



**Gymnázium
České Budějovice
Jírovцова 8**

MATURITNÍ PRÁCE

Neuronová síť

Jonáš Havelka

vedoucí práce: Dr. rer. nat. Michal Kočer


Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně s vyznačením všech použitých pramenů.

V Českých Budějovicích dne podpis

Jonáš Havelka

Abstrakt





Neuronové sítě se dnes objevují všude, ať už je to vyhledávání, překládání, nebo třeba jen zpracovávání dat. Mnoho programovacích jazyků má své knihovny pro práci s umělou inteligencí, ale zrovna  Kotlin, který lze použít skoro kdekoliv (Webové stránky, servery, mobily) takovou knihovnu postrádá.

Klíčová slova

Neuronové sítě,  Kotlin, Umělá inteligence, Multiplatformní knihovna

Poděkování

Poděkování patří hlavně mému učiteli informatiky, který je zároveň vedoucím mé práce, za skvělou výuku na hodinách a velkou trpělivost při kontrole našich prací. Také nesmím zapomenout na Alžbětu Neubauerovou, která mě celý rok podporovala a několikrát provedla korekturu mé práce.

Dále bych rád poděkoval všem komunitám, jejichž nástroje jsem používal, tj. JetBrains, v jejichž programovacím jazyce  Kotlin programuji a jejichž prostředí IntelliJ k tomu využívám,  Gradle, které používám ke kompilaci, L^AT_EX, ve kterém píšu text a dále  git a  GitHub, jež uchovávají má data, ať už text nebo knihovnu.


Obsah


I	Teoretická část	8
1	Strojové učení a umělá inteligence	9
2	Neuronové sítě	10
2.1	Laický náhled	10
2.1.1	Neuron	10
2.1.2	Sítě	10
2.1.3	Dopředná propagace a zpětná propagace	11
2.2	Formální náhled	11
II	Praktická část	12
3	Struktura knihovny	13
3.1	core	13
3.1.1	INeuralNetwork	13
3.1.2	BasicNeuralNetwork	13
3.2	mnistDatabase	13
3.2.1	Databáze MNIST	13
3.2.2	Databáze EMNIST	14
4	Používání knihovny	15
	Apendix	15
	Bibliografie	17
	Slovníček pojmů	18

Přílohy	21
Fotky z pokusů	21
Vlastní program	CD
Dokumentace	CD
Testovací data	CD

Úvod

Neuronové sítě jsou v poslední době velmi skloňované téma. Nikdo vlastně pořádně neví, jak fungují, ale fungují¹. Cílem této práce však nebude zkoumat neuronové sítě, ale implementovat je v co největším rozsahu (ať už struktury bez širšího využití jako asociativní paměť nebo často používané konvoluční sítě na rozpoznávání obrázků).

 Kotlin je ideální programovací jazyk pro vývoj knihovny, protože umožňuje tuto knihovnu používat jak pro JVM

Celá maturitní práce je k dispozici na  GitHubu, text včetně zdrojového L^AT_EXu na adrese https://github.com/JoHavel/Maturitni-Seminarni-Prace/tree/my_work a knihovna samotná pak na <https://github.com/JoHavel/NeuralNetwork>.

¹Tady se hodí podotknout, že alespoň malou představu máme, přece jenom matematicky je to pouze sestup po gradientu, ale překvapivě to dokáže velmi mnoho

Část I

Teoretická část

Kapitola 1

Strojové učení a umělá inteligence

Kapitola 2

Neuronové sítě

2.1 Laický náhled

2.1.1 Neuron

Počítačové neuronové sítě nejsou jen výmysl lidí, jejich základ nalezneme nervových soustavách živočichů. Základní stavební jednotka takové soustavy (stejně tak i neuronové sítě) je neuron. Neuron funguje tak, že přes dendrity přijímá elektrické (přesněji iontové) signály od jiných neuronů a když součet signálů přeteče určitou danou mez, vyšle neuron signál přes axony dál do dalších neuronů.

Přenos signálu z axonu do dendritu se odehrává v malých prostorách mezi nimi zvaných synapse. Vodivost synapsí je ovlivněna jejich chemickým složením, a proto se domníváme, proces učení probíhá měněním těchto chemických spojů (Brookshear et al. 2013, s. 491).

Náš umělý neuron tedy bude mít seznam dendritů (nesoucích informací, z jakého neuronu vedou signál a jak ho mění synapse), tzv. aktivační funkci (tedy jak silný signál posílá dále v závislosti na součtu vstupních signálů) a výstupní signál. Často navíc bude obsahovat základní hodnotu (angl. bias), která reprezentuje hladinu iontů v neuronu.

2.1.2 Sítě

Jelikož nahodilé neurony by se těžko udržovali v paměti a operace na nich by byly velmi pomalé, potřebujeme síť nějak uspořádat. Nejjednodušším uspořádáním jsou vrstvy. Každý neuron z nějaké vrstvy má dendrity ze všech neuronů z vrstvy minulé. Tak se předejde cyklům, které jsou složité na výpočty, a navíc si nemusíme u každého neuronu pamatovat, ze kterých neuronů do něj vede signál.

2.1.3 Dopředná propagace a zpětná propagace

Dopředná propagace (častěji se používá anglický výraz forward propagation) je jednoduše spočítání signálů ve všech neuronech. Tedy u každého neuronu se sečtou vstupní signály (popř. přičte bias) a spočítá se funkční hodnota aktivační funkce v tomto bodě.

Naopak zpětná propagace (častěji se používá anglický výraz backward propagation) je na základě chyby, kterou spočítáme z výstupu neuronové sítě a předpokládaného výstupu, určit, které proměnné hodnoty (synapse a biasy) se na ní nejvíce podílejí. Potom tyto hodnoty posuneme odpovídajícím způsobem (stejně jako příroda mění chemické vlastnosti synapse)

2.2 Formální náhled

Nechť $n_i = (\{w_{x,i}\}, f_i, b_i, v_i)$ je neuron, kde $w_{x,i} \in \mathbb{R}$ je hodnota (angl. weight), kterou synapse jdoucí z neuronu n_x násobí signál, $f_i : \mathbb{R} \leftarrow \mathbb{R}$ je aktivační funkce, $b_i \in \mathbb{R}$ je bias a $v_i \in \mathbb{R}$ je signál vycházející z daného neuronu. Potom dopředná propagace (tedy spočítání v_i) vypadá takto:

$$v_i = f_i \left(b_i + \sum_x v_x \cdot w_{x,y} \right)$$

Což lze zapsat vektorově jako:

$$\vec{v} = (v_{x_1}, v_{x_2}, \dots)$$

$$\vec{w}_i = (w_{x_1,i}, w_{x_2,i}, \dots)$$

$$v_i = f_i(b_i + \vec{w}_i \cdot \vec{v})$$

Nebo můžeme do hodnot „zakomponovat“ i bias:

$$\vec{v} = (v_{x_1}, v_{x_2}, \dots, 1)$$

$$\vec{w}_i = (w_{x_1,i}, w_{x_2,i}, \dots, b_i)$$

$$v_i = f_i(\vec{w}_i \cdot \vec{v})$$

Část II

Praktická část

Kapitola 3

Struktura knihovny

Knihovna je rozdělena do dvou částí:

- První, a ta hlavní, je `core` (česky jádro), které obsahuje definice neuronových sítí (tj. konvoluční neuronovou síť, obyčejnou neuronovou síť, asociativní paměť) a definice pro ně potřebné (například aktivační funkce)
- Druhá je `mnistDatabase`, která se stará o učení neuronových sítí na datových databázích.

3.1 `core`

3.1.1 `INeuralNetwork`

Rozhraní (anglicky interface), které implementuje základní funkce neuronových sítí, které mají jako vstup i výstup `Double ()` vektor

3.1.2 `BasicNeuralNetwork`

Tato třída

3.2 `mnistDatabase`

3.2.1 Databáze MNIST

„Databáze MNIST, databáze ručně psaných číslic dostupná na stránkách <http://yann.lecun.com/exdb/mnist/> obsahuje 60 000 tréninkových a 10 000 ověřovacích příkladů. MNIST

je vytvořena z databáze spravované NIST (National Institute of Standards and Technology). Číslice mají normalizovanou velikost a jsou vycentrované v obrázcích shodné velikosti.“ (LeCun et al. 2019, přeloženo)

Tuto databázi jsem použil pro první testování BasicNeuralNetwork, jelikož má pro první testování dostačující velikost. Pro pozdější testování využívám převážně EMNIST.

3.2.2 Databáze EMNIST

„Databáze MNIST se stala standardem pro učení umělého vidění. Databáze MNIST je odvozená z databáze NIST Special Database 19, která obsahuje ručně psané číslice a velká i malá písmena. EMNIST (Extended MNIST), varianta celé databáze NIST, přebírá uspořádání z databáze MNIST¹.“ (Cohen et al. 2017, přeloženo)

Tato databáze obsahuje více příkladů než MNIST, navíc obsahuje i sety s písmeny, proto jsem po prvních pokusech s MNIST přešel na tuto databázi.

¹Má však prohozené řádky a sloupce obrázků.

Kapitola 4

Používání knihovny

Závěr

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Bibliografie

- Brookshear, J. Glenn, David T. Smith a Dennis Brylow (2013). *Informatika*. Czech. Přel. English Jakub Goner. 1. vyd. Brno, CZ: Computer Press. 608 **pagetotals**. ISBN: 978-80-251-3805-2.
- Cohen, Gregory et al. (2017). “EMNIST: an extension of MNIST to handwritten letters”. In: *arXiv preprint arXiv:1702.05373*.
- LeCun, Yann, Corinna Cortes a Christopher J.C. Burges (15. pros. 2019). *THE MNIST DATABASE*. of handwritten digits. English. URL: `\url{http://yann.lecun.com/exdb/mnist/}`.

Slovníček pojmů

axon výběžek vedoucí signál z neuronu. 10, 18

dendrit výběžek vedoucí signál do neuronu. 10, 18

interface česky rozhraní je v objektově orientovaném programování zabalení funkcí a vlastností třídy, které by měla každá třída z nějaké skupiny mít (např. každá fronta by měla mít funkci pro přidání a odebrání prvku a jedna z jejích vlastností je velikost). 13

JVM Java Virtual Machine je virtuální stroj, který umožňuje běh Java Bytecodu, kódu, do kterého se překládá Java a Kotlin. 7

Kotlin programovací jazyk vyvíjený firmou JetBrains, založen na Javě. 3, 4, 7

synapse spojení (mezera) mezi axonem a dendritem, jež podle svých chemických vlastností zesílí nebo zeslabí signál předávaný z axonu do dendritu. 10, 11

třída anglicky class je základní prvek objektově orientovaného programování. Obsahuje funkce a vlastnosti, které bude mít objekt, který se vytvoří z dané třídy (popřípadě třídy, jež budou z této třídy dědit). 18

Seznam obrázků

Seznam tabulek

Přílohy

1. Fotky z pokusů
2. Vlastní program
3. Dokumentace
4. Testovací data