

Gymnázium Christiana Dopplera, Zborovská 45, Praha 5

ROČNÍKOVÁ PRÁCE  
**Regresní neuronové sítě**

Vypracoval: Lukáš Čaha  
Třída: 8.M  
Školní rok: 2017/2018  
Seminář: Seminář z programování

Prohlašuji, že jsem svou ročníkovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím s využíváním práce na Gymnáziu Christiana Dopplera pro studijní účely.

V Praze dne 16. prosince 2017

Lukáš Čaha

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Základní pojmy</b>	<b>4</b>
2.1	Neuron . . . . .	4
2.1.1	Jádro . . . . .	4
2.1.2	Synapse . . . . .	4
2.2	Síť . . . . .	4
2.3	Pohyb dat . . . . .	5
2.3.1	Vstup . . . . .	5
2.3.2	Výstup . . . . .	5
<b>3</b>	<b>Vstup</b>	<b>6</b>
3.1	Design vstupní vrstvy . . . . .	6
3.2	Scaling . . . . .	6
3.3	Data . . . . .	6
<b>4</b>	<b>Forward-propagation</b>	<b>7</b>
4.1	Typy sítí . . . . .	7
4.1.1	Normální . . . . .	7
4.1.2	Deep learning . . . . .	7
<b>5</b>	<b>Back-propagation</b>	<b>8</b>
5.1	Loss function . . . . .	8
<b>6</b>	<b>Praktická práce s PC</b>	<b>9</b>
6.1	Jazyk . . . . .	9
6.1.1	Vývojové prostředí . . . . .	9
6.1.2	Knihovny . . . . .	9
6.2	Git . . . . .	9
<b>7</b>	<b>Závěr</b>	<b>10</b>
	<b>Literatura</b>	<b>11</b>
	<b>Přílohy</b>	<b>12</b>

# 1. Úvod

Ve světě se nachází mnoho dat v mnoha podobách a v dnešní době se dostáváme do bodu, kdy nestačíme všechny třídit a využívat na 100 %.

Neuronové sítě jsou vrcholem lidské práce v oblasti informačních technologií. Mohl bych je přirovnat k lidskému mozku. A důvodem proč je zde zmiňuji je právě jejich všestranost. Sítí můžeme pouštět dva typy dat. Jedak lidmi vyhodnocené, a poté nevyhodnocené u nichž budu chtít výsledek. Neuronové sítě se podle prvního typu dat naučí jaká je souvislost mezi vstupem a výstupem a pak můžou přibližně určit výstupy dat druhého typu. Pokud byla v první řadě síť správně designovaná můžeme očekávat výsledky s poměrně velikou přesností a rychlostí zpracování, na jakou jsem zvyklí u počítačů.

Touto prací bych chtěl rozebrat neuronové sítě na úroveň pochopitelnou i pro středoškoláky, kteří by chtěli začít se strojovým učením, což je obor zahrnující moderní způsoby práce s umělou inteligencí.

## 2. Základní pojmy

### 2.1 Neuron

#### 2.1.1 Jádro

**Aktivace** je hodnota mezi nulou a jedničkou ( $a = 0.73$ ). Tahle hodnota určuje míru zapnutosti neuronu. Více aktivované neurony mohou mít větší vliv na neurony v síti přímo následující. Aktivace neuronů jsou závislé na datech, takže není možné měnit jejich hodnoty přímo.

**Normalizační funkce** upravuje příchozí signály, tak aby následně vytvořená hodnota zapadala do rozmezí aktivací. Přijdou-li do neuronu 4 signály všechny s maximální hodnotou, bude aktivace neuronu velmi blízko 1.

#### 2.1.2 Synapse

**Synapse** je spojení mezi dvěma neurony. Toto spojení zajišťuje, že aktivace neuronu v síti je závislá na aktivacích předchozích neuronů.

**Váha** ovlivňuje spoje mezi neurony. Váhy spojení tvoří dohromady povahu sítě. Z libovolných vstupních dat můžeme upravováním síly spojení (synapsí) vyvodit libovolné výstupní data.

### 2.2 Síť

Síť se skládá z několika vrstev, které jsou navzájem propojené.

**Vrstva** je několik neuronů, které se navzájem neovlivňují, ale jsou ovlivněny stejnými neurony a zároveň ovlivňují stejné neurony.

**Bias** je míra vlivu nezávislého na datech. Tato externí síla se stará o vyrovnaní sítě s menším počtem neuronů a tím uspořídá výpočetní výkon. V překladu je bias šum, který rozostřuje data, aby se výsledná síť nepřizpůsobila až příliš moc trénovacím datům.

## 2.3 Pohyb dat

### 2.3.1 Vstup

#### Typy vstupních dat

**Trénovací data** jsou data u nichž používáme vstupy i výstupy pro vylepšování sítě. Pokud výsledná síť uvidí znovu tato data bude na nich mít mnohem lepší výsledky, jelikož je trénovaná speciálně na tyto data a až jako vedlejší produkt je trénink na data podobná.

**Testovací data** je soubor vstupů, u nichž je známý i výsledek. Ten ale nikdy není ukázán síti, slouží totiž pro porovnání výsledku sítě s pravidlým výsledkem. Takto získává uživatel statistiky o kvalitě sítě.

**Produkční data** jsou důvod proč síť vůbec programujeme. Tyto data dostává síť během běžného používání a počítá k nim výsledky. Není však možnost určit jak by tyto výsledky měli vyjít, a proto nám už zbývá pouze doufat, že síť funguje jak popisuje teorie.

**Scaling** je metoda upravení hodnot z našich vstupních dat, tak aby v síti tato data vystupovala pouze jako aktivace. Dobrým příkladem je vstupní hodnota věk. V našich datech se vyskytuje člověk s maximální věkem 100 a minimálním 0. Odpovídající hodnoty aktivace potom budou  $100 \rightarrow 1.0$  a  $0 \rightarrow 0.0$ .

### 2.3.2 Výstup

**Back-scaling** je forma získání dat zpět z neuronové sítě. Pokud zrovna trénujeme, není nutné data získávat a pak je porovnávat s očekávanými výsledky, lepší způsob je očekávané výsledky převést na jazyk, kterým komunikuje síť. Tímto samozřejmě myslím použít scaling a převést výsledek na hodnotu mezi nulou a jedničkou.

## 3. Vstup

### 3.1 Design vstupní vrstvy

### 3.2 Scaling

### 3.3 Data

## 4. Forward-propagation

### 4.1 Typy sítí

#### 4.1.1 Normální

#### 4.1.2 Deep learning



## 5. Back-propagation

### 5.1 Loss function

## 6. Praktická práce s PC

### 6.1 Jazyk

Jako vývojový jazyk jsem si zvolil Python. Konkrétně používám verzi 3.6.3 na operačním systému Windows 7 64-bit. Tento jazyk jsem si zvolil především, protože je v oboru výrazně preferovaný. Tomu odpovídá i množství knihoven, které pro python v oblasti strojového učení vzniklo.

#### 6.1.1 Vývojové prostředí

Pro vývoj používám NotePad++. IDE zvýrazňuje klíčová slova ve většině jazyků a není komplikované.

#### 6.1.2 Knihovny

Knihovny usnadňují práci s programem. Můžeme si je představit právě jako knihy z knihovny plné funkcí, které jsou většinou velmi jednoduché, ale i přes to vyžadují čas na vytvoření a organizaci zdrojových souborů. Navíc jsou dost často optimalizované na to co dělají.

**NumPy** poskytuje značné množství jednoduchých matematických operací a funkcí, ze kterých je strojové učení složené. Hlavní funkce je pro mne násobení matic, kterou je možné rychle vyhodnocovat forward-propagation.

### 6.2 Git

Pro zálohování a verzování používám Git.

## 7. Závěr

Toto je závěr mé ročníkové práce.

# Literatura

- [1] Birge J. R., Wets R. J.-B. (1987): Computing bounds for stochastic programing problems by means of a generalized moment problem. *Mathematics of Operations Research* **12**, 149-162.

# Přílohy