



FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITY KOMENSKÉHO
ÚSTAV INFORMATIKY

Učenie bezkontextového jazyka $L_{a^n b^n}$
pomocou rekurentnej BCM neurónovej siete

Diplomová práca

Vypracoval: Svorad Štolc Jr.
Školiteľ: Igor Farkaš, PhD

BRATISLAVA

2001

Čestne prehlasujem, že predkladanú prácu som vypracoval samostatne
s použitím literatúry uvedenej v zozname.

.....

Bratislava, apríl 2001.

Obsah

1	Úvod do problematiky	3
1.1	Motivácia	3
1.2	Ekvivalencia formálnych automatov a neurónových sietí	5
1.2.1	Konečné automaty	5
1.2.2	Zásobníkové automaty	5
1.2.3	Turingove stroje	6
1.3	Presnosť aritmetiky	6
1.4	Prístupy zhora-nadol a zdola-nahor	6
1.5	Modely učenie pomocou učiteľa	7
1.5.1	Rekurentná autoasociatívna pamäť	7
1.5.2	Jednoduché rekurentné siete	7
1.5.3	Jednoduché rekurentné siete a genetické algoritmy	8
1.6	Samoorganizácia	8
1.6.1	Samoorganizujúce sa sémantické mapy	9
1.6.2	Predikcia komplexných postupností pomocou rekurentnej BCM siete	10
2	Teoretické podklady	12
2.1	Otázky a východiská	12
2.2	Rekurentná BCM sieť	13
2.2.1	Architektúra siete	13
2.2.2	BCM učenie	14
2.3	Predikcia vs. akceptácia	17
2.3.1	Pravdepodobnostné predikčné modely	18
2.3.2	Konštrukcia predikčného modelu	19
2.3.3	Miery úspešnosti predikčných modelov	19
2.4	Analýza dynamiky siete	21
2.4.1	Linearizačná metóda	22

<i>OBSAH</i>	2
3 Experiment	24
3.1 Trénovacia a testovacia množina	24
3.2 Parametre tréovania	25
3.3 Výsledky	25
4 Analýza stavového priestoru	30
4.1 Sieť 1: Cyklická dynamika	30
4.2 Sieť 2: „Cik-cak” dynamika	36
5 Diskusia	42
5.1 Akceptácia	42
5.2 Generalizácia	43
5.3 Záver	43
A Definícia dynamického systému	47

Kapitola 1

Úvod do problematiky

1.1 Motivácia

Často diskutovanou témou v oblasti umelej inteligencie je spracovanie prirodzeného jazyka. Ide o netriviálny mechanizmus analýzy časovo závislých dát so zložitou vnútornou štruktúrou. V súčasnosti existuje viacero parciálnych riešení, ktoré sú založené na klasických symbolických metódach (vyžívajúcich teóriu formálnych jazykov a automatov), alebo na distribuovaných subsymbolických mechanizmoch (napr. umelé neurónové siete či kolónie agentov). Štruktúra viet prirodzeného jazyka je vo všeobecnosti zložitejšia ako lineárne usporiadanie slov. Dokonca je možno nájsť jazyky, ktorých generatívna sila presahuje možnosti bezkontextových gramatík.

Klasickým prístupom k spracovaniu syntaxe prirodzeného jazyka sú konečné automaty (Finite State Automata, FSA) (príp. ich rozšírenia). Spracovanie prebieha na symbolickej báze, výpočet je determinovaný diskretnými vnútornými stavmi a závislosťami prechodov medzi nimi. Takýto prístup možno ľahko formalizovať, avšak jeho plasticita je značne obmedzená.

V kontraste ku klasickým prístupom, paralelné distribuované spracovanie (Parallel Distributed Processing, PDP) ponúka radikálne odlišný pohľad na problémovú oblasť. Ľudský kognitívny aparát sa z biologického pohľadu podobá skôr distribuovanému systému pracujúcemu na paralelnej báze, ako sekvenčnému symbolickému systému. Neuronálne štruktúry v mozgu sa vyznačujú mohutnou plasticitou a robustnosťou, čo je v kontradikcii s klasickými prístupmi pomocou FSA. Rekurentné neurónové siete (Recurrent

Neural Networks, RNN) sú typickým príkladom distribuovaných systémov schopných spracovávať časové sekvencie. Na RNN možno tiež hľadiť ako na nelineárny dynamický systém so spojitým priestorom vnútorných stavov a diskretným časom. Z tohto uhla pohľadu je sieť spracúvajúca časové vzory príkladom dynamického rozpoznávača [13].

V súvislosti s neurónovými sieťami (NN), ako mechanizmom vhodným na spracovávanie lingvistických dát, je tiež potrebné zaoberať sa metódou učenia, čiže spôsobom modifikácie vnútorných váh siete. Klasickým, dobre prebádaným prístupom, je učenie pomocou učiteľa. Založené je obyčajne na algoritmoch minimalizujúcich chybovú funkciu siete, váhy sú modifikované v smere klesajúceho gradientu chyby. Existuje tu štatút učiteľa, ktorého úlohou je usmerňovať učenie „správnym“ smerom, t.j. smerom vedúcim ku (globálnemu) minimu chyby. Druhou veľkou triedou učiacich mechanizmov sú metódy založené na princípoch samoorganizácie. Učenie prebieha bez explicitnej informácie o správnosti výstupov, výhradne na základe štatistických vlastností vstupných dát. Tento prístup je motivovaný biologickou realitou, ktorá sa vzťahuje na kortikálne oblasti ľudského mozgu. Mozgová kôra vykazuje zaujímavé črty priestorového usporiadania (tzv. mapy). Existujú tu totiž oblasti, zachytávajúce topologickým spôsobom podobnosť či súvislosti, medzi jednotlivými podnetmi (napr. tónovo blízke stimuly vyvolávajú aktivitu v blízkych zhlukoch neurónov). Čo je zaujímavé, tieto mapy sa formujú spontánne, samoorganizovane, iba na základe externých podnetov.

V tejto práci sa budeme bližšie zaoberať vlastnosťami konkrétneho rekurentného neuronálneho modelu učiaceho sa bez učiteľa, konkrétne rekurentnou BCM sieťou s laterálnou inhibíciou (Recurrent BCM Network, RBCM) [18]. Jednoduchý model BCM neurónu bol navrhnutý so zámerom modelovať správanie vizuálnej kôry [2]. Jeho kľúčovou vlastnosťou je dynamický prah ovplyvňujúci zmenu synaptických váh. Ukázalo sa [18], že RBCM sieť je schopná predikovať isté, nie triviálne, časové sekvencie s porovnateľnou úspešnosťou ako jednoduché rekurentné siete učené tradičným spôsobom. Možno teda predpokladať, že modely učené bez učiteľa majú podobnú silu ako modely učené pomocou neho.

Vnútorné reprezentácie vytvorené rôznymi učiacimi mechanizmami sa môžu navzájom značne líšiť. Častou metódou popisu vnútorných reprezentácií je analýza dynamiky siete, použitím aparátu dynamických systémov.

V prácach [15] a [19] boli detailne analyzované niektoré z typických reprezentácií určitých počítačových jazykov. Ukázalo sa, že úspech trenovania závisí od rôznych faktorov. Za istých okolností však tréning dospel k uspokojivým výsledkom.

Ambíciou tejto práce je popísať povahu vnútorných reprezentácií jazyka $L_{a^n b^n}$ vyvinutých RBCM sieťou, čiže rekurentným modelom učným bez učiteľa.

1.2 Ekvivalencia formálnych automatov a neurónových sietí

1.2.1 Konečné automaty

Podľa teórie formálnych jazykov [6] sú konečné automaty schopné spracovávať nanajvýš triedu regulárnych jazykov (\mathcal{R}). Vyplýva to z konečného počtu vnútorných stavov a symbolov pracovnej abecedy. Možno tiež ukázať ekvivalenciu medzi deterministickým a nedeterministickým FSA [6]. Rekurentnú neurónovú sieť ekvivalentnú ku deterministickému automatu o n stavoch a m symboloch možno konštruktívne zostrojiť pomocou $\mathcal{O}(n)$ neurónov a $\mathcal{O}(mn)$ váh [11]. Ukázalo sa, že siete tréňované na predikciu regulárnych jazykov sú schopné simulovať i činnosť nekonečných automatov.

1.2.2 Zásobníkové automaty

Jazyk $L_{a^n b^n}$, čiže jazyk „centrálne uzátvorkovaných výrazov“, však nepatrí do triedy \mathcal{R} , ale do triedy bezkontextových jazykov \mathcal{L}_{CF} . Preto je na jeho akceptáciu nutný minimálne zásobníkový automat (Push-Down Automaton, PDA). Automaty tohoto druhu sa vyznačujú externým médiom typu FIFO, nekonečnej hĺbky, na ktoré možno počas výpočtu ukladať a spätne z neho čítať pomocné informácie. Podobne ako konečné automaty, PDA možno priamo konvertovať na ekvivalentnú rekurentnú sieť. Celý neuronálny zásobníkový automat možno rozdeliť na tri prepojené moduly. Prvý zodpovedá riadiacemu konečnému automatu, ďalšie dva realizujú funkcie zásobníka („push“ a „pop“).

1.2.3 Turingove stroje

Najsilnejším z tradičných výpočtových modelov je Turingov stroj (TS), akceptujúci triedu rekurzívne vyčísliteľných jazykov \mathcal{L}_{RE} . TS sa považuje za všeobecný formalizmus algoritmu. Jeho sila je väčšia ako sila súčasných počítačov, práve z dôvodu neobmedzenosti výpočtovej pamäte. Výskum v oblasti dynamických systémov a RNN ukázal, že možno zostrojiť sieť simulujúcu univerzálny Turingov stroj [17]. V dôkaze sa využíva možnosť uložiť nekonečne veľkú informáciu do stavového priestoru a dynamiky siete.

1.3 Presnosť aritmetiky

Problematická situácia nastáva v prípade, keď presnosť aritmetiky, využívanej v simulačnom prostredí, prestane byť postačujúca. Konštruované neurónové siete potom prestávajú byť ekvivalentnou alternatívou ku klasickým symbolickým systémom. Z istého pohľadu sú však vzniknuté omyly prínosom pre celý systém. Prirodzená inteligencia totiž nie je fenomén skladajúci sa výhradne z ideálnych racionálnych procesov. Dalo by sa dokonca povedať, že omylnosť či emócia sú esenciálnymi črtami prirodzenej inteligencie.

1.4 Prístupy zhora-nadol a zdola-nahor

Doteraz sme prezentovali predovšetkým konštrukčné prístupy zhora-nadol. Presnejšie ide o budovanie ekvivalentných neurónových sietí k formálnym automatom riešiacim istý problém na symbolickej báze. Čo však v prípade, keď takýto algoritmus z akýchkoľvek dôvodou nepoznáme? Učenie siete na príkladoch, so zámerom extrahovať podstatné črty tréningovej vzorky, nazývame konštrukciou zdola-nahor. Je zrejmé, že tento prístup nedovoľuje vytvárať ideálne neuronálne modely symbolických formalizmov (na rozdiel od prístupu zhora-nadol), pracuje totiž na báze pozorovania kladných a záporných prípadov istého fenoménu. Prístup zdola-nahor však umožňuje formuláciu „domnienok“ na základe neúplnej informácie, ich reformuláciu v okamihu príchodu nových eventualít (črta často využívaná v OCR softwaroch a i.). Tento prístup je biologicky prijateľnejší a častokrát i efektívnejší. Použitím správne zvolených parametrov učenia možno eliminovať zašumenie

vstupných dát, ktoré by sa potenciálne mohlo prejavíť zložitou sieťou (počtom neurónov či mohutnosťou ich prepojení).

1.5 Modely učené pomocou učiteľa

Pozrime sa teraz konkrétnejšie na niektoré rekurentné modely učené pomocou učiteľa. Zaoberať sa budeme najmä sieťami pracujúcimi nad časovými postupnosťami, špeciálne prediktormi nasledujúceho symbolu sekvencie.

1.5.1 Rekurentná autoasociatívna pamäť

Prvým modelom siete trénovanej metódou zdola-nahor je rekurentná autoasociatívna pamäť (RAAM) [10]. Autor pôvodne navrhol tento model na spracovávanie stromovitých štruktúr. Sieť sa skladá z dvoch oddelených, navzájom inverzných častí: kodér a dekodér. Pôvodne boli obidve časti trénované separátne, no neskôr autor upustil od učenia kodéra, ktorý konštruoval analyticky, ako inverziu dekodéra. Dekodér pozostáva z dvoch vrstiev, n vstupných a $2n$ výstupných neurónov. Vrstvy sú navzájom úplne prepojené použitím $2n^2$ synaptických spojení. Ako aktivačná funkcia neurónov je použitá bežná sigmoidálna funkcia¹.

V priebehu jedného kroku kódovania je $2n$ vstupných neurónov redukovaných na výstup o šírke n neurónov. Kódovanie pozostáva z viacerých redukčných iterácií. Pri dekódovaní je proces opačný, vstup siete šírky n je expandovaný na výstup o veľkosti $2n$.

Optimálne váhy dekodéra Pollack zisťuje pomocou horolezeckého (hill climbing) algoritmu. Takto natrénovaná sieť vie zakódovať a opäť dekódovať sekvencie dĺžky max. 5 symbolov.

1.5.2 Jednoduché rekurentné siete

Ďalším modelom je jednoduchá rekurentná sieť (Simple Recurrent Network, SRN) [4]. Tento model je relatívne silným nástrojom na spracovanie triedy bezkontextových jazykov \mathcal{L}_{CF} . SRN sa skladá z troch vrstiev: vstupnej, kontextovej a výstupnej. Vstupná vrstva je zložená z dvoch odlišných častí, prvou je externý vstup, druhou sú kópie aktivít kontextových neurónov z predošlého kroku výpočtu. Jednotlivé vrstvy pozostávajú z $(m + n)$, n a q

¹Sigma funkcia: $\sigma(x) = \frac{1}{1+e^{-x}}$

neurónov. Aktivita skrytej vrstvy je v každom kroku kopírovaná na príslušný počet vstupných neurónov. Na zvyšné neuróny vstupnej vrstvy je privádzaný externý vstup siete. Tretia, výstupná vrstva pôsobí ako klasifikátor aktivít strednej kontextovej vrstvy. Bežne používanou metódou tréovania SRN je gradientový algoritmus BPTT (Backpropagation Through Time).

Rodriguez [15] tréoval SRN pomocou BPTT na predikciu nasledujúceho symbolu slov z $L_{a^n b^n}$. Natréovaná sieť bola schopná úspešne predikovať slová dĺžky až 32 znakov, hoci najdlhšie slovo tréovacej množiny bolo dlhé 22 znakov ($\leq a^{11}b^{11}$).

1.5.3 Jednoduché rekurentné siete a genetické algoritmy

Zaujímavé výsledky v oblasti SRN a jazyka $L_{a^n b^n}$ boli získané použitím genetického učiaceho algoritmu (GA) [3]. Na rozdiel od BPTT učenia, v stavovom priestore vznikali viac než dva pevné body. Schopnosť zovšeobecnenia u sietí tréovaných pomocou GA tiež podstatne stúpla. Možno teda predpokladať, že systémy s komplexnejšou dynamikou majú lepšie prediktívne schopnosti aj v nepresných simulačných podmienkach.

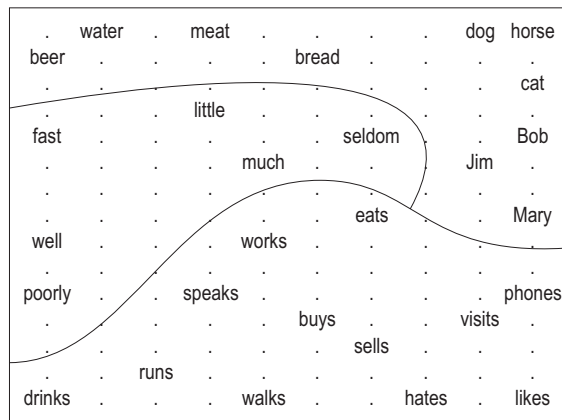
1.6 Samoorganizácia

Učenie bez učiteľa, inak povedané samoorganizácia, je vo väčšine prípadov založená na rôznych verziach Hebbovho učiaceho pravidla. Pôvodne bolo toto pravidlo formulované ako univerzálny mechanizmus, na základe ktorého prebiehajú všetky synaptické modifikácie v biologických neuronálnych štruktúrach. Matematicky ho možno vyjadriť ako

$$\frac{\partial w_{ij}}{\partial t} = \alpha c_i(t) c_j(t), \quad (1.1)$$

kde ľavá strana predstavuje zmenu synaptickej váhy medzi neurónmi i a j , α je skalár určujúci rýchlosť učenia a $c_i(t)$ aktivácia daného neurónu. Neformálne možno povedať, že spojenia medzi synchronne aktivovanými neurónmi silnejú, medzi asynchrónne aktivovanými neurónmi spojenia slabnú.

Samoorganizujúce siete sa vyznačujú takzvanými laterálnymi, bočnými spojeniami prepájajúcimi neuróny jednej vrstvy navzájom. Laterálne spoje



Obr. 1.1: Sémantická mapa vytvorená sieťou typu SOM.

vo všeobecnosti delíme na excitačné a inhibičné. Topológia siete, určená rozložením bočných spojení, determinuje typ samoorganizácie. Obyčajne má intenzita laterálnych spojení z pohľadu každého neurónu tvar tzv. „mexického klobúka“ (t.j. blízke neuróny sa excitujú, vzdialenejšie sa inhibujú, ešte vzdialenejšie neuróny nemajú medzi sebou väzbu).

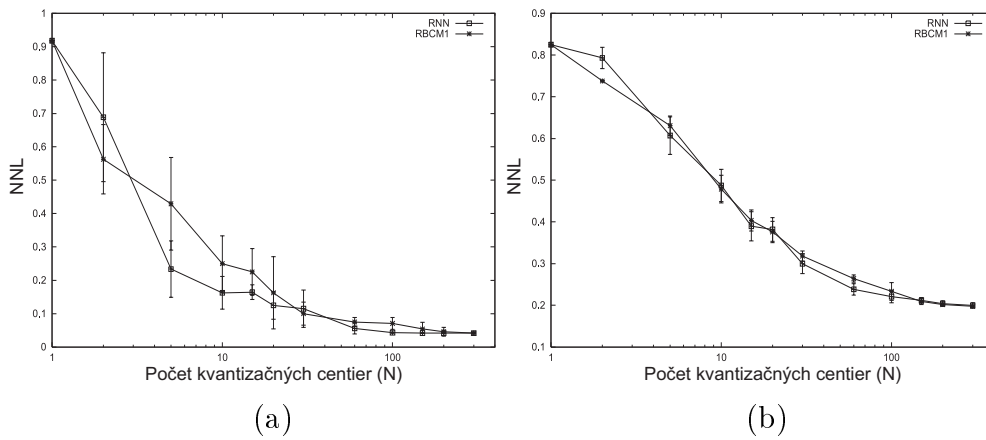
V nasledujúcej časti popíšeme dva modely pracujúce na princípoch samoorganizácie. Zamerané sú na syntaktickú analýzu rôzne zložitých časových sekvencií.

1.6.1 Samoorganizujúce sa sémantické mapy

Samoorganizujúca sa sémantická mapa [14] nepatrí medzi mechanizmy založené na rekurentnom kontexte. Ide o tradičnú SOM architektúru (Self-Organizing Map), ktorej sú predkladané časové sekvencie slov vo vhodnej forme. Predstavme si množinu jednoduchých prirodzených viet tvaru podmet-prísudok-predmet, alebo podmet-prísudok-prívlastok. Napr.:

Mary likes meat
 Jim speaks well
 dog drinks fast
 cat hates Jim
 ⋮

Slová boli sieti prezentované vo forme spriemerneného kontextu. V rámci vety je kontext slova definovaný priamym predchodcom a priamym



Obr. 1.2: NNL v závislosti od počtu kvantizačných centier (a) pre Feigenbaumovú postupnosť, (b) pre laser. Grafy hovoria o úspešnosti RBCM siete v porovnaní s klasickou SRN [18].

následovníkom. Na identifikáciu slov bolo použité lokálne kódovanie (napr. $[0,1,0,0]$), čiže každé z n slov použitého slovníka je reprezentované vo forme n -rozmerného vektora s jedinou dimenziou nastavenou na 1. Kontext slova v rámci určitej vety možno vyjadriť pomocou $2n$ -rozmerného vektora (predchodca, následovník). Spriemernený kontext sumarizuje kontexty určitého slova vo všetkých použitých vetách. Sieť počas tréningu vytvorila mapu slovníka, zachytávajúcu vzťahy jednotlivých pojmov, postrehnutelné zo syntaxe viet (pozri obr. 1.1).

1.6.2 Predikcia komplexných postupností pomocou rekurentnej BCM siete

Model RBCM siete obsahuje rekurentnú kontextovú väzbu umožňujúcu spracovanie časových symbolických sekvencií. Architektúra RBCM siete je detailne popísaná v časti 2.2.

Pravdepodobnostný predikčný model vybudovaný na výstupných aktivitách RBCM siete bol testovaný na predikciu reťazcov Feigenbaumovej postupnosti² a sekvencie generovanej laserom v chaotickom režime [18]. Ukázalo sa, že rekurentné BCM siete sú schopné predikovať obidve netriviálne postupnosti s porovnateľnou úspešnosťou ako SRN učené tradičným

²Feigenbaumová postupnosť: ($w_0 = b$; $w_1 = ba$; $w_n = w_{n-1}w_{n-2}w_{n-2}$; $n > 1$)

spôsobom (pozri obr. 1.2).

Kapitola 2

Teoretické podklady

2.1 Otázky a východiská

Väčšina prác orientovaná na problematiku spracovania prirodzeného jazyka na báze konektivismu využíva modely neurónových sietí učených pomocou učiteľa. Istá pozornosť sa venovala i analýze dynamiky takto trénovaných modelov [3, 15]. Výrazne menej prebádanou oblasťou sú rekurentné samoorganizujúce sa modely, čiže rekurentné modely učené bez učiteľa. Naším zámerom je osvetliť niektoré aspekty učenia pomocou samoorganizácie, hlavne v súvislosti s jednoduchými symbolickými štruktúrami známymi z teórie formálnych jazykov [6]. Oblasti, ktoré nás budú v práci bližšie zaujímať, možno sformulovať do nasledujúcich niekoľko otázok:

1. Je možné aby sa samoorganizujúca sieť, naučila spracovávať jednoduchý bezkontextový jazyk, konkrétne pochopila podstatu počítadla?
2. Aká je generalizačná schopnosť takýchto sietí a čím je jej úspešnosť podmienená?
3. Čím sa vyznačujú vnútorné sieťové štruktúry vyvinuté trénovaním?
4. Čím sa tieto štruktúry odlišujú od klasických SRN učených pomocou učiteľa?

Prvá otázka vyžaduje vybudovanie mechanizmu umožňujúceho akceptáciu slov daného jazyka. Vnútorná reprezentácia siete by mala dovoliť jasne

separovať finálny akceptačný stav, „rovnaký“ pre všetky slová daného jazyka. Práve od tohto faktoru závisí odpoveď na druhú otázku. Ak je separovateľnosť jednotlivých stavov dobrá, teda ak vieme jednoznačne detekovať akceptačný stav systému, je pravdepodobné, že slová jazyka, nezahrnuté do tréningovej množiny, budú sieťou úspešne spracované.

Bežnou metódou skúmania vnútorných reprezentácií je analýza dynamiky stavového priestoru siete. Rekurentnú neurónovú sieť možno tiež považovať za dynamický systém, ktorého dynamika, pohyb vo fázovom priestore, popisuje vnútornú reprezentáciu či štruktúru problému. Odpoveď na tretiu a štvrtú otázku sa pokúsime nájsť bližšou analýzou dynamík dvoch konkrétnych sietí.

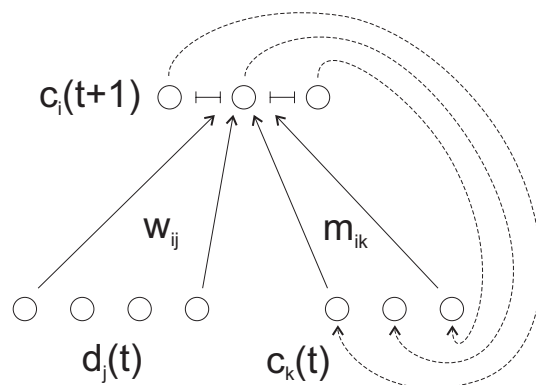
Rozsah nastolených otázok však široko presahuje rámec tejto práce, preto sa v ďalšom texte obmedzíme na konkrétny jazyk, sieťovú architektúru i učenie. Ako reprezentanta jednoduchého deterministického bezkontextového jazyka sme vybrali počítadlový jazyk $L_{a^n b^n}$. Na simuláciu sme použili samoorganizujúci rekurentný model schopný spracovávať časové sekvencie, totiž rekurentná sieť typu BCM. Sieť bola učená na náhodných sekvenciách slov z $L_{a^n b^n}$, ktoré jej boli predkladané po symboloch. Nad natrénovanou sieťou sme vybudovali predikčný model, ktorého cieľom bolo zachytiť mieru úspešnosti vnútornej reprezentácie.

2.2 Rekurentná BCM sieť

V tejto časti detailnejšie popíšeme použitý neuronálny model, tiež zdefinujeme samoorganizujúce učenie typu BCM.

2.2.1 Architektúra siete

RBCMN vznikne rozšírením tradičnej BCM architektúry o rekurentnú (kontextovú) väzbu [18, 1]. Sieť pozostáva z dvoch vrstiev, z vrstvy vstupnej a výstupnej. Vstupná vrstva sa skladá z externých vstupov a kópií aktivity výstupnej vrstvy z predošlého časového kroku. Neuróny výstupnej vrstvy sú navzájom úplne laterálne prepojené konštantnými inhibičnými spojeniami (pozri obr. 2.1). Spojenia tohto typu zohrávajú kľúčovú úlohu pri samoorganizácii neurónov výstupnej vrstvy vzhľadom na vstupnú selektivitú. Aktivitu i -teho výstupného neurónu v čase $t + 1$ možno formálne vyjadriť



Obr. 2.1: Rekurentná BCM sieť prvého rádu s laterálnou inhibíciou dopredného typu. Sieť pozostáva zo vstupnej vrstvy (externý vstup $\vec{d}(t)$ a aktivity výstupnej vrstvy $\vec{c}(t)$ v čase t) a z výstupnej vrstvy (nové výstupné aktivity $\vec{c}(t+1)$). Neuróny výstupnej vrstvy sú úplne laterálne prepojené konštantnými inhibičnými spojeniami

vzťahom

$$c_i(t+1) = \sigma \left(\left[\vec{w}_i(t) - \mu \sum_{j \neq i} \vec{w}_j(t) \right] \otimes \vec{d}(t) + \left[\vec{m}_i(t) - \mu \sum_{j \neq i} \vec{m}_j(t) \right] \otimes \vec{c}(t) \right), \quad (2.1)$$

kde μ je kladná konštanta vyjadrujúca silu laterálnej inhibície, σ je tradičná sigmoidálna aktivačná funkcia, ktorej strmosť a obor hodnôt podstatne ovplyvňujú učiace schopnosti siete. Pod operáciou \otimes rozumieme skalárne násobenie dvoch vektorov¹. Použitá laterálna inhibícia je tzv. dopredného typu, pri výpočte aktivity neurónu sa využíva výhradne vnútorná aktivita bočných neurónov, nie však ich skutočný výstup² (pozri (2.1)) [8].

2.2.2 BCM učenie

BCM učenie patrí do triedy mechanizmov, ktoré nevyžadujú ku svojej činnosti učiteľa. Založené je na modifikovanom Hebbovom pravidle. Učenie neprebíha na laterálnej úrovni, modifikujú sa iba aferentné spoje. Zmena Hebbovho pravidla spočíva predovšetkým v zavedení pohyblivého prahu synaptickej modifikácie θ_M^i , ktorá určuje hranicu medzi posilňovaním a oslabovaním synaptických spojení.

¹skalárny súčin vektorov: $\vec{a} \otimes \vec{b} = \sum_i a_i b_i$

²výstup neurónu = σ (vnútorná aktivita)

Formálne je BCM učenie odvodené z gradientovej minimalizácie celkovej stratovej funkcie $L(t) = \sum L_i(t)$ [8], kde

$$L_i(t) = - \left(\frac{1}{3} c_i^3(t) - \frac{1}{4} E(c_i^2(t)) c_i^2(t) \right). \quad (2.2)$$

Funkcia $E(c_i^2(t)) = \theta_M^i(t)$ je váženým priemerom druhých mocnín aktivít za istý časový interval τ .

$$\theta_M^i(t) = \frac{1}{\mathcal{A}\tau} \int_{-\infty}^t c_i^2(t') e^{-\frac{t-t'}{\tau}} dt' \quad (2.3)$$

Škálovací koeficient \mathcal{A} nazývame konštantou efektívneho prahu $\theta_M^i(t)$. Zo vzťahu (2.3) vidno, že dôležitosť aktivít exponenciálne klesá smerom do minulosti. Druhé mocniny v sume spôsobujú, že θ_M^i nikdy neklesne pod nulu. V našom simulačnom prostredí bol však použitý diskretizovaný modifikačný prah $\theta_{Md}^i(t)$, ktorého časové parametre t a τ môžu nadobúdať výlučne celočíselné hodnoty.

$$\begin{aligned} \theta_{Md}^i(t) &= \frac{1}{\mathcal{A}\lambda} \sum_{k=0}^{\tau-1} c_i^2(t-k) e^{-\frac{k}{\tau}} \\ \lambda &= \sum_{k=0}^{\tau-1} e^{-\frac{k}{\tau}} \end{aligned} \quad (2.4)$$

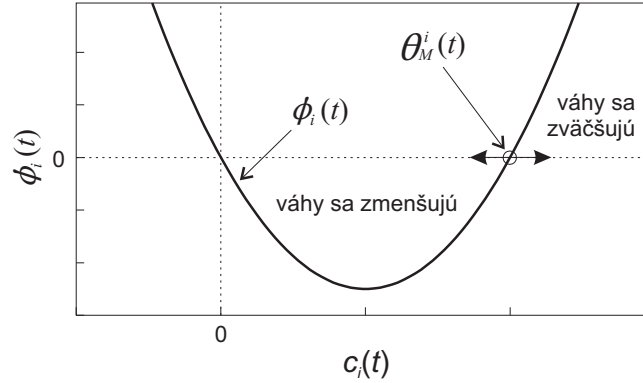
Deriváciou stratovej funkcie podľa jednotlivých váh dostávame základné pravidlo učenia, teda pravidlo podľa ktorého sa modifikujú jednotlivé aferentné a kontextové váhy.

$$\frac{\partial w_{ij}(t+1)}{\partial t} = -\eta \frac{\partial L(t+1)}{\partial w_{ij}(t)} = \eta \left(\sum_{\alpha} \phi_{\alpha}(t+1) \frac{\partial c_{\alpha}(t+1)}{\partial w_{ij}(t)} \right) \quad (2.5)$$

Kladná konštanta η vyjadruje rýchlosť učenia. Vo väčšine prípadov je vhodné, ak η počas tréningu postupne klesá. Z biologického pohľadu je takéto učenie prípustné, znižovanie učiacej schopnosti sa totiž v prírode objavuje na mnohých miestach (napr. kritické obdobie pri učení vizuálnej kôry). Biologické systémy však zostávajú čiastočne plastické počas celého svojho života, preto by η nemala klesnúť až na nulovú hodnotu. Funkcia

$$\phi_i(t) = c_i(t)(c_i(t) - \theta_M^i(t)) \quad (2.6)$$

je synaptická modifikačná funkcia pre neurón i . Zo vzťahov (2.5) a (2.6) vyplýva, že ak sa synaptické spojenie w_{ij} v čase t využíva a zároveň je



Obr. 2.2: Funkcia $\phi_i(t) = (c_i(t)(c_i(t) - \theta_M^i(t)))$ pretína nulu práve v dvoch bodoch: 0 a $\theta_M^i(t)$. Ak aktivita i -teho neurónu padne do podprahového pásma $(0, \theta_M^i(t))$, aferentné váhy sa znižujú úmerne k hodnote $\phi_i(t)$. Ak je $c_i(t) > \theta_M^i(t)$, aferentné váhy daného neurónu rastú. Modifikačný prah sa posúva smerom do strán v závislosti od histórie c_i .

výstupná aktivita $c_i(t)$ menšia ako $\theta_M^i(t)$, potom sa váha spojenia w_{ij} v čase $t + 1$ znižuje. Naopak, ak $c_i(t)$ presiahne $\theta_M^i(t)$, aktivované spojenia sa zosilňujú (pozri obr. 2.2).

Vo vzťahu (2.5) figuruje parciálna derivácia aktivity neurónu podľa príslušnej váhy. Túto veličinu možno vyjadriť pomocou funkcie δ_{ij}^K ³ ako

$$\frac{\partial c_\alpha(t+1)}{\partial w_{ij}(t)} = \sigma'(t+1) \left[(\delta_{i\alpha}^K - \mu(1 - \delta_{i\alpha}^K)) d_j(t) + \left(m_\alpha(t) - \mu \sum_{\beta \neq \alpha} m_\beta(t) \right) \otimes \frac{\partial c(t)}{\partial w_{ij}(t-1)} \right], \quad (2.7)$$

resp.

$$\frac{\partial c_\alpha(t+1)}{\partial m_{ij}(t)} = \sigma'(t+1) \left[(\delta_{i\alpha}^K - \mu(1 - \delta_{i\alpha}^K)) c_j(t) + \left(m_\alpha(t) - \mu \sum_{\beta \neq \alpha} m_\beta(t) \right) \otimes \frac{\partial c(t)}{\partial m_{ij}(t-1)} \right], \quad (2.8)$$

kde $\frac{\partial c(t)}{\partial w_{ij}(t-1)}$ a $\frac{\partial c(t)}{\partial m_{ij}(t-1)}$ sú známe z predošlého kroku učenia.

³Kroneckerova delta: $\delta_{ij}^K = \begin{cases} 1 & \text{ak } i = j \\ 0 & \text{ak } i \neq j \end{cases}$

Možno si všimnúť, že BCM učenie závisí od viacerých voľných parametrov, ktorých výber podstatným spôsobom ovplyvňuje priebeh i samotnú úspešnosť učenia.

Parameter η špecifikuje rýchlosť modifikácie synaptických váh v smere klesajúceho gradientu $L(t)$. Príliš veľká hodnota η spôsobuje takmer stochastické správanie, učenie nekonverguje. Príliš malá hodnota η môže, naopak spôsobiť uviaznutie v lokálnom minime $L(t)$.

Veľkosť laterálnej inhibície μ ovplyvňuje samoorganizáciu v výstupnej vrstve. Malé μ znižuje závislosť jednotlivých neurónov od svojho okolia, viaceré neuróny sa preto môžu po natrénovaní správať identicky. S rastúcou hodnotou μ stúpajú medzi neurónmi vzájomné inhibičné väzby, čím sa podporuje diferenciácia ich správania.

Modifikačný prah θ_M^i v sebe integruje isté obdobie aktivít určitého výstupného neurónu. Dĺžku tohoto obdobia určuje konštanta τ (pozri (2.3) a (2.4)). Možno si tiež všimnúť, že hodnota τ ovplyvňuje rýchlosť modifikácie prahu θ_M^i .

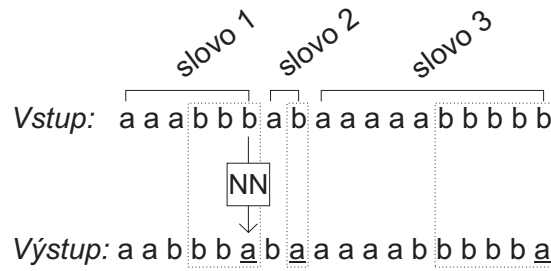
Konštanta efektívneho prahu \mathcal{A} doladuje správne umiestnenie prahu θ_M^i .

Dĺžka učenia môže byť tiež faktorom ovplyvňujúcim efektivitu učiaceho procesu. Príkladom môže byť tréning ukončený skôr, ako sieť skonverguje do minima stratovej funkcie L .

2.3 Predikcia vs. akceptácia

Typickým spôsobom testovania rekurentných neuronálnych modelov je tzv. predikčná úloha. Sieť trénovaná a testovaná takýmto spôsobom má za úlohu predikovať nasledujúci symbol predkladanej sekvencie (pozri obr. 2.3). Predikčná úloha je výhodnejšia oproti akceptačnej úlohe, čiže akceptácii celých slov, predovšetkým v jasnejšom a jednoduchšom tréningu. Na druhej strane akceptácia je priamym dôsledkom série úspešných predikcií pre celé slová. Akceptáciu preto možno považovať za podproblém predikcie, resp. problém v predikcii obsiahnutý.

Je známe, že samoorganizujúce sa siete, extrahujú isté štatistické vlastnosti trénovacej vzorky. Na základe týchto črt prebieha vnútorná modifikácia siete, čiže učenie. V prípade samoorganizujúcich máp ide o detekciu zhlukov. V prípade klasickej BCM siete sú extrahované smery s multimodálnym rozdelením hustoty pravdepodobnosti. Voľné parametre



Obr. 2.3: Príklad predikčnej úlohy na sekvencii slov jazyka $L_{a^n b^n}$. Akceptácia slova pozostáva zo série korektných predikcií typu $b \rightarrow ?$. Prechody typu $a \rightarrow ?$ nie sú predikovatelné, sieť totiž nevie, kedy sa na vstup e objaví prvý "b" symbol.

veľkou mierou ovplyvňujú smer, ktorým sa učenie bude uberať i črty, ktoré budú trénovaním odhalené.

Keďže učenie RBCM siete prebieha bez explicitnej informácie o požadovanom výstupe, nemožno priamo implementovať vstupno výstupné mapovanie. Je nutné definovať predikčný mechanizmus stojaci na vnútorných reprezentáciách siete, ktorý umožní vykonávanie predikčnej úlohy.

2.3.1 Pravdepodobnostné predikčné modely

Pravdepodobnostný predikčný model neurónovej siete je mechanizmus schopný predikovať nasledujúci vstupný symbol na základe vnútorného stavu siete. Ku každému stavu siete je priradená množina pravdepodobností vyjadrujúcich mieru očakávania určitého symbolu v danom okamihu. Nech $\mathcal{A} = \{s_1, \dots, s_n\}$ je pracovná abeceda a C nech je vnútorný stav siete. Formálne možno predikčný model pre sieť N definovať ako

$$\mathcal{M}_N = \{\forall C : [C, P(s_1|C), P(s_2|C), \dots, P(s_n|C)]\}, \quad (2.9)$$

pričom platí, že

$$\forall C : \sum_i P(s_i|C) = 1.$$

2.3.2 Konštrukcia predikčného modelu

Zo vzťahu (2.9) vidno, že množina \mathcal{M}_N vo špecifickosti nemusí byť konečná. Preto sa v praxi vytvárajú ohraničené aproximácie modelu \mathcal{M}_N . Budované sú na základe množiny dosiahnuteľných vnútorných stavov siete, počas prechodu cez tréningovú množinu. Množina dosiahnuteľných stavov vytvára v stavovom priestore zhluky aktivít, ktorých centrá budú tvoriť konečnú stavovú bázu ohraničeného predikčného modelu.

Extrakciu centier zhlukov, klastrovanie, možno realizovať viacerými spôsobmi. V tejto práci bol použitý synchronný K -means klastrovací algoritmus, hoci rovnako dobre môže poslúžiť napr. klastrovanie typu SOM, prípadne iné.

Každému centru C_i priradíme pole počítadiel, pre každý symbol abecedy jedno. Opätovným prechodom cez tréningovú množinu zistíme počty podmienených prechodov zo stavu C_i do niektorého iného stavu, za predpokladu, že je na vstupe práve znak s_j . Prvorádovou normalizáciou počítadiel konkrétneho centra C_i získame pole podmienených pravdepodobností $\mathcal{P}(s_j|C_i)$ aproximujúce ideálne prechodové pravdepodobnosti $P(s_j|C_i)$.

Zaraďovanie reálnych stavov siete ku konkrétnym centrom C_i prebieha pomocou Euklidovskej miery podobnosti. Stav \mathcal{C} je reprezentovaný centrom C_k , ktoré spĺňa:

$$\forall j \neq k : |C_k - \mathcal{C}| \leq |C_j - \mathcal{C}|, \quad (2.10)$$

kde $|a - b|$ je Euklidovská vzdialenosť.

2.3.3 Miery úspešnosti predikčných modelov

Úspešnosť predikčného modelu možno posudzovať z viacerých rôznych aspektov. Môže ním byť štatistická úspešnosť predikcie po znakoch, úspešnosť predikcie dlhých kontinuálnych úsekov či selektívna predikcia na konkrétnych miestach. Rôzne miery majú rôznu výpovednú hodnotu pre rôzne typy riešených problémov. V niekoľkých nasledujúcich statiach uvedieme miery používané v tejto práci.

Miera 1: Pravdepodobnosť NNL

Nech \mathcal{M} je predikčný model, nech A je počet symbolov abecedy \mathcal{A} , nad ktorou je model vybudovaný a nech m je dĺžka testovacej sekvencie S . Potom

NNL (Normalized Negative Log-Likelihood) možno vyjadriť ako

$$\text{NNL}_{\mathcal{M}}(S) = \frac{-1}{m-1} \sum_{t=1}^{m-1} \log_A P_{\mathcal{M}}(s_{t+1}|C_t). \quad (2.11)$$

Zo vzťahu (2.11) vidno, že NNL úspešného modelu ($\forall t : P_{\mathcal{M}}(s_{t+1}|C_t) \approx 1$) je nízke ($\text{NNL}_{\mathcal{M}} \approx 0$), zatiaľ čo NNL neúspešného modelu stúpa k ∞ . Pre model s rovnomerne rozdelenými pravdepodobnosťami v jednotlivých momentoch rozhodovania (hod kockou) je $\text{NNL}_{\mathcal{M}} \approx 1$.

NNL vyjadruje celkovú štatistickú úspešnosť modelu \mathcal{M} . Táto úspešnosť je vhodná ako meradlo podobnosti medzi predikovanou a predkladanou postupnosťou symbolov, nie je však vhodná na indikáciu úspešnej akceptácie celých slov či dlhších kontinuálnych úsekov. Nehovorí totiž nič o mieste omylu a dobe, počas ktorej sa model nepomýlil či pomýlil viac krát.

Miera 2: Percento chybných predikcií (PERR)

Pri zisťovaní úspešnosti predikčného modelu \mathcal{M} je častokrát výhodné poznať absolútnu omylnosť systému vyjadrenú v percentách. NNL takúto informáciu priamo neposkytuje, preto sme použili taktiež alternatívnu mieru PERR (Prediction Error), ktorá vypovedá práve o celkovej chybe predikcie na testovacej sekvencii.

$$\text{PERR}_{\mathcal{M}}(S) = \frac{1}{m-1} \sum_{t=1}^{m-1} \text{err}(P_{\mathcal{M}}(s_{t+1}|C_t)) \quad (2.12)$$

$$\text{err}(x) = \begin{cases} 1 & \text{ak } x \leq 0.5 \\ 0 & \text{ak } x > 0.5 \end{cases}$$

Miera PERR i ostatné ďalej spomenuté miery, sú použiteľné iba za predpokladu použitia dvojsymbolovej abecedy. Rozhodovania totiž, v tomto prípade, pozostávajú z výberu medzi dvoma alternatívami.

Miera 3: Percento akceptovaných slov (CPW)

CPW (Correctly Predicted Words) miera hovorí o celkovom úspechu modelu, nie však na úrovni symbolov, ale na úrovni predikcie celých slov. V našom prípade predkladáme sieti slová tvaru $a^n b^n$. Predikcia takýchto slov je ukončená po príchode prvého symbolu "a" po niekoľkých "b" symboloch

(pozri obr. 2.3). Prvá polovica slov $a^n b^n$ je však nepredikovatelná, počet za sebou idúcich "a" symbolov totiž nie je odhadnuteľný žiadnym spôsobom. Po príchode prvého "b" symbolu nasleduje dobre predvídateľná "b" sekvencia konkrétnej dĺžky, zakončená písmenom "a" na správnom mieste. Pod akceptovaním slova z $L_{a^n b^n}$ budeme preto rozumieť korektnú predikciu všetkých symbolov druhej polovice slova, vrátane predikcie typu $b \rightarrow a$ na konci slova.

$$\text{CPW}_{\mathcal{M}}(S) = \frac{W_{\text{accept}}}{W_{\text{all}}} \quad (2.13)$$

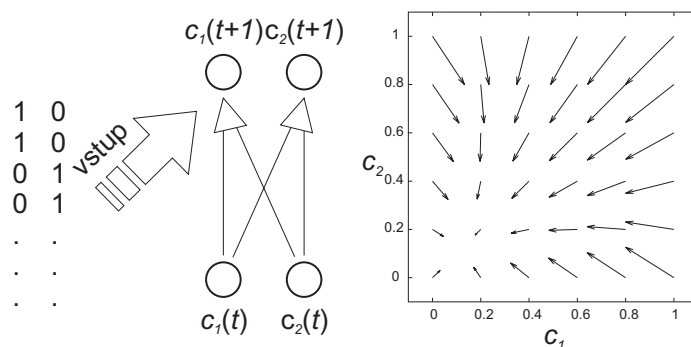
W_{pred} je počet akceptovaných slov, W_{all} je počet všetkých slov testovacej množiny.

Miera 4: Miera generalizácie

Zaujímavou vlastnosťou neuronálnych modelov je schopnosť správne zovšeobecňovať, generalizovať, informácie obsiahnuté v tréningovej vzorke. Ide o istý druh predikčnej extrapolácie mimo tréningovú množinu. Generalizujúce siete úspešne akceptujú slová, ktoré nie sú obsiahnuté v tréningovej vzorke, no patria do požadovaného jazyka. Pokiaľ sieť obsahuje vhodné vnútorné reprezentácie zachytávajúce iba podstatné črty (v našom prípade je to reprezentácia počítadla), možno predpokladať, že schopnosť zovšeobecnenia bude vysoká. Naopak, pokiaľ sieť počas tréningu extrahovala nepodstatnú informáciu, popisujúcu konkrétne slová tréningovej množiny, zovšeobecnenie bude malé. Úroveň generalizácie budeme v našom experimente kvantifikovať dĺžkou maximálneho akceptovaného slova.

2.4 Analýza dynamiky siete

Ako sa spomenulo už vyššie, rekurentnú neurónovú sieť je možno považovať tiež za dynamický systém (DS) s premenlivou dynamikou. Po ukončení tréningu a zafixovaní jednotlivých váh, sa sieť správa ako transformátor stavu minulého do stavu nasledujúceho (pozri obr. 2.4). Vo fázovom priestore dynamických systémov sa často vyskytujú takzvané pevné body, ktoré predstavujú kľúčovú časť vnútorných sieťových reprezentácií [19, 5]. Preto má zmysel detekovať počet, pozíciu a povahu pevných bodov daného systému, prípadne odhadovať správanie celého DS v ich okolí.



Obr. 2.4: V prípade konštantného vstupu sa RNN správa ako dynamický systém, ktorého povahu možno zachytiť pomocou fázového diagramu (na pravej strane). Zmena externého vstupu spôsobuje „prepínanie“ medzi dynamikami rôznych typov.

2.4.1 Linearizačná metóda

Bežnou metódou analýzy nelineárnych dynamických systémov je metóda linearizácie. Linearizačná analýza funguje na princípe aproximácie nelineárnych DS pomocou lineárnych, v okolí každého pevného bodu. Metóda pozostáva zo štyroch krokov:

Krok 1: Detekcia pevných bodov

Pre lineárny dynamický systém možno pevné body získať riešením sústavy lineárnych rovníc o viacerých neznámych. Pre nelineárne DS však neexistuje všeobecné analytické riešenie. Rekurentné neurónové siete so sigmoidálnou aktivačnou funkciou, zaraďujeme medzi nelineárne DS, preto sa na detekciu pevných bodov používa niektorá z numerických metód. Pevné body možno rozdeliť do niekoľkých kategórií: atraktory, repelory, periodické pevné body, sedlá a podivné atraktory (pozri dodatok A). Pevný bod typu atraktor možno jednoducho odhaliť opakovanou iteráciou systému z vhodnej štartovacej pozície. Pre ostatné typy pevných bodov však opäť neexistuje jednoduchý detekčný postup.

Krok 2: Výpočet parciálnych derivácií

Parciálne derivácie $\frac{\partial c_i(t+1)}{\partial c_j(t)}$ sa v RBCM sieťach čiastočne odlišujú od príslušných derivácií v tradičných rekurentných sieťach. Jakobián v bode \vec{x} možno pre RBCM sieť formálne vyjadriť ako

$$J(\vec{x}) = \begin{bmatrix} \frac{\partial c_1(t+1)}{\partial c_1(t)} & \cdots & \frac{\partial c_1(t+1)}{\partial c_n(t)} \\ \vdots & & \vdots \\ \frac{\partial c_n(t+1)}{\partial c_1(t)} & \cdots & \frac{\partial c_n(t+1)}{\partial c_n(t)} \end{bmatrix}, \quad (2.14)$$

pričom

$$\frac{\partial c_i(t+1)}{\partial c_j(t)} = \sigma'(\vec{x}) (m_{ij} - \mu \sum_{k \neq i} m_{kj}). \quad (2.15)$$

Konštanta n predstavuje počet výstupných (stavových) neurónov, μ koeficient laterálnej inhibície a σ' je derivácia sigmoidálnej funkcie.

Krok 3: Konštrukcia lokálnych lineárnych systémov

Pre každý pevný bod \vec{x}_{fix} , detekovaný v kroku 1, vypočítame $J(\vec{x}_{fix})$ podľa vzťahov (2.14) a (2.15). Príslušný lineárny systém je potom definovaný nasledujúcim vzťahom:

$$\vec{c}_{in}(t+1) = J(\vec{x}_{fix}) \otimes \vec{c}_{in}(t). \quad (2.16)$$

Krok 4: Zistenie charakteristických hodnôt a vektorov konkrétneho lineárneho systému, odvodenie správania nelineárneho systému v okolí pevného bodu

Podľa charakteristických hodnôt matice lokálneho lineárneho systému možno presne špecifikovať typ pevného bodu. Povaha pevného bodu popisuje správanie celého DS v jeho blízkom okolí. Podrobnosti analýzy pevných bodov na základe charakteristických hodnôt a vektorov sú uvedené v dodatku A.

Kapitola 3

Experiment

3.1 Trénovacia a testovacia množina

RBCM sieť, použitá v práci, bola trénovná na sekvencii slov jazyka $L_{a^n b^n}$, ktoré sa jej predkladali po jednotlivých symboloch. Trénovacia množina pozostávala zo slov náhodnej dĺžky. Obsahovala však viac kratších slov ako dlhších v nasledujúcom pomere:

n	$k(n)$
1	10
2	6
3	4
4	3
≥ 5	1

Konštanta n určuje polovicu dĺžky slova $a^n b^n$, koeficient $k(n)$ určuje pomer pravdepodobností, s akými sú slova určitej dĺžky generované. Pomery pravdepodobností sú analyticky vyjadriteľné ako $k(n) \approx \left\lceil \frac{10}{n} \right\rceil$. Trénovacia množina použitá v našom experimente pozostávala z 390 náhodných slov dĺžky maximálne 16 znakov ($n \leq 8$). Výber trénovacej množiny s nerovnomernou distribúciou slov rôznej dĺžky bol motivovaný predovšetkým snahou simulovať „kvázi-reálne“ prostredie, čiže také, v ktorom nie je ostrá hranica medzi javmi bežnými a javmi zriedkavými.

Natrénovaná sieť bola testovaná dvoma spôsobmi. Pri prvom teste bola použitá náhodná sekvencia slov dĺžky max. 16 ($n \leq 8$). Zámerom prvého testu bolo zistiť, či sieť úspešne rozpoznáva natrénované údaje. Druhým testom sme chceli určiť schopnosť a hranice generalizácie natrénovaných

RBCM sietí. Na tento účel sme použili testovaciu sekvenciu obsahujúcu slová dĺžky max. 32 ($n \leq 16$). Dĺžky slov oboch testovacích sekvencií boli rovnomerne distribuované.

3.2 Parametre tréovania

Úspešnosť tréovania BCM siete závisí veľkou mierou od vhodne nastavených parametrov a štartovacích podmienok. Stabilné riešenie, t.j. také, pri ktorom je stratová funkcia $L(t)$ už dostatočne dlhú dobu stabilizovaná, môže v závislosti od parametrov tréovania nadobúdať najrôznejšie formy. Je preto dôležité nájsť parametre vhodné pre daný problém, príp. pre povahu tréovacích dát. Hľadanie optimálnych parametrov BCM tréovania je však natoľko komplexný problém, že by sa mu mohla venovať zvláštna štúdia.

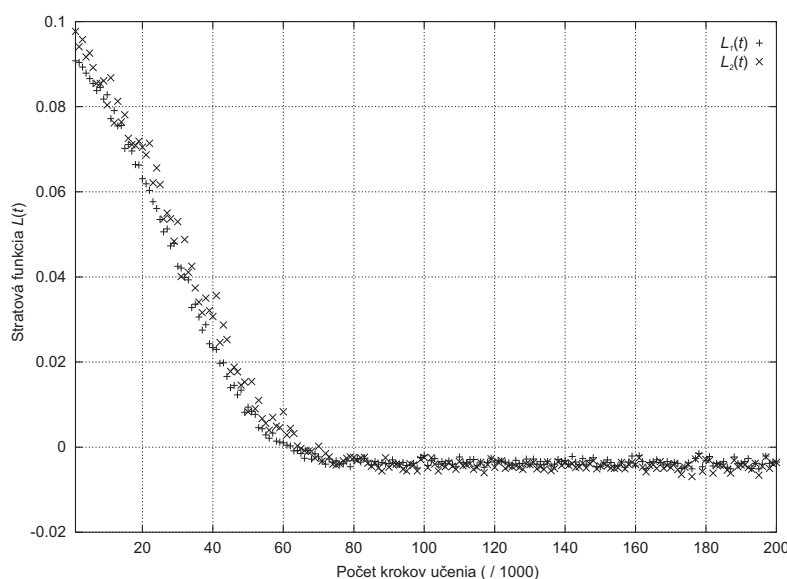
V experimente sme použili RBCM siete s tromi neurónmi vo vstupnej a dvoma vo výstupnej vrstve. Symboly abecedy sme reprezentovali pomocou lokálneho kódovania dvojrozmernými vektormi. Každý vektor, reprezentujúci určitý symbol, mal nastavenú práve jednu dimenziu na hodnotu 1. Rozšírením vstupu o jeden rozmer sme do modelu zaviedli taktiež posun prahu aktivácie (BIAS).

Použité boli tieto parametre učenia: rýchlosť učenia $\eta = 0.0001$, dĺžka histórie aktivít $\tau = 100$, škálovací koeficient $\mathcal{A} = 0.19$ a koeficient laterálnej inhibície $\mu = 0$.

3.3 Výsledky

Experimentálny protokol pozostával zo 120 behov učenia (12 sérií po 10 sietí). Jednotlivé série sa navzájom odlišovali dĺžkou tréovania. Po skončení fázy učenia siete prešli testovaním predikcie, akceptácie a generalizácie. Ukázalo sa, že tréovanie konverguje, stratová funkcia $L(t)$ sa pre jednotlivé siete ustálila v okolí svojej stabilnej hodnoty približne po 80000 iteráciách. Možno však sledovať jemné zväčšovanie variancie, súvisiacej so zlepšovaním predikčných schopností (pozri obr. 3.1). Jednoznačnú konvergenciu možno pozorovať i na grafoch NNL a predikčnej chyby (pozri obr. 3.2 a 3.3).

Treba zdôrazniť, že testovanie NNL, predikčnej chyby a úspešnej akceptácie prebiehalo na sekvencii obsahujúcej slová dĺžky max. 16 znakov ($n \leq 8$).



Obr. 3.1: Hodnoty stratových funkcií $L_i(t)$ v rôznych okamihoch učenia. Ku koncu tréovania stúpa variancia okolo stabilnej hodnoty (cca. -0.003).

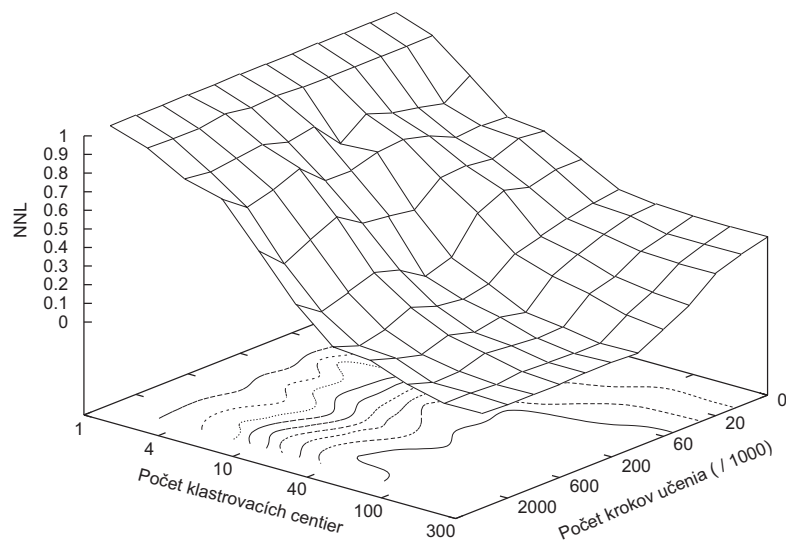
Z experimentu vyplynulo, že natrénované RBCM siete dobre vystihujú povahu tréovacích dát a umožňujú vybudovanie uspokojivých predikčných modelov. Zo štatistického pohľadu túto schopnosť zachytávajú obr. 3.2 a 3.3. Pri použití 100 klastrovacích centier je $NNL \approx 0$, už po uplynutí 200000 iterácií učenia. Absolútna chyba predikcie je taktiež približne rovná 0%. Dobrá predikčná schopnosť implikuje dostatočnú separáciu navzájom nesúvisiacich klastrov stavového priestoru i vhodnú dynamiku prechodov medzi nimi. Hlbšou analýzou stavových priestorov a ich dynamík sa budeme zaoberať v kapitole 4.

Ak sa posunieme z úrovne symbolov na úroveň slov a začneme od predikčných modelov požadovať korektné akceptovanie celých slov, situácia sa mení len čiastočne. Rozdiely medzi jednotlivými sériami učenými nad 200000 iterácií sú o čosi badateľnejšie, maximum úspešnosti sa posúva smerom k dlhšie učeným sériám. Globálne najúspešnejšia séria bola trévaná až 2000000 iterácií (pozri obr. 3.4).

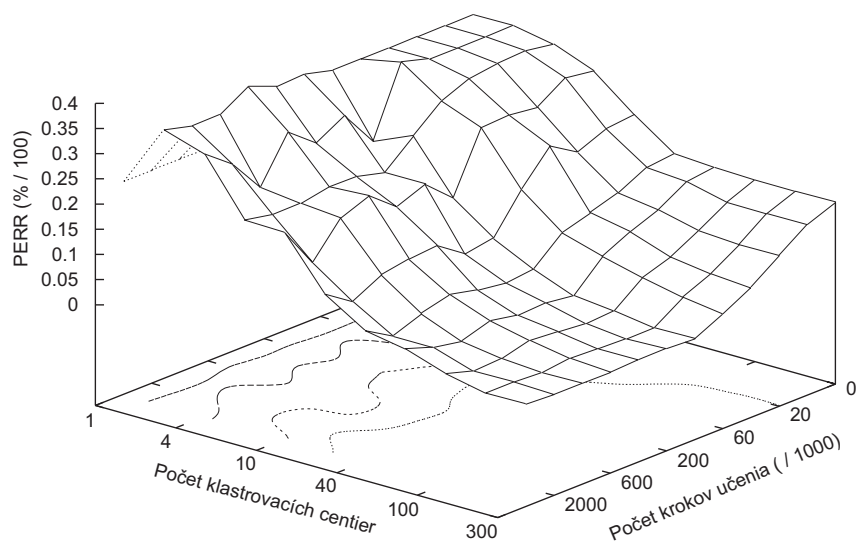
Natrénované RBCM siete preukázali i nezanedbateľnú schopnosť zovšeobecnenia. V priemere boli siete schopné korektné akceptovať slovo dĺžky až 24 znakov ($n \leq 12$), čo je 1.5 násobok dĺžky maximálneho slova zaradeného do tréovacej množiny (pozri graf 3.5). Dĺžka maximálneho

akceptovaného slova však nepopisuje generalizačnú schopnosť dokonale. V prípade niektorých natrénovaných sietí vzniká napríklad problém s akceptáciou slov dĺžky $2k$, kde k je nepárne číslo.

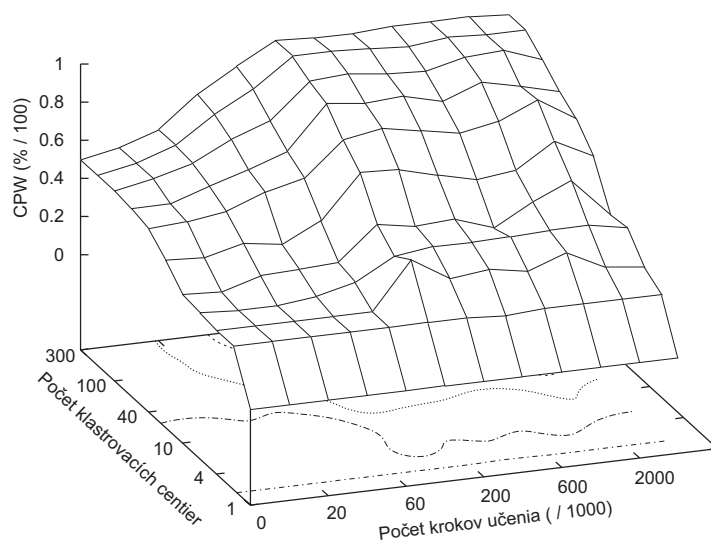
Z experimentu zreteľne vyplýva, a to z predikčného i akceptačného testu, že BCM učenie napomáha budovaniu úspešných predikčných modelov, t.j. vytvára vhodné vnútorné reprezentácie predkladaných dát.



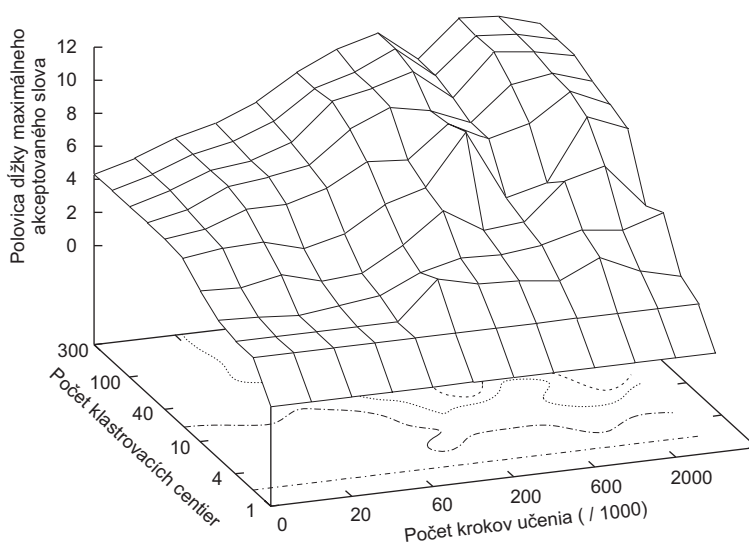
Obr. 3.2: Hodnota NNL v závislosti od dĺžky tréovania a počtu klastrovacích centier. BCM tréovanie zníži hodnotu NNL z cca. 0.4 na takmer 0.



Obr. 3.3: Percento chybnjej predikcie v závislosti od dĺžky tréovania a počtu klastrovacích centier.



Obr. 3.4: Percento dobre akceptovaných slov v závislosti od dĺžky tréovania a počtu klastrovacích centier.



Obr. 3.5: Generalizácia v závislosti od dĺžky tréovania a počtu klastrovacích centier. Optimálna dĺžka tréovania sa pohybuje približne medzi 1000000 a 2000000 iteráciami, hoci séria "600" vytvára zreteľnú nehomogenitu vo vývoji generalizácie. Vznik takejto nekonzistentnosti však nie je jednoduché vysvetliť.

Kapitola 4

Analýza stavového priestoru

V tejto kapitole rozoberieme detailnejšie dve konkrétne siete, vyvinuté pomocou BCM učenia za rôznych okolností. Pozrieme sa na ich dynamiku, vývoj v priebehu učenia, povahu pevných bodov, trajektórie slov vo fázovom priestore a tiež teoreticky zdôvodníme úroveň generalizácie.

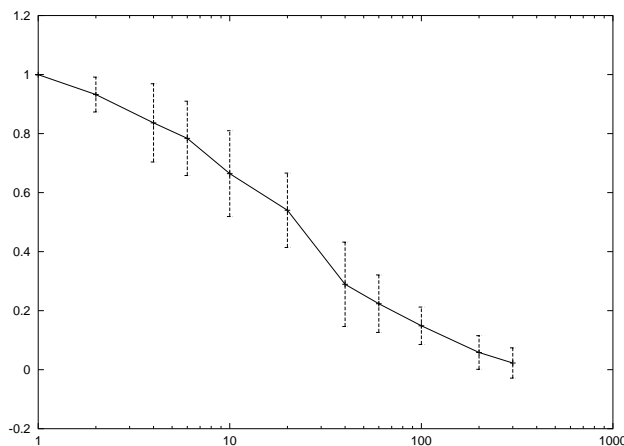
Na simuláciu sme použili RBCM sieť s dvoma neurónmi vo výstupnej vrstve. Stavový priestor siete je tak reprezentovateľný pomocou dvojrozmerných diagramov. Os x bude v ďalšom texte vždy znázorňovať aktivitu prvého výstupného neurónu, os y aktivitu druhého neurónu výstupnej vrstvy (pozri obr. 4.4, 4.5, ...).

4.1 Sieť 1: Cyklická dynamika

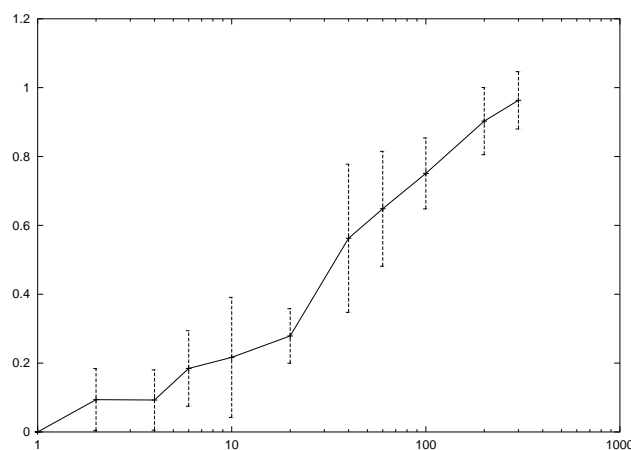
Prvým typom dynamiky, ktorým sa budeme bližšie zaoberať je tzv. cyklická dynamika. Ide o stabilný dynamický model, ktorého charakteristické črty sa od istého momentu ďalším tréňovaním už nemenia.

Parametre tréňovania boli čiastočne odlišné od predchádzajúcich (pozri časť 3.2), použili sa nasledujúce hodnoty: rýchlosť učenia $\eta = 0.001$, dĺžka histórie aktivít $\tau = 100$, škálovací koeficient $\mathcal{A} = 0.1725$ a koeficient laterálnej inhibície $\mu = 0$.

Predikcia a akceptácia nadobudla optimálne hodnoty po uplynutí približne 1350000 krokov učenia. Najlepšie NNL hodnoty sa pohybovali v okolí 0.1, percento správnej akceptácie na tréňovacej vzorke dosiahlo takmer 99% (pozri obr. 4.1). Grafy pozostávajú z priemerných hodnôt pre jednu sériu sietí tréňovaných za tých istých okolností, vyznačené sú i štandardné

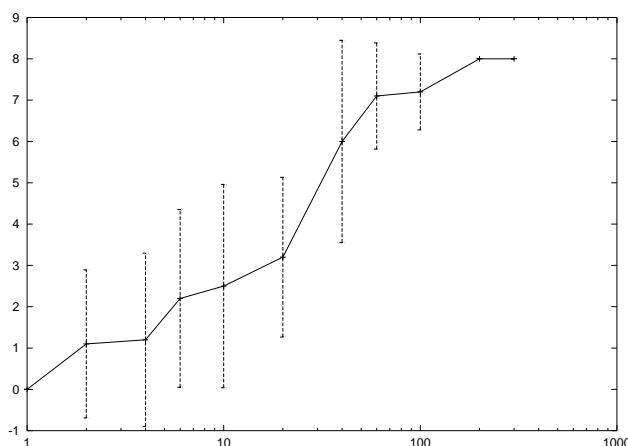


(a)



(b)

Obr. 4.1: Priebek (a) NNL a (b) percento akceptovaných slov pre siete s cyklickou dynamikou (typ 1). Grafy zjednocujú výsledky série trénovanej 1350000 krokov. Uvedená je štandardná odchyľka od priemeru. (Os x - počet klastrovacích centier, os y - (a) NNL / (b) CPW.)



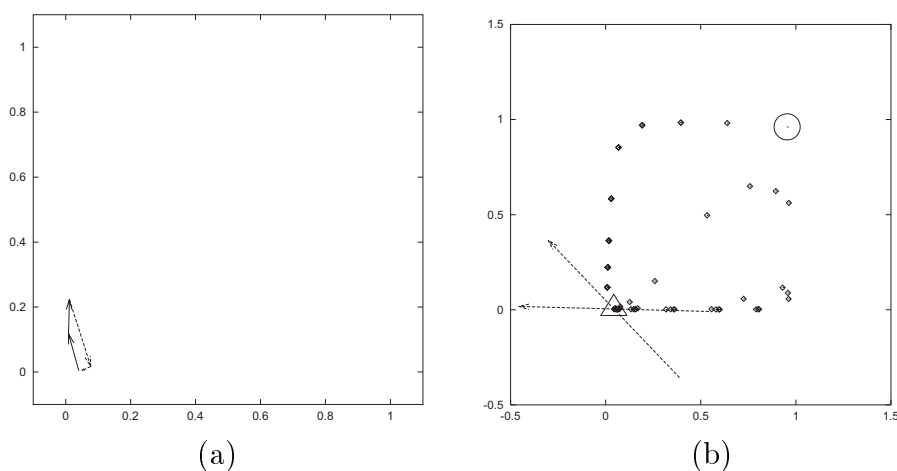
Obr. 4.2: Polovica dĺžky maximálneho akceptovaného slova pre sériu sietí typu 1, trénovanú 1350000 krokov. Ani s použitím maximálneho počtu klastrovacích centier siete nemajú žiadnu schopnosť generalizácie. (Os x - počet klastrovacích centier, os y - generalizácia.)

odchylky od priemeru.

Siete s cyklickou dynamikou nie sú schopné generalizovať naučenú informáciu. Z obr. 4.2 vidno, že aj pri použití 300 klastrovacích centier, nie je maximálne akceptované slovo dlhšie ako 16 symbolov, čo je práve maximum trénujúcej množiny. Prečo je to tak, skúsime vysvetliť v nasledujúcich pár odstavcoch.

Na odhalenie vnútorných reprezentácií sme použili linearizačnú metódu popísanú v časti 2.4.1. Natrénovaná neurónová sieť bola podrobená všetkým štyrom krokom linearizácie.

Zistili sme, že sieť 1 má dynamiku dvoch pevných bodov, pričom každý z nich je atraktor. Pevné body sú rozmiestnené v rohoch stavového priestoru, pevný bod dynamiky "a" (Fix_a) leží na pozícii [0.95, 0.96], pevný bod dynamiky "b" (Fix_b) leží na pozícii [0.04, 0] (pozri obr. 4.4, 4.3b). Podľa charakteristických hodnôt lineárneho systému v každom z pevných bodov, možno určiť typ daného pevného bodu a tým aproximovať správanie celého nelineárneho systému v jeho okolí. Charakteristické hodnoty λ_i sú v našom prípade nasledovné:

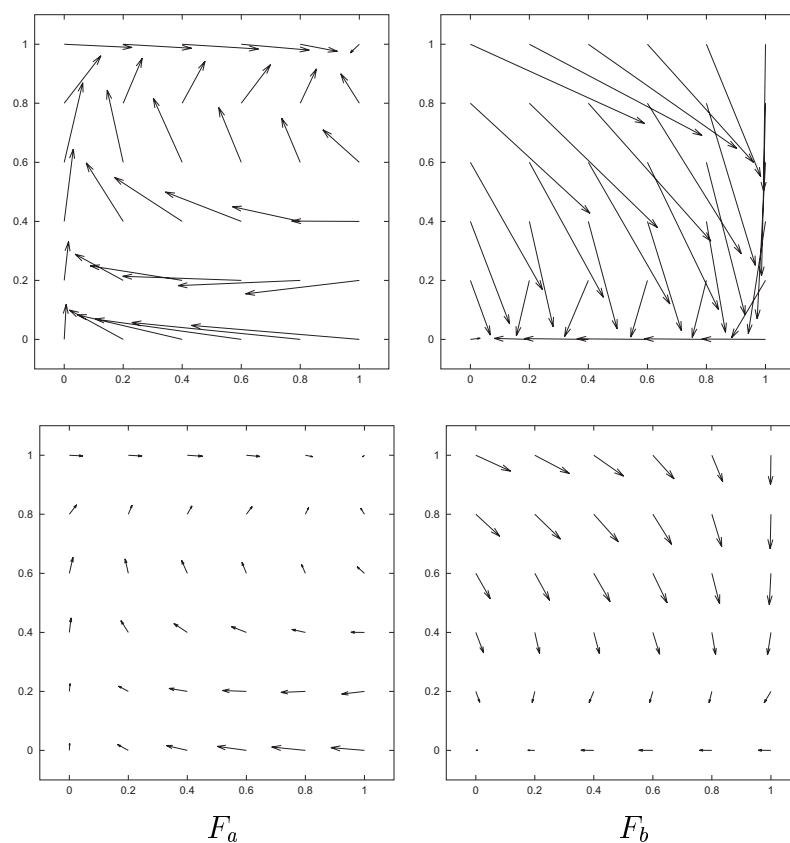


Obr. 4.3: (a) Trajektória slova a^2b^2 v stavovom priestore siete 1, po uplynutí 1350000 krokov učenia. Ide o cyklický dej začínajúci sa a končiaci sa približne v tom istom bode. Tento druh správania je spôsobený dynamikou dvoch atraktívnych pevných bodov, rozmiestnených v protiláhlých rohoch stavového priestoru. (b) Stavový diagram zobrazuje rozmiestnenie pevných bodov siete 1. Krúžok predstavuje pevný bod dynamiky "a", trojuholník pevný bod dynamiky "b". Šípky pretínajúce sa v pevnom bode ukazujú smery lineárnej kontrakcie či expanzie v jeho blízkom okolí (charakteristické vektory). V diagrame možno nájsť takisto množinu všetkých dosiahnuteľných stavov počas prechodu cez tréningovú množinu (malé krúžky).

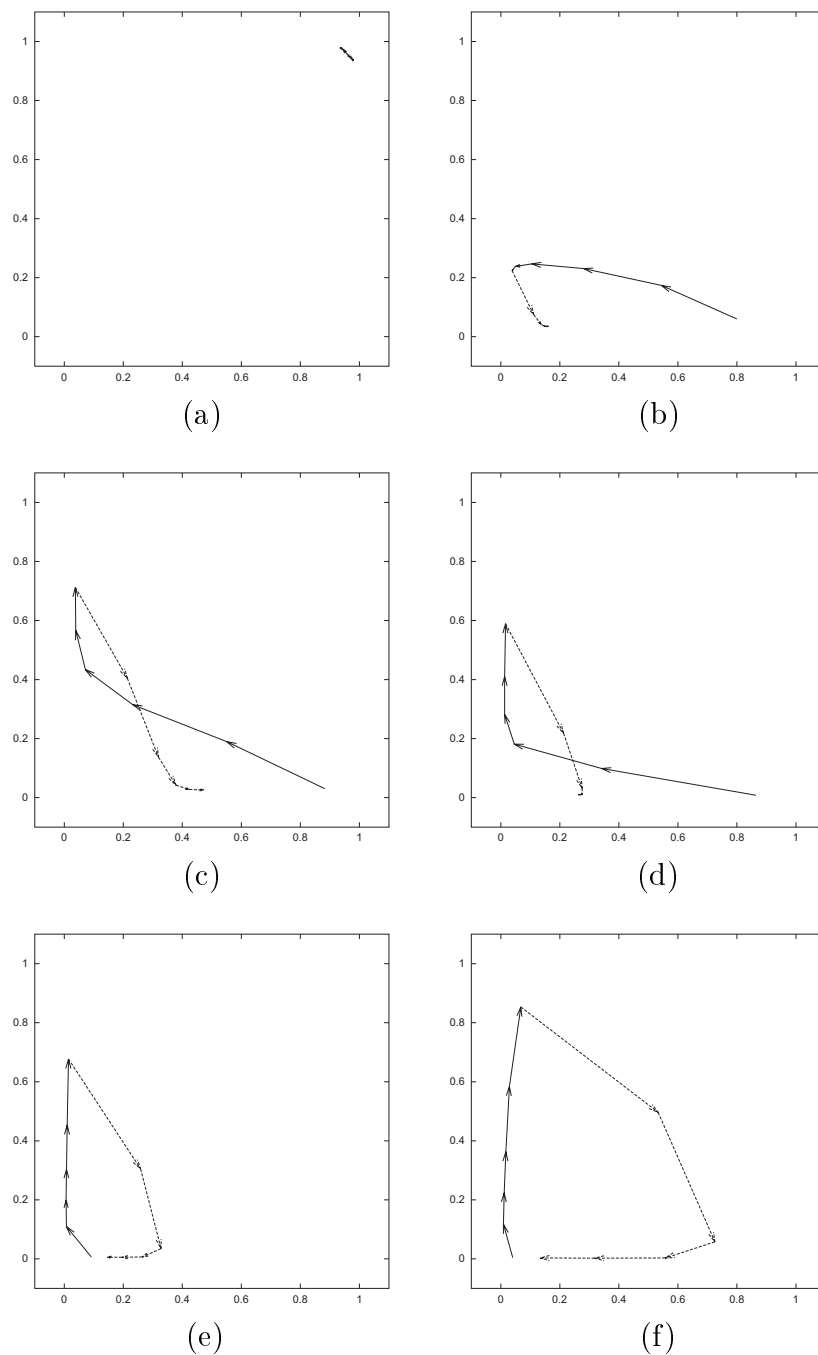
	λ_1	λ_2	$ \lambda_1 $	$ \lambda_2 $
Fix_a	$0.22 + 0.08i$	$0.22 - 0.08i$	0.23	0.23
Fix_b	0.03	0.19	0.03	0.19

Vidno, že dĺžky charakteristických hodnôt pre obidva pevné body sú menšie ako 1. Znamená to, že v oboch prípadoch ide o atraktívne pevné body. Tiež vidno, že kontraktivita atraktorov je približne rovnaká ($0.23 \approx 0.19$), čo je istým predpokladom pre úspešnú akceptáciu. Možno si všimnúť, že pre pevný bod Fix_a neexistujú vlastné (reálne) charakteristické hodnoty ani vektory. Táto situácia nastáva v prípade, keď nie je možné nájsť vlastné osi lineárnej kontrakcie či expanzie. V našom prípade to indikuje špirálovitý atraktor. Povaha atraktorov Fix_a a Fix_b sa zdá byť z istého nadhľadu rovnaká, no nie je to tak. Líšia sa totiž správaním v ich bezprostrednom okolí.

Gradienty celého stavového priestoru siete 1 možno pozorovať vo fázových



Obr. 4.4: Fázové diagramy dynamiky typu 1. Stĺpec F_a zachytáva dynamiku systému so znakom "a" na vstupe. Stĺpec F_b zachytáva dynamiku systému so znakom "b" na vstupe. Vo vrchných grafoch sú uvedené skutočné gradienty v jednotlivých bodoch stavového priestoru. Gradienty v spodných grafoch sú normalizované na $\frac{1}{5}$ originálnej dĺžky. Z diagramov je vidno, že každá z dynamík obsahuje jeden atraktívny pevný bod. Pre F_a sa pevný bod nachádza približne v lokalite $[0.95, 0.96]$, pre F_b v lokalite $[0.04, 0]$.



Obr. 4.5: Trajektórie slova a^5b^5 v stavovom priestore siete 1, v priebehu tréovania. Grafy zachytávajú situáciu po uplynutí určitého počtu krokov učenia: (a) 0, (b) 20, (c) 200, (d) 600, (e) 1000, (f) 1350 (všetky hodnoty $\times 10^3$). Šípky kreslené plnou čiarou reprezentujú zmenu stavu s príchodom symbolu "a". Prerušované šípky reprezentujú zmenu stavu s príchodom symbolu "b".

diagramoch na obr. 4.4. Normalizované fázové diagramy však môžu byť zavádzajúce (spodná polovica obr. 4.4). Zmazávajú totiž rozdiely medzi niektorými typmi pevných bodov, napr. pevný bod typu „sedlo“ nemožno dobre odlíšiť od normálneho atraktora.

Ako sa už spomenulo, vnútorná reprezentácia jazyka $L_{a^n b^n}$ je v prípade siete s cyklickou dynamikou založená na dynamike dvoch atraktorov. Tento druh dynamiky prináša viacero nevýhod, najmä v súvislosti s predikciou či generalizáciou. K dobrej predikcii je bezpodmienečne potrebné, aby kontraktívnosť oboch pevných bodov bola rovnaká. Dôsledkom nevyváženosti spádov jednotlivých atraktorov môže byť neuzavretosť trajektórií slov, alebo znižovanie „rozlišovacej“ schopnosti v okolí niektorého z pevných bodov. V prípade siete 1 je prvá podmienka viac či menej splnená. Trajektórie slov sú v naučenej sieti uzavreté, no rozlišovacia schopnosť v oboch pevných bodoch nepostačuje (pozri obr. 4.5 a 4.3a). Práve tento nedostatok je hlavnou príčinou zlej generalizačnej schopnosti.

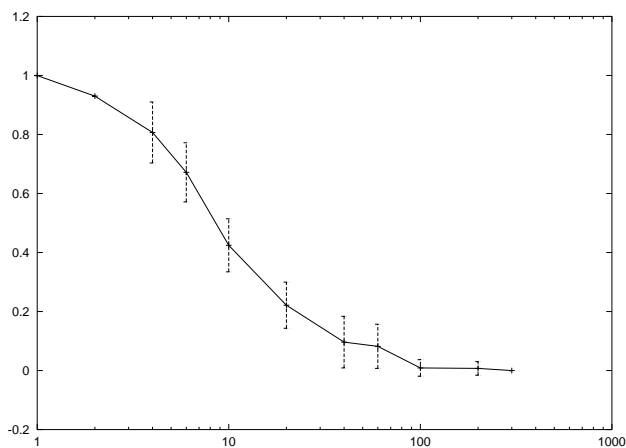
V tejto súvislosti môže byť zaujímavý i vývoj dynamiky siete v priebehu učenia. Najlepšie je možno porozumieť dynamike siete pomocou trajektórií slov v stavovom priestore (pozri obr. 4.5).

4.2 Sieť 2: „Cik-cak“ dynamika

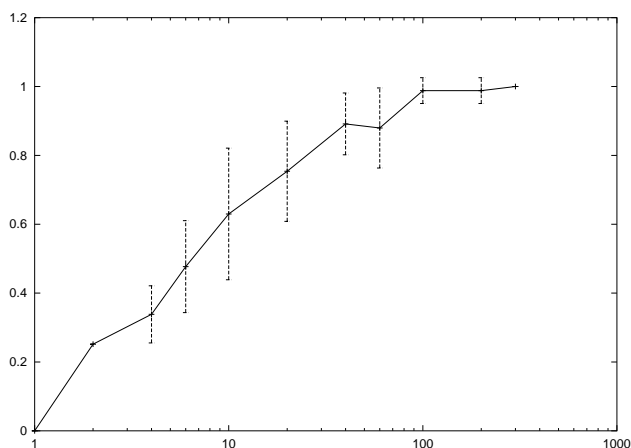
Druhou podrobnejšie rozoberanou dynamikou bude tzv. „cik-cak“ dynamika. Na rozdiel od sietí typu 1, prejavuje táto dynamika tiež isté generalizačné schopnosti. „Cik-cak“ dynamika je dôsledkom už čiastočne popisovaného tréningu z kapitoly 3.

Pre jednoznačnosť opäť uvádzame použité parametre tréningu: učiaci faktor $\eta = 0.0001$, dĺžka histórie aktivít $\tau = 100$, škálovací koeficient $\mathcal{A} = 0.19$ a koeficient laterálnej inhibície $\mu = 0$.

Z kapitoly 3 vidno, že maximum predikčnej a generalizačnej schopnosti nastane približne po 2000000 krokoch učenia (pozri obr. 3.2, 3.3, 3.4, 3.5). Úspešnosť série tréningu práve 2000000 iterácií zachytáva konkrétnejšie obr. 4.6. V porovnaní so sieťami typu 1, bolo možné pozorovať podstatne menšiu varianciu úspešnosti, už pri 100 klastrovacích centrách je predikčná schopnosť pre všetky siete takmer 100%. Priemerná dĺžka maximálneho akceptovaného slova stúpila na čosi viac ako 20 symbolov ($n \approx 10$), čo predstavuje 1.25 násobok maximálneho slova v tréningovej množine (pozri obr. 4.7).

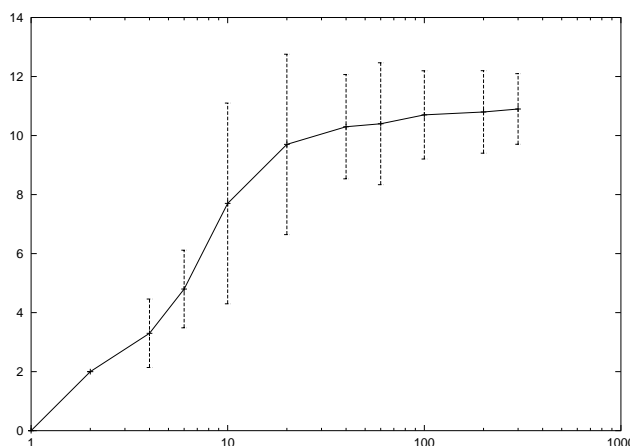


(a)



(b)

Obr. 4.6: Priebeh (a) NNL a (b) percento akceptovaných slov pre siete s „cik-cak“ dynamikou (typ 2). Grafy zjednocujú výsledky série trénovanej 2000000 krokov. Možno si všimnúť podstatne menšiu varianciu úspešnosti pre vyššie počty klastrovacích centier, v porovnaní so sieťou typu 1. (Os x - počet klastrovacích centier, os y - (a) NNL / (b) CPW.)



Obr. 4.7: Polovica dĺžky maximálneho akceptovaného slova pre sériu sietí typu 2, trévanú 2000000 krokov. Na rozdiel od sietí typu 1, je viditeľná istá schopnosť generalizácie. (Os x - počet klastrovacích centier, os y - generalizácia.)

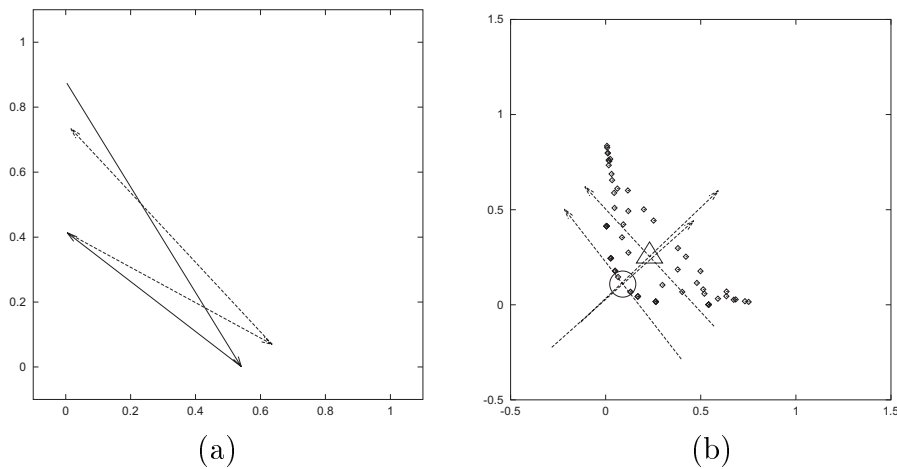
Z výsledku linearizácie vyplynulo, že sieť 2 využíva opäť dynamiku dvoch pevných bodov, ktoré však majú odlišný charakter.

	λ_1	λ_2
Fix_a	-0.3	-0.76
Fix_b	-0.63	-1.56

Pevný bod Fix_a je atraktor, ktorý mení znamienko v smere oboch osí kontrakcie. Pevný bod Fix_b má čiastočne iné vlastnosti. Absolútna hodnota druhej z charakteristických hodnôt je totiž väčšia ako 1, čo zapríčiňuje expaziu v smere jedného z charakteristických vektorov. Toto je typickou vlastnosťou sedlových pevných bodov. Obidve charakteristické hodnoty pre Fix_b sú záporné, čiže menia znamienko tiež pozdĺž oboch osí. V dôsledku toho sa dynamický systém pohybuje po „cik-cak“ trajektóriách v stavovom priestore.

Siete typu 2 spĺňajú podmienku inverznosti väčších charakteristických hodnôt. Táto podmienka je nutná pre úspešnú akceptáciu a generalizáciu u sietí s dynamikou atraktor-sedlo [15]. Podmienku inverznosti možno formulovať ako

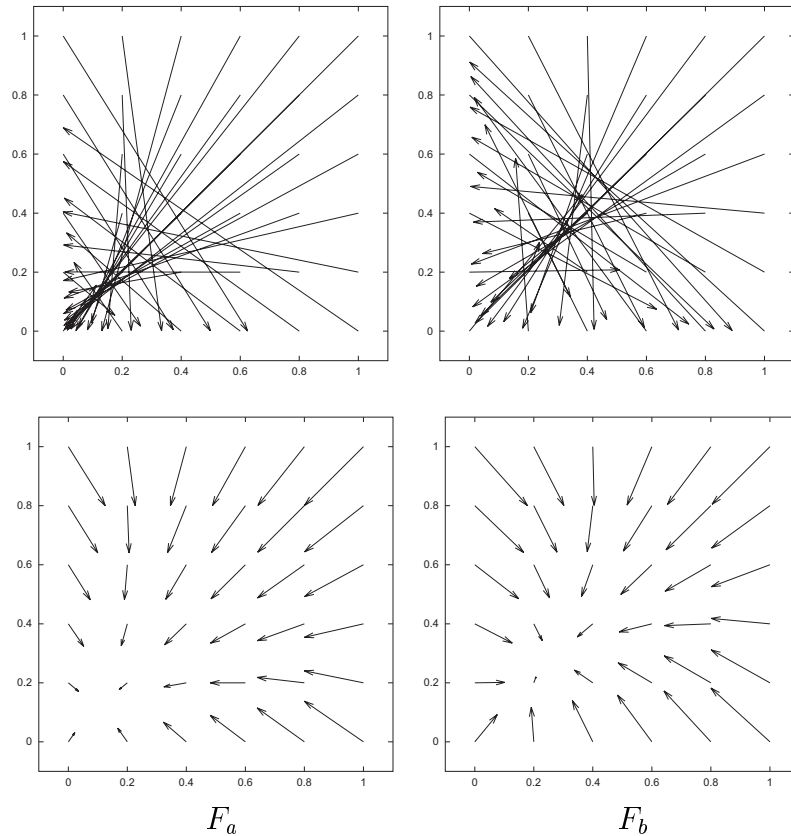
$$\lambda_{max}^a = \frac{1}{\lambda_{max}^b}, \quad (4.1)$$



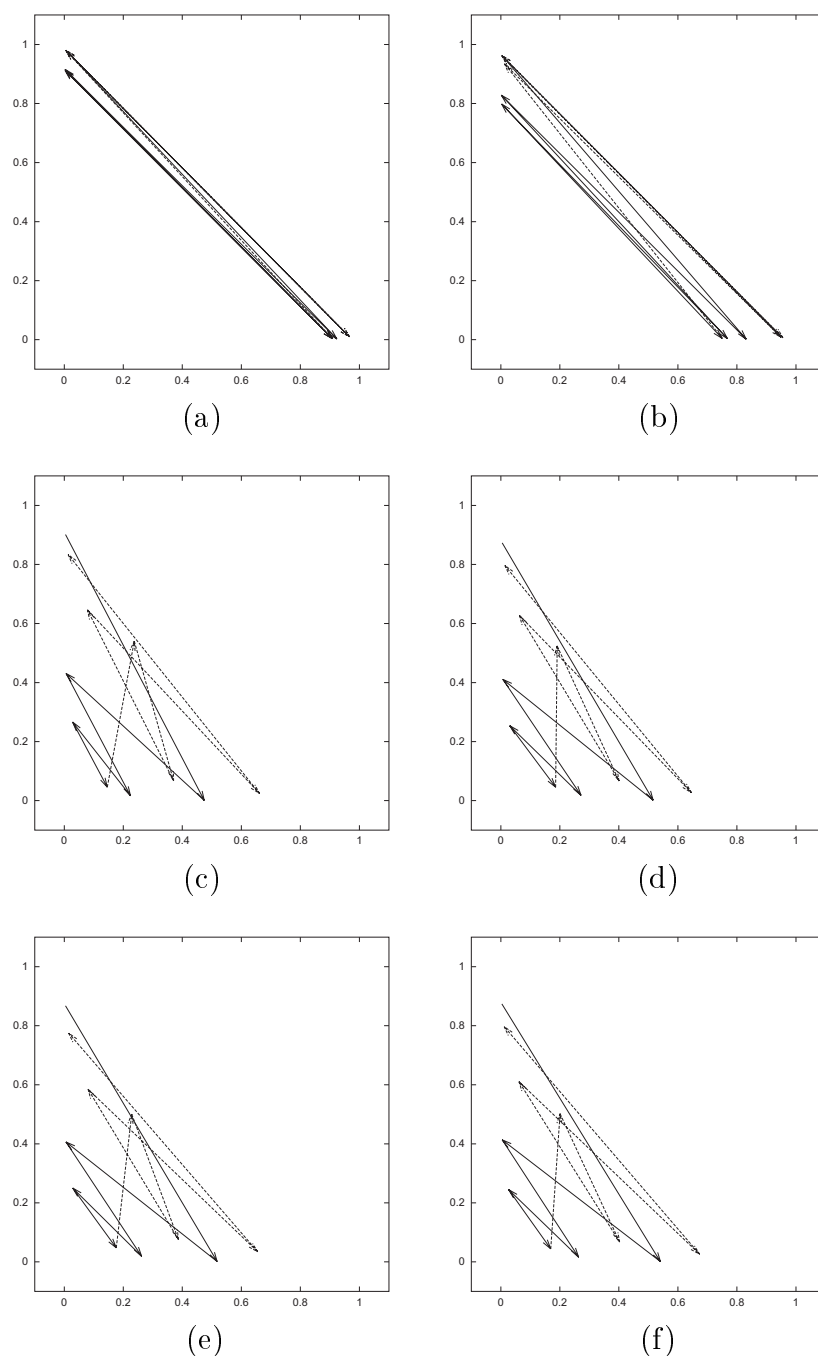
Obr. 4.8: (a) Trajektória slova a^2b^2 v stavovom priestore siete 2, po uplynutí 2000000 krokov učenia. (b) Stavový diagram zobrazuje rozmiestnenie pevných bodov, charakteristické vektory a dosiahnuteľné stavy siete 2. Krúžok predstavuje pevný bod dynamiky "a", trojuholník pevný bod dynamiky "b". Šípky pretínajúce sa v pevnom bode ukazujú smery lineárnej kontrakcie či expancie v jeho blízkom okolí (charakteristické vektory). V diagrame možno nájsť takisto množinu všetkých dosiahnuteľných stavov počas prechodu cez tréningovú množinu (malé krúžky). Na rozdiel od dynamiky dvoch atraktorov siete 1, sieť 2 obsahuje pevne body rôznych typov: atraktor (krúžok) a sedlo (trojuholník). Kombinácia vhodne rozmiestnených pevných bodov tohto typu umožňuje úspešnú akceptáciu i generalizáciu.

kde λ_{max}^i je maximálna z charakteristických hodnôt pevného bodu Fix_i , za predpokladu, že ignorujeme znamienka. V našom prípade sa $-0.76 \approx \frac{1}{-1.56}$, čo zaručuje „kvázi-inverzné“ správanie dynamiky "a" a "b". Slová za ideálnych okolností štartujú a končia exaktne v tom istom bode (na rozdiel od sietí typu 1). Finálny stav je navyše striktne oddelený od ostatných stavov, čo je podstatou lepšej generalizačnej schopnosti.

Bližšie o povahe pevných bodov siete typu 2 hovorí obr. 4.9. Vývoj trajektórií slov možno nájsť na obr. 4.10 a 4.8a. Zaujímavé sú tiež polohy pevných bodov a rozloženie dosiahnuteľných stavov vo fázovom priestore (pozri obr. 4.8b).



Obr. 4.9: Fázové diagramy dynamiky typu 2. Stĺpec F_a zachytáva dynamiku systému so znakom "a" na vstupe. Stĺpec F_b zachytáva dynamiku systému so znakom "b" na vstupe. Vo vrchných grafoch sú uvedené skutočné gradienty v jednotlivých bodoch stavového priestoru. Gradienty v spodných grafoch sú normalizované na $\frac{1}{5}$ originálnej dĺžky. Pevné body nie sú navzájom vzdialené, tak ako v prípade siete 1. Majú však odlišnú povahu, čo zapríčiňuje „kvázi-inverzné“ správanie dynamiky "a" a "b". Prvý atraktívny pevný bod sa nachádza približne v lokalite $[0.09, 0.11]$, druhý sedlový pevný bod sa nachádza v lokalite $[0.23, 0.25]$.



Obr. 4.10: Trajektórie slova a^5b^5 v stavovom priestore siete 2, v priebehu tréovania. Grafy zachytávajú situáciu po uplynutí určitého počtu krokov učenia: (a) 0, (b) 20, (c) 100, (d) 400, (e) 1000, (f) 2000 (všetky hodnoty $\times 10^3$). Možno si všimnúť, že po 1000000 iteráciách sa trajektórie slov takmer nemenia. Nasvedčujú tomu tiež výsledky experimentu popisovaného v kapitole 3 (pozri obr. 3.2, 3.3). Šípky kreslené plnou čiarou reprezentujú zmenu stavu s príchodom symbolu "a". Prerušované šípky reprezentujú zmenu stavu s príchodom symbolu "b".

Kapitola 5

Diskusia

5.1 Akceptácia

Ako sme ukázali v kapitole 3, rekurentná BCM sieť je schopná naučiť sa jazyk $L_{a^n b^n}$ s istou mierou zovšeobecnenia. Táto schopnosť implikuje vnútorné reprezentácie, ktoré separujú navzájom nesúvisiace podnety.

Pokiaľ trénovaciu množinu chápeme ako sekvenciu symbolov, bez ohľadu na konce slov, sú výsledky RBCM sietí veľmi uspokojivé (pozri obr. 3.2 a 3.3) [18]. Za vhodných podmienok kleslo NNL predikčného modelu, vybudovaného nad sieťou s „cik-cak“ dynamikou, až takmer na 0. Znamená to, že model štatisticky veľmi dobre vystihuje povahu predkladanej sekvencie.

Pokiaľ chápeme trénovaciu vzorku ako množinu slov, ktoré sú siete predkladané v náhodnom poradí, situácia o čosi horšia. Výraznejšie sa prejavuje závislosť trénovania na jeho parametrov či počiatočných podmienkach. Akceptačná úloha kladie veľký dôraz na varianciu stavov vo fázovom priestore, veľká variancia je predpokladom lepších akceptačných výsledkov.

Obidva analyzované druhy sietí, siete s cyklickou i „cik-cak“ dynamikou, sú typické vzdialenými pevnými bodmi a relatívne plochými spádovými oblasťami (absolútne hodnoty kontrakčných či expanzných koeficientov sú rádovo rovné 1). Plochosť zapríčiňuje veľkú varianciu stavov a tým tiež lepšiu akceptačnú predispozíciu.

V súvislosti s konštrukciou predikčných modelov, je dôležitou vlastnosťou učenia tiež schopnosť odhaliť charakteristické črty trénovacej vzorky. Učenie takýmto spôsobom determinuje vnútorné reprezentácie siete, ktoré môžu byť viac alebo menej adekvátne. Experimentálne sme potvrdili, že RBCM

siete sú schopné naučiť sa správnym spôsobom reprezentovať počítadlo s ohraničenou hĺbkou.

5.2 Generalizácia

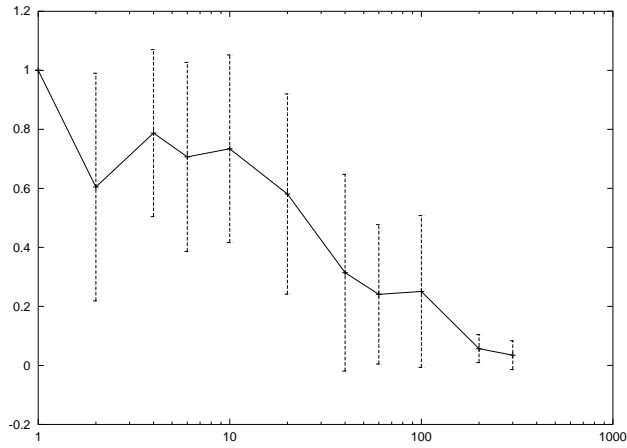
Schopnosť generalizácie objavujúca sa u RBCM siete je zreteľne závislá na parametroch a dĺžke tréovania. Uvedené boli dva druhy dynamík, z ktorých sa každá vyznačovala inou úrovňou generalizácie. Siete s cyklickou dynamikou neboli za žiadnych okolností schopné zovšeobecňovania. Na druhej strane, siete typu 2 („cik-cak” dynamika) preukázali generalizačnú schopnosť takmer na úrovni SRN tréovanej pomocou BPTT [15].

Pozrime sa teraz bližšie na predikčnú chybu jednotlivých sietí, špeciálne na pozície, v ktorých táto chyba nastala. V prípade jazyka $L_{a^n b^n}$ je možné rozdeliť predikovatelné prechody do dvoch základných skupín: 1) predikcie typu $b \rightarrow a$, 2) predikcie typu $b \rightarrow b$. Vo všeobecnosti sú $b \rightarrow b$ predikcie vo výraznej prevahe nad predikciami typu $b \rightarrow a$. Zo štatistického pohľadu sú preto $b \rightarrow b$ predikcie významnejšie než $b \rightarrow a$ predikcie, hoci z akceptáčného pohľadu sú si rovnocenné. Keďže generalizácia stojí na akceptácii celých slov, je nutné od neuronálneho modelu požadovať správnu predikciu pri oboch typoch prechodov (pozn. PERR má binárne príspevky: správne / nesprávne).

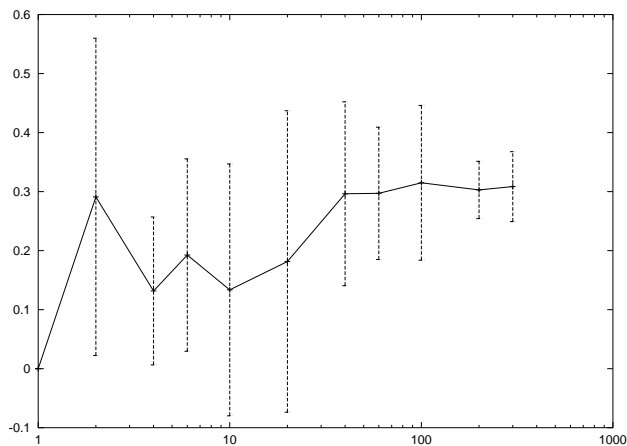
Selektívna predikčná chyba $b \rightarrow a$ natréovanej siete typu 1, klesá s počtom klastrovacích centier (pozri obr. 5.1a). Podobný proces možno sledovať aj v prípade sietí typu 2 (pozri obr. 5.2a). Experiment však ukázal, že v prípade generalizujúcich sietí typu 2, predikčná chyba $b \rightarrow b$ klesá so stúpajúcim počtom centier, na rozdiel od sietí typu 1 (pozri obr. 5.1b a 5.2b). Potvrdilo sa tým, že dynamika dvoch atraktorov nie je vhodná na reprezentáciu počítadla, je schopná obsiahnuť iba obmedzenú množinu slov. Naopak dynamika typu atraktor-sedlo (typ 2) je schopná úspešne reprezentovať jazyk $L_{a^n b^n}$.

5.3 Záver

Ako sme ukázali v kapitole 3 a 4, rekurentná BCM neurónová sieť je za istých okolností schopná naučiť sa „správne počítať” na príkladoch z jazyka $L_{a^n b^n}$. Vnútorne reprezentácie dobre natréovanej RBCM siete pozostávajú z dvoch

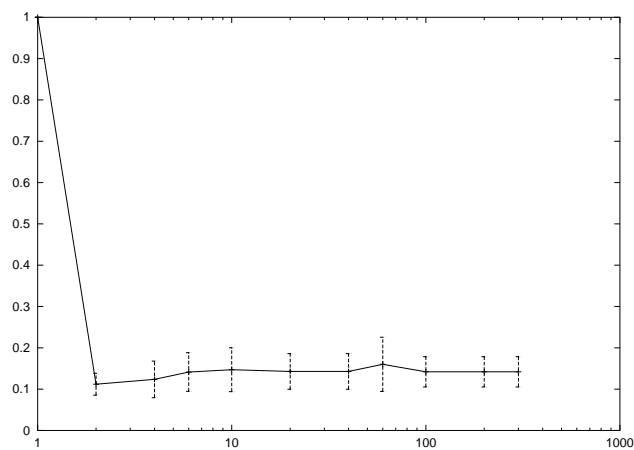


(a)

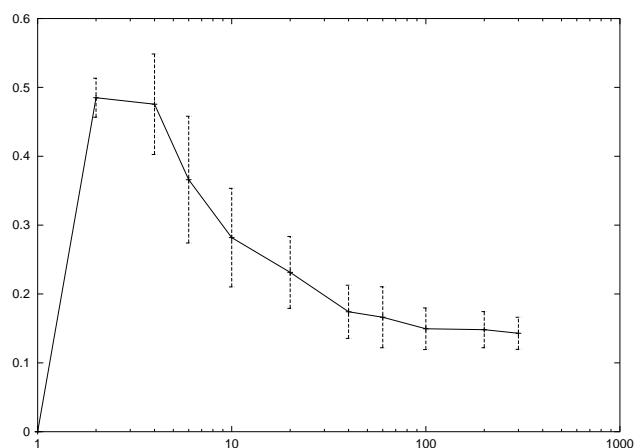


(b)

Obr. 5.1: Selektívna chyba predikcie pre sieť s cyklickou dynamikou (typ 1), v závislosti od počtu klastrovacích centier. Grafy zobrazujú percento chybných predikcií v (a) $b \rightarrow a$ prechodoch, (b) $b \rightarrow b$ prechodoch. Možno si všimnúť, že predikčná chyba $b \rightarrow a$ klesá so stúpajúcim počtom klastrovacích centier. Počet chybných $b \rightarrow b$ predikcií však postupne rastie. (Os x - počet klastrovacích centier, os y - (a) $PERR_{b \rightarrow a}$ / (b) $PERR_{b \rightarrow b}$.)



(a)



(b)

Obr. 5.2: Selektívna chyba predikcie pre siete s „cik-cak” dynamikou, v závislosti od počtu klastrovacích centier (bližší popis grafov možno nájsť pri obr. 5.1). Na rozdiel od sietí s cyklickou dynamikou (typ 1), siete s „cik-cak” dynamikou (typ 2) preukazujú istú schopnosť generalizácie. Z grafov vidno, že predikčná chyba $b \rightarrow b$ klesá zároveň s chybou $b \rightarrow a$. (Os x - počet klastrovacích centier, os y - (a) $PERR_{b \rightarrow a}$ / (b) $PERR_{b \rightarrow b}$.)

vyvážených pevných bodov typu atraktor a sedlo, podobne ako pri SRN [15]. Tento druh dynamiky dovoľuje vybudovať predikčný model so schopnosťou zovšeobecňovať naučenú informáciu.

Úspešnosť (konvergencia) RBCM sietí je veľkou mierou závislá na štartovacích podmienkach a parametroch učenia. Perspektívne by preto bolo zaujímavé venovať sa hľadaniu či extrakcii optimálnych parametrov učenia pre daný problém.

Zaujímavé môže byť tiež nahradenie pravdepodobnostného predikčného modelu doprednou neurónovou sieťou, napojenou na výstupnú vrstvu rekurentnej BCM siete. Išlo by o neuronálnu alternatívu pravdepodobnostného predikčného mechanizmu. Založená by bola na rozdeľovaní stavového priestoru RBCM siete do oblastí s rôznou dôležitosťou i funkčným ohodnotením. Učenie takéhoto modelu by eventuálne mohlo prebiehať na báze Hebbovho učiaceho pravidla s implicitným normovaním. Takéto učenie by zodpovedalo za vytváranie asociácií medzi vnútornými stavmi RBCM siete a požadovanými výstupmi. Prekonaním spomínaných problémov by mohol vzniknúť samoorganizujúci neuronálny model, nevyžadujúci žiaden externý štatistický aparát.

Dodatok A

Definícia dynamického systému

V tejto kapitole formálnejšie zdefinujeme pojem dynamického systému (DS), ktorý je kľúčovým pri popise správania sietí v okolí pevných bodov.

1. Dynamický systém s diskretným časom možno chápať ako iteráciu diferencovateľnej funkcie:

$$f : \mathcal{R}_n \rightarrow \mathcal{R}_n, \text{ napr. } \vec{x}(t+1) = f(\vec{x}(t)), t \in N, \vec{x} \in \mathcal{R}_n,$$

kde N je množina prirodzených čísel a \mathcal{R}_n n -rozmerný priestor nad reálnymi číslami. Ak funkcia f spĺňa vlastnosť superpozície¹, dynamický systém je lineárny, inak je nelineárny. RNN so sigmoidou na výstupe je príkladom nelineárneho dynamického systému.

2. Postupné iterácie generujú sekvenciu bodov určujúcich trajektóriu systému štartujúceho z bodu $x_0 \in \mathcal{R}_n$.

$$\vec{x}(0), \vec{x}(1) = f(\vec{x}(0)), \vec{x}(2) = f(\vec{x}(1)) = f^2(\vec{x}(0)), \dots$$

3. Bod \vec{x} sa nazýva pevným bodom funkcie f ak $\forall m \in N : f^m(\vec{x}) = \vec{x}$.
4. Pevný bod \vec{x} sa nazýva atraktor, ak existuje také jeho okolie, pre ktoré platí, že $\forall \vec{y} \in \mathcal{O}(\vec{x}) : \lim_{m \rightarrow \infty} f^m(\vec{y}) = \vec{x}$
5. Pevný bod \vec{x} sa nazýva repel, ak existuje také jeho okolie, pre ktoré platí, že $\forall \vec{y} \in \mathcal{O}(\vec{x}) : \lim_{m \rightarrow -\infty} f^m(\vec{y}) = \vec{x}$

¹Vlastnosť superpozície: $f(a\vec{x} + b\vec{y}) = af(\vec{x}) + bf(\vec{y})$, kde a a b sú skaláry, \vec{x} a \vec{y} sú vektory.

6. Bod \vec{x} nazývame n-periodickým pevným bodom ak $f^n(\vec{x}) = \vec{x}$.
7. Dynamický systém možno zdefinovať i ako systém rovníc (funkčných závislostí) $F = \{f_i : \mathcal{R}_n \rightarrow \mathcal{R}, i = 1, 2, \dots\}$.

Príklad dvojrozmerného DS:

$$\begin{aligned}x_1(t+1) &= f_1(x_1(t), x_2(t)) \\x_2(t+1) &= f_2(x_1(t), x_2(t))\end{aligned}$$

8. Krok výpočtu lineárneho dynamického systému možno zapísať ako

$$\vec{X}(t+1) = F \otimes \vec{X}(t), \quad (\text{A.1})$$

kde F je matica lineárneho zobrazenia a $\vec{X}(t)$ je vektor reprezentujúci stav systému v čase t .

9. Charakteristické hodnoty λ a vektory v sú tie, pre ktoré platí $F \otimes v = \lambda v$. Pre lineárny systém n-rovníc určujú charakteristické hodnoty veľkosť a povahu kontrakcie či expanzie. Charakteristické vektory určujú osi, pozdĺž ktorých sa systém lineárne deformuje. Pre každý n-rozmerný lineárny systém existuje práve n charakteristických hodnôt a vektorov. V prípade, že lineárny systém aproximuje nelineárny systém v jednom z jeho pevných bodov, charakteristické hodnoty môžu pomôcť popísať povahu daného pevného bodu.

- (a) ak $\forall i : |\lambda_i| < 1$, analyzovaný bod je atraktor
- (b) ak $\forall i : |\lambda_i| > 1$, analyzovaný bod je repelior
- (c) ak $\exists i : |\lambda_i| > 1$ a $\forall j \neq i : |\lambda_j| < 1$ analyzovaný bod je sedlo
- (d) ak $\exists i : |\lambda_i| = 1$ na analýzu je nutná iná metóda

10. Nech F je nelineárny systém funkcií a nech J je matica parciálnych derivácií systému F podľa jednotlivých premenných (Jakobián). Potom $\mathcal{F} = J(a)$ je matica lineárneho systému aproximujúceho správanie F v bode a .

11. V prípade nelineárnych systémov nazývame analógie charakteristických hodnôt množinou stabilných a nestabilných variet (\mathcal{M}_S a \mathcal{M}_U)².

$$\begin{aligned}\mathcal{M}_S &= \{\vec{x} \mid \forall m : F^m(\vec{x}) \in W_S \wedge \lim_{m \rightarrow \infty} F^m(\vec{x}) = \vec{x}_{fix}\} \\ \mathcal{M}_U &= \{\vec{x} \mid \forall m : F^m(\vec{x}) \in W_U \wedge \lim_{m \rightarrow -\infty} F^m(\vec{x}) = \vec{x}_{fix}\}\end{aligned}$$

²Stable / Unstable Manifold (\mathcal{M}_S / \mathcal{M}_U)

Ide o analyticky ťažko detekovateľné a spracovateľné veličiny, preto sa v praxi uprednostňujú aproximačné metódy.

Použitá literatúra

- [1] C. M. Bachman, S. A. Musman, D. Luong, and A. Schultz. Unsupervised BCM projection pursuit algorithms for classification of simulated radar presentations. *Neural Networks*, 7:709–728, 1994.
- [2] E. L. Bienenstock, L.N. Cooper, and P. W. Munro. Theory of the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Neuroscience*, 2:32–48, 1982.
- [3] M. Bodén, H. Jacobsson, and T. Ziemke. Evolving context-free language predictors. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1033–1040, 2000.
- [4] J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14:179–211, 1990.
- [5] J. L. Elman. *Language as a dynamical system. In R. F. Port & T. van Gelder (Eds.), Mind as Motion: Explorations in the Dynamics of Cognition, pp. 195-223.* Cambridge, MA: MIT Press, 1995.
- [6] J. E. Hopcroft and J. D. Ullman. *Formálne jazyky a automaty.* Alfa, 1978.
- [7] R. M. Hristev. *The ANN Book.* 1998.
- [8] N. Intrator and L. N. Cooper. Objective Function Formulation of the BCM Theory of Visual Cortical Plasticity: Statistical Connections, Stability Conditions. *Neural Networks*, 5:3–17, 1992.
- [9] V. Kvasnička, Ľ. Beňušková, J. Pospíchal, I. Farkaš, P. Tiňo, and A. Kráľ. *Úvod do teórie neurónových sietí.* IRIS, 1979.

- [10] O. Melnik, S. Levy, and J. Pollack. *RAAM for Infinite Context-Free Languages. IJCNN 2000*. IEEE Press, 2000.
- [11] C. W. Omlin and C. L. Giles. Constructing Deterministic Finite-State Automata in Recurrent Neural Networks. *Journal of the ACM*, 45(6):937, 1996.
- [12] P. Petrovič, P. Tiño, and Ľ. Beňušková. Processing Symbolic Sequences by the BCM Neuron. *Neural Network World*, 8(5):491–500, 1995.
- [13] J. Pollack. The introduction of dynamical recognizers. *Machine Learning*, 7:227–252, 1991.
- [14] H. Ritter and T. Kohonen. Self-Organizing Semantic Maps. *Biological Cybernetics*, 61:241–254, 1989.
- [15] P. Rodriguez, J. Wiles, and J. L. Elman. A Recurrent Neural Network that Learns to Count. *Connection Science*, 11:5–40, 1999.
- [16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), Parallel Distributed Processing, Vol. 2*. Cambridge, MA: MIT Press, 1986.
- [17] H. T. Siegelmann. *Foundations of Recurrent Neural Networks*. PhD dissertation, New Brunswick Rutgers, The State University of New Jersey, 1993.
- [18] P. Tiño, M. Stančík, and Ľ. Beňušková. Building predictive models on complex symbolic sequences via a first-order recurrent BCM network with lateral inhibition. *Proc. IEEE-INNS-ENNS Int. Joint Conference on Neural Networks*, 2:265–270, 2000.
- [19] J. Wiles and J. L. Elman. *Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. Processing of the Seventeenth Annual Conference of Cognitive Science Society*. Cambridge, MA: MIT Press, 1995.

Zoznam použitých skratiek

BPTT	Backpropagation Through Time
CPW	Correctly Predicted Words
DS	Dynamical System
FIFO	First In, First Out
FSA	Finite State Automaton
GA	Genetical Algorithm
NNL	Normalized Negative Log-Likelihood
OCR	Optical Character Recognition
PDA	Push-Down Automaton
PDP	Paralel Distributed Processing
PERR	Prediction Error
RBCMN	Recurrent BCM Network
RNN	Recurrent Neural Network
SOM	Self-Organizing Map
SRN	Simple Recurrent Network
TS	Turingov stroj