

Spracovanie viet jazyka pomocou neurónovej siete s echo stavmi

Bakalárska práca

Peter Galiovský

UNIVERZITA KOMENSKÉHO
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDA INFORMATIKY



9.2.1 Informatika

Vedúci práce
doc. Ing. Igor Farkaš, PhD.

Bratislava
jún, 2008

Čestne vyhlasujem, že som bakalársku prácu vypracoval samostatne, len s použitím uvedenej literatúry a pod odborným dohľadom školiteľa.

Bratislava, jún 2008

Peter Galiovský

Moje poďakovanie patrí môjmu školiteľovi, doc. Ing. Igorovi Farkašovi, PhD., ktorý ma k napísaniu práce inšpiroval a bez ktorého ochoty by nikdy nezvnikla.

Abstrakt

Prijateľné modely spracovania viet jazyka musia mať schopnosť syntaktickej systematickosti – vedieť zovšeobecňovať z malej množiny tréningových viet. Frank skúmal [6], do akej miery túto vlastnosť prejavujú neurónové siete s echo stavmi na úlohe predikcie nasledujúceho slova. Ukázal, že siete s echo stavmi sa správajú tzv. slabo systematicky a vyslovil domnienku, že s narastajúcou veľkosťou lexikónu sa bude ich správanie naďalej zlepšovať. Najprv overujeme jeho výsledky, potom na ne nadväzujeme simuláciami s použitím väčších lexikónov a dynamických rezervoárov. Ukazujeme, že zväčšovanie lexikónu môže, za podmienky použitia dostatočne veľkého dynamického rezervoáru, viesť k ďalšiemu zlepšeniu systematickosti sietí.

Kľúčové slová: rekurentná neurónová sieť, echo stavy, systematickosť, predikcia nasledujúceho slova

Obsah

1	Úvod	1
1.1	Syntaktická systematickosť	1
1.2	Siete s echo stavmi	2
2	Publikované práce	4
2.1	Van der Velde, Van der Voort van der Kleij, De Kamps	4
2.1.1	Trénovacia a testovacia množina	4
2.1.2	Simulácie a ich výsledky	5
2.2	Frank	7
2.2.1	Kritika záverov Van der Veldeho a kol.	7
2.2.2	Trénovacia a testovacia množina	7
2.2.3	Miera výkonnosti siete – FGP	8
2.2.4	Simulácie s jednoduchými rekurentnými sieťami	9
2.2.5	Simulácie so sieťami s echo stavmi	10
3	Simulácie	13
3.1	Trénovanie úplnou trénovacou množinou	13
3.2	Trénovanie neúplnou trénovacou množinou	19
4	Záver	30

1 Úvod

Na úvod práce si vysvetlíme pojem syntaktickej systematickosti (1.1) a priblížime model siete, ktorej implementácia bola časťou tejto práce – siete s echo stavmi (1.2).

V ďalšej časti sa potom venujeme dvom článkom, z ktorých sme vychádzali pri vlastných simuláciách. Sú to články Van der Veldeho a kol. [18], ktorí skúmali syntaktickú systematickosť jednoduchých rekurentných sietí (2.1), a Franka [6], ktorý Van der Veldeho výsledky kritizoval a v podobných experimentoch pokračoval aj so sieťami s echo stavmi (2.2).

Tretia časť bude venovaná simuláciám na nami implementovanej sieti s echo stavmi a ich výsledkom (3).

V závere potom výsledky zhrnieme a navrhujeme možné pokračovania práce.

1.1 Syntaktická systematickosť

Jednou z charakteristických črt ľudského jazyka je schopnosť kombinovaním obmedzeného počtu slov s vetnými štruktúrami vytvoriť ohromné množstvo gramaticky správnych viet, vrátane takých, s ktorými sme sa nikdy predtým nestretli. Ľudia navyše nemajú problém tieto vety nielen vytvoriť, ale vedia ich aj bez problémov používať či s nimi manipulovať – bezohľadu na to, či sú tieto vety zmysluplné alebo nie. V [18] sa uvádza ako príklad veta *domy jedia nábytok*, ktorá je síce nezmyselná, ale ľudia nemajú žiadne ťažkosti odpovedať na otázku, kto/čo konzumuje nábytok.

Akýkoľvek model prirodzeného jazyka musí vedieť túto schopnosť – *syntaktickú systematickosť* – ak nie vysvetliť, tak aspoň preukázať. V dnes už klasickom článku [5] Fodor a Pylyshyn argumentujú, že konekcionistické modely nemôžu touto schopnosťou disponovať, pokiaľ nie sú priamou implementáciou modelu symbolického. V následnej debate, ktorá dodnes neustala, začalo množstvo autorov uvádzať kontrapríklady. Hadley v [9] tvrdil, že žiaden z nich nie je dostatočný.

Hadley ďalej uvažuje, že systematickosť je dôsledkom učenia a zovšeobecňovania. Neurónová sieť by sa mala na základe učenia na tréningových vetách naučiť spracovať aj všetky ostatné vety. Ďalej tvrdil, že pokiaľ ľuďom postačuje na získanie systematickosti v používaní jazyka prísť do styku len s veľmi malým zlomkom možných viet, rovnako neurónovej sieti, ktorá má disponovať kognitívne prijateľnou syntaktickou systematickosťou, musí na jej naučenie stačiť podobne malý zlomok viet z celkového možného počtu.

Systematickosť rozdelil na silnú a slabú. Siete označil ako *slabo systematické*, ak dokážu

úspešne spracovať novú vstupnú vetu, no so slovami na takých vetných pozíciách, kde sa už vyskytli počas tréovania. *Silno systematická* potom bude sieť, ktorá dokáže úspešne spracovať aj vetu so slovami na nových syntaktických pozíciách. Pokiaľ teda máme v tréovacej množine vety *chlapec ľúbi dievča* a *pes naháňa mačku*, slabo systematická sieť by si mala poradiť aj s vetou *chlapec naháňa mačku*. Pokiaľ sa navyše v tréovacej množine nikdy nevyskytol chlapec ako predmet vety, silno systematická sieť by mala vedieť korektne spracovať aj vetu *pes naháňa chlapca*.

Podľa Hadleyho teda neurónová sieť prijateľná ako model ľudského jazyka musí vykazovať silnú systematickosť. Tvrdil však, že všetky dovtedajšie modely boli prinajlepšom slabo systematické.

Azda najpoužívanejším spôsobom skúmania systematickosti u neurónových sietí je úloha *predikcie nasledujúceho slova*. Sieti sú postupne slovo po slove predkladané vety a úlohou siete je predpovedať, aké slovo bude na vstupe nasledovať v ďalšom kroku. Na riešenie tejto úlohy sa najčastejšie používajú *jednoduché rekurentné siete* (simple recurrent networks, SRN; nazývané podľa ich autora aj Elmanove siete, [1]). Jednoduchá rekurentná sieť sa architektúrou podobná trojvrstvej doprednej sieti. Aktivácia neurónov prostrednej – rekurentnej – vrstvy sa však ukladá a používa v ďalšom časovom kroku ako prídavný vstup.

Van der Velde a kol. v [18] tvrdia, že aj slabá systematickosť je nad možnosť SRN. Frank však tento záver spochybňuje [6] a okrem SRN skúma aj systematickosť na inom modeli neurónových sietí, *sieťach s echo stavmi* (echo state networks, ESN; [13]). Práve Frankov článok sa stal hlavným východiskom pre túto prácu, ktorá si dala za cieľ implementovať sieť s echo stavmi, overiť Frankove výsledky a nadviazať na ne ďalšími simuláciami. Oba články podrobnejšie popisujú časti 2.1 a 2.2. O architektúre sietí s echo stavmi stručne hovorí časť nasledujúca.

1.2 Siete s echo stavmi

V rokoch 2001 a 2002 sa nezávisle na sebe objavili dva príbuzné nové modely rekurentných neurónových sietí: *siete s echo stavmi* (echo-state networks, ESN; [10]) a *automaty s kvapalinovými stavmi* (liquid state machines; [16]). Obe spájala spoločná myšlienka použiť ako rekurentnú vrstvu náhodne inicializovaných, náhodne poprepájanú, netrénovanú neurónovú sieť – *dynamický rezervoár* – a požadovaný výstup získať ako lineárnu kombináciu jej aktivácií. Preto sa čoraz viac zaraďujú pod spoločnú hlavičku *rezervoárové počítanie*.

Použitie netrénovanú rekurentnú sieť však nie je úplne nová myšlienka. O *architekturálnom biase*, teda schopnosti rekurentnej siete vykazovať isté užitočné predikčné vlastnosti už pred

učením, sa píše napríklad v [17].

Názov automatov s kvapalinovými stavmi vychádza z autormi uvedenej analógie, ktorá fungovanie modelu prirovnávala k hodeniu kameňa do vody: vstup (pád kameňa) bol pretransformovaný na časopriestorový vzor (sčerenú hladinu). Ako “kvapalina” sa zväčša používa sieť s impulznými neurónmi. V iných prácach sa však objavilo napríklad aj riešenie XOR problému za použitia vedierka skutočnej vody. [4] Automatom s kvapalinovými stavmi sa vo všeobecnosti venuje pozornosť skôr kvôli ich prípadnej biologickej plauzibilite.

Siete s echo stavmi boli naproti tomu od počiatku cielené najmä na predikciu náhodných časových radov v praktických aplikáciách. Pomalé a komplikované učenie bol podľa Jaegera hlavný dôvod, prečo sa SRN doteraz nepresadili v praktických aplikáciách. ESN si môžeme predstaviť aj ako tradičnú SRN so vstupnou, rekurentnou a výstupnou vrstvou, v ktorej ale budeme po náhodnej inicializácii upravovať len váhy spojené z rekurentnej vrstvy do výstupných neurónov.

Pre funkčnosť ESN je nevyhnutné, aby rezervoár vykazoval *vlastnosť echo stavov*. Tá zjednodušene vraví, že rekurentná sieť sa musí pre rovnakú vstupnú sekvenciu po určitom čase ustáliť na rovnakej aktivácii bezohľadu na svoj počiatkový stav. Empiricky bolo zistené, že požadovaná vlastnosť platí, pokiaľ má váhová matica rezervoáru spektrálny polomer menší než 1. Spektrálny polomer váhovej matice rezervoáru okrem toho určuje aj rýchlosť, s akou v rezervoári odznievajú vstupné impulzy – väčší spektrálny polomer má za následok pomalšie odznievanie. Spektrálny polomer váhovej matice tak treba empiricky prispôbiť konkrétnej úlohe.

V literatúre sa ako ďalšia dôležitá vlastnosť váhovej matice rezervoára zvykne uvádzať jej *riedka náhodná prepojenosť* (sparse connectivity). Podľa [13] však prax ukázala, že plne prepojené modely vykazujú rovnakú výkonnosť.

Pre aplikácie predpovedajúce časový rad sa najčastejšie používa nelineárny rezervoár s tangenciálnou sigmoidálnou aktivačnou funkciou. Do rezervoáru je navyše zavedená spätná väzba z výstupných jednotiek. ESN sa potom učí off-line – lineárnou regresiou výstupných váh hľadáme spôsob, akým signály z rezervoáru čo najlepšie skombinovať do požadovaného výstupu. Podrobnejšie pozri [12] alebo [13]. ESN boli veľmi úspešne použité napríklad na chybovú korekciu bezdrôtovej komunikácie [15] alebo finančné prognózy [14].

Frank však ukázal [6], že ako off-line tréning, tak použitie nelineárneho rezervoára zhoršovalo výsledky v úlohe predikcie nasledujúceho slova. Budeme preto používať lineárny rezervoár a on-line tréning váh pomocou algoritmu spätného šírenia chyby. Viac v časti 2.2.5.

2 Publikované práce

2.1 Van der Velde, Van der Voort van der Kleij, De Kamps

2.1.1 Trénovacia a testovacia množina

Van der Velde a kol. [18] skúmali syntaktickú systematickosť jednoduchých rekurentných sietí (SRN) riešiacich úlohu predikcie nasledovného slova. Použitý lexikón o veľkosti 18 (anglických) slov pozostával z ôsmich podstatných mien, ôsmich sloves, zámena *who* a slova označujúceho koniec vety (ekvivalent bodky v písanom prirodzenom jazyku). Trénovacie a testovacie množiny boli tvorené tromi druhmi viet:

- **jednoduché vety** (simple sentences) tvaru podstatné meno – sloveso – podstatné meno (N-V-N)
napr. *Boy sees girl.*
- **súvetia s vedľajšou vetou pripojenou na konci** (right-branching complex sentences) tvaru podstatné meno – sloveso – podstatné meno – *who* – sloveso – podstatné meno (N-V-N-*who*-V-N)
napr. *Boy sees girl who hears girl.*
- **súvetia s vedľajšou vetou vloženou doprostred** (centre-embedded complex sentences) tvaru podstatné meno – *who* – podstatné meno – sloveso – sloveso – podstatné meno (N-*who*-N-V-V-N)
napr. *Boy who boy hears sees girl.*

Každé podstatné meno a sloveso sa dalo vo vete použiť na ľubovoľnej pozícii bez narušenia jej gramatickosti. “Obsahové” slová (podstatné mená a slovesá) boli rozdelené do štyroch skupín, v každej boli dve podstatné mená a dve slovesá.

σ	\rightarrow	Simple Right Centre	N	\rightarrow	N_1 N_2 N_3 N_4
Simple	\rightarrow	N V N .	V	\rightarrow	V_1 V_2 V_3 V_4
Right	\rightarrow	N V N <i>who</i> V N .	N_x	\rightarrow	$n_{x,1}$ $n_{x,2}$
Centre	\rightarrow	N <i>who</i> N V V N .	V_x	\rightarrow	$v_{x,1}$ $v_{x,2}$

Tabuľka 1: Gramatika použitá pre generovanie trénovacích a testovacích viet.

Trénovacia množina pozostávala zo všetkých gramaticky správnych viet, z ktorých každá sa skladala vždy len slov z rovnakej skupiny. Vety v trénovacej množine mali teda tvar $n_{x,i} v_{x,j} n_{x,k}$.

(jednoduché), $n_{x,i} v_{x,j} n_{x,k}$ who $v_{x,l} n_{x,m}$. (súvetia s v. v. na konci) a $n_{x,i}$ who $n_{x,j} v_{x,k} v_{x,l} n_{x,m}$. (súvetia s v. v. uprostred).

Ak prvá skupina pozostávala z podstatných mien *boy* a *girl* a sloviess *sees* a *hears*, druhá skupina z podstatných mien *dog* a *cat* a sloviess *chases* a *kicks*, do trénovacej množiny patrili napríklad vety *boy sees girl*, *girl hears boy* alebo *dog chases cat*, nie však už vety *boy chases cat* alebo *dog sees girl*. Trénovacia množina teda obsahovala celkom 32 jednoduchých viet a 256 súvetí.

Lexikón bol kódovaný lokalisticky. Každému slovu bol priradený jedinečný vektor, v ktorom mala práve jedna zložka hodnotu 1, ostatné boli nulové. Vo vstupnej a výstupnej vrstve tak vždy jeden neurón zodpovedal jednému konkrétnemu slovu z lexikóna.

2.1.2 Simulácie a ich výsledky

Trénovaných bolo desať sietí, uvádzané výsledky boli spriemerované. Siete mali päť vrstiev: vstupnú a výstupnú o veľkosti 20 neurónov¹, rekurentnú vrstvu s veľkosťou taktiež 20 neurónov a dve skryté vrstvy po 10 neurónov – jednu medzi vstupnou a rekurentnou vrstvou a jednu medzi rekurentnou a výstupnou vrstvou.

Trénovanie bolo rozdelené na štyri fázy, počas ktorých sa postupne menil pomer medzi jednoduchými vetami a súvetiami od 1:0 až po 1:4. Vety z trénovacej množiny boli náhodne zoradené a zreťazené. Sieťam boli postupne predkladané slovo po slove. Po slove označujúcom koniec vety bolo ako ďalší vstup v poradí predložené prvé slovo z nasledujúcej vety. Úlohou siete bolo predikovať slovo, ktoré bude vo vstupnom prúde nasledovať. Počas učenia bolo sietí predložených celkom 157 440 viet. Následne boli váhy siete zafixované.

Počas testovania boli sieťam predkladané vety pozostávajúce z podstatných slov a sloviess z N skupín, s N z rozsahu 1 až 3 pre jednoduché vety a 1 až 4 pre súvetia. (Pre $N = 1$ to teda boli vety obsiahnuté aj v trénovacej množine.) Ako miera správnosti predikcie bola použitá *gramatická chyba predikcie* (grammatical prediction error; GPE) zadaná ako podiel súčtu aktivácií výstupných jednotiek zodpovedajúcich slovám, ktoré nie sú gramaticky správnym pokračovaním dosiaľ predloženého vstupu, a celkovej aktivácii výstupných jednotiek

$$GPE = \frac{a(-G)}{a(-G) + a(G)}$$

kde G je množina všetkých slov predstavujúcich gramatické pokračovanie dosiaľ predloženého

¹Hoci veľkosť lexikónu bola len 18, na kódovanie slov použili autori 20bitové vektory.

vstupu, $a(G) = \sum_{i \in G} a_i$, kde a_i je aktivácia výstupnej jednotky i , a $a(-G) = \sum_{i \notin G} a_i$.

Výsledky ukázali, že zatiaľ čo siete si vedeli celkom dobre poradiť s vetami z trénovacej množiny (priemerná GPE bola 0,061, v najproblematickejších vetných pozíciách sa pohybovala okolo 0,1), nové vstupy – teda vety vytvorené kombinovaním slov z rôznych skupín – spracúvali veľmi zle. Priemerné hodnoty GPE boli 0,250 pre $N = 2$, 0,273 pre $N = 3$ a 0,309 pre $N = 4$. Sieť mala najväčšie problémy s predikciou pri spracovaní štvrtých a šiestych slov zo súvetí. V prípade prvého slovesa v súvetí s v. v. uprostred boli priemerné hodnoty GPE až okolo 0,7 a aj v ostatných troch pozíciách sa pohybovali medzi 0,5 a 0,6.

Nízke hodnoty GPE pri testovaní na vetách z trénovacej množiny podľa autorov svedčia o tom, že sieť sa síce dokázala naučiť skladbu viet trénovacej množiny, nedokázala však už tieto poznatky zovšeobecniť a použiť ich pri spracovaní nových vstupov, resp. rozoznať vetnú štruktúru vstupnej sekvencie, s ktorou sa počas učenia nestretla. Analýza priebehu GPE podľa vetnej pozície vstupu viedla autorov k domnienke, že sieť sa pri spracúvaní viet so slovami z rôznych skupín snaží svoju predikciu založiť na základe asociácií dvojíc po sebe nasledujúcich slov odpozorovaných z trénovacích údajov. Výsledky ďalších simulácií, v ktorých ako vstupné údaje použili agramatický *slovný šalát*, boli v súlade s touto domnienkou. Autori preto dospeli k záveru, že testované siete neprejavili schopnosť slabej kombinatorickej systematickosti.

2.2 Frank

2.2.1 Kritika záverov Van der Veldeho a kol.

Na prácu Van der Veldeho a kol. reagoval v článku [6] Stefan L. Frank, podľa ktorého boli závery publikované v [18] unáhlené. Prvú výčitku smeroval voči odsúdeniu sietí ako nejaviacich schopnosť systematickosti bez porovnania so správaním siete, ktorá by bola preukázateľne úplne nesystematická. Ďalej kritizoval absenciu simulácií so sieťami rôznych veľkostí. Van der Veldeho a kol. zvolená veľkosť rekurentnej vrstvy mohla byť podľa Franka buď príliš malá na to, aby dokázala spracovať množstvo rôznych viet, alebo, naopak, príliš veľká vzhľadom k rozsahu tréningových údajov – vďaka veľkému množstvu voľných parametrov siete mohlo dôjsť k premodelovaniu dát (overfitting), kedy sa sieť namiesto všeobecnej štruktúry naučila rozoznávať individuálne prípady z tréningovej množiny, čím sa zhoršila jej schopnosť zovšeobecňovať. Z podobného dôvodu kritizoval aj malý rozsah použitého lexikónu. Rozsiahlejší lexikón by sieť nútil implicitne sa naučiť rozoznávať slovné druhy namiesto ukladania konkrétnych slov.

Frank množinu lexikónov rozšíril na štyri veľkosti $w \in \{18, 26, 34, 42\}$. Slovníky naďalej obsahovali zámeno *who* i slovo značiace koniec vety a boli rozdelené na štyri skupiny. Každú skupinu tvorilo $(w - 2)/8$ podstatných mien a rovnaký počet slovies. Gramatika použitá pre generovanie tréningovej a testovacej množiny bola rovnaká. Slová boli opäť kódované lokalisticky – vektorom s práve jednou zložkou o veľkosti 1, ostatnými zložkami nulovými, jedinečným pre každé slovo zo slovníka.

Veľkosť rekurentnej vrstvy a jej vplyv na výkonnosť siete boli predmetom skúmania. Veľkosť rekurentnej vrstvy v jednotlivých situáciách s pomocou SRN bola $n \in \{10, 20, 40, 70, 100\}$.

2.2.2 Tréningová a testovacia množina

Tréningová množina opäť pozostávala zo všetkých gramatických viet obsahujúcich vrámci jednej vety vždy len slová z jednej skupiny. Veľkosti tréningových množín sú uvedené v tabuľke 2. So zväčšujúcou sa veľkosťou lexikónov sa tak zároveň zmenšoval podiel, ktorý vrámci všetkých gramatických viet predstavovali tréningové vety. Pri $w = 42$ predstavovali tréningové vety len 0,398 % všetkých gramatických viet.

Na rozdiel od simulácií Van der Veldeho a kol., sieťam boli od počiatku učenia predkladané jednoduché vety a súvetia v pomere 1:4. Siete boli tréningované po dobu 10 epôch, po 31 250 viet²

²Pre $w = 42$ máme 25 000 rôznych tréningových súvetí, ktoré, doplnené jednoduchými vetami v pomere 1:4, vytvárajú celkový tréningový súbor o veľkosti 31 250 viet. Rovnaká veľkosť tréningového súboru bola použitá aj pri lexikónoch menších rozsahov.

v každej. Počas učenia im teda bolo predložených celkom 312 250 viet (= 2 000 000 slov).

V simuláciách Van der Veldeho a kol. dosahovali siete najhoršie výsledky pri testovaní na vetách zložených zo slov z najväčšieho možného počtu rôznych skupín. Frank preto testovacie množiny vytvoril zo všetkých “najťažších”jednoduchých viet – viet, kde každé z troch významových slov bolo z inej skupiny – a doplnil ich 100 súvetiami z každého druhu, obsahujúcimi významové slová vždy zo všetkých štyroch skupín.

2.2.3 Miera výkonnosti siete – FGP

Frank kritizoval použitie GPE na vyhodnocovanie zovšeobecňovacej schopnosti siete, pretože pri GPE absentuje základná úroveň zodpovedajúca sieti, ktorá vôbec nezovšeobecňuje. Preto preto navrhol inú mieru, ktorá porovnáva predikciu siete s predikciou bigramového štatistického modelu.

Nech uvažovaná sieť používa lokalistické kódovanie slov a nech výstupné neuróny používajú softmaxovú aktivačnú funkciu:

$$a_{out,i} = \frac{e^{v_i}}{\sum_j e^{v_j}}$$

kde $a_{out,i}$ je výsledná aktivácia neurónu i vo výstupnej vrstve a v_i je jeho celková vážená vstupná aktivácia. Celková aktivácia výstupnej vrstvy bude vždy rovná 1. Aktiváciu neurónu i preto môžeme interpretovať ako predikovanú pravdepodobnosť, s akou bude slovo i nasledujúcim vstupom.

Nech G je množina všetkých gramatických pokračovaní doterajšieho vstupu. Označme si $a(G)$ súčet aktivácií výstupných neurónov zodpovedajúcich slovám, ktoré tvoria gramatické pokračovanie doterajšieho vstupu, teda $a(G) = \sum_{i \in G} a_i$. Dokonale zovšeobecňujúca sieť bude mať aktívne práve tie výstupné jednotky, ktoré zodpovedajú gramaticky správnym pokračovaniam, teda $a(G) = 1$. Takejto predikcii bude náležať ohodnotenie výkonnosti 1. Naopak, predikcii, ktorá je úplne nesprávna, teda $a(G) = 0$, bude náležať ohodnotenie -1 . Ohodnotenie 0 má náležať predikcii siete, ktorá sa síce dokonale naučila spracovať trénovacie vety, ale nedokáže vôbec zovšeobecňovať.

Označme si p_i celkovú pravdepodobnosť, s akou slovo i nasleduje po súčasnom vstupe v trénovacej množine, bezohľadu na syntaktickú pozíciu súčasného vstupu. Ako $p(G)$ si označme súčet týchto pravdepodobností pre slová, ktoré sú gramaticky správnym pokračovaním doterajšieho vstupu, teda $p(G) = \sum_{i \in G} p_i$.

Sieť, ktorá nedokáže zovšeobecniť poznatky získané učením na trénovacej množine, nebude

schopná pri spracovaní pre ňu novej vstupnej vety rozoznať jej štruktúru. Bude tak nútená založiť predikciu nasledujúceho slova len na aktuálnom vstupe bezohľadu na to, v akej vete a na akej pozícii sa nachádza. (Podobne sa podľa Van der Veldeho a kol. správali nimi testované siete.) Jej predikcia by preto mala zodpovedať pravdepodobnostiam, s akou sa jednotlivé slová vyskytujú po aktuálnom vstupe vo vetách tréningovej množiny bez ohľadu na kontext. Výstupná aktivácia jednotky i v takejto sieti bude preto práve $a_{out,i} = p_i$, a v tom prípade aj $a(G) = p(G)$. Takejto predikcii priradíme ohodnotenie 0, čo bude základná úroveň, nad ktorou budem sieť považovať za preukazujúcu schopnosť zovšeobecňovania.

Pri sieťach, ktoré dokážu zovšeobecňovať, bude potom $a(G) > p(G)$. Naopak, siete, ktoré nebudú preukazovať ani takú úspešnosť predikcií, aká by zodpovedala dokonale nacvičenej sieti s dopredným šírením alebo bigramovému štatistickému modelu, budú mať $a(G) < p(G)$.

Formálny zápis Frankom navrhovanej miery je

$$FGP = \begin{cases} \frac{a(G)-p(G)}{p(G)} & \text{if } a(G) \leq p(G) \\ \frac{a(G)-p(G)}{1-p(G)} & \text{if } a(G) > p(G) \end{cases}$$

Budeme ju nazývať *Frankova zovšeobecňovacia výkonnosť* (Frank's generalization performance; FGP).

V prípade, kedy je možné úplne správne predikovať nasledujúce slovo len na základe súčasného vstupu bez predchádzajúceho kontextu, teda keď $p(G) = 1$, bude aj sieť, ktorá predikuje nasledujúci vstup bezchybne, ohodnotená 0. Takáto situácia v použitej gramatike nastáva len na konci viet, kedy nasledujúci vstup môže byť ľubovoľné podstatné meno bezohľadu na vstupy predošlé. Preto FGP v tejto pozícii nebola vyhodnocovaná.

2.2.4 Simulácie s jednoduchými rekurentnými sieťami

V prvej sade simulácií boli opäť použité jednoduché rekurentné siete zložené zo štyroch vrstiev: vstupnej a výstupnej vrstvy, obe veľkosťou zodpovedajúcej použitému lexikónu $w \in \{18, 26, 34, 42\}$ (každý neurón na vstupe a na výstupe teda reprezentoval jedno slovo z lexikónu), rekurentná vrstva veľkosti $n \in \{10, 20, 40, 70, 100\}$ a skrytá vrstva veľkosti 10 medzi rekurentnou a výstupnou vrstvou. V porovnaní s SRN použitými Van der Veldem a kol. bola vynechaná skrytá vrstva medzi vstupnou a rekurentnou vrstvou (kvôli možnosti porovnať výkonnosť SRN s ESN).

Pre každú kombináciu w a n bolo vytvorených, trénovaných a testovaných 10 sietí (výsledky predstavujú vždy priemerné hodnoty z daných 10 sietí). Váhy spojení boli inicializované náhodne, rovnomerne rozložené v intervale $\pm 0,15$. Upravované boli počas učenia algoritmom spätného

široenia chyby bez momentu. Rýchlosť učenia bola na počiatku 0,1, po každej epoche sa znížila o 0,01. Skryté neuróny používali logistickú aktivačnú funkciu, ako chybová funkcia bola použitá krížová entropia, výstupné neuróny používali softmaxovú aktivačnú funkciu.

Pri $w = 18$, $n = 20$ (simulácia porovnateľná s Van der Velde a kol.) sa siete dokázali naučiť tréningovú množinu takmer dokonale ($FGP \geq 0,997$ pre všetky vetné pozície). Ich výkonnosť značne poklesla pri spracovaní testovacej množiny, aj v najproblematickejších miestach (spracovanie slov na tretej a štvrtej pozícii) sa však držala nad hodnotou 0,5. Z testovacích súvetí bolo pritom 61,7% spracovaných správne, čím je myslené dosiahnutie $FGP > 0$ pri všetkých slovách danej vety. Frank preto dospel k záveru, že aj táto sieť preukázala istú schopnosť slabej systematickosti.

Simulácie pre zvyšné dvojice w a n ukázali, že ako zväčšovanie lexikónu, tak (spočiatku) aj zväčšovanie rekurentnej vrstvy malo pozitívny vplyv na schopnosť sietí zovšeobecňovať – ako FGP, tak podiel správne spracovaných testovacích súvetí narastal (pri $w = 42$, $n = 40$ to bolo 84,2%). Vzorec ich výkonnosti zostával viacmenej rovnaký, najhoršie hodnotenie dosahovali pri spracúvaní tretích a štvrtých slov zo všetkých druhov testovacích viet. Pri veľkostiach rekurentnej vrstvy nad 40 však opätovne nastal výrazný pokles FGP, najmä na najviac problematických vetných pozíciách. Siete pravdepodobne trpeli problémom premodelovania dát. Veľký počet tréningovateľných spojení v sieti im umožnil naučiť sa vety z tréningovej množiny “na spamäť”, bez nutnosti zovšeobecňovať. Frank preto pre ďalšie simulácie použil architektúru s veľkou rekurentnou vrstvou, no menším podielom tréningovateľných spojení – siete s echo stavmi.

2.2.5 Simulácie so sieťami s echo stavmi

Prvé simulácie Frank realizoval so štandardnými ESN so vstupnou vrstvou, dynamickým rezervoárom (rekurentnou vrstvou) a výstupnou vrstvou. Jednotky rezervoáru používali logistickú aktivačnú funkciu, použitý bol lexikón o veľkosti $w = 18$. Absencia skrytej vrstvy medzi rezervoárom a výstupnou vrstvou umožnila použiť off-line tréningovanie popísané v [12]. Napriek pomerne rozsiahlemu prehľadávaniu parametrov siete (veľkosť rezervoáru, interval, z akého boli inicializované váhy, spektrálny polomer váhovej matice dynamického rezervoáru) sa nepodarilo nájsť sieť, ktorá by mala lepšiu výkonnosť ako najlepšia SRN pre daný lexikón.

Štandardná ESN nemá možnosť vývinu vnútorných reprezentácií viet či slov pre konkrétnu úlohu. Tie sú podľa [2] dôležité pre úlohu predikcie nasledujúceho slova, Frank preto medzi dynamický rezervoár a výstupnú vrstvu pridal skrytú vrstvu o veľkosti 10 neurónov. (ESN sa tak topologicky podobala SRN použitým pre predošlé simulácie.) Sieť bola následne učena on-

line, štandardným algoritmom spätného šírenia chyby bez momentu. Výkonnosť najúspešnejšej siete však bola oproti štandardným ESN ešte zhoršená.

Pre ďalšie simulácie Frank v ESN skrytú vrstvu ponechal, logistický dynamický rezervoár však zamenil rezervoárom lineárnym. Aktivácia pre jednotlivé vrstvy v čase t teda bola:

$$\mathbf{a}_{\text{rec}}(t) = \mathbf{W}_{\text{in}}\mathbf{a}_{\text{in}}(t) + \mathbf{W}_{\text{rec}}\mathbf{a}_{\text{rec}}(t-1)$$

$$\mathbf{a}_{\text{hid}}(t) = \mathbf{f}(\mathbf{W}_{\text{hid}}\mathbf{a}_{\text{rec}}(t) + \mathbf{b}_{\text{hid}})$$

$$\mathbf{a}_{\text{out}}(t) = \mathbf{f}_{\text{out}}(\mathbf{W}_{\text{out}}\mathbf{a}_{\text{hid}}(t) + \mathbf{b}_{\text{out}})$$

kde \mathbf{a}_{in} je predkladaný vstupný vektor, \mathbf{a}_{rec} je aktivácia dynamického rezervoáru, \mathbf{a}_{hid} aktivácia skrytej vrstvy a \mathbf{a}_{out} aktivácia výstupnej vrstvy (teda predikcia nasledujúceho vstupu), \mathbf{W}_{in} je matica váh spojení zo vstupných neurónov do rezervoáru, \mathbf{W}_{rec} je matica váh rekurzívnych spojení, \mathbf{W}_{hid} je matica váh spojení z rezervoáru do skrytej vrstvy, \mathbf{W}_{out} je matica váh spojení zo skrytej vrstvy do výstupnej vrstvy, \mathbf{b}_{hid} je bias vektor skrytej vrstvy, \mathbf{b}_{out} je bias vektor výstupnej vrstvy, \mathbf{f} je sigmoidná aktivačná funkcia a \mathbf{f}_{out} je softmaxová aktivačná funkcia.

Matice váh \mathbf{W}_{in} , \mathbf{W}_{hid} a \mathbf{W}_{out} boli inicializované náhodnými hodnotami rovnomerne rozloženými v intervale $\pm 0,15$. Matica váh rekurentných spojení \mathbf{W}_{rec} bola inicializovaná náhodnými hodnotami tak, aby len 15% váh bolo nenulových, a následne upravená na zvolený spektrálny polomer $|\lambda_{\text{max}}| = 0,7$. \mathbf{W}_{in} a \mathbf{W}_{rec} boli okamžite zafixované, \mathbf{W}_{hid} a \mathbf{W}_{out} spolu s príslušnými bias vektormi trénované on-line pomocou algoritmu spätného šírenia chyby bez momentu. Rýchlosť učenia bola napočiatku nastavená na 0,1 a po každej epoche sa znižovala o 0,01. Trénovaných a testovaných bolo vždy 10 sietí pre každú kombináciu veľkostí dynamického rezervoáru $n \in \{10, 20, 40, 70, 100\}$ a lexikónu $w \in \{18, 26, 34, 42\}$.

Výkonnosť ESN s veľkosťou rezervoáru $n = 20$ a lexikónom rozsahu $w = 18$ bola horšia než pri SRN rovnakých parametrov – z testovacích súvetí bolo len 28,1% spracovaných správne. (Napriek tomu bola priemerná výkonnosť pre jednotlivé vetné pozície stále kladná.) Frank však poznamenáva, že kvôli výrazne odlišnému množstvu trénovateľných parametrov nie je celkom korektné porovnávať ESN a SRN s rovnako veľkými rekurentným vrstvami. Ďalšie simulácie ukázali, že výkonnosť sietí so zvyšujúcimi sa n a w neustále rástla, pričom sa navyše zmenšovali rozdiely medzi jednotlivými vetnými pozíciami. Najlepšie výsledky dosahovala najväčšia testovaná sieť s najväčším lexikónom, teda $n = 100$, $w = 42$. Priemerná FGP pre vety z testovacej množiny bola 0,98 pre jednoduché vety, 0,97 pre súvetia s vedľajšou vetou na konci a 0,93 pre súvetia s vedľajšou vetou uprostred. Sieť spracovala správne až 89% testovacích súvetí. Jej

výkonnosť bola vo všetkých sledovaných kritériách vyššia než výkonnosť najlepšej SRN ($n = 40$, $w = 42$).

ESN navyše nevykazovali znaky premodelovania dát ani pri skúšobných simuláciách s veľmi veľkým rezervoárom. FGP siete s $n = 1530$ na testovacích vetách pri $w = 42$ bola 0,98, 0,97, resp. 0,96 pre jednoduché vety, súveta s vedľajšou vetou na konci, resp. súvetia s vedľajšou vetou uprostred. Sieť spracovala správne až 93,3% testovacích súvetí.

Frank sa domnieva, že rozdielny vzťah medzi výkonnosťou siete a veľkosťou jej rekurentnej vrstvy pri SRN a ESN môže spočívať v ich rozdielnych pamäťových schopnostiach. Zväčšovanie rekurentnej vrstvy SRN môžeme chápať ako zväčšovanie jednak “krátkodobej pamäte” (počet rekurentných neurónov), jednak “dlhodobej pamäte” (počet trénovateľných spojení). V krátkodobej pamäti si sieť uchováva doterajšiu sekvenciu vstupov, zatiaľ čo v dlhodobej jednotlivé trénovacie vety. Zatiaľ čo zväčšenie krátkodobej pamäte môže byť vždy len prospešné, zväčšovaním pamäte dlhodobej môže dôjsť k premodelovaniu dát. Pri zväčšovaní rekurentnej vrstvy SRN pritom počet trénovateľných spojení (dlhodobá pamäť) rastie kvadraticky, zatiaľ čo počet rekurentných neurónov (krátkodobá pamäť) len lineárne.

Pri ESN s pridanou skrytou vrstvou je nárast dlhodobej pamäte v porovnaní s SRN oveľa menší, k efektu premodelovania dát tak príde neskôr a pri zväčšovaní rekurentnej vrstvy môžeme dlhšie ťažiť z výhody zväčšovania krátkodobej pamäte. Jaeger v [11] formálne rozpracoval vzťah medzi veľkosťou dynamického rezervoáru ESN a kapacitou jej krátkodobej pamäte. Dokázal, že za optimálnych okolností je kapacita rovná veľkosti dynamického rezervoáru, pričom navyše dokázal, že je zhodná pre lineárne a sigmoidálne rezervoáre. V praxi je však kapacita krátkodobej pamäte ESN väčšinou výrazne nižšia než je veľkosť jej dynamického rezervoáru.

Frank preto okrem záveru, že ako SRN, tak ESN javia známky slabej syntaktickej systematickosti, predpokladá, že ďalšie zväčšovanie rezervoáru a použitého lexikónu môže v prípade ESN viesť k ešte lepším výsledkom. To sme sa rozhodli overiť vo vlastných simuláciách.

3 Simulácie

3.1 Trénovanie úplnou trénovacou množinou

Cieľom prvej série simulácií bolo overiť časť výsledkov z [6], konkrétne simulácie ESN s veľkosťou dynamického rezervoáru $n = 100$ a lexikónmi o veľkostiach $w \in \{18, 26, 34, 42\}$. Snahou bolo do maximálnej možnej miery zopakovať simulácie z citovanej práce. Podrobnejšie informácie o vytváraní trénovacej a testovacej množiny, architektúre, trénovaní a testovaní sietí a hodnotení ich výkonnosti preto čitateľ nájde v častiach 2.1 a 2.2 popisujúcich práce [18] a [6], z ktorých vychádzame.

Pre každý lexikón bola vygenerovaná testovacia množina pozostávajúca zo všetkých gramatických viet obsahujúcich vrámci jednej vety vždy len slová z jednej skupiny. Keďže nejstvovala žiadna ďalšia jednoduchá veta alebo súvetie pozostávajúca výlučne zo slov z jednej skupiny, ktorá by v trénovacej množine nebola, nazvali sme si takéto trénovacie množiny *úplnými*. Počet viet odprezentovaných sieťam počas jednej epochy tréningu vychádzal z veľkosti takejto množiny pre $w = 42$. Pri rozdelení slov do štyroch skupín je počet slov jedného slovného druhu v jednej skupine $g = (w - 2)/8$, počet možných súvetí pozostávajúcich vždy len zo slov z jednej skupiny je teda rovný $8g^5$. Pre $w = 42$ tak máme 25 000 možných trénovacích súvetí. Keďže pomer jednoduchých viet a súvetí má byť počas trénovania 1 : 4, musíme v každej epoche sieť trénovať 6250 jednoduchými vietami. Rovnaký počet trénovacích viet v jednej epoche bol použitý aj pri simuláciách s menšími lexikónmi. Testovacie množiny pozostávali zo všetkých gramatických jednoduchých viet, vrámci ktorých každé slovo bolo vždy z inej skupiny a z 200 náhodne vygenerovaných súvetí so slovami zo všetkých skupín (v každom súvetí teda boli vždy nanejvýš dve slová z rovnakej skupiny). Presné veľkosti trénovacích a testovacích množín sú uvedené v tabuľke 2.

w	počet viet v množinách			
	trénovacej		testovacej	
	jednoduchých	súvetí	jednoduchých	súvetí
18	32	256	192	200
26	108	1944	648	200
34	256	8192	1536	200
42	500	25000	3000	200
82	4000	800000		5000

Tabuľka 2: Počty viet v trénovacích a testovacích množinách.

Pre každý pár trénovacích a testovacích množín bolo vytvorených 10 sietí s echo stavmi s

w	viet za epochu	epôch	viet celkom
18	31250	10	312500
26	31250	10	312500
34	31250	10	312500
42	31250	10	312500
82	1000000	1	1000000

Tabuľka 3: Dĺžka tréovania sietí s $n = 100$ úplnými tréovacími množinami.

dynamickým rezervoárom o veľkosti $n = 100$ a skrytou vrstvou medzi rezervoárom a výstupnými neurónmi o veľkosti 10 neurónov. Vstupy a výstupy boli kódované lokalisticky – každému slovu z lexikónu zodpovedal jeden neurón. Parametre sme použili rovnaké ako Frank v [6]: váhy spojení zo vstupných neurónov do dynamického rezervoáru (\mathbf{W}_{in}), z dynamického rezervoáru do skrytej vrstvy (\mathbf{W}_{hid}) a zo skrytej vrstvy do výstupnej (\mathbf{W}_{out}) boli inicializované náhodne, rovnomerne rozdelné z intervalu $\pm 0,15$; podiel náhodne inicializovaných nenulových hodnôt v matici váh rekurentných spojení (\mathbf{W}_{rec}) bol 15 %, následne bola matica upravená na spektrálny polomer $|\lambda_{max}| = 0,7$. Rezervoár bol lineárny, skryté neuróny používali sigmoidnú aktivačnú funkciu, výstupné neuróny softmaxovú aktivačnú funkciu. Ako učiaci algoritmus pre úpravu váh \mathbf{W}_{hid} a \mathbf{W}_{out} a príslušných bias vektorov bola použitá metóda spätného šírenia chyby bez momentu, rýchlosť učenia bola spočiatku nastavená na 0,1 a po každej epoche sa znižovala o 0,01.

Výsledky simulácií sú v podstate zhodné s výsledkami v [6]. Siete sa takmer dokonale naučili tréovaciu množinu, dobré výsledky dosahovali aj s testovacími dátami. Všetky ukazovatele výkonnosti na testovacích dátach sa so zväčšujúcim lexikónom zvyšovali. Siete dokázali z testovacej množiny správne spracovať priemerne 81,8–91,7 %. Priemerá GPE bola veľmi nízka – v najhoršom prípade ($w = 18$, súvetia s v. v. uprostred) bolo len 4,8 % aktivácie výstupnej vrstvy agramatickej. Priemerná FGP sa pri testovacích vetách pohybovala medzi 0,923 a 0,956. Najproblematickejším miestom bol spravidla prísudok v súvetí s vedľajšou vetou uprostred. Zaujímavým je však pozorovanie, že zatiaľ čo pri testovacích súvetiach výkonnosť sietí so zväčšujúcim sa lexikónom narastá, pri tréovacích súvetiach je trend presne opačný (viď. tabuľky 10 a 12). Kompletne výsledky sú uvedené v tabuľkách 4 až 8.

Množinu lexikónov sme ešte rozšírili o lexikón s veľkosťou $w = 82$. Tréovaciu množinu sme generovali rovnakým spôsobom, kvôli jej veľkosti (800 000 súvetí) sme však učenie skrátili na jednu epochu – počas nej bolo sieťam odprezentovaných viac tréovacích vstupov než počas desiatich epôch pri predchádzajúcich simuláciach. Testovaciu množinu tvorilo 5000 náhodne generovaných súvetí, každé so slovami zo všetkých štyroch skupín. Jednoduché vety sme do

testovacej množiny nezahrnuli. Výsledky simulácií ukázali značný pokles výkonnosti, a to ako pri tréningoch, tak pri testovacích dátach. Podiel správne spracovaných viet (pri ktorých *FGP* neklesla na žiadnom slove pod 0), ako aj samotná výkonnosť pri spracovaní komplexných viet boli z celej skupiny sietí tréningovaných úplnou množinou vôbec najnižšie. Neskoršie simulácie naznačili, že pravdepodobnou príčinou bola nedostatočná veľkosť dynamického rezervoáru, ktorého pamäťová kapacita už nepostačovala pre úspešné spracovanie viet zo slovníka daného rozsahu.

w	Úspešne spracované vety (%) po tréningu úplnou množinou z tréningovej množiny				z testovacej množiny			
	s	r-b	c-e	c	s	r-b	c-e	c
18	99,8	99,7	99,6	99,6	94,3	83,7	79,8	81,8
26	99,7	99,4	99,3	99,4	97,1	88,8	80,7	84,8
34	99,5	99,1	99,1	99,1	97,8	91,7	86,7	89,2
42	99,5	98,6	98,5	98,6	98,2	93,7	89,7	91,7
82	99,7	98,4	97,4	97,9		80,4	69,4	74,9

Tabuľka 4: Podiel správne spracovaných viet sieťami $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréningovanými úplnou tréningovou množinou. **s** označuje jednoduché vety, **r-b** súvetia s vedľajšou vetou na konci, **c-e** súvetia s vedľajšou vetou uprostred a **c** celkový priemer pre súvetia.

\overline{GPE} – jednoduché vety – tréovanie úplnou množinou								
w	trénovacia množina				testovacia množina			
	N	V	N	priemer	N	V	N	priemer
18	0,002	0,002	0,003	0,002 ($\pm 0,001$)	0,001	0,008	0,047	0,018 ($\pm 0,027$)
26	0,003	0,002	0,004	0,003 ($\pm 0,001$)	0,001	0,009	0,020	0,010 ($\pm 0,011$)
34	0,005	0,003	0,006	0,005 ($\pm 0,002$)	0,001	0,006	0,019	0,009 ($\pm 0,010$)
42	0,005	0,003	0,006	0,005 ($\pm 0,002$)	0,001	0,006	0,012	0,006 ($\pm 0,006$)
82	0,005	0,002	0,003	0,004 ($\pm 0,003$)				

Tabuľka 5: Priemerné hodnoty GPE pri spracovaní jednoduchých viet sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou tréovacou množinou.

\overline{FGP} – jednoduché vety – tréovanie úplnou množinou								
w	trénovacia množina				testovacia množina			
	N	V	N	priemer	N	V	N	priemer
18	0,994	0,993	0,992	0,993 ($\pm 0,001$)	0,998	0,968	0,890	0,952 ($\pm 0,060$)
26	0,992	0,992	0,989	0,991 ($\pm 0,002$)	0,998	0,970	0,946	0,972 ($\pm 0,031$)
34	0,987	0,989	0,983	0,986 ($\pm 0,003$)	0,996	0,982	0,952	0,977 ($\pm 0,025$)
42	0,987	0,988	0,983	0,986 ($\pm 0,003$)	0,997	0,979	0,969	0,982 ($\pm 0,017$)
82	0,985	0,992	0,991	0,989 ($\pm 0,005$)				

Tabuľka 6: Priemerné hodnoty FGP pri spracovaní jednoduchých viet sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou tréovacou množinou.

\overline{FGP} – všetky typy viet z tréovacej množiny				
w	súvetia			
	jednoduché	s v. v. na konci	s v. v. uprostred	priemer
18	0,993 ($\pm 0,001$)	0,991 ($\pm 0,005$)	0,990 ($\pm 0,006$)	0,991 ($\pm 0,010$)
26	0,991 ($\pm 0,002$)	0,987 ($\pm 0,013$)	0,985 ($\pm 0,012$)	0,987 ($\pm 0,021$)
34	0,986 ($\pm 0,003$)	0,982 ($\pm 0,018$)	0,980 ($\pm 0,012$)	0,982 ($\pm 0,025$)
42	0,986 ($\pm 0,003$)	0,977 ($\pm 0,030$)	0,971 ($\pm 0,023$)	0,976 ($\pm 0,042$)
82	0,989 ($\pm 0,005$)	0,983 ($\pm 0,022$)	0,969 ($\pm 0,028$)	0,979 ($\pm 0,042$)

Tabuľka 7: Priemerné hodnoty FGP pri spracovaní všetkých typov viet z tréovacej množiny sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou tréovacou množinou.

\overline{FGP} – všetky typy viet z testovacej množiny				
w	súvetia			
	jednoduché	s v. v. na konci	s v. v. uprostred	priemer
18	0,952 ($\pm 0,060$)	0,925 ($\pm 0,153$)	0,907 ($\pm 0,229$)	0,923 ($\pm 0,291$)
26	0,972 ($\pm 0,031$)	0,939 ($\pm 0,124$)	0,916 ($\pm 0,268$)	0,936 ($\pm 0,310$)
34	0,977 ($\pm 0,025$)	0,956 ($\pm 0,090$)	0,927 ($\pm 0,163$)	0,948 ($\pm 0,202$)
42	0,982 ($\pm 0,017$)	0,961 ($\pm 0,062$)	0,938 ($\pm 0,138$)	0,956 ($\pm 0,164$)
82		0,914 ($\pm 0,083$)	0,856 ($\pm 0,307$)	0,885 ($\pm 0,332$)

Tabuľka 8: Priemerné hodnoty FGP pri spracovaní všetkých typov viet z testovacej množiny sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou tréovacou množinou.

\overline{GPE} – súvetia s vedľajšou vetou na konci (right-branching complex sentences) – tréovanie úplnou množinou														
	trénovacia množina							testovacia množina						
w	N	V	N	who	V	N	priemer	N	V	N	who	V	N	priemer
18	0,003	0,002	0,003	0,007	0,003	0,003	0,004 ($\pm 0,007$)	0,001	0,006	0,045	0,094	0,010	0,057	0,037 ($\pm 0,142$)
26	0,003	0,002	0,004	0,010	0,005	0,005	0,006 ($\pm 0,011$)	0,001	0,008	0,020	0,079	0,030	0,020	0,029 ($\pm 0,142$)
34	0,005	0,003	0,007	0,014	0,008	0,006	0,008 ($\pm 0,013$)	0,001	0,005	0,023	0,059	0,011	0,019	0,025 ($\pm 0,110$)
42	0,005	0,003	0,007	0,017	0,011	0,010	0,010 ($\pm 0,020$)	0,001	0,006	0,013	0,042	0,014	0,026	0,021 ($\pm 0,087$)
82	0,005	0,002	0,003	0,017	0,005	0,009	0,007 ($\pm 0,012$)	0,006	0,045	0,045	0,051	0,039	0,043	0,038 ($\pm 0,036$)

Tabuľka 9: Priemerné hodnoty GPE pri spracovaní súvetí s vedľajšou vetou na konci sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou trénovacou množinou.

17

\overline{FGP} – súvetia s vedľajšou vetou na konci (right-branching complex sentences) – tréovanie úplnou množinou														
	trénovacia množina							testovacia množina						
w	N	V	N	who	V	N	priemer	N	V	N	who	V	N	priemer
18	0,993	0,992	0,991	0,987	0,990	0,994	0,991 ($\pm 0,005$)	0,998	0,976	0,894	0,812	0,967	0,901	0,925 ($\pm 0,153$)
26	0,992	0,991	0,988	0,980	0,978	0,992	0,987 ($\pm 0,013$)	0,998	0,972	0,947	0,843	0,904	0,968	0,939 ($\pm 0,124$)
34	0,987	0,988	0,982	0,972	0,969	0,990	0,982 ($\pm 0,018$)	0,997	0,983	0,944	0,881	0,959	0,969	0,956 ($\pm 0,090$)
42	0,987	0,987	0,982	0,966	0,955	0,984	0,977 ($\pm 0,030$)	0,997	0,978	0,968	0,915	0,951	0,957	0,961 ($\pm 0,062$)
82	0,985	0,992	0,991	0,966	0,978	0,986	0,983 ($\pm 0,022$)	0,984	0,884	0,899	0,898	0,892	0,926	0,914 ($\pm 0,083$)

Tabuľka 10: Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou na konci sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou trénovacou množinou.

\overline{GPE} – súvetia s vedľajšou vetou uprostred (centre-embedded complex sentences) – tréovanie úplnou množinou														
w	trénovacia množina							testovacia množina						
	N	who	N	V	V	N	priemer	N	who	N	V	V	N	priemer
18	0,003	0,004	0,007	0,007	0,004	0,006	0,005 ($\pm 0,004$)	0,001	0,001	0,004	0,101	0,094	0,088	0,048 ($\pm 0,107$)
26	0,003	0,008	0,009	0,011	0,007	0,008	0,008 ($\pm 0,006$)	0,001	0,002	0,007	0,101	0,117	0,023	0,042 ($\pm 0,117$)
34	0,005	0,011	0,012	0,014	0,008	0,013	0,010 ($\pm 0,007$)	0,001	0,005	0,018	0,103	0,065	0,044	0,039 ($\pm 0,088$)
42	0,005	0,014	0,017	0,019	0,012	0,023	0,015 ($\pm 0,012$)	0,001	0,005	0,013	0,084	0,052	0,041	0,033 ($\pm 0,072$)
82	0,005	0,006	0,021	0,023	0,012	0,034	0,017 ($\pm 0,019$)	0,006	0,007	0,039	0,197	0,122	0,091	0,077 ($\pm 0,167$)

Tabuľka 11: Priemerné hodnoty GPE pri spracovaní súvetí s vedľajšou vetou uprostred sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou trénovacou množinou.

18

\overline{FGP} – súvetia s vedľajšou vetou uprostred (centre-embedded complex sentences) – tréovanie úplnou množinou														
w	trénovacia množina							testovacia množina						
	N	who	N	V	V	N	priemer	N	who	N	V	V	N	priemer
18	0,993	0,991	0,989	0,991	0,985	0,991	0,990 ($\pm 0,006$)	0,998	0,997	0,993	0,862	0,745	0,849	0,907 ($\pm 0,229$)
26	0,992	0,984	0,987	0,986	0,975	0,987	0,985 ($\pm 0,012$)	0,998	0,995	0,989	0,856	0,694	0,963	0,916 ($\pm 0,268$)
34	0,987	0,978	0,981	0,982	0,970	0,980	0,980 ($\pm 0,012$)	0,997	0,990	0,970	0,853	0,825	0,927	0,927 ($\pm 0,163$)
42	0,987	0,973	0,973	0,975	0,956	0,965	0,971 ($\pm 0,023$)	0,997	0,990	0,980	0,881	0,850	0,933	0,938 ($\pm 0,138$)
82	0,985	0,987	0,967	0,970	0,958	0,946	0,969 ($\pm 0,028$)	0,984	0,987	0,936	0,676	0,708	0,844	0,856 ($\pm 0,307$)

Tabuľka 12: Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou uprostred sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ tréovanými úplnou trénovacou množinou.

3.2 Trénovanie neúplnou trénovacou množinou

Pokračovať v konštrukcii trénovacej množiny pre rozsiahlejšie lexikóny rovnakým spôsobom by bolo neúnosné, pre ďalšie simulácie sme preto zvolili iný postup. Trénovacu množinu viet sme konštruovali tak, aby sa každé slovo z lexikónu vyskytovalo na každej vetnej pozícii 100krát v jednoduchých vetách a 400krát v súvetiach. Takýmto spôsobom dostaneme pre lexikón o veľkosti w trénovaciu množinu s $250w - 500$ vetami. Testovaciu množinu tvorilo náhodne vybraných 500 jednoduchých viet, v ktorých bolo každé slovo z inej skupiny, a 2000 súvetí (po 1000 z každého druhu), každé so slovami zo všetkých štyroch skupín (v každom súvetí boli teda nanajvýš dve slová z jednej skupiny). Veľkosti množín pre jednotlivé lexikóny uvádza tabuľka 13.

w	počet viet v množinách			
	trénovacej		testovacej	
	jednoduchých	súvetí	jednoduchých	súvetí
34	1600	6400	500	2000
42	2000	8000	500	2000
82	4000	16000	500	2000
162	8000	32000	500	2000
242	12000	48000	500	2000

Tabuľka 13: Počty viet v trénovacích a testovacích množinách pre siete trénované neúplnými trénovacími množinami.

Pre siete s dynamickým rezervoárom veľkosti $n = 100$ sme robili simulácie nad lexikónmi veľkostí $w \in \{34, 42, 82, 162, 242\}$. Siete boli učené po dobu desať epôch, dĺžka jednej epochy bola vždy rovná veľkosti trénovacej množiny pre konkrétny lexikón, viď. tabuľka 14.

w	n	viet za epochu	epôch	viet celkom
34	100	8000	10	80000
42	100	10000	10	100000
82	100	20000	10	200000
162	100	40000	10	400000
242	100	60000	10	600000
162	200	40000	10	400000
242	200	60000	10	600000
162	500	40000	3	120000
242	500	60000	2	120000

Tabuľka 14: Dĺžka tréovania sietí s $n \in \{100, 200, 500\}$ neúplnými trénovacími množinami.

Pri porovnaní výsledkov pre $w = 32$ až $w = 82$ s predošlými simuláciami, v ktorých boli ESN učené úplnou trénovacou množinou, vidíme vo väčšine ukazovateľov mierny výkonnostný

pokles, hoci sa nájdu aj ojedinelé výnimky (napríklad podiel správne spracovaných testovacích súvetí pri $w = 32$ je 91 % oproti 89,2 %, a pri $w = 82$ je 75,5 % oproti 74,9 %, či spracovanie niektorých vetných pozícií v testovacích súvetiach pre $w = 32$). Zaujímavé sú však predovšetkým dve zistenia.

Po prvé, zatiaľ čo pri predošlých simuláciách, počas ktorých boli siete učené úplnou trénovacou množinou, bola schopnosť ESN zovšeobecňovať pri spracúvaní testovacích viet vyššia pri použití lexikónu s veľkosťou $w = 42$ než pri $w = 34$, v tomto prípade je to naopak. Možným vysvetlením by bola akási vyššia “pamäťová náročnosť” zovšeobecňovania na základe neúplnej trénovacej množiny v porovnaní s tým istým procesom nad množinou úplnou. Potenciálny prínos väčšieho lexikónu pre generalizáciu môže byť znegovaný nemožnosťou nadobudnuté poznatky úspešne uložiť.

Po druhé, siete mali výrazne horšiu výkonnosť pri spracovaní trénovacej množiny v porovnaní s predošlými simuláciami. Napríklad v prípade $w = 82$ je výkonnosť siete na trénovacích a testovacích vetách veľmi podobná, v oboch prípadoch slabšia oproti výsledkom s $w = 34$ alebo $w = 42$. V predošlých simuláciách s úplnou trénovacou množinou pritom sieť $w = 82$ síce podávala nad testovacou množinou slabšie výsledky oproti simuláciám s $w = 34$ alebo $w = 42$, pri trénovacích dátach však výraznejšie nezaostávala. V iných prípadoch bola výkonnosť nad testovacou množinou dokonca vyššia než nad trénovacou.

Samo sa núka vysvetlenie, že za tento výkonnostný prepád nad trénovacou množinou môže kratšie učenie sietí. Čiastkové hodnotenia výkonnosti zbierané už počas trénovania sietí však ukazujú, že ESN učené úplnou trénovacou množinou dosahovali nad ňou vyššie hodnoty FGP už v čase, keď bol počet dosiaľ spracovaných trénovacích viet porovnateľný. (Na druhú stranu je však taktiež pravdou, že učiaci algoritmus na daných sieťach v tom čase pracoval ešte s vyššou rýchlosťou učenia.) Opäť akoby učenie sa z neúplnej trénovacej množiny bolo pre siete “pamäťovo náročnejšie”. Pamäťová kapacita dynamického rezervoáru na hranici zvládnuteľnosti danej úlohy im neumožní si o trénovacej množine uložiť podrobnejší obraz a núti ich založiť predikciu nasledujúceho vstupu podobne ako pri testovacích vetách na generalizácii.

V simuláciách sme pokračovali aj pre rozsiahlejšie slovníky. Pri $n = 100$ a $w \in \{162, 242\}$ sme zaznamenali ďalší výkonnostný prepád, s rovnakými znakmi ako v predošlom prípade: ešte výraznejšie sa prehĺbili rozdiely v FGP pri spracovaní vetných pozícií s rôznou náročnosťou predikcie, taktiež v niektorých prípadoch bola výkonnosť nad trénovacou množinou horšia než nad testovacou. Pre danú veľkosť dynamického rezervoára bolo spracovanie podobne rozsiahlych lexikónov už príliš náročné. Priemerná FGP však bola stále kladná – siete teda aj naďalej javili

znaky slabšej systematickosti.

So zväčšením rezervoára na $n = 200$ prišlo aj k očakávanému zlepšeniu výkonov sietí. V prípade $w = 162$ boli siete vo väčšine ukazovateľov lepšie než ostatné ESN trénované neúplnou trénovacou množinou a v mnohom prekonal aj siete trénované úplnou trénovacou množinou pri lexikónoch menších než 42. Pozoruhodné bolo, že napriek tomu v niektorých situáciách – napríklad pri spracovaní súvetí – ešte stále dosahovali nižšiu FGP nad trénovacími vetami v porovnaní so spracovaním viet testovacích.

Ďalším zaujímavým zistením je, že sieť s $n = 200$, $w = 162$ dosahovala oveľa lepšie hodnotenia ako sieť s $n = 100$, $w = 82$. Pomer medzi veľkosťami dynamického rezervoára a lexikónu n/w je v oboch prípadoch pritom takmer rovnaký. Navyše, aj sieť s $n = 200$, $w = 242$ dosahovala zväčša lepšie hodnotenia než sieť $n = 100$, $w = 82$. Nemusí to nutne znamenať zneplatnenie našej domnienky, že slabá výkonnosť siete s $n = 100$, $w = 82$ bola spôsobená nízkou pamäťovou kapacitou rezervoára. Môže to ale naznačovať, že “pamäťové nároky” konkrétnej úlohy nie sú závislé len od veľkosti lexikónu.

Návrat späť k situácii, kedy siete zvládali trénovacie údaje lepšie než testovacie, nastal až s ďalším zväčšením dynamického rezervoára, na $n = 500$. Dĺžku učenia sme pri oboch testovaných veľkostiach lexikónu $w \in \{162, 242\}$ skrátili na celkom 120 000 viet. (Pri $w = 162$ to teda bol pokles o 70 %, pri $w = 242$ o 80 %.) Napriek tomu – alebo azda aj preto – boli výsledné siete v predikcii nasledujúceho slova vôbec najúspešnejšie zo všetkých testovaných.

Pri $n = 500$, $w = 162$ bol podiel správne spracovaných trénovacích viet 100 %, v prípade testovacích súvetí to bolo 99,6 %. Priemerná FGP pre testovacie vety bola 0,988 pre jednoduché vety, 0,985 pre súvetia s v. v. na konci a 0,986 pre súvetia s v. v. uprostred. Výkonnosť dosahovaná na problematických syntaktických pozíciách sa od výkonnosti dosahovanej na pozíciách jednoduchších líšila len veľmi mierne – v aj v najhoršom prípade (piate slovo súvetia s v. v. uprostred) bola dosiahnutá FGP v priemere 0,949. GPE sa pohybovala priemerne v rozsahu 0 až 0,01. Najvyššia bola v štvrtom slove súvetí s v. v. uprostred, a aj to len 0,021.

Pri $n = 500$, $w = 242$ boli hodnoty veľmi mierne horšie. V svetle predchádzajúcich výsledkov to nie je vôbec prekvapivé. Je veľmi pravdepodobné, že ďalšie zväčšenie dynamického rezervoára by prinieslo zlepšenie výsledkov aj pre tento lexikón. Stále sa však jednalo o druhú najúspešnejšiu sieť z celej testovacej množiny.

Podrobné výsledky simulácií sú uvedené v tabuľkách 15 až 18. Grafy 1 až 8 znázorňujú výkonnosť vybraných sietí vo vzťahu k syntaktickej pozícii aktuálneho vstupu, graf 9 priemernú výkonnosť sietí vo vzťahu k spôsobu ich trénovania a veľkosti dynamického rezervoára.

Úspešne spracované vety (%) po trénovaní neúplnou množinou									
		z trénovacej množiny				z testovacej množiny			
w	n	s	r-b	c-e	c	s	r-b	c-e	c
34	100	98,1	96,5	96,2	96,3	99,4	93,7	88,2	91,0
42	100	97,7	95,2	95,1	95,1	96,2	90,0	84,1	87,1
82	100	95,2	85,6	85,0	85,3	91,0	77,9	73,1	75,5
162	100	86,1	63,8	63,9	63,8	82,3	65,1	61,7	63,4
242	100	80,0	51,1	48,0	49,6	79,2	54,3	51,2	52,7
162	200	98,0	93,5	93,8	93,6	97,2	92,7	89,8	91,3
242	200	96,3	84,3	86,1	85,2	95,8	83,9	84,5	84,2
162	500	100,0	100,0	100,0	100,0	99,9	99,8	99,5	99,6
242	500	100,0	99,9	99,8	99,9	99,8	99,5	98,8	99,2

Tabuľka 15: Podiel správne spracovaných viet sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými čiastočnou trénovacou množinou. **s** označuje jednoduché vety, **r-b** súvetia s vedľajšou vetou na konci, **c-e** súvetia s vedľajšou vetou uprostred a **c** celkový priemer pre súvetia.

\overline{FGP} – jednoduché vety – trénovanie neúplnou množinou										
		trénovacia množina				testovacia množina				
w	n	N	V	N	priemer	N	V	N	priemer	
34	100	0,969	0,955	0,950	0,958 ($\pm 0,012$)	0,996	0,984	0,975	0,985 ($\pm 0,012$)	
42	100	0,965	0,952	0,944	0,954 ($\pm 0,013$)	0,990	0,957	0,932	0,960 ($\pm 0,034$)	
82	100	0,932	0,911	0,909	0,917 ($\pm 0,017$)	0,965	0,865	0,901	0,910 ($\pm 0,071$)	
162	100	0,884	0,825	0,756	0,822 ($\pm 0,073$)	0,906	0,784	0,752	0,814 ($\pm 0,103$)	
242	100	0,841	0,731	0,711	0,761 ($\pm 0,091$)	0,856	0,749	0,710	0,772 ($\pm 0,094$)	
162	200	0,964	0,962	0,963	0,963 ($\pm 0,001$)	0,986	0,950	0,974	0,970 ($\pm 0,026$)	
242	200	0,942	0,942	0,936	0,940 ($\pm 0,004$)	0,959	0,927	0,952	0,946 ($\pm 0,023$)	
162	500	0,994	0,989	0,997	0,993 ($\pm 0,005$)	0,993	0,976	0,996	0,988 ($\pm 0,014$)	
242	500	0,994	0,988	0,997	0,993 ($\pm 0,006$)	0,993	0,975	0,996	0,988 ($\pm 0,015$)	

Tabuľka 16: Priemerné hodnoty FGP pri spracovaní jednoduchých viet sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.

\overline{FGP} – súvetia s vedľajšou vetou na konci (right-branching complex sentences) – tréovanie neúplnou množinou															
		trénovacia množina							testovacia množina						
w	n	N	V	N	who	V	N	priemer	N	V	N	who	V	N	priemer
34	100	0,968	0,955	0,949	0,912	0,903	0,957	0,940 ($\pm 0,057$)	0,995	0,985	0,975	0,900	0,923	0,940	0,953 ($\pm 0,084$)
42	100	0,966	0,954	0,945	0,911	0,871	0,958	0,934 ($\pm 0,079$)	0,990	0,957	0,932	0,887	0,896	0,959	0,937 ($\pm 0,087$)
82	100	0,933	0,912	0,911	0,848	0,761	0,909	0,879 ($\pm 0,142$)	0,965	0,866	0,898	0,864	0,795	0,882	0,878 ($\pm 0,123$)
162	100	0,884	0,824	0,755	0,738	0,540	0,760	0,750 ($\pm 0,260$)	0,905	0,793	0,772	0,795	0,647	0,773	0,781 ($\pm 0,184$)
242	100	0,841	0,731	0,710	0,684	0,399	0,619	0,664 ($\pm 0,330$)	0,864	0,754	0,713	0,715	0,505	0,644	0,699 ($\pm 0,262$)
162	200	0,964	0,962	0,963	0,941	0,859	0,951	0,940 ($\pm 0,090$)	0,986	0,949	0,973	0,961	0,898	0,957	0,954 ($\pm 0,067$)
242	200	0,943	0,943	0,936	0,904	0,725	0,907	0,893 ($\pm 0,187$)	0,957	0,927	0,939	0,929	0,776	0,922	0,908 ($\pm 0,147$)
162	500	0,994	0,989	0,997	0,990	0,977	0,999	0,991 ($\pm 0,016$)	0,994	0,975	0,996	0,981	0,968	0,997	0,985 ($\pm 0,025$)
242	500	0,994	0,988	0,997	0,984	0,970	0,996	0,988 ($\pm 0,022$)	0,993	0,976	0,997	0,975	0,962	0,993	0,983 ($\pm 0,030$)

Tabuľka 17: Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou na konci sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ tréovanými neúplnou trénovacou množinou.

\overline{FGP} – súvetia s vedľajšou vetou uprostred (centre-embedded complex sentences) – tréovanie neúplnou množinou															
		trénovacia množina							testovacia množina						
w	n	N	V	N	who	V	N	priemer	N	V	N	who	V	N	priemer
34	100	0,968	0,935	0,931	0,938	0,897	0,934	0,934 ($\pm 0,050$)	0,995	0,979	0,968	0,850	0,818	0,913	0,920 ($\pm 0,164$)
42	100	0,966	0,931	0,919	0,930	0,880	0,929	0,926 ($\pm 0,062$)	0,990	0,980	0,965	0,800	0,769	0,923	0,904 ($\pm 0,215$)
82	100	0,933	0,869	0,856	0,857	0,761	0,846	0,854 ($\pm 0,123$)	0,965	0,932	0,904	0,712	0,701	0,807	0,837 ($\pm 0,253$)
162	100	0,884	0,796	0,651	0,701	0,569	0,650	0,709 ($\pm 0,248$)	0,905	0,825	0,705	0,636	0,608	0,640	0,720 ($\pm 0,258$)
242	100	0,841	0,757	0,558	0,573	0,393	0,486	0,601 ($\pm 0,360$)	0,864	0,792	0,623	0,579	0,451	0,513	0,637 ($\pm 0,341$)
162	200	0,964	0,953	0,929	0,932	0,890	0,934	0,934 ($\pm 0,056$)	0,986	0,978	0,956	0,872	0,880	0,940	0,935 ($\pm 0,109$)
242	200	0,943	0,921	0,847	0,886	0,825	0,884	0,884 ($\pm 0,098$)	0,957	0,941	0,876	0,864	0,823	0,912	0,895 ($\pm 0,112$)
162	500	0,994	0,991	0,991	0,993	0,981	0,999	0,991 ($\pm 0,011$)	0,994	0,991	0,990	0,972	0,967	0,997	0,985 ($\pm 0,026$)
242	500	0,994	0,986	0,986	0,986	0,971	0,996	0,986 ($\pm 0,017$)	0,993	0,986	0,983	0,958	0,949	0,994	0,977 ($\pm 0,040$)

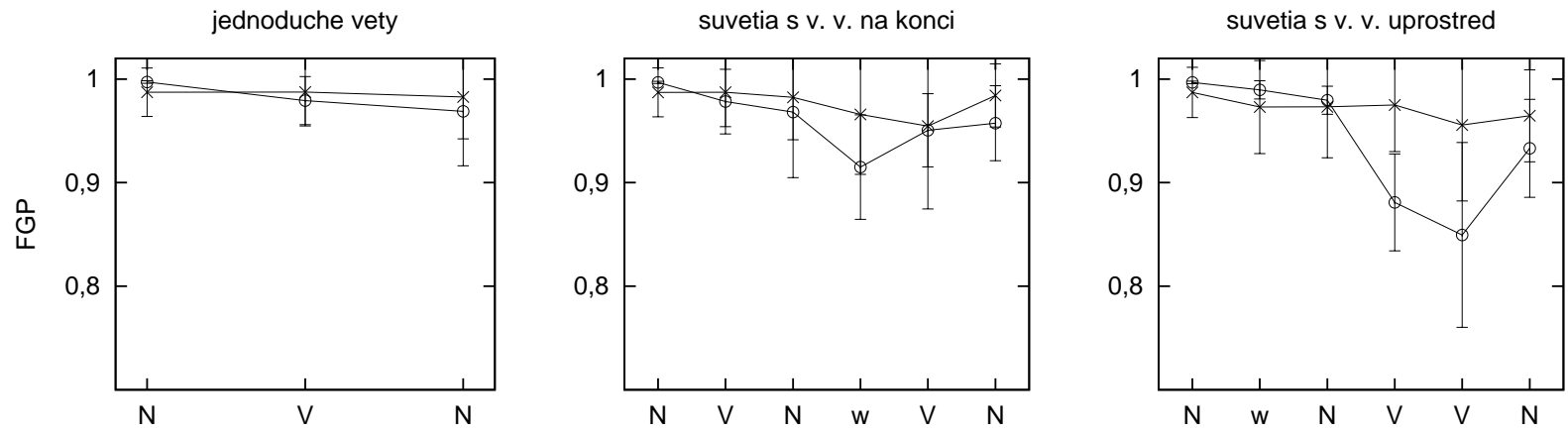
Tabuľka 18: Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou uprostred sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ tréovanými neúplnou trénovacou množinou.

\overline{FGP} – všetky typy viet z trénovacej množiny					
súvetia					
w	n	jednoduché	s v. v. na konci	s v. v. uprostred	priemer
34	100	0,958 ($\pm 0,012$)	0,940 ($\pm 0,057$)	0,934 ($\pm 0,050$)	0,941 ($\pm 0,086$)
42	100	0,954 ($\pm 0,013$)	0,934 ($\pm 0,079$)	0,926 ($\pm 0,062$)	0,935 ($\pm 0,111$)
82	100	0,917 ($\pm 0,017$)	0,879 ($\pm 0,142$)	0,854 ($\pm 0,123$)	0,877 ($\pm 0,209$)
162	100	0,822 ($\pm 0,073$)	0,750 ($\pm 0,260$)	0,709 ($\pm 0,248$)	0,748 ($\pm 0,396$)
242	100	0,761 ($\pm 0,091$)	0,664 ($\pm 0,330$)	0,601 ($\pm 0,360$)	0,658 ($\pm 0,534$)
162	200	0,963 ($\pm 0,001$)	0,940 ($\pm 0,090$)	0,934 ($\pm 0,056$)	0,942 ($\pm 0,115$)
242	200	0,940 ($\pm 0,004$)	0,893 ($\pm 0,187$)	0,884 ($\pm 0,098$)	0,899 ($\pm 0,227$)
162	500	0,993 ($\pm 0,005$)	0,991 ($\pm 0,016$)	0,991 ($\pm 0,011$)	0,992 ($\pm 0,022$)
242	500	0,993 ($\pm 0,006$)	0,988 ($\pm 0,022$)	0,986 ($\pm 0,017$)	0,988 ($\pm 0,032$)

Tabuľka 19: Priemerné hodnoty FGP pri spracovaní všetkých typov viet z trénovacej množiny sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.

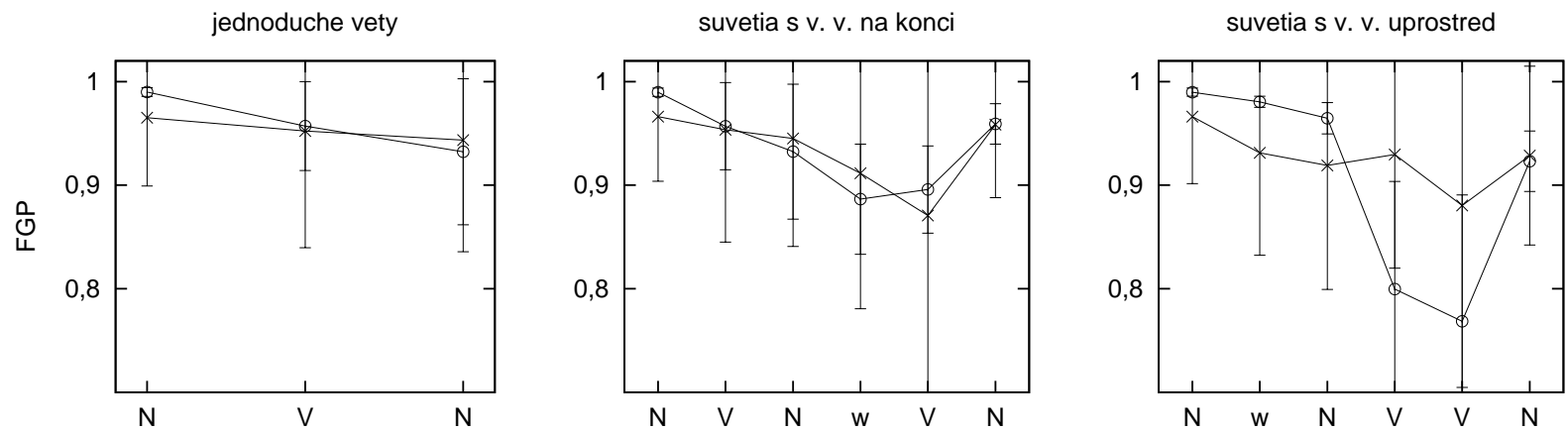
\overline{FGP} – všetky typy viet z testovacej množiny					
súvetia					
w	n	jednoduché	s v. v. na konci	s v. v. uprostred	priemer
34	100	0,985 ($\pm 0,012$)	0,953 ($\pm 0,084$)	0,920 ($\pm 0,164$)	0,946 ($\pm 0,205$)
42	100	0,960 ($\pm 0,034$)	0,937 ($\pm 0,087$)	0,904 ($\pm 0,215$)	0,928 ($\pm 0,251$)
82	100	0,910 ($\pm 0,071$)	0,878 ($\pm 0,123$)	0,837 ($\pm 0,253$)	0,868 ($\pm 0,306$)
162	100	0,814 ($\pm 0,103$)	0,781 ($\pm 0,184$)	0,720 ($\pm 0,258$)	0,763 ($\pm 0,354$)
242	100	0,772 ($\pm 0,094$)	0,699 ($\pm 0,262$)	0,637 ($\pm 0,341$)	0,689 ($\pm 0,469$)
162	200	0,970 ($\pm 0,026$)	0,954 ($\pm 0,067$)	0,935 ($\pm 0,109$)	0,950 ($\pm 0,140$)
242	200	0,946 ($\pm 0,023$)	0,908 ($\pm 0,147$)	0,895 ($\pm 0,112$)	0,911 ($\pm 0,201$)
162	500	0,988 ($\pm 0,014$)	0,985 ($\pm 0,025$)	0,985 ($\pm 0,026$)	0,986 ($\pm 0,041$)
242	500	0,988 ($\pm 0,015$)	0,983 ($\pm 0,030$)	0,977 ($\pm 0,040$)	0,982 ($\pm 0,056$)

Tabuľka 20: Priemerné hodnoty FGP pri spracovaní všetkých typov viet z testovacej množiny sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.

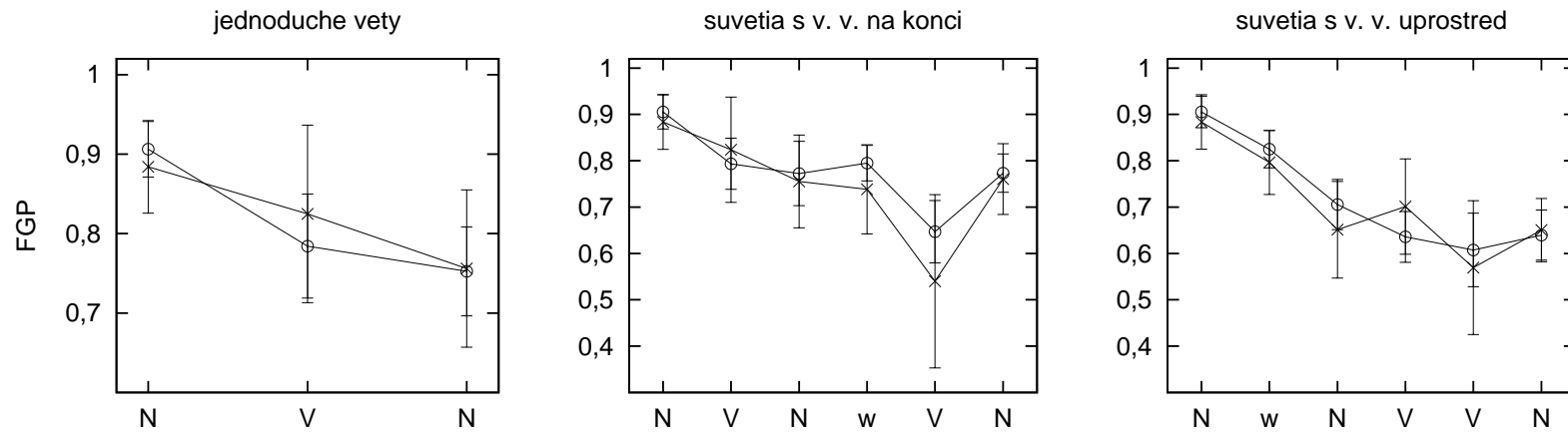


Obrázok 1: Priemerné hodnoty FGP siete s $n = 100$ trénovanej úplnou množinou pri $w = 42$. (x – trénovacie dáta, o – testovacie dáta)

25

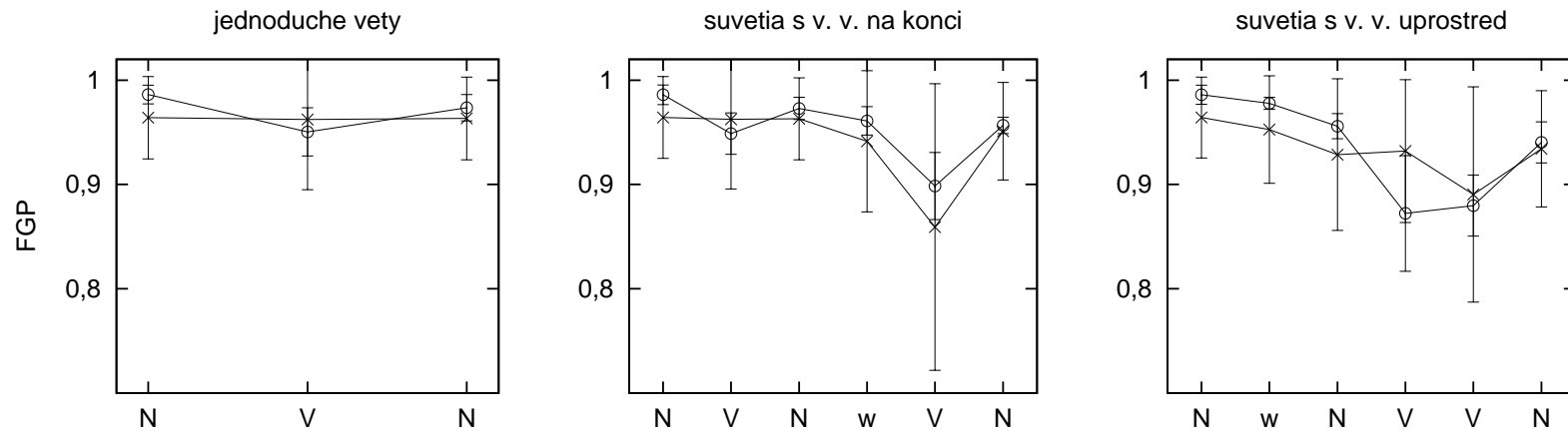


Obrázok 2: Priemerné hodnoty FGP siete s $n = 100$ trénovanej neúplnou množinou pri $w = 42$. (x – trénovacie dáta, o – testovacie dáta)

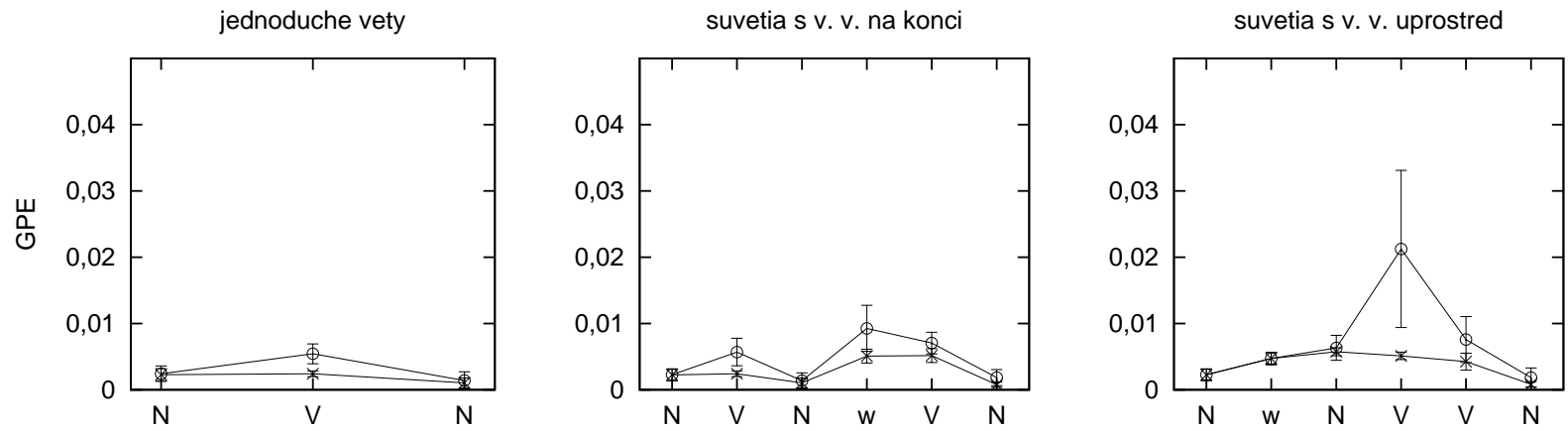


Obrázok 3: Priemerné hodnoty FGP siete s $n = 100$ trénovanej neúplnou množinou pri $w = 162$. (x – trénovacie dáta, o – testovacie dáta)

26

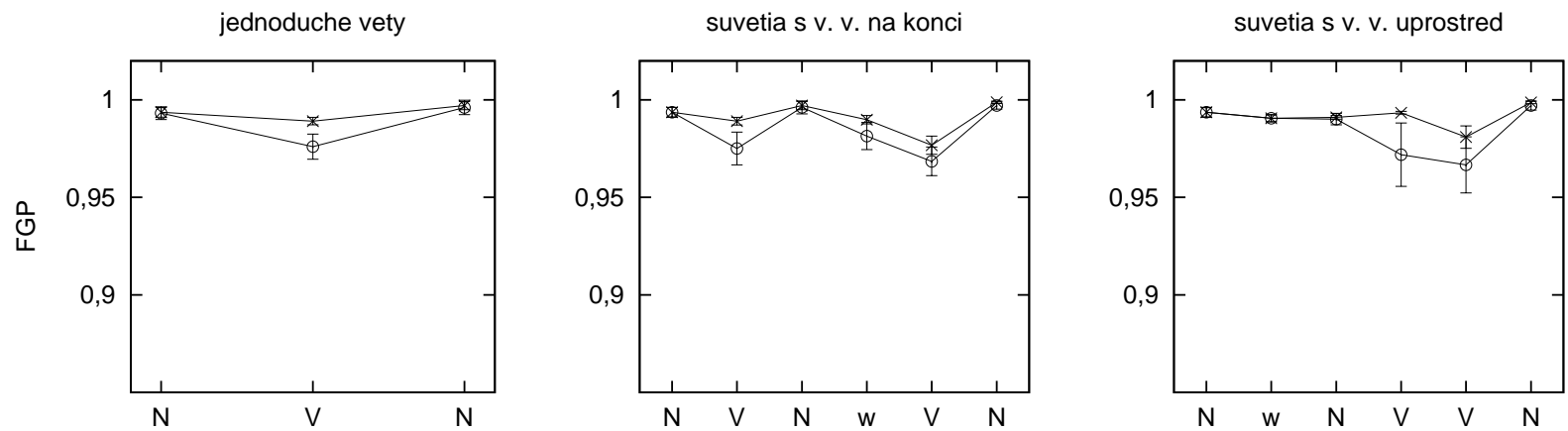


Obrázok 4: Priemerné hodnoty FGP siete s $n = 200$ trénovanej neúplnou množinou pri $w = 162$. (x – trénovacie dáta, o – testovacie dáta)

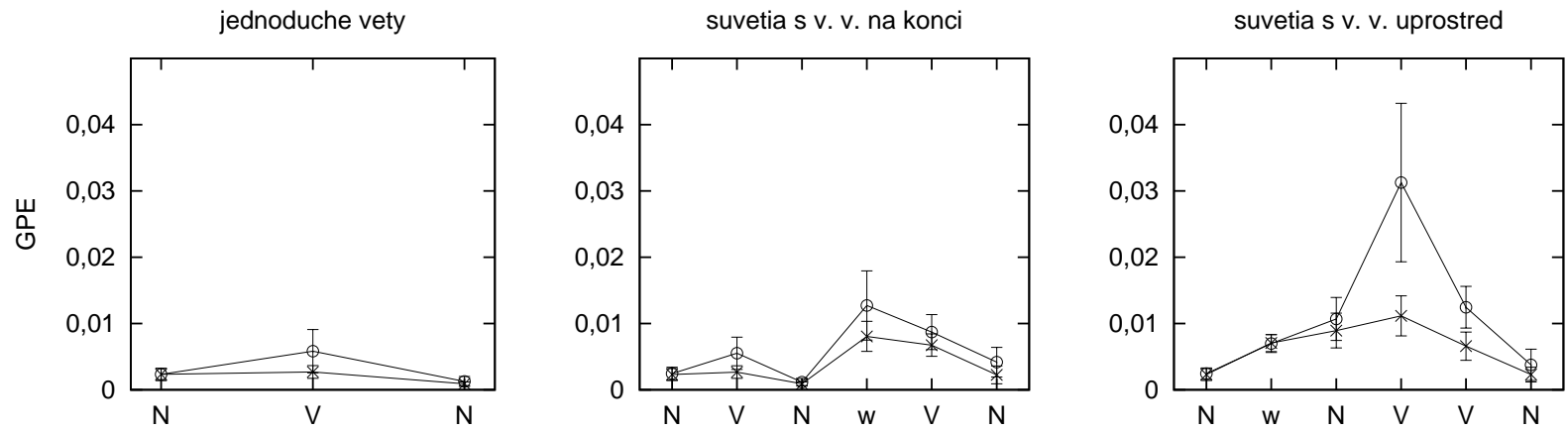


Obrázok 5: Priemerné hodnoty GPE siete s $n = 500$ trénovanej neúplnou množinou pri $w = 162$. (x – trénovacie dáta, o – testovacie dáta)

27

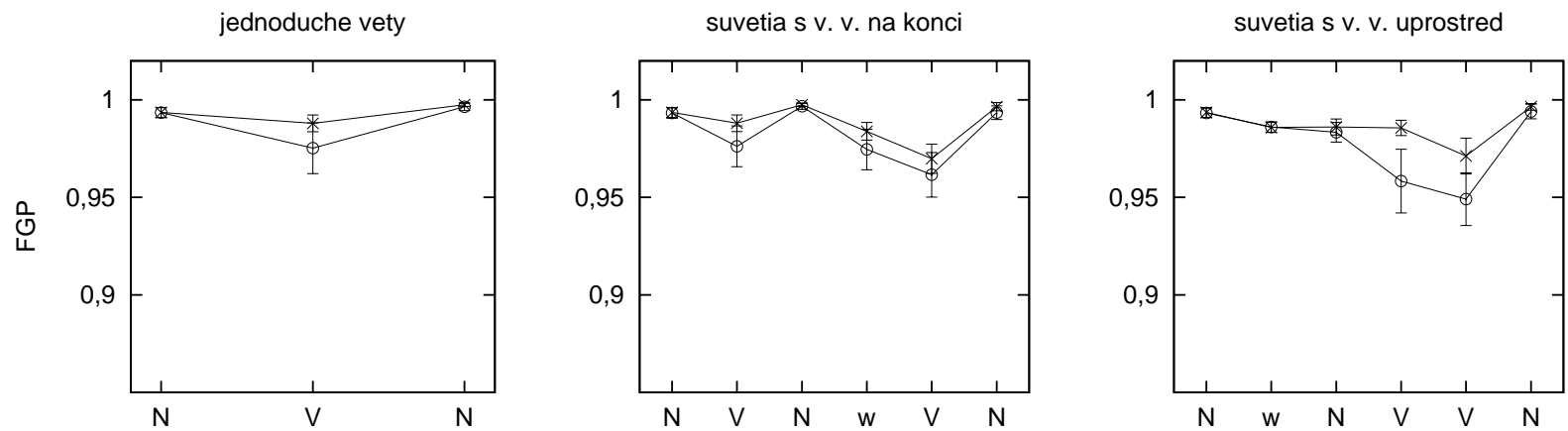


Obrázok 6: Priemerné hodnoty FGP siete s $n = 500$ trénovanej neúplnou množinou pri $w = 162$. (x – trénovacie dáta, o – testovacie dáta)

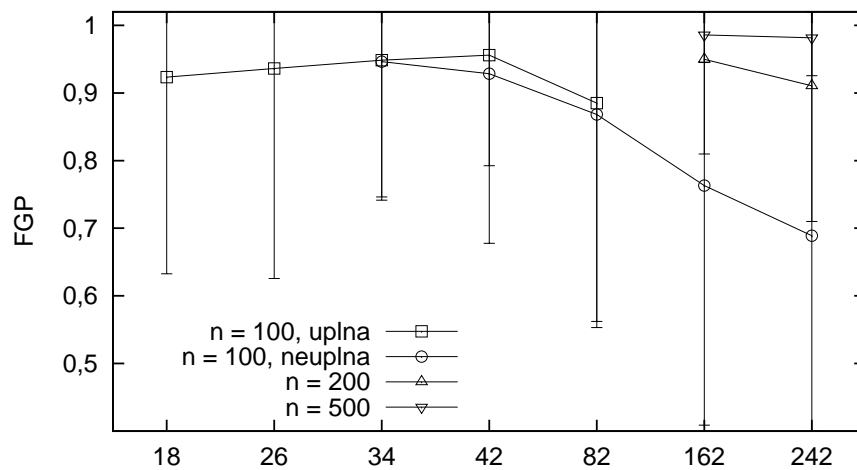


Obrázok 7: Priemerné hodnoty GPE siete s $n = 500$ trénovanej neúplnou množinou pri $w = 242$. (x – trénovacie dáta, o – testovacie dáta)

28



Obrázok 8: Priemerné hodnoty FGP siete s $n = 500$ trénovanej neúplnou množinou pri $w = 242$. (x – trénovacie dáta, o – testovacie dáta)



Obrázok 9: Závislosť FGP od veľkosti dynamického rezervoáru a spôsobu trénovania siete.

4 Záver

Vlastnými simuláciami sa nám podarilo ako overiť výsledky z [6], tak potvrdiť Frankom vyslovenú domnienku, že systematickosť siete sa bude so zväčšujúcim sa lexikónom zlepšovať.

So zväčšovaním lexikónu je nevyhnutné zväčšovať aj dynamický rezervoár siete. Nepodarilo sa nám však určiť vzťah medzi veľkosťou lexikónu a veľkosťou dynamického rezervoáru potrebnou pre optimálne spracovanie danej úlohy. Sieť s $w = 162$, $n = 200$ podávala lepšie výkony ako sieť s $w = 82$, $n = 100$, hoci pomer w/n je v oboch prípadoch prakticky zhodný. Sieť $w = 82$, $n = 100$ bola navyše prekonaná aj sieťou s $w = 242$, $n = 200$, ktorá má pomer w/n výrazne horší.

Na výkonnosť mohla mať vplyv dĺžka učenia, ktorá sa zväčšovala so zväčšujúcim sa lexikónom. Zároveň však klesal podiel tréningových viet na celkovej počte gramatických viet pre daný lexikón: pre $w = 162$ bol $1,79 \cdot 10^{-7}$, pre $w = 242$ len $3,61 \cdot 10^{-8}$. Určiť vzťah medzi veľkosťou lexikónu a veľkosťou dynamického rezervoáru potrebnou pre optimálne spracovanie danej úlohy by mohlo byť predmetom ďalšieho výskumu.

Najlepšie siete zo simulácií, a sice $n = 500$, $w = 162$ a $n = 500$, $w = 242$ podávali skoro dokonalé výsledky. Podiel správne spracovaných testovacích súvetí bol 99,6 %, resp. 99,2 %, priemerná FGP pre jednotlivé druhy testovacích viet 0,988, 0,985 a 0,985, resp. 0,988, 0,983 a 0,977. Frank najlepšie výsledky uvádzal z pokusnej simulácie s $n = 1530$ a $w = 42$, a sice 93,3 %, 0,98, 0,97 a 0,93. Aj tu je zrejmy pozitívny vplyv zväčšeného lexikónu na schopnosť zovšeobecňovať. Pri $n = 500$ sme navyše skrátili učenie sietí na (celkovo) 120 000 viet. Frank pravdepodobne pokračoval v tréningu 312 500 vetami (celkom). Či aj kratšie tréningové množstvo malo vplyv na mierne lepší výsledok našich simulácií, ostáva otvorenou otázkou.

Ďalšou možnosťou pokračovania práce by mohlo byť skúmanie prejavov silnej systematickosti či sledovanie vplyvu rozdielnych kódovanií slov na výkonnosť siete. Taktiež by sme mohli, porovnať použitý model ESN s niektorým z novších variantov, ktoré sa objavili v literatúre (napríklad *delenou ESN s laterálnou inhibíciou* [19]), prípadne sa snažiť nadviazať na výsledky iných autorov vychádzajúcich z [6].

Farkaš a Crocker v [3] napríklad opakujú simulácie z [6] na modeloch využívajúcich rekurzívne samoorganizujúce mapy. Najlepšie výsledky dosiahli skombinovaním modelu RecSOMsard (v úlohe modulu učiaceho sa časovo-kontextové reprezentácie) s dvojvrstvovou doprednou sieťou (v úlohe modulu predikujúceho nasledujúce slovo). Výsledky boli porovnateľné s [6], autori existenciu ďalšieho modelu javiaceho znaky slabšej systematickosti uvádzajú ako ďalší argu-

ment proti kritike konekcionizmu. Použitý model založený na rekurzívnych samoorganizujúcich mapách je navyše biologicky plauzibilnejší než ESN.

Frank taktiež pokračoval v skúmaní systematickosti ESN a tvrdí [7, 8], že vykazujú aj znaky silnej systematickosti. V [8] (spolu s Čerňanským) však na rozdiel od staršej práce použili štandardnú ESN trénovanú lineárnou regresiou výstupných váh, no najmä lokalistické kódovanie slov nahradili vektormi odvodenými od vzájomneho výskytu slov v trénovacej množine. Autori však v závere [8] tvrdia, že správanie ich modelu plne vyhovuje definícii silnej systematickosti podľa Hadleyho [9].

Zoznam obrázkov

1	Priemerné hodnoty FGP siete s $n = 100$ trénovej úplnou množinou pri $w = 42$.	25
2	Priemerné hodnoty FGP siete s $n = 100$ trénovej neúplnou množinou pri $w = 42$.	25
3	Priemerné hodnoty FGP siete s $n = 100$ trénovej neúplnou množinou pri $w = 162$.	26
4	Priemerné hodnoty FGP siete s $n = 200$ trénovej neúplnou množinou pri $w = 162$.	26
5	Priemerné hodnoty GPE siete s $n = 500$ trénovej neúplnou množinou pri $w = 162$.	27
6	Priemerné hodnoty FGP siete s $n = 500$ trénovej neúplnou množinou pri $w = 162$.	27
7	Priemerné hodnoty GPE siete s $n = 500$ trénovej neúplnou množinou pri $w = 242$.	28
8	Priemerné hodnoty FGP siete s $n = 500$ trénovej neúplnou množinou pri $w = 242$.	28
9	Závislosť FGP od veľkosti dynamického rezervoáru a spôsobu trénovej siete.	29

Zoznam tabuliek

1	Gramatika použitá pre generovanie trénovacích a testovacích viet.	4
2	Počty viet v trénovacích a testovacích množinách.	13
3	Dĺžka trénovej siete s $n = 100$ úplnými trénovacími množinami.	14
4	Podiel správne spracovaných viet sieťami $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou.	15
5	Priemerné hodnoty GPE pri spracovaní jednoduchých viet sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou.	16
6	Priemerné hodnoty FGP pri spracovaní jednoduchých viet sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou.	16
7	Priemerné hodnoty FGP pri spracovaní všetkých typov viet z trénovej množiny sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou.	16
8	Priemerné hodnoty FGP pri spracovaní všetkých typov viet z testovacej množiny sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou.	16
9	Priemerné hodnoty GPE pri spracovaní súvetí s vedľajšou vetou na konci sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou.	17

10	Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou na konci sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou. . .	17
11	Priemerné hodnoty GPE pri spracovaní súvetí s vedľajšou vetou uprostred sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou. . .	18
12	Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou uprostred sieťami s $n = 100$ a $w \in \{18, 26, 34, 42, 82\}$ trénovanými úplnou trénovacou množinou. . .	18
13	Počty viet v trénovacích a testovacích množinách pre siete trénované neúplnými trénovacími množinami.	19
14	Dĺžka trénovania sietí s $n \in \{100, 200, 500\}$ neúplnými trénovacími množinami. .	19
15	Podiel správne spracovaných viet sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.	22
16	Priemerné hodnoty FGP pri spracovaní jednoduchých viet sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.	22
17	Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou na konci sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.	23
18	Priemerné hodnoty FGP pri spracovaní súvetí s vedľajšou vetou uprostred sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.	23
19	Priemerné hodnoty FGP pri spracovaní všetkých typov viet z trénovacej množiny sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.	24
20	Priemerné hodnoty FGP pri spracovaní všetkých typov viet z testovacej množiny sieťami s $n \in \{100, 200, 500\}$ a $w \in \{34, 42, 82, 162, 242\}$ trénovanými neúplnou trénovacou množinou.	24

Referencie

- [1] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. 2
- [2] Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225, 1991. 10
- [3] Igor Farkaš and Matthew W. Crocker. Syntactic systematicity in sentence processing with a recurrent self-organizing network. *Neurocomputing*, 71:1172–1179, 2008. 30
- [4] Chrisantha Fernando and Sampsa Sojakka. Pattern recognition in a bucket. In *Advances in Artificial Intelligence. ECAL 2003*, pages 589–597, 2003. 3
- [5] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: a critical analysis. In S. Pinker and J. Mehler, editors, *Connections and Symbols*. MIT Press, Cambridge, Mass., 1988. 1
- [6] Stefan L. Frank. Learn more by training less: systematicity in sentence processing by recurrent networks. *Connection Science*, 18(3):287–302, 2006. i, 1, 2, 3, 7, 13, 14, 30
- [7] Stefan L. Frank. Strong systematicity in sentence processing by an echo state network. In *Artificial Neural Networks – ICANN 2006*, pages 505–514, 2006. 31
- [8] Stefan L. Frank and Michal Čerňanský. Generalization and systematicity in echo state networks. In *Proceedings of the 30th annual conference of the Cognitive Science Society*, 2008. 31
- [9] Robert F. Hadley. Systematicity in connectionist language learning. *Mind and Language*, 9(3):247–272, 1994. 1, 31
- [10] Herbert Jaeger. The echo state approach to analysing and training recurrent neural networks. GMD Report 148, German National Research Center for Information Technology, 2001. 2
- [11] Herbert Jaeger. Short term memory in echo state networks. GMD Report 152, German National Research Center for Information Technology, 2001. 12
- [12] Herbert Jaeger. Adaptive nonlinear system identification with echo state networks. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 593–600. MIT Press, 2003. 3, 10

- [13] Herbert Jaeger. Echo state network. *Scholarpedia*, 2(9):2330, 2007. Available from World Wide Web: http://www.scholarpedia.org/article/Echo_state_network. 2, 3
- [14] Herbert Jaeger. Jacobs university team wins international financial forecasting competition, 2007. Available from World Wide Web: <http://www.faculty.iu-bremen.de/hjaeger/TimeSeriesCompetitionOverview.html>. 3
- [15] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304:78–80, 2004. 3
- [16] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002. 2
- [17] Peter Tiño, Michal Čerňanský, and Ľubica Beňušková. Markovian architectural bias of recurrent neural networks. *IEEE Transactions on Neural Networks*, 15(1):6–15, 2004. 3
- [18] Frank van der Velde, Gwendid T. van der Voort van der Kleij, and Marc de Kamps. Lack of combinatorial productivity in language processing with simple recurrent networks. *Connection Science*, 16(1):21–46, 2004. 1, 2, 4, 7, 13
- [19] Yanbo Xue, Le Yang, and Simon Haykin. Decoupled echo state networks with lateral inhibition. *Neural Networks*, 20:365–376, 2007. 30