

6. prednáška

Holografická redukovaná reprezentácia v umelej inteligencii a kognitívnej vede

6.1 Úvodné poznámky

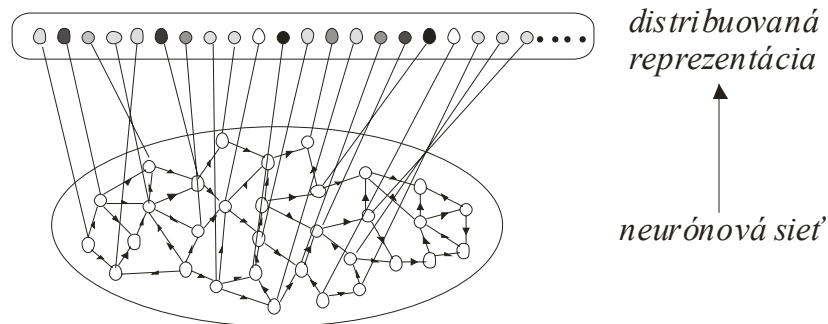
Moderný pohľad na vzťah medzi mozgom a myslou vychádza z neurovednej paradigmy [3], podľa ktorej architektúra mozgu je určená spojmi medzi neurónmi, ich inhibičným, alebo excitačným charakterom a taktiež aj ich intenzitou. Ľudský mozog vykazuje neobyčajnú plasticitu, v priebehu učenia neustále vznikajú (ale taktiež aj zanikajú) synaptické spoje. Možno konštatovať, že *schopnosť mozgu vykonávať nielen kognitívne aktivity, ale byť aj pamäťou a aj riadiacim centrom pre našu motoriku, je plne zakódovaná jeho architektúrou*. Môžeme konštatovať, že predstava o ľudskom mozgu ako o počítači sa musí formulovať tak, že mozog je *paralelný distribuovaný počítač* (obsahujúci mnoho miliárd neurónov, elementárnych procesorov, ktoré sú medzi sebou poprepájané do zložitej neurónovej siete). Program v tomto paralelnom počítači je priamo zabudovaný do architektúry neurónovej siete, t. j. ľudský mozog je jednocelový paralelný počítač reprezentovaný neurónovou sieťou, ktorý nie je možné preprogramovať bez zmeny jeho architektúry.

Z vyššie uvedených všeobecných úvah vyplýva, že myseľ s mozgom tvoria jeden integrálny celok, ktorý je charakterizovaný *komplementárnym dualizmom*. Myseľ je v tomto prístupe chápaná ako program vykonávaný mozgom, pričom tento program je špecifikovaný architektúrou distribuovanej neurónovej siete reprezentujúcej mozog. Mozog a myseľ tvoria dva rôzne pohľady na ten istý objektu:

- (1) Keď hovoríme o mozgu, myslíme tým „hardwarovú“ štruktúru, biologicky realizovanú neurónmi a ich synaptickými spojmi (formálne reprezentovanú neurónovou sieťou), v opačnom prípade,
- (2) keď hovoríme o myšli, myslíme tým kognitívne a iné podobné aktivity mozgu, ktoré sú vykonávané na symbolickej úrovni, kde prebieha transformácia symbolickej informácie na základe (jednoduchých) pravidiel.

Komplementárny dualizmus medzi mozgom a myslou spôsobuje určité ťažkosti pri interpretácii kognitívnych aktivít mysle. Čisto neurálny prístup k interpretácii kognitívnych aktivít mysle sa upriamuje na hľadanie neurálnych korelátov medzi aktivitami neurónov a kognitívnymi aktivitami (konekcionizmus). Použitie neurálnej paradigmy k interpretácii symbolických kognitívnych aktivít má „vedľajší“ efekt v tom, že tieto sa nám „rozpušťajú“ v ich mikroskopickom popise, symboly sa nám akoby strácajú v detailnom popise aktivít neurónov, intenzít synaptických spojov a pod. V opačnom prípade, absolutizovanie symbolickej paradigmy pri interpretácii kognitívnych aktivít mysli (kognitivizmus), ignorovanie skutočností, že myseľ je pevne ukotvená v mozgu, vedie ku koncepcnej sterilite, k snahe vysvetľovať kognitívne procesy ľudskej mysle len pomocou fenomenologických

termínov odvodených od koncepcie symbolu. Navrhujú sa umelé symbolické konštrukty (metódy, algoritmy a pod.), pre ktoré obvykle neexistuje žiadna experimentálna podpora v neurovede. Cieľom tejto práce je poukázať na alternatívny prístup, ktorý prekonáva „priepasť“ medzi konekcionistickým a kognitivistickým prístupom k popisu a interpretácii kognitívnych aktivít ľudského mozgu [10-13]. Ukážeme, že použitie distribuovanej reprezentácie umožňuje integrovať konekcionizmus a kognitivismus, kde mentálne reprezentácie (symboly) sú špecifikované pomocou distribuovaných obrazcov neurálnych aktivít, pričom nad týmito distribuovanými obrazcami môžeme zaviesť formálne algebraické operácie, ktoré umožňujú matematicky modelovať nielen kognitívne operácie, ale aj simulovať procesy prebiehajúci pri ukladaní a vyberaní informácie uloženej v pamäti.



Obrázok 1. Znáznornenie prechodu od neurónovej siete k distribuovanej reprezentácii. Stav neurónovej siete v čase t je zadaný aktivitami jednotlivých neurónov, ktoré sú určené aktivitami v predchádzajúcom čase $t-1$ a váhovými koeficientmi jednotlivých orientovaných spojov. V určitej abstrakcie, môžeme tieto aktivity usporiadať do veľkého jednorozmerného poľa (vektora) útvaru, ktorého komponenty sú reálne čísla (ich veľkosť je určená stupňom šede príslušnej zložky – neurónu). V distribuovanej reprezentácii je ignorovaná architektúra neurónovej siete, t.j. dve distribuované reprezentácie musia byť chápané ako úplne nezávislé bez vzájomných súvislostí, ich kauzálna súvislosť pochádzajúca z neurónovej siete je teraz úplne ignorovaná. V distribuovanej reprezentácii zavádzame nové unárne a binárne operácie, pomocou ktorých vznikajú z pôvodných nové distribuované reprezentácie.

Našu pozornosť usmerníme na netradičný štýl uskutočňovania výpočtov pomocou distribuovaných obrazcov, ktorý sa diametrálne líši od klasických numerických a symbolických výpočtov a je vhodným modelovým prostriedkom pre pochopenie globálnych vlastností neurónových sietí. Ukážeme, že takéto „neuropočítanie“ je založené na rozsiahlych náhodne vytvorených obrazcoch (reprezentovaných mnohorozmernými vektormi s náhodnými zložkami), pozri obr. 1. Tento prístup, ktorého základné prístupy boli formulované už koncom 60-tich rokov [2,4,5,9,14], bol zavŕšený sériou prác Tony Plateom [7-9] o „*holografickej redukovanej reprezentácii*“ (HRR). Ukážeme, aké typy výpočtov sú týmto prístupom uskutočniteľné a či nám umožnia pochopiť procesy prebiehajúce v mozgu pri kognitívnych aktivitách. Náš príspevkom k rozvoju HRR spočíva v jeho aplikácia k modelovaniu kognitívnych procesov usudzovania pomocou pravidiel modus ponens a modus tollens. Kanerva [16-18] v polovici 90-tich rokov navrhol určitú alternatívu k HRR, ktorá je založená na náhodne generovaných binárnych vektoroch.

2. Matematická formulácia holografickej reprezentácie

Cieľom tejto kapitoly je prezentácia základných vlastností holografickej reprezentácie, ktorá bola vypracovaná A. Plateom [7-9]. Základný pojem je *konceptuálny vektor*, ktorý je reprezentovaný n -rozmerným vektorom

$$\mathbf{a} \in R^n \Rightarrow \mathbf{a} = (a_0, a_2, \dots, a_{n-1}) \quad (1)$$

pričom jeho komponenty sú náhodné čísla so štandardným normálnym rozdelením

$$a_i = N(0,1/n) \quad \forall i \in \{0,1,\dots,n-1\} \quad (2)$$

kde $N(0,1/n)$ je náhodné číslo so stredom v 0 a so strednou odchýlkou $1/n$.

Nad konceptuálnymi vektormi definujeme binárnu operáciu „konvolúciu“, ktorá dvojici vektorov priradí tretí vektor, $\otimes : R^n \times R^n \rightarrow R^n$, alebo

$$\mathbf{c} = \mathbf{a} \otimes \mathbf{b} \quad (3)$$

Zložky výsledného vektora $\mathbf{c} = (c_0, c_1, \dots, c_{n-1})$ sú určené vzťahom

$$c_i = \sum_{j=0}^{n-1} a_j b_{[i-j]} \quad (i = 0,1,\dots,n-1) \quad (4)$$

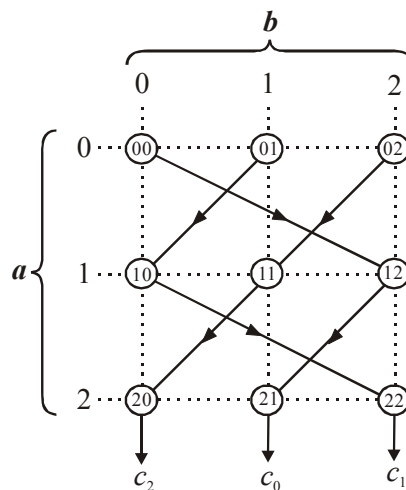
kde index v hranatej zátvorke, $[k]$, pomocou operácie *modulo n* takto¹

$$k' = k \text{ mod } n \quad (5a)$$

$$[k] = \begin{cases} k' & (\text{if } k' \geq 0) \\ n + k' & (\text{if } k' < 0) \end{cases} \quad (5b)$$

Konvolúcia dvoch vektorov \mathbf{a} a \mathbf{b} pre $n=3$ má tento tvar (pozri obrázok 2)

$$\begin{aligned} c_0 &= a_0 b_0 + a_1 b_2 + a_2 b_1 \\ c_1 &= a_0 b_1 + a_1 b_0 + a_2 b_2 \\ c_2 &= a_0 b_2 + a_1 b_1 + a_2 b_0 \end{aligned} \quad (6)$$



Obrázok 2. Znázornenie konvolúcie dvoch vektorov \mathbf{a} a \mathbf{b} pre $n=3$ (pozri (6)).

Konvolúcia vyhovuje týmto vlastnostiam:

- (1) komutatívnosť, $\mathbf{a} \otimes \mathbf{b} = \mathbf{b} \otimes \mathbf{a}$
- (2) asociatívnosť, $(\mathbf{a} \otimes \mathbf{b}) \otimes \mathbf{c} = \mathbf{a} \otimes (\mathbf{b} \otimes \mathbf{c})$
- (3) distributívnosť, $\mathbf{a} \otimes (\alpha \mathbf{b} + \beta \mathbf{c}) = \alpha (\mathbf{a} \otimes \mathbf{b}) + \beta (\mathbf{a} \otimes \mathbf{c})$
- (4) existencia jednotkového vektora, $\mathbf{1} \otimes \mathbf{a} = \mathbf{a}$ ($\mathbf{1} = (1,0,\dots,0)$)

¹ Štandardná definícia aritmetickej operácie *k modulo n* je určená ako zvyšok po celočíselnom delení číslom n . Je potrebné poznamenať, že použitá definícia operácie *k modulo n* sa líši od tejto štandardnej definície pre záporné čísla k , ktorá poskytuje záporný výsledok. V našom prípade, ak je tento výsledok záporný, potom je transformovaný tak, že sa k nemu pripočíta n .

Konvolúcia môže byť taktiež vyjadrená pomocou *cirkulantnej matice*[1]

$$\left. \begin{aligned} c_0 &= a_0b_0 + a_1b_2 + a_2b_1 \\ c_1 &= a_0b_1 + a_1b_0 + a_2b_2 \\ c_2 &= a_0b_2 + a_1b_1 + a_2b_0 \end{aligned} \right\} \Rightarrow \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \underbrace{\begin{pmatrix} a_0 & a_2 & a_1 \\ a_1 & a_0 & a_2 \\ a_2 & a_1 & a_0 \end{pmatrix}}_{\text{circ}(a)} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} \Rightarrow a \otimes b = \text{circ}(a)b \quad (7)$$

Tento konkrétny príklad je zovšeobecnený pre ľubovoľnú dimenziu n takto

$$c = a \otimes b \Leftrightarrow c = \text{circ}(a)b \Leftrightarrow \text{circ}(a) = \begin{pmatrix} a_0 & a_{n-1} & \dots & a_2 & a_1 \\ a_1 & a_0 & \dots & a_3 & a_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n-2} & a_{n-3} & \dots & a_0 & a_{n-1} \\ a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \end{pmatrix} \quad (8)$$

kde všeobecná cirkulantná matica má maticové elementy

$$(\text{circ}(a))_{ij} = a_{[i-j]} \quad (9)$$

Cirkulantná matica má tieto vlastnosti

$$\text{circ}(a \otimes b) = \text{circ}(a)\text{circ}(b) \quad (10)$$

pretože konvolúcia je komutatívna operácia, potom cirkulantné matice navzájom komutujú

$$\text{circ}(a)\text{circ}(b) = \text{circ}(b)\text{circ}(a) \quad (11)$$

Nech X je inverzná matica k cirkulantnej matici $\text{circ}(a)$

$$X\text{circ}(a) = \text{circ}(a)X = E \quad (12)$$

Jej alternatívny tvar je

$$X = \text{circ}^{-1}(a) \quad (13)$$

Nech a^{-1} je *inverzný vektor* vzhľadom k vektoru a , $a^{-1} \otimes a = 1 = (1, 0, \dots, 0, 0)$, potom za predpokladu, že cirkulantná matica je regulárna, $|\text{circ}(a)| \neq 0$, dostaneme

$$\text{circ}^{-1}(a) = \text{circ}(a^{-1}) \quad (14)$$

Definujme unárnu operáciu *involúcie* (pozri obrázok 3)

$$(\)^* : R^n \rightarrow R^n \quad (15)$$

vzťahom

$$b = a^* = (a_{[0]}, a_{[-1]}, \dots, a_{[-n+2]}, a_{[-n+1]}) \quad (16)$$

$$(a_0, a_1, a_2, \dots, a_{n-2}, a_{n-1})^* = (a_0, a_{n-1}, a_{n-2}, \dots, a_2, a_1)$$

Obrázok 3. Znázornenie unárnej operácie involúcie.

Operácia involúcie spĺňa tieto vzťahy

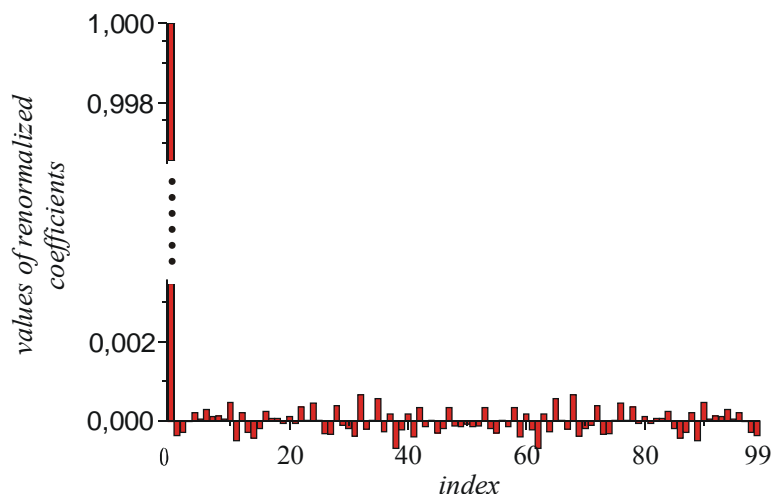
$$(a + b)^* = a^* + b^* \quad (17a)$$

$$(a \otimes b)^* = a^* \otimes b^* \quad (17b)$$

$$(a \otimes b^*) \cdot c = a \cdot (b \otimes c) \quad (17c)$$

$$a^{**} = a \quad (17d)$$

$$\text{circ}(a^*) = \text{circ}^T(a) \quad (17e)$$



Obrázok 4. Histogram jednotlivých zložiek súčiny $\mathbf{c}^* \otimes \mathbf{c}$, kde \mathbf{c} je náhodne generovaný konceptuálny vektor pre $n=100$. Absolútna hodnota „prvej“ zložky $(\mathbf{c}^* \otimes \mathbf{c})_0$ je väčšia o niekoľko rádov ako absolútne hodnoty ostatných zložiek $(\mathbf{c}^* \otimes \mathbf{c})_i$, pre $i \geq 1$. To znamená, že súčin $\mathbf{c}^* \otimes \mathbf{c}$ po vhodnej normalizácii je približne rovný jednotkovému vektoru $(\mathbf{c} \cdot \mathbf{c})^{-1} (\mathbf{c}^* \otimes \mathbf{c}) \doteq \mathbf{1} = (1, 0, \dots, 0, 0)$.

Dolážeme, že involúcia \mathbf{c}^* je približne rovná inverznému vektoru \mathbf{c}^{-1} , $\mathbf{c}^* \otimes \mathbf{c} \approx \mathbf{1}$. Študujme i -tú zložku konvolúcie $\mathbf{c}^* \otimes \mathbf{c}$

$$\begin{aligned}
 (\mathbf{c}^* \otimes \mathbf{c})_i &= \sum_{k=0}^{n-1} c_k^* c_{[i-k]} = \sum_{k=0}^{n-1} c_{[-k]} c_{[i-k]} \\
 &= \begin{cases} \mathbf{c} \cdot \mathbf{c} & (\text{pre } i = 0) \\ \sum_{k=0}^{n-1} c_{[-k]} c_{[i-k]} & (\text{pre } i > 0) \end{cases} \quad (18)
 \end{aligned}$$

Nultá zložka konvolúcie $(\mathbf{c}^* \otimes \mathbf{c})_0$ sa rovná skalárnemu súčinu $\mathbf{c} \cdot \mathbf{c}$, ktorý je vyňjadrený pomocou kladných „diagonálnych“ členov c_i^2 , zatiaľ čo ostatné zložky $(\mathbf{c}^* \otimes \mathbf{c})_i$, pre $i \geq 1$, sú určené sumami „nediagonálnych“ členov $c_i c_j$, ktoré majú náhodné znamienko. Dôsledkom tejto skutočnosti je, že $(\mathbf{c}^* \otimes \mathbf{c})_0$ je podstatne väčšia než ako $(\mathbf{c}^* \otimes \mathbf{c})_i$, pre $i \geq 1$, z čoho priamo vyplýva požadovaná skutočnosť $\mathbf{c}^* \otimes \mathbf{c} \approx \mathbf{1}$, (pozri obr. 4).

Jedna zo základných vlastností holografickej reprezentácie je možnosť rekonštrukcie pôvodných konceptuálnych vektorov, ktoré boli použité pri konštrukcii konvolúcie dvoch vektorov. Táto možnosť je veľmi dôležitá, pretože umožňuje dekódovať pôvodnú informáciu zo zložených konceptuálnych vektorov. *Rekonštrukcia* \mathbf{x} z $\mathbf{c} \otimes \mathbf{x}$ je založená na skutočnosti, že involúcia vektora je približne rovná jeho inverznej hodnote, t.j. $\mathbf{c}^* \otimes \mathbf{c} \approx \mathbf{1}$

$$\tilde{\mathbf{x}} = \mathbf{c}^* \otimes (\mathbf{c} \otimes \mathbf{x}) = (\mathbf{c}^* \otimes \mathbf{c}) \otimes \mathbf{x} \approx \frac{1}{\mathbf{c} \cdot \mathbf{c}} \mathbf{1} \otimes \mathbf{x} = \frac{1}{\mathbf{c} \cdot \mathbf{c}} \mathbf{x} \quad (19)$$

podľa ktorej, konvolúcia \mathbf{c}^* s vektorom $\mathbf{c} \otimes \mathbf{x}$ produkuje vektor $\tilde{\mathbf{x}}$, ktorý je podobný pôvodnému vektoru \mathbf{x} , $\tilde{\mathbf{x}} \approx \mathbf{x}$. Tento výsledok vyjadríme takto

$$\frac{1}{(\mathbf{c} \cdot \mathbf{c})} \tilde{\mathbf{x}} = \begin{pmatrix} x_0 \\ x_1 + \eta_1 \\ \dots \\ x_{n-1} + \eta_{n-1} \end{pmatrix} = \mathbf{x} + \boldsymbol{\eta} \quad (20)$$

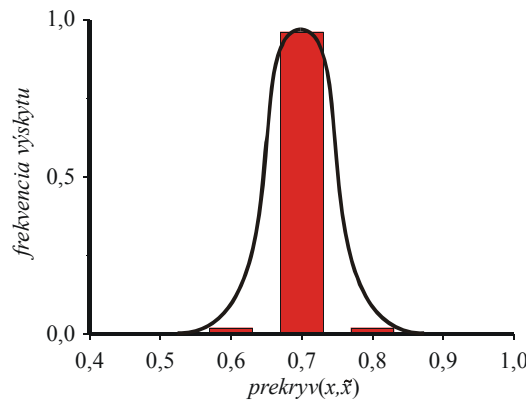
kde vektor $\boldsymbol{\eta}$ je interpretovaný ako náhodný šum s normálnou distribúciou so strednou hodnotou v nule a so štandardnou odchýlkou omnoho menšou než akú má \mathbf{x} .

Prekryv rezultujúceho vektora $\tilde{\mathbf{x}}$ s pôvodným vektorom \mathbf{x} je určený pomocou skalárneho súčinu takto

$$-1 \leq \text{prekryv}(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{\mathbf{x} \cdot \tilde{\mathbf{x}}}{|\mathbf{x}| |\tilde{\mathbf{x}}|} \leq 1 \quad (21)$$

kde nerovnosti vyplývajú priamo zo Schwartzovej nerovnosti z lineárnej algebry. Čím je táto hodnota bližšie k maximálnej hodnote 1, tým sú si vektory $\tilde{\mathbf{x}}$ a \mathbf{x} podobnejšie².

Na obrázku 4 je znázornený histogram prekryvov pre súčin $\mathbf{c} \otimes \mathbf{x}$, obsahujúci dvojicu náhodne vygenerovaných rôznych konceptuálnych vektorov \mathbf{c} a \mathbf{x} dimenzie $n=1000$. Z obrázku vyplýva, že najčastejší prekryv medzi $\tilde{\mathbf{x}} = \mathbf{c}^* \otimes \mathbf{c} \otimes \mathbf{x}$ a \mathbf{x} je okolo 0.7, z čoho vyplýva, že vektory $\tilde{\mathbf{x}}$ a \mathbf{x} sú si podobné, $\tilde{\mathbf{x}} \approx \mathbf{x}$.



Obrázok 5. Histogram prekryvov medzi vektormi $\tilde{\mathbf{x}}$ a \mathbf{x} (dimenzie $n=1000$) je s najväčšou frekvenciou okolo 0.7, z čoho vyplýva, že vektory $\tilde{\mathbf{x}}$ a \mathbf{x} sú si podobné.

Obráťme pozornosť na druhú možnosť verifikácie formuly (20b) pomocou prístupu nazývaného „aditívna pamäť“. Majme množinu obsahujúcu $p+q$ náhodne vygenerovaných konceptuálnych vektorov, $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p, \mathbf{x}_{p+1}, \dots, \mathbf{x}_{p+q}\}$, pričom $p < q$. Pomocou prvých p vektorov z X definujme pamäťový vektor \mathbf{t} ako ich súčet

$$\mathbf{t} = \sum_{i=1}^p \mathbf{x}_i \quad (22)$$

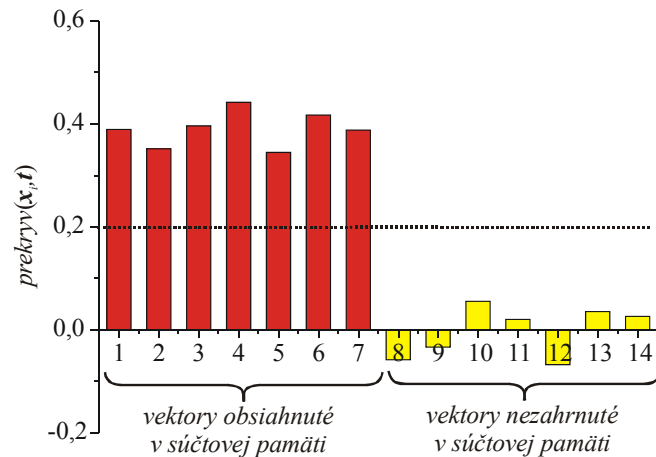
Vektor \mathbf{t} reprezentuje aditívnu pamäť, ktorá jednoduchým aditívnym spôsobom obsahuje vektory z množiny X . Rozhodnutie o tom, či nejaký vektor $\mathbf{x} \in X$ je obsiahnutý v \mathbf{t} musí byť založené na hodnote prekryvu (21)

$$\text{prekryv}(\mathbf{x}, \mathbf{t}) = \frac{\mathbf{x} \cdot \mathbf{t}}{|\mathbf{x}| |\mathbf{t}|} \quad (23)$$

Ak je táto hodnota väčšia ako vopred zvolená prahová hodnota, $\text{prekryv}(\mathbf{x}, \mathbf{t}) \geq \vartheta$, potom

² V prípade, že prekryv sa blíži k hodnote -1, potom vektory $\tilde{\mathbf{x}}$ a \mathbf{x} sú si taktiež podobné, aj keď sú opačne orientované (antikolineárne).

vektor \mathbf{x} je zahrnutý v aditívnej pamäti \mathbf{t} , v opačnom prípade, ak $\text{prekryv}(\mathbf{x}, \mathbf{t}) < \vartheta$, potom vektor \mathbf{x} nie je zahrnutý v \mathbf{t} (pozri obr. 5).



Obrázok 6. Znázornenie súčtovej pamäti pre prvých 7 vektorov množiny X , ktorá obsahuje 14 náhodne vygenerovaných konceptuálnych vektorov dimenzie $n=1000$. Prahovú hodnotu ϑ v tomto prípade môžeme zvoliť 0.2.

V oboch predchádzajúcich príkladoch (pozri obr. 3 a 4) bol použitý rovnaký spôsob určenia konceptuálnych vektorov, ktoré sa „vyskytujú“ v nejakom inom komplexnom konceptuálnom vektore (ktorý môže byť výsledkom zložitých predchádzajúcich výpočtov – transformácií). Použitý spôsob sa nazýva „čistenie“ (angl. *clean-up*), ktorý je formálne špecifikovaný takto: Majme množinu vektorov $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ a nejaký vektor \mathbf{t} . Stojíme pred problémom, rozhodnúť, či pamäťový vektor \mathbf{t} obsahuje aditívnu komponentu, ktorá je podobná (alebo nie je podobná) nejakému vektoru z množiny X . Tento problém riešime pomocou výpočtu tzv. prekryvu (23), formálne

$$\mathbf{x} \approx \mathbf{t} = \begin{cases} \text{áno} & (\text{prekryv}(\mathbf{x}, \mathbf{t}) \geq \vartheta) \\ \text{nie} & (\text{prekryv}(\mathbf{x}, \mathbf{t}) < \vartheta) \end{cases} \quad (24)$$

kde ϑ je zvolená *prahová hodnota* akceptovania veľkosti prekryvu ako pozitívnej odpovedi. Výsledkom tohto procesu čistenia je podmnožina vektorov

$$X(\mathbf{t}) = \{\mathbf{x} \in X; \mathbf{x} \approx \mathbf{t}\} \subseteq X \quad (25)$$

Môžeme si položiť otázku aj v trochu inej podobe, a to, či pamäťový vektor \mathbf{t} je podobný nejakému vektoru z množiny X ? Odpoveď na túto všeobecnejšiu otázku rozhodneme podľa maximálnej hodnoty prekryvu

$$\text{prekryv}(\mathbf{t}, X) = \max_{\mathbf{x} \in X} \text{prekryv}(\mathbf{t}, \mathbf{x}) \quad (26)$$

Potom môžeme prepísať (24) do tvaru

$$\mathbf{x} \approx X = \begin{cases} \text{áno} & (\text{prekryv}(\mathbf{x}, X) \geq \vartheta) \\ \text{nie} & (\text{prekryv}(\mathbf{x}, X) < \vartheta) \end{cases} \quad (27)$$

3. Asociačná pamäť

Konštrukcia asociačnej pamäti patrí medzi hlavné výsledky holografickej redukovanej reprezentácie, ktorá môže byť ďalej zovšeobecnená pomocou tzv. agregácie (angl. *chunking*).

Majme množinu konceptuálnych vektorov $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ a tréningovú množinu $A_{train} = \{\mathbf{c}_i/\mathbf{x}_i; i=1,2,\dots,m\}$, ktorá obsahuje $m < n$ asociačných dvojíc konceptuálnych vektorov $\mathbf{c}_i/\mathbf{x}_i$, kde \mathbf{c}_i je vstup do asociačnej pamäti (angl. *cue*) a \mathbf{x}_i je výstup z pamäti. Zostrojíme pamäťový vektor \mathbf{t} , ktorý reprezentuje *asociačnú pamäť* nad tréningovou množinou A_{train}

$$\mathbf{t} = \mathbf{c}_1 \otimes \mathbf{x}_1 + \dots + \mathbf{c}_m \otimes \mathbf{x}_m = \sum_{i=1}^m \mathbf{c}_i \otimes \mathbf{x}_i \quad (28)$$

Predpokladajme, že vopred poznáme len vstupy asociačnej pamäti \mathbf{c}_i , nepoznáme možné výstupy z množiny $X_{train} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$. Odozva asociačnej pamäti na vstup – narážku \mathbf{c}_i je určená pomocou procesu „čistenia“ reprezentovaného formulou (27). V prvom kroku spočítame vektor $\tilde{\mathbf{x}}_i = \mathbf{c}_i^* \otimes \mathbf{t}$, potom pomocou procesu založenom na maximálnej hodnote prekryvu zistíme či $\tilde{\mathbf{x}}_i \approx \mathbf{x}_i \in X$

$$prekryv(\tilde{\mathbf{x}}_i, X) = \max_{\mathbf{x} \in X_{train}} prekryv(\tilde{\mathbf{x}}_i, \mathbf{x}) \quad (29)$$

Asociačnú pamäť bude ilustrovaná nasledujúcimi dvoma príkladmi.

Prvý príklad

Tento príklad využíva len tréningovú množinu $A_{train} = \{\mathbf{c}_i/\mathbf{x}_i; i=1,2,\dots,m\}$, ktorá je náhodne vygenerovaná pre $m=8$, pričom dimenzia konceptuálnych vektorov je $n=1000$. Pre každú asociačnú dvojicu $\mathbf{c}_i/\mathbf{x}_i$ spočítame $\mathbf{t}_i = \mathbf{c}_i \otimes \mathbf{x}_i$. Hodnoty $prekryv(\mathbf{c}_i^* \otimes \mathbf{t}_i, \mathbf{x}_j)$ sú uvedené v tabuľke

	1	2	3	4	5	6	7	8
1	0.71703	-0.01820	0.01452	0.02776	-0.01488	-0.01922	-0.02442	0.01358
2	-0.03998	0.73804	0.01510	0.01430	0.00276	0.02346	-0.00545	-0.01626
3	-0.02757	-0.01736	0.64667	0.00474	-0.11580	-0.00812	0.01476	0.00379
4	0.00785	0.00374	-0.01899	0.68728	-0.15340	0.00005	-0.00561	0.00136
5	-0.00466	0.00426	-0.01831	-0.00827	0.70767	0.04175	-0.03384	-0.00668
6	-0.01467	0.02522	-0.01403	-0.01316	-0.03000	0.71444	0.00078	-0.00526
7	0.02966	0.00892	-0.00301	-0.00358	0.01285	0.00971	0.70790	0.01816
8	-0.00344	-0.01080	0.00843	-0.01871	0.00324	-0.02629	0.00851	0.58957

Z tejto tabuľky jasne vyplýva, že prekryvy sú dostatočne veľké len pre diagonálne hodnoty, zatiaľ čo nediagonálne prekryvy sú rádovo menšie. Môžeme teda jednoznačne rozhodnúť pomocou prekryvu, či $\mathbf{c}_i^* \otimes \mathbf{t}_i \approx \mathbf{x}_i$ je asociantom s „narážkou“ \mathbf{c}_i .

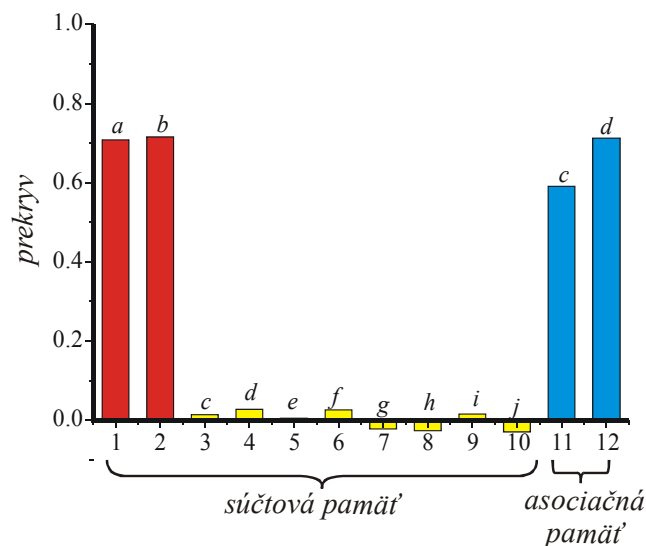
Druhý príklad

V tomto ilustračnom príklade budeme využívať tréningovú množinu $A_{train} = \{\mathbf{c}_i/\mathbf{x}_i\}$, generovanú pre $m=10$ asociačných párov – vektorov dimenzie $n=1000$. Táto pamäť je reprezentovaná pamäťovým vektorom $\mathbf{t} = \mathbf{c}_1 \otimes \mathbf{x}_1 + \dots + \mathbf{c}_m \otimes \mathbf{x}_m$. Nasledujúca tabuľka obsahuje 20 pokusov „čistenia“, pri ktorých sme s 50% pravdepodobnosťou použili ako asociačný vstup vektor \mathbf{c}_i z tréningovej množiny alebo náhodne vygenerovaný konceptuálny

vektor. Tabuľka obsahuje maximálne hodnoty prekryvov (29), pomocou ktorých môžeme jednoznačne identifikovať, či použitý vstup má asociáciu v rámci tréningovej množiny.

#	max. prekryv	index vstupu	index výstupu s max. prekryvom
1	0.311	6	6
2	0.047	náhod.gener.	neexistuje
3	0.383	5	5
4	0.373	10	10
5	0.316	3	3
6	0.397	4	4
7	0.074	náhod.gener.	neexistuje
8	0.065	náhod.gener.	neexistuje
9	0.069	náhod.gener.	neexistuje
10	0.039	náhod.gener.	neexistuje
11	0.344	7	7
12	0.402	8	8
13	0.032	náhod.gener.	neexistuje
14	0.073	náhod.gener.	neexistuje
15	0.017	náhod.gener.	neexistuje
16	0.004	náhod.gener.	neexistuje
17	0.033	náhod.gener.	neexistuje
18	0.056	náhod.gener.	neexistuje
19	0.373	10	10
20	0.037	náhod.gener.	neexistuje

Z tabuľky vyplýva, že asociačná pamäť pomocou procesu čistenia je jednoznačne identifikovateľná, pre konceptuálne vektory hodnoty maximálneho prekryvu dobre špecifikujú existenciu (alebo neexistenciu) príslušných asociačných výstupov.



Obrázok 6. Znázornenie rekognoskácie kombinácie súčtovej a asociačnej pamäti. V prvom kroku sme vykonali proces „čistenia“, pomocou ktorého sme zistili, že pamäť obsahuje len dve položky *a* a *b*, ktoré v ďalšom kroku slúžia ako vstup pre rekognoskáciu asociačnej pamäti, kde sme zistili ako výstupy vektory *c* a *d*. Pri procese „čistenia“ sme ešte navyše verifikovali s negatívnym výsledkom, či súčtová časť pamäti neobsahuje aj ďalšie vektory *c, d, ..., i, j*. S veľkou pravdepodobnosťou môžeme teda rekognoskovať pamäťový vektor *t* ako kombináciu súčtovej a asociačnej pamäti $t = a + b + a \otimes c + b \otimes d$. Dimenzia použitých vektorov je $n=1000$.

Kombinácia súčtovej a asociačnej pamäti

Ukážeme, že aj kombinácia súčtovej a asociačnej pamäti je spoľahlivo rekognoskovateľná, čo sa ukáže v našich ďalších aplikáciách ako výhodná črta holografickej reprezentácie.

Predpokladajme, že máme 10 konceptuálnych vektorov a, b, c, d, \dots, i, j , pomocou prvých štyroch zostrojíme kombináciu súčtovej a asociatívnej pamäti takto

$$\mathbf{t} = \mathbf{a} + \mathbf{b} + \mathbf{a} \otimes \mathbf{c} + \mathbf{b} \otimes \mathbf{d} \quad (30)$$

Pomocou procedúry „čistenia“ získame, že vektor \mathbf{t} obsahuje ako zložky vektory \mathbf{a} a \mathbf{b} , ktoré v následnom kroku odstránim z vektora \mathbf{t}

$$\mathbf{t}' = \mathbf{t} - \mathbf{a} - \mathbf{b} = \mathbf{a} \otimes \mathbf{c} + \mathbf{b} \otimes \mathbf{d} \quad (31)$$

Z odstránenej aditívnej časti použitím techniky rekognoskácia asociatívnej pamäti zistíme, že obsahuje dve dvojice \mathbf{a}/\mathbf{c} a \mathbf{b}/\mathbf{d} , pozri obr. 6.

4. Postupnosť symbolov

Konstruktia asociačnej pamäti neumožňuje uchovať štruktúrovanú informácia, cieľom tejto kapitoly je ukázať, že holografická distribuovaná reprezentácia je schopná spracovať lineárny reťazec znakov, ktoré sú reprezentované postupnosťou konceptuálnych vektorov.

Pre konkrétnosť našich úvah, študujme postupnosť 6 konceptuálnych vektorov dimenzie $n=1000$

$$\text{postupnosť} = \{\mathbf{a} \rightarrow \mathbf{b} \rightarrow \mathbf{c} \rightarrow \mathbf{d} \rightarrow \mathbf{e} \rightarrow \mathbf{f}\} \quad (32)$$

Pre tieto vektory zostrojíme pamäťový vektor

$$\mathbf{t}_0 = \mathbf{a} + \mathbf{a} \otimes \mathbf{b} + \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} + \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d} + \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d} \otimes \mathbf{e} + \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d} \otimes \mathbf{e} \otimes \mathbf{f} \quad (33)$$

Vieme, že tento vektor obsahuje postupnosť vektorov zakódovanú vektorom (33), avšak nevieme ktorých vektorov a v akom poradí. Ukážeme, že pomocou procedúry „čistenia“ z vektora \mathbf{t}_0 môžeme postupne rekonštruovať pôvodnú postupnosť (32) nasledujúcou procedúrou (pozri obr. 7):

1. krok: $\mathbf{a} = \text{clean_up}(\mathbf{t}_0)$, $\mathbf{t}_1 := \mathbf{t}_0 - \mathbf{a}$,

$$\tilde{\mathbf{t}}_1 := \mathbf{a}^* \otimes \mathbf{t}_1,$$

2. krok: $\mathbf{b} = \text{clean_up}(\tilde{\mathbf{t}}_1)$, $\mathbf{t}_2 := \mathbf{t}_1 - \mathbf{a} \otimes \mathbf{b}$,

$$\tilde{\mathbf{t}}_2 := (\mathbf{a} \otimes \mathbf{b})^* \otimes \mathbf{t}_2,$$

3. krok: $\mathbf{c} = \text{clean_up}(\tilde{\mathbf{t}}_2)$, $\mathbf{t}_3 := \mathbf{t}_2 - \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}$,

$$\tilde{\mathbf{t}}_3 := (\mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3)^* \otimes \mathbf{t}_3,$$

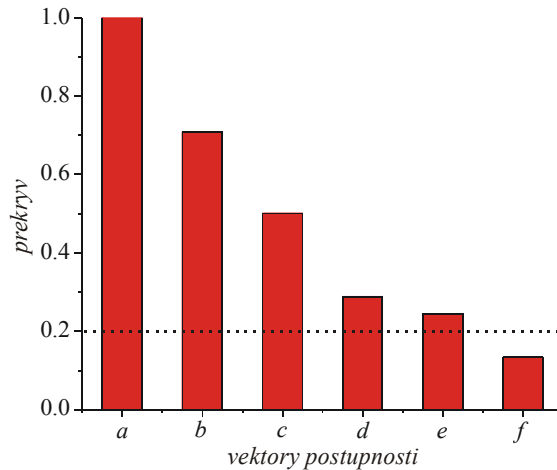
4. krok: $\mathbf{d} = \text{clean_up}(\tilde{\mathbf{t}}_3)$, $\mathbf{t}_4 := \mathbf{t}_3 - \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d}$,

$$\tilde{\mathbf{t}}_4 := (\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d})^* \otimes \mathbf{t}_4,$$

5. krok: $\mathbf{e} = \text{clean_up}(\tilde{\mathbf{t}}_4)$, $\mathbf{t}_5 := \mathbf{t}_4 - \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d} \otimes \mathbf{e}$,

$$\tilde{\mathbf{t}}_5 := (\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d} \otimes \mathbf{e})^* \otimes \mathbf{t}_5,$$

6. krok: $\mathbf{f} = \text{clean_up}(\tilde{\mathbf{t}}_5)$.



Obrázok 8. Prekryv pre jednotlivé vektory s postupnosti z (30), ktoré boli získané rekonštrukciou z vektora t_0 . Z obrázku vyplýva dôležitý poznatok, že nastáva pomerne rýchla degradácia rekonštrukcie, už šiesty vektor f je rekonštruovaný s pravdepodobnosťou menšou ako 0.20.

Funkcia $clean_up(\cdot)$ vykonáva „čistenie“ pre daný vektor t vzhľadom k množine vektorov $X = \{a, b, \dots, f, g, h, \dots\}$. Jednotlivé kroky rekonštrukcie postupnosti konceptuálnych vektorov – znakov sú znázornené na obr. 8, z ktorého vyplýva, že proces rekonštrukcie postupnosti symbolov pomerne rýchlo degraduje, už pri šiestom vektore je prekryv menší ako 0.2.

Postupnosť symbolov môže byť kódovaná taktiež pomocou asociačnej pamäti, kde vektor vstupu c_i špecifikuje i -tú pozíciu daného symbola. Vyššie uvedený ilustračný príklad je reprezentovaný pamäťovým vektorom

$$t = c_1 \otimes a + c_2 \otimes b + c_3 \otimes c + c_4 \otimes d + c_5 \otimes e + c_6 \otimes f \quad (34)$$

Potom rekognoskácia tejto postupnosti spočíva v hľadaní asocianta k vstupnému vektoru c_i , pomocou procesu čistenia zostrojíme „tréningovú množinu“

$$A_{train} = \{c_1/a, c_2/b, c_3/c, c_4/d, c_5/e, c_6/f\} \quad (35)$$

ktorá jednoznačne špecifikuje postupnosť vektorov. Výhoda tohto postupu je v tom, že presnosť rekognoskácia nedegraduje tak rýchlo ako pri pôvodnom postupe založenom na pamäťovom vektore (33).

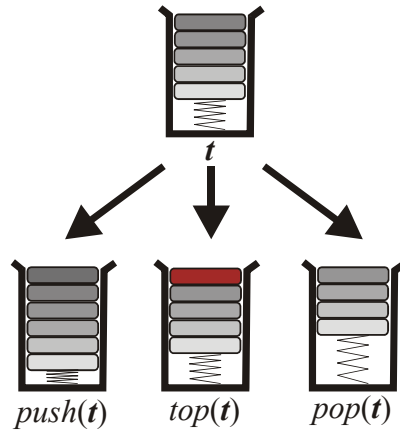
Asociačný prístup k implementácii pamäti pre postupnosť symbolov (reprezentovaných vektormi a, b, \dots) je možné jednoducho zmodifikovať do tvaru tzv. „zásobníkovej pamäti“. Asociačné vstupy c_i sú určené takto

$$c_i = p^i \quad (i = 1, 2, \dots) \quad (34)$$

kde p^i je i -tá (konvolučná) mocnina náhodne generovaného konceptuálneho vektora p , $p^i = p^{i-1} \otimes p$. Potom pamäťový vektor (32) má tvar

$$t = p \otimes x_1 + p^2 \otimes x_2 + p^3 \otimes x_3 + \dots + p^n \otimes x_n \quad (35)$$

kde x_i sú jednotlivé položky pamäti $\{a, b, \dots\}$. Takto interpretovaná asociatívna pamäť pre postupnosť symbolov sa nazýva „zásobníková pamäť“, pomocou mocninových vstupov p^i môžeme jednoducho meniť jej obsah, pozri obr. 9. Nad touto pamäťou definujeme tri rôzne operácie, pomocou ktorých môžeme meniť jej obsah:



Obrázok 9. Tri možné operácie nad zásobníkovou pamäťou reprezentovanou vektorom.

(1) $push(t, x) = p \otimes x + p \otimes t$, nová položka x je zavedená na vrchol zásobníka.

(2) $top(t) = clean_up(p^* \otimes t)$, rekognoskuje vrchnú položku v zásobníku.

(3) $pop(t) = p^{-1} \otimes t - top(t)$, zo zásobníka je odstránená vrchná položka.

Najproblematickejšia je tretia operácia, pomocou ktorého je zo zásobníka odstránená vrchná položka. Korektná implementácia požaduje použitie presného inverzného vektora p^{-1} , aproximácia tohto inverzného vektora pomocou involúcie, $p^{-1} \doteq p^*$, vedie k rýchlejšej degradácii zásobníkovej pamäti.

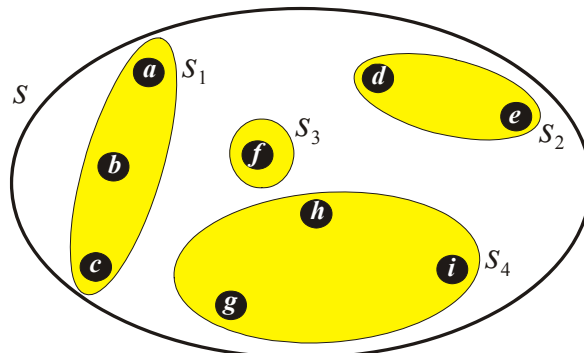
5. Agregovaná pamäť

Agregovaná pamäť pomáha prekonávať problémy s degradáciou pamäti pre postupnosť symbolov (pozri kap. 4). Majme množinu konceptuálnych vektorov $S = \{a, b, \dots, k, l, \dots\}$, túto množinu rozložíme na disjunktné podmnožiny - agregáty

$$S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup \dots \quad (S_i \cup S_j = \emptyset, \text{ pre } i \neq j) \quad (36)$$

Študujme množinu $S = \{a, b, c, d, e, f, g, h\}$, jej rozklad na agregáty vypadá takto (pozri obr. 8)

$$S_1 = \{a, b, c\}, S_2 = \{d, e\}, S_3 = \{f\}, \text{ and } S_4 = \{g, h\} \quad (37)$$



Obrázok 10. Ilustrácia rozkladu množiny S na disjunktné agregáty, pozri (36).

Agregovaná pamäť je reprezentovaná pamäťovým vektorom, ktorý reprezentuje postupnosť agregátov $\{s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4\}$

$$t = s_1 + s_1 \otimes s_2 + s_1 \otimes s_2 \otimes s_3 + s_1 \otimes s_2 \otimes s_3 \otimes s_4 \quad (38b)$$

pričom jednotlivé agregáty sú definované takto pomocou príslušných postupností vektorov (pozri obr. 9)

$$s_1 = a + a \otimes b + a \otimes b \otimes c \quad (38c)$$

$$s_2 = d + d \otimes e \quad (38d)$$

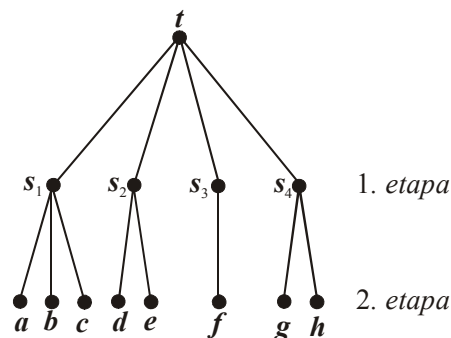
$$s_3 = f \quad (38e)$$

$$s_4 = g + g \otimes h \quad (38f)$$

Proces spracovania agregovanej pamäti obsahuje dve etapy:

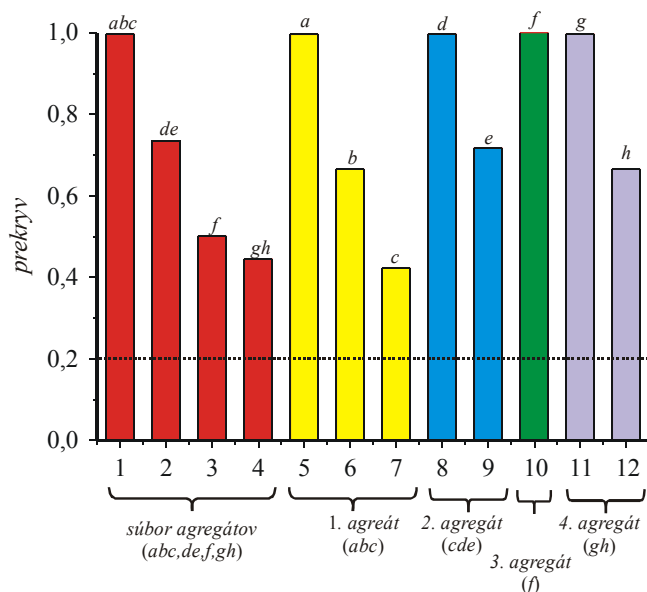
1. etapa – pomocou procesu „čistenia“ identifikujeme agregáty vyskytujúce sa v t (predpokladáme, že „upratovanie“ má množinu $X = \{x_1, x_2, \dots, x_n\}$ z konca 2. kapitoly, kde bol tento proces špecifikovaný, rozšírenú aj o agregáty s_1, s_2, s_3, s_4 , t.j. v našom ilustračnom príklade $X = \{a, b, c, d, e, f, g, h, s_1, s_2, s_3, s_4\}$).

2. etapa – identifikované agregáty sú ďalej analyzované pomocou procesu upratovania.



Obrázok 11. Agregácia 8 conceptuálnych vektorov na 4 agregáty. V 1. etape proces „upratovania“ identifikuje agregáty, ktoré sú potom v 2. etape ďalej identifikované na elementárne conceptuálne vektory.

Výsledky dvoj-etapového procesu „čistenia“ sú znázornené na obr. 11. Z tohto obrázku vyplýva, že v prípade dlhej sekvencie conceptuálnych vektorov rýchla degradácia procesu „čistenia“ môže čiastočne prekonaná pomocou agregácia conceptov na agregáty, ktoré sú na najvyššej úrovni samostatne kódované.



Obrázok 12. Znárodnenie dvoj-etapového procesu „čistenia“, pričom v prvej etape sú identifikované agregáty, zatiaľ čo v druhej etape sú identifikované elementárne konceptuálne vektory z už identifikovaných agregátov.

Agregačná metóda môže byť zovšeobecnená tak, že sa vytvárajú agregáty vyšších rádov, t.j. agregáty zložené z agregátov, pozri obr. 12, kde jednotlivé agregáty sú definované takto (použitie konceptuálne vektory a, b, \dots, c majú dimenziu $n=1000$)

$$s_1 = a + a \otimes b + a \otimes b \otimes c \quad (39a)$$

$$s_2 = d + d \otimes e \quad (39b)$$

$$s_3 = f \quad (39c)$$

$$s_4 = g + g \otimes h \quad (39d)$$

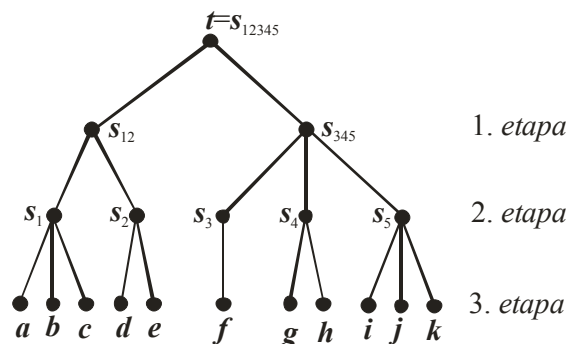
$$s_5 = i + i \otimes j + i \otimes j \otimes k \quad (39e)$$

$$s_{12} = s_1 + s_1 \otimes s_2 \quad (39f)$$

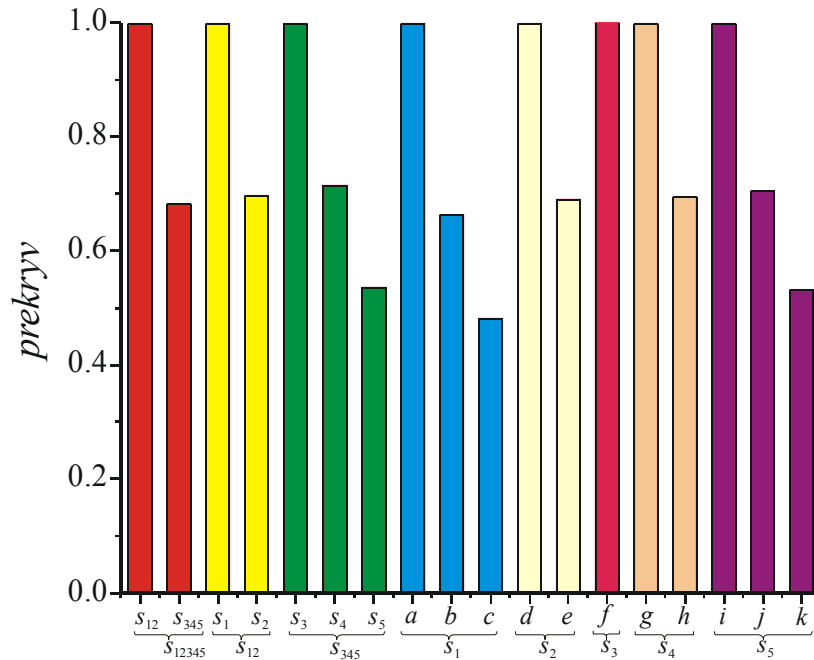
$$s_{345} = s_3 + s_3 \otimes s_4 + s_3 \otimes s_4 \otimes s_5 \quad (39g)$$

$$t = s_{12345} = s_{12} + s_{12} \otimes s_{345} \quad (39h)$$

Proces „čistenia“ agregovaného vektora t špecifikovaného (39h) je znázornená na obr. 13. V 1. etape rekognoskujeme vektor t , zistíme, že obsahuje dva agregáty s_{12} a s_{345} . V 2. etape analyzujeme agregáty z predchádzajúcej 1. etapy, zistíme, že obsahujú agregáty s_1, s_2, \dots, s_5 . V poslednej tretej etape analyzujeme agregáty z predchádzajúceho kroku, ktoré už obsahujú elementárne konceptuálne vektory a, b, \dots, k . Prekrývy resultujúcich vektorov v procese „čistenia“ sú znázornené na obr. 14.



Obrázok 13. Znárodnenie agregácie vyššieho rádu, kde sú použité agregáty zložené z agregátov.



Obrázok 14. Znázornenie 3-etapového čistenia agregovaného vektora t špecifikovaného vzťahom (39h). V prvej etape sa rekognoskuje vektor t , zistí sa, že obsahuje dva agregáty s_{12} a s_{345} . V druhej etape sa rekognoskujú dva agregáty z predchádzajúcej etapy, zistí sa, že obsahujú agregáty s_1, s_2, \dots, s_5 . V poslednej tretej etape sa rekognoskuje postupne 5 agregátov identifikovaných v predchádzajúcej etape, obsahujú konceptuálne vektory a, b, \dots, k .

Z uvedených ilustračných príkladov vyplýva, že prístup agregovanej pamäti reprezentuje efektívny spôsob prekonania rýchlej degradácie pôvodnej verzie postupnej rekognoskácie vektora (33). Tým, že združíme niekoľko konceptuálnych vektorov do agregátu, získame jednoduchú možnosť rozšírenia našich možností korektné rekognoskovať väčšie množiny konceptuálnych vektorov. Proces agregácie môže mať niekoľko hierarchických úrovní, čím sa naše možnosti zapamätania a rekognoskácie konceptuálnych vektorov stávajú skoro neohraničené.

6. Kódovanie relácií

Holografická redukovaná reprezentácia je vhodným prostriedkom aj na kódovanie relácií (predikátov). Študujme binárnu reláciu $P(x,y)$, použijúc Pascalovský zápis, táto relácia je formálne špecifikovaná touto hlavičkou

$$\text{function } P(x : \text{type}_1; y : \text{type}_2) : \text{type}_3 \quad (40)$$

Jednotlivé argumenty relácie sú špecifikované typmi type_1 a type_2 , ktoré špecifikujú doménu nad ktorou sú tieto premenné určené; podobne aj samotná relácia P je chápaná ako funkcia, ktorej doména hodnôt je špecifikovaná typom type_3 . V mnohých prípadoch doména premenných a aj samotnej relácie je jedna, preto ich špecifikácie môžu byť vynechané, čo podstatne zjednodušuje holografickú reprezentáciu relácií. Redukovaný tvar relácie (40) vypadá takto

$$\text{function } P(x; y) \quad (41)$$

kde už vopred vieme akého typu sú premenné x, y , a taktiež aj samotná relácia P . Holografická reprezentácia relácie (40) môže mať tento tvar

$$t = P + \text{variable}_1 + \text{variable}_2 + P \otimes (\text{type}_3 + \text{variable}_1 \otimes (x + \text{type}_1) + \text{variable}_2 \otimes (y + \text{type}_2)) \quad (42)$$

Jej dekódovanie prebieha postupne. V prvej etape pomocou procedúry „čistenia“ rekognoskujeme názov (identifikátor) relácie P a taktiež aj názvy (identifikátory) jej jednotlivých premenných x a y . V druhom kroku rekognoskujeme typ $type_3$ relácie P , v záverečnom treťom kroku pomocou predchádzajúcich výsledkov rekognoskujeme premenné x , y a taktiež aj ich typy $type_1$ a $type_2$. V mnohých prípadoch reprezentácia predikátora $P(x,y)$ je postačujúca v tomto zjednodušenom tvare (pozri (41))

$$t = P + variable_1 \otimes x + variable_2 \otimes y \quad (43)$$

Zvolená metóda holografickej reprezentácie relácie je ľahko zovšeobecniteľná aj zložitejšie relácie, kde premenné sú taktiež predikáty, napr. $P(x, Q(y,z))$, kde „vnútorný“ predikát Q je charakterizovaný

$$function Q(y : type_3; z : type_4) : type_5 \quad (44)$$

K tomu, aby sme mohli vytvoriť zloženú reláciu $P(x, Q(y,z))$, musíme predpokladať typovú kompatibilitu druhej premennej relácie P a typu relácie Q , t.j. $type_2 = type_4$. V zjednodušenom prístupe, kde všetky typy sú rovnaké, nie je potrebné rozlišovať typy jednotlivých premenných a samotných relácií. Zjednodušená holografická reprezentácia relácie (44) má tento tvar

$$t' = Q + variable_1 \otimes y + variable_2 \otimes z \quad (45)$$

Dosadením reprezentácie (45) za premennú y v reprezentácii (43) dostaneme túto výslednú reprezentáciu zloženej relácie $P(x, Q(y,z))$

$$\begin{aligned} t &= P + variable_1 \otimes x + variable_2 \otimes (Q + variable_3 \otimes y + variable_4 \otimes z) \\ &= P + variable_1 \otimes x + variable_2 \otimes Q + \\ &\quad variable_2 \otimes variable_3 \otimes y + variable_2 \otimes variable_4 \otimes z \end{aligned} \quad (46)$$

Prvý ilustračný príklad – podobnosť medzi geometrickými obrázkami

Na obrázku 14 je znázornených 48=6×8 geometrických útvarov, ktoré obsahujú v horizontálnom alebo vertikálnom usporiadaní dva objekty, ktoré navyše môžu byť ešte malé alebo veľké. Označme holografické reprezentácie príslušných atomárnych koncepcií takto:

Obrázce: **tr** (triangle), **sq** (square), **ci** (circle), **st** (star)

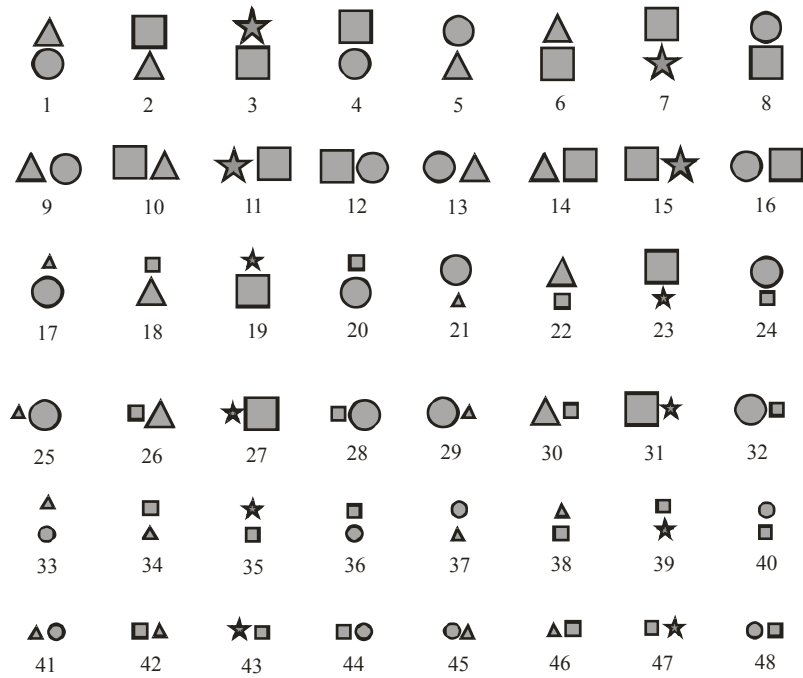
Unárne relácie: **sm** (small), **lg** (large)

Binárne relácie: **hor** (horizontal), **ver** (vertical)

Premenné: **ver_var₁** (1st variable for binary relation **ver**), **ver_var₂** (2nd variable for binary relation **ver**), **hor_var₁** (1st variable for binary relation **hor**), **hor_var₂** (2nd variable for binary relation **hor**)

Jednotlivé obrázky z obr. 14 sú charakterizované reláciami uvedenými v nasledujúcej tabuľke.

riadok	špecifikácia
1	ver(lg(x),lg(y))
2	hor(lg(x),lg(y))
3	hor(sm(x),lg(y)) a hor(lg(x),sm(y))
4	ver(sm(x),lg(y)) a ver(lg(x),sm(y))
5	ver(sm(x),sm(y))
6	hor(sm(x),sm(y))



Obrázok 15. Množina 48 podobných obrázkov, ktoré obsahujú dva objekty, umiestnené buď vedľa seba alebo nad sebou a objekty sú buď malé alebo veľké. Každý stĺpec obsahuje dvojicu podobných objektov, ktoré sa líšia len svojím umiestnením alebo veľkosťou.

Holografické reprezentácie jednotlivých prípadov z tejto tabuľky majú tento tvar (porovnaj s rovnicou (43)).

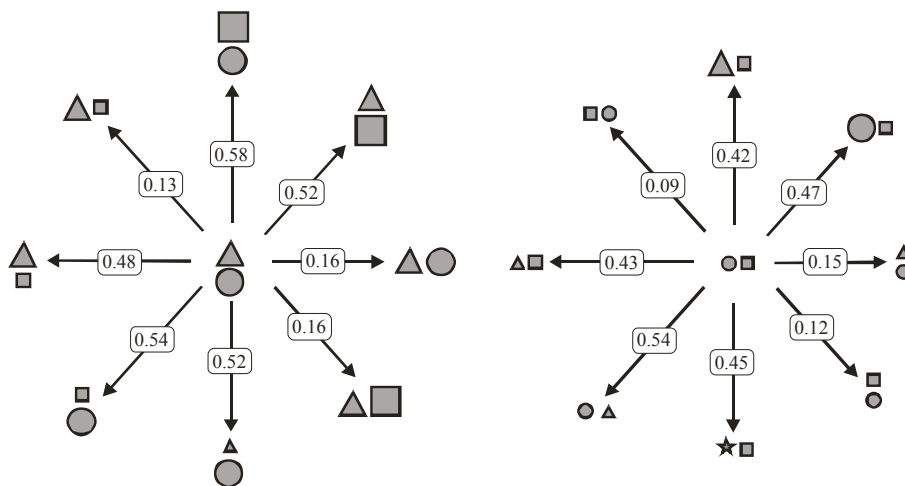
$$\begin{aligned}
 t_{1,x,y} &= \mathit{ver} + \langle \mathit{ver_var}_1 \otimes \mathit{lg} \otimes x + \mathit{ver_var}_2 \otimes \mathit{lg} \otimes y \rangle \\
 t_{2,x,y} &= \mathit{hor} + \langle \mathit{hor_var}_1 \otimes \mathit{lg} \otimes x + \mathit{hor_var}_2 \otimes \mathit{lg} \otimes y \rangle \\
 t_{3,x,y} &= \begin{cases} \mathit{ver} + \langle \mathit{ver_var}_1 \otimes \mathit{lg} \otimes x + \mathit{ver_var}_2 \otimes \mathit{sm} \otimes y \rangle \\ \mathit{ver} + \langle \mathit{ver_var}_1 \otimes \mathit{sm} \otimes x + \mathit{ver_var}_2 \otimes \mathit{lg} \otimes y \rangle \end{cases} \\
 t_{4,x,y} &= \begin{cases} \mathit{hor} + \langle \mathit{hor_var}_1 \otimes \mathit{lg} \otimes x + \mathit{hor_var}_2 \otimes \mathit{sm} \otimes y \rangle \\ \mathit{hor} + \langle \mathit{hor_var}_1 \otimes \mathit{sm} \otimes x + \mathit{hor_var}_2 \otimes \mathit{lg} \otimes y \rangle \end{cases} \\
 t_{5,x,y} &= \mathit{ver} + \langle \mathit{ver_var}_1 \otimes \mathit{sm} \otimes x + \mathit{ver_var}_2 \otimes \mathit{sm} \otimes y \rangle \\
 t_{6,x,y} &= \mathit{hor} + \langle \mathit{hor_var}_1 \otimes \mathit{sm} \otimes x + \mathit{hor_var}_2 \otimes \mathit{sm} \otimes y \rangle
 \end{aligned} \tag{48}$$

kde x a y sú holografické reprezentácie jednotlivých obrázkov (tr , sq , ci , st) a zátvorka $\langle u \rangle$ indikuje, že vektor u je normalizovaný. Podobnosť medzi jednotlivými obrázkami je určená prekryvom ich holografickými reprezentáciami

$$\mathit{podobnost}'(X, X') = \mathit{overlap}(t, t') \tag{48}$$

Získané výsledky sú znázornené na obr. 15. Dominantným determinantom podobnosti je ich vertikálne alebo horizontálne usporiadanie. Prekryv (t.j. aj podobnosť) medzi dvoma obrázkami, ktoré majú rôzne usporiadanie je obvykle menší ako 0.1.

Vo všeobecnosti, holografická redukovaná reprezentácia umožňuje pomerne jednoduché stanovenie podobnosti medzi objektmi, ktoré sú špecifikované predikátovou štruktúrou (40) alebo jej zovšeobecnením pomocou ďalších vložených (nested) predikátov (pozri (46)). Táto možnosť otvára úplne nové možnosti ako hľadať podobné objekty alebo riešiť problém analógie, ktorý je štandardne považovaný za veľmi obtiažny problém umelej inteligencie vyžadujúci špeciálne symbolické techniky [xx].



Obrázok 16. Znáoznenie podobných obrázkov pre dva vybrané obrázky 1 a 48 (pozri obr. 16). Jednotlivé šípky obsahujú prekryv medzi obrázkami spočítaný vzťahom (48).

Druhý ilustračný príklad – podobnosť medzi binárnymi číslami

Budeme študovať podobnosť medzi binárnymi číslami dĺžky 3, ktoré sú reprezentované reťazcom $(\alpha_1\alpha_2\alpha_3) \in \{0,1\}^3$. Toto číslo môžeme chápať ako usporiadanú trojicu binárnych symbolov, ktoré v distribuovanej reprezentácii sú reprezentované takto (pozri (33))

$$t_{(\alpha_1\alpha_2\alpha_3)} = t_{\alpha_1} + t_{\alpha_1} \otimes t_{\alpha_2} + t_{\alpha_1} \otimes t_{\alpha_2} \otimes t_{\alpha_3} \tag{49a}$$

$$t_{\alpha} = \begin{cases} \text{nula} & (\text{pre } \alpha = 0) \\ \text{jedna} & (\text{pre } \alpha = 1) \end{cases} \tag{49b}$$

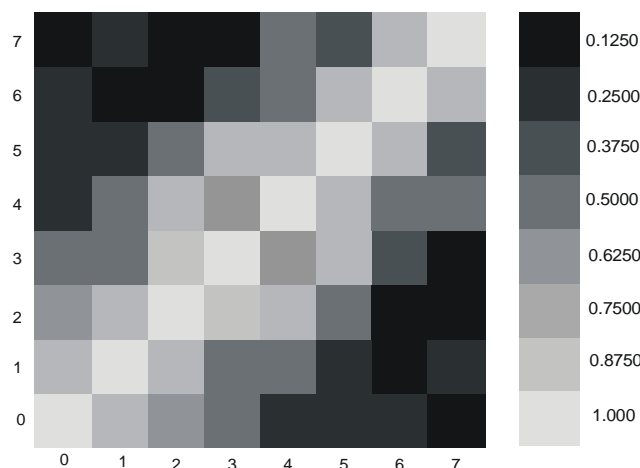
kde *nula* a *jedna* sú distribuované reprezentácie čísiel ‘0’ resp. ‘1’. Tak napríklad, binárne číslo (101) je reprezentované takto

$$t_{(101)} = \text{jedna} + \text{jedna} \otimes \text{nula} + \text{jedna} \otimes \text{nula} \otimes \text{jedna} \tag{50}$$

Podobnosť medzi jednotlivými reprezentáciami odráža aj podobnosť medzi príslušnými binárnymi číslami; reprezentácia dvoch blízkych binárnych čísel je nepriamo úmerná ich vzdialenosti (napr. absolútnej hodnote rozdielu). V nasledujúcej tabuľke sú uvedené podobnosti medzi reprezentáciami binárnych čísel, ktoré boli počítane pomocou ich prekryvu (23).

	000	001	010	011	100	101	110	111
000	1.00	0.74	0.50	0.46	0.21	0.16	0.14	0.11
001	0.74	1.00	0.71	0.42	0.45	0.15	0.12	0.15
010	0.50	0.71	1.00	0.75	0.70	0.43	0.09	0.08
011	0.46	0.42	0.75	1.00	0.46	0.70	0.37	0.04
100	0.21	0.45	0.70	0.46	1.00	0.74	0.43	0.39
101	0.16	0.15	0.43	0.70	0.74	1.00	0.70	0.35
110	0.14	0.12	0.09	0.37	0.43	0.70	1.00	0.69
111	0.11	0.15	0.08	0.04	0.39	0.35	0.69	1.00

Maximálna podobnosť je medzi dvojicami reprezentácií, ktoré sú priradené dvom susedným celým číslam, minimálna podobnosť medzi dvoma reprezentáciami existuje vtedy, ak priradené čísla majú maximálnu vzdialenosť 7. Tieto údaje z tabuľky sú graficky znázornené na obr. 17.



Obrázok 17. Grafické znázornenie podobnosti medzi distribuovanými reprezentáciami binárnych čísel, veľkosť prekryvu medzi jednotlivými reprezentáciami je úmerná svetlosti odpovedajúcej štvorcovej oblasti (oblasti na vedľajšej diagonále sú najsvetlejšie). V pravom dolnom a v ľavom hornom rohu sú oblasti najtmavšie, čo odpovedá skutočnosti, že tieto oblasti sú priradené reprezentáciám maximálne vzdialených čísel.

Z tohoto jednoduchého ilustračného príkladu vyplýva, že v rámci holografickej distribuovanej reprezentácie môžeme navrhovať (aspoň potenciálne) asociačné reprezentácie typu (28), kde asociačné „narážky“ odpovedajú číslam. To znamená, že v tomto distribuovanom prístupe existuje možnosť asociačnej simulácie ľubovoľnej funkcie, čo podstatne zvyšuje možnosti metódy smerom k jej univerzálnosti.

7. Uvažovanie pomocou *modus ponens* a *modus tollens*

Simulácia procesov uvažovania (inferencie) patrí medzi základné problémy, ktoré sa znovu a znovu riešia v umelej inteligencii a v kognitívnej vede [xx]. Fodorova kritika konekcionizmu [xx] bola práve založená na unáhlenom závere, že umelé neurónové siete nie sú schopné simulovať vyššie kognitívne aktivity, ktoré sú vraj výhradnou doménou klasického symbolického prístupu. Tento Fodorov názor sa ukázal ako nesprávny, ďalší rozvoj teórie neurónových sietí ukázal, že konekcionizmus je univerzálnym výpočtovým prostriedkom, ktorý nemá hranice aplikovateľnosti, neexistujú preň domény neaplikovateľnosti, ktoré by boli preň zakázané, samozrejme môže sa ukázať, že v niektorých doménach jeho použitie je neobyčajne ťažkopádne a zbytočne komplikované, že existujú iné prístupy, ktoré v danej doméne poskytujú podstatne jednoduchšie a priamočiarejšie riešenie, než ako umožňujú neurónové siete.

V tejto kapitole ukážeme na možnosť reprezentácie dvoch základných módov deduktívneho uvažovania výrokovej logiky,

$$\frac{p \Rightarrow q}{p} \quad \text{a} \quad \frac{p \Rightarrow q}{\bar{q}} \quad \bar{p} \quad (51)$$

ktoré sa nazývajú *modus ponens* resp. *modus tollens*. Tieto módy uvažovania sú ekvivalentné týmto tautológiám výrokovej logiky

$$((p \Rightarrow q) \wedge p) \Rightarrow q \quad (52a)$$

$$((p \Rightarrow q) \wedge \bar{q}) \Rightarrow \bar{p} \quad (52b)$$

Implikáciu ' \Rightarrow ' môžeme chápať ako binárnu reláciu, ktorú je možno holografickým distribuovaným spôsobom reprezentovať takto (pozri vzťah (45))

$$t_{p \Rightarrow q} = op \otimes impl + var_1 \otimes p + var_2 \otimes q \quad (53)$$

ktorá obsahuje spčet troch častí, prvá časť špecifikuje typ relácie (implikáciu), druhá a tretia časť špecifikuje prvú (antedecedent) resp. druhú (konsekvent) premennú relácie implikácie. Tento konceptuálny vektor reprezentujúci reláciu implikácie môže byť transformovaný takto

$$\tilde{t}_{p \Rightarrow q} = t_{p \Rightarrow q} \otimes T \quad (54a)$$

kde

$$T = var_1^* \otimes p^* \otimes p^* \otimes q + var_2^* \otimes q^* \otimes \bar{q}^* \otimes \bar{p} \quad (54b)$$

Transformovaná reprezentácia implikácie je reprezentuje súčet dvoch asociačných dvojíc

$$\tilde{t}_{p \Rightarrow q} \approx p^* \otimes q + \bar{q}^* \otimes \bar{p} \quad (55)$$

ktorá poskytuje holografickú reprezentáciu pravidiel *modus ponens* a *modus tollens*

$$p \otimes \tilde{t}_{p \Rightarrow q} \approx q \quad (56a)$$

$$\bar{q} \otimes \tilde{t}_{p \Rightarrow q} \approx \bar{p} \quad (56b)$$

Prvý vzťah (56a) môžeme chápať ako holografickú reprezentáciu *modus ponens* (pozri (51) a (52a)), zatiaľ čo druhý vzťah je holografická reprezentácia *modus tollens* (pozri (51) a (52b)).

Podobný výsledok môže byť získaný aj iným alternatívnym spôsobom, ktorý je založený na disjunktnej tvare implikácie

$$(p \Rightarrow q) \equiv (\bar{p} \vee q) \quad (57)$$

Distribúovaná reprezentácia implikácie v tomto alternatívnom tvare má túto podobu

$$t_{\bar{p} \vee q} = op \otimes disj + var_1 \otimes \bar{p} + var_2 \otimes q \quad (58)$$

Transformáciou tejto reprezentácie dostaneme (porovnaj (53))

$$\tilde{t}_{\bar{p} \vee q} = t_{\bar{p} \vee q} \otimes T \approx \bar{p} \otimes \bar{q}^* + q \otimes p^* \quad (59a)$$

kde

$$T = var_1^* \bar{q}^* + var_2^* p^* \quad (59b)$$

Táto transformácia je podstatne jednoduchšia než ako v predchádzajúcom prípade (54b). Pravidlá *modus ponens* a *modus tollens* sú teraz realizované vzťahmi podobnými s (56a-b). Navyše dostávame ešte tieto dve „pravidlá“

$$q^* \otimes \tilde{t}_{\bar{p} \vee q} \approx p^* \quad (60a)$$

$$\bar{p}^* \otimes \tilde{t}_{\bar{p} \vee q} \approx \bar{q}^* \quad (60b)$$

ktoré pripomínajú dobre známe pravidlá chybného uvažovania

$$\begin{array}{c} p \Rightarrow q \\ q \\ p \end{array} \quad \text{a} \quad \begin{array}{c} p \Rightarrow q \\ \bar{p} \\ \bar{q} \end{array} \quad (61)$$

ktoré sa nazývajú „*potvrdenie dôsledku*“ resp. „*poprenie predpokladu*“. Táto skutočnosť je spôsobená tým, že transformované reprezentácie implikácie $\tilde{t}_{p \Rightarrow q}$ a $\tilde{t}_{\bar{p} \vee q}$ nie sú identické, reprezentácia $\tilde{t}_{\bar{p} \vee q}$ vedie práve k neočakávaným výsledkom (61), ktoré reprezentujú nesprávne módy usudzovania (ktoré sú však často, aj napriek tomu, často používané ľuďmi nepoznajúcimi zásady logiky).

8. Predikátová logika

Budeme sa zaoberať jednoduchou formou predikátovej logiky, ktorá je založená na unárnych predikátoch, $P(x)$, ktorých distribuovaná reprezentácia má tvar (pozri kapitolu 6)

$$t_{P(x)} = pred \otimes P + pred_var \otimes x \quad (62)$$

Spojenie tohoto predikátu s univerzálnym kvantifikátorom, $(\forall x)P(x)$, môže byť reprezentované nasledujúcim spôsobom

$$\mathbf{t}_{(\forall x)} = \mathbf{uni_quant} \otimes \mathbf{uni} + \mathbf{uni_quant_var} \otimes \mathbf{x} \quad (63a)$$

$$\mathbf{t}_{(\forall x)P(x)} = \mathbf{t}_{(\forall x)} + \mathbf{t}_{(\forall x)} \otimes (\mathbf{pred} \otimes \mathbf{P} + \mathbf{pred_var} \otimes \mathbf{x}) \quad (63b)$$

Oba konceptuálne vektory $\mathbf{t}_{P(x)}$ a $\mathbf{t}_{(\forall x)P(x)}$ môžu byť rekognoskované pomocou „čistiacej“ procedúry.

Pre naše ďalšie účely štúdia procesov uvažovania v rámci predikátovej logiky a ich distribuovanej reprezentácie je tento prístup zbytočne zložitý, používanie konceptuálneho vektora $\mathbf{t}_{(\forall x)}$ pre reprezentáciu symbolu $(\forall x)$ v podstate len zbytočne komplikuje proces rekognoskácie zložených konceptuálnych vektorov obsahujúcich ako zložku $\mathbf{t}_{(\forall x)}$. Preto opustíme od explicitného používania symbolu $(\forall x)$, jeho význam bude substituovaný používaním „univerzálnej premennej“ x , t.j. predikát $P(x)$ obsahujúci univerzálnu premennú x sa interpretuje ako $(\forall x)P(x)$, môžeme teda s určitou obozretnosťou používať „formulu“ $(\forall x)P(x) \equiv P(x)$.

V predikátovej logike platí pravidlo univerzálneho odlúčenia, pomocou ktorého sa konkretizuje predikát s univerzálnym kvantifikátorom na predikát s konkrétnou premennou a , $(\forall x)P(x) \Rightarrow P(a)$, ktorý je dôsledkom jednoduchšej tautológie výrokovej logiky $((p \wedge q) \wedge p) \Rightarrow q$. Použitím konvencie univerzálnej premennej x túto konkretizáciu prepíšeme do jednoduchšieho tvaru

$$P(x) \Rightarrow P(a) \quad (64)$$

Zostrojíme distribuovanú reprezentáciu tejto univerzálnej eliminácie jednoduchého unárneho replikátora pomocou transformačného vektora \mathbf{T} , pozri formuly (54-56). Distribuovaná reprezentácia (64) má tvar

$$\mathbf{t}_{P(a)} \approx \mathbf{t}_{P(x)} \otimes \mathbf{T} \quad (65)$$

kde $\mathbf{T} = \mathbf{x}^* \otimes \mathbf{a}$, potom

$$\mathbf{t}_{P(a)} \approx \mathbf{t}_{P(x)} \otimes \mathbf{x}^* \otimes \mathbf{a} \quad (66)$$

Sumarizujúc naše úvahy, distribuovaná reprezentácia pravidla univerzálnej eliminácie (konkretizácie) sa uskutočňuje pomocou transformačného vektora \mathbf{T} , pomocou ktorého sa univerzálna premenná x substituuje „konkrétnou“ premennou a .

Pomocou tohto zjednodušenej reprezentácie kvantifikovaných predikátov budeme študovať tzv. *zovšeobecnený modus ponens* a *zovšeobecnený modus tollens*

$$\frac{(\forall x)(P(x) \Rightarrow Q(x)) \quad P(a)}{Q(a)} \quad \text{a} \quad \frac{(\forall x)(P(x) \Rightarrow Q(x)) \quad \bar{Q}(a)}{\bar{P}(a)} \quad (67a)$$

alebo v zjednodušenej podobe pomocou univerzálnej premennej x

$$\frac{P(x) \Rightarrow Q(x) \quad P(a)}{Q(a)} \quad \text{a} \quad \frac{P(x) \Rightarrow Q(x) \quad \bar{Q}(a)}{\bar{P}(a)} \quad (67b)$$

Tieto zovšeobecnené schémy deduktívneho uvažovania vyplývajú priamo z ich štandardnej výrokovej formy (51) a konkretizácie (61). Distribuovaná reprezentácia hlavnej (vrchnej) premisy týchto pravidiel má tvar

$$t_{P(x) \Rightarrow Q(x)} = op \otimes impl + var_1 \otimes (pred \otimes P + pred_var \otimes x) + var_2 \otimes (pred \otimes Q + pred_var \otimes y) \quad (68)$$

Konkretizácia implikácie $P(x) \Rightarrow Q(x)$ na $P(a) \Rightarrow Q(b)$, je formálne vyjadriteľná pomocou implikácie (pozri (64))

$$(P(x) \Rightarrow Q(x)) \Rightarrow (P(a) \Rightarrow Q(b)) \quad (69)$$

kde práva strana má nasledujúcu distribuovanú reprezentáciu

$$t_{P(a) \Rightarrow Q(b)} = op \otimes impl + var_1 \otimes (P + a) + var_2 \otimes (Q + b) \quad (70)$$

Podobne ako v úvodnom ilustračnom príklade (pozri (65)), tento prechod vyjadrený implikáciou (69), môže byť v distribuovanej reprezentácii vyjadrený pomocou transformácie

$$\tilde{t}_{P(a) \Rightarrow Q(b)} = t_{P(a) \Rightarrow Q(b)} \otimes T \approx (P + a)^* \otimes (Q + b) + (\bar{Q} + b)^* \otimes (\bar{P} + a) \quad (71)$$

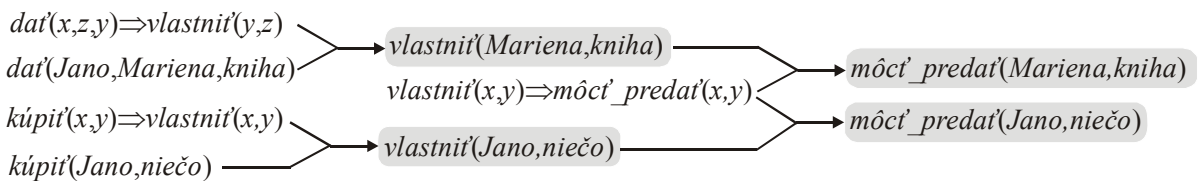
kde nová transformovaná distribuovaná reprezentácia $\tilde{t}_{P(a) \Rightarrow Q(b)}$ vyhovuje vzťahom, ktoré reprezentujú pravidlá (67) modus ponens resp. modus tollens

$$(P + a) \otimes \tilde{t}_{P(a) \Rightarrow Q(b)} \approx (Q + b) \quad (72a)$$

$$(\bar{Q} + b) \otimes \tilde{t}_{P(a) \Rightarrow Q(b)} \approx (\bar{P} + a) \quad (72b)$$

Ilustračný príklad – modelovanie reflexívneho uvažovania

V tomto ilustračnom príklade ukážeme, že holografická distribuovaná reprezentácia poskytuje formálne prostriedky, pomocou ktorého môžeme simulovať proces usudzovania založeného na zovšeobecnenom modus ponens (67). Tento proces bol široko študovaný Shastrim a Ajanagaddom [xx] pomocou konekcionistického systému nazývaného SHRUTI, ktorý bol schopný simulovať reflexívne uvažovanie založenom na predikátovej logike. Podobné výsledky budú dosiahnuté aj pomocou formalizmu holografickej distribuovanej reprezentácie.



Obrázok 18. Znázornenie použitia zovšeobecneného pravidla modus ponens (64) pre vyvedenie poznatkov (označených tmavým podkladom a taktiež aj výstupnými šípkami) z implikácií (1-3) a zo vstupných pozorovaní (a-c), ktoré sú označené vstupnými šípkami.

Majme formálny systém obsahujúcou tieto tri všeobecné pravidlá (pozri obr.(18))

(1) $dat'(x, y, z) \Rightarrow vlastnit'(y, z)$, *typ* x : darca; *typ* y : príjemca; *typ* z : vec,

(2) $kupit'(y, z) \Rightarrow vlastnit'(y, z)$, *type* y : kupujúci; *typ* z : vec,

(3) $vlastnit'(y, z) \Rightarrow mocit'_predat'(y, z)$, *typ* y : vlastník; *typ* z : vec,

a tri pozorovania (skutočnosti – fakty)

(a) $dat'(Jano, Mariena, kniha)$,

(b) *vlastniť (Mariena, kniha)*

(c) *kúpiť (Jano, niečo)*.

Aké sú deduktívne závery z tohto systému? Výsledky sú naznačené na obrázku 15, teraz pristúpime k ich odvodeniu pomocou distribuovanej reprezentácie založenej na konceptuálnych vektoroch a operáciách nad nimi.

Sledujme prvý zovšeobecnený modus ponens z obr. 16

$$\begin{aligned} dat'(x, y, z) &\Rightarrow vlastniť(y, z) \\ \underline{dat'(Jano, Mariena, kniha)} & \\ \underline{vlastniť(Mariena, kniha)} & \end{aligned} \quad (73)$$

Pomocou postupu (68-72) môžeme realizovať túto schému uvažovania pomocou reprezentácie konceptuálnych vektorov, jej jednotlivé zložky (idúc zhora nadol) sú reprezentované takto

$$\mathbf{t}_1^{(1)} = op \otimes impl + var_1 \otimes \mathbf{t}_{dat'(x,y,x)} + var_2 \otimes \mathbf{t}_{vlastniť(y,x)} \quad (74a)$$

$$\begin{aligned} \mathbf{t}_2^{(1)} = \mathbf{t}_{dat'(Jano, Mariena, kniha)} &= dat' + dat'_var_1 \otimes Jano + \\ &dat'_var_2 \otimes Mariena + dat'_var_3 \otimes kniha \end{aligned} \quad (74b)$$

$$\begin{aligned} \mathbf{t}_3^{(1)} = \mathbf{t}_{vlastniť(Mariena, kniha)} &= vlastniť + vlastniť'_var_1 \otimes Mariena + \\ &vlastniť'_var_2 \otimes kniha \end{aligned} \quad (74c)$$

kde konceptuálne vektory $\mathbf{t}_{dat'(x,y,x)}$ a $\mathbf{t}_{vlastniť(y,x)}$ sú zostrojené analogickým spôsobom ako v (43). V prvom kroku musíme vykonať konkretizáciu implikácie $dat'(x, y, z) \Rightarrow vlastniť(y, z)$, tak že všeobecné premenné x, y, z sú substituované konkrétnymi premennými *Jano, Mariena, kniha*. Táto konkretizácia sa vykoná pomocou transformácie T , ktorá je špecifikovaná formulami (65-66)

$$\hat{\mathbf{t}}_1^{(1)} \approx \mathbf{t}_1^{(1)} \otimes T \quad (75)$$

kde

$$\hat{\mathbf{t}}_1^{(1)} = op \otimes impl + var_1 \otimes \mathbf{t}_{dat'(Jano, Mariena, kniha)} + var_2 \otimes \mathbf{t}_{vlastniť(Mariena, kniha)} \quad (76a)$$

$$T = \mathbf{t}_1^{*(1)} \otimes \hat{\mathbf{t}}_1^{(1)} \quad (76b)$$

Takto konkretizovaná reprezentácia $dat'(Jano, Mariena, kniha) \Rightarrow vlastniť(Mariena, kniha)$ v následnom kroku je použiteľná pre modus ponens uskutočnenej pomocou formúl (71-72)

$$\tilde{\mathbf{t}}_1^{(1)} \approx \hat{\mathbf{t}}_1^{(1)} \otimes T' \quad (77a)$$

kde rezultujúci konceptuálny vektor $\tilde{\mathbf{t}}_1^{(1)}$ už reprezentuje modus ponens, t.j. platí

$$\mathbf{t}_{dat'(Jano, Mariena, kniha)} \otimes \tilde{\mathbf{t}}_1^{(1)} \approx \mathbf{t}_{vlastniť(Mariena, kniha)} \quad (78)$$

Podobným spôsobom môžu byť uskutočniteľné aj ostatné tri zovšeobecnené modus ponens z obr. 16.

Ilustračný príklad –generalizácie pomocou indukcie

Uvažujme „tréningovú“ množinu tvorenú postupnosťou jednoduchých unárnych predikátory, ktoré môžeme interpretovať ako pozorovania

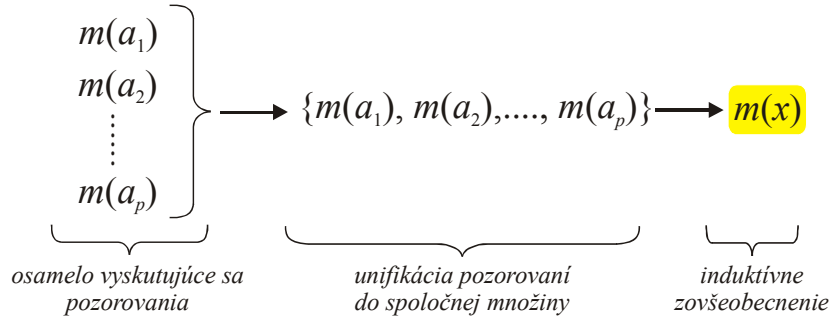
$$A_{train} = \{m(a_i); i = 1, 2, \dots, q\} \quad (79)$$

Naším cieľom bude zovšeobecniť tieto partikulárne predikáty do tvaru s univerzálnym kvantifikátorom (alebo v našom jednoduchšom formalizme, s univerzálnou premennou)

$$(\forall x)[m(x)] \equiv m(x) \quad (80)$$

To znamená, že jednotlivé partikulárne prípady $m(a_i)$ z tréningovej množiny (75) sú zovšeobecnené do formuly, ktorá nie je bezprostredne odvoditeľná z nich (pozri obr. 19)

$$m(a_1) \wedge \dots \wedge m(a_p) \stackrel{?}{\Rightarrow} m(x) \quad (81)$$



Obrázok 19. Znázornenie indukčného zovšeobecnenia, ktoré obsahuje tri etapy. V prvej etape máme izolované pozorovania $m(a_i)$, ktoré medzi sebou nesúvisia. V druhej etape izolovanosť jednotlivých pozorovaní je nahradená ich unifikáciou do jednej množiny, čo vyjadruje skutočnosť, že pozorovania nie sú izolované a nezávislé, ale majú niečo spoločné. V záverečnej tretej etape, unifikovaná forma pozorovaní je indukčným zovšeobecnená pomocou univerzálnej premennej x , ktorá reprezentuje triedu objektov majúcich rovnakú vlastnosť m .

Nech predikát $m(a_i)$ z tréningovej množiny A_{train} je holograficky reprezentovaný takto

$$\mathbf{t}_{m(a_i)} = \mathbf{rel} \otimes \mathbf{m} + \mathbf{m_var} \otimes \mathbf{a}_i \quad (82)$$

Tieto reprezentačné konceptuálne vektory môžu byť navzájom transformované pomocou transformačných vektorov $T_{i,j}$

$$\mathbf{t}_{m(a_j)} \approx \mathbf{t}_{m(a_i)} \otimes T_{i,j} \quad (83a)$$

$$T_{i,j} = \mathbf{t}_{m(a_i)}^* \otimes \mathbf{t}_{m(a_j)} \approx \mathbf{a}_i^* \otimes \mathbf{a}_j \quad (83b)$$

Definujme nový transformačný vektor \tilde{T}_i

$$\tilde{T}_i = \sum_{j=1}^q T_{i,j} = \mathbf{a}_i^* \otimes \mathbf{x} \quad (84a)$$

$$\mathbf{x} = \mathbf{a}_1 + \mathbf{a}_2 + \dots + \mathbf{a}_p \quad (84b)$$

kde konceptuálny vektor \mathbf{x} je priradený novému "zovšeobecnému" vektoru, ktorý reprezentuje každý argument a_i z tréningovej množiny A_{train} (ktorý je v našich ďalších úvahách o indukcii chápaný ako samostatná konceptuálna entita rovnocenná s reprezentáciami pôvodných objektov $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$)

$$\mathbf{t}_{m(a_i)} \otimes \tilde{T}_i \approx \mathbf{t}_{m(x)} \quad (85)$$

Táto formula môže byť interpretovaná ako indukčné zovšeobecnenie, kde partikulárne objekty a_i z tréningovej množiny v reprezentáciách $\mathbf{t}_{m(a_i)}$ sú substituované novým objektom x , ktorý môže byť interpretovaný ako nový univerzálny objekt.

9. Formálne jazyky

Vo štvrtej kapitole sme študovali použitie holografickej distribuovanej reprezentácie k štúdiu postupnosti symbolov, ktoré môžeme chápať ako najjednoduchší typ štruktúrovanej informácie. Majme konečnú abecedu $\mathcal{A} = \{a, b, \dots, p, q, \dots\}$, ktorý obsahuje malé písmena a ktoré sú nazývané terminálne symboly. Množina všetkých možných reťazcov zostrojených nad touto abecedou obsahuje je označená $L = \mathcal{A}^+ = \{a, b, \dots, aa, ab, \dots, aaa, aab, \dots, ppp, \dots\}$. Táto nekonečná (ale spočítateľná) množina môže byť rozdelená na dve disjunktné podmnožiny, $L = L_G \cup L_{NG}$, pričom stojíme pred problémom ako jednoducho charakterizovať reťazce z jednotlivých podmnožín. Formálne riešenie tohoto problému spočíva v definovaní *charakteristickej funkcie*, ktorá špecifikuje elementy podmnožín takto

$$\eta_G(x) = \begin{cases} 1 & (\text{ak } x \in L_G) \\ 0 & (\text{ak } x \in L_{NG}) \end{cases} \quad (88)$$

potom $L_G = \{x \mid x \in L \wedge \eta_G(x) = 1\}$. Tento prístup špecifikuje reťazce z podmnožiny L_G pomocou ich enumerácie, t.j. taxatívny vymenovaním, či reťazec $x \in L$ patrí do podmnožiny L_G alebo do L_{NG} .

Predpokladajme, že reťazce z L_G sú hierarchicky klastrované na dve alebo viac zložiek, ktoré navyše môžu byť do seba vnorené, pričom na záver tohoto procesu klastrovania získame jeden globálny klaster. Tento proces môže byť diagramaticky vyjadrený mnohými spôsobmi (pozri obr. 20, kde tri jednoduché spôsoby klastrovania sú znázornené). Prístup koreňového stromu z obrázku patrí medzi najjednoduchšie spôsoby formalizovania klastrovací proces reťazca na hierarchicky organizovanú štruktúru, jeho formalizácia môže byť vykonaná pomocou tzv. prepisovacích pravidiel. Tak napríklad, ilustračný príklad z prvého riadku obr. 20 tieto pravidlá majú tvar

$$\begin{aligned} P_1 &: S \rightarrow BC \\ P_2 &: B \rightarrow Ac \\ P_3 &: C \rightarrow ab \\ P_4 &: A \rightarrow ab \end{aligned} \quad (89)$$

kde veľké písmena sa nazývajú neterminálne symboly (volne povedané, tieto symboly reprezentujú „podklastre“, ktoré sú súčasťou finálneho globálneho klastra). Písmeno S sa nazýva „štartovací“ neterminál, je priradený hierarchicky najvyššiemu klastru, ktorý zahrňuje všetky terminálne symboly z daného reťazca). Použitím pravidiel z (89) môžeme postupne vytvoriť reťazec $x = (abcab)$ takto

$$S \rightarrow BC \left\{ \begin{array}{l} \rightarrow AcC \left\{ \begin{array}{l} \rightarrow abcC \\ \rightarrow Acab \end{array} \right\} \\ \rightarrow Bab \rightarrow Acab \end{array} \right\} \rightarrow abcab \quad (90)$$

We see that there exist three different ways how to construct the string $x = (abcab)$.

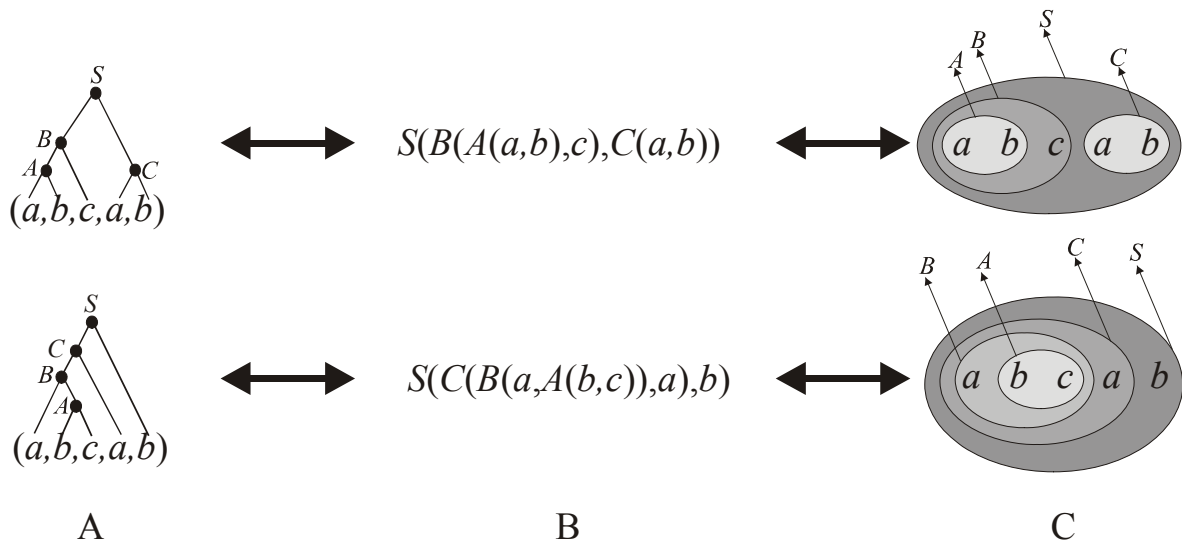


Figure 20. Two illustrative conversions of a sequence (a,b,c,a,b) into different „nested“ structures: (A) Rooted tree structure that aggregates simple sequence elements into higher-level hierarchic structures, (B) composite two argument functions, and (C) nested subsets encompass one or more symbols such that the resulting structure correspond to an aggregation of symbols.

The production system (89) may be considerably simplified such that we removed all nonterminal symbols that are redundant (in this specific case we may remove the nonterminal C , which is identical to A).

$$\begin{aligned}
 P_1 &: S \rightarrow BA \\
 P_2 &: B \rightarrow Ac \\
 P_3 &: A \rightarrow ab
 \end{aligned}
 \tag{91}$$

Moreover, the production system may be extended about one or more productions such that the construction of the original string $x = (abcab)$ is saved, but we may potentially construct many other new different strings $x \in L$, e.g.

$$\begin{aligned}
 P_1 &: S \rightarrow BA \\
 P_2 &: B \rightarrow Ac \\
 P_3 &: A \rightarrow ab \\
 P_4 &: A \rightarrow Aa
 \end{aligned}
 \left. \vphantom{\begin{aligned} P_3 \\ P_4 \end{aligned}} \right\} P_{34} : A \rightarrow ab | Ab
 \tag{92}$$

where new production $P_4' : A \rightarrow Aa$ introduces a generative feature of the resulting used scheme.

The above considerations may be generalized onto a concept called the *grammar* $[xx]$ (initially introduced by Noam Chomsky [20] almost 50 years ago) specified as follows

$$G = (N, T, S, P) \tag{93}$$

where N is a set composed of nonterminal symbols (in our above preliminary considerations represented by capital letters), $T \subset \mathcal{A}$ is a set composed of terminal symbols (represented by lower case letters), $S \in N$ is a *starting symbol* especially distinguished from elements of nonterminals N , and finally, P is a set composed of the so-called *productions* (often called also production rules or rewriting rules)

$$P = \{\alpha \rightarrow \beta\} \tag{94}$$

where α is a substring composed of terminal and at least one nonterminal symbols and β is a substring composed of terminal or nonterminal symbols. In particular, the set P contains a

production of the form $S \rightarrow \beta$, this production will initialize the procedure of creation of strings that are induced by the grammar G . These strings, initialized by the starting symbol S and composed entirely of terminal symbols form a subset denoted by L_G , which is called the *language*. It means that the characteristic function (88) is now unambiguously specified by the production set (94) such its elements are (1) composed only of terminal symbols and (2) its creation procedure is initialized the nonterminal starting symbol S .

We have seen that each grammatical string $x \in L_G$ may be interpreted by a rooted tree $t(x)$ called the *derivation tree*, where a top vertex is labeled by the starting symbol S , its leafs (terminal vertices) are labeled by terminal symbols, and all other remaining intermediate vertices are labeled by nonterminal symbols (cf. Fig. 20). Derivation trees have an extraordinary position in artificial intelligence and cognitive science [21,22], a semantic meaning of the grammatical string $x \in L_G$ is specified not only by the string itself but also by its derivation tree $t(x)$. If a string has two or more derivation trees, then we may say that it has more than one different semantic meanings. An actual semantic meaning of $x \in L_G$ will depend on the used derivation tree $t(x)$, which specifies an actual chunking of string symbols onto hierarchically higher clusters that contribute to the whole semantic meaning by a specific contribution.

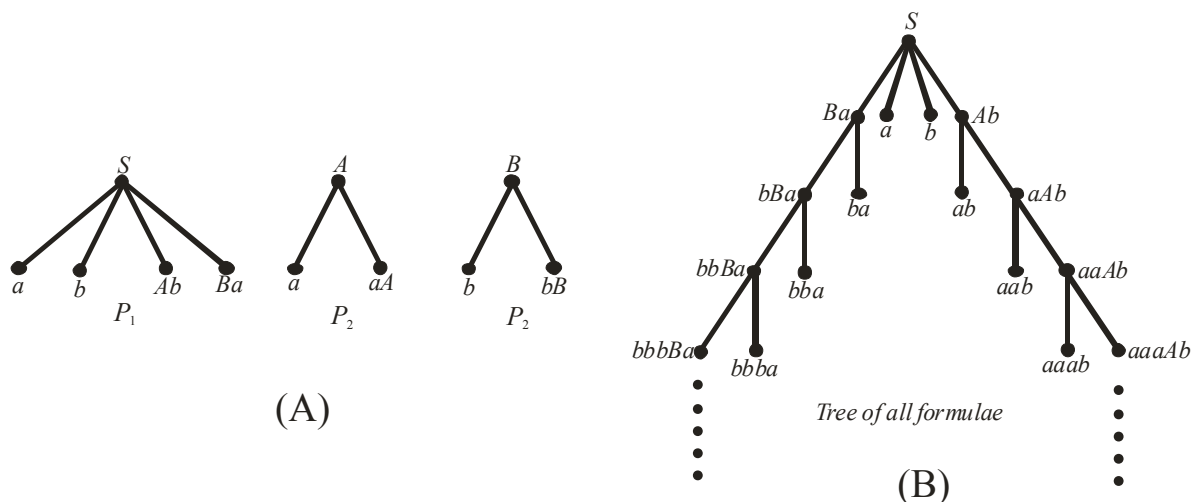


Figure 21. (A) Diagrammatic representation of productions (1).. (B) An infinite tree of all possible solutions – formulae.

Let us consider the following simple grammar $G = (N = \{S, A, B\}, T = \{a, b\}, S, P)$ with productions specified by

$$\begin{aligned}
 P_1 : S &\rightarrow a \mid b \mid Ab \mid Ba \\
 P_2 : A &\rightarrow a \mid aA \\
 P_3 : B &\rightarrow b \mid bB
 \end{aligned}
 \tag{95}$$

with productions diagrammatically visualized by These productions are diagrammatically visualized in Fig. 21, diagram A.

From fig. 21 we may construct all possible formulae that are specified by the productions (95)

$$\begin{aligned}
 L_G &= \{b^n a; n \geq 0\} \cup \{a^n b; n \geq 0\} = \\
 &= \{a, b, ab, ba, bba, aab, bbba, aaab, \dots\}
 \end{aligned}
 \tag{96}$$

For each grammatical formula there exists on tree of all formulae an oriented path beginning in the root of tree (labeled by a starting nonterminal S) and ending at a leaf evaluated by the respective formula, e.g.

$$\begin{aligned}
 a \in L_G &\Rightarrow S \rightarrow a \\
 b \in L_G &\Rightarrow S \rightarrow b \\
 ab \in L_G &\Rightarrow S \rightarrow Ab \rightarrow ab \\
 ba \in L_G &\Rightarrow S \rightarrow Ba \rightarrow ba \\
 aab \in L_G &\Rightarrow S \rightarrow Ab \rightarrow aAb \rightarrow aab \\
 bba \in L_G &\Rightarrow S \rightarrow Ba \rightarrow bBa \rightarrow bba
 \end{aligned}
 \tag{97}$$

The corresponding derivation trees are listed in Fig. 22.

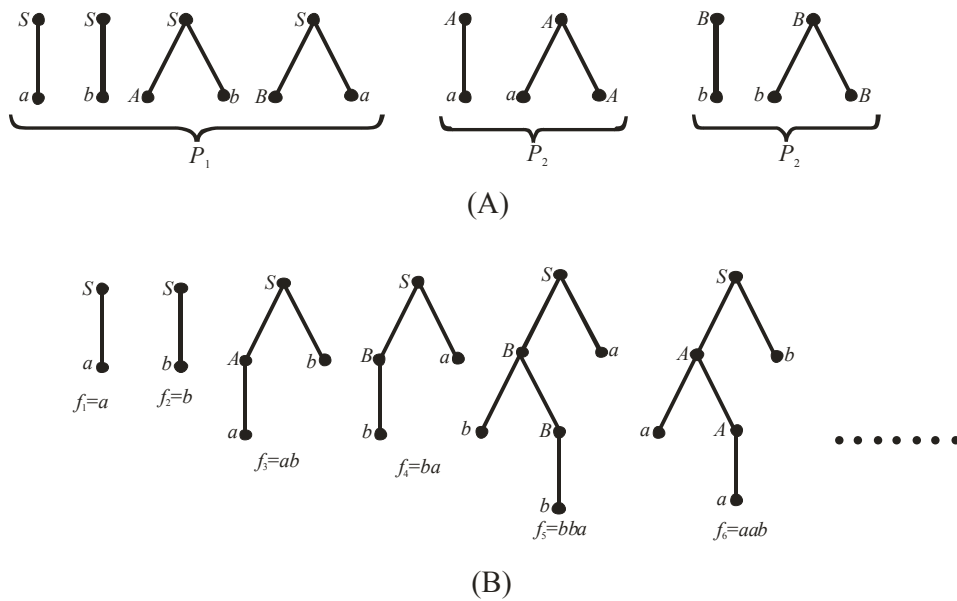


Figure 22. (A) Detailed diagrammatic representation of productions (1). (B) Derivation trees assigned to the first six formulae.

HRR representations of derivation trees in Fig. 22 are (cf. (44))

$$\begin{aligned}
 t_1 &= \mathbf{left} \otimes a \\
 t_2 &= \mathbf{left} \otimes b \\
 t_3 &= \mathbf{left} \otimes (A + \mathbf{left} \otimes a) + \mathbf{right} \otimes b \\
 t_4 &= \mathbf{left} \otimes (B + \mathbf{left} \otimes b) + \mathbf{right} \otimes a \\
 t_5 &= \mathbf{left} \otimes (B + \mathbf{left} \otimes b + \mathbf{right} \otimes (B + \mathbf{left} \otimes b)) + \mathbf{right} \otimes a \\
 t_6 &= \mathbf{left} \otimes (A + \mathbf{left} \otimes a + \mathbf{right} \otimes (A + \mathbf{left} \otimes a)) + \mathbf{right} \otimes b
 \end{aligned}
 \tag{98}$$

where we have omitted the vector representation \mathbf{S} of the nonterminal starting symbol, which appears as the first in each representation of derivation trees.

HRR representation of the first grammatical strings (96) are (cf. (34))

$$\begin{aligned}
f_1 &= \mathit{first} \otimes a \\
f_2 &= \mathit{first} \otimes b \\
f_3 &= \mathit{first} \otimes a + \mathit{second} \otimes b \\
f_4 &= \mathit{first} \otimes b + \mathit{second} \otimes a \\
f_5 &= \mathit{first} \otimes b + \mathit{second} \otimes b + \mathit{third} \otimes a \\
f_6 &= \mathit{first} \otimes a + \mathit{second} \otimes a + \mathit{third} \otimes b
\end{aligned} \tag{99}$$

The used holographic representations of derivation trees and string of symbols have a serious shortcoming; different t 's have great similarities between them (the same property is also true for vectors t 's), see Fig. 23. This fact causes that an immediate application of (28) to form an analogy of associative memory vector $T = t_1 \otimes f_1 + \dots + t_6 \otimes f_6$ could not be well performed, since this particular memory vector does not satisfy the following two important properties $t_i^* \otimes T \approx t_i$ and $f_i^* \otimes T \approx f_i$ that are of basic importance for correct recognition of T . Therefore we have to use slightly another approach to the construction of associative memory vector T based on the so-called *label vectors*.

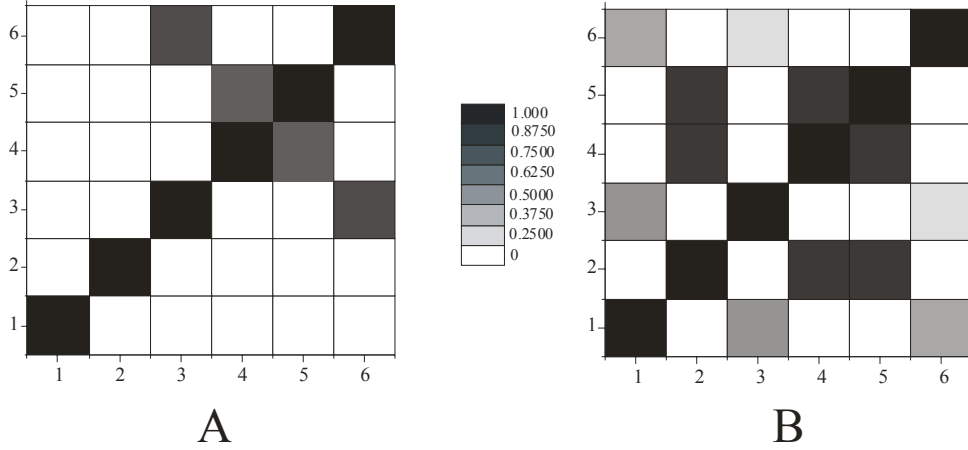


Figure 23. Graphic representation of similarities (A) between holographic distributed representations of derivation trees (98) and (B) between their terminal symbol sequences.

Let us have a set of six randomly generated label vectors x_1, x_2, \dots, x_6 , which will serve in our forthcoming considerations as labels for distinguishing of vector representations of derivation trees (that may be very similar from the standpoint of their mutual overlap specified by (21)). Applying these “label” vectors we construct the following new modified vectors

$$\tilde{t}_i = x_i \otimes t_i \quad \text{and} \quad \tilde{f}_i = x_i \otimes f_i \tag{100}$$

A transformation vector is determined as follow

$$T = \sum_{i=1}^6 \tilde{t}_i^* \otimes \tilde{f}_i \tag{101}$$

The modified vectors are specified by

$$\tilde{t}_i \otimes T \approx \tilde{f}_i \quad \text{and} \quad \tilde{f}_i \otimes T^* \approx \tilde{t}_i \tag{102}$$

for $i=1,2,\dots,6$. It means that the transformation vector T^* plays a role of parser, which assigns to a „formula” f the respective „derivation tree” t . In other words, we have „algebraized” the highly symbolic parsing process of formulae, which assigns derivation trees to formulae, where the trees are very important for a construction of semantic interpretations of the

formulae (i.e. the semantic meaning of formulae is specified not only by its terminal symbols but also by its respective derivation tree).

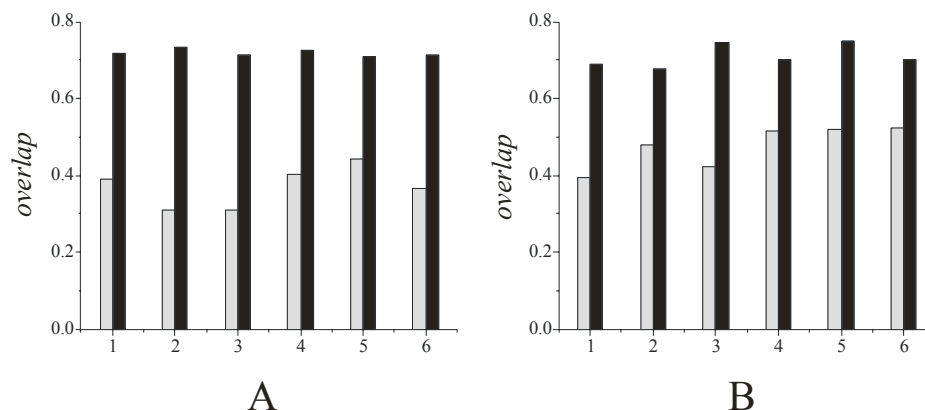


Figure 24. Reconstruction of holographic representations of derivation trees (A) and symbol sequences (B) induced by the grammar (95). The light/dark columns correspond to overlap between reconstructed and original vectors

10. Závěry

Holografická redukovaná reprezentácia ponúka nové nekonvenčné riešenie jedného zo základných problémov umelej inteligencie a kognitívnej vedy, ktorý spočíva v distribuovanom kódovaní štruktúrovanej informácie (postupnosť znakov, vnorené relačné štruktúry, a pod.). Použitá distribuovaná reprezentácia je založená na dvoch binárnych operáciách: unárnej operácii „involúcie“ a binárnej operácii „konvolúcia“ nad doménou n -rozmerých náhodne generovaných konceptuálnych vektorov, ktorých elementy spĺňajú normálnu distribúciu $N(0,1/n)$. Pomocou tejto distribuovanej reprezentácie môžeme modelovať rôzne typy asociatívnej pamäti, ktoré sú reprezentované konceptuálnym vektorom a taktiež dekódovať pamäťový vektor, t.j. určiť konceptuálne vektory, ktoré ho vytvorili. Táto rekognoskácia pamäťového vektora sa vykonáva pomocou procedúry čistenia (clean-up), ktorá nám určí pomocou vzájomného prekryvu vektor, ktorý je mu najviac podobný. Popísali sme taktiež proces agregácie vektorov, ktorý umožňuje prekonať ťažkosti rýchlej degradácie rekognoskácie pamäťového vektora tak, niekoľko vektorov sa združí do agregátu a tento sa zapamätáva prostredníctvom pamäťového vektora. Holografická redukovaná reprezentácia umožňuje hľadať podobnosť medzi dvoma štruktúrovanými konceptami pomocou jednoduchej algebraickej operácie skalárneho súčinu medzi ich distribuovanými reprezentáciami. Táto skutočnosť môže byť veľmi užitočnou pre modelovanie procesov prehľadávania pamäti na základe podobnosti (analógie) jej jednotlivých položiek. V záverečnej časti práce sme ukázali, že holografická redukovaná reprezentácia môže byť taktiež použitá k modelovaniu procesov usudzovania založených na pravidlách modus ponens a modus tollens, taktiež sme ukázali plodnosť tohto prístupu pri modelovaní procesu zovšeobecnenia pomocou indukcie.

Literatúra

- [1] Davis, P. J.: *Circulant Matrices*. Chelsea Publishing, New York, 1999.
- [2] Gabor, D.: Holographic model for temporal recall. *Nature*, **217** (1968), 1288–1289.

- [3] McClelland, J. L. M., Rumelhart, D. E., and the PDP research group (eds.): *Parallel distributed processing: Explorations in the microstructure of cognition*, volumes 1 and 2. The MIT Press, Cambridge, MA (1986).
- [4] Metcalfe Eich, J.: Levels of processing, encoding specificity, elaboration, and charm. *Psychological Review*, **92** (1985), 1–38.
- [5] Murdock, B. B.: A theory for the storage and retrieval of item and associative information. *Psychological Review*, **89** (1982), 316–338.
- [6] Plate, T. A.: *Distributed Representations and Nested Compositional Structure*. Department of Computer Science, University of Toronto. Ph.D. Thesis, 1994.
- [7] Plate, T. A.: *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*, CSLI Publications, Stanford, CA, 2003.
- [8] Plate, T. A.: Holographic Distributed Representations. *IEEE Transaction on Neural Networks*, **6** (1995), 623-641.
- [9] Slack, J. N.: The role of distributed memory in natural language processing. In O'Shea, T. (ed.): *Advances in Artificial Intelligence: Proceedings of the Sixth European Conference on Artificial Intelligence, ECAI-84*. Elsevier Science Publishers, New York, 1984.
- [10] Smolensky, P., Legendre, G., Miyata, Y.: *Principles for an Integrated Connectionist/Symbolic Theory of Higher Cognition*. Technical Report CU-CS-600-92, Department of Computer Science and 92-8, Institute of Cognitive Science. University of Colorado at Boulder, 1992, 75 pages.
- [11] Smolensky, P.: On the proper treatment of connectionism. *The Behavioral and Brain Sciences*, **11** (1988), 1-74.
- [12] Smolensky, P.: Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, **46** (1990), 159-216.
- [13] Smolensky, P.: Neural and conceptual interpretations of parallel distributed processing models. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models*. MIT Press - Bradford Books, Cambridge, MA, 1986, pp. 390-431.
- [14] Willshaw, D. J., Buneman, O. P., and Longuet-Higgins, H. C.: Non-holographic associative memory. *Nature*, **222** (1969), 960–962.
- [15] Shastri, L., Ajjanagadde, V.: From simple associations to systematic reasoning: connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, **16** (1993), 417-494.
- [16] Kanerva, P.: The Spatter Code for Encoding Concepts at Many Levels. In Marinaro, M., Morasso, P. G. (eds.): *ICANN '94, Proceedings of the International Conference on Artificial Neural Networks*. Springer Verlag, London, 1994, pp. 226–229.
- [17] Kanerva, P.: Binary Spatter-Coding of Ordered K-tuples. In von der Malsburg, C., von Seelen, W., Vorbruggen, J. C., Sendhoff, B. (eds.): *ICANN'96, Proceedings of the International Conference on Artificial Neural Networks*. Springer Verlag, Berlin, 1996, pp. 869-873.
- [18] Kanerva, P.: Analogy as a Basis of Computation, In Uesaka, Y., Kanerva, P., Asoh, H.: *Foundations of real-world intelligence*. CSLI Publications, Stanford, CA, 2001.

- [19] Fodor, J.A., Pylyshyn, Z.W.: Connectionism and cognitive architecture: A critical analysis., *Cognition*, **28** (1988) 3-71.
- [20] Chomsky, N.: Three models for the description of language. *IEEE Trans. Info. Theory* **2** (1956), 113 - 124.
- [21] Johnson – Laird, P.N.: *Mental Models. Toward a Cognitive Science of Language, Inference, and Consciousness*. Cambridge University Press, Cambridge 1983.
- [22] Russel, S., Norvig, P.: *Artificial Intelligence. A Modern Approach*. Prentice Hall, New Jersey 1995.