



KATEDRA INFORMATIKY
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO, BRATISLAVA

ANALÝZA REKURENTNÝCH SIETÍ S ECHO STAVMI
PRI GENEROVANÍ TRAJEKTÓRIÍ
(bakalárska práca)

TOMÁŠ MALÍK

Študijný odbor: Informatika 9.2.1
Vedúci práce: doc. Ing. Igor Farkaš, PhD.

Bratislava, 2009

Abstrakt

Neurónové siete s echo stavmi sú pomerne novým modelom rekurentných sietí, ktorý umožňuje spracovanie sekvenčných dát a ktorého výhodou je efektívne trénovanie. V teoretickej časti uvádzame formálny popis modelu, vlastnosť echo stavov, trénovacie algoritmy, ako aj možnosti optimalizácie úspešnosti siete. Experimentálna časť zahŕňa viacero úloh, zameraných na generovanie spojitých trajektorií rôznych signálov v čase. Zopakovali sme niektoré výsledky, ktoré vo svojich prácach uvádza Jaeger, a každú úlohu sme súčasne obohatili o vlastné modifikácie, vďaka ktorým sme mohli demonštrovať vplyv voľby parametrov siete na jej úspešnosť predikcie. Úspešne sme overili schopnosť modelu ESN pracovať ako generátor harmonického signálu, preladiteľný frekvenčný generátor, či krátkodobá pamäť.

Kľúčové slová: rekurentná sieť s echo stavmi, dynamický rezervoár, generovanie signálov.

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne použitím citovaných zdrojov.

Bratislava, dňa

Tomáš Malík

Ďakujem môjmu vedúcemu doc. Ing. Igorovi Farkašovi, PhD. za výber témy,
študijné materiály, konzultácie a ostatnú odbornú pomoc.

Obsah

1	Úvod	5
2	Neurónová sieť s echo stavmi	7
2.1	Štruktúra ESN	7
2.2	Vlastnosť echo stavov	9
2.3	Parametre dynamického rezervoára	13
2.4	Trénovanie ESN	14
2.5	Optimalizácia parametrov ESN	17
3	Experimentálna časť	20
3.1	Zavedenie označenia parametrov ESN	20
3.2	Základné vlastnosti ESN	21
3.3	Generátor harmonického signálu	23
3.4	Frekvenčný generátor	28
3.5	Krátkodobá pamäť	33
4	Záver	38
	Literatúra	40
	Zoznam príloh	42

Zoznam obrázkov

2.1	Všeobecná štruktúra ESN, Voliteľné prepojenia sú znázornené bodkovanými šípkami (podľa Jaeger (2001))	8
3.1	Priebeh aktivácie štyroch vybraných neurónov z DR.	21
3.2	Rozdiel v rýchlosti zabúdanie siete (celkový výstup siete).	22
3.3	Zabúdanie počiatočného stavu siete.	22
3.4	Konfigurácia siete pre úlohu generátora harmonického signálu.	23
3.5	Odozva vybraných neurónov DR na tréningový signál.	24
3.6	Správanie siete po nastavení náhodného stavu rezervoára v časoch 0 a 100.	26
3.7	Vývoj rádu chyby ($\log_{10}(\text{MSE}_{test})$) v závislosti od veľkosti spektrálneho polomeru váhovej matice DR.	27
3.8	Vývoj rádu chyby ($\log_{10}(\text{MSE}_{test})$) v závislosti od frekvencie tréningového signálu.	28
3.9	Konfigurácia siete pre úlohu frekvenčného generátora.	29
3.10	Príklad tréningovej postupnosti - druhá polovica.	30
3.11	Predikcia siete na konci vstupnej postupnosti v porovnaní s požadovaným výstupom siete.	31
3.12	Generalizácia natréningovanej siete.	31
3.13	Odozva 5 vybraných neurónov DR v úlohe frekvenčného generátora. Posledný graf (vpravo dole) zobrazuje prislúchajúcu vstupnú postupnosť.	32
3.14	Tréningová postupnosť pre úlohu krátkodobej pamäte.	34
3.15	Predikcia siete v porovnaní s požadovaným výstupom v úlohe krátkodobej pamäte.	34
3.16	Stabilita siete pri generovaní melódie po natréningovaní s použitím šumu.	35
3.17	Výstup siete natréningovanej na generovanie dvoch rôznych postupností.	36

Kapitola 1

Úvod

Pri štúdiu neurónových sietí sa často stretávame s úlohami, ktoré si vyžadujú spracovanie dát s časovou štruktúrou. Klasické dopredné siete sa na tento účel nedajú použiť, nakoľko ich výstup závisí len od aktuálneho vstupu. Isté riešenie ponúka tzv. neurónová sieť s časovým posunom (angl. Time Delay Neural Network, TDNN). TDNN „vidí“ okrem aktuálneho vstupu aj vstupy z niekoľkých predošlých krokov. Táto architektúra je však v niektorých prípadoch nevhodná na reprezentáciu časovo-priestorovej štruktúry vstupných dát. Kvasnička a kol. (1997) poukazujú na tento fakt v kapitole 6.

Primárne sa na spracovanie dát s časovou štruktúrou používajú rekurentné neurónové siete (angl. Recurrent Neural Networks, RNN). Za rekurentnú môžeme vo všeobecnosti považovať každú sieť, v ktorej existuje podmnožina neurónov, ktorá si uchováva informáciu o svojom predošlom stave. Vo výpočte aktivácie neurónu v čase t sa potom zohľadňuje aj informácia o stave neurónu z časového kroku $t - 1$. Stav siete teda nie je závislý len od aktuálneho vstupu, ale aj od predchádzajúcich stavov siete. Preukázalo sa, že rekurentné siete si dokážu vytvoriť vnútornú stavovú reprezentáciu časového kontextu vstupných dát.

Na základe týchto vlastností majú RNN využitie v nasledujúcich úlohách:

- Klasifikačné alebo asociačné úlohy, sieť má určiť, či práve ukončená postupnosť patrí alebo nepatrí do nejakej triedy, prípadne do ktorej z tried ju možno zaradiť.
- Predikčné, sieť má na základe štruktúry prezentovanej vstupnej informácie pred časom t , predpovedať vývoj v čase nasledujúcom po t .
- Generatívne úlohy, sieť má na základe pozorovania určitého vývoja dát pokračovať v časovom rade, zohľadňujúc základnú tendenciu dát obsiahnutú v dostupnom úseku.

Architektúra RNN je oproti dopredným sieťam zložitejšia, čo kladie aj vyššie nároky na tréningové algoritmy. Medzi najznámejšie algoritmy patrí rekurentné učenie v reálnom čase (Real Time Recurrent Learning, RTRL) (Williams a Zipser, 1989) a spätné šírenie chýb v čase (Backpropagation Through Time, BPTT) (Rumelhart a kol., 1986). Obe metódy pracujú na princípe minimalizácie gradientu chyby metódou najstrmšieho spádu. Základný rozdiel je v spôsobe šírenia chybovej informácie. Podrobnejší popis algoritmov RTRL a BPTT uvádza Williams (1992) resp. Williams a Zipser. (1995). Nevýhodou gradientových metód je však problém s dosiahnutím minima chybovej funkcie. Výsledok tréningovania nie je vždy optimálny, nakoľko nie je vylúčené uviaznutie v lokálnom minime povrchu chybovej funkcie. Tieto nedostatky viedli k hľadaniu vhodnejších spôsobov tréningovania. Medzi najúspešnejšie prístupy momentálne patria algoritmy založené na Kalmanovej filtrácii (Haykin, 2002). Klasické modely RNN počas tréningovania upravujú váhy všetkých prepojení, čo vedie k problémom s časovou zložitou. Veľkosť RNN sa preto v praxi pohybuje maximálne v desiatkach neurónov.

V roku 2001 predstavil H. Jaeger nový model rekurentných neurónových sietí: siete s echo stavmi (angl. echo state networks, ESN)(Jaeger, 2001b). Koncepcia tohto modelu umožňuje, na rozdiel od RNN, tréningovanie pomocou jednoduchých tréningových algoritmov, za použitia veľkého počtu vnútorných neurónov. Ukázalo sa, že využitie ESN sietí je v mnohých úlohách veľmi efektívne. V tejto práci sa budeme venovať práve vlastnostiam modelu ESN. V teoretickej časti uvidíme detailný popis ESN siete. V experimentálnej časti budeme prezentovať výsledky, ktoré sme dosiahli s ESN sieťami v rôznych úlohách.

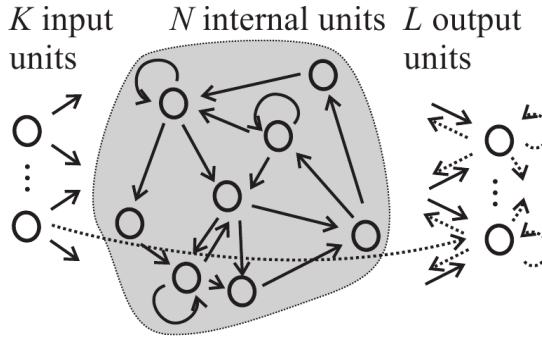
Kapitola 2

Neurónová sieť s echo stavmi

Štruktúra ESN je podobná klasickým RNN. V oboch prípadoch sú neuróny organizované do troch vrstiev: vstupnej, vnútornej (skrytej, rekurentnej) a výstupnej. Základný rozdiel je v rekurentnej vrstve, ktorá sa v prípade ESN skladá z veľkého počtu neurónov. Tie sú navzájom poprepájané spojeniami, ktoré sa netrénujú. Trénujú sa len prepojenia na výstupnej vrstve. Vnútoraná vrstva neurónov je označovaná ako dynamický rezervoár (DR). Nevýhodou klasických RNN je nutnosť použitia komplikovaných a výpočtovo náročných tréningových algoritmov. Keďže model ESN nemení prepojenia vrámci DR, tento môže obsahovať stovky neurónov. Odpadá nutnosť šírenia chybového signálu v čase a na tréning sa dá použiť akýkoľvek algoritmus lineárnej regresie. V prípade RNN sa snažíme dosiahnuť požadované správanie úpravou váh malého počtu neurónov a je preto dôležité optimálne zvoliť hodnoty váh všetkých prepojení. Naopak podstatou ESN je použitie náhodného DR s veľkým počtom neurónov, na zabezpečenie bohatej odozvy na vstupný signál. Odozva DR na vstupný signál má predstavovať jeho obraz vo vysoko rozmernom stavovom priestore rezervoára. Úlohou výstupnej vrstvy je vytvoriť výstup ako lineárnu kombináciu zaujímavých vlastností, ktoré zo vstupného signálu extrahovali vnútorné neuróny. V nasledujúcej časti uvidíme formálny popis ESN (Jaeger, 2001b).

2.1 Štruktúra ESN

Diskrétny časový model ESN siete je odvodený z klasických rekurentných neurónových sietí. Je tvorený vstupnou a výstupnou vrstvou neurónov, ktoré sú prepojené s vnútornou vrstvou neurónov. Uvažujme teda K vstupných neurónov, N vnútorných neurónov a L výstupných neurónov.



Obr. 2.1: Všeobecná štruktúra ESN, Voliteľné prepojenia sú znázornené bodkovanými šípkami (podľa Jaeger (2001))

Aktivácie neurónov v časovom kroku n sú charakterizované:

- vstupným vektorom $\mathbf{u}(n) = (u_1(n), \dots, u_K(n))^T$
- aktivačným vektorom rezervoára $\mathbf{x}(n) = (x_1(n), \dots, x_N(n))^T$
- výstupným vektorom $\mathbf{y}(n) = (y_1(n), \dots, y_L(n))^T$

Hodnoty synaptických váh sú uložené v maticiach:

- $\mathbf{W}^{in} = (\mathbf{w}_{ij}^{in})$, veľkosti $N \times K$, pre vstupné váhy
- $\mathbf{W} = (\mathbf{w}_{ij})$, veľkosti $N \times N$, pre vnútorné prepojenia
- $\mathbf{W}^{out} = (\mathbf{w}_{ij}^{out})$, veľkosti $L \times (K + N + L)$, pre výstupné váhy
- $\mathbf{W}^{back} = (\mathbf{w}_{ij}^{back})$, veľkosti $N \times L$, pre spätné prepojenia (angl. back-projection) od výstupných neurónov k vnútorným neurónom

Povolené sú aj prepojenia z vstupných neurónov priamo na výstupné neuróny, ako aj prepojenia medzi výstupnými neurónmi navzájom. Pre vnútorné prepojenia sa predpokladá, že matica \mathbf{W} indukuje cyklické prepojenia medzi vnútornými neurónmi. Konkrétne modely ESN sietí sa od seba navzájom odlišujú v dôsledku rôznych modifikácií všeobecnej štruktúry s cieľom dosiahnuť optimálne správanie sa siete v danej úlohe. Typickým príkladom modifikácie je vynechanie priameho prepojenia medzi vstupnými a výstupnými

neurónmi, či vzájomného prepojenia výstupných neurónov. Niektoré úlohy si naopak vyžadujú zaradenie spätného prepojenia z výstupu späť do dynamického rezervoára.

Aktivácia vnútorných neurónov je aktualizovaná podľa vzťahu

$$\mathbf{x}(n+1) = f(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{y}(n)), \quad (2.1)$$

kde $f = (f_1, \dots, f_N)$ sú výstupné aktivačné funkcie vnútorných neurónov. Výstup siete sa vypočíta pomocou vzorca

$$\mathbf{y}(n+1) = f^{out}(\mathbf{W}^{out}(\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{y}(n))), \quad (2.2)$$

kde $f^{out} = (f_1^{out}, \dots, f_N^{out})$ sú aktivačné funkcie výstupných neurónov (spravidla lineárne, poprípade sigmoidálne). Vektor $(\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{y}(n))$ je spojenie vstupného vektora, aktivácie rezervoára a predošlého výstupu.

Ako funkciu f budeme používať sigmoidálnu aktivačnú funkciu

$$f(x) = \tanh(x). \quad (2.3)$$

V prípade vynechania niektorého voliteľného prepojenia na výstup sa v 2.2 vypustí aj príslušná zložka z vektora $(\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{y}(n))$. Zároveň sa použije zmenšená matica \mathbf{W}^{out} v ktorej sa vynechajú stĺpce zodpovedajúce váham vynechaných prepojení.

Pre sieť, ktorá z voliteľných prepojení obsahuje len priame prepojenia zo vstupu na výstup a ktorá nepoužíva spätné prepojenie výstupu s rezervoárom, budú aktualizácie rovnice upravené nasledovne. V rovnici pre výpočet aktivácie rezervoára

$$\mathbf{x}(n+1) = f(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n))$$

sa vynechá príspevok $\mathbf{W}^{back}\mathbf{y}(n)$ do aktivácie vnútorných neurónov, keďže spätné prepojenie nie je použité. Pre výpočet výstupu siete dostaneme vzťah

$$\mathbf{y}(n+1) = f^{out}(\mathbf{W}^{out}(\mathbf{u}(n+1), \mathbf{x}(n+1))),$$

kde matica \mathbf{W}^{out} má rozmer $L \times (K \times N)$.

2.2 Vlastnosť echo stavov

Z pohľadu úspešnosti tréningu ESN siete, je kľúčová takzvaná vlastnosť echo stavov (Echo state property). Táto vlastnosť zaručuje správnu odozvu siete na diskretnú vstupnú sekvenciu. Zjednodušene môžeme povedať, že rekurentná neurónová sieť má echo stavy, ak aktivácie neurónov $\mathbf{x}(n)$ v čase n

závisia len od vstupného signálu $\mathbf{u}(n)$ (teda nie od pôvodného stavu siete). Pre každý neurón x_i existuje echo funkcia e_i taká, že ak sieť bežala dostatočne dlhú dobu, potom aktuálny stav neurónu sa dá jednoznačne vyjadriť ako $\mathbf{x}_i(n) = e_i(u(n), u(n-1), u(n-2), \dots)$.

Prakticky to znamená, že ak je sieť riadená vstupnou sekvenciou dostatočnej dĺžky, výsledný stav (aktivácia) siete $\mathbf{x}(n)$ nezávisí od počiatočného stavu siete, ale len od histórie vstupov a výstupov. V Sieti, ktorá má echo stavy môžeme pozorovať „zabúdanie“ siete. Signály privedené do vnútra rezervoára sú tlmené, v dôsledku čoho sa časom ich vplyv na aktiváciu siete vytráca. Teda sieť časom zabúda svoj počiatočný stav.

Ako sa možno dočítať v Maass a kol. (2001), na to aby sme poznali výstup siete s určitou presnosťou, stačí nám poznať len históriu posledných vstupov s podobnou presnosťou. Nepotrebujeme teda poznať stav siete pred privedením vstupnej sekvencie na vstup.

Prítomnosť echo stavov v RNN sieti je vlastnosť, ktorá závisí od váhových matíc \mathbf{W}^{in} , \mathbf{W} , eventuálne aj od matice \mathbf{W}^{back} , ak je použitá. Zároveň je táto vlastnosť závislá na tréningovej množine. V dôsledku toho môže mať tá istá nenatrénovaná sieť echo stavy pre jednu tréningovú množinu, no pre inú už echo stavy nemusí mať.

Pre formálnu definíciu echo stavov, ako ju uvádza Jaeger (2001b), si najskôr zdefinujeme niektoré pojmy a označenia. Budeme uvažovať vstupné sekvencie $(\mathbf{u}(n))_{n \in J} \in U^J$, kde U je kompaktná množina. Skrátené zápisy $\bar{\mathbf{u}}^{\pm\infty}$, $\bar{\mathbf{u}}^{+\infty}$, $\bar{\mathbf{u}}^{-\infty}$, $\bar{\mathbf{u}}^h$, budú v rovnakom poradí označovať postupnosti, ktoré sú zľava a sprava neohraničené ($J = \mathbb{Z}$), sprava neohraničené ($J = k, k+1, \dots$ pre nejaké $k \in \mathbb{Z}$), zľava neohraničené a konečné postupnosti dĺžky h . Analogicky budeme označovať vstupné sekvencie, na ktoré okrem kompaktnosti nebudeme klásť žiadne ďalšie podmienky.

Ďalej zavedieme operátor T , pre zmenu stavu siete (vykonanie jedného časového kroku). Prechod zo stavu $\mathbf{x}(n)$ do stavu $\mathbf{x}(n+h)$ zapíšeme ako $\mathbf{x}(n+h) = T(\mathbf{x}(n), \mathbf{y}(n), \bar{\mathbf{u}}^h)$, čo zodpovedá iterácii siete pomocou pravidiel 2.1 a 2.2 cez vstupnú postupnosť $\bar{\mathbf{u}}^h$, dĺžky h , z počiatočného stavu $\mathbf{x}(n)$ a výstupu $\mathbf{y}(n)$. Pre siete bez spätnej väzby sa operátor zjednoduší na tvar $T(\mathbf{x}(n), \bar{\mathbf{u}}^h)$, pretože informácie o výstupe siete nie sú potrebné pre výpočet nového stavu.

Pre rekurentné neurónové siete je charakteristické rozdelenie ich stavového priestoru do dvoch disjunktných množín A, B . Dynamiku siete preto budeme skúmať v závislosti na množine stavov $A \subset \mathbb{R}^N$, ktoré nadobúda. Budeme požadovať aby $\forall u, u \in U \wedge \forall x \in A$ platilo, že $T(\mathbf{x}, \mathbf{u}) \in A$. Ak sa teda sieť nachádza v stave z množiny A , tak po prezentovaní ľubovoľného vstupu z množiny U prejde sieť opäť do niektorého zo stavov z množiny A . Hovoríme, že sieť spĺňa *standardné podmienky kompaktnosti* ak

- vstupy vyberáme z kompaktného vstupného priestoru U
- stavy siete sú z kompaktnej množiny A

Definícia 2.2.1 *Predpokladajme štandardné podmienky kompaktnosti. Predpokladajme sieť bez spätnoväzbových prepojení. Potom, sieť má echo stavy, ak stav siete $\mathbf{x}(n)$ je jednoznačne definovaný ľubovoľnou zľava neohraničenou vstupnou postupnosťou $\bar{\mathbf{u}}^{-\infty}$. To znamená, že pre ľubovoľnú vstupnú postupnosť $\dots, \mathbf{u}(n-1), \mathbf{u}(n) \in U^{-N}$, pre všetky postupnosti stavov $\dots, \mathbf{x}(n-1), \mathbf{x}(n) \in A^{-N}$, kde $\mathbf{x}(i) = T(\mathbf{x}(i-1), \mathbf{u}(i))$ a $\mathbf{x}'(i) = T(\mathbf{x}'(i-1), \mathbf{u}(i))$ platí, že $\mathbf{x}(n) = \mathbf{x}'(n)$.*

V nasledujúcej definícii zavedieme terminológiu potrebnú pre alternatívnu charakteristiku echo stavov.

Definícia 2.2.2 (a) *Postupnosť stavov $\bar{\mathbf{x}}^{-\infty} = \dots, \mathbf{x}(n-1), \mathbf{x}(n) \in A^{-N}$ sa nazýva kompatibilná zo vstupnou postupnosťou $\bar{\mathbf{u}}^{-\infty} = \dots, \mathbf{u}(n-1), \mathbf{u}(n)$, ak $\forall i < n : T(\mathbf{x}(i), \mathbf{u}(i+1)) = \mathbf{x}(i+1)$.*

(b) *Podobne, neohraničená postupnosť stavov $\bar{\mathbf{x}}^{\infty}$ sa nazýva kompatibilná so vstupnou postupnosťou $\bar{\mathbf{u}}^{\infty}$, ak $\forall i : T(\mathbf{x}(i), \mathbf{u}(i+1)) = \mathbf{x}(i+1)$.*

(c) *Stav siete $\mathbf{x} \in A$ sa nazýva koncovo kompatibilný so vstupnou postupnosťou $\bar{\mathbf{u}}^{-\infty}$ ak existuje postupnosť stavov $\dots, \mathbf{x}(n-1)$ taká, že $T(\mathbf{x}(i), \mathbf{u}(i+1)) = \mathbf{x}(i+1)$ a $\mathbf{x} = \mathbf{x}(n)$.*

(d) *Stav siete $\mathbf{x} \in A$ sa nazýva koncovo kompatibilný s konečnou vstupnou postupnosťou $\bar{\mathbf{u}}^h$ ak existuje postupnosť stavov $\mathbf{x}(n-h), \dots, \mathbf{x}(n) \in A^{h+1}$ taká, že $T(\mathbf{x}(i), \mathbf{u}(i+1)) = \mathbf{x}(i+1)$ a $\mathbf{x} = \mathbf{x}(n)$.*

Definícia 2.2.3 *Predpokladajme štandardné podmienky kompaktnosti a sieť bez spätnoväzbových prepojení.*

1. *Sieť „kontrahuje stavy“, ak pre všetky sprava neohraničené vstupné postupnosti $\bar{\mathbf{u}}^{+\infty}$ existuje nulová postupnosť $(\delta_h)_{h \geq 0}$ taká, že pre všetky stavy $\mathbf{x}, \mathbf{x}' \in A$, pre všetky $h \geq 0$, pre všetky vstupné začiatkové podpostupnosti $\bar{\mathbf{u}}_h = \mathbf{u}(n), \dots, \mathbf{u}(n+h)$, platí, že $d(T(\mathbf{x}, \bar{\mathbf{u}}_h), T(\mathbf{x}', \bar{\mathbf{u}}_h)) < \delta_h$, kde d je Euklidovská vzdialenosť na \mathbb{R}^N*
2. *Sieť „zabúda stavy“, ak pre všetky zľava neohraničené vstupné postupnosti $\dots, \mathbf{u}(n-1), \mathbf{u}(n) \in U^{-N}$ existuje nulová postupnosť $(\delta_h)_{h \geq 0}$ taká, že pre všetky stavy $\mathbf{x}, \mathbf{x}' \in A$, pre všetky $h \geq 0$, pre všetky vstupné koncové podpostupnosti $\bar{\mathbf{u}}_h = \mathbf{u}(n-h), \dots, \mathbf{u}(n)$, platí, že $d(T(\mathbf{x}, \bar{\mathbf{u}}_h), T(\mathbf{x}', \bar{\mathbf{u}}_h)) < \delta_h$.*

3. Sieť „zabúda vstupy“, ak pre všetky zľava neohraničené vstupné postupnosti $\bar{\mathbf{u}}^{-\infty}$ existuje nulová postupnosť $(\delta_h)_{h \geq 0}$ taká, že pre všetky $h \geq 0$, pre všetky vstupné koncové podpostupnosti $\bar{\mathbf{u}}_h = \mathbf{u}(n-h), \dots, \mathbf{u}(n)$, pre všetky zľava neohraničené vstupné postupnosti tvaru $\bar{\mathbf{w}}^{-\infty} \bar{\mathbf{u}}_h, \bar{\mathbf{v}}^{-\infty} \bar{\mathbf{u}}_h$, pre všetky stavy \mathbf{x} koncovy kompatibilné s $\bar{\mathbf{w}}^{-\infty} \bar{\mathbf{u}}_h$ a pre všetky stavy \mathbf{x}' koncovy kompatibilné s $\bar{\mathbf{v}}^{-\infty} \bar{\mathbf{u}}_h$ platí, že $d(\mathbf{x}, \mathbf{x}') < \delta_h$.

Nasledujúce dve vety boli dokázané (Jaeger, 2001b).

Veta 2.2.1 *Predpokladajme štandardné podmienky kompaktnosti a sieť bez spätnoväzbových prepojení. Predpokladajme, že T je spojitý. Potom vlastnosti kontrakcie stavov, zabúdania stavov a zabúdania vstupov siete sú ekvivalentné s prítomnosťou echo stavov v sieti.*

Podmienky pre prítomnosť echo stavov z definície 2.2.3 sa v praxi dajú len ťažko overiť. Pre zabezpečenie existencie echo stavov je preto vhodnejšie riadiť sa vetou 2.2.2, ktorá na základe Lipschitzovej vlastnosti váhovej matice v (a) uvádza postačujúcu podmienku pre echo stavy a v (b) postačujúcu podmienku pre neexistenciu echo stavov.

Veta 2.2.2 *Uvažujme sigmoidálnu sieť s výstupnými neurónmi, ktorých aktivačná funkcia je $f_i = \tanh$. (a) Nech váhová matica \mathbf{W} má $\sigma_{max} = \Lambda < 1$, kde σ_{max} je jej maximálna singulárna hodnota. Potom $d(T(\mathbf{x}, \mathbf{u}), T(\mathbf{x}', \mathbf{u})) < \Lambda d(\mathbf{x}, \mathbf{x}')$ pre všetky vstupy \mathbf{u} a pre ľubovoľné stavy $\mathbf{x}, \mathbf{x}' \in [-1, 1]^N$. Čo implikuje prítomnosť echo stavov pre všetky vstupy \mathbf{u} , pre všetky stavy $\mathbf{x}, \mathbf{x}' \in [-1, 1]^N$.*

(b) *Nech váhová matica \mathbf{W} má spektrálny polomer $|\lambda_{max}| > 1$, kde λ_{max} je najväčšie vlastné číslo matice \mathbf{W} . Potom má sieť asymptoticky nestabilný nulový stav. To implikuje, že sieť nemá echo stavy pre žiadnu vstupnú množinu U , ktorá obsahuje 0 a k nej prípustnú množinu stavov $A = [-1, 1]^N$.*

Z formálneho hľadiska doposiaľ nemáme tvrdenie, ktoré by nám dávalo jednoznačný návod, ako zostrojiť sieť s echo stavmi pre požadovanú vstupnú množinu. Prakticky sa pri konštrukcii ESN sietí vychádza z hypotézy 2.2.1, ktorá sa zatiaľ vždy ukázala ako platná.

Hypotéza 2.2.1 *Nech δ, ϵ sú malé kladné čísla. Potom existuje veľkosť siete N taká, že jej dynamický rezervoár s N neurónmi je vytvorený náhodným vygenerovaním váhovej matice \mathbf{W}_0 s hodnotami váh z uniformného rozdelenia $[-1, 1]$, normalizáciou \mathbf{W}_0 na maticu \mathbf{W}_1 s jednotkovým spektrálnym polomerom daným vzťahom $\mathbf{W}_1 = 1/|\lambda_{max}|\mathbf{W}_0$, kde $|\lambda_{max}|$ je spektrálny polomer matice \mathbf{W}_0 , preškálovaním \mathbf{W}_1 na $\mathbf{W} = (1-\delta)\mathbf{W}_1$, čím \mathbf{W} nadobudne spektrálny polomer $(1-\delta)$, potom sieť $(\mathbf{W}_{in}, \mathbf{W}, \mathbf{W}_{back})$ je sieť s echo stavmi s pravdepodobnosťou $1-\epsilon$.*

Na základe hypotézy 2.2.1 môžeme pre vytvorenie siete s echo stavmi použiť nasledovný postup:

1. Náhodne vygenerujeme maticu interných váh \mathbf{W}_0 .
2. Normalizujeme maticu \mathbf{W}_0 na maticu \mathbf{W}_1 so spektrálnym polomerom 1 a to vynásobením prvkov matice \mathbf{W}_0 hodnotou $1/|\delta_{max}|$, kde $|\delta_{max}|$ je spektrálny polomer matice \mathbf{W}_0 .
3. Preškálujeme maticu \mathbf{W}_1 na maticu $\mathbf{W} = \alpha \mathbf{W}_1$, kde $\alpha < 1$ je škálovací koeficient, čím získame maticu \mathbf{W} , ktorá má spektrálny polomer α .
4. Týmto získame nenatréňovanú sieť $(\mathbf{W}_{in}, \mathbf{W}, \mathbf{W}_{back})$, ktorá má (doposiaľ nebol preukázaný opak) vlastnosť echo stavov bez ohľadu na výber matic \mathbf{W}_{in} a \mathbf{W}_{back} .

2.3 Parametre dynamického rezervoára

Dynamický rezervoár je charakterizovaný maticou prepojení W , preto za jeho parametre budeme považovať parametre práve tejto matice. Ako sme už spomenuli, pre zabezpečenie prítomnosti echo stavov je dôležité preškálovať maticu vnútorných prepojení W tak, aby bola hodnota jej spektrálneho polomeru menšia ako 1. Na to sme v postupe pre vytvorenie ESN siete využili škálovací koeficient $\alpha < 1$, ktorého hodnota zodpovedá spektrálnemu polomeru výslednej matice W .

Škálovací parameter α zohráva dôležitú úlohu aj v charaktere dynamiky rezervoára. Pre úspešný výsledok tréňovania ESN siete je kľúčová práve voľba hodnoty α s prihliadnutím na charakter úlohy. Pre malé hodnoty α dostaneme rezervoár s rýchlou dynamikou, v ktorom privedené signály veľmi rýchlo doznievajú. Naopak pre hodnoty α veľmi blízke 1 dostávame pomalý rezervoár, ktorý signály tlmí len veľmi pomaly. V takomto rezervoári sa posilňuje vplyv predošlých vstupov na nasledujúci stav. Vstupné a výstupné neuróny zase svoj vplyv strácajú. Podľa doterajších empirických pozorovaní závisí časová škála dynamického rezervoára exponenciálne od $1 - \alpha$. Pre postupnosť $\alpha = 0.99, 0.98, 0.97$ tak dostávame exponenciálny nárast rýchlosti rezervoára. Štandardne sa používajú hodnoty α v rozmedzí 0.7 – 0.98.

Od dynamického rezervoára sa vyžaduje, aby odozvy na vstupný signál vo vnútri rezervoára boli čo najbohatšie. Naším cieľom je zvoliť takú konfiguráciu DR, aby sa reakcie jednotlivých neurónov v rezervoári čo najviac odlišovali. To možno dosiahnuť viacerými spôsobmi:

- nehomogénna konektivita v rámci DR

- náhodná voľba váh v DR
- riedka konektivita v DR

Práve riedku konektivitu vo vnútri dynamického rezervoára označoval Jaeger za zdroj rôznorodnej odozvy neurónov v DR. Jeho úvaha vychádzala z rozdelenia dynamiky rezervoára do voľne previazaných oddelených podsystémov v dôsledku riedkej konektivity. Napriek tejto intuícii, mnoho autorov poukázalo na fakt, že siete s plne prepojenými rezervoármi pracujú rovnako dobre ako siete s riedko prepojenými rezervoármi. Tento poznatok je zaujímavý hlavne z hľadiska výpočtovej náročnosti. Totižto ak vytvoríme rezervoár tak, že každý neurón má konštantný počet prepojení na ďalšie neuróny, dostaneme sieť s lineárnym nárastom výpočtovej zložitosti vzhľadom na veľkosť rezervoára N . I keď bolo poukázané na rovnakú funkčnosť plne prepojených sietí, v literatúre sa často odporúča voliť konektivitu v rámci DR v rozmedzí 1–20% na dosiahnutie bohatej odozvy rezervoára. Výber váh, pred preškálovaním na spektrálny rádius α , môže byť napríklad z uniformného rozdelenia z $[-1, 1]$.

2.4 Trénovanie ESN

Základný rozdiel v trénovaní ESN sietí oproti klasickým rekurentným sieťam je v tom, že sa v trénovacom procese upravujú len váhy prepojení výstupnej vrstvy. Matica \mathbf{W} dynamického rezervoára sa teda od svojej počiatočnej inicializácie nemení. Takýto prístup značne urýchľuje proces trénovania a umožňuje pracovať s veľkým počtom neurónov v rezervoári. Kým klasické rekurentné neurónové siete používajú na skrytej vrstve maximálne zopár desiatok neurónov, pri ESN sieťach sa bežne používajú DR so stovkami neurónov. Správna inicializácia DR zabezpečí bohatú odozvu siete na vstupné signály. Úlohou výstupnej vrstvy je zrekonštruovať požadovaný výstupný signál ako lineárnu kombináciu signálov jednotlivých neurónov. Proces trénovania je zodpovedný za výpočet respektíve úpravu výstupnej matice \mathbf{W}^{out} tak, aby sa minimalizovala stredná kvadratická chyba.

Pri trénovaní ESN siete môžeme postupovať podľa nasledovného algoritmu, ktorý uvádza Jaeger (2002). V popise algoritmu budeme uvažovať sieť ktorá používa na výstupe neuróny so sigmoidálnou aktivačnou funkciou. Ďalej budeme predpokladať prítomnosť spätnoväzbových prepojení. Algoritmus je tak vyčerpávajúci a prípadné použitie zjednodušeného modelu si vyžaduje zrejme úpravy vzťahov, ktoré sme načrtli už v kapitole 2.1.

Majme vstupno - výstupnú postupnosť $(\mathbf{u}(1), \mathbf{d}(1)), \dots, (\mathbf{u}(T), \mathbf{d}(T))$. Potom ako výsledok trénovania požadujeme ESN sieť $(\mathbf{W}^{in}, \mathbf{W}, \mathbf{W}^{back}, \mathbf{W}^{out})$,

ktorej výstupy $\mathbf{y}(n)$ aproximujú požadovaný výstup $\mathbf{d}(n)$, ak je sieť riadená vstupnou postupnosťou $\mathbf{u}(n)$.

Treba si však uvedomiť, že natrénovaná sieť je schopná aproximovať požadovaný výstup až potom, čo odznie vplyv prvotnej konfigurácie siete. Budeme teda požadovať, aby natrénovaná sieť aproximovala požadovaný signál v časoch $n = T_0, \dots, T$, kde $T_0 > 1$. Hodnota T_0 závisí od dynamických vlastností siete a pohybuje sa v hodnotách $T_0 = 10$ (pre rýchle siete), až po $T_0 = 500$ (pre pomalé siete). Okrem dobrej aproximácie požadovaného výstupu sa na natrénovanú sieť kladie hlavne požiadavka na dobrú schopnosť generalizácie, čo znamená dobrú aproximáciu nezávislej množiny testovacích dát, ktoré sú generované tým istým systémom ako aj tréningové dáta.

1. krok: Vytvoríme sieť $(\mathbf{W}^{in}, \mathbf{W}, \mathbf{W}^{back})$, ktorá má vlastnosť echo stavov. Na vytvorenie matice vnútorných prepojení \mathbf{W} použijeme postup uvedený v 2.2. Prvky matíc \mathbf{W}^{in} a \mathbf{W}^{back} náhodne vygenerujeme z intervalu $[-1, 1]$ a následne preškálujeme podľa potreby.

Výsledná matica \mathbf{W} by mala byť riedka, aby sa podporila rôznorodosť odoziev od vnútorných neurónov. Veľkosť N matice \mathbf{W} volíme s ohľadom na zložitnosť úlohy a dĺžku tréningovej postupnosti T . Aby sme zabránili efektu preučenia siete na tréningovú postupnosť, volíme N na úrovni $T/10$ až $T/2$ (pre pravidelné, periodické signály volíme hodnoty N bližšie k $T/2$). Pri vytváraní matice \mathbf{W} potrebujeme zvoliť škálovací parameter α . Presnú hodnotu treba doladiť na základe pozorovania. Ako východiskovú môžeme zvoliť hodnotu 0.8.

Správnu funkčnosť siete ovplyvňuje aj škálovanie matíc \mathbf{W}^{in} a \mathbf{W}^{back} . Pre nízke hodnoty prvkov \mathbf{W}^{in} vstup len málo ovplyvňuje chovanie siete. Aktivácie vnútorných neurónov sa pohybujú v lineárnej časti sigmoidy. Sieť zotrúva v stavoch, ktoré sú blízke nulovému stavu. Naopak veľké hodnoty váh \mathbf{W}^{in} spôsobujú, že stav siete v ďalšom kroku je silno závislý od vstupného signálu. Aktivácie vnútorných neurónov sa posúvajú do oblasti saturácie, čo zavádza do rezervoára značne nelineárne správanie. V extrémnych prípadoch pracujú vnútorné neuróny v čisto bipolárnom režime, keď nadobúdajú aktiváciu iba z hodnôt 1, -1.

Analogický vplyv na správanie siete má škálovanie matice \mathbf{W}^{back} . Najmä v úlohách kde sa vyžadujú oscilácie sa ukázalo vhodné škálovanie spätnej matice na vyššie hodnoty.

2. krok: Navzorkovanie tréningovej dynamiky siete.

1. Inicializácia siete do ľubovoľného stavu, napr. $\mathbf{x}(0) = 0$.
2. Necháme sieť bežať cez tréningovú postupnosť, pre $n = 0, \dots, T$, so

vstupným signálom $\mathbf{u}(n)$. Namiesto výstupu siete vnucujeme signál $\mathbf{d}(n-1)$. Výpočet stavu sa tak realizuje pomocou rovnice:

$$x(n+1) = f^{out}(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{d}(n)) \quad (2.4)$$

3. V čase $n = 0$, kde $d(n)$ nie je definované, použijeme $\mathbf{d}(n) = 0$
4. V každom časovom kroku po uplynutí času T_0 zaznamenáme stav siete $\mathbf{x}(n)$, ako nový riadok do matice stavov M . Nakoniec tak dostaneme maticu veľkosti $(T - T_0 + 1) \times (K + N + L)$.
5. Tak isto v každom kroku po uplynutí času T_0 zaznamenáme invertovaný (inverznou funkciou k aktivačnej funkcii výstupných neurónov) požadovaný výstup siete $\tanh^{-1}(\mathbf{d}(n))$, ako nový riadok matice \mathbf{D} . Takto nakoniec dostaneme maticu o veľkosti $(T - T_0 + 1) \times L$.

Pri vytváraní matíc \mathbf{M} a \mathbf{D} je dôležité dať si pozor na to aby sme pridávali vektory $\mathbf{x}(n)$ a $\tanh^{-1}(\mathbf{d}(n))$. Ľahko sa totiž môže stať, že do matíc vložíme dvojicu vektorov $\mathbf{x}(n)$ a $\tanh^{-1}(\mathbf{d}(n-1))$.

3. krok: Výpočet výstupnej matice sa redukuje na výpočet súčinu pseudo-inverznej matice \mathbf{M}^+ s maticou \mathbf{D} , čím dostaneme maticu $(\mathbf{W}^{out})^+$ veľkosti $(K + N + L) \times L$, ktorej i -ty stĺpec obsahuje váhy od všetkých neurónov k i -tému výstupnému neurónu. Po transponovaní tejto matice dostávame hľadanú maticu \mathbf{W}^{out} :

$$\mathbf{W}^{out} = (\mathbf{M}^+\mathbf{D})^T \quad (2.5)$$

4. krok: V tejto fáze máme natrénovanú ESN sieť $(\mathbf{W}^{in}, \mathbf{W}, \mathbf{W}^{back}, \mathbf{W}^{out})$, ktorá je pripravená na použitie. Od tohto okamihu môžeme sieť používať pomocou aktualizáčnych rovníc 2.1 a 2.2.

V druhom kroku je použitá metóda forsírovania učiaceho signálu (angl. teacher - forcing). Ide o to, že počas tréningu sa cez spätné prepojenia neprivádza do rezervoára výstupný signál siete $\mathbf{y}(n)$, ale požadovaný výstupný signál $\mathbf{d}(n)$. Pre niektoré úlohy, ako napríklad generovanie periodických sekvencií, môže byť potrebné použitie forsírovania učiaceho signálu aj po skončení tréningu, aby sa inicializovala správna predikcia siete.

Uvedený algoritmus predstavuje offline metódu dávkového tréningu, čiže najskôr sa len zbierajú informácie o správaní sa siete na tréningovej postupnosti. Potom sa v jednom kroku vypočíta výstupná matica. Ak by sme chceli sieť tréningovať „online“, čiže meniť váhy postupne po každej prichádzajúcej

vzorke, môžeme použiť niektorý z algoritmov *Least mean square (LMS)* alebo *Recursive least square*.

Príklad LMS trénovacieho algoritmu:

1. Sieť necháme bežať s trénovacím signálom $\mathbf{d}(n)$, pričom jej nanucujeme požadovaný výstup. Stav siete počítame podľa vzťahu 2.4, rovnako ako pri offline metóde.
2. V každom kroku upravíme výstupné váhy podľa vzťahu

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \lambda(\mathbf{d}(n) - \mathbf{y}(n))\mathbf{x}(n), \quad (2.6)$$

kde λ je rýchlosť učenia.

V prípade ESN sietí so spätnými prepojeniami sa po natrénovaní siete môžu vyskytnúť problémy so stabilitou a to najmä v prípade, že trénovacie dáta sú nezašumené. Každá sieť má po natrénovaní určitú výstupnú chybu. Nestabilná sieť pri testovaní v každom kroku zväčšuje veľkosť chyby na výstupe, čo spôsobí že sieť sa správa inak, ako sme pri trénovaní požadovali. Tento problém sa dá pomerne jednoducho vyriešiť. Pri trénovaní zavedieme do siete malé množstvo šumu o veľkosti $0.0001, \dots, 0.1$. Počas trénovania použijeme upravenú rovnicu na výpočet stavu

$$\mathbf{x}(n+1) = f(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{d}(n) + v(n)), \quad (2.7)$$

kde $v(n)$, je pridané malé množstvo bieleho šumu.

2.5 Optimalizácia parametrov ESN

Vzhľadom na množstvo parametrov, ktoré vplývajú na výkon ESN, nie je možné uviesť jednoznačný návod, ktorý by viedol k zostrojeniu optimálnej ESN siete. Ak sa pokúšame navrhnuť sieť pre nejakú konkrétnu úlohu a tá nedosahuje požadované výsledky, príčin môže byť viacero. V najhoršom prípade je dôvodom nefunkčnosti neschopnosť modelu ako takého pracovať v danej úlohe. V lepšom prípade môže byť na vine zlá voľba trénovacích dát, ktoré nedostatočne vystihujú charakter problému. Ak sieť javí aspoň náznaky požadovanej funkcionality, no nie však v dostatočnej kvalite a stabilite, je veľmi pravdepodobné, že riešením bude optimalizácia parametrov ESN.

Spektrální polomer

Spektrální polomer α váhovej matice rezervoára W , je základným parametrom ESN siete, ktorý určuje rýchlosť dynamiky vo vnútri rezervoára. Pre úspešné tréningovanie siete je kľúčové zosúladiť rýchlosť vrámci DR s povahou úlohy. Pre veľmi rýchle deje a krátke časové okno pamäte volíme malé hodnoty α . Pre pomalé úlohy, s väčšou pamäťovou náročnosťou volíme α blízke jednej. Výber hodnoty α je tým kritickejší, čím sa hodnoty viacej blížia k jednotke. V praxi môžeme postupovať tak, že na základe intuície a metódy pokus - omyl vyberieme hodnotu α , pri ktorej sieť dostatočne dobre pracuje. Ak chceme minimalizovať chybu siete, čo najlepším výberom spektrálneho polomeru, sieť testujeme s iteratívne sa meniacou hodnotou α , pričom sledujeme priemernú výstupnú chybu. Za optimálnu považujeme tú hodnotu α , použitím ktorej sieť dosiahla najmenšiu chybu.

Veľkosť rezervoára

Od počtu neurónov v DR závisí, ako presne bude sieť schopná aproximovať dynamiku úlohy. Príliš veľký počet neurónov môže spôsobiť preučenie siete na tréningové dáta, čo vedie k zlej schopnosti generalizácie pre vstupy mimo tréningovej množiny. Počet neurónov sa dá odhadnúť na základe veľkosti vstupnej postupnosti. Pri použití väčšieho počtu neurónov ako je dĺžka vstupnej postupnosti by bola odozva siete naučená naspamäť. Pre dosiahnutie dobrej generalizácie sa môžeme riadiť pravidlom, podľa ktorého má na jeden neurón pripadať 2 až 10 hodnôt zo vstupnej postupnosti.

Vizualizácia odozvy rezervoára

Podstatou dobre fungujúcej siete, je správna odozva neurónov vo vnútri DR. Pri úprave parametrov siete je veľmi nápomocné vykresliť si aktivitu niekoľko vybraných neurónov počas tréningovania, či testovania. Tieto informácie nám môžu odhaliť dôvod zlyhania siete, poprípade poslúžia pri ručnom doladení parametrov siete (najmä škálovania matíc).

Ak pozorujeme rýchle oscilácie v úlohe, v ktorej ich neočakávame, môže to byť dôsledok príliš veľkého spektrálneho polomeru matice W , ako aj príliš veľké hodnoty v spätnoväzbovej matici. Problém by sa mal odstrániť po preškálovaní matíc na nižšie hodnoty.

Vo všeobecnosti veľkosť preškálovania vstupnej a spätnoväzbovej matice určuje, aké veľké odozvy budú v DR. Ak chceme dosiahnuť nelineárne správanie, preškáľujeme matice na vyššie hodnoty. Ak si úloha vyžaduje

lineárne správanie použijeme nižšie hodnoty preškálovania matíc, aby sa odozvy neurónov pohybovali v lineárnej časti ich aktivačnej funkcie.

Pridanie šumu

Pri trénovaní siete so spätnými prepojeniami pomocou presných matematických dát sa môžu objaviť problémy so stabilitou siete. Napríklad pri generovaní periodického signálu sieť s časom zvyšuje nepresnosť predikcie, až napokon sieť stratí schopnosť generovať trénovaný signál. Riešenie tohto problému je prekvapivo jednoduché a spočíva v pridaní malého množstva šumu do rezervoára v procese trénovania. Odolnosť siete na malé nepresnosti v predikcii sa podstatne zlepšuje. Alternatívne môže byť šum použitý na zlepšenie schopnosti generalizácie. V praxi je však vhodnejšie na zabránenie preučenia znížiť počet neurónov v DR.

Ďalšie možnosti optimalizácie

V prípade, že vstupné dáta majú strednú hodnotu vzdialenú od nuly, odporúča sa pridať jeden vstupný neurón s konštantnou hodnotou. Hodnota z pridaného vstupu umožní sieti posunúť vstupné dáta na korektnú strednú hodnotu na výstupe. Príliš veľká pridaná hodnota môže budiť neuróny do ich nelineárnej časti. Ak je to nežiadúce, použijeme ako prídavný vstup veľmi malé hodnoty.

Kapitola 3

Experimentálna časť

Jedným z cieľov tejto bakalárskej práce, bola implementácia modelu ESN a následná analýza tohto modelu v úlohe generovania spojitých trajektórií. V tejto kapitole podrobne popíšeme jednotlivé experimenty. Najskôr sme si prakticky overili, či správanie implementovaného modelu korešponduje s požiadavkami uvedenými v teoretickej časti. Potom sme model adaptovali pre prácu v rôznych úlohách. Pre každú úlohu sme analyzovali vplyv jednotlivých parametrov na výkonnosť siete. Model ESN bol implementovaný v MATLAB-e. Zdrojové súbory sa nachádzajú na priloženom CD.

3.1 Zavedenie označenia parametrov ESN

Pri popisoch experimentov budeme používať nasledujúce označenie parametrov:

- α - spektrálny polomer matice W
- d_W - hustota matice W , percentuálny počet nenulových prvkov matice W
- K–N–L - konfigurácia siete, kde K, N a L sú v poradí počty vstupných, vnútorných a výstupných neurónov W
- S_{in} - škálovací parameter vstupnej matice, matica má prvky s intervalu $[-S_{in}, S_{in}]$
- S_{back} - škálovací parameter výstupnej matice, matica má prvky s intervalu $[-S_{out}, S_{out}]$
- MSE_{train}, MSE_{test} - stredná kvadratická chyba pri tréningu, resp. testovaní

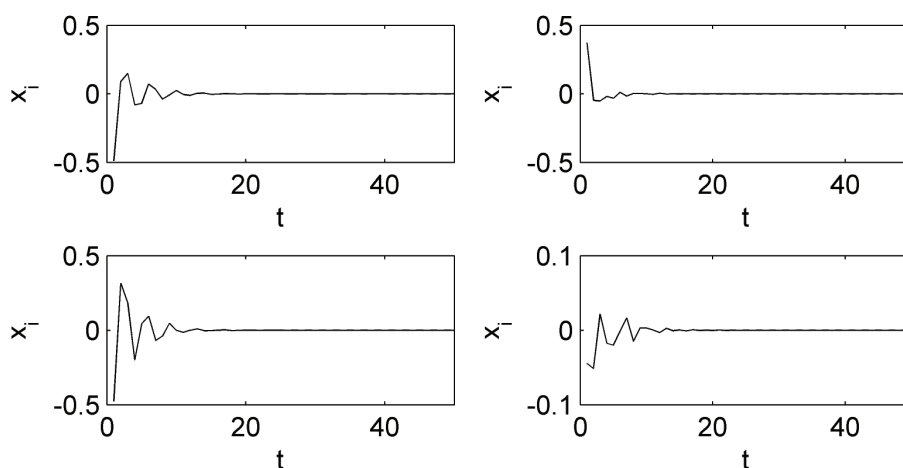
- f - aktivačná funkcia vnútorných neurónov
- f_{out} - aktivačná funkcia výstupných neurónov
- v - veľkosť pridaného bieleho šumu

3.2 Základné vlastnosti ESN

Ako sme spomenuli v teoretickej časti, každá ESN sieť musí spĺňať podmienku existencie echo stavov. Z definície echo stavov intuitívne vyplývajú niektoré základné vlastnosti ESN:

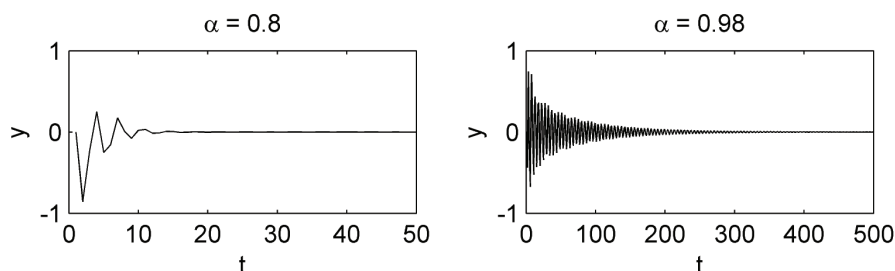
- Signály vo vnútri DR sú tlmené, čoho dôsledkom je zabúdanie siete
- Stav siete závisí len od histórie vstupov, nie od počiatočnej konfigurácie siete

Tieto vlastnosti si overíme na zjednodušenom modeli ESN, ktorá nebude obsahovať vstupné neuróny, nakoľko nebudeme potrebovať do siete privádzať signály zvonka. Dynamický rezervoár pozostáva z 20 neurónov. Na výstupe je jeden neurón, z ktorého odoberáme výsledný výstupný signál zo siete. Spätne prepojenia z výstupu tiež vynecháme. Aktivačná funkcia na vnútorných ako aj vonkajších neurónoch bola $f = f_{out} = \tanh$. Stav siete, čiže vektor x , bol náhodne inicializovaný. Sieť sme nechali bežať dostatočný počet krokov, až kým vplyv počiatočnej aktivácie úplne neodznel.



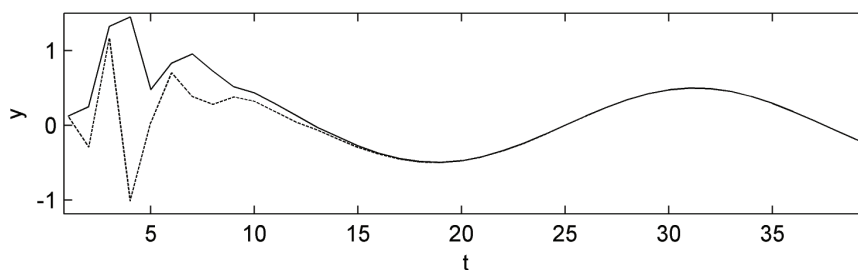
Obr. 3.1: Priebeh aktivácie štyroch vybraných neurónov z DR.

Na obrázku 3.1 je graficky znázornená aktivácia štyroch vybraných interných neurónov počas prvých 50 časových krokov. Ako je vidieť, aktivita neurónov je tlmená a sieť postupne konverguje do nulového stavu. Ďalšia dôležitá vlastnosť, ktorú možno pozorovať, je odlišná reakcia vnútorných neurónov na počiatočný stav siete.



Obr. 3.2: Rozdiel v rýchlosti zabúdanie siete (celkový výstup siete).

Vplyv spektrálneho polomeru matice vnútorných prepojení α na dynamiku siete je zobrazený na obrázku 3.2. Na ľavom grafe pozorujeme výstup siete pre $\alpha = 0.8$. Vplyv počiatočnej konfigurácie siete sa stráca už po 20. krokoch. Na pravom grafe je znázornené výrazné spomalenie dynamiky pri nastavení $\alpha = 0.98$. Signál je vo vnútri tlmený zjavne slabšie a vplyv počiatočnej konfigurácie pretrváva ešte aj po 200 krokoch.



Obr. 3.3: Zabúdanie počiatočného stavu siete.

Druhú spomínanú vlastnosť ESN sme si overili na rovnakej konfigurácii siete ako v predošlom prípade, len s výnimkou použitia lineárnej výstupnej funkcie a pridania spätných prepojení. Tú istú sieť sme nechali bežať dvakrát s rôznou počiatočnou aktiváciou rezervoára. Do siete sme cez spätné prepojenia privádzali nanútený sínusový signál s periódou 25 krokov. Na obrázku 3.3 vidíme výstup siete odoberaný z jediného výstupného neurónu siete.

Prerušovaná čiara zobrazuje priebeh výstupu siete s inou počiatočnou konfiguráciou ako súvislá čiara. Signál na tomto neuróne je lineárnou kombináciou aktivácií neurónov DR a teda istým spôsobom vyjadruje stav siete. Podľa očakávaní sa spočiatku reakcia siete líši, ale postupom času vplyv počiatočnej inicializácie stavu vymizne. Po 30. krokoch je už stav totožný a je zrejme, že závisí len od vstupnej postupnosti.

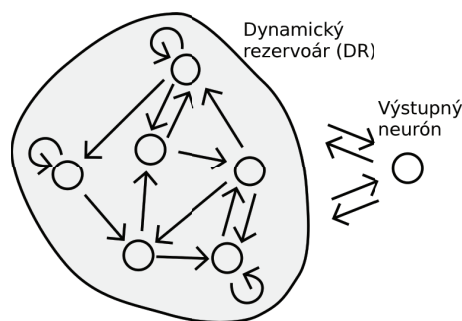
Na záver úvodných pozorovaní uvedieme poznatky, ktoré sme sledovali v správaní sa netrénovanej siete. Sieť nemala vstupné neuróny. Spätné prepojenia boli použité. Na výstupe bol jeden neurón. Ostatné parametre siete boli náhodne generované. Keď sme sieť nechali bežať, pozorovali sme nasledovné tendencie:

- Sieť sa časom ustálila na konštantnej výstupnej hodnote.
- Na výstupe siete sa v nejakej forme objavil periodický signál, najmä pri preškáľovaní spätných prepojení na vyššie hodnoty.
- Na výstupe siete sa objavilo chaotické správanie, najmä pri hodnotách spektrálneho polomeru $\alpha > 1$, keď je veľká pravdepodobnosť, že sieť nemá echo stavu.

Skúmanie doteraz prezentovaných výsledkov si nevyžadovalo tréning siete. Na výpočet prezentovaných výsledkov stačilo použitie vzťahov 2.1 a 2.2, po patričnej úprave.

3.3 Generátor harmonického signálu

Cieľom tohto experimentu je natréning modelu ESN v úlohe frekvenčného generátora, ktorý bude generovať sínusový signál s fixnou frekvenciou.



Obr. 3.4: Konfigurácia siete pre úlohu generátora harmonického signálu.

Použitý model je znázornený na obrázku 3.4. Do rezervoára nevedie žiaden vstup. Nevyhnutná súčasť, ktorú si vyžaduje táto úloha, sú spätné prepojenia z výstupu do rezervoára. Naučená frekvencia závisí od tréningovej sekvencie, ktorá je počas tréningu privádzaná cez spätné váhy do rezervoára. Po natrénovaní sa cez spätné prepojenia už šíri výstupný signál siete z predošlého časového kroku. Sieť sme vytvorili pomocou algoritmu, ktorý sme uviedli v podkapitole 2.4 o tréningu ESN. Detaily konfigurácie siete sú uvedené v tabuľke 3.1.

K–N–L	α	d_W [%]	S_{back}	f	f_{out}	v
0–20–1	0.8	25	1.5	tanh	id	0

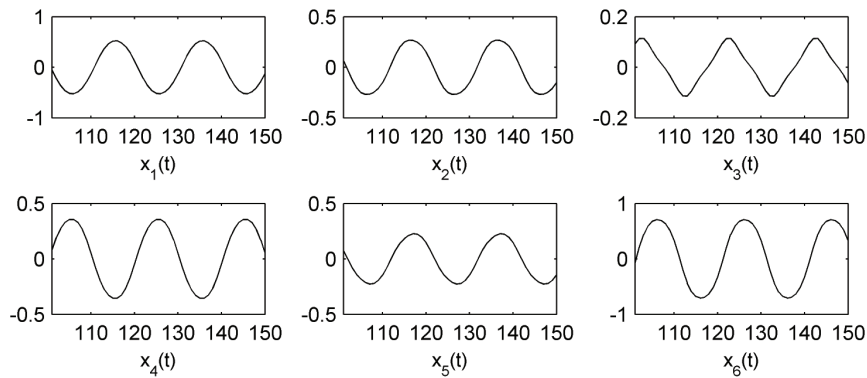
Tabuľka 3.1: Parametre siete pre úlohu generátora harmonického signálu

Tréning prebiehal pomocou offline algoritmu. Vstupná postupnosť bola vygenerovaná pomocou funkcie sínus tak, aby dĺžka periódy bola 25 krokov. Amplitúda signálu dosahovala výšku 0.5. Dĺžka tréningovej postupnosti bola 500. Keďže použitý model je mierne modifikovaný, bolo potrebné nasledovne upraviť rovnice pre výpočet stavu a výstupu siete:

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{y}(n)), \quad (3.1)$$

$$\mathbf{y}(n+1) = f^{out}(\mathbf{W}^{out}\mathbf{x}(n+1)). \quad (3.2)$$

Počas tréningu sú zmenené len výstupné váhy v matici \mathbf{W}^{out} . Pôvodná hodnota váh v matici \mathbf{W}^{out} nie je definovaná a v priebehu tréningu sa nikde nepoužíva. Tréning je rozdelený do dvoch fáz. V prvej fáze necháme bežať sieť 500 krokov s ľubovoľného počiatočného stavu v čase $n = 1$. Zároveň cez



Obr. 3.5: Odozva vybraných neurónov DR na tréningový signál.

maticu spätných prepojení \mathbf{W}^{back} privádzame do siete trérovací signál, čo vo vnútri DR vyvoláva aktivačnú odozvu. Na obrázku 3.5 je vykreslená odozva šiestich neurónov z DR na trérovací signál.

Na jednotlivých odozvách si môžeme všimnúť dve podstatné veci:

- Aktivačné odozvy neurónov v DR sa od seba líšia tvarom, amplitúdou a fázou signálu.
- Všetky aktivačné odozvy neurónov sú periodické signály, pričom ich perióda je zhodná s periódou trérovacieho signálu.

Počas prvej fázy trérovania sú stavové vektory rezervoára $\mathbf{x}(n)$ ukladané ako riadky do matice \mathbf{M} . Súčasne sa do riadkov matice \mathbf{D} ukladá požadovaný výstup v čase n . Riadky do matíc \mathbf{M} a \mathbf{D} začíname pridávať až po odznení vplyvu počiatočného stavu siete. V tomto prípade sme vynechali aktivácie z prvých 100 krokov siete, počas ktorých vplyv počiatočného stavu bezpečne odznel. Takto sme dostali maticu \mathbf{M} veľkosti 400×25 a maticu \mathbf{T} veľkosti 400×1 .

V druhej fáze potrebujem vypočítať maticu \mathbf{W}^{out} tak, aby sme minimalizovali odchýlku od požadovaného výstupu. Výstup siete v kroku n môžeme vyjadriť nasledovne:

$$y(n) = \sum_{i=1}^{N=25} \mathbf{w}_i^{out} \mathbf{x}_i(n) \quad (3.3)$$

Vo výsledku chceme zvoliť maticu \mathbf{W}^{out} tak, aby sme minimalizovali veľkosť strednej kvadratickej chyby.

$$\text{MSE}_{train} = 1/400 \sum_{n=101}^{500} (d(n) - y(n))^2 = 1/400 \sum_{n=101}^{500} (d(n) - \sum_{i=1}^{N=25} \mathbf{w}_i^{out} \mathbf{x}_i(n))^2 \quad (3.4)$$

Existuje viacero spôsobov ako dosiahnuť minimalizáciu uvedenej strednej kvadratickej chyby. V princípe možno použiť ľubovoľný algoritmus lineárnej regresie. V našom prípade pri offline metóde trérovania sa výpočet výstupných váh redukuje na výpočet jednej pseudoinverznej matice. Výsledné váhy dostaneme vynásobením pseudoinverznej matice k \mathbf{M} s maticou \mathbf{D} .

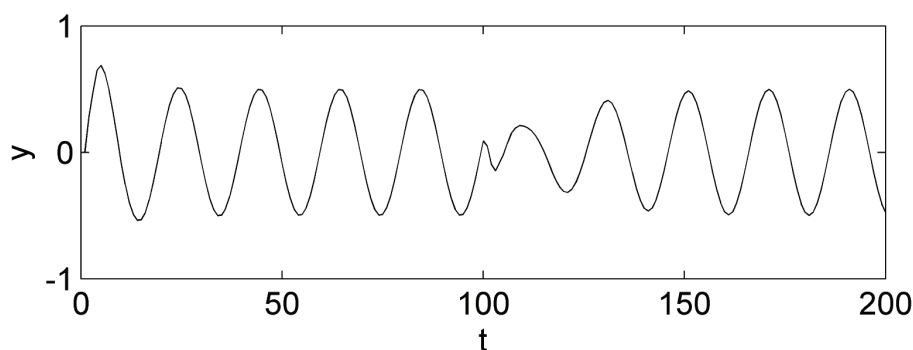
$$\mathbf{W}^{out} = (\mathbf{M}^+ \mathbf{D})^T \quad (3.5)$$

Po nastavení vypočítanej výstupnej matice sme sieť nechali bežať 100 krokov s forsírovaním učiaceho signálu. Takto sme dosiahli trérovaciu chybu $\text{MSE}_{train} = 9.6\text{e}-29$. Potom sme sieť nechali bežať ešte ďalších 100 krokov, ale už bez trérovacieho signálu, ktorý sme vymenili za vlastný výstup siete. Výsledná testovacia chyba dosiahla $\text{MSE}_{test} = 1.9\text{e}-25$. Testovacia chyba je o

dva rády vyššia oproti trénovacej, napriek tomu je veľmi nízka. Sieť sa teda perfektne naučila generovať sínusoidu s dĺžkou periódy 25 krokov.

Odkúšali sme aj tréovanie pomocou online trénovacieho algoritmu, keď sme v každom kroku upravovali váhy matice \mathbf{W}^{out} pomocou rovnice 2.6. Týmto spôsobom sa podarilo sieť natréovať, no výsledky boli horšie ako v prípade offline metódy. Prudký pokles výstupnej chyby trval len krátku dobu a následne jej veľkosť klesala už len veľmi pomaly. Je možné že toto správanie súvisí s periodickým charakterom úlohy.

Nakoľko sieť vykazuje po natréovaní istú chybu, prichádza do úvahy otázka, či sa táto chyba nebude časom kumulovať a viesť až k absolútnej nestabilite siete. Táto domnienka sa v tomto prípade, ale nepotvrdila. Chyba siete počas jej behu zostáva ustálená aj po veľmi veľkom počte krokov. Dokonca pri tejto úlohe ani nezáleží na tom, z akého stavu sieť po natréovaní spustíme. Sieť sa v krátkom čase prispôbi na generovanie natréovaného signálu. Vysoká stabilita siete spočíva v jednoduchosti úlohy generovania sínusoidy. Natréovaná sieť predstavuje dynamický systém s jediným atraktorom, ktorým je perióda trénovacieho signálu.

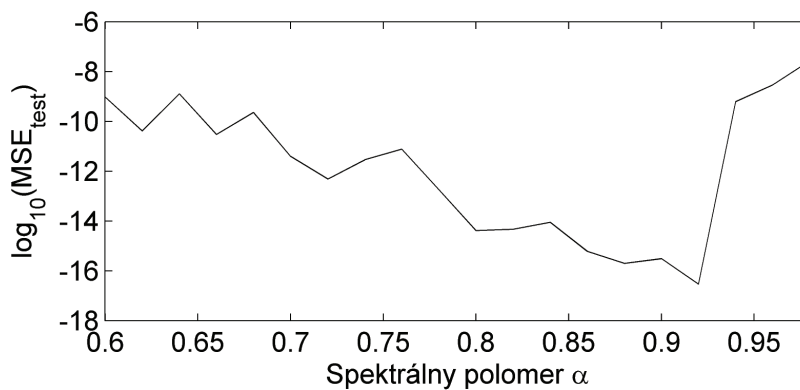


Obr. 3.6: Správanie siete po nastavení náhodného stavu rezervoára v časoch 0 a 100.

Obrázok 3.6 znázorňuje stabilitu natréovanej siete. V krokoch 0 a 100 bol nastavený náhodný stav rezervoára, napriek tomu bola sieť v priebehu niekoľkých krokov schopná bezchybne pokračovať v generovaní naučeného signálu. Stabilita, ktorú sme pozorovali pri tejto úlohe je výnimočne veľká a ako sme už spomínali, súvisí s jednoduchosťou úlohy. V zložitejších úlohách je práve dosiahnutie požadovanej stability rozhodujúcim problémom.

V podkapitole 2.3 o parametroch dynamického rezervoára sme sa zmienili o dôležitosť správnej voľby spektrálneho polomeru α . Od jeho hodnoty totiž závisí rýchlosť zabúdania siete a teda vnútorná dynamika rezervoára.

Z definície vlastností echo stavov vieme, že pre zaručenie prítomnosti echo stavov v sieti je vhodné zvoliť $\alpha < 1$. Pre malé hodnoty α dostávame rýchlu sieť, naopak pre hodnoty blízke 1 dostávame pomalú sieť. Z tohto hľadiska je zaujímavé sledovať, ako vplýva výber hodnoty spektrálneho polomeru DR na výkonnosť siete (na jej testovaciu chybu) v úlohe generátora sínusového signálu. Vyskúšali sme teda vytvoriť sieť s 50 neurónmi v DR s konektivitou 20%. Túto sieť sme opakovane trénovali generovať sínusoidu s periódou 40 krokov, pre rôzne veľkosti spektrálneho polomeru α . Pri zmene spektrálneho polomeru sa zachovávala vnútorná štruktúra DR, čiže vzájomné poprepájanie neurónov zostalo rovnaké a menila sa len veľkosť váh tak, aby sa dosiahol požadovaný spektrálny polomer.

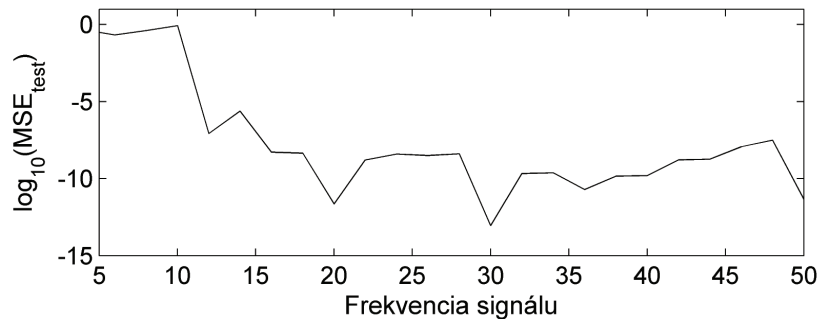


Obr. 3.7: Vývoj rádu chyby ($\log_{10}(\text{MSE}_{\text{test}})$) v závislosti od veľkosti spektrálneho polomeru váhovej matice DR.

Vplyv spektrálneho polomeru na veľkosť testovacej chyby je znázornený na obrázku 3.7. Ako je vidieť, dynamiku danej úlohy najlepšie vystihuje DR preškálovaný približne na hodnotu $\alpha = 0.92$. Konkrétne pri tejto úlohe je pre $\alpha = 0.6$ testovacia chyba dvojnásobná, no aj napriek tomu je na výstupe siete stále signál s požadovanou periódou a amplitúdou. Až pri testovacej chybe rádo vyššej ako 10^{-6} začínala mať sieť problémy s udrжанím požadovanej amplitúdy.

Frekvencia signálu resp. dĺžka periódy závisí od pripravenej trénovacej postupnosti. Samozrejme tu existujú limity a nie je možné naučiť sieť generovať signál s ľubovoľnou dĺžkou periódy. Čím viac krokov pripadne na jednu periódu, tým väčšie pamäťové nároky sú na sieť kladené. Na obrázku 3.8 je znázornený vývoj testovacej chyby vzhľadom na frekvenciu trénovacieho signálu. V tomto prípade bol použitý DR s 50 neurónmi a $\alpha = 0.88$. Dĺžka trénovacej postupnosti bola 500 krokov. Frekvencia signálu vyjadruje počet

sínusových periód, ktoré sa vyskytovali v tréningovej postupnosti. Pre frekvenciu 25, teda pri počte 25 periód na 500 krokov, dostávame 20 krokov na jednu periódu. V ľavej časti grafu vidieť, že so znižujúcou frekvenciou začína rásť testovacia chyba. Od momentu, keď má jedna perióda väčšiu dĺžku ako 50 krokov, už sieť nebola schopná zrekonštruovať tréningový signál. Na zlepšenie schopnosti siete predikovať pomalšie signály, môžeme skúsiť zmeniť hodnotu α . Avšak hlavným parametrom, ktorý vplýva na veľkosť pamäte, je počet neurónov v DR. Ak je signál na výstupe deformovaný, poprípade má dvojnásobnú frekvenciu oproti tréningovému signálu, pravdepodobne sieť pracuje vo svojich pamäťových limitoch.



Obr. 3.8: Vývoj rádu chyby ($\log_{10}(\text{MSE}_{\text{test}})$) v závislosti od frekvencie tréningového signálu.

K optimalizačným postupom patrí vizualizácia odozvy vo vnútri DR. Táto metóda môže pomôcť pri určení tzv. wash-out time, teda času zabúdania siete. Ďalej môžeme pozorovať vplyv preškálovania spätnej matice na veľkosť aktivácie neurónov v DR. Nám sa osvedčilo také škálovanie matice \mathbf{W}^{back} , pri ktorom najviac aktivované neuróny začali vykazovať nelineárne správanie. Vo všetkých sieťach v úlohe generovania sínusového signálu to v našom prípade znamenalo hodnoty spätných váh z rozsahu $[-1.5, 1.5]$.

3.4 Frekvenčný generátor

V prvom experimente sme si na úlohe generátora sínusoidy ukázali spôsob tréningovania a základné vlastnosti sietí s echo stavmi. Teraz prejdeme na zložitejšiu úlohu, v ktorej ESN sieť funguje ako preladiteľný frekvenčný generátor. Štruktúru siete z predošlého experimentu rozšírime o vstup. Vstupná postupnosť $\mathbf{u}(n)$ nastavuje frekvenciu signálu, ktorý budeme očakávať na

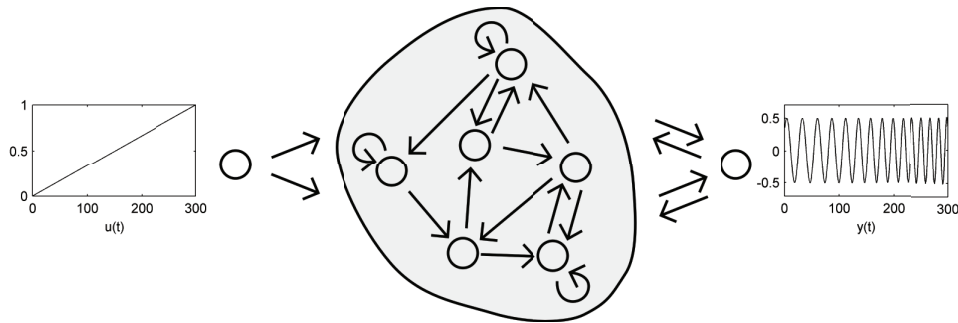
výstupe siete. Vstupná hodnota nula predstavuje minimálnu frekvenciu, hodnota 1 zase maximálnu frekvenciu. Keďže už ide o zložitejšiu úlohu, zvýšili

$\mathbf{K-N-L}$	α	d_W [%]	S_{in}	S_{back}	f	f_{out}	v
1-200-1	0.8	20	0.5	3.4	tanh	tanh	0.01

Tabuľka 3.2: Parametre siete pre úlohu frekvenčného generátora

sme počet neurónov DR na 200. Kompletne parametre použitej siete sú uvedené v tabuľke 3.2.

Schematicky je úloha frekvenčného generátora znázornená na obrázku 3.9. Prítomné bolo aj priame prepojenie zo vstupu na výstup, hoci nie je na obrázku zaznačené. Vstupný signál z intervalu $[0, 1]$ určuje frekvenciu výstupného signálu, ktorá sa plynule mení. Aj v tejto úlohe je použitá matica spätných prepojení \mathbf{W}^{back} . Trénované sú len prepojenia smerujúce na výstup. Amplitúda výstupného signálu opäť závisí na trénovacej postupnosti, my sme zvolili sínusový signál z intervalu $[-0.5, 0.5]$.



Obr. 3.9: Konfigurácia siete pre úlohu frekvenčného generátora.

Keďže bol do siete pridaný vstup a s ním aj nové prepojenia, je nutné príslušne upraviť aj systémové rovnice:

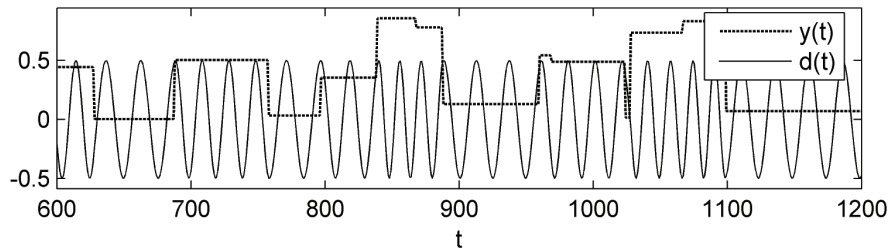
$$\mathbf{x}(n+1) = f(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{y}(n)), \quad (3.6)$$

$$\mathbf{y}(n+1) = f^{out}(\mathbf{W}^{out}\mathbf{z}(n+1)), \quad (3.7)$$

kde $\mathbf{z}(n) = [\mathbf{u}(n); \mathbf{x}(n)]$ je stĺpcový vektor, ktorý vznikne spojením vektorov \mathbf{u} a \mathbf{x} . Pri tejto úlohe sme z výstupných prepojení nepoužili prepojenia medzi výstupnými neurónmi navzájom. Takto dostávame výstupnú maticu \mathbf{W}^{out} veľkosti $L \times (K + N)$.

S pozorovaní sme zistili, že v úlohe frekvenčného generátora je možné rovnako dobre použiť lineárnu ako aj sigmoidálnu výstupnú funkciu. Aby

sme poukázali na rozdiely v tréningovom algoritme pri použití nelineárnej výstupnej funkcie, budeme prezentovať model s výstupnou funkciou \tanh . Tréningová postupnosť mala dĺžku 1200 krokov a pozostávala z vstupnej postupnosti u a výstupnej postupnosti d . Najskôr sme vygenerovali vstupnú postupnosť, čiže signál z intervalu $[0, 1]$, ktorý v čase menil výšku amplitúdy s pravdepodobnosťou 1,5%. Na základe postupnosti $\mathbf{u}(n)$ bola vygenerovaná príslušná výstupná postupnosť $\mathbf{d}(t)$, tak aby v čase t zodpovedala frekvencia signálu $\mathbf{d}(t)$ vstupnej hodnote $\mathbf{u}(t)$. Frekvenčný rozsah bol udaný minimálnou a maximálnou frekvenciou. Prezentovaný model bol tréningovaný na dĺžku periódy 14 – 26 krokov.



Obr. 3.10: Príklad tréningovej postupnosti - druhá polovica.

V tejto úlohe sme použili len offline metódu tréningovania. V prvej fáze tréningovania sa na vstup siete privádza vstupná postupnosť $u(t)$. Zároveň, keďže používame maticu \mathbf{W}^{back} , je cez spätné váhy forsírovaný požadovaný výstupný signál $d(t)$. Za použitia systémových rovníc 3.6 a 3.7 získame postupnosť rozšírených stavov siete $\mathbf{z}(1), \dots, \mathbf{z}(n_{max})$. Tieto zapíšeme ako riadky do matice stavov \mathbf{M} . Ďalej potrebujeme vytvoriť maticu \mathbf{D} . V prípade že na výstupe siete používame nelineárnu funkciu, neukladáme do riadkov tejto matice priamo prvky $d(t)$, ale ich invertované hodnoty. V našom prípade $\tanh^{-1}(d(t))$. Pred vytvorením matic \mathbf{M} a \mathbf{D} sme odstránili prvých 200 stavov, ktoré by mohli byť ovplyvnené počiatočnou inicializáciou siete.

Tréningová chyba, ktorú sa snažíme minimalizovať:

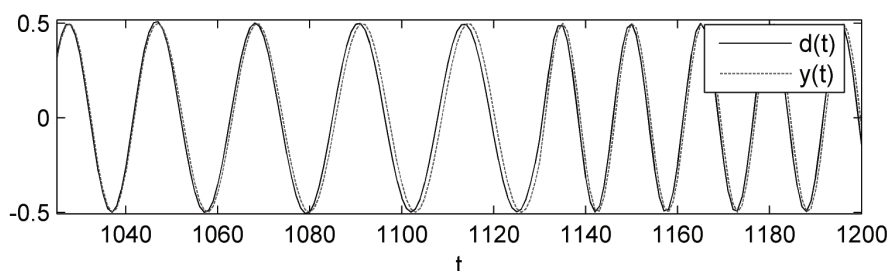
$$MSE_{train} = 1/1000 \sum_{n=201}^{1200} (\tanh^{-1}(d(n)) - \tanh^{-1}(y(n)))^2. \quad (3.8)$$

Výpočet výstupnej matice $\mathbf{W}^{out} = (\mathbf{M} + \mathbf{D})^T$ zostáva nezmenený. Sieť sme po natréningovaní nechali bežať 200 krokov s forsírovaním signálu $d(t)$. Potom až do skončenia vstupnej postupnosti sieť bežala so svojím vlastným výstupom. Takto sme dosiahli tréningovú chybu $MSE_{train} = 7.1e-7$ a testovaciu chybu $MSE_{test} = 7.9e-4$. Sieť bola aj v tomto prípade schopná pracovať v danej

úlohe s ľubovoľného počiatočného stavu. Na dosiahnutie takejto stability, ale bolo potrebné pridať pri tréningu do rezervoára malé množstvo uniformného bieleho šumu. Sieť sa tak do istej miery naučila korigovať výstupnú chybu. V prvej fáze tréningu sme použili upravenú stavovú rovnicu:

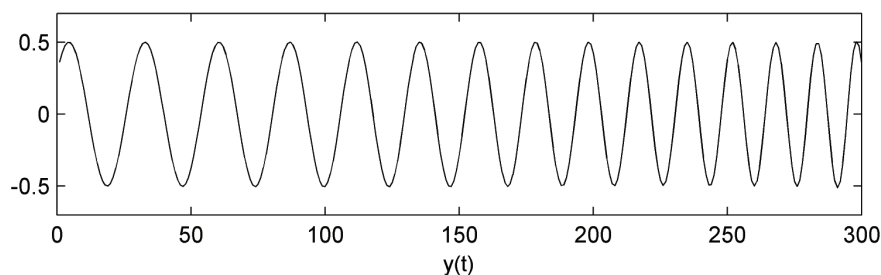
$$\mathbf{x}(n+1) = f(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{y}(n) + v(n)), \quad (3.9)$$

kde $v(n)$ je malé množstvo uniformného bieleho šumu.



Obr. 3.11: Predikcia siete na konci vstupnej postupnosti v porovnaní s požadovaným výstupom siete.

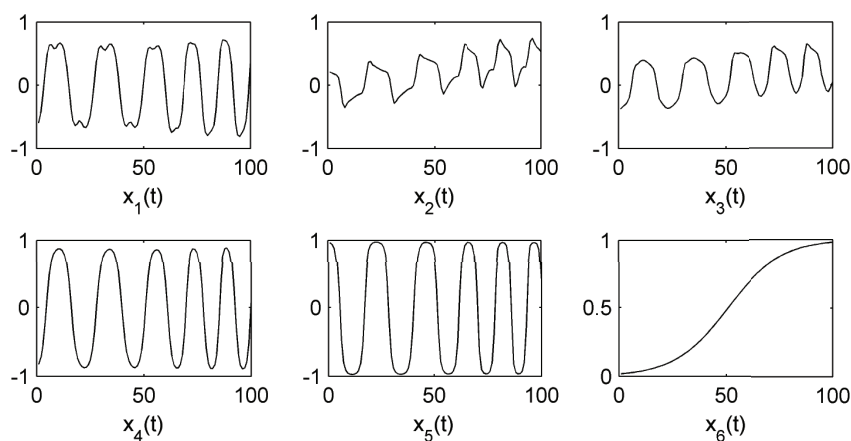
Na obrázku 3.11 je znázornená predikcia natrénovanej siete. Prvých 200 krokov bolo použité forsírovanie učiaceho signálu. Vykreslený úsek grafu zobrazuje výstup siete z konca vstupnej postupnosti. Ako je vidno, sieť aj po takmer 1000 krokoch generuje signál s veľkou presnosťou. Amplitúda ako aj frekvencia je presne dodržaná. Chybu siete spôsobujú hlavne fázové posuny, ktoré vznikajú pri zmene frekvencie, čo môžeme pozorovať okolo kroku 1080, keď výstupný signál mierne spomalí a fázový posun vzrastie. Následne v čase 1130 sa fázový posun zmenší potom, čo nastáva zvýšenie frekvencie.



Obr. 3.12: Generalizácia natrénovanej siete.

Pri skúmaní rôznych modelov neurónových sietí nás často zaujíma schopnosť generalizácie. Tréningové dáta nemôžu pokrývať celú množinu vstupov

danej úlohy a preto sa snažíme vybrať vhodnú reprezentatívnu vzorku. K úlohe frekvenčného generátora sme príklad vstupnej postupnosti uviedli na obrázku 3.10. Vstupná postupnosť obsahuje náhodne vybrané hodnoty z intervalu $[0, 1]$. Od siete požadujeme nielen to, aby bola schopná generovať správny výstup pre vstupnú tréningovú postupnosť, ale hlavne schopnosť správne reagovať na nové vstupy, s ktorými sa sieť pri tréningu nestretla. Aby sme demonštrovali schopnosť generalizácie ESN v úlohe frekvenčného generátora, nechali sme sieť bežať na vstupnej postupnosti, ktorá lineárne prechádza naprieč celým vstupným rozsahom. Vstupná postupnosť mala dĺžku 300 krokov a bola vytvorená lineárnou funkciou $f(t) = t/300$. Odozvu siete zachytáva obrázok 3.12. Výstupný signál plynule zvyšuje svoju frekvenciu od minimálnej až po maximálnu a sieť teda adekvátne reaguje na vstupné hodnoty, ktoré sú pre ňu nové.



Obr. 3.13: Odozva 5 vybraných neurónov DR v úlohe frekvenčného generátora. Posledný graf (vpravo dole) zobrazuje prislúchajúcu vstupnú postupnosť.

V tejto úlohe je frekvencia výstupného signálu riadená hodnotou vstupu. Zaujímalo nás preto, ako sa prejavuje zmena vstupnej hodnoty na odozvu v DR. Opäť sme si pripravili vstupnú postupnosť, ktorá prechádza celým vstupným rozsahom. Tentokrát bola vygenerovaná vhodne upravenou funkciou \tanh a jej dĺžka bola 100 krokov. Takto sme získali odozvu neurónov DR, na ktorej možno dobre pozorovať vplyv vstupného signálu na ich aktiváciu. Priebeh aktivácie piatich vybraných neurónov je na obrázku 3.13. Posledný graf (vpravo dole) znázorňuje použitú vstupnú postupnosť. Odozvy neurónov sú opäť periodické signály so spoločnou periódou, ktorá sa navyše

v čase mení. U niektorých neurónov okrem zmeny periódy pozorujeme aj vertikálny posun, či zmenu vzorky signálu v závislosti od hodnoty vstupu.

Na záver tohto experimentu ešte uvedieme vplyv niektorých parametrov na výkon siete. Dôležitú úlohu zohráva preškáľovanie vstupnej a spätnej matice. Ručne na základe pozorovania treba zvoliť vhodný pomer vplyvu signálov prechádzajúcich týmito maticami. Príliš veľká matica \mathbf{W}^{in} môže spôsobiť, že na výstupe sa v nejakej forme objaví vstupný signál bez oscilácií. V opačnom prípade sa vôbec neprejaví vplyv vstupu na frekvenciu signálu na výstupe. Príliš veľká matica \mathbf{W}^{back} môže mať za následok, že na výstupe sa objavia rýchle nekontrolované oscilácie. Ak má sieť problém udržať požadovanú amplitúdu výstupného signálu, pravdepodobne pomôže zvýšenie vplyvu spätnej väzby. Ďalej treba pamätať na to, že čím väčšie je preškáľovanie matic \mathbf{W}^{in} a \mathbf{W}^{back} , tým väčšia je nelinearita siete, čo môžeme kontrolovať vykreslením aktivácií neurónov DR. Na dosiahnutie dobrej stability siete bolo potrebné pri tréňovaní použiť biely šum. Stabilizačný účinok sa prejavil pre široký rozsah veľkosti šumu, no šum väčší ako 0.01 bol už na úkor presnosti. Optimálna hodnota závisí aj od ostatných parametrov siete. Vplyv šumu možno pozorovať aj pri vykreslení aktivity DR.

3.5 Krátkodobá pamäť

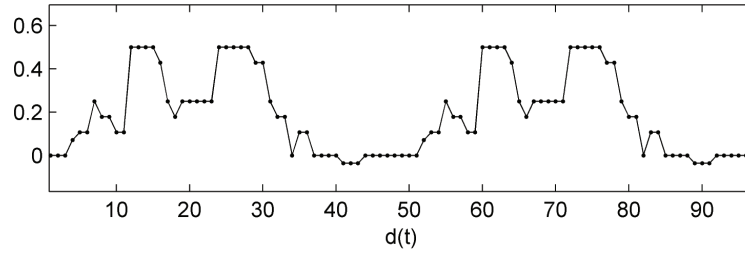
V tomto experimente sa budeme snažiť naučiť sieť generovať periodický signál s celočíselnou periódou. Architektúra siete je podobná ako v úlohe generátora sínusoidy 3.3. Sieť nebude mať vstup, naopak spätné prepojenia budú použité. Presná konfigurácia je uvedená v tabuľke 3.3

K–N–L	α	d_W [%]	S_{back}	f	f_{out}	v
0–400–1	0.908	1.25	2	tanh	tanh	0.0005

Tabuľka 3.3: Parametre siete pre úlohu krátkodobej pamäte.

Tréňovaciu postupnosť sme zvolili podobne ako Jaeger (2001b), melódiu piesne „The House of the Rising Sun“. Notám melódie boli priradené číselné hodnoty od -1(g#) po 14 (a'), pre každý poltón jedno číslo. Tieto hodnoty sme potom vydělili číslom 28, aby sme dostali hodnoty zapadajúce do vstupného intervalu neurónov $[-1, 1]$. Výsledná melódia pozostáva z 48 tónov, čo je teda zároveň aj periódou tréňovacieho signálu. Ten je zobrazený na obrázku 3.14.

Na tréňovanie sme použili postupnosť dĺžky 1500 krokov, z čoho sme prvých 500 krokov zahodili. Sieť sme najskôr natréňovali bez použitia šumu, pričom sme dosiahli extrémne nízku tréňovaciu chybu $MSE_{train} = 1.6e-30$.



Obr. 3.14: Trénovacia postupnosť pre úlohu krátkodobej pamäte.

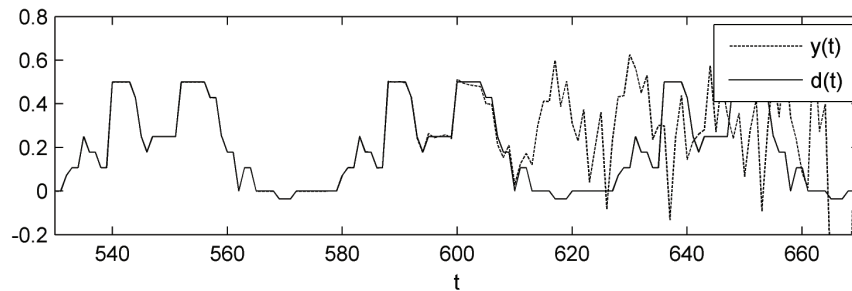
Túto malú chybu Jaeger vysvetľuje nasledovne. Po uplynutí úvodných 500 krokov sieť skonvergovala do periodickej sekvencie 48 stavov. Výstupné váhy minimalizujú chybu:

$$\text{MSE}_{\text{train}} = 1/1000 \sum_{n=501}^{1500} (\tanh^{-1}(d(n)) - \tanh^{-1}(y(n)))^2. \quad (3.10)$$

Keďže zaznamenané stavy sa periodicky opakujú, výpočet sa zjednodušuje na

$$\text{MSE}_{\text{train}} = 1/48 \sum_{n=501}^{548} (\tanh^{-1}(d(n)) - \tanh^{-1}(y(n)))^2, \quad (3.11)$$

čo vedie k výpočtu lineárneho mapovania $\mathbf{W}^{\text{out}} : \mathbb{R}^{400} \rightarrow \mathbb{R}$ určeného 48 nezávislými parametrami, pre ktoré existuje mnoho perfektných riešení.



Obr. 3.15: Predikcia siete v porovnaní s požadovaným výstupom v úlohe krátkodobej pamäte.

Takto natrénovanú sieť sme otestovali, aby sme zistili či je schopná sama generovať tréningovú postupnosť. Prvých 500 krokov bol sieť nanútený učiaci signál. Správanie siete po ukončení forsírovania učiaceho signálu je zachytené

na obrázku 3.15. Potom, čo sme od kroku 500 nechali sieť bežať s vlastným výstupom, nasledoval úsek asi 100 krokov, počas ktorého sieť spoľahlivo generovala požadovanú výstupnú sekvenciu. Postupne sa však na výstupe objavuje narastajúca chyba, až napokon na výstupe prevládne chaotické správanie.

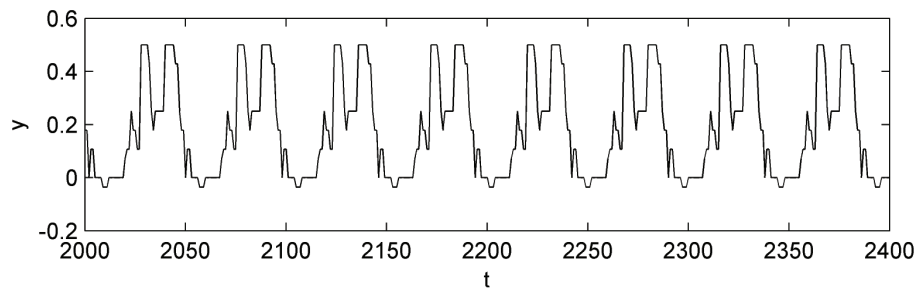
Ukázalo sa teda, že stav $\mathbf{x}(501)$ s operátorom zmeny stavu T^{48} nie je stabilným pevným bodom. Tento problém so stabilitou vyriešime podobne ako v predošlej úlohe a to pridaním šumu do rezervoára v procese tréovania. Použijeme upravenú stavovú rovnicu:

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}(\mathbf{y}(n) + v(n))). \quad (3.12)$$

Malé množstvo šumu zabezpečí, že stavy siete sa budú pohybovať okolo pôvodnej nezašumenej periodickej sekvencie stavov. Takto dostaneme reálne 1000 rôznych parametrov $\mathbf{x}'(n)$ pre minimalizáciu chyby:

$$\text{MSE}_{train} = 1/1000 \sum_{n=501}^{1500} (\tanh^{-1}(d(n)) - \tanh^{-1}(\mathbf{W}^{out}\mathbf{x}'(n)))^2. \quad (3.13)$$

Riešenie ktoré sme týmto spôsobom dostali je už stabilné. Sieť je schopná bez rastúcej chyby nepretržite generovať naučený signál. Pri použití uniformného bieleho šumu z intervalu $[-0.0005, 0.0005]$ sme dosiahli tréovacia chybu $\text{MSE}_{train} = 1.1\text{e-}9$ a testovaciu chybu $\text{MSE}_{test} = 1.3\text{e-}9$. Voľba veľkosti šumu je možné voliť so širokého rozsahu. Nám sa podarilo dosiahnuť najmenšiu chybu pre veľkosť šumu rádovo 10^{-8} . Tréovacia chyba dosiahla hodnotu $\text{MSE}_{train} = 1.6\text{e-}14$ a testovacia chyba $\text{MSE}_{test} = 4.0\text{e-}14$.



Obr. 3.16: Stabilita siete pri generovaní melódie po natréovaní s použitím šumu.

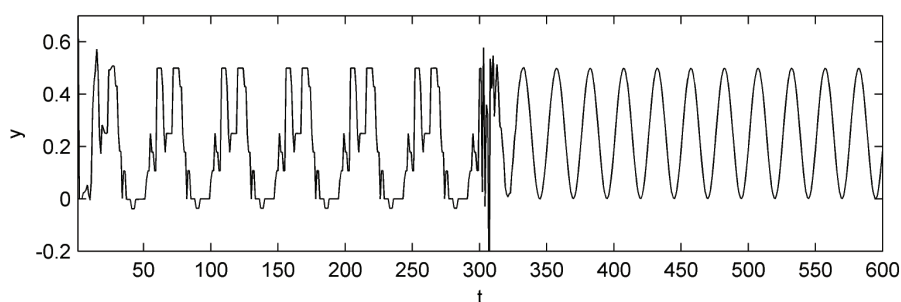
Stabilitu siete pri použití šumu znázorňuje obrázok 3.16. Do kroku 500 bol sieti nanútený tréovacia signál. Ako vidieť na obrázku, sieť aj po uplynutí

2000 krokov bez problémov generuje požadovaný signál. Mohlo by sa zdať, že tréovací signál je príliš krátky vzhľadom na počet neurónov v DR. Okrem uvedenej melódie sme skúšali aj iné signály, ako napríklad modulovanú sínusoidu s periódou 500 krokov, pri ktorých sme dosiahli rovnakú úspešnosť.

V tejto úlohe sme skúmali aj možnosť naučiť sieť viacero rôznych postupností. Jedným z riešení je použiť osobitný výstupný neurón pre každú postupnosť. Podľa toho, ktorú postupnosť by sme chceli generovať, by sme použili príslušný výstupný neurón. Pomocou nanútenia signálu by sme cez tento neurón inicializovali sieť na generovanie zvolenej postupnosti. Toto riešenie je intuitívne funkčné, nakoľko DR je univerzálny a potrebná natrénovaná informácia je uložená vo váhach príslušného výstupného neurónu.

Ďalšou možnosťou je skúsiť natrénovať sieť len s jedným výstupným neurónom na generovanie rôznych postupností. Na tento účel sme vytvorili tréovaciu postupnosť v ktorej sme skombinovali melódiu z tejto úlohy a sínusoidu z úlohy frekvenčného generátora. Každý signál zaberá polovicu tréovacej postupnosti s celkovou dĺžkou 3000 krokov. Pre takto modifikovanú úlohu sme zvýšili počet neurónov DR na 600. Upravili sme aj proces tréovania. Okrem stavov z prvých 500 krokov sme zahodili aj 500 stavov bezprostredne nasledujúcich po stave 1500, keď došlo k zmene tréovacieho signálu.

Na obrázku 3.17 je zobrazený výstup siete. Prvých 200 krokov sme sieti nanútili postupnosť tónov melódie. Sieť bola schopná stabilne pokračovať v generovaní melódie. V krokoch 300 - 500 sme začali sieti nútiť signál sínusoidy, na čo sa sieť prispôbila a opäť bola schopná stabilne generovať požadovaný signál.



Obr. 3.17: Výstup siete natrénovanej na generovanie dvoch rôznych postupností.

V tomto experimente sa nám podarilo natrénovať model ESN na generovanie dvoch rôznych postupností. Výber signálu, ktorý chceme sieťou ge-

nerovať, pozostáva s inicializačného nanútenia signálu na dostatočnú dobu. Kvôli prehľadnosti sme vykreslili len 100 krokov v ktorých sieť samostatne generovala natrénovaný signál, sieť však dokázala pokračovať v generovaní signálu ľubovoľne dlho.

Kapitola 4

Záver

V roku 2001 prišiel Jaeger z modelom echo sietí, ktoré eliminujú nedostatky klasických rekurentných sietí a to použitím rezervoára dynamického správania s pevne nastavenými váhami. ESN sa osvedčili v úlohách predikcie, opakovania, či aproximácie časových radov. V práci sme sa snažili zhrnúť informácie potrebné na úspešné oboznámenie sa s koncepciou ESN, ako aj na ich praktickú implementáciu.

Jadro teoretickej časti tvorí formálny popis všeobecného modelu ESN. Pre úplnosť sme spomenuli presné formálne definície vlastnosti echo stavov, na ktorých je celý model postavený. Pre aplikáciu ESN v praxi sme uviedli komplexný postup na zostrojenie ako aj natrénovanie siete. Na záver teoretickej časti sme sa zmienili o možnostiach a postupoch pri optimalizácii parametrov ESN sietí.

Experimentálna časť obsahuje popis výsledkov, ktoré sme dosiahli s implementovaným modelom ESN. Ako prvé sme uviedli výsledky, ktoré demonštrujú vlastnosti ESN sietí. Čiastočne sme tak overili správnosť implementácie. Ďalej sme model ESN modifikovali pre tri rôzne úlohy, v ktorých sa nám podarilo zopakovať doposiaľ prezentované výsledky, ako aj priniesť vlastné modifikácie a pozorovania. V popise jednotlivých úloh uvádzame spôsob úpravy a použitia trénovacieho algoritmu vzhľadom na použitú konfiguráciu siete, ako aj vplyv výberu parametrov na výkonnosť siete.

V úlohe generátora harmonického signálu sa nám podarilo natrénovať sieť na generovanie sínusoidy s konštantnou frekvenciou. Demonštrovali sme vplyv veľkosti spektrálneho polomeru váhovej matice DR na úspešnosť siete. Poukázali sme aj na obmedzený rozsah frekvencií, respektíve dĺžky periódy signálu, ktorý je možné pri použitej konfigurácii siete spracovať. Pre ďalšie skúmanie môže byť zaujímavý vplyv veľkosti DR na úspešnosť predikcie pre signály s veľkou dĺžkou periódy.

V druhom experimente sa nám podarilo natrénovať model ESN ako frek-

venčný generátor. Úspešne sme overili schopnosť generalizácie. Natrénovaná sieť bola schopná generovať aj signál s frekvenciou, ktorá nebola prítomná v tréningovej postupnosti. Túto úlohu sme sa pokúsili rozšíriť na generátor modulovaného signálu. Dosahované výsledky ale neboli uspokojivé. Realizácia tejto modifikácie by preto mohla byť predmetom pre ďalšie skúmanie.

Použitie ESN ako krátkodobej pamäte sme otestovali na rôznych signáloch. Sieť preukázala schopnosť generovať periodické signály primeranej dĺžky. Navyše sa nám podarilo sieť natrénovať na generovanie dvoch rôznych sekvencií. Sieť generovala tú sekvenciu, na ktorú sme ju inicializovali pomocou forsírovania učiaceho signálu.

Informácie uvedené v tejto práci by mali byť dostatočujúce na úvodné zoznámenie sa modelom ESN a mali by zodpovedať základné otázky týkajúce sa prípadnej implementácie modelu.

Literatúra

- S. Haykin.: *Kalman filtering and neural networks*. John Wiley & Sons, Inc., New York, 2002, online ISBN 0-471-36998-5.
- H. Jaeger.: *A tutorial on training recurrent neural networks. covering bptt, rtrl, ekf, and the echo state network approach*. GMD report 159. German National Research Center for Information Technology, 2002.
- H. Jaeger.: *Echo State Network*. Scholarpedia, 2007.
http://www.scholarpedia.org/article/Echo_state_network
- H. Jaeger.: *Short term memory in echo state networks*, GMD Report 152, German National Research Center for Information Technology, 2001
- H. Jaeger.: *The echo state approach to analysing and training recurrent neural networks*. Technical report GMD report 148. German National Research Center for Information Technology, 2001.
- V. Kvasnička a kol.: *Úvod do teórie neurónových sietí*. IRIS, Bratislava, 1997. 118-132 s.
- M. Lukoševičius, H. Jaeger.: *Overview of Reservoir Recipes*, Jacobs University technical report Nr. 11, 2007
- W. Maass, T. Natschlaeger, H. Markram.: *Real-time computing without stable states: A new framework for neural computation based on perturbations*. Submitted, 2001.
- D. E. Rumelhart, G. E. Hinton, R. J. Williams.: *Learning internal representations by error propagation*. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing*, volume 1, MIT Press.
- B. Schrauwen, D. Verstraeten, J. V. Campenhout.: *An overview of reservoir computing: theory, applications and implementations* Electronics and Information Systems Department, Ghent University, Belgium, 2007.

- R. Williams, J. D. Zipser.: *Experimental analysis of the real-time recurrent learning algorithm*. Connection Science, 1989.
- R. J. Williams, D. Zipser.: *Gradient-based learning algorithms for recurrent networks and their computational complexity*. In: Chauvin, Y., Rumelhart, D.E.: *Back-propagation: Theory, architectures and applications*. Hillsdale, NJ: Erlbaum, 1995.
- R. J. Williams.: *Training recurrent networks using extended Kalman filter*. Int. Joint Conf. Neural Networks, vol. 4, 1992, 241-246 s.
- NIPS 2006 Workshop on Echo State Networks and Liquid State Machines*, 2006. <http://www.esn-lsm.tugraz.at/>.

Prílohy

Príloha 1. - CD médium so zdrojovými kódmi k experimentálnej časti.

Na priloženom CD je uložená táto bakalárska práca v elektronickej podobe. Okrem toho sú na ňom uložené zdrojové kódy implementovanej ESN siete.

Názov adresára	Popis obsahu
STM	Zdrojové súbory pre úlohu krátkodobej pamäte.
STM_2	Zdrojové súbory pre úlohu krátkodobej pamäte s rozšírením na tréning dwoch rôznych postupností.
SWG	Zdrojové súbory pre úlohu generátora harmonického signálu.
TSWG	Zdrojové súbory pre úlohu frekvenčného generátora.

Názov súboru	Popis funkcie
createReservoir.m	Vytvorí dynamický rezervoár.
create_esn.m	Vytvorí model ESN siete.
scaling.m	Preškáluje matice modelu ESN siete.
train.m	Implementácia offline tréningového algoritmu.
runNetwork.m	Nechá sieť bežať zadaný počet krokov.
runNetworkWF.m	Nechá sieť bežať s inicializačným forsírovaním učiaceho signálu.
getMSEtest.m	Vypočíta testovaciu chybu.
getMSEtrain.m	Vypočíta tréningovaciu chybu.
simpleSine.m	Hlavný súbor pre úlohu generátora harmonického signálu.
tunableSine.m	Hlavný súbor pre úlohu frekvenčného generátora.
stm.m	Hlavný súbor pre ESN v úlohe krátkodobej pamäte.