VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra Informatiky

Deep learning v analýze obrazu

Deep learning of image Analysis

2018  David Kuchař

Naskenované zadání

Podepané prohášení

**Poděkování**

Rád bych poděkoval panu …………………….., který byl mým vedoucím práce, za všechny rady, poskytnuté materiály a čas strávený nad problematikou této diplomové práce.

**Abstrakt**

**Klíčová slova**

**Abstract**

**Key words**

**Obsah**

[1. Seznam použitých zkratek 8](#_Toc498692698)

[2. Seznam použitých symbolů 9](#_Toc498692699)

[3. Seznam tabulek 10](#_Toc498692700)

[4. Seznam obrázků 11](#_Toc498692701)

[5. Úvod 12](#_Toc498692702)

[6. Neuronové sítě 13](#_Toc498692703)

[6.1. Bilogický neuron 13](#_Toc498692704)

[6.2. Umělý neuron 13](#_Toc498692705)

[6.3. Umělá neuronová síť 15](#_Toc498692706)

[7. Modely neuronové sítě 16](#_Toc498692707)

[6.4. Perceptron 16](#_Toc498692708)

[8. Návrh prohramování 18](#_Toc498692709)

[9. Praktická část 19](#_Toc498692710)

[10. Závěr 20](#_Toc498692711)

[11. Použitá literatura I](#_Toc498692712)

[12. Přílohy III](#_Toc498692713)

# Seznam použitých zkratek

# Seznam použitých symbolů

# Seznam tabulek

# Seznam obrázků

# Úvod

# Neuronové sítě

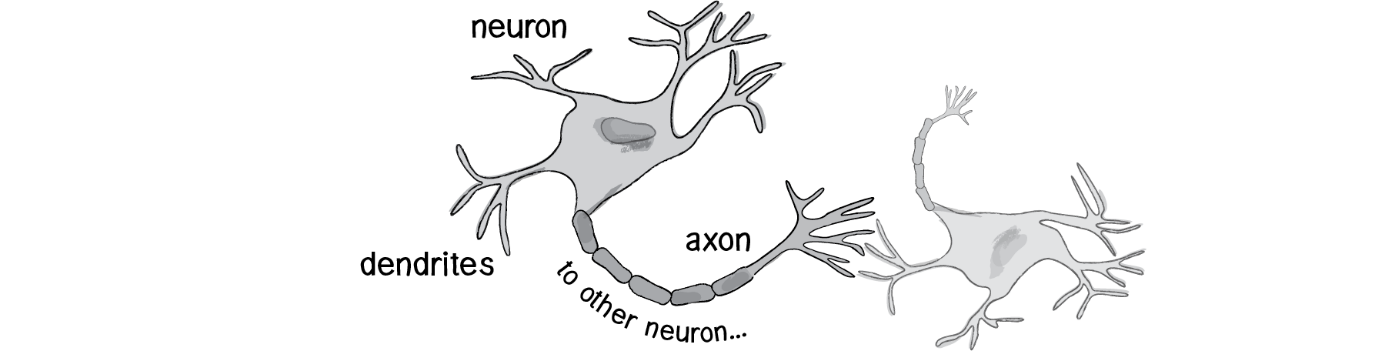
Teorie neuronových sítí je založena na poznacích o spracování informací v nervových buňkách. Světoví odbornící již bylo dlouho dobu inspirování funkcí lidského mozku, když v roce 1943 neurofyziolog Warren S. McCulloch a Walter Pillts vyvinuli první konceptuálni model umělé neuronové sítě. Jejich práce nebyla určená k tomu, jak přesně biologický mozek funguje, ale jak pomocí umělé neuronové sítě řešit určité druhy problémů. Existují problémy, se kterými sis počítač poradí velice snadno např. výpočty nebo vzorce. Naopak jsou pak problémy se kterými si člověk poradí, ale počítač nikoliv. Například rozpoznávání vzoru nebo lidských tváři. Pro tyto případy se začli využívat umělé neuronové sítě. [1]

## Bilogický neuron

Neurony jsou buňky v živých organismech, které umožnují vést signály a reagovat na ně.Každý neuron má několik dendritů, které mají za úkol přijímat signál z okolí a jeden výběžek tzv. axon schopný signál vyslat. Na konci je axon také rozvětven. Axon každého neuronu je zakončen tzv. synapsemi, které dosedají na jiné neurony. Přes synapse se přenášejí vzruchy mezi neurony. Rychlost přenosu je v rozmezí 5–125 m/s.

Nervová soustava člověka obsahuje cca 1011–1012 neuronů. S rostoucím věkem počet neuronů ubývá. Neurony mohou mít desitký, ale někdy až stovky tisíce dendritů, kterí jsou vzájemně propojeny. Odumře-li velká část neuronů, ztrátu lze nahradit zvětšením počtu spojů – dendritů.

Pomocí dendritů neurony zachycují signály. Tyto signály se šíří dovnitř buňky, kde vzniká potenciál. Je-li tento potenciál dostatečně velký, neuron je schopen sám vyslat signál dál [2]



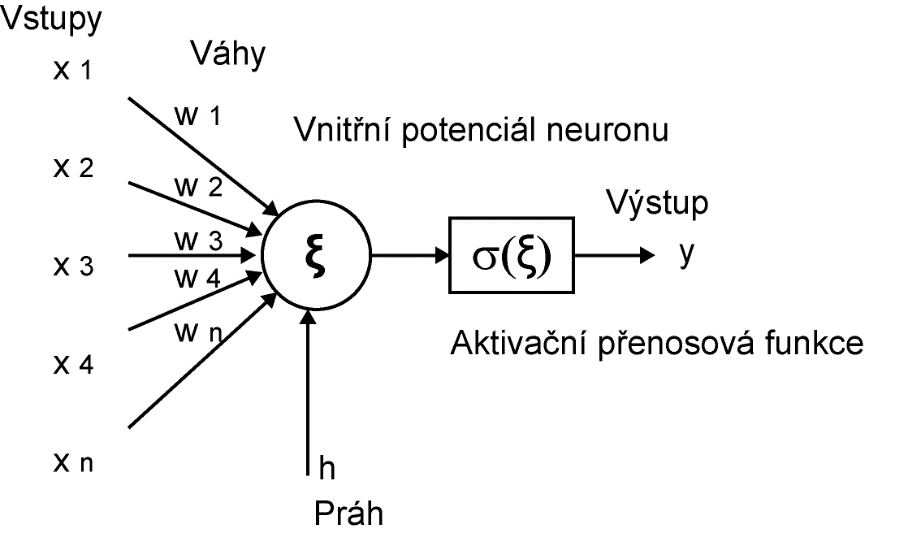
Obrázek 1 Schéma biologockého neuronu [1]

## Umělý neuron

Model umělého neuronu představuje zvětší části abstraktní mechanizmus, jak nervové bunky spracovávají informace. Nelze totiž vytvořit přesnou analogii modelu skutečného neuronu. Proto modely umělých neuronů, které se v součastnoti používají, opisují pouze základní funkci neuronu.

Klasický umělý neuron McCulloch-Pitts(MCP), nebo také perceptron je tvořen několika vstupy x1-xn a jedním výstupem y. Každý vstup má svou vlastná vahu, která muže být i záporná. Vážená suma vstupních hodnot představuje vnitřní potenciál neuronu, který lze vypočítat podle vztahu:

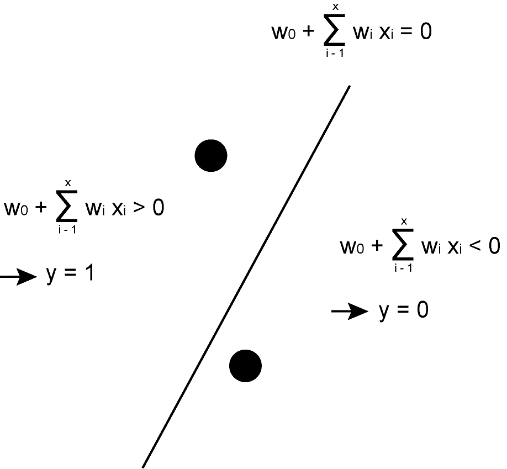
kde x1 až xn jsou vstupy neuronu, w1 až wn je vektor vah jednotlivých vstupů a bias(práh) je konstatní vstup neuronu. Z praktických důvodů se práh modeluje jako jedna z vah tak, že vstupní vektor i vektor vah je rozšířen o nultou pozici. Vstup na nulté pozici je vždy uvažován za roven 1 a nultá váha je nastavena na hodnutu h(práh). V takovém případě ses prah stává jednou z vah a v průbehu trénování podléhá adaptaci.



Obrázek 2 Schéma umělého neuronu [3]

Pokud potenciál neuronu má dostatečně velkou hodnotu vyšle signál. Aktivační přenosová funkce se obecně používá nelineární funkce transformující hodnotu vnitřního potencialu neuronu. Nejčastějí sigmoid. Pro ilustraci však předpokládejme, že použijeme nejjednodušší nelineární typ, ostrou nelinearitu, kdy platí:

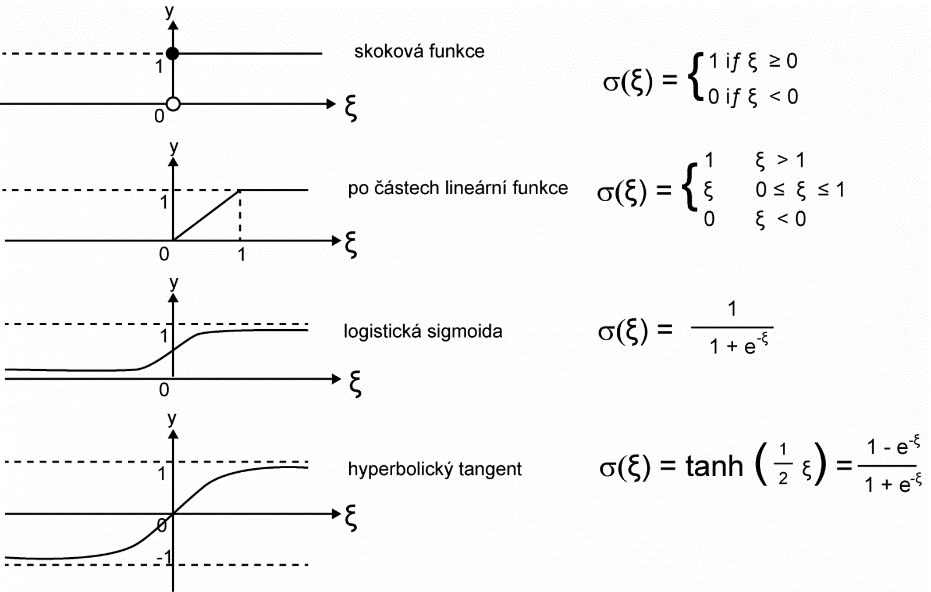
Předpokládejmě neuron pro n = 2 se dvěma reálnými vstupu x1 a x2 a vahami w1 a w2. Takto definovaný neuron {R}2 -> {0,1}. Neuron ve své aktivní dynamice reaguje na vstupy a přiřazuje jim hodnotu 0 nebo 1. Provádí tak klasifikaci techto bodů do dvou skupin podle hondoty aktivační funkce tz. Výstup neuron. V tomto konkrétním přápade je zařazení bodu dáno jejich pozicí vůči přímce definované aktivační funkcí (váhou neuronu). Dělící přímkra rozděluje dvojrozměrný proctor na dvě skupiny viz. obrázek c.2.



Obrázek 3 Ilustrace klasifikace neuronu v rovině [3]

Aktivační přenosové funkce nejčastějí používané v neuronových sítí lze vidět na obrázku č.3.

Existují i odlišné koncepty neuronu, kde výstup neuron je kalkulován jiným způsobem např. Sítě na radiální bázi (RBF sítě), zde neuron vyčísluje vzdálenost vstupního vektoru x id vektoru vah w nebo tkaé vlnkové sítě.[2][3]



Obrázek 4 Aktivační přenosové funkce[3]

Hlavní vlastností neuronových sítě je schopnost se učit. Neuronová sít je složitý systém, který jes schopný dle inforamcí, které zpracovává se vyvíjet. Jsou-li výsledky sítě dobré váhy neuronu se nezmění, v jíném případě se vahý pozmění tak, aby výsledky vycházeli lépe.[1]

**Máme několik kategorii učení:**

* **Učení s učitelem** (Supervised Learning) – při učení jsou neuronové sítí předkládány požadované výsledky a srovnávány s výstupem neuronové síte. Podle rozdílů pak určujeme další kola učení.
* **Učení bez učitele** (Unsupervised Learning) – není založeno na vyhodnocování výstupu, sít se sama snaží třídí vstupy dle podobnosti a charakteristických znaků.
* **Kombinace učení s učitelem a bez** (Semi Supervised Learning) -část výstupu je porovnána s požadovaným výstupem. Další data jsou poté vyhodnocena bez známého výstupu.
* **Zpetnovazebné učení** (Reinforcement Learning) – založené na předchozích zkušenostech a okolnostech. Např. Když robot narazí zapamatuje si, že tudy neprojede.

## Umělá neuronová síť

Každá umělá neuronová síť se skládá ze vzajmeně propojených neuronů, takže te výstup u neuronu je vstup do jiných neuronů. Počet neuronů a jejich vzájmené propojení určuje topologie sítě.

**Topologii rozdělujeme na dva zakladnbí typy:**

* Neuronovou sít s dopředným šířením signáluli (feedforward)
* Neuronová sít se zpetnovazebním šířením signalu (feedback)

U neuronové síte s dopředným šíření signalu postupují všechny signalýze ve směru ze vstupní vrstvy do vrstvy vystupní bez zpetné vazby. Zatím co u sítí se zpětnovazebním šířením signálu je vstup každého neuronu zavislá na hodnotš výstupu z předchozího cyklu.

Neurony jsou strukturované do vrstev. Každá neuronová síť se skládá minimálně ze dvou vrstev, a to vstupní a výstupní případně dalších vrstev tzv. skryté. Cílem vstupní vrstvy je zabezpečit distribuci vstupních signalů sítě do ostatních vrstev. Neuron v ní má pouze jeden vstup a posílá vstupní sígnál na výstupní beze změny. Výstupní vrstva určuje výstup neuronové sítě.[1][2][4]

# Modely neuronové sítě

## Perceptron

Perceptron je nejednodušší model neuronové sítě, kterou v roce 1957 vyvinul Frank Rossenblatt. Prvotně navrhnut, jako model zrakové soustavy. V dněšní době se často nazývá neuron. Nejzákladnější úlohou,kterou řeší Perceptron je klasifikace např.klasifikace výrobku do dvou skupin.Později se však došli k závěru že Perceptron nemá tak široké vužití kvůli použitelnosti pouze v klasifikaci lineárně separovaných skupin.Proto došlo k rozšíření na více vrstvý perceptron MLP (Multi Layered Perceptron).

Perceptron se typická skládá z n vstupu, procesoru a jednoho výstupu. Každá vstup je vynásoben vaho,která je mezi hodnotami <-1,1> .Tato vážené vstupy se nasledně sečtou a vyhodnoti v aplikační funkci, která pote vyšle výstupní signal. Například vyhodnocení zda je číslo kladné či záporné. Jeli kledné vyšle 1 jinak -1.

Při učení perceptronu se používá metoda učení s učitelem Hebbova metoda pro adaptaci perceptronu ve které je navrženo učící pravidlo. Do perceptronu se posílá udak o očekávaném výsledku. Jsou-li výsledky odlišné aktivuje sae funkce,která pozmení váhy u vstupů dle velikosti chyby.Ta se definuje jako rozdíl mezi očekávaným výsledkem a vypočteným.Jsou-li výsledky schodné výsledek chyby je 0,jeli výpočet -1 a očekávaný výsledek +1 chyba je -2. Chyby jsou vyobrazeny v talbulce 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Očekavaný | | Pravý | | Chyba | |
| -1 | -1 | | 0 | |
| -1 | +1 | | -2 | |
| +1 | -1 | | +2 | |
| +1 | +1 | | 0 | |

Tabulka 1 Tabulka přehledu chyb

Vzorec pro výpočet chyby:

Základná vzorec pro novou váhu pro vstup:

Základná vzorec pro novou váhu pro vstup se zohledněním rychlosti učení:

Vysoký hodnota kontanty učení znamená, že se váha bude více drasticky měnit, tím dojde k řešení rychleji, ale s možností že překročí optimální váhu. S nižší hodnout této konstany se dospěje k výsledku pomeleji, ale za zo s lepší přesnotí.

U perceptronu můžeme použít i bias(práh), který napomíhá k lepšímu učení. Bias je vstupní konstatanta, který je po celou dobu uření stejná nadefinovaná na začátku. Pouze se u ní mění váha.[1][4][5]

1. class Perceptron {
2. float[] vahy; // vahy
3. float ucebni\_konstanta = 0.01; // ucebni konstanta
4. Perceptron(int n) {
5. vahy = new float[n]; // nahodne hodnoty vah
6. for (int i = 0; i < vahy.length; i++) {
7. vahy[i] = random(-1, 1);
8. }
9. }
10. int feedforward(float[] vstupy) {
11. float soucet = 0; // vypocet vysledku
12. for (int i = 0; i < vahy.length; i++) {
13. soucet += vstupy[i] \* vahy[i];
14. }
15. return activation(soucet);
16. } // aktivacni funkce
17. int activation(float soucet) {
18. if (soucet > 0) return 1;
19. else return -1;
20. } //trenovani pomoci vah a predpokladaneho vysledku
21. void train(float[] vstupy, int ocekavany\_vystup) {
22. int vystup = feedforward(inputs); // odhadovany vypocet
23. float chyba = ocekavany\_vystup - vystup; // vypocete chyby
24. for (int i = 0; i < vahy.length; i++) {
25. vahy[i] += ucebni\_konstanta \* chyba \* vstupy[i]; // vypocet novych vah
26. }
27. }

Příklad navrhu perceptronu klasifikující kladné a zaporné číslo[1]

1. }

## Více vrstvá neuronová sít

Více vrstvá neuronová sít je nejpoužívanější neuronová sít. Je tvořena opakovýnám základního prvku perceptronu a obsahuje tři hlavní vrstvy:

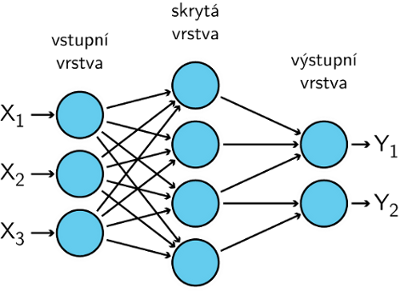
1. vstupní vrstva (input layer)

2. skrytá vrstva (hidden layer) – muže obsahovat více vrstev

3. výstupní vrstva (output layer).

## Feed forward sítě

Neuron je schopen rozděli prostor pouze na dva poloprostory viz, předešla kapitola.Ve složitější úlohách si však s rodělením do dvou poloprostor nevystačíme.A tak sestavuje rozsáhle neuronové, sitě složené s několik vrstev neuronové sítě vzájemně propojoných každý s každám. obrazek niže [6]



Obrázek 4 [7] Model vícevrstvé neuronové sítě

Každý neuron řeši pouze část problému. Výsledek je poté,kompozicí všech výstupních funkcí neuronu v síti. Přístup v přírodě je velmi podobný, pokud by nějaký neuron nefungoval z jakýchkoliv důvodů. Výsledek bude i tak správný, protože celková informace je rozprostěna mezi více neuronů.

Každý neuron rozděluje prostor na dva podprostory. První vrstva teda rozdělí prostor na několik polorovin. Výstup lze popsat více detailně. Druhá vrstava už nepřijmá přáznaky, ale příslušnoust k těmto rovinám.Třídy tedy můžou být rozděleny ne jen přímkami ale i křivkami a celý výsledek se tak zpřesňuje.

Nejčastěji se pro učení používá metoeda BackPropagation,která funguje na porovnýní s očekávaným výsledkem a podlě chyby upraví váhy neuronu.

# #TODO

Další Modely

Convulční neuronové síte

-jak co proc

Deep learning fremwork

Tensorflow

-popsat

Caffe

Theano

# Návrh prohramování

# Praktická část

# Závěr

# Použitá literatura

1. Chapter 10. Neural Networks [online]. [cit. 2017-10-15]. Dostupné z: Chapter 10. Neural Networks [online]. [cit. 2017-10-15]. Dostupné z: <http://natureofcode.com/book/chapter-10-neural-networks/>
2. Biologické algoritmy (4) - Neuronové sítě [online]. 2012 [cit. 2017-10-15]. Dostupné z: <https://www.root.cz/clanky/biologicke-algoritmy-4-neuronove-site/?ic=serial-box&icc=text-title>
3. Matematický model a aktivní dynamika neuronu [online]. [cit. 2017-10-15]. Dostupné z: <http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--jednotlivy-neuron--matematicky-model-a-aktivni-dynamika-neuronu>
4. KRAJČOVIČOVÁ, Mária. Konvulční neurovnová síť pro zpracování obrazu. Antonínská 548/1, 601 90 Brno-střed, 2015. Diplomová práce. Vysoké učení technice v Brně. Vedoucí práce Doc. Ing. RADIM BURGET, Ph.D.
5. Umělá inteligence I: Neuronové sítě [online]. [cit. 2017-11-17]. Dostupné z: <https://is.mendelu.cz/eknihovna/opory/index.pl?cast=21471>
6. Biologické algoritmy (5) - Neuronové sítě [online]. [cit. 2018-01-03]. Dostupné z: <https://www.root.cz/clanky/biologicke-algoritmy-5-neuronove-site/?ic=serial-box&icc=text-title>
7. Neuronové sítě. In: Neuronové sítě [online]. [cit. 2018-01-03]. Dostupné z: <https://is.mendelu.cz/eknihovna/opory/download.pl?objekt=23134>

Zdroje pro mě na využiti z edisonu

[1] Chapter 10. Neural Networks. The Nature of Code [online]. Dostupné z: <http://natureofcode.com/book/chapter-10-neural-networks/>

[2] JUERGEN, Schmidhuber. Deep Learning in Neural Networks: An Overview. ArXiv [online]. 2015, 88 DOI: 10.1016/j.neunet.2014.09.003. Dostupné z: <http://arxiv.org/abs/1404.7828>

[3] Convolutional Neural Networks (LeNet). Deep Learning: . . . moving beyond shallow machine learning since 2006! [online]. 2016 Dostupné z: <http://deeplearning.net/tutorial/lenet.html>

[4] SZEGEDY, Christian, WEI LIU, YANGQING JIA, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [online]. IEEE, 2015, s. 1-9 [cit. 2017-10-12]. DOI: 10.1109/CVPR.2015.7298594. ISBN 978-1-4673-6964-0. Dostupné z: <http://ieeexplore.ieee.org/document/7298594/>

[5] LECUN, Yann, Léon BOTTOU, Yoshua BENGIO a Patrick HAFFNER. Gradient-Based Learning Applied to Document Recognition [online]. 1998, 46 Dostupné z: <http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf>

# Přílohy