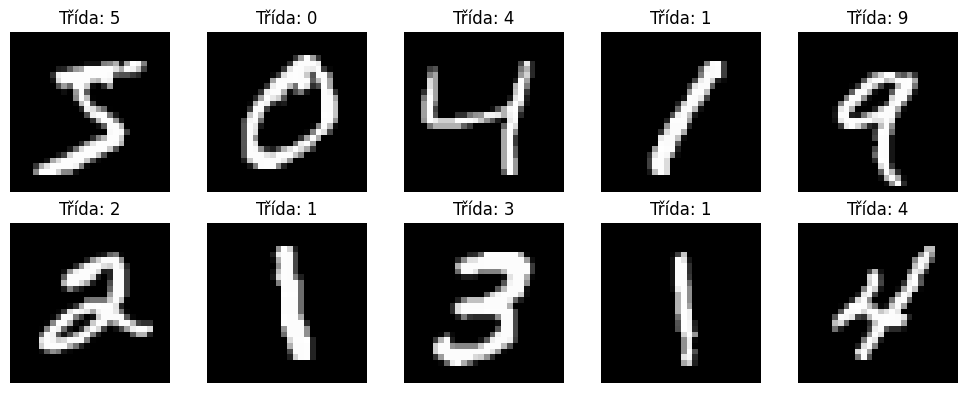
Analýza signálů II – Úloha: Neuronové sítě

## 1. Načtení datasetu

Dataset byl načten pomocí funkce mnist.load\_data() z knihovny TensorFlow.

## 2. Ukázka obrázků z datasetu

Bylo zobrazeno 10 náhodných obrázků s přiřazenou třídou (číslovkou).  
  


## 3. Počet obrázků v každé třídě

Dataset obsahuje přibližně stejné množství vzorků pro každou třídu, což je výhodné pro trénování, protože síť není zatížená nevyváženým učením.

Počet obrázků v každé třídě:

0: 6903, 1: 7877, 2: 6990, 3: 7141, 4: 6824 ,5: 6313, 6: 6876, 7: 7293, 8: 6825, 9: 6958

## 4. Velikost obrázků

Všechny obrázky byly velikosti 28x28, takže nebylo potřeba je upravovat. Normalizovali jsme hodnoty pixelů na rozsah 0–1.

## 5. Rozdělení datasetu

Dataset byl rozdělen na trénovací (60 %), validační (20 %) a testovací (20 %) sadu, aby bylo možné efektivně model natrénovat, ladit a objektivně otestovat jeho výkonnost.  
Rozdělení bylo stratifikované, aby každá sada měla stejné zastoupení tříd.

20 % pro test umožňuje robustní odhad generalizační chyby

## 6. Architektura neuronové sítě

Použita konvoluční neuronová síť se strukturou:  
- Conv2D(32) + MaxPooling  
- Conv2D(64) + MaxPooling  
- Flatten  
- Dense(128) + Dropout(0.5)  
- Dense(10, softmax)  
  
Použitá konvoluční neuronová síť je navržena tak, aby byla dostatečně výkonná pro rozpoznávání jednoduchých obrazových vzorů. MNIST je jednoduchý dataset a hlubší síť by byla zbytečná

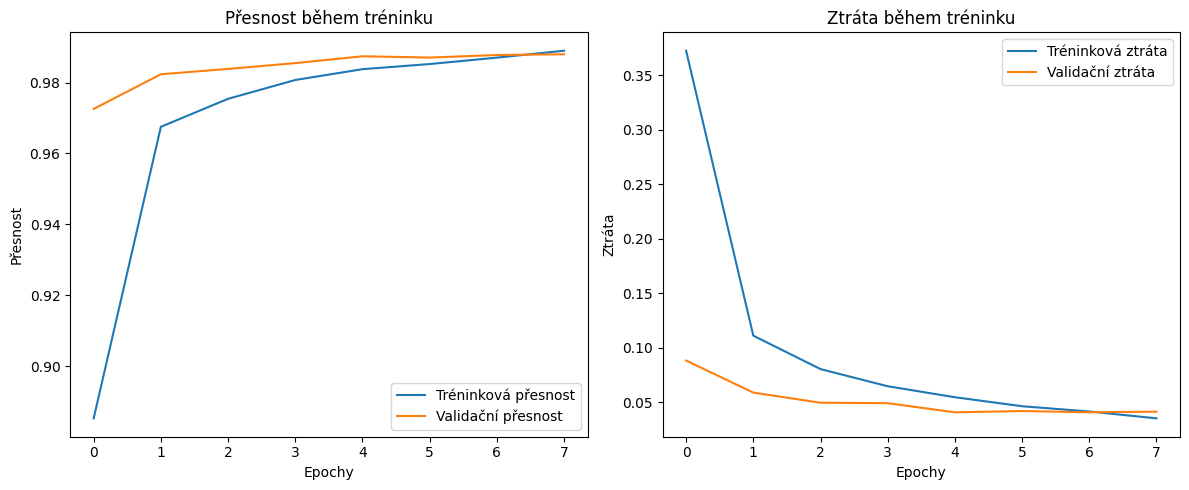
## 7. Parametry trénování

Použili jsme:  
- Adam optimizer s learning\_rate = 0.001 - je výchozí a zároveň stabilní volba, která často vede k rychlé a spolehlivé konvergenci bez nutnosti složitého ladění.  
- categorical\_crossentropy jako ztrátová funkci - one-hot zakódovanými výstupy. Porovnává pravděpodobnosti produkované modelem se skutečnými třídami a penalizuje nesprávné předpovědi  
- EarlyStopping - použili k automatickému ukončení tréninku, pokud se validační ztráta nezlepšuje několik epoch po sobě. Tím se zabrání přeučení  
- batch\_size = 128, epochs = 15 - Zvolena jako kompromis mezi výpočetní efektivitou a kvalitou aktualizace vah, Počet epoch byl zvolen s ohledem na velikost a složitost trénovacího datasetu

## 8. Průběh trénování

Trénování probíhalo velmi rychle. Síť dosáhla validační přesnosti nad 99 %.  
  
Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.



## 9. Testování sítě

Síť byla otestována na testovací sadě. Výsledky:  
Klasifikační report:

precision recall f1-score support

0 0.99 1.00 0.99 1381

1 0.99 0.99 0.99 1575

2 0.99 0.99 0.99 1398

3 0.99 0.99 0.99 1428

4 0.99 0.99 0.99 1365

5 0.99 0.99 0.99 1263

6 0.99 0.99 0.99 1375

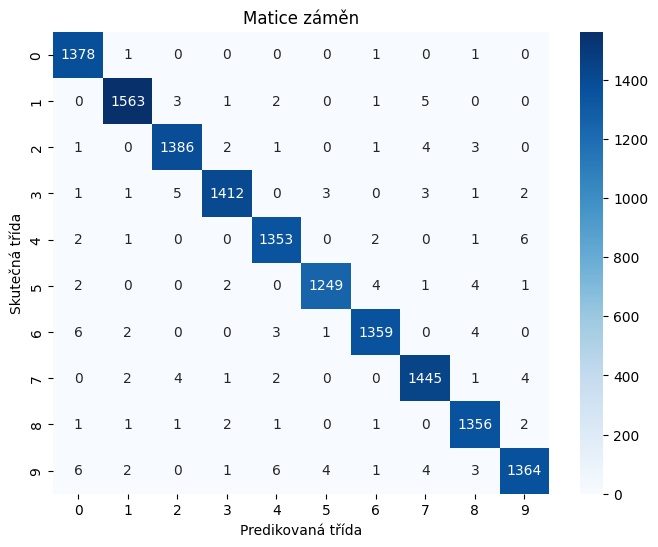
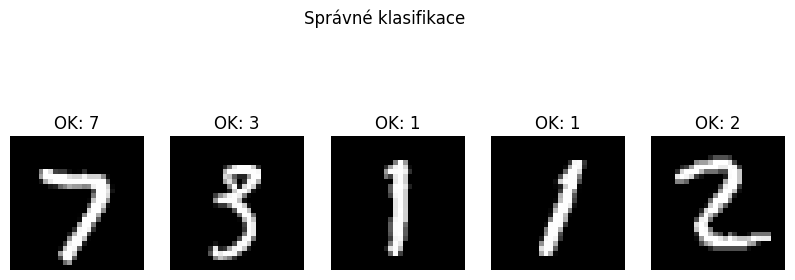
7 0.99 0.99 0.99 1459

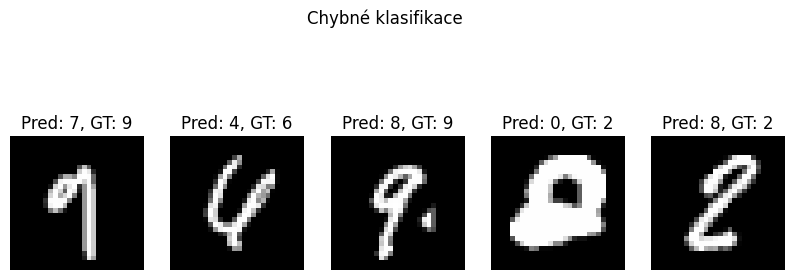
8 0.99 0.99 0.99 1365

9 0.99 0.98 0.98 1391

accuracy 0.99 14000

macro avg 0.99 0.99 0.99 14000

weighted avg 0.99 0.99 0.99 14000  




## 10. Ladění parametrů – Learning Rate

Vyzkoušeli jsme různé hodnoty learning rate:  
  
Val accuracy (combo 1 – konzervativní): 0.9811428785324097  
Val accuracy (combo 2 – rovnovážné): 0.9812856912612915  
Val accuracy (combo 3 – agresivní): 0.9559999704360962

## Shrnutí

Neuronová síť byla úspěšně navržena, natrénována a otestována. Dosáhla vysoké přesnosti a byla optimalizována s ohledem na parametry trénování.