

Experiment 01 – Perceptron a Wine dataset

Popis úlohy

Cílem experimentu bylo implementovat jednoduchý perceptron pro binární klasifikaci dat z datasetu **Wine**. Úkolem bylo rozlišit třídu 1 od ostatních tříd (spojených do třídy 0). Dataset obsahuje 178 vzorků s 13 příznaky.

Parametry trénování

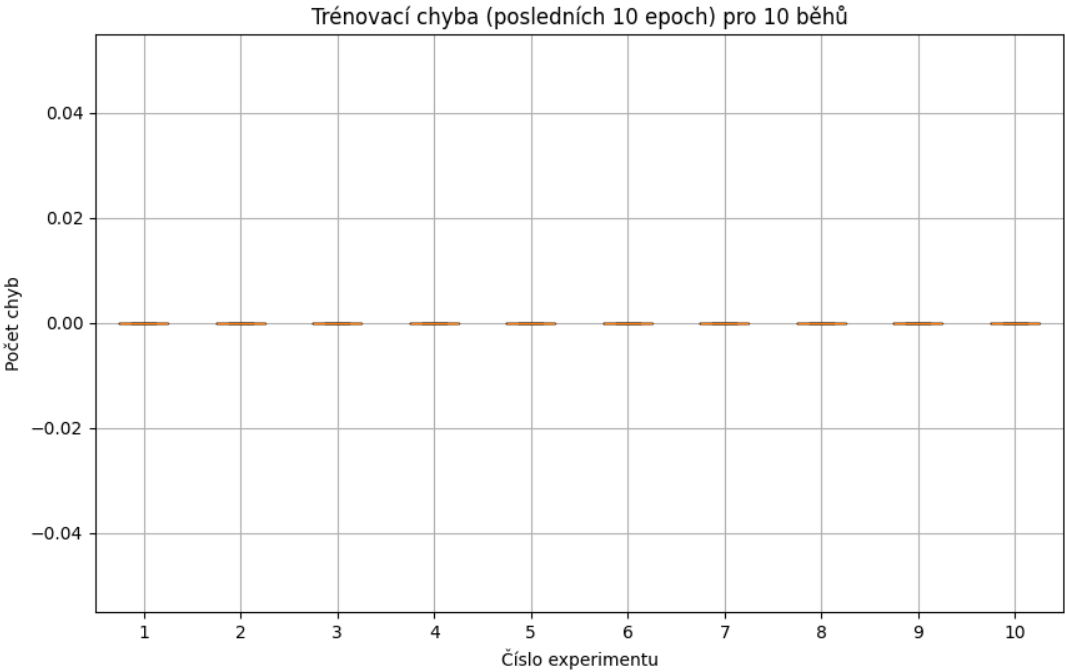
- **Dataset:** [Wine \(UCI\)](#)
- **Počet trénování:** 10
- **Počet epoch:** 100
- **Learning rate:** 0.1
- **Aktivační funkce:** Kroková funkce step function (0/1)
- **Rozdělení dat:** 80 % trénink, 20 % test (142 vzorků trénink, 36 vzorků test)
- **Normalizace vstupů:** StandardScaler()

Výsledky experimentu

- **Nejlepší model:** Běh č. 0
- **Přesnost na testovací množině:** 100.0 %
- **Váhy nejlepšího modelu:**

[-0.80845798 -0.62797949 -0.34409233 -0.95182429 0.6040847 0.15578245
-0.16055299 0.36387226 0.51721521 -0.01992135 -1.23524501 0.50447961
0.30845586 -1.54322013]

Boxplot trénovacích chyb (posledních 10 epoch)



Experiment 02 – Trénování neuronové sítě na funkci $\sin(x)$

Popis úlohy

Cílem experimentu bylo natrénovat jednoduchou plně propojenou neuronovou síť pro aproximaci matematické funkce $\sin(x)$. Síť měla jednu skrytou vrstvu s proměnlivým počtem neuronů a lineární výstupní vrstvu.

Parametry trénování

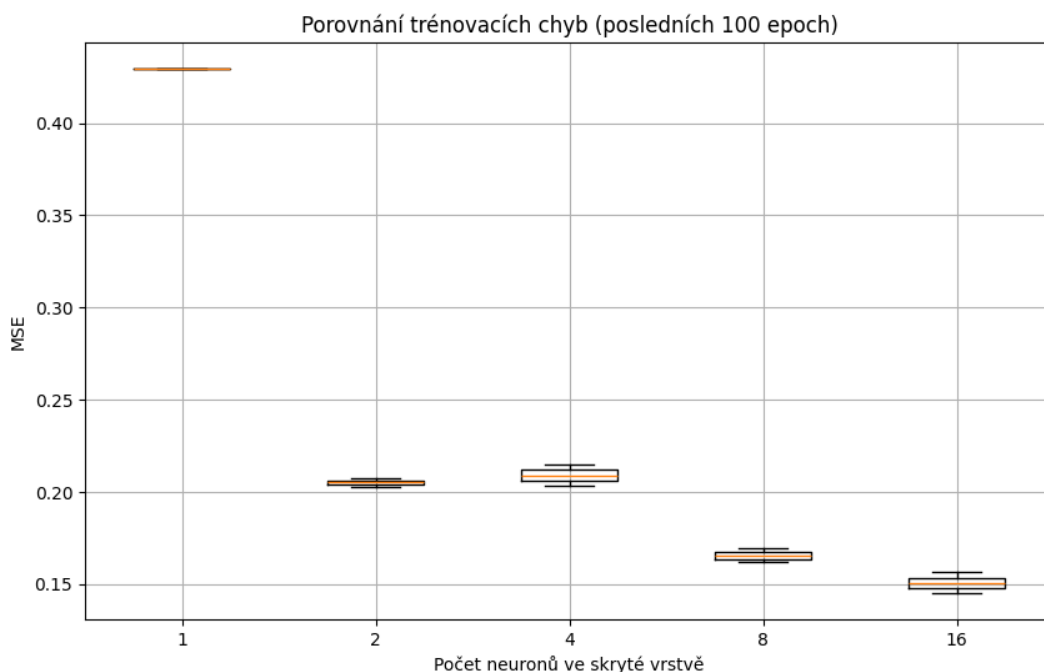
- **Vstupní vrstva:** 1 neuron
- **Výstupní vrstva:** 1 neuron
- **Aktivační funkce:** tanh (skrytá vrstva)
- **Počet epoch:** 1000
- **Learning rate:** 0.01
- **Ztrátová funkce:** Mean Squared Error (MSE)
- **Trénovací data:** 200 bodů rovnoměrně rozložených na intervalu $[-2\pi, 2\pi]$

Experiment: Počet neuronů ve skryté vrstvě

Provedl jsem experiment s následujícími velikostmi skryté vrstvy: [1, 2, 4, 8, 16] Pro každý model jsem zaznamenal průběh trénovací chyby a finální MSE.

Výsledky

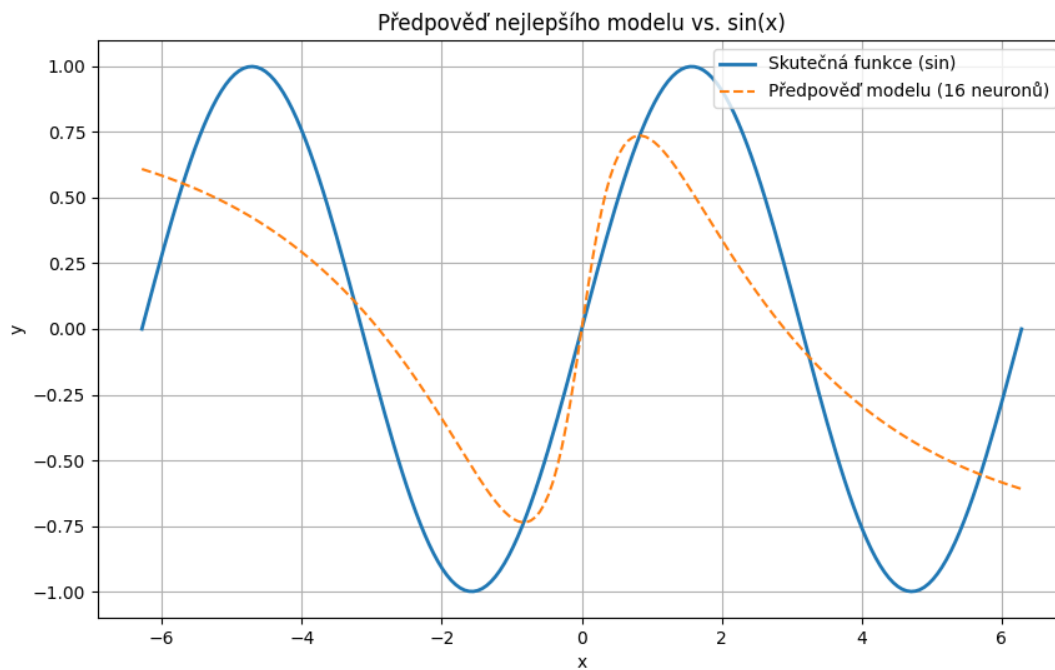
Trénovací chyba (MSE) – posledních 100 epoch



Nejlepší model

- Počet neuronů ve skryté vrstvě: 16
- Finální trénovací chyba (MSE): 0.1453

Porovnání výstupu nejlepšího modelu s funkcí $\sin(x)$



Experiment 03 – Tvorba FFNN pomocí Keras (sekvenční model)

Zadání

- Seznámení se s tvorbou dopředné neuronové sítě (Feedforward Neural Network – FFNN) pomocí sekvenčního modelu v Keras.
- Návrh více topologií a jejich vyhodnocení na klasifikační úloze.
- Experimentování se stabilitou trénování pomocí více běhů.
- Vyhodnocení nejlepšího modelu.

Úloha

- Klasifikační úloha rozpoznávání květin pomocí datasetu Iris.
- Klasifikace vzorku do jedné ze tří tříd na základě čtyř numerických atributů.

Příprava dat

- Dataset Iris byl normalizován pomocí StandardScaler.
- Cílové hodnoty byly zakódovány pomocí LabelBinarizer (one-hot encoding).
- Data byla rozdělena na:
 - 70 % trénovací sada
 - 15 % validační sada
 - 15 % testovací sada

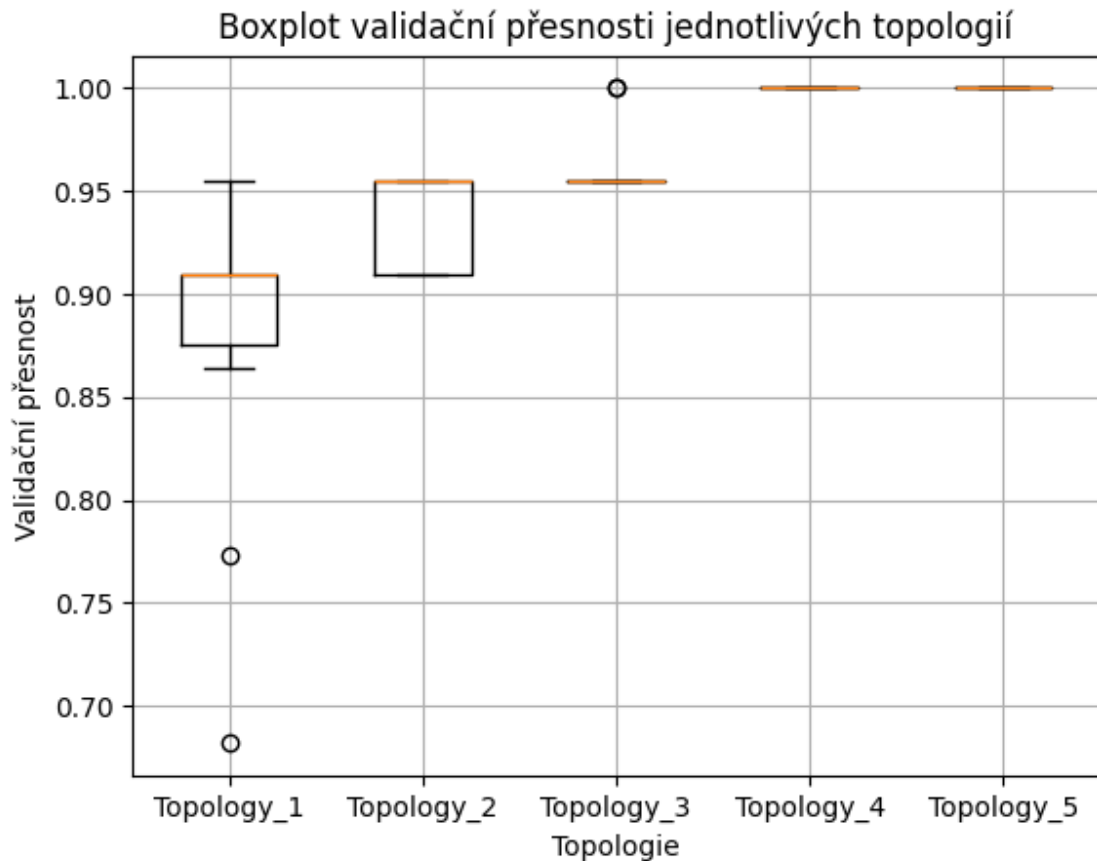
Topologie modelů

Topologie	Popis
Topology_1	10 neuronů → výstupní vrstva
Topology_2	20 neuronů → 10 neuronů → výstupní vrstva
Topology_3	40 neuronů → 20 neuronů → 10 neuronů → výstupní vrstva
Topology_4	80 neuronů → 40 neuronů → 20 neuronů → 10 neuronů → výstupní vrstva
Topology_5	160 neuronů → 80 neuronů → 40 neuronů → 20 neuronů → 10 neuronů → výstupní vrstva

- Každá architektura byla trénována 10krát pro stabilizaci vlivu náhodné inicializace vah.

Výsledky

- Pro každou topologii byla zaznamenána validační přesnost.
- Výsledky jsou zobrazeny pomocí boxplotu:



Nejlepší model

- Nejlepší výsledky dosáhla topologie **Topology_4**.
- Nejlepší model byl natrénován na trénovacích datech a ověřen na testovací sadě.

Výsledky na testovacích datech:

- **Test Loss:** 0.0411
- **Test Accuracy:** 1.0000

Závěr

- Složitější topologie s více vrstvami dosahují lepších výsledků na této klasifikační úloze.
- Testovací přesnost 1.0000 ukazuje velmi dobré naučení modelu.
- Pro menší dataset Iris byla dostačující i jednoduchá FFNN architektura.
- Pro trénink byl použit CPU, což bylo pro rozsah úlohy dostatečné.

Experiment 04 – Trénink FFNN na GTSRB datasetu (PyTorch)

Zadání

- Seznámení s tvorbou dopředné neuronové sítě (Feedforward Neural Network – FFNN) v PyTorch.
- Návrh, trénování a vyhodnocení 5 různých architektur na klasifikační útlize (GTSRB).
- Trénování každé architektury 5× pro vyrovnaní vlivu náhodnosti.
- Vyhodnocení přesnosti a ztráty pomocí boxplotů.
- Vyhodnocení nejlepší topologie na testovacích datech.

Parametry experimentu

- Dataset: GTSRB (German Traffic Sign Recognition Benchmark)
- Framework: PyTorch 2.7.0
- Batch size: 64
- Epochs: 2
- Počet běhů každé topologie: 5
- Learning rate: 0.0005

Použité architektury (topologie)

1. [128]
2. [256, 128]
3. [512, 256, 128]
4. [512, 256, 128, 64]
5. [1024, 512, 256, 128, 64]

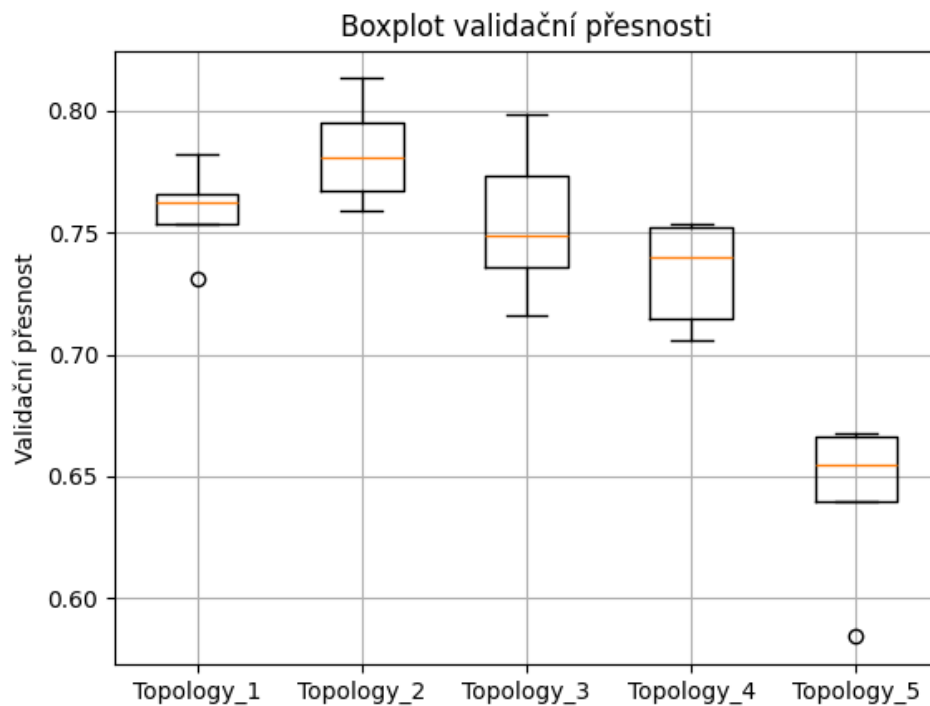
Každá síť končí plně propojenou vrstvou na 43 výstupních tříd.

Výsledky trénování

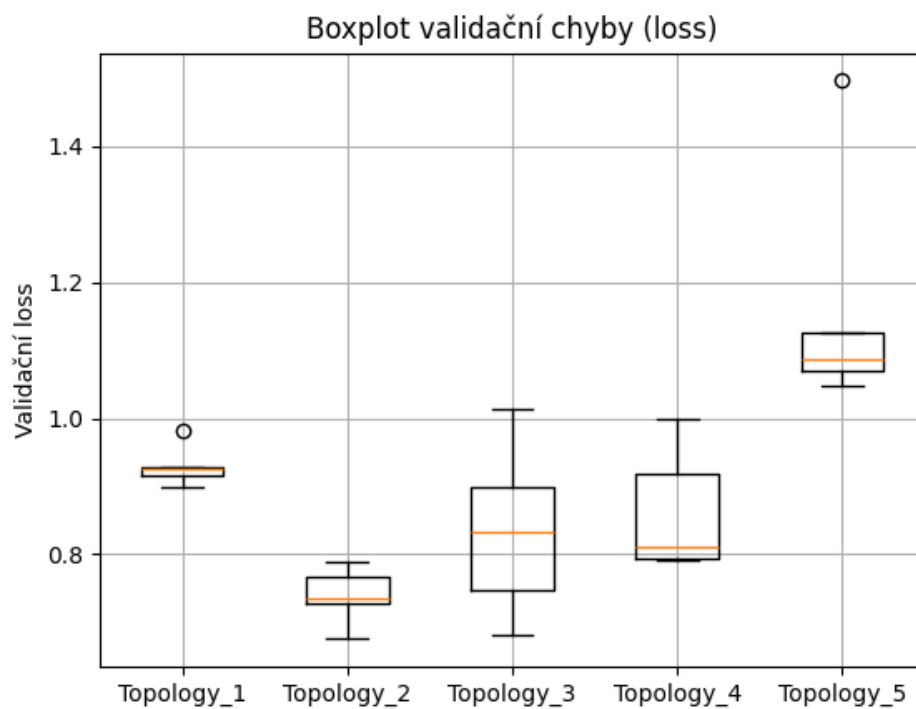
Boxploty

Byly vytvořeny boxploty validační přesnosti a validační chyby (val_loss) přes 5 běhů pro každou architekturu.

Boxplot validační přesnosti:



Boxplot validační chyby (loss):



Nejlepší model

- Nejlepší topologie na základě validační přesnosti: **Topology_2**
- Průměrná přesnost: 78.31%
- Průměrná ztráta: 0.7391

Výsledky na testovacích datech

Metrika	Výsledek
Accuracy	69%

- Nejmenší síť se ukázala jako nejstabilnější (malý rozptyl v boxplotu) a dosáhla nejlepšího výkonu.
- Zvětšování velikosti sítě nepřineslo zlepšení výsledků, což může být způsobeno omezenou velikostí trénovacích dat nebo přeučením větších modelů.

Shrnutí

Experiment ukázal, že pro dataset GTSRB a jednoduché FFNN síť:

- menší architektura je dostatečná,
- náhodnost inicializace vah má vliv – proto bylo vhodné provést 5 běhů,
- správné nastavení batch_size a počtu workers má významný vliv na rychlost trénování.

Experiment 06 – Klasifikace dopravních značek pomocí CNN (PyTorch)

Cíl

- Navrhnout a otestovat různé architektury konvolučních neuronových sítí (CNN) pro klasifikaci dopravních značek z datasetu GTSRB.
- Porovnat jejich výkon a vybrat nejefektivnější model.

Použité nástroje a knihovny

- PyTorch 2.4 (nebo novější s CUDA podporou)
- torchvision, sklearn, matplotlib, seaborn
- Dataset: GTSRB (German Traffic Sign Recognition Benchmark)

Parametry experimentu

- Počet CNN architektur: 5
- Počet tréninků pro každou architekturu: 10 (pro vyrovnání náhodnosti)
- Epochs: 5
- Batch size: 64
- Rozměr vstupu: 32x32 RGB
- Optimalizátor: Adam (LR = 0.0008)
- Datum a čas: 18:07:49 Центральная Европа (лето) on Wednesday, May 28, 2025

Architektury CNN

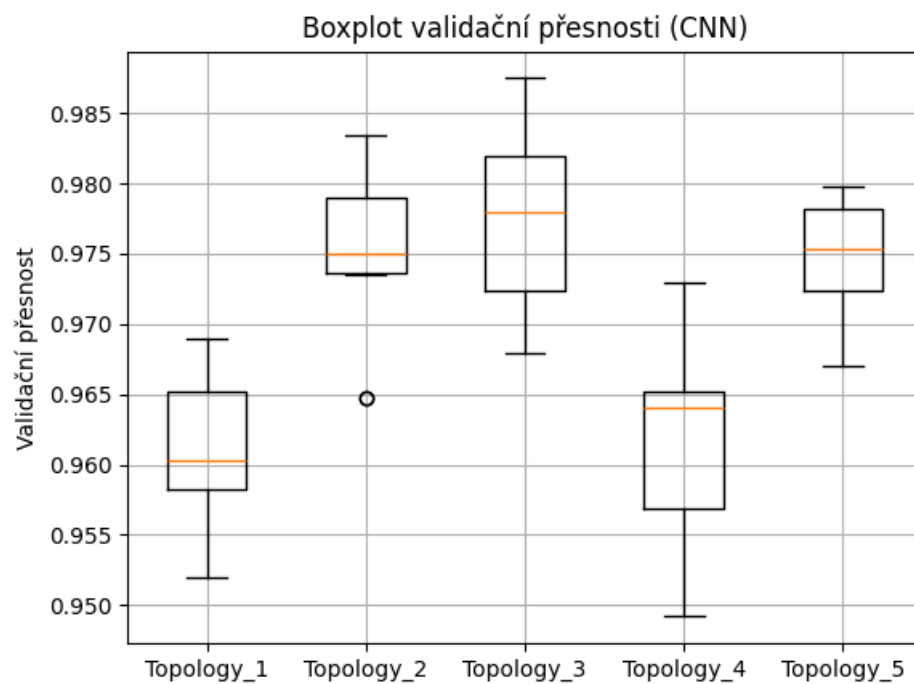
Každá topologie byla definována jako seznam konvolučních vrstev (filtry a velikost kernelu):

1. [(16, 3)]
2. [(32, 5), (16, 3)]
3. [(64, 3), (32, 5)]
4. [(32, 3), (64, 5), (32, 3)]
5. [(64, 5), (64, 3), (32, 5)]

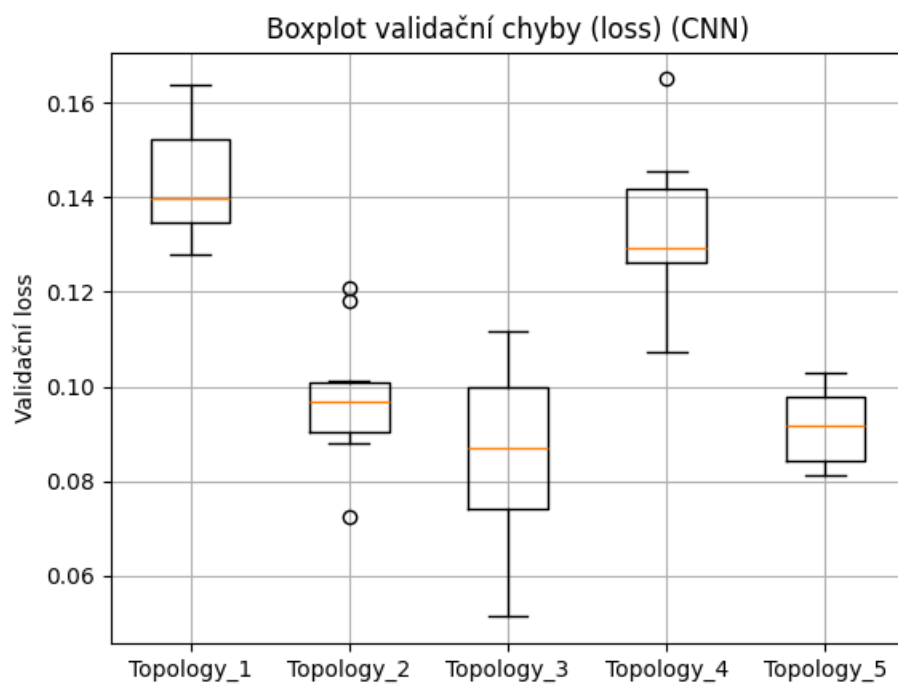
Boxploty

Výsledky trénování 10 běhů pro každou topologii byly zaznamenány a zobrazeny v boxplot grafech:

- **Validační přesnost:**



- **Validační ztráta:**



Nejlepší model

- **Topologie:** [(64, 3), (32, 5)]
- Průměrná přesnost: 97.74%
- Průměrná ztráta: 0.0858 Model byl natrénován znovu a vyhodnocen na testovacích datech.

Výsledky na testovacích datech

Metrika	Výsledek
Accuracy	87%

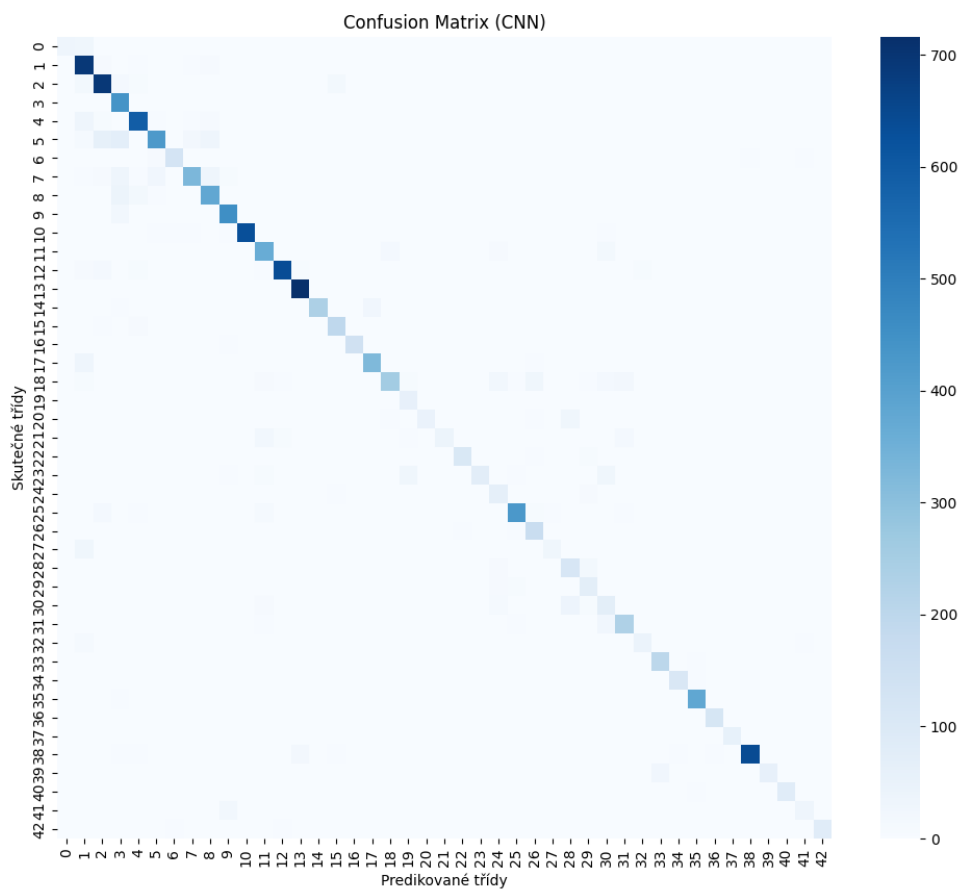
Klasifikační report

- macro avg: precision 0.86, recall 0.82, f1-score 0.83
- weighted avg: precision 0.88, recall 0.87, f1-score 0.87

Pozorování

- Model dosahuje vysoké přesnosti, zejména u častých tříd.
- Zlepšení oproti předchozím experimentům díky většímu počtu epoch a optimalizované topologii.
- Některé méně zastoupené třídy (např. třída 1 nebo 35) vykazují nižší přesnost.

Confusion Matrix



Shrnutí

- CNN sítě nadále převyšují jednoduché FFNN v klasifikaci obrazových dat.
- Větší počet epoch a upravené architektury vedly k lepším výsledkům.
- Pro GTSRB se jako nejlepší ukázala středně hluboká síť [(64, 3), (32, 5)].
- Experiment proběhl úspěšně a všechna data byla správně uložena.