

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: B 2301 Strojní inženýrství
Studijní zaměření: Průmyslové inženýrství a management

BAKALÁŘSKÁ PRÁCE

Neuronové sítě a jejich využitelnost v rámci diskrétní simulační optimalizace

Autor: **Richard SLADKÝ**
Vedoucí práce: **Ing. Pavel Raška, Ph.D.**

Akademický rok 2016/2017

Poděkování

Rád bych tímto poděkoval vedoucímu bakalářské práce Ing. Pavlu Raškovi, Ph.D. za pomoc a odborné vedení při vypracování této práce. Dále bych chtěl poděkovat své rodině a nejbližším za jejich podporu.

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této bakalářské práce.

V Plzni dne:

.....

ANOTAČNÍ LIST BAKALÁŘSKÉ PRÁCE

AUTOR	Příjmení Sladký	Jméno Richard	
STUDIJNÍ OBOR	B2301 – Strojní inženýrství		
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Ing. Raška, Ph.D..	Jméno Pavel	
PRACOVISŤE	ZČU - FST - KPV		
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ	Nehodící se škrtněte
NÁZEV PRÁCE	Neuronové sítě a jejich využitelnost v rámci diskrétní simulační optimalizace		

FAKULTA	strojn	KATEDRA	KPV	ROK ODEVZD.	2017
----------------	--------	----------------	-----	--------------------	------

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	47	TEXTOVÁ ČÁST	47	GRAFICKÁ ČÁST	0
---------------	----	---------------------	----	----------------------	---

STRUČNÝ POPIS (MAX 10 ŘÁDEK) ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY	Tato bakalářská práce je zaměřena na neuronové sítě. V první části této práce jsou popsány základní typy neuronových sítí a jejich fungování. Druhá část je zaměřena na možnosti využití neuronových sítí v několika odvětvích a následně analýza použití v rámci diskrétní simulační optimalizace, pro kterou nejsou neuronové sítě nejvhodnější. Poslední kapitola obsahuje novinky v oblasti neuronových sítí.
KLÍČOVÁ SLOVA ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE	Neuronové sítě, perceptron, dopředné, síť, backpropagation, rekurentní, Hopfieldova síť, Elmanova síť, samoorganizační, Kohonenova síť, Counterpropagation, neuron, váha, práh

SUMMARY OF BACHELOR SHEET

AUTHOR	Surname Sladký		Name Richard	
FIELD OF STUDY	B2301 – Mechanical engineering			
SUPERVISOR	Surname (Inclusive of Degrees) Ing. Raška, Ph.D..		Name Pavel	
INSTITUTION	ZČU - FST - KPV			
TYPE OF WORK	DIPLOMA	BACHELOR	Delete when not applicable	
TITLE OF THE WORK	Neural Networks Used for Discrete Event Simulation Optimization			

FACULTY	Mechanical Engineering	DEPARTMENT	KPV	SUBMITTED IN	2017
----------------	------------------------	-------------------	-----	---------------------	------

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	47	TEXT PART	47	GRAPHICAL PART	0
----------------	----	------------------	----	-----------------------	---

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	This thesis is focused on a neural networks. Basic types of the neural networks and their applications are described in the first part of this thesis. Second part is focused on the possibilities of their use in several sectors and subsequently on their use for discrete event simulation optimization. Neural networks are not appropriate for this use. The last chapter contains news about neural networks.
KEY WORDS	Neural networks, perceptron, feedforward, backpropagation, recurrent, Hopfield's net, Elman's net, self-organized, Kohonen's net, Counterpropagation, neuron, weight, threshold

Obsah

Seznam obrázků	8
Seznam symbolů	9
1 Úvod do řešené problematiky	10
2 Historie neuronových sítí.....	10
2.1 Biologický neuron	11
2.2 Perceptron	12
3 Analýza současného stavu	15
3.1 Rozdělení neuronových sítí	15
3.2 Dopředné neuronové sítě	16
3.2.1 Perceptronová síť	16
3.2.2 Backpropagation.....	19
3.3 Rekurentní neuronové sítě	22
3.3.1 Hopfieldova síť.....	22
3.3.2 Elmanova síť	24
3.5 Samoorganizační neuronové sítě	25
3.5.1 Kohonenova síť	25
3.5.2 Counterpropagation	27
4 Použití umělých neuronových sítí	29
4.1 Technika a řízení	30
4.1.1 Couvání s návěsem.....	30
4.1.2 Samopilotované letadlo	31
4.2 Zpracování signálu	32
4.3 Informační technologie.....	32
4.3.1 Rozpoznávání vzorů	32
4.3.2 Produkce řeči.....	32
4.3.3 Rozpoznávání řeči	33
4.3.4 Identifikace témat textových zpráv	33
4.4 Ekonomika a finance	34
4.4.1 Hypotéka	34
4.4.2 Správa financí.....	35
4.5 Reklama a marketing	35

4.6	Zdravotnictví	36
4.7	Budoucnost	36
5	Neuronové sítě v rámci diskrétní simulační optimalizace.....	37
5.1	Ukázka	38
6	Novinky v rámci neuronových sítí	41
6.1	Neuronová síť Googlu začala mluvit, mlaskat a funět	41
6.2	Pomocí neuronové sítě můžeme navrátit efekt rozmazání textu či tváře	42
6.3	Český Google překladač začal používat umělou inteligenci	42
6.4	Neuronová síť se naučila odezírat ze rtů	43
6.5	Robotické stroje budou vybaveny umělou inteligencí, aby se automaticky zlepšovali ve výrobě.....	43
7	Závěr.....	45
	Bibliografie.....	46

Seznam obrázků

Obrázek 1: Schéma biologického neuronu [2].....	11
Obrázek 2: Perceptron [3]	12
Obrázek 3: Standartní (logistická) sigmoida [3]	12
Obrázek 4: Ostrá nelinearita [3]	13
Obrázek 5: Saturovaná lineární funkce [3]	13
Obrázek 6: Hyperbolický tangens [3]	13
Obrázek 7: Topologie perceptronové sítě [3].....	16
Obrázek 8: Vícevrstvá perceptronová síť [7]	18
Obrázek 9: Topologie Hopfieldovy sítě [10]	22
Obrázek 10: Topologie Elmanovy sítě [13]	24
Obrázek 11: Topologie Kohonenovy sítě [15].....	25
Obrázek 12 : Dopředná síť counterpropagation [2]	27
Obrázek 13: LoFLYTE [16].....	31
Obrázek 14: Boeing X-51 [17] a NASA X-43 [18]	31
Obrázek 15: Rozhraní aplikace	38
Obrázek 16: Různé výsledky aplikace (10 měst, 20 neuronů, 500 iterací)	38
Obrázek 17: Různé výsledky aplikace (10 měst, 20 neuronů, 2000 iterací)	39
Obrázek 18: Výsledky aplikace (30 měst, 50 neuronů, 500 iterací)	39
Obrázek 19: Výsledky aplikace (30 měst, 50 neuronů, 10000 iterací)	40
Obrázek 20: Výsledky testů jednotlivých technologií a porovnání s lidským hlasem [21].....	41
Obrázek 21: Ukázka využití programovatelných robotů [25].....	43
Obrázek 22: Robotické systémy od firmy Fanuc [25]	44

Seznam symbolů

α	-	Koeficient učení
c	-	Vzdálenost mezi městy
d	-	Požadovaný výstup
δ	-	Částečné váhové korekce
$E(w)$	-	Chyba sítě
$E_k(w)$	-	Parciální chyba sítě
h	-	Práh neuronu
i	-	Číslo vstupního neuronu
j	-	Číslo výstupního neuronu
m	-	Celkový počet výstupních neuronů
n	-	Celkový počet vstupních neuronů
r	-	Vzdálenost neuronů
R	-	Poloměr sousedství
s	-	Tréninkový vstup
t	-	Tréninkový výstup
τ	-	Tréninková množina
u	-	Fiktivní proměnná
v_0	-	Bias neuronu ve skryté (vnitřní) vrstvě
w	-	Synaptická váha spoje
w_0	-	Bias neuronu ve výstupní vrstvě
x	-	Informace ve formě reálného čísla (vstup neuronu)
X	-	Množina vstupních neuronů
ξ	-	Potenciál neuronu
y	-	Výstup neuronu
Y	-	Množina výstupních neuronů
Z	-	Množina vnitřních neuronů
z	-	Vnitřní neuron
\mathbb{Z}	-	Množina celých čísel

1 Úvod do řešení problematiky

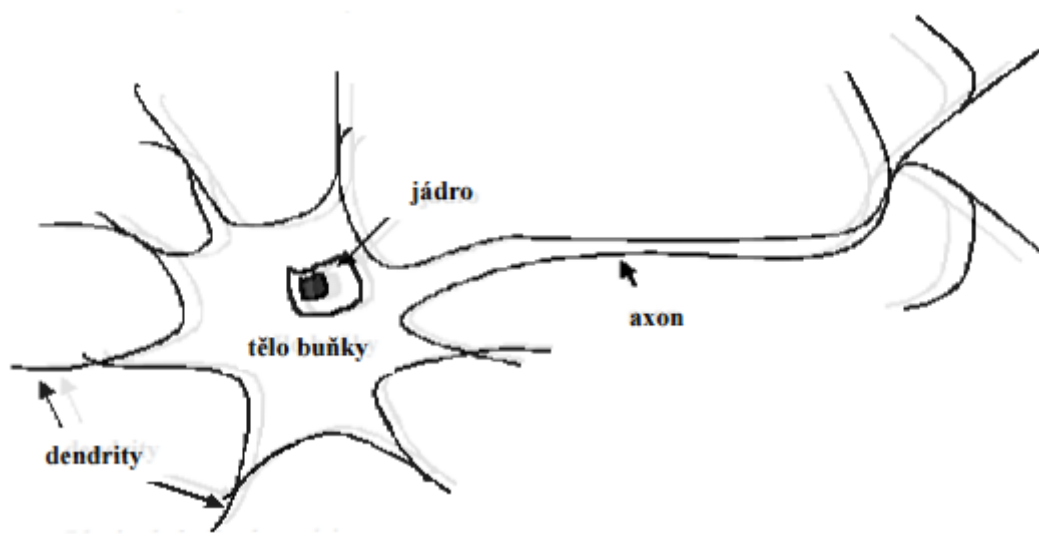
V úvodní kapitole je nastíněna historie neuronových sítí, dále je zde popsán biologický neuron, ze kterého vychází matematický model umělého neuronu. Tento model se nazývá Perceptron a je blíže popsán na konci této kapitoly.

2 Historie neuronových sítí

Člověk se o mozek a myšlení zajímal už od pradávna. Snaha lépe pochopit funkce mozku vedla člověka k potřebě jeho matematického modelování. Poprvé se práce s modely vytvořenými na základě neurofyziologie objevily ve 40. letech minulého století. Warren S. McCulloch a Walter Pitts v roce 1943 publikovali článek, ve kterém vytvořili model neuronu jako dvoustavové prahové zařízení. Tím položili základy k umělým neuronovým sítím. V roce 1949 Donald Hebb odvodil první zákon učení. První algoritmus učení pro vícevrstvé neuronové sítě s dopředným šířením signálu byl publikován až o devět let později, tedy v roce 1958, autorem Franckem Rosenblattem. Tato síť byla nazvána Perceptron. To byl začátek prvního optimistického období rozvoje umělých neuronových sítí. Roku 1969 zveřejnili matematici Marvin L. Minski a Seymour S. Papert zásadní kritiku schopností umělých neuronových sítí. Jejich závěrem bylo, že neuronové sítě nejsou schopny nahradit klasické metody, protože nemohou simulovat všechny logické zákony (jednovrstvý Perceptron nedokáže řešit problém XOR). Vysvětlili také, že problém může být řešen zkonstruováním vícevrstvého Perceptronu, pro který musíme ovšem parametry najít „ručně“ a nelze je automaticky předpovědět, a tedy nalézt algoritmus učení. Tato kritika zapříčinila pozastavení výzkumu neuronových sítí na mnoho let. Výzkum se začal ubírat směrem k umělé inteligenci, ale zde se objevili velké problémy především v oblasti zpracování řeči a obrazu. Teprve v roce 1983 v USA vědci ze sdružení DARPA (Defence Advance Research Project Association) znovu začali investovat do výzkumu neuronových sítí a zapříčinili tak další rozvoj. Po třech letech výzkumu přišli dva vědci (David Rumelhart a Yann LeCunov) nezávisle na sobě s novým, dodnes používaným algoritmem zpětného šíření chyby (Error Back-propagation of gradient BPG). V osmdesátých letech minulého století se po celém světě objevilo mnoho vědců, kteří posunuli vývoj kupředu. Mezi nejznámější z Ameriky patří John Hopfield, Stefen Grossberg a Terrence Sejnowski, z Evropy potom Jenie Héroult a Jürgen Jutten nebo Teuvo Kohonen. V dnešní době se snažíme zejména o zrychlení algoritmů učení, snížení potřebné paměti a rozvoj různých aplikací v různých odvětvích (technice, medicíně ale i společenských vědách). [1]

2.1 Biologický neuron

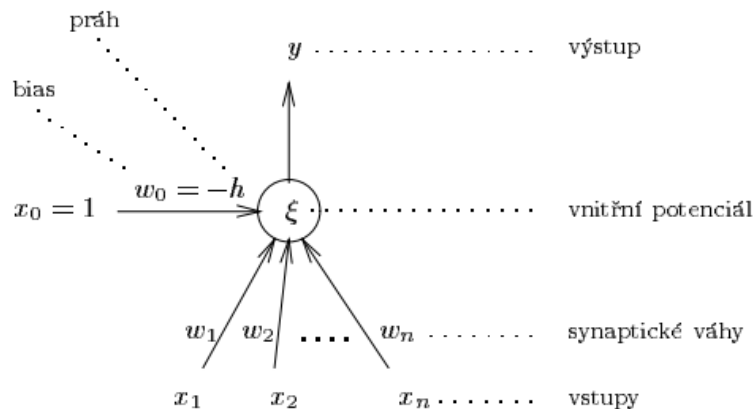
Pochopit fungování nervové soustavy člověka je velmi obtížné, proto nám postačí uvést pouze velmi zjednodušené principy, pomocí nichž dokážeme formulovat matematický model neuronové sítě. Základní stavební jednotkou nervové soustavy je nervová buňka neboli neuron. Jedná se o samostatnou specializovanou buňku, která je určena k přenosu, zpracování a uchování informací. Tyto informace jsou nezbytné pro vykonávání životních funkcí organismu. Obrázek 1 znázorňuje schématický popis biologického neuronu. [2]



Obrázek 1: Schéma biologického neuronu [2]

Neuron je složen z vlastního těla (somatu), vstupních kanálů (dendritů) a jednoho výstupního kanálu (axonu). Axon je dále propojen s dendrity dalších neuronů. Informace je přenášena pomocí speciálního rozhraní (synapse). Synapse z funkčního hlediska můžeme rozdělit na excitační (rozšiřují vzruch v nervové soustavě) a na inhibiční (utlumují). Somat i axon jsou obaleny membránou, která dokáže za jistých okolností vytvářet elektrické impulsy, pomocí nichž se může šířit informace. Elektrické impulsy jsou z axonu na dendrity přenášeny synaptickými branami, které mají různou propustnost, a tím určují intenzitu podráždění dalších neuronů. Pokud jsou tyto neurony dostatečně podrážděny (překročení tzv. prahu), tak samy vytváří další impuls a šíří tak danou informaci. Předpoklad paměťové schopnosti neuronů je to, že po každém průchodu signálu se změní synaptická propustnost. Během života organismu se mění i propojení jednotlivých neuronů. Při učení se mohou vytvářet nové paměťové stopy a při zapomínání se synaptické spoje přerušují. [2]

2.2 Perceptron



Obrázek 2: Perceptron [3]

Nejjednodušší model neuronové sítě se skládá z jediného neuronu a nazývá se Perceptron. Do tohoto neuronu se sbíhá n spojů (dendritů), které reprezentují buď podněty z vnějšího okolí, nebo výstupy jiných neuronů (axony). Po každém spoji (i -tém) přichází informace x_i ve formě reálného čísla. Toto číslo charakterizuje různé příznaky (např. teplotu, barvu, tlak, booleovskou hodnotu atd.). Každý takový spoj má navíc přiřazeno reálné číslo w_i , které udává (tzv. synaptickou) váhu spoje (např. jeho důležitost). Každý neuron má ještě tzv. práh h . Každý neuron, s výjimkou neuronů v první vrstvě, má ještě tzv. bias $w_0 = -h$ (neboli vychýlení), díky němuž neuronová síť při učení nezůstane v lokálních minimech. Bias tedy síť nepatrně rozvažuje, aby algoritmus učení divergoval až do globálního minima.

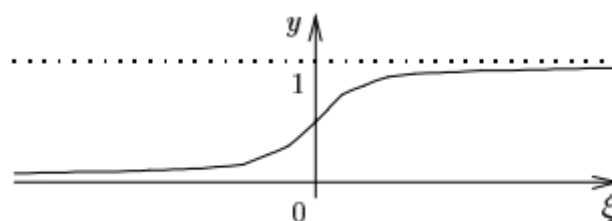
Celkový potenciál (podnět) neuronu udává součet:

$$\xi = \sum_{i=1}^n w_i * x_i - \vartheta \quad (1)$$

Kde n je celkový počet vstupů do neuronu.

Na tento potenciál reaguje neuron výstupní odezvou: $y = \sigma(\xi)$, kde σ je předepsaná nelineární přenosová funkce, obvykle ve tvaru sigmoidy. Formálně výhodná funkce má například tvar:

$$\sigma(\xi) = \frac{1}{1 + \exp(-\lambda\xi)} \quad (2)$$



Obrázek 3: Standardní (logistická) sigmoida [3]

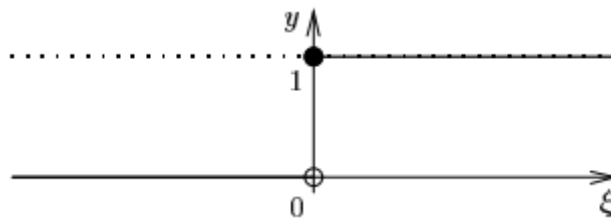
Celkem je tedy odezva neuronu dána vztahem:

$$y = \sigma \left(\sum_{i=1}^n w_i * x_i - \vartheta \right) \quad (3)$$

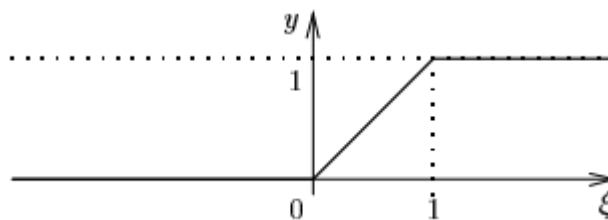
Podle toho, zda je odezva kladná nebo záporná, je neuron schopný klasifikovat vstupy do dvou tříd, což je základ pro bohatší funkce neuronových sítí. Je dobré podotknout, že jako přenosová funkce nám může posloužit jakákoli omezená rostoucí diferencovatelná funkce.

Mezi nejpoužívanější přenosové (aktivační) funkce patří kromě již zmíněné sigmoidy například ostrá nelinearita, saturovaná lineární funkce nebo hyperbolický tangens.

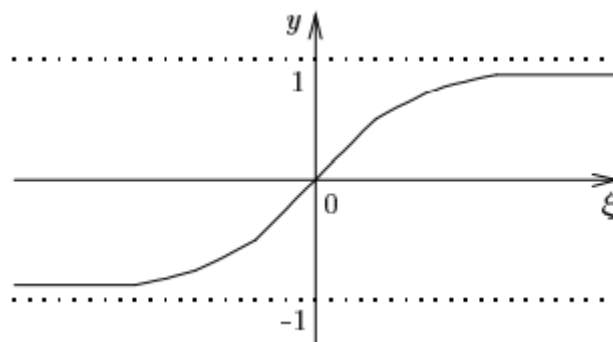
Obsah této kapitoly vychází z literatury [3].



Obrázek 4: Ostrá nelinearita [3]



Obrázek 5: Saturovaná lineární funkce [3]



Obrázek 6: Hyperbolický tangens [3]

Adaptační algoritmus perceptronu: [4]

1. Inicializace vah w_i ($i = 1, \dots, n$) a biasu w_0 malými náhodnými čísly.
Přiřazení inicializační hodnoty koeficientu učení $\alpha \in (0, 1)$.
2. Dokud není splněna podmínka ukončení výpočtu, opakovat kroky (3 až 7).
3. Pro každý tréninkový vstup (s_i) a příslušný tréninkový výstup (t_i) provádět kroky (4 až 6).
4. Aktivovat vstupní neurony ($x_i = s_i$).
5. Vypočítat skutečnou hodnotu na výstupu:

$$y_{in} = w_0 + \sum_{i=1}^n x_i w_i$$

$$y = \begin{cases} 1 & \text{pokud } y_{in} > h \\ 0 & \text{pokud } -h \leq y_{in} \leq h \\ -1 & \text{pokud } y_{in} < -h \end{cases}$$

6. Aktualizovat váhové hodnoty a bias pro daný vzor jestliže se y nerovná vzorovému výstupu ($y \neq t$):

$$w_i(\text{new}) = w_i(\text{old}) + \alpha t x_i, \quad (i = 1, \dots, n)$$

$$w_0(\text{new}) = w_0(\text{old}) + \alpha t$$

Jinak:

$$w_i(\text{new}) = w_i(\text{old})$$

$$w_0(\text{new}) = w_0(\text{old})$$

7. Podmínka ukončení:

Jestliže ve 3. kroku již nenastává žádná změna váhových hodnot:

stop

Jinak:

pokračovat krokem 4

3 Analýza současného stavu

V současné době existuje mnoho typů neuronových sítí. Každý typ má své výhody i nevýhody a každý je vhodný pro jiný typ úlohy. Všechny neuronové sítě se skládají z neuronů, které ovšem nemusí být stejné. Mohou mít různé přenosové funkce, různá spojení, různé učící algoritmy atd. Na základě těchto vlastností můžeme provést různá rozdělení neuronových sítí. [5]

3.1 Rozdělení neuronových sítí

- Podle počtu vrstev:
 - Jednovrstvé (Hopfieldova síť, Kohonenova síť...)
 - Vícevrstvé (ART síť, Perceptronová síť, klasická vícevrstvá síť s algoritmem Backpropagation)
- Podle typu algoritmu učení:
 - S učitelem (Backpropagation...)
 - Bez učitele (Hopfieldova síť...)
- Podle stylu učení na síti s učícím:
 - Deterministickým (Backpropagation)
 - Stochastickým (náhodné nastavování vah)

Neuronové sítě se v čase vyvíjí a mění. Změny můžeme pozorovat například u propojení jednotlivých neuronů, stavu neuronů nebo vah jednotlivých neuronů. Na základě těchto změn je dobré rozdělit celkovou dynamiku neuronové sítě do tří dynamik a následně tedy i práci sítě rozdělit do tří režimů: [3]

- Organizační (změna topologie)
- Aktivní (změna stavu)
- Adaptivní (změna konfigurace)

V následujícím textu budou popsány některé konkrétní typy neuronových sítí a blíže specifikována jejich topologie a vlastnosti.

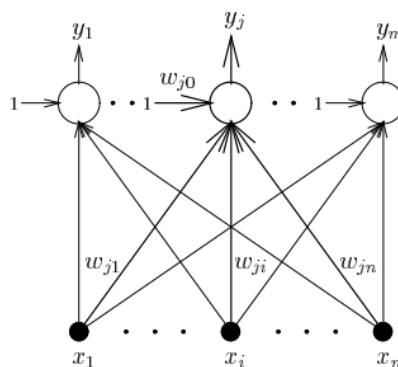
3.2 Dopředné neuronové sítě

Dopřednou neuronovou síť si lze představit jako orientovaný graf, kde jednotlivé uzly jsou spojeny ohodnocenými orientovanými (obecně oboustranně) hranami. Ohodnocení těchto hran (váhy), jsou parametrem zpracování signálu. Váhy mohou být ohodnoceny jak kladně, tak záporně, takže neurony se mohou navzájem povzbuzovat nebo potlačovat. Váha spoje určuje schopnost a intenzitu přenášení informace. Spojení mezi dvěma neurony (uzly) je orientováno, je tedy přesně dáno, ze kterého a do kterého neuronu směřují informace. Signál se v dopředných sítích šíří po jednotlivých vrstvách od vstupu směrem k výstupu. Rozeznáváme uzly vstupní, výstupní a skryté.

3.2.1 Perceptronová síť

Jedná se o jeden z prvních úspěšných modelů neuronové sítě.

Organizační dynamika specifikuje na začátku pevnou architekturu jednovrstvé sítě n - m . To znamená, že síť má n vstupních neuronů a každý z těchto neuronů je vstupem každého z m výstupních neuronů. Pro lepší pochopení topologie této sítě nám poslouží Obrázek 7.



Obrázek 7: Topologie perceptronové sítě [3]

Aktivní dynamika perceptronové sítě určuje způsob výpočtu funkce sítě. V tomto případě se reálné stavy neuronů ve vstupní vrstvě nastaví na vstup sítě a výstupní neurony spočítají svůj binární stav, který pak určuje výstup sítě. Jinak řečeno každý neuron nejprve musí spočítat svůj vnitřní potenciál jako příslušnou afinní kombinaci vstupů:

$$\xi_j = \sum_{i=0}^n w_{ji} x_i \quad j = 1, \dots, m \quad (4)$$

jejíž koeficienty $w = (w_{10}, \dots, w_{1n}, \dots, w_{m0}, \dots, w_{mn})$ tvoří konfiguraci sítě. Index i je číslo vstupního neuronu, n je celkový počet vstupních neuronů, j je číslo výstupního neuronu a m je celkový počet výstupních neuronů, takže například váha w_{ji} znázorňuje, že tato váha je na vazbě z i -tého vstupního neuronu do j -tého výstupního neuronu. Stav perceptronu se pak určí z jeho vnitřního potenciálu za použití aktivační funkce, která má tvar ostré nelinearity (Obrázek 4).

Adaptivní dynamika spočívá v naučení sítě. Síť je předložena tréninková množina τ k naučení požadované funkce:

$$\tau = \left\{ (s_k, t_k) \mid \begin{array}{l} s_k = (s_{k1}, \dots, s_{kn}) \in \mathbb{R}^n \\ t_k = (t_{k1}, \dots, t_{km}) \in \{0,1\}^m \end{array} \quad k = 1, \dots, p \right\} \quad (5)$$

kde s_k je reálný vstup k -tého tréninkového vzoru a t_k je odpovídající požadovaný binární výstup. Adaptivní dynamika má za cíl, aby síť pro každý vstup s_k ($k = 1, \dots, p$) z tréninkové množiny (s celkovým počtem vzorů p) odpovídala v aktivním režimu požadovaným výstupem t_k , tedy aby bylo v platnosti:

$$y(w, s_k) = t_k, \quad k = 1, \dots, p \quad (6)$$

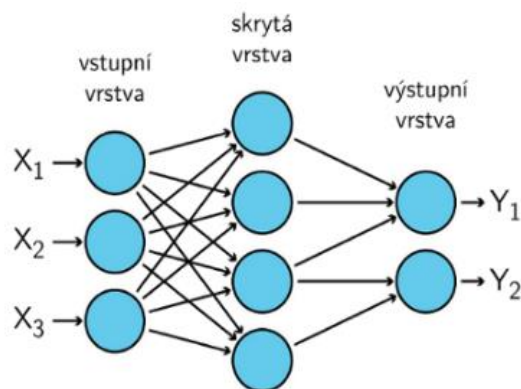
Podmínku (6) nelze splnit vždy. Jedním perceptronem nemůžeme totiž počítat každou funkci (např. funkci XOR) nebo tréninková množina nemusí být funkcí (tj. k jednomu vstupu jsou požadované dva různé výstupy). V takovém případě se snažíme, aby se síť naučila co nejvíce vzorů. Někdy nemusí být cílem sítě naučit se všechny vzory stoprocentně, protože příkladové vzory nemusí být úplně přesné.

Na začátku adaptace v čase 0 jsou váhy konfigurace $w(0)$ nastaveny náhodně blízko nuly. V každém časovém kroku učení $T = 1, 2, 3, \dots$ je síti předložen jeden vzor z tréninkové množiny a síť se ho snaží naučit, tj. adaptuje podle něj své váhy. Pořadí vzorů při učení může být různé (záleží na tréninkové strategii). Různé strategie lze například porovnat s lidským učením. Některý student si na zkoušku několikrát přečte učebnici, jiný se hned při prvním čtení vše důkladně učí a oba si na konci opakují části, které neumí. Rychlost učení sítě je také různá. Z počátku se volí nižší rychlost učení, v pozdější fázi během adaptace rychlost roste. To v analogii s přípravou studenta na zkoušku odpovídá prvnímu povrchnímu seznámení s předmětem a pozdějšímu důkladnému doučení detailů.

Význam tohoto modelu je spíše teoretický, protože perceptronová síť je schopna počítat pouze omezenou třídu funkcí (lineárně oddělitelné problémy). Tento jednoduchý model je však základem složitějších modelů jako je obecná vícevrstvá síť s učícím algoritmem backpropagation.

Obsah této kapitoly vychází z literatury [3].

Pro složitější (nelineární) rozdělení je nutné použít více vrstev. Vícevrstvé (dopředné) sítě mají jednu nebo více *skrytých* vrstev, jejichž neurony jsou nazývány skryté neurony. Obrázek 8 znázorňuje vícevrstvou síť s jednou skrytou vrstvou. Tato síť je plně propojena, protože každý neuron v jedné vrstvě je propojen s každým neuronem z další vrstvy. Pokud by některá spojení chyběla, mluvíme o částečně propojené síti. Pokud řekneme, že daná síť má n vrstev, počítáme pouze skryté vrstvy a výstupní vrstvu. Vstupní vrstva neprovádí žádný výpočet, proto ji nezapočítáváme do celkového počtu vrstev. Obrázek 8 tedy znázorňuje dvouvrstvou perceptronovou síť. [6]



Obrázek 8: Vícevrstvá perceptronová síť [7]

3.2.2 Backpropagation

Tento model neuronové sítě patří mezi nejnámější a nejpoužívanější (použití přibližně v 80% případů). Model vychází z vícevrstevných perceptronových sítí (můžeme říci, že se jedná o jistou optimalizaci). Neuronové sítě s učícím algoritmem backpropagation (zpětné šíření chyby) se učí za pomoci příkladů. To znamená, že musíme mít připravenou skupinu vstupů, u kterých známe správné (požadované) výstupy. Pomocí těchto příkladů ukážeme neuronové síti, jaké výstupy očekáváme při zadání určitých vstupů, a algoritmus backpropagation dovolí síti se adaptovat. [3] [8]

Učící proces Backpropagation probíhá v malých iterativních krocích:

Jeden z příkladů z učící skupiny je aplikován na síť a tato síť nám vrátí výstup závislý na současném nastavení synaptických vah (zpočátku bude výstup náhodný). Tento výstup je pak porovnán se správným výstupem z učící skupiny a je spočtena chyba (odchylka od správného výstupu). Tato chyba poté postupuje sítí zpět (od výstupu ke vstupu) a v každé vrstvě je provedena malá změna jednotlivých vah. Změnou vah se snažíme o minimalizaci celkové chyby výsledku. Celý tento proces opakujeme pro každý příklad z učící skupiny a můžeme ho dokonce opakovat stále dokola pro celou učící skupinu, dokud není celková chyba dostatečně malá. Poté můžeme říci, že síť je na daný problém dostatečně naučena. Síť se nikdy nedokáže naučit ideální funkci pro řešení daného problému, pouze se jí bude asymptoticky blížit. [8]

Chyba sítě $E(w)$ je vzhledem k tréninkové množině definována jako součet parciálních chyb sítě $E_k(w)$ vzhledem k jednotlivým tréninkovým vzorům ($k = 1, \dots, p$) a je závislá na konfiguraci sítě w :

$$E(w) = \sum_{k=1}^p E_k(w) \quad (7)$$

Parciální chyba $E_k(w)$ sítě vzhledem ke k -tému tréninkovému vzoru je úměrná součtu mocnin odchylek skutečných hodnot výstupů sítě pro vstup k -tého tréninkového vzoru od odpovídajících požadovaných hodnot výstupů u tohoto vzoru:

$$E_k(w) = \frac{1}{2} \sum_{j \in Y} (y_j(w, s_k) - t_{kj})^2 \quad (8)$$

kde Y je množina výstupních neuronů, y_j jsou jednotlivé výstupy sítě a t_{kj} jsou jednotlivé požadované výstupy tréninkových vzorů. [3]

Jedním z velkých problémů u backpropagation je (kromě minimalizace chybové funkce) volba vhodné topologie pro řešení konkrétního problému. Většinou přesně neznáme vztahy mezi vstupy a výstupy, které by nám pomohli při návrhu konkrétní topologie. Ve většině případů používáme vícevrstvou topologii s jednou nebo dvěma vnitřními vrstvami a očekáváme, že algoritmus backpropagation příslušné vztahy z tréninkové množiny zobecní za pomoci vah jednotlivých spojů mezi neurony. Stále je však nutné vhodně zvolit počet neuronů ve vnitřních vrstvách. Tento problém organizační dynamiky blíže souvisí s adaptací a generalizací neuronové sítě.

Topologie konkrétní vícevrstvé neuronové sítě by měla odpovídat řešenému problému, respektive jeho složitosti. Za pomoci malé sítě nemůžeme řešit komplikovaný problém, protože se může stát, že se síť v průběhu učení zastaví v nějakém mělkém lokálním minimu a je potřeba topologii doplnit o další vnitřní neurony, aby adaptace měla větší stupeň volnosti. Na druhou stranu pokud budeme mít příliš velkou síť, tak sice dokážeme nalézt globální minimum chybové funkce, ale nalezená konfigurace sítě obvykle velmi zobecňuje tréninkové vzory včetně jejich chyb a nepřesností, tudíž dává chybné výsledky pro nenaučené příklady (neboli špatně generalizuje). Pokud dojde k zapamatování tréninkové skupiny bez zobecnění obsažených zákonitostí, mluvíme o přeučení (overfitting). Naším cílem je tedy nalezení optimální topologie, která bude dostatečně bohatá pro řešení daného problému, a zároveň nebude příliš velká, aby nedošlo k nesprávnému zobecnění potřebných vztahů mezi vstupy a výstupy. V praxi většinou architekturu sítě volíme heuristicky, např. zavedeme v první vrstvě o něco více neuronů než vstupů a v druhé vrstvě bude počet neuronů roven aritmetickému průměru mezi počtem výstupů a neuronů v první vnitřní vrstvě. Po adaptaci můžeme v případě velké chyby sítě neurony přidat, nebo v případě horší generalizace několik neuronů odebrat. Celý adaptivní režim se poté opakuje pro novou architekturu. Nakonec se počítá chyba sítě vzhledem k testovací množině, což je část tréninkové množiny, která se záměrně nevyužila k adaptaci, čímž provedeme test kvality generalizace neuronové sítě. [2]

Pro nastínění implementace backpropagation nám poslouží následující algoritmus: [4]

1. Váhové hodnoty a bias jsou inicializovány malými náhodnými čísly.
Přiřazení inicializační hodnoty koeficientu učení α .
2. Dokud není splněna podmínka ukončení výpočtu, opakovat kroky (3 až 10).
3. Pro každý tréninkový pár provádět kroky (4 až 9)

Feedforward:

4. Aktivovat vstupní neurony z množiny X .
5. Vypočítat vstupní hodnoty vnitřních neuronů z množiny Z : $z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$.

Stanovení výstupních hodnot vnitřních neuronů $z_j = f(z_{in_j})$, kde f je aktivační funkce.

6. Stanovení skutečných výstupních hodnot signálu neuronové sítě:

$$y_{in_k} = w_{0k} + \sum_{j=1}^m z_j w_{jk},$$
$$y_k = f(y_{in_k}).$$

Backpropagation:

7. Ke každému neuronu z výstupní vrstvy Y je přiřazena hodnota očekávaného výstupu pro vstupní tréninkový vzor. Dále jsou vypočteny částečné váhové korekce pro w_{jk} příslušející chybě na spojeních vedoucích k neuronu Y_k ve výstupní vrstvě podle vzorce $\delta_k = (t_k - y_k)f'(y_{in_k})$. δ_k je součástí váhové korekce $\Delta w_{jk} = \alpha \delta_k z_j$ i korekce biasu $\Delta w_{0k} = \alpha \delta_k$.
8. Ke každému neuronu ve vnitřní vrstvě Z je přiřazena sumace jeho delta vstupů (tj. z neuronů, které se nacházejí v následující vrstvě), $\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$. Vynásobením získaných hodnot derivací jejich aktivační funkce obdržíme $\delta_j = \delta_{in_j} f'(z_{in_j})$, což jsou částečné váhové korekce pro v_{ij} příslušející chybě na spojeních vedoucích k neuronu Z_j ve skryté vrstvě. δ_j je součástí váhové korekce $\Delta v_{ij} = \alpha \delta_j x_i$ i korekce biasu $\Delta v_{0j} = \alpha \delta_j$.

Aktualizace vah a prahů:

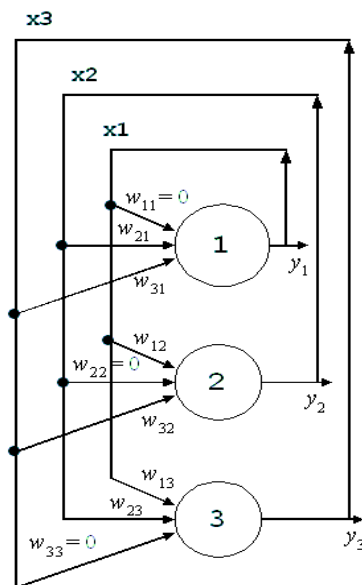
9. Každý neuron ve výstupní vrstvě Y aktualizuje na svých spojeních váhové hodnoty včetně svého biasu: $w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$. Každý neuron ve vnitřní vrstvě Z aktualizuje na svých spojeních váhové hodnoty včetně svého biasu: $v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$.
10. Podmínka ukončení:
Pokud již nenastávají žádné změny váhových hodnot nebo pokud již bylo vykonáno maximálně definované množství váhových změn, stop; jinak, pokračovat.

3.3 Rekurentní neuronové sítě

V následující kapitole je ukázáno, jakým způsobem lze implementovat do vícevrstvé neuronové sítě časový kontext. Jinak řečeno, odezva sítě nebude dána pouze aktuálním vstupním stimulem, ale bude odrážet i vliv stimulů, které tento aktuální předcházely. Realizovat tento princip umožňuje rekurentní neuronová síť. Oproti předcházejícím modelům se v tomto případě signál nešíří pouze od vstupní vrstvy směrem k vrstvě výstupní, ale dochází i ke zpětnému přenosu informace od vrstev vyšších zpět do vrstev nižších. Tato zpětná vazba je řešena prostřednictvím tzv. rekurentních neuronů. [9]

3.3.1 Hopfieldova síť

Jedná se o rekurentní (zpětnovazebnou) síť, která se skládá pouze z jedné vrstvy. V této vrstvě jsou neurony plně propojeny. To znamená, že každý výstup neuronu je propojen se vstupy všech ostatních neuronů (ne však se vstupem sebe samotného). Jedná se tedy o *symetrickou síť*.



Obrázek 9: Topologie Hopfieldovy sítě [10]

Adaptivní fáze Hopfieldovy sítě se řídí *Hebbovým zákonem*. Algoritmus Hebbova zákona pro asociativní neuronové sítě je následující: [4]

1. Inicializace všech váhových hodnot $w_{ij} = 0$.
2. Pro každý testovací vzor (tréninkový pár) opakovat kroky 3 až 5.
3. Inicializovat vrstvu X pomocí vzorových vstupů $x_i = s_i$ ($i = 1, \dots, n$).
4. Inicializovat vrstvu Y pomocí vzorových výstupů $y_j = t_j$ ($j = 1, \dots, m$).
5. Nastavit váhové hodnoty $w_{ij}(\text{new}) = w_{ij}(\text{old}) + x_i y_j$.

Na rozdíl od perceptronových modelů probíhá adaptivní fáze pouze jednou. Opět zde máme učící skupinu vzorů, kde každý ze vzorů je „učení“ pouze jednou a následně jsou stanoveny výsledné váhy sítě podle pravidla:

$$w_{ji} = \sum_{k=1}^p s_{kj}s_{ki} \quad 1 \leq j \neq i \leq n \quad (9)$$

Kde p je počet vzorů a w_{ji} značí váhu mezi neurony i a j . s_{kj} a s_{ki} i -tý a j -tý neuron v k -tém vzoru.

V aktivní fázi probíhá výpočet po jednotlivých neuronech. Pro každý z neuronů je vypočten vnitřní potenciál podle:

$$\xi_j^{(t-1)} = \sum_{i=1}^n w_{ji}y_i^{(t-1)} \quad (10)$$

Kde y_i značí výstup i -tého neuronu.

Následně je pomocí $\xi_j^{(t-1)}$ určen nový výstup y_j :

$$y_j^{(t)} = \begin{cases} 1, & \xi_j^{(t-1)} > 0 \\ y_j^{(t-1)}, & \xi_j^{(t-1)} = 0 \\ -1, & \xi_j^{(t-1)} < 0 \end{cases} \quad (11)$$

Odtud je vidět, že výsledné stavy jsou blíže k biologické předloze umělých neuronů, protože jsou bipolární (pouze hodnoty 1 a -1). Takto je výpočet aktivní fáze opakován, dokud se síť neustálí (stavy neuronů se přestanou měnit).

V této kapitole bylo čerpáno ze zdroje [11].

3.3.2 Elmanova síť

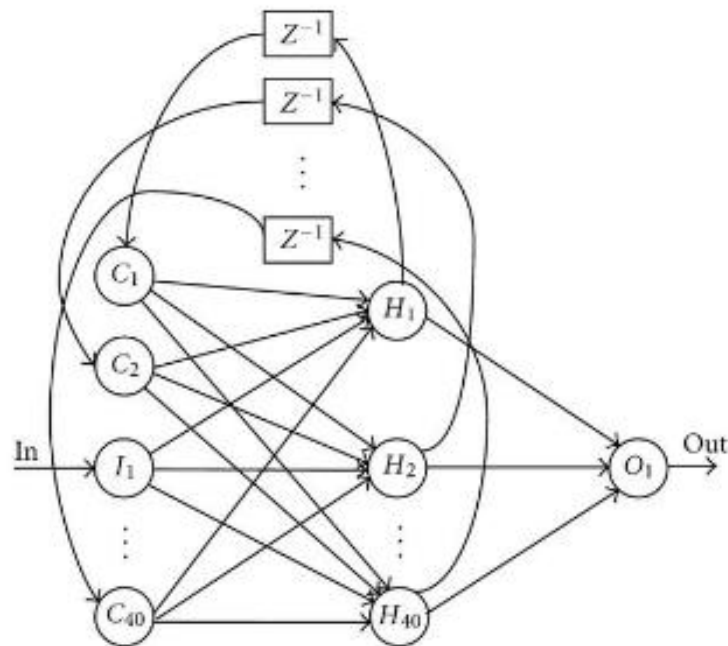
Jedná se o dvouvrstvou síť s učícím algoritmem Backpropagation a se zpětnou vazbou ze skryté vrstvy na vrstvu vstupní. Výstup ze zpětné vrstvy je možné v každém časovém kroku využít jako vstup do kontextové vrstvy, která je dále využita jako vstup do celé neuronové sítě. Jinak řečeno vstup v čase t může ovlivnit vstup v čase $t+1$. Zde je několik nejvyužívanějších algoritmů učení Elmanovy sítě:

Učící algoritmus zpětného šíření chyby v čase – využívá se pro výpočet derivace chyby pro rekurentní síť. Jde vlastně o modifikaci algoritmu Backpropagation, kde se nejdříve uskuteční rozvinutí rekurentní sítě v čase do sítě s dopředným šířením a poté se využívá klasický algoritmus zpětného šíření chyby.

Učící algoritmus učení v reálném čase – je sice výpočtově náročnější než algoritmus zpětného šíření chyby v čase ale nepotřebuje přesně definovaný časový interval určující rozvinutí sítě.

Učící algoritmy na základě Kalmanova filtru – výpočtově velmi náročný algoritmus (proto se objevují různé modifikace), který patří mezi nejvyužívanější. Kalmanův filtr můžeme využít pro odhad skrytého stavu na základě pozorování.

Jako zdroj pro popis Elmanovy sítě byla využita literatura [12].



Obrázek 10: Topologie Elmanovy sítě [13]

3.5 Samoorganizační neuronové sítě

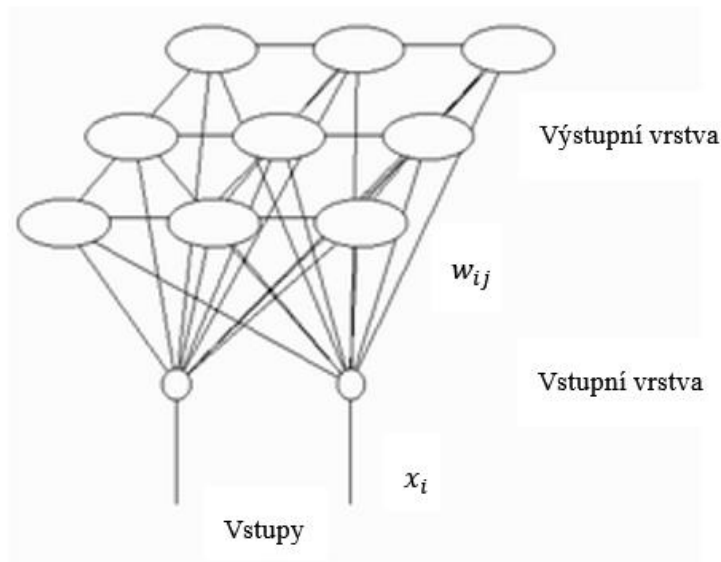
Následující kapitola bude popisovat modely neuronových sítí, které využívají soutěžní strategie učení. Tyto modely mají společný princip, že výstupní neurony sítě soutěží mezi sebou o to, který z nich bude aktivní. Oproti jiným učícím principům (např. Hebbovo učení) je v určitém časovém okamžiku aktivní pouze jeden neuron. [2]

3.5.1 Kohonenova síť

Jednovrstvá samoorganizující se síť, pracující na principu shlukování neuronů. Snaží se o napodobení fungování lidského mozku, který si uchovává informace pomocí vnitřní prostorové reprezentace dat. V Kohonenově síti jsou propojeny pouze nejbližší sousední neurony (na rozdíl od Hopfieldovy sítě, kde jsou neurony propojeny každý s každým). Neuron nemá ani přenosovou funkci, pouze počítá vzdálenosti r mezi vzorem zakódovaným ve vahách a vzorem vstupním podle vzorce: [12]

$$r_j = \sum_{i=1}^n [x_i(t) - w_{ij}(t)]^2 \quad (12)$$

Aby síť mohla co možná nejlépe klasifikovat zadaný problém, řadí neurony do jednotlivých oblastí. Při každém kroku se také mění váhy (adaptují se). Neurony mezi sebou soutěží (kompetice) o to, který je nejbližším vzorem ke vstupnímu a následně všechny okolní neurony upraví své váhy a vektory dle vítěze. Dochází tak k optimalizaci sítě. [14]



Obrázek 11: Topologie Kohonenovy sítě [15]

Kohonenův algoritmus: [4]

1. Inicializace všech váhových hodnot w_{ij} .

Inicializace poloměru sousedství, tj. okolí R .

Inicializace parametru učení α .

2. Pokud není splněna podmínka ukončení, provádět kroky (3 až 9).
3. Pro každý vstup x_1 až x_n opakovat kroky (4 až 6).
4. Pro každý výstupní neuron j ($j = 1, \dots, m$) vypočítat:

$$r(j) = \sum_{i=1}^n (w_{ij} - x_i)^2$$

Kde m je počet výstupních neuronů, i je vstupní neuron a n je počet vstupních neuronů.

5. Najít index J takový, že $r(J)$ je minimum.
6. Aktualizace váhových hodnot všech neuronů ($j \in J$) tvořících topologické sousedství charakterizované indexem J , tj. pro všechna i ($i = 1, \dots, n$) platí:

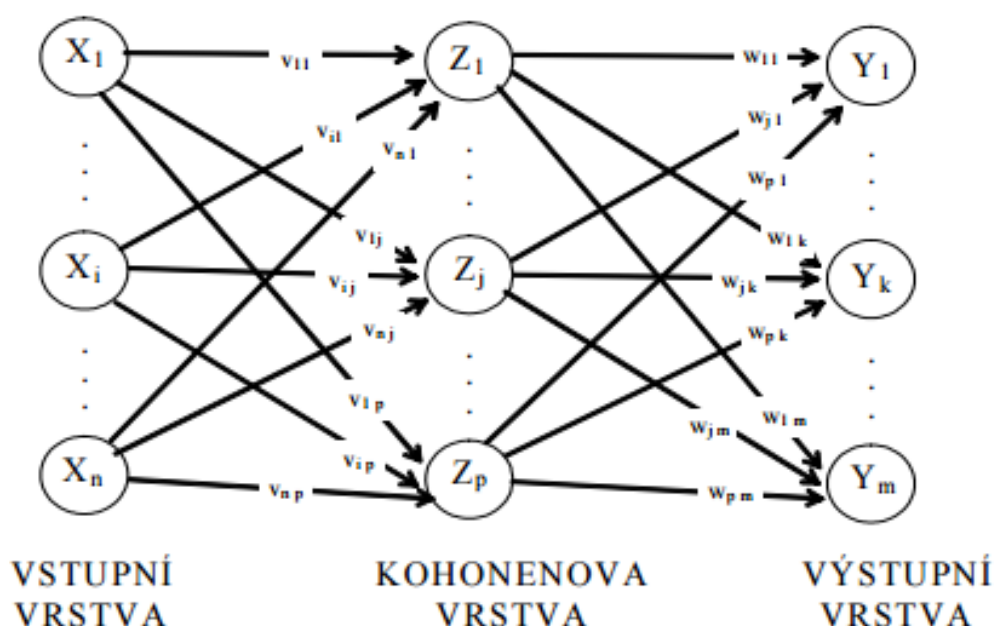
$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha[x_i - w_{ij}(\text{old})]$$

7. Aktualizace parametru učení.
8. Zmenšení poloměru R topologického sousedství.
9. Test podmínky ukončení.

3.5.2 Counterpropagation

Jedná se o model umělé neuronové sítě, která se snaží využít samoorganizační síť v kombinaci s dalším přídavným mechanismem k řešení problémů učení s učitelem. Následující popisovaná síť je jen jedna z možných variant této neuronové sítě. Síť counterpropagation hledá k danému vstupu nejbližšího reprezentanta a odpovídá výstupní hodnotou, která je spojena s tímto reprezentantem, tedy pracuje jako vyhledávací tabulka (lookup table).

Síť se skládá z neuronů rozdělených do třech vrstev. První vstupní vrstva pouze posílá vstupní signál do další vrstvy. Druhá vrstva se skládá z p samoorganizačních jednotek, neboli Kohonenových samoorganizačních map. Tyto jednotky jsou vzájemně propojeny. Poslední třetí vrstva se skládá z Grossbergových jednotek instar a výstupy těchto jednotek tvoří výstupy celé neuronové sítě. Schéma této sítě je znázorněno na



Obrázek 12 : Dopředná síť counterpropagation [2]

Adaptivní dynamika u této sítě se rozděluje do dvou fází. Pomocí Kohonenova učení bez učitele se nejprve nastaví váhy v_{ij} ($i = 1, \dots, n; j = 1, \dots, p$) samoorganizačních jednotek mezi vstupní a vnitřní vrstvou. V této fázi se používají jen vstupní části tréninkových vzorů. Na konci první fáze se váhy v zafixují a přechází se ke druhé fázi učení, ve které se nastaví váhy w_{jk} ($j = 1, \dots, p; k = 1, \dots, m$) mezi vnitřní a výstupní vrstvou. Následně je pro všechny neurony ve výstupní vrstvě spočten aktuální výstup, který je porovnán s požadovaným výstupem sítě. Pokud výstup sítě neodpovídá požadovanému výstupu, jsou váhy w modifikovány pomocí Grossbergova adaptačního pravidla.

Při popisu modelu Counterpropagation bylo čerpáno ze zdroje [2].

Algoritmus sítě counterpropagation: [4]

1. Inicializovat všechny váhové hodnoty a parametry učení.
2. Pokud není splněna podmínka ukončení 1. fáze adaptace, provádět kroky 3 až 8.
3. Pro každý vstupní vektor $x = (x_1, \dots, x_n)$ opakovat kroky 4 až 6.
4. Aktivovat vstupní vrstvu vektorem x .
5. Najít vítěze kompetice ve vnitřní vrstvě a označit jeho index J .
6. Aktualizace váhových hodnot na spojeních vedoucích k neuronu Z_J , neboli pro všechna i ($i = 1, \dots, n$) platí: $v_{ij}(\text{new}) = (1 - \alpha)v_{ij}(\text{old}) + \alpha x_i$.
7. Snížit hodnotu parametru učení α .
8. Test podmínky ukončení 1. fáze.
9. Pokud není splněna podmínka ukončení 2. fáze adaptace, provádět kroky 10 – 16. (α má během celé 2. fáze velmi malou konstantní hodnotu.)
10. Pro každý tréninkový vstupní pár vektorů $x: y$, ($x = (x_1, \dots, x_n), y = (y_1, \dots, y_m)$), opakovat kroky 11 až 14.
11. Aktivovat vstupní vrstvu vektorem x ; Aktivovat výstupní vrstvu vektorem y .
12. Najít vítěze kompetice ve vnitřní vrstvě a označit jeho index J .
13. Aktualizace váhových hodnot na spojeních ze vstupní vrstvy do neuronu Z_J (α je velmi malé), tj. pro všechna i ($i = 1, \dots, n$) platí: $v_{ij}(\text{new}) = (1 - \alpha)v_{ij}(\text{old}) + \alpha x_i$.
14. Aktualizace váhových hodnot na spojeních vedoucích z neuronu Z_J do výstupní vrstvy, tj. pro všechna k ($k = 1, \dots, m$) platí: $w_{jk}(\text{new}) = (1 - \alpha)w_{jk}(\text{old}) + \alpha y_k$.
15. Snížit hodnotu parametru učení α .
16. Test podmínky ukončení 2. fáze.

4 Použití umělých neuronových sítí

Nyní se neuronové sítě používají v mnoha odvětvích, ale v oblastech, kde by byli zodpovědné za významnější finanční prostředky nebo dokonce za lidský život ještě nejsou naplno využívány. Důvodem je jejich spolehlivost. Umělé neuronové sítě, přesto že pracují správně, můžou být nepřesné. Samotná myšlenka neomylnosti počítačů také není zcela správná. V reálu mohou počítače dělat chyby stejně jako lidé. Ať už se jedná o chyby z důvodu technických problémů nebo chyb v programech, či z nedostatku profesionality. To znamená, že při řešení závažnějších úloh, by neuronové sítě neměly sloužit jako jediný prostředek při varování na zvláštní situace. Je nezbytné, aby pro zvláště kritické úkoly existovalo více systémů, které by pracovaly paralelně, a pojistili sebe navzájem.

Dalším úskalím použití neuronových sítí je, že u tradičních neuronových sítí není možné vysvětlit, jakým způsobem je řešen určitý problém. Resp. je sice možné fungování vysvětlit, ale většinou je vnitřní reprezentace výsledků učení tak složitá, že není možné ji analyzovat (až na několik jednodušších případů, které se většinou nepoužívají).

Před nedávnem se začali objevovat aktivní pokusy, které kombinují umělé neuronové sítě s expertními systémy. Při tomto použití umělá neuronová síť reaguje na většinu jednodušších případů a zbytek zasílá expertnímu systému.

Existují také různé softwarové balíčky neuronových sítí vyvinuté různými společnostmi, které dovolují uživatelům používat různé typy neuronových sítí s různými způsoby jejich učení. Jedná se jak o speciální balíčky (např. na určování budoucích cen akcií), tak o univerzální použitelné všestranně.

Umělé neuronové sítě je tedy možné využít pro širokou škálu problémů. V následujícím textu bude popsáno několik případů, ve kterých je použití neuronových sítí obzvláště zajímavé. Některé příklady se neřadí mezi novinky, nicméně jejich popis slouží k lepšímu pochopení aplikace neuronových sítí.

4.1 Technika a řízení

4.1.1 Couvání s návěsem

Potíže vznikající při couvání s návěsem jsou známy každému, kdo se někdy pokoušel, nebo sledoval začátečníka provádět tento manévru. Nicméně, řidič se zkušenostmi provede tento kousek s pozoruhodnou lehkostí. Jako příklad aplikace neuronových sítí na problém s řízením si můžeme představit neuronovou síť, která je schopna určit jak zacouvat k nakládací rampě s kamionem s připojeným návěsem. Jako informace nám postačí popis pozice kabiny kamionu, pozice zadní části návěsu, pozice (fixní) nakládací rampy a úhly mezi kamionem, návěsem a nakládací rampou. Neuronová síť je schopna se naučit, jak zatočit s kamionem, aby návěs dosáhl rampy ze všech možných počátečních konfigurací, ze kterých je řešení reálné. Aby byl problém ještě náročnější, kamion je schopen pouze couvat.

Neuronová síť používá k řešení tohoto problému dva moduly. První (emulátor) se naučí počítat novou pozici kamionu, vzhledem k jeho stávající pozici a úhlu natočení volantu. Kamion se v každém časovém okamžiku pohne o fixní vzdálenost. Tento modul se učí „cit“ pro to, jak kamion s návěsem reaguje na různé řídicí signály, v podstatě stejným způsobem, jakým se řidič učí chování takové soupravy s návěsem. Emulátor má několik skrytých jednotek a je trénován pomocí backpropagation.

Druhý modul je ovladač. Poté co je emulátor naučený, ovladač se učí správnou sérii řídicích signálů, kterou musí dát kamionu, aby zadní dveře návěsu byly zároveň s rampou. V každém časovém kroku, ovladač vyšle řídicí signál a emulátor určí novou pozici kamionu a návěsu. Tento proces pokračuje, dokud návěs nedosáhne rampy, nebo se souprava nezkrříží. Poté je určena chyba a upraví se váhy na ovladači.

Jako u řidiče se výkon zlepšuje s praxí a ovladač se učí poskytnout sérii řídicích signálů, které nasměrují kamion a návěs k rampě, bez ohledu na počáteční pozici (pokud je řešení reálné). Na začátku tedy může kamion směřovat k rampě, může směřovat od rampy nebo může svírat s rampou jakýkoli úhel. Podobně úhel mezi kamionem a návěsem může mít počáteční hodnotu blížící se situaci zkrřížení. Proces tréninku pro ovladač funguje na principu rekurentní backpropagation.

4.1.2 Samopilotované letadlo

Roku 1996 společnost Accurate Automation Corp z USA na základě objednávky NASA a Air Force vyvinula experimentální samopilotované hypersonické špionážní letadlo LoFLYTE (Low-Observable Flight Test Experiment). Toto letadlo, navržené pro výzkum nových zásad pilotování, bylo dlouhé 2,5 metru a vážilo pouhých 32 kilogramů. LoFLYTE využíval umělé neuronové sítě, díky kterým se mohl autopilot učit kopírovat techniku pilotáže od skutečného letce.

Letadlo dosahovalo vysokých rychlostí (4 až 5 Machu), tudíž rychlost reakce pilota člověka nebyla vždy dostatečná. Pokud se tak stalo, přišla na řadu neuronová síť, která díky získaným zkušenostem pilota a velmi rychlému zpracování informací umožňovala rychle najít řešení kritických situací.



Obrázek 13: LoFLYTE [16]

Na stejném principu jako LoFLYTE pracují i modernější experimentální robotická letadla, jako například Boeing X-51 a NASA X-43, které jsou ještě rychlejší a mají dokonalejší systém řízení.



Obrázek 14: Boeing X-51 [17] a NASA X-43 [18]

4.2 Zpracování signálu

V oblasti zpracování signálu se vyskytuje mnoho aplikací neuronových sítí. Jedno z prvních komerčních použití bylo (a stále je) potlačení šumu při telefonování. Neuronová síť použita k tomuto účelu je forma ADALINE. Potřeba adaptivního potlačování ozvěny se stala zásadní s vývojem mezikontinentálních satelitních linek pro dálkové telefonní okruhy, kde je zpoždění při přenosu kolem půl vteřiny. Dokonce i v případě drátové telefonní linky způsobuje opakovací zesilovače ozvěnu. Myšlenka potlačení šumu je celkem jednoduchá. Na konci dlouhé linky je příchozí signál aplikován do telefonního systémového komponentu (hybridu) a do přidavného filtru (neuronová síť). Rozdíl mezi výstupem hybridu a výstupem neuronové sítě je chyba, která je použita k nastavení vah v neuronové síti. Neuronová síť je trénována k odstranění zvuku (ozvěny) z výstupního signálu hybridu.

4.3 Informační technologie

4.3.1 Rozpoznávání vzorů

Do oblasti rozpoznávání spadá mnoho zajímavých problémů. Jedna ze specifických oblastí, ve které bylo vyvinuto mnoho aplikací neuronových sítí, je automatické rozpoznávání ručně psaných znaků (číslic nebo písmen). Velké množství variací velikosti, pozice a stylů psaní dělají tento problém velmi obtížný pro tradiční techniky. Jedná se o dobrý příklad použití neuronových sítí, nicméně se jedná o typ zpracování informací, který lidé provádí relativně snadno. Vícevrstvé neuronové sítě (například backpropagation) se používají k rozpoznávání ručně psaných poštovních směrovacích čísel. Přesto, že je aplikace založena na standardním učícím algoritmu, je zcela běžné upravovat architekturu pro dosažení lepších výsledků. [4]

4.3.2 Produkce řeči

Učení se číst Anglický text nahlas je složitá otázka, protože správná fonetická výslovnost písmen záleží na kontextu, ve kterém se dané písmeno nachází. Tradiční přístup k tomuto problému by znamenal vytvoření sady pravidel pro standardní výslovnost různých skupin písmen, společně s vyhledávací tabulkou pro výjimky.

Jeden z nejznámějších příkladů použití neuronových sítí v problému produkce řeči je NETtalk (vícevrstvá neuronová síť). Na rozdíl od potřeby vytvoření pravidel a vyhledávací tabulky pro výjimky, jediné co je potřeba u NETtalk je set příkladů s psaným vstupem společně se správnou výslovností. Psaný vstup obsahuje jednak písmeno, které je právě vyslovováno a dále tři písmena před a za vyslovovaným písmenem (kvůli kontextu). Další symboly jsou použity k indikaci konce slova nebo interpunkci. Síť je trénována použitím 1000 nejběžnějších Anglických slov. Po naučení je síť schopna číst nová slova pouze s několika chybami. Tyto chyby jsou jen drobné chyby ve výslovnosti a srozumitelnost mluvy je celkem dobrá.

Je zajímavé, že se zde nachází několik dosti odlišných etap odpovídání sítě, jak učení sítě postupuje. Síť se celkem rychle naučí rozeznávat samohlásky od souhlásek. Nicméně v první fázi učení používá stejnou samohlásku pro všechny samohlásky a stejnou souhlásku pro všechny souhlásky. Výsledkem je nesrozumitelné „žvatláni“. Druhá fáze učení spočívá v naučení sítě rozpoznávat hranice mezi jednotlivými slovy. To produkuje pseudo-slovní typ odezvy. Po nejméně deseti průchodech přes tréninková data se text stane srozumitelným. To znamená, že učící proces odezvy sítě je podobný jako vývoj řeči u malých dětí.

4.3.3 Rozpoznávání řeči

Zaznamenáváme pokrok ve složité oblasti rozpoznávání řeči bez závislosti na mluvčího. Mnoho užitečných systémů má nyní omezenou slovní zásobu nebo gramatiku, nebo potřebuje přeučení při změně mluvčího. Několik typů neuronových sítí je používáno pro rozpoznávání řeči, včetně vícevrstevných sítí (včetně rekurentních).

Jedna síť je předmětem zvláštního zájmu, jednak kvůli její úrovni vývoje směrem pro použití v praxi a také pro její konstrukci. Jedná se o Kohonenovy samoorganizující se mapy. Výstupem této sítě je dvourozměrné pole. Vstup do sítě je založen na krátkých segmentech (několik milisekund dlouhých). Síť shromažďuje podobné vstupy, klastry, které jsou vytvořeny tak, že jsou umístěny různé příklady stejného fonému vyskytující se na výstupních jednotkách, které jsou blízko u sebe ve výstupním poli.

Poté co jsou vstupní signály zmapovány na fonetické „regiony“ (což je provedeno bez toho, aby bylo sítí sděleno, co to je foném), výstupní jednotky mohou být propojeny s odpovídající klávesou pro zkonstruování fonetického psacího stroje. Protože korespondence mezi fonémy a psanými písmeny je velmi běžná ve Finštině (pro kterou byla síť vynalezena), hláskování je většinou správné.

Dalším příkladem je systém pro hlasové ovládání vestavěné kalkulačky ve Windows od ruské firmy NejroProjekt. Tento systém byl schopen bez předchozího tréninku rozpoznat 36 slov řečených libovolným člověkem. Při klasifikaci byla využívána hierarchická neuronová síť, která měla dvě kaskády. První měla na starost hrubé rozřazení slov do jedné z šesti tříd a druhá kaskáda přesně určila slovo uvnitř jednotlivých tříd. Na učení této neuronové sítě se podílelo 19 hlasatelů.

4.3.4 Identifikace témat textových zpráv

Mezi další příklad úspěšného použití umělých neuronových sítí se řadí zpravodajský server Convectis. Server byl roku 1997 vybrán společností PointCast, která byla vůdcem doručením personalizovaných zpráv z internetu, pro automatické rozřazení textových zpráv podle kategorie do jednotlivých rubrik. Convectis byl, díky určení klíčových slov podle kontextu, schopný rozpoznat v reálném čase a automaticky rozřadit obrovské proudy textových zpráv posílaných přes datové sítě (Reuters, NBC a CBS).

4.4 Ekonomika a finance

Neuronové sítě jsou používány v mnoha obchodních problémech. Zde bude zmíněno pouze několik z mnoha příkladů.

4.4.1 Hypotéka

Ačkoliv se může zdát, že pravidla, které formují základ pro tvorbu hypotéky, jsou dobře známá, je složité specifikovat celý proces, ve kterém experti rozhodují v okrajových případech. Kromě toho, je zde velká finanční odměna i za malou redukci počtu hypoték, které se stanou delikventní. Základní myšlenka použití neuronových sítí při posouzení rizik hypotéky je použití dosavadních zkušeností k vytrénování sítě, která zajistí více konzistentní a spolehlivé vyhodnocení hypotečních žádostí.

Za použití dat od několika zkušených hypotečních hodnotitelů byli neuronové sítě vytrénovány k zobrazení uchazečů o hypotéku, kteří chtějí pouze hypotéku, nebo kteří chtějí hypoteční pojištění. Účelem toho všeho bylo rozhodnout, kterým žadatelům by měla být dána půjčka a kterým nikoliv. Rozhodnutí u druhého typu uchazečů je více náročné, protože jen ti žadatelé, kteří byli vyhodnoceni jako vyšší riziko, jsou vyhodnoceni pro hypoteční pojištění. Tréninkový vstup zahrnuje informace o žadatelovo zaměstnání, počet rodinných příslušníků, současný příjem atd., stejně jako informace o samotné hypotéce, jako třeba úrok, charakteristika nemovitosti (např. odhadní hodnotu). Cílovým výstupem ze sítě je potom odezva „schváleno“ nebo „zamítnuto“.

U obou typů žádostí neuronová síť dosáhla vysokého počtu stejných výsledků jako lidští experti. Pokud došlo k odchylce, byl to většinou krajní případ, kde by i hypoteční expert mohl mít stejný názor jako neuronová síť. Při nezávislém srovnání kvality potvrzených hypoték si neuronová síť vedla celkově lépe než experti. Ve výsledku se síť naučila formulovat schválení na základě zkušeností všech expertů, jejichž akce byli formovány v základu pro trénink.

Druhá neuronová síť byla trénována ke zhodnocení rizika selhání při půjčce, založena na datech obsahujících 111 080 aplikací, z nichž u 109 072 nebyla známa historie splácení. Bylo vybráno celkem 4 000 tréninkových příkladů z databáze. Přestože neplacení může vyplývat z mnoha příčin, které nejsou obsaženy v dostupných informacích při žádosti o úvěr, předpověď sítě byla schopna zredukovat neplatičství o 12 procent.

Vznikl také program KnowledgeSeeker od firmy Angoss, s jehož pomocí můžou odborníci v bance zjistit, který z jejich žadatelů o hypotéku bude pravděpodobně zdržovat platby. Zpočátku se předpokládalo, že na prvním místě budou ti, kteří už dříve zdržovali platby. Studie ale ukázaly, že v budoucnu budou problémy spíše s platbami u dlužníků, kteří někdy údajně zapomněli splátku zaplatit. Jak se ukázalo, jejich „zapomětlivost“ souvisela s vážnými finančními problémy.

4.4.2 Správa financí

Neuronové sítě jsou značně používány na finančních trzích. Například americká Citibank začala využívat neuronové sítě v roce 1990 a v následujících dvou letech od zavedení, podle časopisu The Economist, dosáhlo automatické obchodování výnosu 25% za rok. Další, kdo využívá neuronové sítě je Chemical Bank, která je využívá k předzpracování transakcí v zahraničních devizových trzích a sleduje podezřelé transakce. Také Deere & Co LBS Capital využívá cca 900 neuronových sítí v kombinaci s expertními systémy.

Již v roce 1992 společnost HNC vydala software nazývaný Falcon, díky kterému bylo možné detekovat a zabránit podezřelým operacím odcizených kreditních a debetních karet (v reálném čase). Umělé neuronové sítě se postupně učily obvyklé chování zákazníků a byli následně schopny detekovat náhlé změny při nakupování, což signalizuje možnou krádež karty. Tyto krádeže činily velkým bankám ročně ztráty desítky milionů dolarů, ale po zavedení Falconu v roce 1994 se poprvé od zavedení karet tyto ztráty snížili. Podobný systém byl také vyvinut firmou ITC a sloužil ke sledování transakcí pomocí kreditních karet Visa.

4.5 Reklama a marketing

Krásným příkladem použití neuronových sítí je takzvaná „chytrá“ reklama. Neuronová síť SelectCast od společnosti Aptex Software dokázala určit oblasti zájmu uživatelů internetu a nabídnout jim následně reklamu na podobná témata. Roku 1997 společnost Excite licencovala tuto síť pro použití ve svých vyhledávačích. Po zavedení sítě na servery byla pokryta „chytrou“ reklamou asi třetina uživatelů. Následné studie prokázaly, že odezva na tuto „chytrou“ reklamu byla v průměru dvakrát vyšší než na klasickou reklamu a v některých případech se účinnost zvýšila až pětkrát.

Podobně využila neuronové sítě i společnost Neural Innovation při spolupráci s marketingovými společnostmi. Jednalo se o strategii direct mail. Nejdříve bylo rozesláno pouze 25% ze všech návrhů a shromážděné informace z odpovědí a reakcí spotřebitelů byli předány neuronové síti, která měla za úkol najít nejlepší segment spotřebitelského trhu pro každý produkt. Následně bylo rozesláno zbylých 75% návrhů s ohledem na „doporučení“ neuronové sítě. Účinnost druhé etapy rozesílání se výrazně zvýšila v porovnání s původním stavem.

Zpětná vazba od zákazníků je velmi důležitá při podnikání v konkurenčním prostředí. Většinou tedy společnosti provádí různé průzkumy mezi zákazníky, což umožňuje zjistit, jaké faktory jsou rozhodující při nákupu určitého zboží nebo služby. Následná analýza výsledků ovšem není snadný úkol, protože je zde velké množství souvisejících parametrů a je nutné identifikovat nejdůležitější faktory. Pro tento případ mohou také posloužit neuronové sítě, které jsou schopny zjistit a předvídat chování spotřebitelů při změně marketingové politiky, tedy najít optimální strategii firmy.

4.6 Zdravotnictví

Svého času byl ve Spojených státech zaveden systém na odhalování podvodů ve zdravotnictví. Rozpočtové ztráty z tohoto druhu podvodu byli ročně cca 730 milionů dolarů. Firma ITC tedy začala vytvářet specializovaný neuronový systém. Vývoj trval více než rok a stál 2,5 milionu dolarů. Testování tohoto nového systému ovšem ukázalo, že neuronová síť je schopna rozpoznat 38% případů podvodů. Dosavadně používaný expertní systém byl schopen rozpoznat pouze 14%.

V lékařské diagnostice jsou neuronové sítě nejčastěji používány spolu s expertními systémy. Již dříve zmiňovaná firma NejroProjekt vytvořila systém objektivní diagnostiky sluchu u dětí. Při klasické diagnostice jsou registrované reakce mozku na sluchový stimul (pomocí elektroencefalogramu). K diagnostice sluchu dítěte zkušeným odborníkem je třeba cca 2000 testů. Neuronová síť je schopna provést diagnostiku sluchu dítěte se stejnou spolehlivostí po pouhých 200 kontrolách během několika minut a navíc bez účasti odborníka.

Jedním z mnoha dalších příkladů použití neuronových sítí v medicíně byl také „instantní lékař“ vynalezený Andersonem v polovině osmdesátých let. Myšlenka této aplikace spočívala ve vytrénování aplikace, která by obsahovala velké množství lékařských záznamů, kde by každý záznam obsahoval informace o symptomech, diagnózách a léčbě konkrétní nemoci. Po naučení byla síť schopna po předložení určité kombinace příznaků najít „nejlepší“ diagnózu a léčbu. Síť dosahovala překvapivě dobrých výsledků, vzhledem k její jednoduché struktuře. Když se v tréninkové skupině často objevovala specifická kombinace symptomů společně se specifickou diagnózou a léčbou, síť většinou navrhovala stejnou diagnózu a léčbu. V případech, kde byly nejasnosti v tréninkových datech, síť navrhla nejčastější diagnózu a léčbu. V nových situacích, síť předepsala léčbu odpovídající symptomům, které se objevili už dříve, bez ohledu na další symptomy, které byly přítomny.

4.7 Budoucnost

Jak je vidět, neuronové sítě mají široké využití ve všech možných odvětvích. V některých oblastech, jako je například rozpoznávání obrazů a predikce cen akcií, jsou už známé a velmi využívané. V následné době lze tedy očekávat rozšíření jejich použití i do jiných oblastí ale zavádění nových technologií je složitý a zdlouhavý proces, ale praxe ukazuje, že i přes velké investice se vývoj neuronových sítí opravdu vyplácí.

V této kapitole (Použití umělých neuronových sítí) bylo čerpáno ze zdrojů [4] a [19].

5 Neuronové sítě v rámci diskretní simulační optimalizace

Jako konkrétní příklad využití neuronových sítí v rámci diskretní simulační optimalizace bylo zvoleno řešení problému obchodního cestujícího (Travelling Salesman Problem). Jedná se o obtížný diskretní optimalizační problém, kde hlavní úlohou je nalezení nejkratší možné cesty, která prochází všemi zadanými body na mapě.

Problém obchodního cestujícího může být formulován jako celočíselný lineární program. Města jsou označena čísly $1, \dots, n$ a je definováno:

$$a_{ij} = \begin{cases} 1 & \text{cesta vede z města } i \text{ do města } j \\ 0 & \text{jinak} \end{cases} \quad (13)$$

Pro $i = 1, \dots, n$ nechť je c_i vzdálenost mezi městy i a j . Poté může být problém obchodního cestujícího napsán jako následující celočíselný problém lineárního programování:

$$\min \sum_{i=1}^n \sum_{j \neq i, j=1}^n c_{ij} a_{ij} \quad (14)$$

Kde:

$$0 \leq x_{ij} \leq 1 \quad \text{pro} \quad i, j = 1, \dots, n;$$

$$u_i \in \mathbb{Z} \quad \text{pro} \quad i = 1, \dots, n;$$

$$\sum_{i=1, i \neq j}^n x_{ij} = 1 \quad \text{pro} \quad j = 1, \dots, n;$$

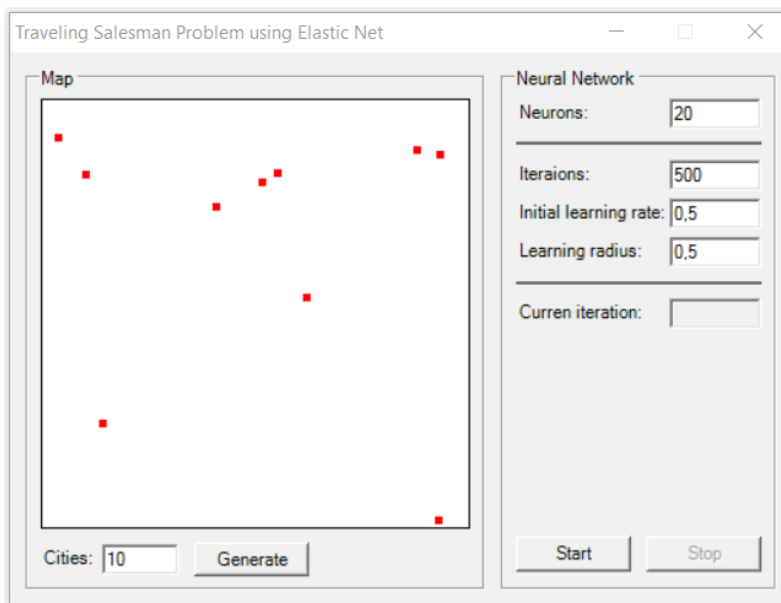
$$\sum_{j=1, j \neq i}^n x_{ij} = 1 \quad \text{pro} \quad i = 1, \dots, n;$$

Pro řešení tohoto problému je využívána takzvaná Elastická síť, která je podobná Kohonenově samoorganizující se síti. Tato Elastická neuronová síť využívá stejný princip samoorganizace ale je odlišná v interpretaci. Kohonenova síť je interpretována jako 2D mapa uzlů, ale Elastická síť se interpretuje jako kruh uzlů. Během tréninkové fáze jsou popisné vektory předkládány síti jeden po druhém, což umožňuje síti získat nějaký tvar, který reprezentuje možné řešení. V případě problému obchodního cestujícího má každý neuron dvě váhy, které reprezentují souřadnice v osách x a y . Během tréninkové fáze jsou souřadnice jednotlivých měst předávány na vstup síti postupně za sebou a neuronová síť upravuje své váhy tak, že reprezentují cestu obchodního agenta.

Nevýhodou této metody je to, že neposkytuje přesné řešení. Jednotlivé váhy neuronů mohou být blízko souřadnicím jednotlivých měst, ale nemusí být totožné. Další nevýhodou této metody je to, že neposkytuje dobré výsledky při vysokém počtu měst. Přesto je tato metoda dobrou demonstrací použití neuronových sítí.

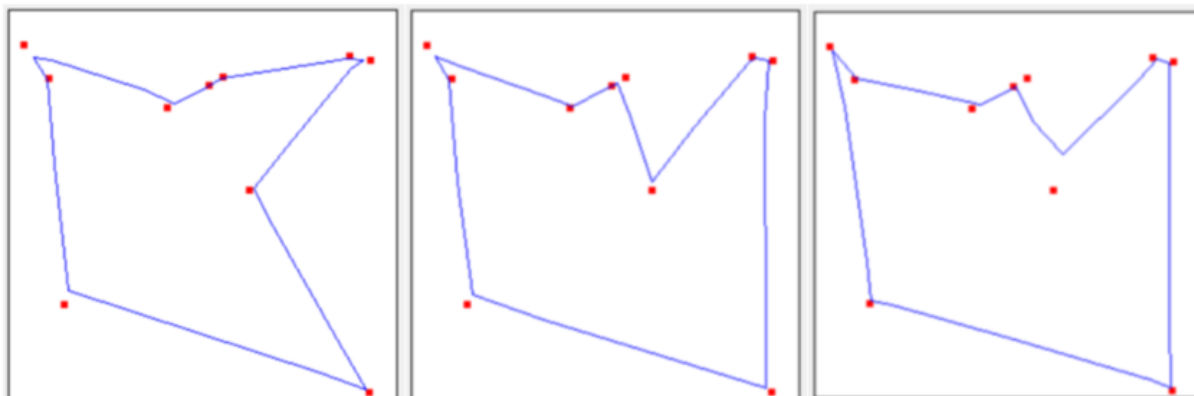
5.1 Ukázka

Jako ukázka řešení výše zmíněného problému poslouží aplikace napsána v jazyce C#, která byla převzata ze zdroje [20]. Aplikace umožňuje uživateli zadat počet měst, počet neuronů, počet iterací, počáteční rychlost učení a rádius učení.



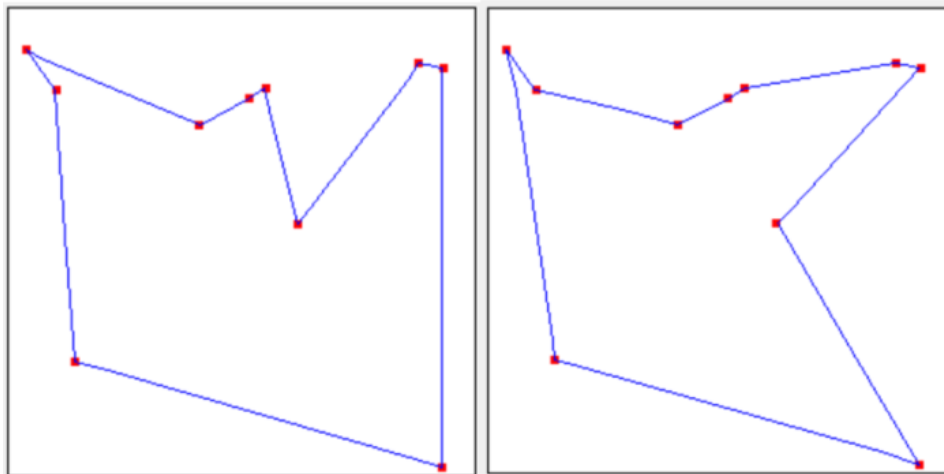
Obrázek 15: Rozhraní aplikace

Při spuštění aplikace s defaultním nastavením (10 měst, 20 neuronů, 500 iterací) nám jako výsledek vrátí trasu, po které by se měl cestující vydat. Při provedení opakovaných spuštění se stejným nastavením si můžeme povšimnout, že ne vždy dostaneme od neuronové sítě stejný výsledek. Výsledky se sice často opakují ale s trochou „štěstí“ získáme více různých tras. Jak již bylo zmíněno výše, neuronové sítě nejsou pro tento úkol perfektní a je vidět, že skrz některá „města“ výsledná trasa vůbec neprochází, ale jen se k nim přibližuje.



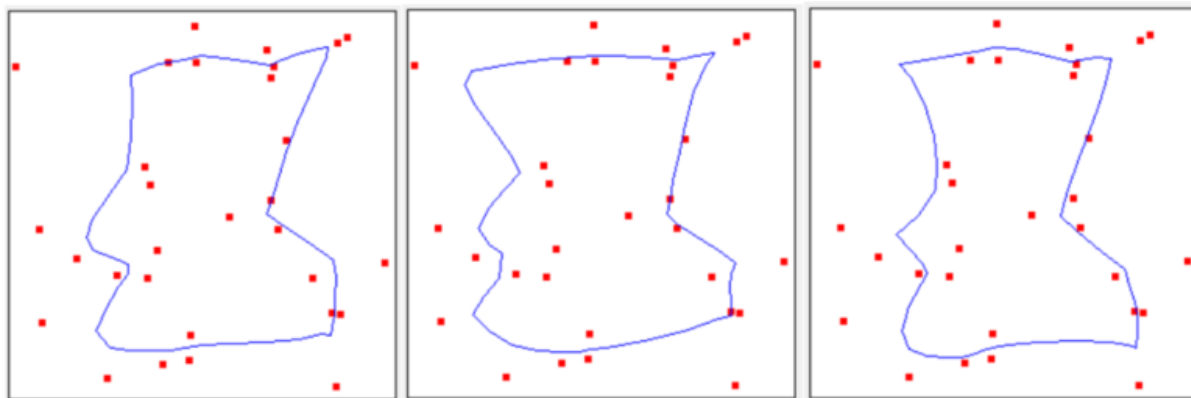
Obrázek 16: Různé výsledky aplikace (10 měst, 20 neuronů, 500 iterací)

To, že výsledná trasa některými body neprochází, můžeme částečně vyřešit zvýšením počtu iterací. Při zvýšení počtu iterací na 2000 jsou výsledné trasy mnohem přesnější, a jen málokdy se stane, že by některým bodem trasa vůbec neprocházela. Počet různých výsledných tras se také zmenšil a při třiceti spuštěních aplikace dávala jako výsledek jen tyto dvě trasy.



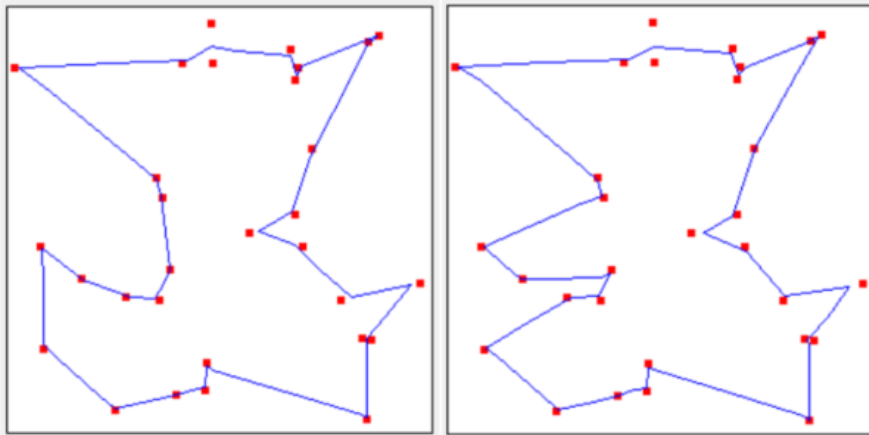
Obrázek 17: Různé výsledky aplikace (10 měst, 20 neuronů, 2000 iterací)

Pokud zvýšíme počet vygenerovaných bodů, začneme dostávat dost nepřesné výsledky. Pro 30 měst, 50 neuronů a 500 iterací dostáváme vcelku nepřesné výsledky, které se navíc liší (mírně) při každém spuštění. Jak je vidět, takovéto výsledky jsou nedostačující.



Obrázek 18: Výsledky aplikace (30 měst, 50 neuronů, 500 iterací)

Při zvýšení počtu iterací na 10000 již dostáváme použitelné výsledky (Obrázek 19). Nicméně doba výpočtu rapidně vzrostla. Pokud bychom chtěli dosáhnout lepších výsledků, bylo by nutné zvýšit počet neuronů, a také počet iterací. To nám bohužel aplikace nedovoluje, a i kdybychom splnili tyto kritéria, není jisté, zda by byl výsledek ideální.



Obrázek 19: Výsledky aplikace (30 měst, 50 neuronů, 10000 iterací)

Tato ukázka potvrzuje, že neuronové sítě se sice dají použít při řešení optimalizačních problémů, nicméně nejsou zcela ideální a jistě existují jiné algoritmy, které by v této problematice posloužily lépe. Alternativou by mohlo být vylepšení neuronových sítí a použití nových učících algoritmů, což může nastat v blízké budoucnosti.

Podklady pro tuto kapitolu byly převzaty ze zdroje [20].

6 Novinky v rámci neuronových sítí

6.1 Neuronová síť Googlu začala mluvit, mlaskat a funět

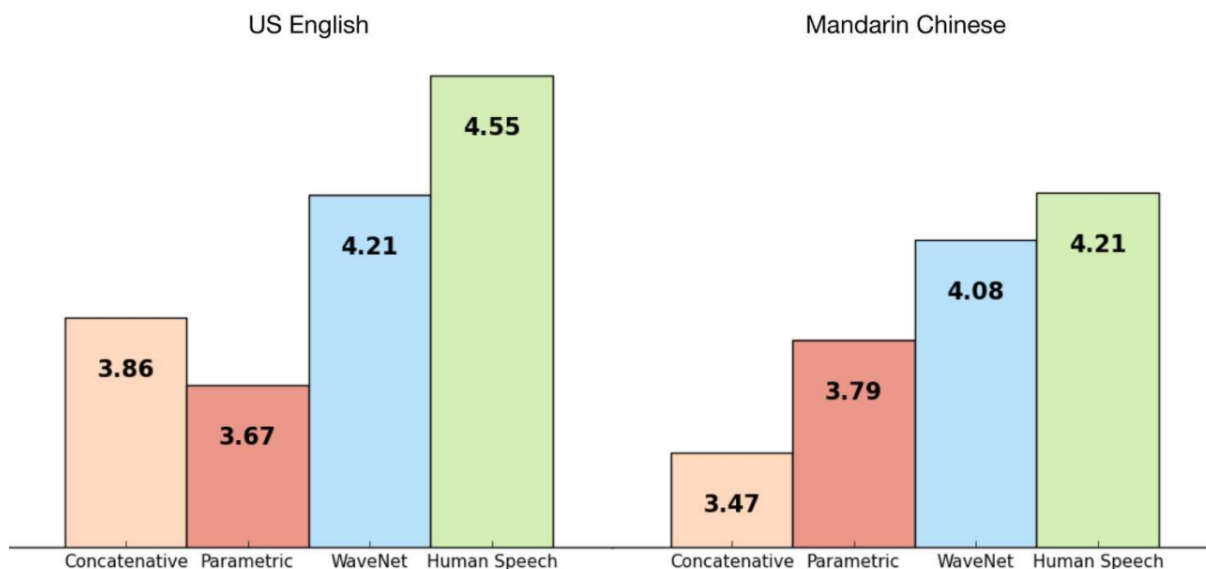
Věhlasná britská laboratoř Google DeepMind vyvinula novou konvoluční neuronovou síť WaveNet, která se specializuje na generování zvuku.

Systémů a programů na generování zvuku je sice mnoho, žádný se ale nepodobá WaveNetu. Pro generování umělého hlasu se využívají dvě základní technologie: konkatenativní a parametrická.

Konkatenativní technologie vytváří robotický hlas kombinováním skutečných hlasových úryvků člověka do jednoho celku. Technologie tedy vytváří hlas, který je velmi podobný reálnému hlasu. S touto formou se můžeme setkat například při hlášení v MHD a na nádražích, nebo například u hlasových asistentů, jako je třeba Apple Siri. Konkatenativní technologie má ovšem jednu podstatnou nevýhodu. Syntetický hlasový projev se skládá z nahrávek skutečného mluvčího, a není tedy příliš tvárný. Z mužského hlasu nelze udělat ženský a tak dále.

Tento problém řeší parametrická technologie. Jedná se o kompletní hlasový model, který je schopný generovat zcela umělý zvuk. Tento zvuk je ovšem až příliš robotický. Parametrický model je tedy tvárnější, ale každý hned pozná, že s ním mluví robot a nikoliv člověk.

V případě WaveNetu vědci začali neuronovou síť učit poslouchat hlasové a hudební nahrávky (16 kHz), a poté chtěli, aby neuronová síť sama něco řekla. Výsledky neuronové sítě jsou v mnoha testech mnohem lepší, než u konkatenativní i parametrické technologie převodu textu na řeč. Výsledky z testů pro jednotlivé technologie jsou na následujícím obrázku.



Obrázek 20: Výsledky testů jednotlivých technologií a porovnání s lidským hlasem [21]

WaveNet je neuronová síť se strojovým učením, která se díky sběru informací neustále zdokonaluje. Nemá tedy pevná pravidla „od výroby“ a na rozdíl od ostatních technologií dokáže vygenerovat i lidský dech nebo mlaskání, protože stejné zvuky slyšela při učení. Každý výstup může být trochu odlišný a do jisté míry unikátní, čímž je opět blíže lidské mluvě.

Nejpodstatnějším průlomem je to, že se jedná o zcela tvárný systém generování syntetického zvuku. V případě, že by WaveNet měl k dispozici všechny typické lidské hlasy a pomocí dalších softwarů je klasifikoval, mohl by je následně i interpretovat. To znamená, že v budoucnosti by byl schopen napodobit jakoukoli osobu. [21]

6.2 Pomocí neuronové sítě můžeme navrátit efekt rozmazání textu či tváře

Tímto problémem se zabývá čerstvá studie ze září 2016, ve které se skupina vědců z Cornell Tech a Texaské univerzity v Austinu věnuje technologii odhalování rozmazaných fotografií pomocí strojového učení. Studie je založena na tom, že pokud na text nebo lidský obličej na fotografii použijeme mozaikový efekt nebo klasické rozmazání, tak i při vysoké síle daného efektu v obrazu stále zůstává určitý korelační vztah k originálu. Člověk tento vztah samozřejmě nevidí ale stroj už ano. Pokud vezmeme nějaký velký soubor fotografií osob, použijeme na tyto fotografie několik podobných efektů a necháme neuronovou síť učit se rozpoznávat tyto jemné korelační rozdíly, můžeme následně celý proces otočit a odstranit rozmazání či jiný efekt z námi zvolené fotografie.

Úspěšnost neuronové sítě v této studii byla ohromující. Pokud bylo síti sděleno, jakým typem deformace byla fotografie zničena, dokázala neuronová síť zobrazit původní fotografii s přesností 90-100%. Pokud síť nevěděla, jaký efekt byl použit, byla úspěšnost pouze 0,19-50% v závislosti na použité deformaci. I toto číslo je však velmi vysoké, zvláště pokud si uvědomíme, že použitý efekt by měl člověka nebo text schovat na 100%. [22]

6.3 Český Google překladač začal používat umělou inteligenci

Dlouhé roky Google ve svém strojovém překladači využíval statistickou technologii, kterou ovšem v roce 2015 doplnil pomocí neuronových sítí a umělé inteligence.

Webové strojové překladače lze rozdělit do tří skupin. První skupina překladačů funguje na primitivním principu. Stroj překládá pomocí slovníku jedno slovo po druhém a případně použije sadu pokročilejších gramatických instrukcí. Jelikož překladač nezná kontext celé věty, překlad je mnohdy nesprávný a přeložená věta ani nedává smysl.

Druhou skupinu tvoří statistické překladače. Tyto překladače nejprve v režimu učení porovnávají dva stejné texty v různých jazycích a statisticky zkoumají jejich podobnost (korelaci). Problém je, že nemáme k dispozici dostatek učebních textů. Většina firem používajících statistickou technologii k učení využívali například úřední dokumenty EU nebo Bibli. Tyto texty ovšem neobsahují reálné fráze používané v běžné mluvě.

Třetí a nejnovější skupinou jsou překladače využívající strojové učení a neuronové sítě. Tyto překladače jsou dosti podobné statistickým překladačům, s tím rozdílem, že strojové učení je v tomto případě univerzálnější a vnitřní pravidla si v mnoha ohledech vytváří samo. Strojové učení se díky neuronovým sítím dokáže učit i více jazyků naráz, nebo například, pokud umí překládat jazykové páry japonština-angličtina, angličtina-japonština, korejština-angličtina, angličtina-korejština, tak dokáže přeložit i doposud neznámý jazykový pár japonština-korejština. Tato technologie se tedy neučí jazyky jako takové, ale učí se, jak překládat zadané texty aby se co nejvíce podobali textům, které byly poskytnuty při učení. [23]

6.4 Neuronová síť se naučila odezírat ze rtů

Koncem roku 2016 přišli vědci z Oxfordu s novou technologií zvanou LipNet. Tato technologie byla schopna odezírat z úst s přesností 93,4% na předem připraveném vzorku. LipNet se nejprve díky obrazové analýze naučil, jakým způsobem se deformují ústa při vyslovování jednotlivých slov, a následně jen stačilo vytvořit model.

Na stejném problému v Oxfordu pracoval ještě druhý tým (sponzorovaný laboratoří Google DeepMind), který svoji neuronovou síť učil pomocí kanálu BBC. Po zhlédnutí tisíců televizních pořadů byla tato neuronová síť schopna odezírat z úst s úspěšností 46,8%. Nutno podotknout, že LipNet sice dosáhl přibližně dvakrát takové úspěšnosti, ale na předem pečlivě připraveném vzorku. Neuronová síť od DeepMindu dokáže s téměř poloviční úspěšností odezírat z jakéhokoliv obrazového vstupu. Také stojí za zmínku, že když z videí odezíral lidský profesionál, dosáhl úspěšnosti pouze 12,4%.

V běžném životě by tato technologie mohla pomoci mnoha lidem s poruchou sluchu, ovšem je zde i kontroverznější využití. Pokud bychom tuto technologii spojili s kamerovým systémem, mohli bychom odezírat, o čem se někdo baví na kilometry daleko, což by mohlo značně narušovat soukromí. [24]

6.5 Robotické stroje budou vybaveny umělou inteligencí, aby se automaticky zlepšovali ve výrobě

V hromadné výrobě jsou dnes pokročilí roboti využíváni napříč mnoha odvětvími, od výroby mobilních telefonů až po výrobu automobilů. Díky neustálému vylepšování produktů a zavádění nových modelů, přestávají klasičtí roboti v pokročilejších segmentech stačit. Programování robotů se v posledních letech dosti zlepšilo a usnadnilo, ani to ale nestačí a dochází k integraci umělé inteligence.



Obrázek 21: Ukázka využití programovatelných robotů [25]

Jeden z největších výrobců programovatelných průmyslových robotů Fanuc (jehož roboty využívají například i v Tesla Motors nebo Applu) se rozhodl, že začne spolupracovat s firmou Nvidia, kvůli možnosti využití výpočetního hardwaru Tegra, který podporuje akceleraci strojového učení a umělou inteligenci (hluboké neuronové sítě). Díky tomu bude moci každý robot mít vlastní „mozek“, který bude schopen analyzovat veškerou práci, kterou dělá nejen robot samotný, ale i ostatní roboti. Díky tomu se robot může zdokonalovat v daném procesu mnohem rychleji, než by tou bylo při použití klasického ručního programování. Stroje musí často vyrábět různé věci v různý čas, a tak se počítá i s jistým předáváním zkušeností mezi roboty. Roboti se budou moci mezi sebou „školit“ a předávat si zdokonalené postupy při výrobě. Roboti a celá továrna se budou moci „přeučit“ doslova přes noc. To může mít značné výhody, zvláště když se technologický vývoj exponenciálně zrychluje a nové produkty jsou uváděny do výroby stále častěji.



Obrázek 22: Robotické systémy od firmy Fanuc [25]

Fanuc tedy chce implementovat neuronové sítě pro programování robotů, kde optimální sled příkazů nebude muset vymýšlet a psát programátor, ale stroj na ně postupně přijde sám. Otázkou ovšem zůstává, jestli se bude opravdu jednat o optimální postup. [25]

7 Závěr

První část této bakalářské práce obsahuje základní popis některých významných typů neuronových sítí. V úvodní kapitole byla popsána historie neuronových sítí, následně pak zjednodušený model biologického neuronu a dále matematický model neuronu neboli Perceptron. Perceptron je základní stavební jednotkou neuronových sítí, které byly rozděleny do několika kategorií: Dopředné sítě, Rekurentní sítě a Samoorganizační sítě. Pro každou kategorii byli zvoleni dva zástupci, díky nimž byl nastíněn základní princip fungování. Všechny neuronové sítě fungují za pomoci jednotlivých perceptronů a není snadné je jednoznačně zařadit do určitých kategorií, proto se může stát, že v jiných publikacích se setkáme s odlišným rozdělením.

Druhá část bakalářské práce pojednává o možném použití neuronových sítí. Záměrem bylo nastínění možností využití a dále analýza možnosti implementace v rámci diskrétní simulační optimalizace. Při této analýze byl řešen problém obchodního cestujícího. Tento problém byl vybrán z důvodu podobnosti s problémy, které se vyskytují v průmyslovém inženýrství. Obchodní cestující musí navštívit všechna zadaná města tak, aby jeho celková uražená cesta byla co možná nejkratší. Obdobné problémy také řeší průmyslové inženýrství v rámci logistiky, kde je snaha o co možná nejkratší trasy přepravovaného materiálu.

Neuronové sítě mají opravdu široké spektrum využití (což naznačují i uvedené příklady), ale v rámci diskrétní simulační optimalizace nejsou nejvhodnější volbou. Jak již bylo naznačeno v dřívější kapitole, pro řešení problematiky diskrétní simulační optimalizace můžeme nalézt vhodnější algoritmy než zrovna neuronové sítě (genetické algoritmy, evoluční strategie, stochastické metody atd.). Využitím neuronových sítí sice můžeme daný problém vyřešit, ale pokud se bude jednat o něco složitějšího, výpočet bude trvat velmi dlouho, což není žádoucí. Neuronové sítě představují mnoho možností a mají velký potenciál, k plnému využití je ale třeba do budoucnosti neuronové sítě ještě zdokonalit.

V poslední kapitole jsou zmíněny některé novinky v oblasti neuronových sítí a umělé inteligence, které mohou sloužit jako další ukázka nynějšího využití neuronových sítí a poukazovat na směr aktuálního vývoje.

Bibliografie

- [1] J. Tučková, Vybrané aplikace umělých neuronových sítí při zpracování signálů, Praha: České vysoké učení technické v Praze, 2009.
- [2] E. Volná, „Neuronové sítě 1,“ Ostravská univerzita v Ostravě, Ostrava, 2008.
- [3] J. Šíma a R. Neruda, „Teoretické otázky neuronových sítí,“ Matfyzpress, Praha, 1996.
- [4] L. Fausett, Fundamentals of Neural Networks, New Jersey: Prentice-Hall, Inc., Englewood, 1994.
- [5] I. Zelinka, Umělá inteligence: hrozba nebo naděje?, Praha: BEN - technická literatura, 2003.
- [6] J. Jantzen, „Introduction To Perceptron Networks,“ Technical University of Denmark, Lyngby, 1998.
- [7] „Univerzitní informační systém MENDELU,“ Mendelova univerzita v Brně, [Online]. Available: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471. [Přístup získán 10 Listopad 2016].
- [8] P. McCollum, „An Introduction to Back-Propagation Neural Networks,“ Encoder, [Online]. Available: <http://www.seattlerobotics.org/encoder/nov98/neural.html>. [Přístup získán 1 Listopad 2016].
- [9] I. Vondrák, „Neuronové sítě,“ Technická univerzita Ostrava, Ostrava, 1994.
- [10] N. Nikolaev a H. Iba, „Adaptive Learning of Polynomial Networks: Genetic Programming, Backpropagation and Bayesian Methods,“ Springer, New York, 2006.
- [11] D. Kabáth, „Fakulta Informatiky Masarykovy Univerzity,“ 1 Listopad 2009. [Online]. Available: https://nlp.fi.muni.cz/uui/referaty2009/kabath_david/referat.pdf. [Přístup získán 1 Listopad 2016].
- [12] B. Stískal, „Návrh algoritmů pro neuronové sítě řídicí síťový prvek,“ Vysoké učení technické v Brně, Brno, 2008.
- [13] S. Reid, „10 misconceptions about Neural Networks,“ Turing Finance, 8 Květen 2014. [Online]. Available: <http://www.turingfinance.com/misconceptions-about-neural-networks/>. [Přístup získán 5 Prosinec 2016].
- [14] V. Jirsík a P. Hráček, Neuronové sítě, expertní systémy a rozpoznávání řeči, Brno: Vysoké učení technické, 2002.
- [15] A. Vojáček, „Samoučící se neuronová síť - SOM, Kohonenovy mapy,“ Automatizace.hw.cz, 14 Květen 2006. [Online]. Available: <http://automatizace.hw.cz/clanek/2006051401>. [Přístup získán 5 Prosinec 2016].

- [16] A. Parsch, „LoFLYTE,“ Directory of U.S. Military Rockets and Missiles, Appendix 4, 2 Červen 2004. [Online]. Available: <http://www.designation-systems.net/dusrm/app4/loflyte.html>. [Přístup získán 28 Únor 2017].
- [17] C. Bellay, „X-51A Waverider,“ U.S. Air Force, 2 Březen 2011. [Online]. Available: <http://www.af.mil/AboutUs/FactSheets/Display/tabid/224/Article/104467/x-51a-waverider.aspx>. [Přístup získán 28 Únor 2017].
- [18] R. Nemiroff a J. Bonnell, „Astronomy Picture of the Day,“ NASA, 29 Březen 2004. [Online]. Available: <https://apod.nasa.gov/apod/ap040329.html>. [Přístup získán 28 Únor 2017].
- [19] A. Podyablonskiy, „Bankovní institut vysoká škola: Závěrečné práce,“ 1 Duben 2013. [Online]. Available: https://is.bivs.cz/th/15623/bivs_b/130426_BP_Podyablonskiy.pdf. [Přístup získán 28 Únor 2017].
- [20] A. Kirillov, „Neural Networks on C#,“ Code Project, 19 Listopad 2006. [Online]. Available: <https://www.codeproject.com/Articles/16447/Neural-Networks-on-C->. [Přístup získán 29 Březen 2017].
- [21] J. Čížek, „Další průlom: Neuronová síť Googlu začala mluvit, mlaskat a funět,“ Žive.cz, 13 Zář 2016. [Online]. Available: <http://www.zive.cz/clanky/dalsi-prulom-neuronova-sit-googlu-zacala-mluvit-mlaskat-a-funet/sc-3-a-184186/default.aspx>. [Přístup získán 10 Květen 2017].
- [22] J. Čížek, „Rozmazali jste text a tvář kvůli ochraně soukromí? Neuronová síť to stejně rozlouskne,“ Živě.cz, 16 Zář 2016. [Online]. Available: <http://www.zive.cz/clanky/rozmazali-jste-text-a-tvar-kvuli-ochrane-soukromi-neuronova-sit-to-stejne-rozlouskne/sc-3-a-184239/default.aspx>. [Přístup získán 10 Květen 2017].
- [23] J. Čížek, „Český Google Překladač začal používat umělou inteligenci. Konec „drahoušků““, Živě.cz, 20 Duben 2017. [Online]. Available: <http://www.zive.cz/clanky/cesky-google-prekladac-zacal-pouzivat-umelou-inteligenci-konec-drahousku-zakazniku/sc-3-a-187234/default.aspx>. [Přístup získán 10 Květen 2017].
- [24] J. Čížek, „Neuronová síť Googlu sledovala BBC tak dlouho, dokud se nenaučila odezírat z úst. Nejlépe ze všech,“ VTM.cz, 25 Listopad 2016. [Online]. Available: <https://vtm.zive.cz/bleskovky/neuronova-sit-googlu-sledovala-bbc-tak-dlouho-dokud-se-nenaučila-odezírat-z-úst-nejlépe-ze-všech/sc-871-a-185060/default.aspx>. [Přístup získán 10 Květen 2017].
- [25] K. Javůrek, „Japonský výrobce dá robotickým strojům umělou inteligenci, aby se automaticky zlepšovaly ve výrobě,“ VTM.cz, 23 Říjen 2016. [Online]. Available: <https://vtm.zive.cz/clanky/japonsky-vyrobce-da-robotickym-strojum-umelou-inteligenci-aby-se-automaticky-zlepsovaly-ve-vyrobe/sc-870-a-184668/>. [Přístup získán 10 Květen 2017].