

Neuronové sítě

aneb to málo, co o nich vím

J. Doubravová

Seminář o seismologickém softwaru, 2014

Outline

- 1 Teorie
 - Úvodem
 - Matematický model neuronové sítě
 - Neuronové sítě
 - Klasické modely neuronových sítí
 - Asociativní sítě
- 2 Ukázka
- 3 Webnet a výhledy

Outline

- 1 Teorie
 - Úvodem
 - Matematický model neuronové sítě
 - Neuronové sítě
 - Klasické modely neuronových sítí
 - Asociativní sítě
- 2 Ukázka
- 3 Webnet a výhledy

Troška historie

- W. McCulloch, W. Pitts 1943 - matematický model jednoduchého neuronu, teoretická práce, parametry $\{-1,0,1\}$
- D. Hebb 1949 The Organization of Behavior - učící pravidlo pro synapse neuronů
- M. Minsky 1951 - první neuropočítač Snark, nebyl použit k řešení praktických problémů
- F. Rosenblat 1957 - perceptron = zobecnění Pitts+Culloch neuronu s reálnými parametry, učící algoritmus, dokázal, že pro daná tréninková data nalezne odpovídající váhy, pokud řešení existuje. Společně s C. Wightmanem sestrojili první úspěšný neuropočítač Mark I Perceptron pro rozpoznávání obrazců. Znak promítali na světelnou tabuli, odkud ho snímali 20x20 fotovodiči = 400 intenzit "pixelů"

Troška historie

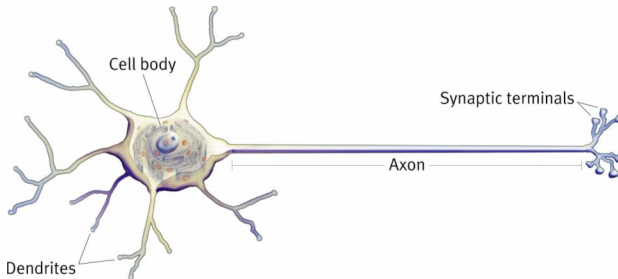
- B. Widrow (1960's) ADALINE adaptive linear element, firma na výrobu hardware pro neuropočítače
- pak útlum - Minsky, Papert zdiskreditoval (XOR kauza)
- renesance 80. léta J. Hopfield ukázal souvislosti s fyzikálními modely magnetických materiálů
- D. Rumelhart, G. Hinton, R. Williams 1986 - učící algoritmus - zpětné šíření chyb pro vícevrstvou síť. Aplikován T. Sejnowksim a C. Rosenbergem pro NETtalk - převod psané angličtiny na mluvenou
- 1987 první IEEE International Conference on Neural Networks v San Diegu a pak už jenom sluníčko, výzkumy, výuka, aplikace...

Troška biologie

- nervová soustava je interface mezi organismem a okolním prostředím, reaguje na podněty jak z vnějšku tak na vnitřní stavy systému
- senzory nazýváme receptory, reagují na různé typy podnětů (tepelné, světelné, mechanické...)
- informace se šíří pomocí sítě neuronů (zde dochází k filtraci a kompresi) k efektorům (svaly, žlázy)
- celý proces je reflex, reflexy se šíří po projekčních drahách do mozkové kůry - 6 oblastí = paralelní zpracování jednotlivých smyslů
- neurony: jen v mozkové kůře 13-15 mld. neuronů, jeden může být propojen až s 5000 dalšími

Troška biologie

- anatomie neuronu: soma (tělo), dendrity (vstupy), axon a axonální/synaptická zakončení (výstupy, dotýkají se dendritů jiných neuronů)
- propojení dendritu a axonu = chemická synapse s různou propustností, ma dělíme je na excitační (rozšiřují informaci) a inhibiční (tlumicí)



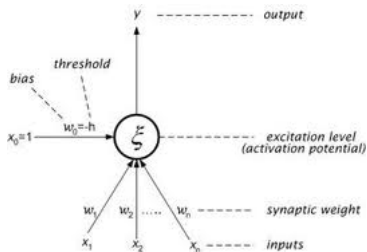
Troška biologie

- přenos informace elektrický impuls, když překročí na dendritech určitý práh, neuron generuje elektrický impuls a šíří tak informaci dál
- synaptická propustnost se po každém průchodu signlu mění, může se i úplně prorušit (zapomeneme) nebo vytvořit synapse nová (učíme se)

Outline

- 1 Teorie
 - Úvodem
 - **Matematický model neuronové sítě**
 - Neuronové sítě
 - Klasické modely neuronových sítí
 - Asociativní sítě
- 2 Ukázka
- 3 Webnet a výhledy

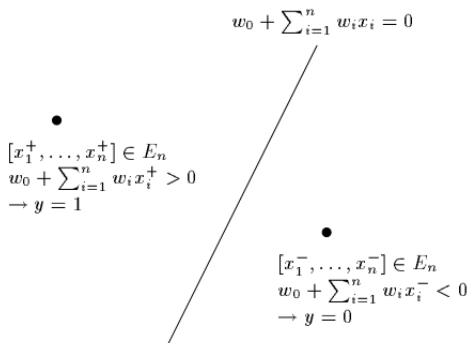
Formální neuron



- n reálných vstupů $x =$ dendrity, vstup $x_0 = 1$
- váhy $w =$ synaptické váhy (kladné excitační, záporné inhibiční), váha $w_0 = -h$ tzv. bias
- vnitřní potenciál neuronu
- práh h
- výstup je dán aktivační funkcí

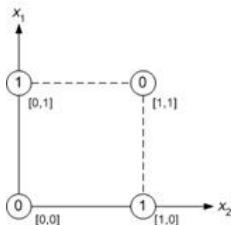
$$y = \sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$

Formální neuron



- n rozměrný euklidovský prostor
- x_i souřadnice
- rovnice nadroviny
 $w_0 + \sum_{i=1}^n w_i x_i = 0$
- jeden neuron tedy rozděluje prostor na dva poloprostory

XOR problém



- jednoduchá logická funkce
- 2D prostor pro dvouvstupý XOR
- 1 neuron rozdělí rovinu přímkou na dva poloprostory
- nelze vyřešit
- potřeba zapojit do sítě

Outline

- 1 Teorie
 - Úvodem
 - Matematický model neuronové sítě
 - **Neuronové sítě**
 - Klasické modely neuronových sítí
 - Asociativní sítě
- 2 Ukázka
- 3 Webnet a výhledy

Neuronová síť

- soustava formálních neuronů, které jsou vzájemně propojeny
- výstup neuronu je obecně vstupem více neuronů
- počet a propojení neuronů = topologie sítě (architektura sítě)
- vstupní (jako receptory), výstupní (jako efekторы) a skryté neurony
- stavy všech neuronů v síti = stav sítě
- všechny synaptické váhy = konfigurace sítě
- tři úrovně dynamiky sítě: organizační (změna topologie), aktivní (změna stavu), adaptivní (změna vah) = v nervové soustavě se to děje najednou

Dynamika

- organizační - dělíme na cyklické (rekurentní) = existuje skupina (alespoň jeden) neuronů zapojená v cyklu, a acyklické (dopředné)
- acyklická lze vždy uspořádat do vrstev, kde každý neuron jedné vrstvy propojíme s každým neuronem v další vrstvě, chybějící propojení formálně nahradíme $w = 0$
- aktivní - počáteční stav a způsob změny v čase, obecně výpočet ve spojitém čase ale obvykle diskrétní
- funkce každého neuronu - obvykle stejná pro všechny, ale nemusí být jako v nervové soustavě

Aktivační funkce

ostrá nelinearita

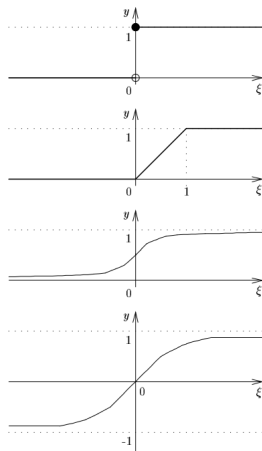
$$\sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases}$$

saturovaná lin. fce

$$\sigma(\xi) = \begin{cases} 1, & \xi > 0 \\ \xi, & 0 \leq \xi \leq 1 \\ 0, & \xi < 0 \end{cases}$$

standard sigmoid $\sigma(\xi) = \frac{1}{1+e^{-\xi}}$

$$\tanh \sigma(\xi) = \frac{1-e^{-\xi}}{1+e^{-\xi}}$$

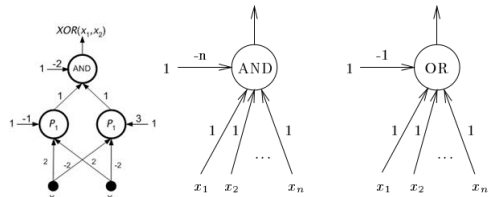
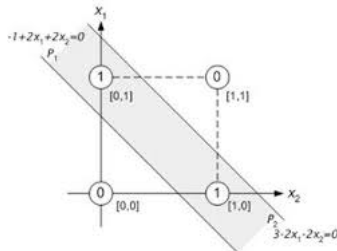


Adaptivní dynamika

- specifikuje počáteční stavy vah (např. náhodně) a jak se budou v čase měnit = učit
- optimalizace nelineárního problému - může být časově náročné
- požadavovanou fci definujeme tréninkovou množinou - známe vstup a požadovaný výstup = s učitelem (supervised learning)
- naopak samoorganizace (self-organizing map) nemá učitele a sama seskupí vstupní vektory do shluků na základě podobnosti

Geometrická reprezentace

- víme, že jeden neuron rozdělí prostor na dva poloprostory oddělené nadrovinou
- budeme-li mít první vrstvu sítě, kde každý neuron takto rozdělí prostor, ve druhé vrstvě můžeme pomocí logických fcí AND a OR implementovat průnik a sjednocení těchto oblastí
- příkladem budiž problém XOR

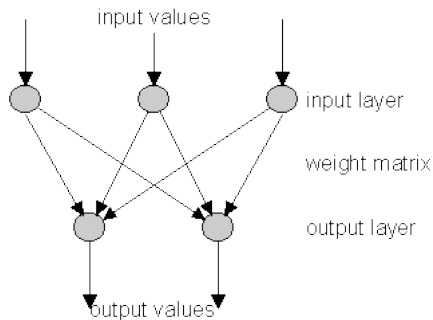


Outline

- 1 Teorie
 - Úvodem
 - Matematický model neuronové sítě
 - Neuronové sítě
 - **Klasické modely neuronových sítí**
 - Asociativní sítě
- 2 Ukázka
- 3 Webnet a výhledy

Síť perceptronů

- n reálných vstupů, m logických výstupů, 1 vrstva, propojení každý s každým, ostrá nelinearita, funkce sítě zadána tréninkovou množinou



Síť perceptronů

- adaptace nemusí mít řešení - viz XOR kontrabeispiel, nebo není fce = požadujeme více řešení pro stejný vstup
- adaptace probíhá v adaptačních cyklech, dle perceptronového pravidla, kde $0 < \varepsilon \leq 1$ je rychlost učení (často jako u lidí, napřed malá, pak přidáme na detailech)
- $w_{ji}^t = w_{ji}^{t-1} - \varepsilon x_{ki} (y_j(\mathbf{w}^{t-1}, \mathbf{x}_k) - d_{kj})$ $\begin{matrix} j = 1, \dots, m \\ i = 0, \dots, n \end{matrix}$
- velká závorka = rozdíl j-tého výstupu pro k-tý vzor a požadovaného správného výstupu = chyba $\in \{-1, 0, 1\}$
- Rosenblatt dokázal, že pokud řešení existuje (lze oddělit nadrovinou), pak je nalezeno po konečném počtu adaptačních kroků = celkem k ničemu pořádnému, ale teoretický základ

Vícevrstvá síť

- nejznámější typ s učícím algoritmem zpětného šíření chyb - backpropagation
- zobecnění sítě perceptronů - přidáme skryté vrstvy, aktivační funkce je tanh, sigmoida... = potřebujeme ji v backpropagation diferencovat

- chyba vzhledem k tréninkové množině (p

vzorů) $E(\mathbf{w}) = \sum_{k=1}^P E_k(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^P \sum_j (y_j(\mathbf{w}, \mathbf{x}_k) - d_{kj})^2$ - tu

chceme minimalizovat

- gradientová metoda $\Delta w_{ji}^t = -\varepsilon \frac{\partial E}{\partial w_{ji}}(\mathbf{w}^{t-1})$, takže klasicky - může být pomalé, najít lokální min. ne globální, potřebujeme tu derivaci chybové funkce

$$\frac{\partial E}{\partial w_{ji}} = \sum_{k=1}^P \frac{\partial E_k}{\partial w_{ji}} = \sum_{k=1}^P \frac{\partial E_k}{\partial y_j} \cdot \frac{\partial y_j}{\partial \xi_j} \cdot \frac{\partial \xi_j}{\partial w_{ji}} = \sum_{k=1}^P \frac{\partial E_k}{\partial y_j} \cdot y_j(1 - y_j) \cdot y_i$$

(např. pro sigmoidu)

Metoda backpropagation

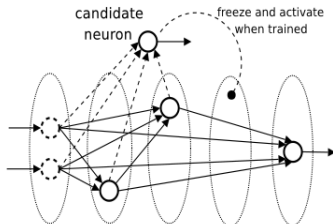
- $\frac{\partial E_k}{\partial y_j} = y_j - d_{kj}$ pro j je výstup
- pro skryté neurony potom $\frac{\partial E_k}{\partial y_j} = \sum_r \frac{\partial E_k}{\partial y_r} \cdot \frac{\partial y_r}{\partial \xi_r} \cdot \frac{\partial \xi_r}{\partial w_{rj}}$, kde r jsou neurony do kterých j -tý vstupuje např. $\sum_r \frac{\partial E_k}{\partial y_r} \cdot y_r (1 - y_r) w_{rj}$
- takto odkonce spočteme tu derivaci (poslední skrytou vrstvu z výstupu, předposlední z poslední skryté atd..)
- platí pro acyklickou síť
- omezeně místo gradientů sdružené gradienty nebo newtonova metoda
- pro cyklické topologie = zbecněná metoda rekurentní backpropagation

MADALINE

- Multiple ADaptive LINear Element (Widrow, Hoff)
- jako síť perceptronů, aktivační funkce ale lineární $y_j = \sum_{i=1}^n w_{ji}x_i$
- výstupy dávají “vzdálenost” od nadroviny $w_{j0} + \sum_{i=1}^n w_{ji}x_i = 0$
(není to vzdálenost ale je to v abs.h. lin. proporcionální)
- učení nejčastěji backpropagation = lineární, dává paraboloidní chybovou funkci - nejsou problémy s lok. minimy
- Widrow+Hoff LMS (Least mean square) - oprava po každém tréninovém vzoru $w_{ji}^t = w_{ji}^{t-1} - \varepsilon x_{ki} (y_j(\mathbf{w}^{t-1}, \mathbf{x}_k) - d_{kj})$

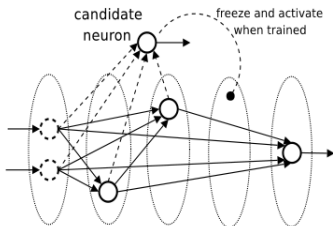
Sítě s kaskádovou architekturou

- Fahlman, Lebiere (1990), ve skryté vrstvě jsou výstupy odspoda nahoru předávány jako vstupy neuronů stejné vrstvy (všech)
- na začátku nedefinujeme topologii skryté vrstvy, napřed naučíme bez skryté vrstvy a když už se výsledek nezlepšuje, přidáme skrytou



Sítě s kaskádovou strukturou

- přidáme “kandidáta”
- připojíme jeho vstupy ale ne výstupy, učíme tak, že hledáme maximální korelaci mezi chybou sítě a výstupem kandidáta
- Když naučíme (adaptace vah je hotova, nezlepšuje se už korelace), zapojíme do sítě a celou síť znova adaptujeme = složitost sítě se najde sama, učení je velmi rychlé



Outline

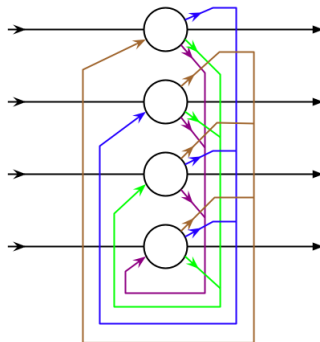
- 1 Teorie
 - Úvodem
 - Matematický model neuronové sítě
 - Neuronové sítě
 - Klasické modely neuronových sítí
 - **Asociativní sítě**
- 2 Ukázka
- 3 Webnet a výhledy

Asociativní síť

- Anderson - lineární asociativní síť = jako asociativní paměť - vybavení na základě částečné informace
- autoasociativní - doplnění informací, detailů na základě zevrubné informace, heteroasociativní - doplnění souvislostí
- formálně jako MADALINE ale bez bias vstupu jako $w_{j0} = 0$, geometricky ty nadroviny prochází nulou, maticově $\mathbf{y} = \mathbf{W}\mathbf{x}$
- adaptace podle Hebbova zákona (zase z biologie): synaptická váha mezi dvěma neurovy je dána součinem jejich stavů - čili pokud jsou oba aktivovány, vazba se ještě posiluje, jinak naopak zeslabuje $w_{ji}^t = w_{ji}^{t-1} - d_{kj}x_{ki}$ pro všechny tréninkové vzory, na konci je tedy $w_{ji} = \sum_{k=1}^p d_{kj}x_{ki}$
- pokud jsou tréninkové vzory ortonormální, pak síť dobře asociuje - tedy máme-li vstup podobný nějakému vzoru, přiřadí

Hopfieldova síť

- McCulloch, Pitts, později profláknul právě Hopfield
- na počátku je úplná cyklická topologie, každý s každým (ne sám se sebou)



Hopfieldova síť

- výstupy bipolární, celočíselné, +1,-1, adaptace podle Hebba

$$w_{ji} = \sum_{k=1}^p x_{kj}x_{ki}$$

- $w_{ij} = w_{ji}$; neorientované spoje, čili můžeme překreslit, aktivní dynamika = zvolíme bod a počítáme výstup, paralelně nemusí fungovat
- fyzika - modely mg. materiálů ve statistické fyzice

Další

- Boltzmannův stroj
- Sítě se samoorganizací
- Spousta šílený teorie o složitosti sítí atd

Úskalí

- přeučení sítě - naučíme ji details, nepřesnosti měření a chyby - musí zůstat schopnost generalizace
- musíme vhodně zvolit topologii, a to se taky těžko dělá...cit?

Backpropagation

Hopfield

Konfigurace sítě

- SLRNN - single layer recurrent neural network (zobecnění RTRNN Williams, Zipser 1989) - některé vstupy sítě jsou zpět zavedené výstupy z předchozího kroku a to s různým zpožděním
- síť má pak paměť - reaguje na předchozí stavy
- vstupy - signál ve frekvenčních pásmech (9x), polarizace ve 3 pásmech, horizontální a vertikální tj. $18+6=24$
- každý ještě STA/LTA poměry, a podzorkovat (mean v okně)
- výstup - 1 pokud je tam jev, 0 pokud není
- aktivační funkce tanh
- učení - modifikace backpropagation - backpropagation through time
- nějaké konfigurace naučeny, nutno zkusit na nových datech....