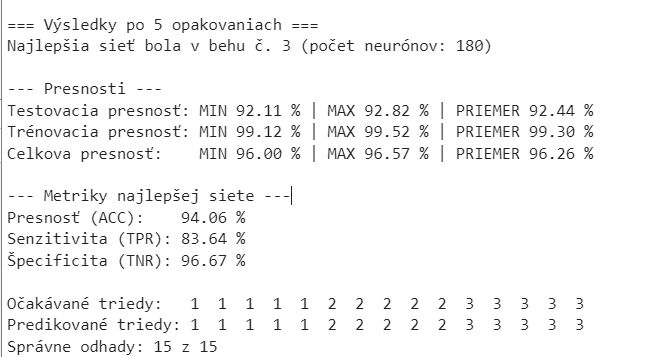




1. **180 neurónov** – veľmi vysoká presnosť, ale pomalšie trénovanie a tendencia k pretrénovaniu.

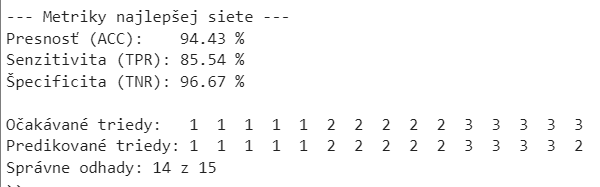




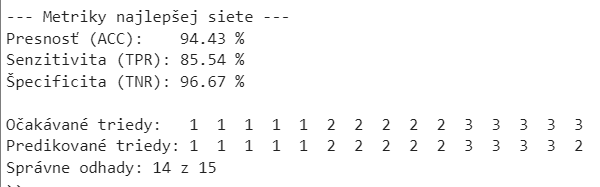


🔹 **8. Postup testovania vybraných vzoriek:**

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



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



**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 dáta:**

* **Vstupné dáta:** Obrázky číslic vo formáte 28×28 pixelov rozložené do vektora veľkosti 784 prvkov (XDataall)
* **Výstupné dáta:** Triedy číslic (0–9) zakódované pomocou one-hot reprezentácie (ind2vec)
* **Rozdelenie dát:**
  + Trénovacie dáta: 60 %
  + Testovacie dáta: 40 %
  + Validačné dáta 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 číslice 0–9), aktivačná funkcia softmax

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

* Trénovacia metóda: Levenberg-Marquardt (trainlm)
* **Počet epoch:** 300
* **Cieľová 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 trénovania. Zdokumentovať výsledky pre najlepšie trénovanie (proces trénovania, 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.

Obrázok, na ktorom je text, snímka obrazovky, vývoj, softvér

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

**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:



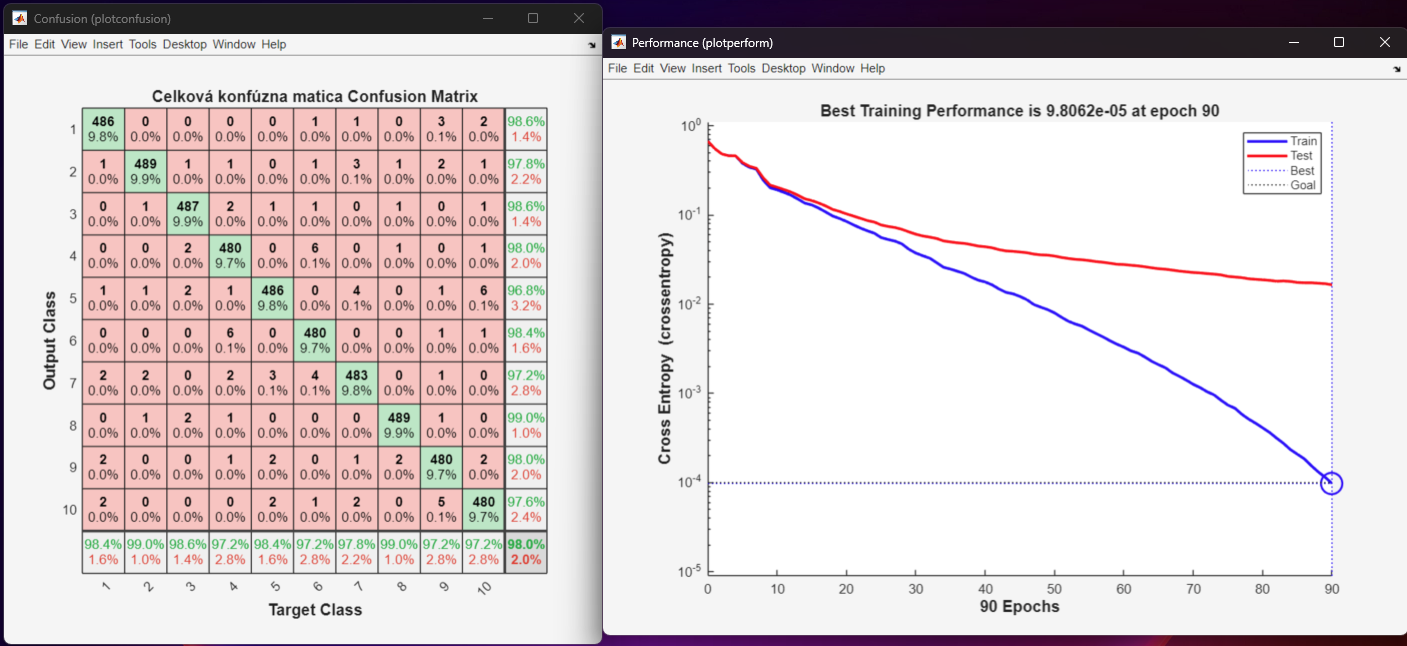
**7. Vedieť ukázať aj iné nastavenie parametrov NS:**

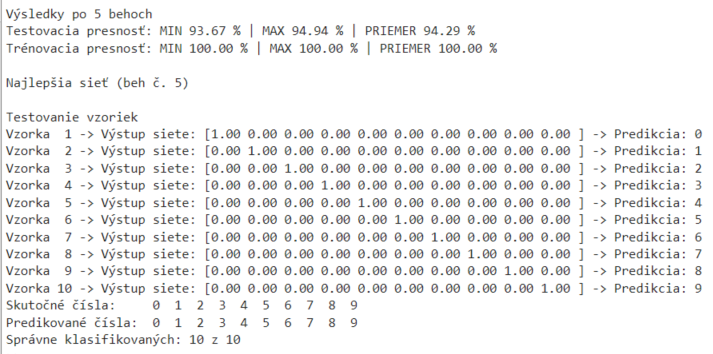
* **10 neurónov**

****

****

* **130 neurónov**

****

****

**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).

Obrázok, na ktorom je text, snímka obrazovky, písmo, dokument

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.

**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 číslic (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 dáta:** Obrázky číslic vo formáte 28×28 pixelov, vstup CNN mal tvar 28×28×1 (Ximgs)
* **Výstupné dáta:** Triedy číslic 0–9 (kategorizované pomocou categorical)
* **Rozdelenie dát:**
  + Trénovacie dáta: 60 %
  + Testovacie dáta: 40 %
  + Validačné dáta 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 5×5
  + Dávková normalizácia a ReLU
  + Druhá konvolučná vrstva: 12 filtrov veľkosti 5×5
  + Dávková normalizácia, ReLU a Max pooling (2×2, krok 2)
* **Plne prepojené vrstvy:**
  + fullyConnectedLayer(32), následne dropout(0.5)
  + fullyConnectedLayer(10)
* **Výstupné vrstvy:** softmaxLayer, classificationLayer

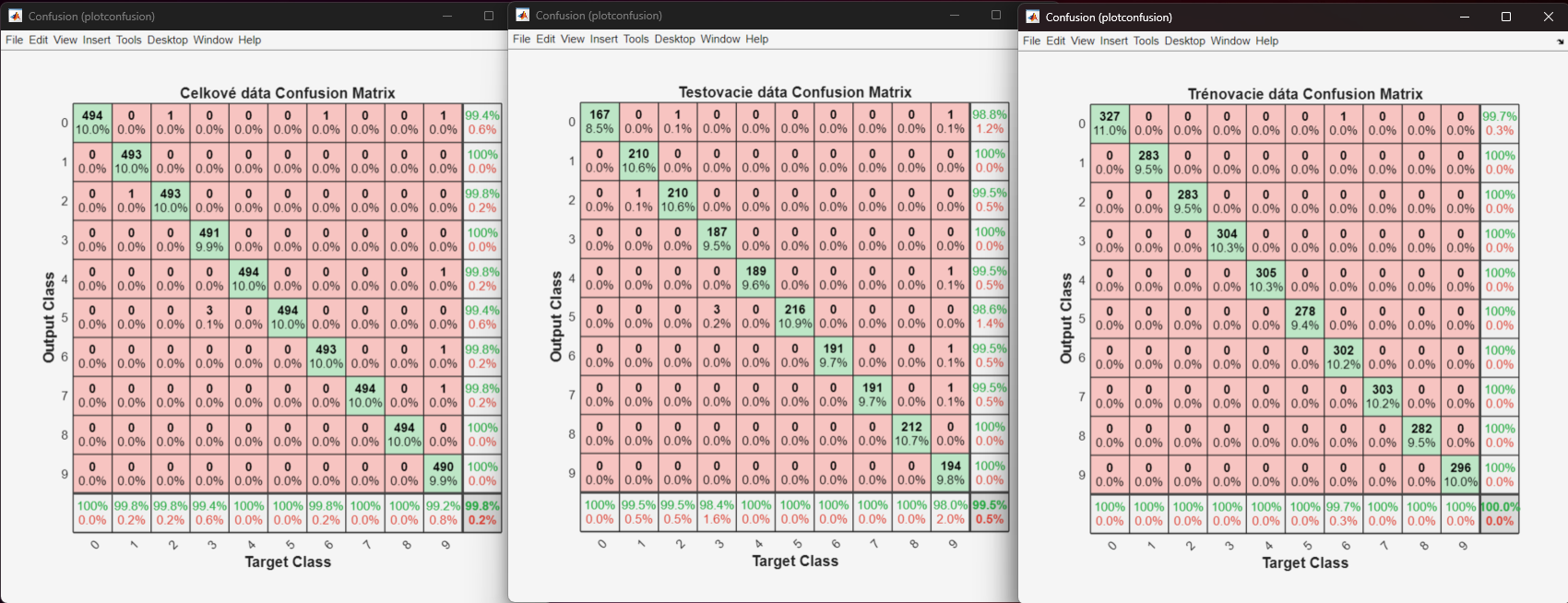
**4. Nastavené parametre trénovania (ukončovacie podmienky, použitá kriteriálna funkcia, krok učenia, veľkosť dávky, ...):**

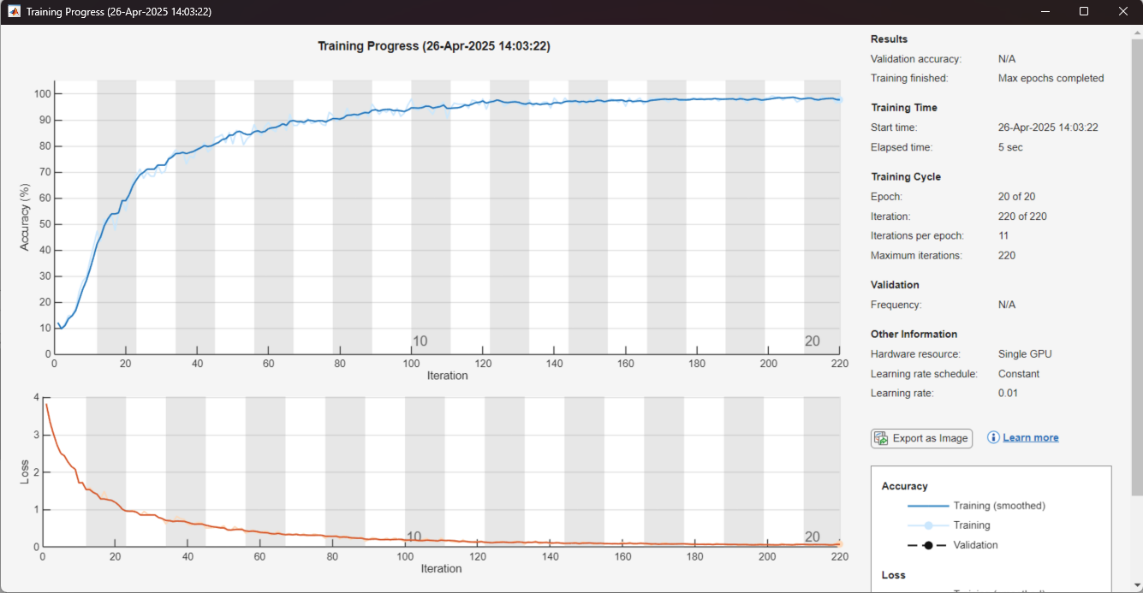
* **Optimalizačný algoritmus:** sgdm (Stochastic Gradient Descent with Momentum)
* **Počet epoch:** 20
* **Počiatočná rýchlosť učenia (InitialLearnRate):** 0.01
* **Veľkosť dávky (MiniBatchSize):** 256
* **Miešanie dát (Shuffle):** **raz pred začiatkom trénovania** (once)
* **Kriteriálna funkcia:** Kategorizovaná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

**5. Najprv zvoliť architektúru CNN (2 alebo 3 konvolučné vrstvy), čo spĺňa požiadavky**

**klasifikácie. Následne ladiť parametre trénovania. Zdokumentovať výsledky pre najlepšie trénovanie (proces trénovania, kontingenčná matica, úspešnosť klasifikácie trénovacie a testovacie data):**

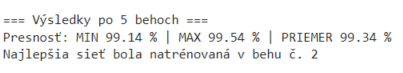
* Použité boli **2 konvolučné vrstvy**. *(viď. kód na konci strany)*
* Po 5 behoch bola vybraná najlepšia sieť na základe najvyššej testovacej presnosti.
* Výsledky boli zdokumentované pomocou plotconfusion (pre trénovacie, testovacie a celkové dáta) a priebehu učenia (training-progress).





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

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

****

**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()

];

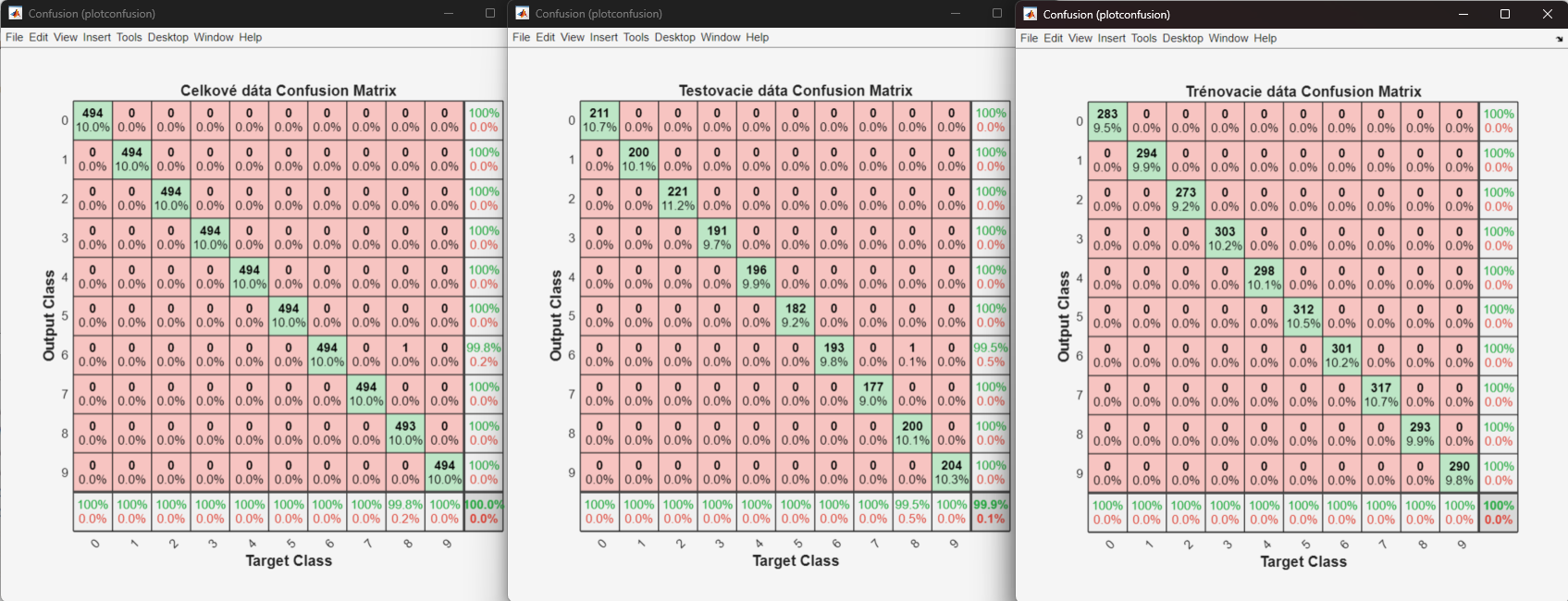
-----------

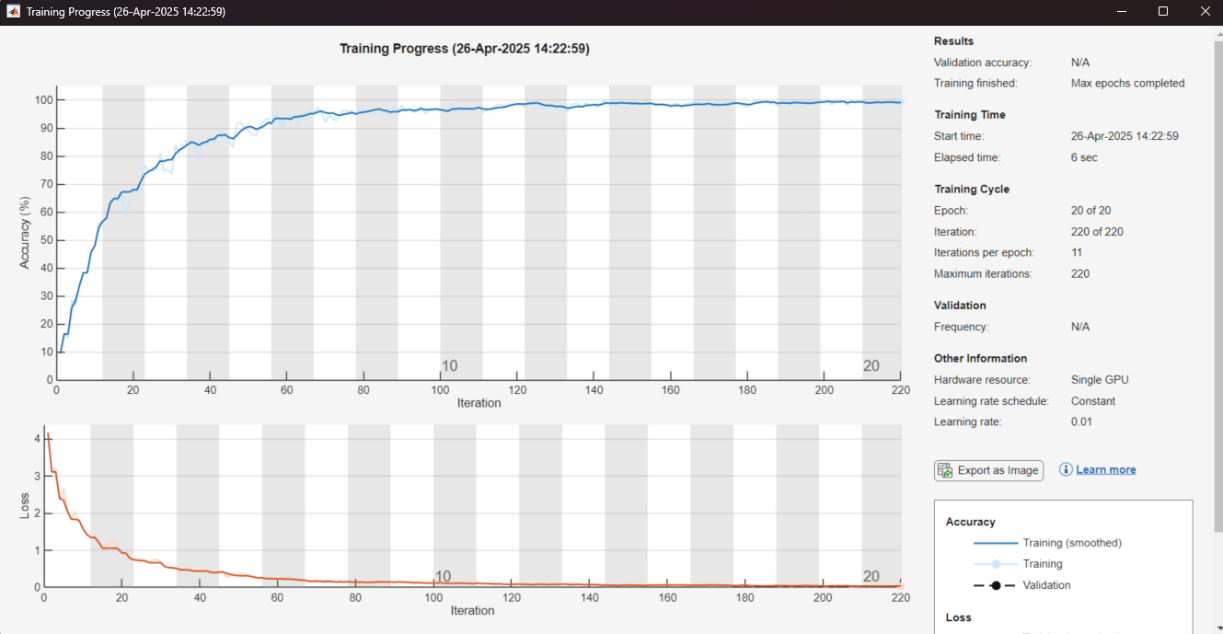
Dopad tejto zmeny:

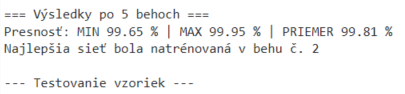
* Väčší počet filtrov a pridaná tretia konvolučná vrstva umožnia sieti zachytiť viac detailov a komplexnejších vzorov na obrázkoch.
* Väčšinou vedie k zvýšeniu presnosti na trénovacích aj testovacích dátach, hlavne pri zložitejších úlohách.
* Nevýhoda: vyššia výpočtová náročnosť (dlhší tréning), a ak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

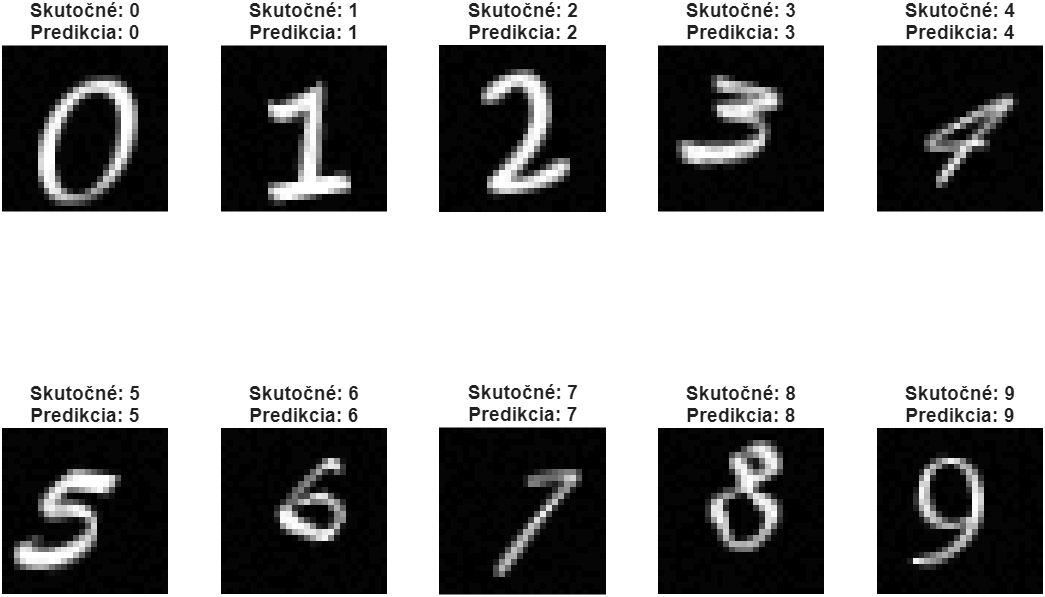
**8. Porovnať dve štruktúry CNN (proces trénovania, kontingenčná matica, úspešnosť klasifikácie trénovacie a testovacie data):**

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









**9. Porovnať MLP a CNN sieť (proces trénovania, kontingenčná matica, úspešnosť klasifikácie trénovacie 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 dáta.

**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. *(viď. kód na konci strany)*

**Obrázok, na ktorom je text, písmo, snímka obrazovky, číslo

Obsah vygenerovaný umelou inteligenciou môže byť nesprávny.**

***\*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

];