

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky

Zápočtová práce 1  
Lukáš Pírko

## Cíl práce

Úkolem bylo navrhnout vhodnou dopřednou vícevrstvou neuronovou síť, která bude aproximovat výpočet následujícího postupu

1. Vstupem budou 3 čísla  $a, b, c$  o libovolných reálných hodnotách v rozsahu  $[-1, 1]$
2. Výstupem bude dvojice hodnot  $x, y$  pro kterou platí:
  - a.  $x = \max(a, b, c) \times b$
  - b.  $y = a^2 - b \times c$

Výsledná neuronová síť má tři vstupní parametry  $a, b, c$  a dvě výstupní  $x, y$  které by měli odpovídat zadanému výpočtu výše.

## Struktura projektu

Projekt je strukturován do dvou zdrojových souborů

main1.py – soubor který slouží k tvorbě neuronové sítě. Strukturován je následovně. V úvodu souboru jsou importy potřebné ke spuštění, následuje tvorba trénovací množiny, tvorba modelu, vykreslení grafu, a nakonec samotné uložení modelu.

main2.py – zde dochází k načtení modelu neuronové sítě, který bych vytvořený v předešlém kroku. Výsledky sítě jsou vypsány do konzole a vypsány do grafu, pro lepší názornost.

## Řešení

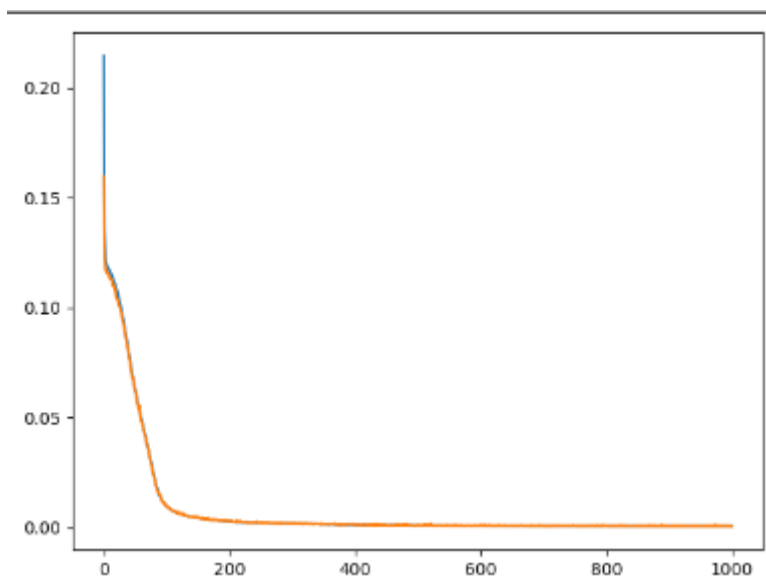
Neuronová síť, která vzešla z testování má jednu skrytou vrstvu s 36 neurony. Na základě diskuze s vyučujícím a jeho doporučením byla jako aktivační funkce skryté vrstvy použita hyperbolický tangens a jako výstupní aktivační funkce lineární. Řešený problém se zdál být relativně jednoduchý, z tohoto důvodu byla využita tato topologie neuronové sítě. Předpoklad že není potřeba mít více vrstev vyplynulo i z testování. A tak můžeme úspěšně potvrdit, že tato topologie byla vybrána jako vhodná. K tomuto faktu vedl i pokus o přidání více vrstev, či více neuronů. Nicméně ani po těchto změnách nedocházelo k zásadnímu zlepšení výsledků.

Síť jsem testoval na dvou tisících datech. Tyto data byla generována na základě seedu. K tomuto řešení jsem se uchýlil, z důvodu možnosti znovuspuštění a porovnání výsledných hodnot, na základě stejných vstupních dat. Tímto způsobem byla neuronová síť optimalizována až do uspokojivého výsledku. Počet testovacích dat by měl bohatě stačit vzhledem k faktu, že mnohonásobně převyšuje počet optimalizovaných vah a prahů. Mé zkoušení a testování různých kombinací a pokusů se ustálilo na následujícím nastavením.

- Počet vstupních dat: 2000
- Počet neuronů: 36
- Velikost dávky: 32
- Data používá na validaci: 15%
- EPOCHa: 1000

## Výsledek

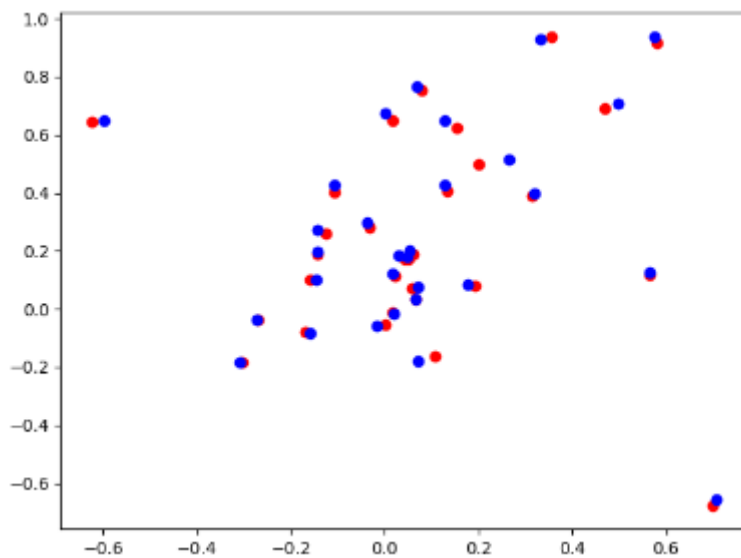
Obrázek 1 popisuje průběh trénování samotné neuronové sítě. Konkrétně sledujeme změnu hodnot chyby neuronové sítě. Kdy modrá křivka popisuje trénovací množinu a oranžová popisuje validační množinu. Už z grafu je patrné že se tyto dvě hodnoty téměř shodují. Zároveň klesající křivku lze označit za očekávané a správné chování.



Obrázek 1 - Graf natrénování

Obrázek 2 popisuje přesnost výsledné neuronové sítě, kde je zobrazeno zhruba 30 pseudo náhodně vygenerovaných výsledků. Výsledné hodnoty jsou velmi zajímavé, ovšem mnohem více zajímavé je pro nás rozdíl těchto hodnot. Odchytky se při posledním testu pohybovali v intervalu (0.006 – 0.017).

Zároveň je zde patrné, že v určitých oblastech má síť větší přesnost než v jiných. Toto je způsobeno nepokrytí celého rozsahu řešené úlohy. Pro zlepšení by bylo zapotřebí mít testovací data, která budou zastupovat celý rozsah řešené úlohy. Pak naroste i přesnost výsledné neuronové sítě.



Obrázek 2 - Testování výsledků (síť vs. cíl)

## Závěr

Neuronová síť by měla dosahovat dostatečné přesnosti. Odchytky se vždy projevuje až na druhém desetinném místě, což v aktuální situaci lze považovat za dostatečné. Pro zmenšení odchytky je potřeba mít trénovací data která lépe zastoupí celý rozsah řešené úlohy.