

Samoučící se neuronová síť - SOM, Kohonenovy mapy

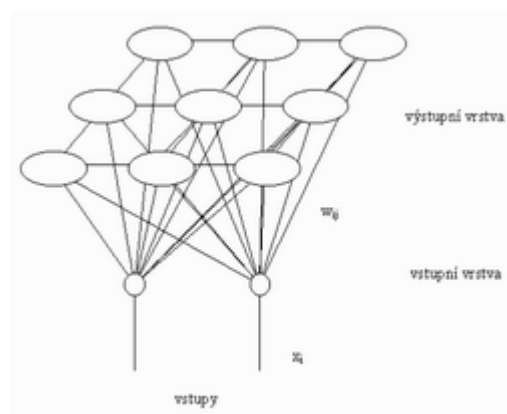
Antonín Vojáček, 14 Květen, 2006 - 10:33

- [Měření a regulace](#)

Samooorganizující neuronové sítě s učením bez učitele jsou stále více využívány pro rozlišení, rozpoznávání a třídění neznámých číslicových signálů a dat. Hlavním představitelem jsou Kohonenovy mapy. Ty sami rozpoznávají shodné prvky nebo naopak rozdíl mezi signály a je tak s nimi možné zpracovat úplně neznámé signály a data.

Co je to SOM a Kohonenovy mapy ?

Dnes již mezi základní typy neuronových sítí a také mezi nejpoblárnější, patří tzv. SOM = Self-Organizing Maps (Samooorganizující se mapy), častěji známé po svém "stvořiteli" jako Kohonenovy mapy. Ty patří do skupiny samoučících se neuronových sítí, tzn. sítí s učením bez učitele, které ke svému nastavování nepotřebují ideální vzory. To znamená, že k učení sítě stačí jen velká skupina reálných signálů, z nichž některé mají určitou společnou vlastnost nebo naopak výrazné odlišnosti a již k nim nemusí být přiřazeny žádné ideální učící signály nebo informace (target = cílové hodnoty). Ty v případě tzv. učení s učitelem udávají konečný cílový stav, do kterého se má síť učením dostat. A právě jejich získání bývá často velkým problémem. Naopak u SOM (Kohonenovy mapy) nám například stačí jen skupina nahraných řečových signálů a během učení si síť již sama nalezne společné znaky a odlišnosti, podle kterých se bude ve své aktivní činnosti rozhodovat. A to je ta výhoda, která za cca 20 let od vzniku Kohonenových map z nich udělala velmi často využívanou a velmi oblíbenou neuronovou síť.



Obr. 1. Příklad jednoduché fyzické struktury Kohonenovy mapy

K čemu to je ?

Svojí schopností samooorganizace a shlukování objektů s podobnými vlastnostmi do skupin jsou Kohonenovy mapy přímo předurčeny pro aplikace rozhodování, rozlišování a třídění objektů, signálů, značek apod. Častou aplikací je rozpoznávání řeči (např. přepis mluveného slova na napsané apod.) nebo přeložení psaného textu na tištěný (grafity). Množství aplikací je však nevyčísitelné a ne nadarmo i odborníci v oboru říkají: "Pokud nevíš, jaký algoritmus pro zpracování signálů použít, použij SOM (Kohonenovy mapy)." V tomto případě totiž o zpracovávání signálů není teoreticky nutné cokoli vědět a SOM si je samo přebere a roztrídí.

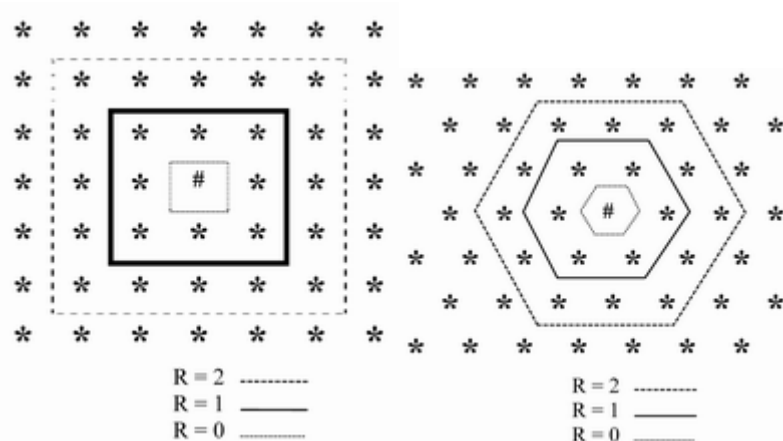
Možné aplikace:

- zpracování řeči
- úprava zvuku
- zpracování obrazu (videa a fotek)
- hledání a detekce osob podle fotografií
- bezpečnostní aplikace
- přepis ručně psaného textu na tištění
- hledání podobných znaků v úplně neznámých signálech
- odstranění neznámého rušení
- automatické třídění

Princip a struktura

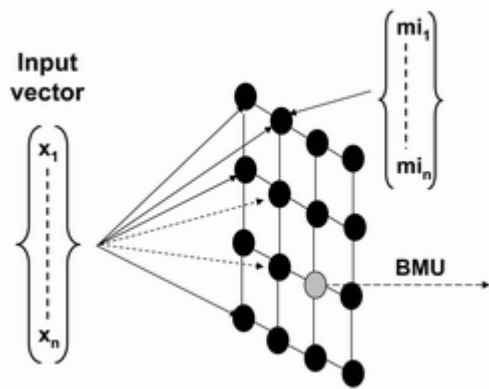
I když základní princip Kohonenovy mapy není příliš obtížný, vyžaduje jeho pochopení trošku představivosti. Pokusím se ho však popsat bez nutnosti bližších znalostí neuronových sítí, pro ty, co s nimi ještě nemají žádné zkušenosti.

Základ tvoří uspořádaná struktura neuronů, které se v tomto případě dají představit jako body (kroužky), kde ke každému přísluší unikátní vektor koeficientů označované jako váhy W (m_i na obrázku 3.). Nejčastěji má struktura formu dvourozměrné ($k=2$) čtvercové nebo obdélníkové matice, hexagonálního útvaru nebo někdy i jednorozměrného vektoru ($k=1$).



Obr. 2. Možné struktury uspořádání neuronů (*) s definicí okolí R vítězného neuronu (#)

Rozměr struktury (k) však nemá nic společného s počtem vah každého neuronu (bodu), tedy dimenzí neuronu (n). Obvykle platí, že $k < n$, stejně jako počet neuronů $m < n$. Naopak tvar struktury uspořádání neuronů má vliv na učení mapy a počet vah je vždy shodný s počtem parametrů vstupních vzorků, hodnot nebo koeficientů vstupujícího zpracovávaného signálu – viz obrázek 3.



Obr. 3. Struktura neuronů Kohonenovy mapy s vítězným neuronem BMU (vstupní vektor s hodnotami $x_1 \dots x_n$, neuron s váhami $m_i = W$)

Tvar uspořádání neuronů má vliv na volbu tzv. okolí neuronu R , které vymezuje jeho sousedy (sousední, nejbližší postavené neurony). V maticovém uspořádání neuronů (čtvercové nebo hexagonální) je velikost okolí rovná počtu "řad" neuronů od centrálního neuronu – viz obrázek 2. Váhy každého neuronu naopak definují polohu neuronu v prostoru.

Princip učení SOM

Matici neuronů se postupně předkládají vektory vstupního signálu (x) tak, že se zvlášť porovnává rozdíl příslušných hodnot vektoru vah (koeficientů w) každého neuronu s hodnotami vektoru vstupního signálu. K vyjádření rozdílu se může využít různých algoritmů, ale nejčastěji se dává přednost výpočtu euklidovské vzdálenosti D , tj. součet rozdílů příslušných hodnot:

$$D = (x_1 - w_1)^2 + (x_2 - w_2)^2 + \dots + (x_n - w_n)^2.$$

Výsledkem je tedy počet hodnot D , rovný počtu neuronů ve struktuře (např. 100 hodnot v matici 10 x 10 neuronů). Následně se vybere jediný neuron s nejmenším D a označí se jako tzv. vítěz (winner). Váhy tohoto neuronu totiž nejvíce ze všech odpovídají hodnotám právě předloženého signálu. Při předkládání první učícího vstupního vektoru se jeho hodnoty porovnávají s náhodně vygenerovanými hodnotami vah (koeficientů) jednotlivých neuronů.

Váhy W vítězného neuronu se pak upravují (updatují), aby se co nejvíce přiblížily hodnotám právě předloženého vstupního vektoru (x). Využívá se vzorce:

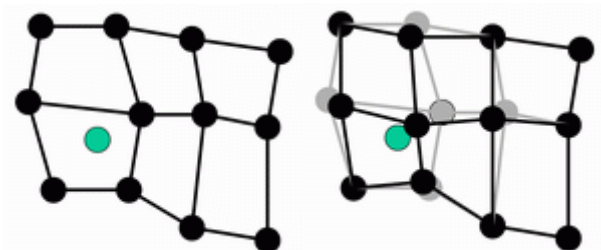
$$W_{i \text{ nové}} = W_{i \text{ staré}} + \alpha (x - W_{i \text{ staré}}),$$

kde α je učící koeficient vyjadřující rychlost učení (může nabývat hodnot 0 až 1, např. $\alpha = 0.6$), W_i je vektor vah (koeficientů) i -tého neuronu $W_i = [W_{i1}, W_{i2}, \dots, W_{in}]$ a x je vstupní učící vektor $x = [x_1, x_2, \dots, x_n]$.

Při opětovném opakování dávky učících vektorů nebo postupným předkládáním dalších nových dávek se učící koeficient obvykle snižuje. Spolu s vítězným neuronem se mění i ty sousední v definovaném okolí R (viz obrázek 2.). Jejich váhy se upravují stejným způsobem jako u vítěze, pouze s tím rozdílem, že koeficient α je nahrazen koeficientem β , přičemž platí $\alpha < \beta$. Při opětovném opakování dávky učících vektorů se může provádět i snižování hodnoty okolí R až na $R = 0$, tzn. adaptuje se pouze vítěz.

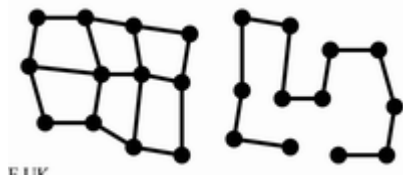
Ve výsledku by se mělo dosáhnout stavu, kdy v maticové struktuře neuronů vznikne několik významných center, tzv. shluky, mezi nimiž se výrazně liší hodnoty vah neuronů. Neurony,

jejichž váhy během učení dosáhly nulových hodnot, se ze struktury mohou vyloučit. Počet shluků by měl být shodný s počtem odlišných vlastností nebo parametrů, které Kohonenova mapa našla v předložených dávkách učících vstupních vektorů. To také znamená, že funkčnost mapy a neuronových sítí obecně, výrazně závisí na složení signálů a informací v učících dávkách.



Obr. 4. Příklad adaptace vah neuronu v mapě rozložení shluků ve dvojdimenzionálním prostoru

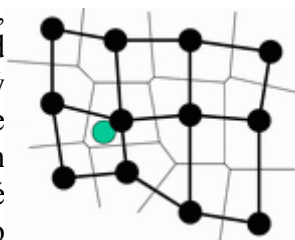
Pro jednodušší kontrolu a přehlednější dohled nad učением mapy se využívá grafického zobrazení shluků, které vyjadřuje prostorové vztahy mezi neurony v prostoru vah. V diagramu jsou váhové vektory (=neurony) zobrazeny jako černé body v dvojdimenzionálním prostoru, které zároveň tvoří centra shluků. Černé čáry představují přímky spojující váhové vektory sousedních neuronů. Na obrázku 4. je ukázaná změna "pozice" neuronu před a po adaptaci vah na vstupní vektor (zelený bod). Na obrázku 5. jsou pak uvedeny příklady diagramů shluků naučené mapy maticové a vektorové struktury uspořádání neuronů. Správné naučení představuje rovnoměrné rozprostření vzájemně propojených bodů po celé ploše, která popisuje vstupní datový prostor, v němž jsou vstupní vektory dat rovnoměrně rozloženy.



Obr. 5. Mapa rozložení shluků ve dvojdimenzionálním prostoru pro maticovou (vlevo) a vektorovou (vpravo) strukturu neuronů

Aktivní funkce naučené sítě SOM

Po naučení SOM (Kohonenovy mapy) se na vstup vždy přikládá analyzovaný neznámý vektor hodnot (zelený bod) podobného druhu, jako byly učící vektory a opět výpočtem podobnosti-vzdálenosti od vektorů vah jednotlivých neuronů (černé body), se vybere ten vítězný neuron, nejvíce podobný hodnotám na vstupu. Ten již představuje určitou definovanou skupinu (shluk) a tím je znám výsledek. Ten představuje zařazení analyzovaného neznámého signálu do některé skupiny či kategorie a tím i jeho pojmenování a nalezení jeho vlastností.



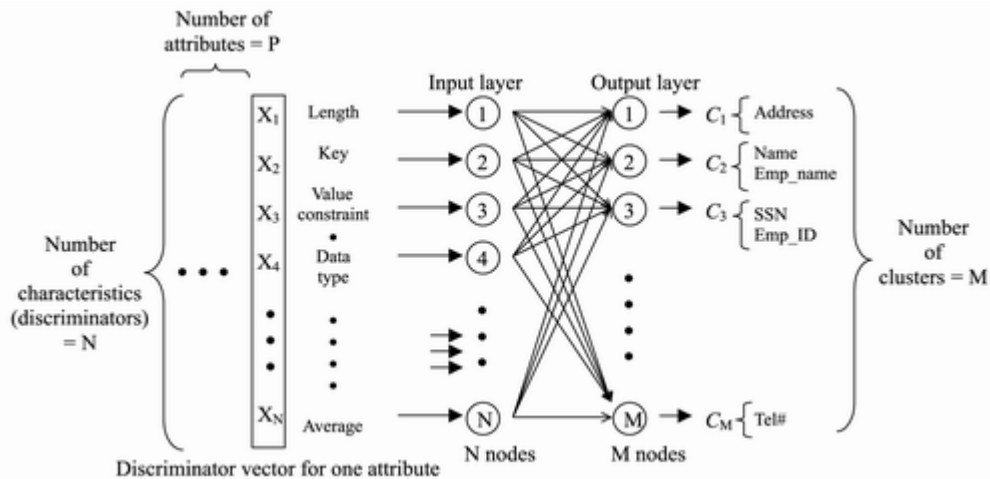
Obr. 6. Zařazení analyzovaného vektoru do shluku vyhrazeného okolím bodu

V principu jsou možné dva režimy:

- **neadaptivní režim** - dojde k zařazení vstupního signálu do některé již v mapě vytvořené shluku - kategorie, což je zároveň i výsledek a na výstupu sítě se může objevit číslo kategorie,

její jméno, nebo nějaký jiný údaj, který neznámý vstupní vektor charakterizuje - viz obrázek 7.

- **adaptační režim** - analýza vstupního signálu je stejná jako v neadaptačním režimu, pouze následně ještě dochází k malé úpravě vah vítězného neuronu. To znamená, že mapa neustále doučuje a je tedy schopna reagovat na dlouhodobě působící postupné změny, podobně jako například náš mozek a naše přizpůsobování chování ke změně okolního prostředí.



Obr. 7. Třídění vstupních vektorů do skupin C_1 až C_M (clusters) dle parametrů v nich obsažených (X_1 až X_N)

Příklady učení

Na následujícím obrázku 8. je uveden počátek praktického jednoduchého příkladu učení Kohonenovy mapy s jednorozměrným uspořádáním ($k=1$) dvou neuronů/shluků ($m=2$), kde každý obsahuje 4 váhy (koeficienty). Pro učení jsou k dispozici 4 vstupní binární vektory (obsahují pouze stavy 1 a 0), které mají tu společnou vlastnost, že vždy dva a dva obsahují hodnotu "1" v prvních nebo posledních dvou pozicích.

Příklad:

Mějme 4 vektory: (1, 1, 0, 0); (0, 0, 0, 1); (1, 0, 0, 0); (0, 0, 1, 1).

Maximální počet shluků je: $m = 2$.

Předpokládejme, že parametr učení je definován vztahy: $\alpha(0) = 0,6$;

$$\alpha(t+1) = 0,5 \alpha(t).$$

Protože jsou k dispozici pouze dva shluky, okolí bodu J (krok 4) je nastaveno tak, že v každém kroku aktualizuje své váhové hodnoty pouze jeden neuron výstupní vrstvy, tj. $R = 0$.

Krok 0. Inicializace váhové matice:

$$\begin{bmatrix} 0,2 & 0,8 \\ 0,6 & 0,4 \\ 0,5 & 0,7 \\ 0,9 & 0,3 \end{bmatrix}$$

Inicializace poloměru sousedství:

$$R = 0.$$

Inicializace parametru učení:

$$\alpha(0) = 0,6.$$

Krok 1. Adaptace:

Krok 2. Pro první vektor (1, 1, 0, 0) opakovat kroky 3-5.

$$\text{Krok 3.} \quad D(1) = (0,2 - 1)^2 + (0,6 - 1)^2 + (0,5 - 0)^2 + (0,9 - 0)^2 = 1,86;$$

$$D(2) = (0,8 - 1)^2 + (0,4 - 1)^2 + (0,7 - 0)^2 + (0,3 - 0)^2 = 0,98.$$

Krok 4. Vstupní vektor je blíže uzlu 2, tak $J = 2$.

Krok 5. Aktualizace váhových hodnot vítězného neuronu:

$$\begin{aligned} w_{j_2}(\text{new}) &= w_{j_2}(\text{old}) + 0,6[x_i - w_{j_2}(\text{old})] \\ &= 0,4w_{j_2}(\text{old}) + 0,6x_i. \end{aligned}$$

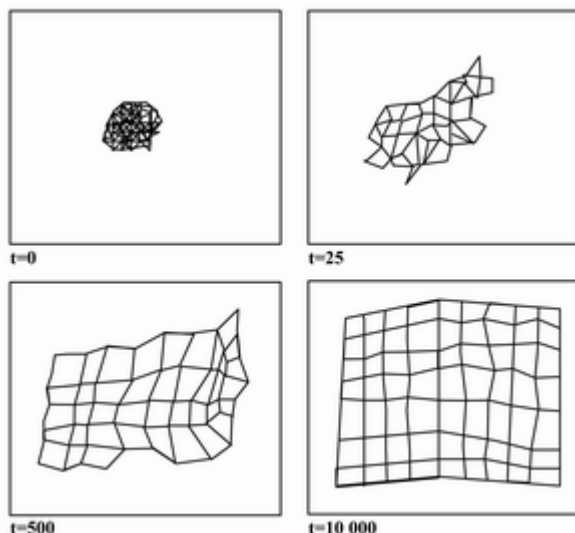
Obr. 8. Úvod do příkladu učení jednoduché Kohonenovy mapy

Pokračování celého postupu učení, kde se prakticky stále za sebou opakují na obrázku uvedené kroky, je možné vzhlednout v česky psaném dokumentu "Neuronové sítě 1" na straně 52 až 54 v sekci [Download & Odkazy](#), ze kterého je úvod příkladu převzat. Výsledkem je adaptace náhodně vygenerovaných vah neuronů před učením, na konečné hodnoty vah po učení:

$$\begin{bmatrix} 0,0 & 1,0 \\ 0,0 & 0,5 \\ 0,5 & 0,0 \\ 1,0 & 0,0 \end{bmatrix}$$

První sloupec, tzn. váhy prvního neuronu, v sobě postihuje oba vstupní vektory (1,1,0,0) a (1,0,0,0), zatímco druhý sloupec, druhý neuron, naopak zbylé vektory (0,0,0,1) a (0,0,1,1). Výsledkem je tedy nalezené společné vlastnosti a rozdíly 4 vstupních vektorů a naučení Kohonenovy mapy. Jedinou nevýhodou, která platí i obecně pro většinu neuronových sítí, je "pomalost" učení, kdy uvedený výsledek byl naučen až po 10 000krát předložených vstupních vektorů. Rychlost lze sice zvýšit volbou vyšších hodnot koeficientů α , příp. β , více blížíci se hodnotě 1, ale to pak může snížit přesnost a kvalitu učení. Je tedy nutné nalézt kompromis.

Na obrázku 9. je pak ukázka vývoje rozložení neuronů ve datovém prostoru během režimu učení pro 0, 25, 500 a 10 000 kroků. Z posledního obrázku vpravo dole je patrné, že neurony byly optimálně rozloženy tak, aby pokryly vstupní datový prostor a byly tak podchyceny a rozlišeny všechny důležité vlastnosti.



Obr. 9. Rozložení neuronů ve vstupním datovém prostoru po různém počtu kroků učení.

Závěr

Kohonenovy mapy, resp. samoorganizující se mapy – SOM, jsou silným a kvalitním nástrojem pro identifikaci neznámých vlastností a parametrů, skrytých v digitalizovaných vzorcích libovolného signálu. V některých aplikacích mohou pracovat jako alternativa k jiným algoritmům, v některých aplikacích jsou již nenahraditelné. S použitím programu Matlab a jeho toolboxu Neural Networks, lze již v celku po krátké době vytvořit například jednoduchý rozpoznávač samohlásek v nahraném řečovém signálu. SOM lze nalézt v programech typu grafity, tedy rozpoznávání rukou psaného textu a jeho přepis na text tištění, nebo se již několik let zkouší na zjišťování nebo předvídání vývoje či vzniku nemocí a poruch mozku, které se vždy projevují jen zprostředkovaně přes plynulost mluvení, schopnost koordinovaného pohybu apod. Využití v řídicích systémech robotů se již samo nabízí. Schopnost se neustálého doučování během činnosti umožňuje přizpůsobení funkce při dlouhodobě trvajících změnách. Tím se již blíží k samotné funkci našeho mozku, který též dokáže přizpůsobovat naše chování změně prostředí. Samoorganizace a samoučení je tedy základem opravdu "inteligentní" umělé inteligence. .

Antonín Vojáček
vojacek@hwg.cz

DOWNLOAD & Odkazy

- Eva Volná: "**Neuronové síť 1**", Ostravská univerzita - Přírodovědecká fakulta, OSTRAVA 2002, http://ki.fpv.ukf.sk/materialy_public/Umela%20inteligencia/neuronovesite.pdf
- Václav Hlaváč: "**Umělé neuronové síť z pohledu rozpoznávání**", prezentace, Centrum strojového vnímání Katedra kybernetiky FEL ČVUT, hlavac@fel.cvut.cz
- "**Dokumentografické Informační Systémy**", Slidy k přednášce DBI010 KSI MFF UK, www.ms.mff.cuni.cz/~kopecky/vyuka/dis/

- Tomáš Skopal: "**Neuronové síte a Information Retrieval**", VŠB-Technical University Ostrava, Department of Computer Science, www.cs.vsb.cz/arg/techreports/neural.pdf
- Prof. Ing. A. Procházka, CSc. & kolektiv: "**Analýza a zpracování dat reprezentující znečištění ovzduší získaných pozemním měřením**", Ústav počítačové a řídicí techniky VŠCHT Praha.