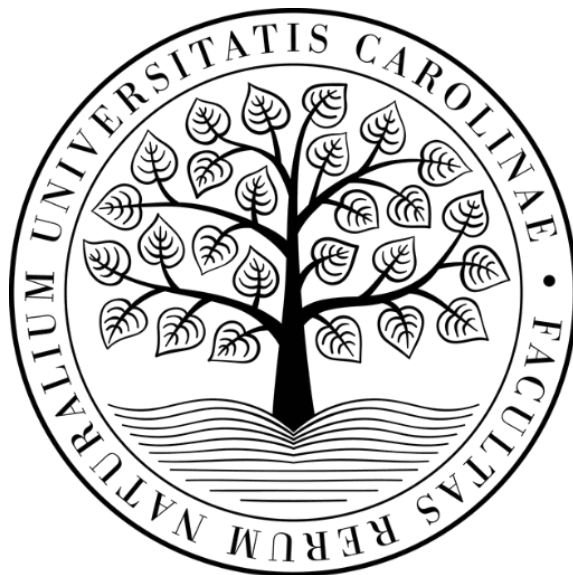


UNIVERZITA KARLOVA  
PŘÍRODOVĚDECKÁ FAKULTA



Úvod do programování  
Neuronové sítě

Řádová, Martina

1. ročník doktorského studia

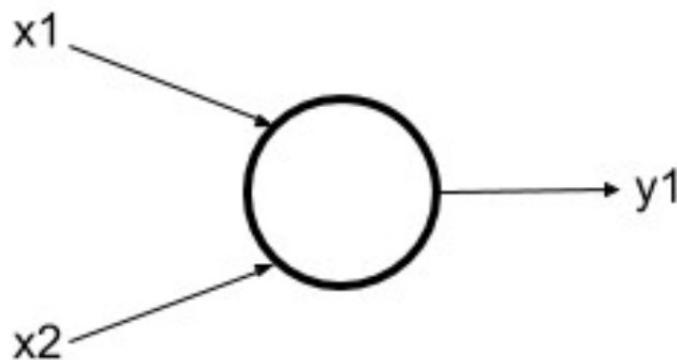
Praha, 2022

## 1. Zadání

Cílem úkolu je sestavit a zejména pochopit fungování základní neuronové sítě. Síť je testována na datasetu MNIST (Modified National Institute of Standards and Technology), který obsahuje databázi ručně psaných digitálních číslic. V prvním kódu *basic\_NN* jsou použity jen základní číslice od 1 do 9 načtené pomocí funkce *load\_digits*. V druhém kódu *basic\_NN\_02* jsou načtena data pomocí funkce *fetch\_openml*, které mají lepší rozlišení než předešlá data.

## 2. Popis a rozbor problému + vzorce

Neuronové sítě (NN) jsou populárními modely *strojového učení* (ML). Mimo jiné se používají i ke zpracování obrazových dat. NN sestávají z neuronů (uzlů). Ty jsou schopné použít vstupní data, provést výpočty a vytvořit výstup (viz obrázek 1). Počet vstupů i výstupů může být jakýkoliv.



Obrázek 1: Model neuronu

Uvnitř neuronu je vstup vložen do následující rovnice:

$$w_1x_1 + w_2x_2 + b$$

kde  $w_1$  a  $w_2$  jsou váhy,  $x_1$  a  $x_2$  jsou vstupy a  $b$  je bias. Takto získaná hodnota je vložena do aktivační funkce, která nabývá hodnot od 0 do 1.

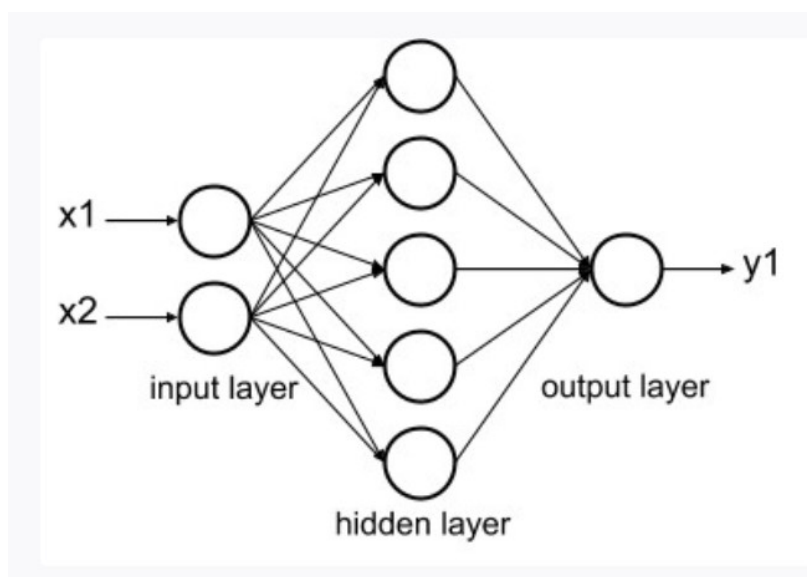
$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Aktivačních funkcí je více, výše představená *sigmoid*, *tanh* a *ReLU*. Tanh je podobná sigmoid, ale její rozmezí je od -1 do 1. Vzorec je:

$$f(x) = \tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

ReLU je Rectified Linear Unit a je to funkce pro pozitivní čísla, ta jsou rovna  $x$ , negativní čísla jsou rovna 0. Kterou z těchto funkcí použít záleží na konkrétních používaných datech.

Neuronové sítě mají zpravidla vstupní, skrytou a výstupní vrstvu. Počet skrytých vrstev může být od jedné do stovek podle složitosti sítě. Na začátku lze i stanovit kolik neuronů (uzlů) bude skrytá vrstva mít. Na obrázku 2 je znázorněna neuronová síť dopředná, to znamená, že ze vstupní data jdou do skryté vrstvy (z každého neuronu vrstvy vstupní do každého neuronu vrstvy skryté) a následně do výstupní vrstvy. Žádné váhy nebo cokoliv jiného se nevrací.



Obrázek 2: Model sítě

Jelikož NN mají množství parametrů, které lze upravovat, není snadné je udržovat stále aktualizované a dosáhnout tak co nejlepšího výkonu. K tomu slouží *backpropagation* (zpětné šíření), které umožňuje se z výstupu iteracemi vracet zpět do skrytých vrstev a aktualizovat koeficienty.

Důležitým ukazatelem přesnosti sítě je *ztrátová funkce* (loss function). Jedním z příkladů takovéto funkce je například *cross entropy*.

### 3. Popisy algoritmů formálním jazykem

Oba algoritmy mají za úkol načíst data z datasetu MNIST. V jednom případě se ptají uživatele na vstupy jako jsou počet iterací a solver. Algoritmus následně rozdělí data na trénovací a testovací a nad trénovacími provádí výpočty a predikce. Nakonec je vypsána dosažená přesnost v určení digitálních ručně psaných čísel.

#### 4. Problematické situace a jejich rozbor + ošetření těchto situací v kódu

Úkolem bylo proniknout do základů neuronových sítí a jak takovouto síť sestavit. Díky online kurzu na webové stránce Sololearn – ML – NN, bylo možné prakticky a uceleně nahlédnout do základních principů. Jistým problémem bylo správné pořadí všech úkonů a funkcí, které byla snaha do tvořeného kódu zařadit. Řešením bylo vše s určitými obměnami vyzkoušet, zopakovat si principy jednotlivých částí a dosáhnout správného výsledku.

#### 5. Vstupní data, formát vstupních dat, popis

Vstupními daty jsou výše zmiňované digitální ručně psané číslice ze známého datasetu MNIST. První verze čísel je načítána funkcí *load\_digits*. Jedná se o snímky digitálních číslic o rozměrech 8 x 8 pixelů. Jedná se o velmi jednoduché snímky, které jsou vhodné pro použití a testování základních sítí. Další číslice datasetu MNIST byly načteny pomocí funkce *fetch\_openml*. Jsou to o něco přesnější snímky o rozměrech 28 x 28 pixelů.

#### 6. Výstupní data, formát výstupních dat, popis

Výstupní data z těchto algoritmů nejsou. Výstupem je pouze několik řádků v Terminálu, kde jsou vypsané celkové přesnost sítě, chybová matice a snímek s číslicemi.

#### 7. Dokumentaci

Před samotným použitím sítě je nutné doinstalovat dva nástroje. Tento krok se provádí v Terminálu, kam se vypíšu příkazy:

```
pip install -U scikit-learn
```

```
pip install matplotlib
```

Následně je možné importovat všechny moduly a příslušné funkce.

Co program *basic\_NN* dělá?

Nejprve je pomocí funkce *load\_digits* naimportován dataset MNIST s číslicemi. V dalším kroku je set rozdělen na trénovací a testovací data. Následně je uživatel dotazován na zadání volitelných parametrů jako je počet iterací a solver. Pomocí funkce *MLPClassifier* je použita trénovací část datasetu pro trénování. Vložená funkce *warning* zachytává hlášku *Continuous*

*Integration* a ignoruje ji. Tato hláška upozorňuje na nedostatečný počet iterací pro kvalitní trénování.

Funkcí *mlp.predict* síť predikuje o jakou číslici by se mělo jednat. Následně se připravuje chybová matice, která je jako výstup vytisknuta a uvádí, kde došlo k záměně.

### Co program *basic NN 02* dělá?

V této části bylo úkolem vyzkoušet zejména načtení další části datasetu MNIST, k čemuž slouží funkce *fetch\_openml*. Stažení datasetu chvíli trvá (záleží na výkonosti počítače) proto jsou přidány informativní printy: Downloading dataset a Downloading complete. V dalším kroku je set rozdělen na trénovací a testovací data. Pomocí funkce *MLPClassifier* je použita trénovací část datasetu pro trénování. Opět je vložena i funkce *warning* vysvětlená výše. Následně jsou vytisknuty skóre dosažené testováním a trénováním.

V poslední části jsou vytvořeny podgrafy a zobrazeny váhy se stejným měřítkem.

## 8. Závěr, možné či neřešené problémy, náměty na vylepšení

Závěrem se převážně podařilo proniknout do základů NN a pochopit první principy fungování. Dále by bylo vhodné lépe proniknout do využití aktivačních funkcí, jejich použití a aplikace. Následujícím úkolem je otestování co největšího počtu volitelných parametrů, otestovat jejich vliv na běh a přesnost výpočtů a mít tak zásobu, ze které bude možné do budoucna čerpat.

Následně bude cílem proniknout a otestovat ML knihovny jako jsou PyTorch, TensorFlow – Keras. Obsahují silné nástroje strojového učení, které jsou vhodné mimo jiné i ke zpracovávání obrazových dat. S tímto krokem souvisí i pokusy o sestavení složitějších algoritmů nejen základních neuronových sítí, ale i dalších složitějších sítí jako jsou konvoluční neuronové sítě nebo například rekurentní neuronové sítě.

## 9. Seznam literatury

Online learning course: <https://www.sololearn.com/learning/1094>

Oficial webpage of scikit – learn: <https://scikit-learn.org/stable/index.html>

Documentation for Python: <https://docs.python.org/3/>