Obsah

1. Soutěživé sítě

1.1 Základní informace

Následující text je součástí učebních textů předmětu Umělá inteligence a je určen hlavně pro studenty Matematické biologie. Tato kapitola se zabývá soutěživými sítěmi, pro které je typická existence soutěživé, vzájemně propojené vrstvy neuronů. Neurony této sítě jsou aktivovány přiložením vstupu a v ustáleném stavu po vzájemné kompetici vítězí zpravidla jeden "nejsilnější" z nich. Podrobněji bude rozebrán model samoorganizující se Kohonenovy mapy.

1.2 Výstupy z učení

Zvládnutí učebního textu umožní studentům:

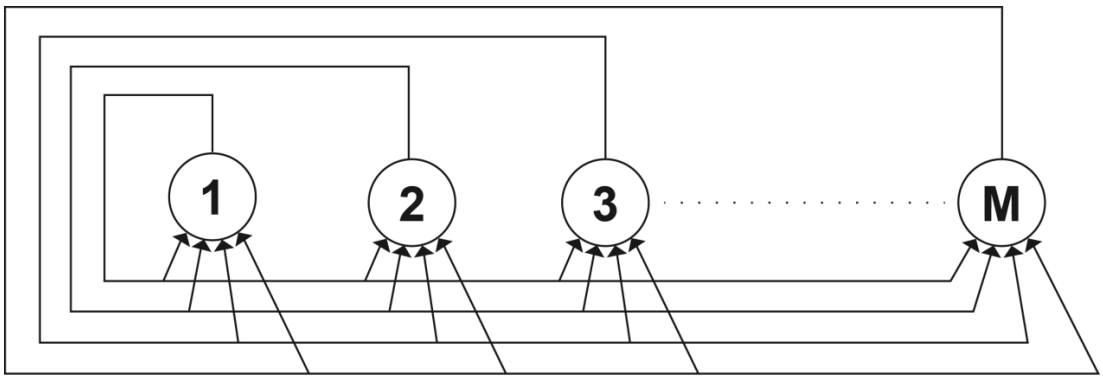
- Popsat princip soutěživých sítí a demonstrovat jej na dynamice sítě MAXNET
- Seznámit se s dynamikou Hammingovy sítě a demonstrovat její aktivní dynamiku na modelovém příkladu
- Pochopit organizační a aktivní dynamiku Kohonenovy samoorganizační mapy
- Formálně definovat adaptační dynamiku Kohonenovy mapy, popsat úskalí Kohonenova učení a provést jeho rozšíření o algoritmy LVQ1, LVQ2, LVQ3

1.3 Jednoduchá soutěživá síť MAXNET

Základní koncept soutěživých sítí si osvětlíme na příkladu nejjednodušší sítě, sítě pro výběr maxima, MAXNET. Stejně jako v případě ostatních soutěživých sítí je základní součástí této sítě tzv. soutěživá vrstva, ve které jsou jednotlivé, vzájemně propojené neurony buzeny vstupním signálem - přiloženým vstupním klasifikovaným vektorem. V další fázi, která probíhá už v jednotlivých diskrétních časových krocích k, je vždy rekurentně a synchronně aktualizován stav celé vrstvy. V ustáleném stavu sítě je nakonec aktivní (má nenulový výstup) pouze jeden neuron z této kompetiční vrstvy. Tento vítězný neuron představuje reprezentanta shluku přiřazeného k předloženému vstupu. V případě sítě MAXNET je vítězem vždy neuron s nejvyšší počáteční hodnotou vstupu, vzhledem k rovnosti vah mezi neurony soutěživé vrstvy tedy neuron s nejvyšší vnitřním potenciálem v kroku 0. Tato soutěživá vrstva tedy odpovídá bloku pro výběr maxima.

1.3.1 Organizační dynamika

MAXNET se skládá ze soutěživé vrstvy, ve které se nachází M klasických binárních neuronů, jak byly zavedeny v kapitole týkající se perceptronů. Neurony jsou propojeny v této vrstvě každý s každým.



Obr. 1 *Soutěživá vrstva sítě MAXNET.*

Prahy všech neuronů jsou nulové

$$\vartheta_i = 0 \; pro \; \forall i \; ,$$

(1)

Hodnota váhy zpětnovazební vazby každého neuronu je

$$w_{ij} = 1 \, pro \; i = j$$

(2)

a vazby mezi neurony jsou nastaveny všechny na stejnou hodnotu (-ε)

$$w_{ij} = -\varepsilon \; pro \; i \neq j \; .$$

(3)

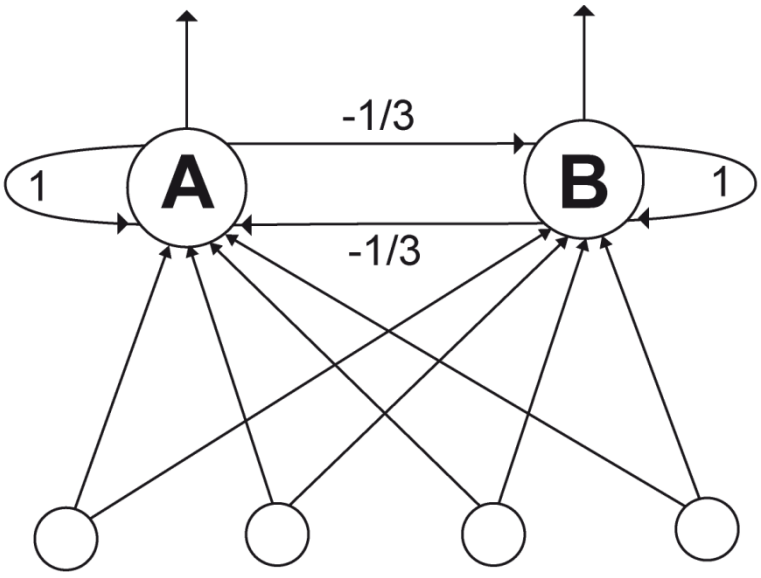
Hodnotu ε volíme v intervalu od nuly do 1/M, kde M je počet neuronů soutěživé vrstvy

$$0 < \varepsilon < \frac{1}{M} \; .$$

(4)

Tyto váhy zůstávají konstantní. Počet neuronů soutěživé vrstvy volíme obvykle dle předpokládaného počtu shluků, do kterých chceme klasifikovat.

Pro přivedení signálu (vstupu) do soutěživé vrstvy je tato síť doplněna o vstupní vrstvu úplně propojenou se soutěživou vrstvou váhovanými vazbami, které ve fázi učení podléhají adaptaci.



Obr. 2 *Jednoduchá soutěživá síť MAXNET se dvěma neurony v kompetiční vrstvě.*

1.3.2 Adaptační dynamika

Váhy soutěživé vrstvy zůstávají po celou dobu adaptace i aktivní dynamiky konstantní. Adaptaci podléhají pouze váhy, kterými jsou ohodnoceny vazby ze vstupní do kompetiční vrstvy. Učení probíhá na základě vstupů z trénovací množiny. Není zde obvykle nutné, stejně jako v případě Hopfieldovy sítě určovat výstup, který má síť poskytnout na předkládané vstupy.

M = {$\mathbf{x}^1$, $\mathbf{x}^2$, …. $\mathbf{x}^{pmax}$}

(5)

Typické kroky soutěživého učení platné i v síti MAXNET jsou:

- Nastavení pevných hodnot vah v kompetiční vrstvě
- Nastavení hodnot vah vstupní vrstvy na náhodné hodnoty, nebo lze tyto hodnoty nastavit na základě apriorní znalosti o předpokládaných shlucích (typický reprezentant). Jednotlivou váhu mezi i-tým prvkem vstupního vektoru a j-tým neuronem soutěživé vrstvy zpravidla volíme v intervalu (0,1) tak, aby $\sum_{\forall i} w_{ij} = 1$.
- Předložení vstupního vektoru **x** síti
- Výpočet primárních výstupů neuronů soutěživé vrstvy
- Kompetice neuronů v jednotlivých krocích v soutěživé vrstvě (vstup odpojen)
- Určení vítězného neuronu, reprezentanta s nenulovým výstupem
- Úprava vah vítězného neuronu $w_{ij} = w_{ij} + \Delta w_{ij}$ o Δw_{ij} (6) a jejich normalizace na $\sum_{\forall i} w_{ij} = 1$.
  $$\Delta w_{ij} = \eta x_i \frac{1}{M} - \eta w_{ij}, \forall i$$
  (7)
- Opakování postupu pro další vstupní vektor **x**. Koeficient η se obvykle v průběhu učení sítě postupně snižuje. Tento postup urychluje konvergenci sítě.

Mějme síť z obrázku 2. Předpokládejme počáteční nastavení vah neuronů A a B $w_A = (0{,}2; 0{,}2; 0{,}3; 0{,}3)$ a $w_B = (0{,}2; 0{,}3; 0{,}3; 0{,}2)$. Například pro vektor **x** = (1; 1; 0; 0) bude výstup neuronu A před zahájením laterální inhibice roven y_A = 0,4 a vektoru B y_B = 0,5. Po ukončení kompetice bude vítězným neuron B, provedeme tedy úpravu vektoru jeho vah w_B dle vztahu (7). Zvolme η =1/2. Počet neuronů M = 2.

Pak budou váhy po předložení prvního vstupu rovny

$$w_{1B} = 0{,}2 + 0{,}5 * 0{,}5 - 0{,}5 * 0{,}2 = 0{,}35$$

$$w_{2B} = 0{,}3 + 0{,}5 * 0{,}5 - 0{,}5 * 0{,}3 = 0{,}4$$

$$w_{3B} = 0{,}3 + 0 - 0{,}5 * 0{,}3 = 0{,}15$$

$$w_{4B} = 0{,}2 + 0 - 0{,}5 * 0{,}2 = 0{,}1$$

Po takto upravených vahách by síť klasifikovala pro vstupy (0;0;0;1), (0;0;1;0), (0;0;1;1), (0;1;1;1), (1;0;0;1), (1;0;1;1) do shluku reprezentovaného neuronem A, pro vstupy (0;1;0;0), (0;1;1;0), (1;0;0;0), (1;1;0;0), (1;1;0;1), (1;1;1;0) pak B. Ostatní vektory nebudou zařazeny do žádné z tříd.

Podobně bychom v adaptaci vah pokračovali i pro další předkládané vstupy. Důležitou vlastností je, že v uvedeném algoritmu neurčujeme explicitně třídu, do které má být ten který vstup zařazen, není a nemůže být tedy vyhodnocována žádná chyba klasifikace sítě. Síť vytváří shluky samostatně, pouze na základě předkládaných vstupů. Jedná se tedy o učení bez učitele.

1.3.3 Aktivní dynamika

V procesu aktivní dynamiky jsou síti předkládány jednotlivé vstupy, vstupní vektory. Síť odpoví po realizaci výpočtu aktivací jediného neuronu ze soutěživé vrstvy, který představuje reprezentanta shluku. Vstup je tak zařazen do shluku představovaného vybraným reprezentantem. V případě sítě MAXNET se jedná o neuron, který byl po přiložení vstupu nejvíce buzen, tedy ten neuron s maximální hodnotou aktivační funkce. Jinak řečeno, je vybrán ten neuron, jehož váhy maximálně korelují s předkládaným vstupním vektorem.

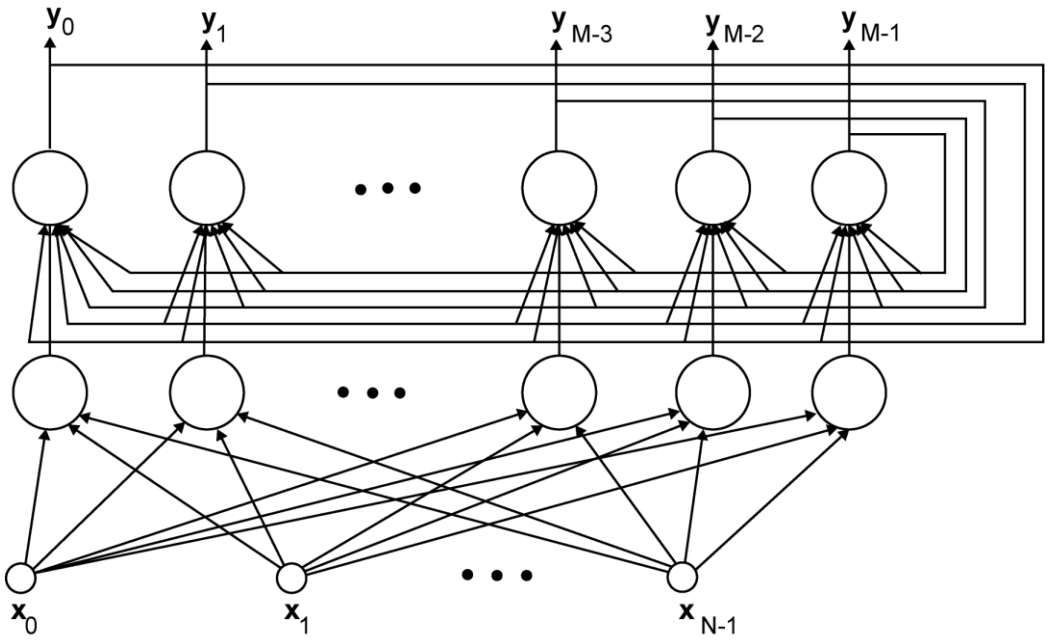
Mějme opět síť z obrázku 2 a vektory vstupních vah do neuronů A a B $w_A = (0{,}2; 0{,}2; 0{,}3; 0{,}3)$ a $w_B = (0{,}2; 0{,}3; 0{,}3; 0{,}2)$.

Takto nastavená síť zvolí pro vstupy (0;0;0;1), (0;0;1;1), (1;0;0;1), (1;0;1;1) reprezentanta A, pro vstupy (0;1;0;0), (1;1;1;0), (1;1;0;0), (1;1;1;0) pak reprezentanta B. Ostatní vektory nebudou zařazeny do žádné z tříd, neboť nezvítězí žádný z neuronů (oba jsou buzeny stejně a soutěživá vrstva jejich vliv vzájemně vyruší).

Je třeba poznamenat, že aktivní a adaptační dynamika nemusejí být v kategorii samorganizujících se sítí se vzájemnými vazbami vždy striktně oddělenými fázemi, ale vzhledem k samoučícímu se algoritmu sítí může adaptace probíhat teoreticky donekonečna, na základě stále nových předkládaných vstupů.

1.4 Hammingova síť

1.4.1 Organizační dynamika



Obr. 3 *Hammingova síť.*

Organizanizační dynamika Hammingovy sítě je zřejmá z obrázku. Skládá se ze tří vrstev, kdy první vrstva je vstupní a přenáší N vstupů z vektoru **x** úplným propojením na druhou, Hammingovu vrstvu s M neurony. Tato vrstva, respektive každý neuron v ní, počítá doplněk Hammingovy vzdálenosti každého neuronu (vektoru jeho vah) od předkládaného vstupu. Výstupy druhé vrstvy jsou přímo přenášeny (s vahou = 1) do poslední, soutěživé vrstvy vybírající maximum z M neuronů.

1.4.2 Adaptační dynamika

Dále předpokládejme N rozměrné bipolární vstupní vektory, tedy vektory s N prvky nabývajícími hodnot 1 nebo (-1). Trénovací množina obsahuje M vzorových bipolárních vektorů **x**.

M = {$\mathbf{x}^1$, $\mathbf{x}^2$, …. $\mathbf{X}^M$}

(8)

Parametry Hammingovy vrstvy, tedy váhy a prahy i-tého neuronu, jsou nastaveny přímým výpočtem založeným na reprezentantech z trénovací množiny.

Prahy i-tého neuronu jsou nastavena na

$$\vartheta_i = \frac{N}{2} \; pro \forall i \; .$$

(9)

Vektor vah $\mathbf{w}_i$ i-tého neuronu je nastaven dle hodnot i-tého reprezentanta **x** dle vztahu

$$\vec{w}_i = \frac{1}{2}\vec{x}_i \; pro \forall i$$

(10)

Každý neuron Hammingovy vrstvy tedy odpovídá jednomu vstupu z množiny vzorů, představuje jeho reprezentanta.

Soutěživá vrstva je nastavena dle pravidel popsaných v kapitole věnované MAXNET.

1.4.3 Aktivní dynamika

Aktivní dynamika Hammingovy sítě je zřejmá. Po předložení vstupního vektoru vypočítá Hammingova vrstva doplněk Hammingovy vzdálenosti vstupního vektoru od všech neuronů, reprezentantů. Nejvyšší hodnotu výstupu bude mít ten neuron, který je vstupu nejblíže ve smyslu Hammingovy vzdálenosti. Poslední vrstva pak zajistí, že tento neuron bude po kompetici zvolen vítězným reprezentantem.

Mějme například množinu vzorů

M = {x^1(1;-1;-1;1;1), x^2(-1;1;-1;1;-1), x^3(1;-1;1;-1;1)}

Předpokládejme vstupní vektor **x** = (1;1;1;-1;-1).

Váhy a prahy neuronů Hammingovy vrstvy jsou nastaveny dle vztahů (9), (10).Pro odezvu i-tého neuronu y_i neuronů Hammingovy vrstvy můžeme psát

y_1 = ((x^1/2)*x)-5/2 = ½(x^1*x-5) = ½(-3-5) = -4

y_2 = ((x^2/2)*x)-5/2 = ½(x^2*x-5) = ½(-1-5) = -3

y_3 = ((x^3/2)*x)-5/2 = ½(x^3*x-5) = ½(1-5) = -2

Po boji tedy zvítězí třetí neuron s i = 3, který je vektoru **x** nejpodobnější ve smyslu Hammingovy vzdálenosti.

Hammingova síť vlastně představuje model, který ve svých parametrech ukládá konkrétní množinu reprezentantů a vysvětlení její činnosti interpretace výstupů je například na rozdíl od vícevrstvých dopředných neuronových sítí zřejmá. Parametry sítě snadno nastavíme výpočtem, bez nutnosti iterací. Tyto výhodné vlastnosti jsou na druhou stranu vykoupeny menší schopností sítě zobecňovat a pro praktické implementace pracující s vícerozměrnými vstupními vektory by reálně pracující Hammingova síť vedla na velmi rozsáhlou množinu vzorů a tedy velký počet neuronů v síti.

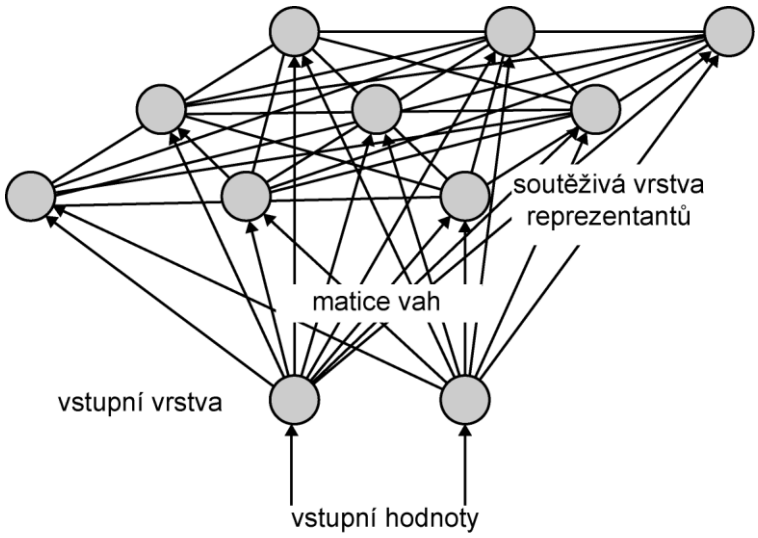
1.5 Samoorganizující se mapy

1.5.1 Jednoduchá samoorganizační mapa - organizační a aktivní dynamika

V případě samoorganuzijících se map působí neuronová síť jen pro představu, obvykle je reálně implementována pomocí jednodušších datových struktur, jako jsou pole (vektory) či matice.

Organizační struktura jednoduché samoorganizující mapy je shodná se sítí, která byla popsána v odstavci věnujícímu se MAXNET. Jedná se o dvouvrstvou síť, kde je vstupní vrstva úplně propojena

s kompetiční vrstvou. Síť obvykle realizuje zobrazení z prostoru reálných čísel do prostoru binárních hodnot.



Obr. 4 *Obecná topologie samoorganizující se mapy.*

Vektory vah mezi neurony vstupní a kompetiční vrstvy představují podobně jako v případě Hammingovy sítě pozice reprezentantů ve vstupním prostoru. Při aktivní dynamice tedy síť počítá podobně jako pro Hammingovu síť vzdálenost přiloženého vstupu od pozice všech reprezentantů daných vahami neuronů. Kompetiční vrstva poté zajistí opět vítězství nejsilnějšího.

1.5.2 Adaptační dynamika jednoduché samoorganizační mapy– kohonenovo učení

Základní metodou adaptační dynamiky je učení bez učitele. Síť tedy organizuje své váhy sama, pouze na základě vstupů předkládaných z trénovací množiny.

$M = \{\mathbf{x}^1, \mathbf{x}^2, ... \mathbf{x}^{pmax}\}$

(11)

Síť během své adaptace vytváří reprezentanty pro shluky nalezené sítí v trénovací množině, do kterých následně při aktivní dynamice síť zařazuje předkládané vstupy.

Proces adaptace probíhá metodou kohonenova učení. Při tomto procesu adaptace, jsou nejprve nastaveny váhy na náhodné hodnoty. Následně je na vstup v k-tém kroku přiložen jeden ze vstupů z trénovací množiny. Síť určí vítězný, i-tý neuron. Váhy vítězného neuronu jsou modifikovány dle

$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta\,(\mathbf{x}(k) - \mathbf{w}_i(k))$.

, kde η představuje parametr z intervalu (0,1>.

(12)

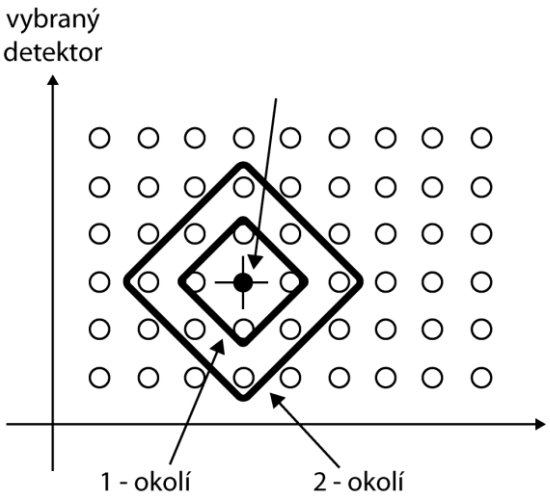
Parametr η určuje míru posunu vektoru **w** směrem k vektoru **x** a určuje rychlost učení. Na počátku učení je zpravidla jeho hodnota blízká 1 a v procesu učení je postupně snižována k 0. Myšlenka

adaptace je taková, že modifikací vah posouváme pozici vítězného reprezentanta ve vstupním prostoru blíže k aktuálně klasifikovanému vstupu. Takto procházíme postupně celou trénovací množinu až do chvíle, kdy již dochází k minimální změně pozic reprezentantů.

Tento způsob adaptace má však i své podstatné nevýhody, je velmi citlivý na vhodnou počáteční inicializaci vah neuronů tak, aby pokrývala tu část vstupního prostoru, kde předpokládáme klasifikované vstupy a tedy hledané shluky. Představme si situaci, kdy chceme provést identifikaci shluků v klasickém euklidovském dvojrozměrném prostoru. Předpokládejme, že po počáteční náhodné inicializaci vah neuronů jsou tito počáteční reprezentanti rozmístěni nerovnoměrně a jsou soustředěni pouze do jedné oblasti vstupního prostoru. Dále předpokládejme, že trénovací množina, ve které chceme identifikovat shluky, obsahuje vektory ze samozřejmě stejného euklidovského prostoru, ale umístěné v jiné oblasti, než jsou inicializovány váhy. Oblast reprezentantů daných počátečním nastavením vah a oblast vstupů z trénovací množiny jsou tak od sebe v prostoru vzdálené. Po předložení prvního vstupu z trénovací množiny dojde k tomu, že zvítězí jeden, nejbližší z reprezentantů. Jeho váhy jsou upraveny dle (12) a tento reprezentant je tak přesunut mnohem blíže k oblasti vstupů z trénovací množiny. Co se ale bude dít po předložení dalšího vstupu z trénovací množiny? Jelikož jsou si vstupní vektory relativně blízké, zvítězí opět ten stejný, v minulém kroku korigovaný reprezentant. Takto tedy algoritmus postupuje dále a cyklicky upravuje pouze jednoho jediného reprezentanta. Ostatní zůstávají beze změny. Tento základní algoritmus tedy v praxi často selhává. S ilustrovaným problémem se umí vypořádat rošířená samoorganizační neuronová síť nazývaná Kohonenova mapa.

1.5.3 Kohonenova samoorganizační mapa – organizační a aktivní dynamika

Organizační dynamika Kohonenovy mapy je zcela stejná, jako v případě jednoduché samoorganizační mapy z obrázku 4. Rozdíl se nachází v kompetiční vrstvě. Její organizace je opět naprosto shodná s předcházejícími případy soutěživých sítí, ale neurony (detektory) jsou navíc logicky uspořádány do geometrické struktury. Nejčastěji je to přímka či mřížka, ale může být v zásadě libovolná. Obvykle má její rozměr výrazně nižší dimenzi, než originální prostor vstupů.



Obr. 5 *Definice okolí detektoru dle [2].*

Geometrické uspořádání detektorů umožňuje definovat pojem „1 - n okolí“ každého z detektorů. V procesu aktivní dynamiky se toto geometrické uspořádání neuplatňuje, aktivní dynamika po

předložení vstupu je zcela shodná s jednoduchou samoorganizační mapou definovanou v předchozí kapitole.

1.5.4 Kohonenova samoorganizační mapa – adaptační dynamika

Odlišnost Kohonenovy mapy od jednoduché samoorganizační mapy je ve fázi adaptační dynamiky, která bere v úvahu geometrické uspořádání detektorů. Adaptace je opět založena na předkládání vstupů z trénovací množiny. Po předložení vektoru je opět určen vítězný neuron, stejně jako v předcházejícím případě. Nicméně poté nejsou upravovány jen váhy vítězného neuronu, ale dochází i k modifikaci vah, které se nacházejí v geometrickém okolí vítěze dle vztahu (12).

Obvyklý postup je takový, že v počáteční fáze adaptace je upravované široké okolí vítěze, zpravidla zahrnující celou síť. Poté modifikované okolí postupně klesá. Obvykle s klesajícím okolím klesá i hodnota parametru η, která určuje rychlost učení. Tento postup přitáhne reprezentanty do míst, kde se nachází nevyšší koncentrace vstupů z trénovací množiny. Dochází tak k minimalizaci chyby klasifikace přes trénovací množinu ve smyslu minimalizace střední kvadratické odchylky vstupních vektorů od reprezentantů.

Parametr η nemusí být konstantní ani z pohledu vzdálenosti od vítězného reprezentanta a jeho hodnota může se vzdáleností od vítěze klesat, například dle gaussovy funkce se středem ve vítězném neuronu. Uvedený postup vede také k tomu, že reprezentanti, kteří jsou si blízcí ve smyslu topologického uspořádání například na mřížce, jsou si po adaptaci blízcí i v originálním prostoru vstupů.

1.5.5 Kohonenova samoorganizační mapa – učením s učitelem

Uvedené postupy adaptace Kohonenovy mapy probíhaly bez učitele. Rozdělení vstupů do jednotlivých shluků bylo dáno pouze adaptační dynamikou sítě. Nicméně často chceme dát nalezeným shlukům určitý význam, nalezené shluky pojmenovat, tedy klasifikovat předkládané vstupy do předem známého počtu pojmenovaných kategorií. K tomu slouží algoritmy učení Kohonenovy mapy s učitelem, nazývané jako LVQ1, LVQ2 a LVQ3 (od Linear Vector Quantization), dle [1].

První fáze učení je pro všechny tři algoritmy shodná a spočívá v aplikaci standardního algoritmu adaptace Kohonenovy mapy, jak byl představen v předchozí kapitole. V této fázi jsou z trénovací množiny použity pouze vstupní vektory **x**, příslušné výstupní kategorie **y** nejsou v této fázi použity.

$$M = \{[\mathbf{x}^1, y_d^1], [\mathbf{x}^2, y_d^2] \ldots . [\mathbf{x}^{pmax}, y_d^{pmax}]\} \quad (13)$$

Po naučení sítě zjistíme, jak síť klasifikuje přes celou trénovací množinu. Pro každý vstupní vektor **x** z trénovací množiny určíme reprezentanta, který je danému vstupu **x** přiřazen. Současně máme k dispozici pro každý ze vstupů korektní kategorii y_d. Pro každého reprezentanta tak můžeme určit množinu pojmenovaných kategorií y_d a četnost zařazení vstupů **x** do každé z nich. Na základě údaje o četnosti přiřazení kategorií jednotlivým reprezentantům pojmenujeme reprezentanty tou kategorií, kterou reprezentovali přes trénovací množinu nejčastěji.

Posledním krokem je korekce vah a tedy pozice reprezentantů jedním se zmíněných LVQ algoritmů.

Algoritmus **LVQ1** pracuje tak, že síti předkládáme vstup z trénovací množiny a určíme vítěze. Poté vyhodnotíme, zda je klasifikace předloženého vstupu korektní, je tedy určen správný shluk. Následně upravujeme váhy pouze tohoto reprezentanta a to dle vztahu shodného s (12) pro korektně zařazený vstupní vektor

$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta\,(\mathbf{x}(k) - \mathbf{w}_i(k))$

a

$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) - \eta\,(\mathbf{x}(k) - \mathbf{w}_i(k))$

(14)

pro chybně klasifikovaný vstup.

V případě korektní klasifikace tak dochází tedy k přiblížení reprezentanta ke klasifikovanému vzoru, v opačném případě pak naopak dochází k oddálení reprezentanta od chybně klasifikovaného vzoru. LVQ1 provádí pouze korekci pozic nalezených reprezentantů, hodnota η by měla být velmi malá, dle [1] mezi 0,02 – 0,01 a měla by se postupně blížit k nule. Uvedený postup vede k vyprázdnění hraničních oblastí mezi sousedními shluky. Nalezená hranice aproximuje bayesovskou rozhodovací hranici a leží uprostřed spojnice mezi reprezentanty různých tříd.

Algoritmus **LVQ2** se používá pouze pro několik (tisíc) iterací, protože bylo experimentálně zjištěno, že zpočátku zlepšuje umístění rozhodovací hranice, nicméně po velkém počtu iterací začíná klasifikaci zhoršovat. Opět přiložíme vstup a určíme tentokrát dva vstupu nejbližší neurony - reprezentanty. Musí platit, že jeden z nich klasifikuje korektně a druhý chybně. Dále musí platit, že vstup není příliš blízký žádnému ze dvou reprezentantů, nachází se tedy "ve středu mezi nimi" v okně, jehož obvyklá šíře je cca 10 -30% vzdálenosti neuronů. Při splnění těchto podmínek následně upravíme oba neurony dle obvyklých vztahů, kdy dle korektnosti klasifikace pak přiblížíme, či naopak oddálíme daný z dvojice neuronů od přeloženého vstupu (12), (14).

Algoritmus **LVQ3** je stabilním rozšířením LVQ2. Pracuje stejně jako algoritmus LVQ2, tedy koriguje pozice dvou nejbližších reprezentantů dle LVQ2, (12), (14). Navíc ale ještě provádí úpravu vah korektně klasifikujícího reprezentanta z vybrané dvojice dle

$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \varepsilon\eta\,(\mathbf{x}(k) - \mathbf{w}_i(k))$.

(15)

Posouvá jej tedy blíže ke korektně klasifikovanému vstupu. Parametr ε je po dobu adaptace konstantní a je volen v rozmezí 0,1 – 0,5 [1].

1.6 Seznam použité literatury

[1] Šíma, J., Neruda, R.: Teoretické otázky neuronových sítí, MATFYZPRESS, 1996, ISBN 80-85863-18-9.

[2] Jan, J.: Číslicová filtrace, analýza a restaurace signálů. VUT v Brně, VUTIUM, 2002. ISBN 80-214-1558-4.

[3] Volná, E.: Neuronové sítě 1. Skripta Ostravská universita v Ostravě, Ostrava, 2008.

[4] V. Kvasnička, L. Beňušková, J. Pospíchal, I. Farkaš, P. Tiňo, and A. Kráľ. Úvod do teórie neurónových sietí, IRIS, Bratislava, 1997, ISBN 80-88778-30-1.