Echo-state network and its memory capcity

Jakub Bahyl[1]
FMFI, UK v Bratislave

V tejto práci prezentujem výsledky pre pamäťovú kapacitu jednoduchšej verzie rekurzívnej neurónovej siete "echo-state network". Sieť je trénovaná dátami z náhodnej uniformnej distribúcie a testovaná na schopnosti rekonštruovať minulosť. V práci skúmam vhodnosť rôznych parametrov, akými sú hĺbka minulosti, inicializácia, aktivácia a sparsita váhových matíc.

Keywords: recurrent NN, memory capacity

"The human brain has 100 billion neurons, each neuron connected to 10 thousand other neurons. Sitting on your shoulders is the most complicated object in the known universe."
Michio Kaku

I. INTRO

Programovacia časť bola tvorená v jazyku Julia v0.5.1. Prílohou k tomuto projektu je zdrojový kód, ktorý je spustiteľný bez akýchkoľvek inštalácií cez Jupyter na stránke Juliabox. Prihlásiť sa dá voľne cez Google+, Github alebo LinkedIn. Po uploadnutí kódu by mal bez problémov bežať.

V tomto projekte využívam Pythonové knižnice *PyPlot* a *Numpy*.

II. ECHO-STATE NETWORK IN SHORT

Echo-state network (ESN) je typ architektúry rekurentnej neurónovej siete vhodný pre učenie s učiteľom (supervised learning). Hlavná myšlienka tejto architektúry spočíva v tom, že pre každý vstup si skrytá vrstva neurónov (vrstva je len jedna) vyrobí svoju kópiu, ktorú sieť neskôr skombinuje s ďalším vstupom. Týmto sieť efektívne používa svoju *pamäťovú schopnosť* a moduluje signál vstupujúci do skrytej vrstvy. V literatúre sa skrytá vrstva často nazýva *rezervoárom*, preto ju od teraz tak budem nazývať aj ja.

V tejto práci sa obmedzíme na jednoduchšiu verziu ESN:

- Žiadne priame prepojenia medzi vstupom a výstupom
- Len jednosmerné prepojenie rezervoáru k výstupu (bez feedbacku)
- Jednolevelová pamäť (pamäť ESN si uloží maximálne minulý stav rezervoáru.)
- Konštrukcia W_{out} pseudoinverziou.

Na obrázku (FIG.1) je znázornená schéma takejto siete:

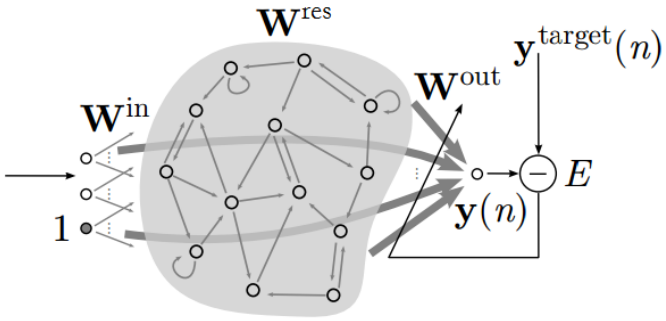

FIG. 1. Schéma zjednodušenej ESN.

III. INITIALISATION OF MODEL

Počas jednej epochy do siete vkladám postupne po jednom dáte $x_i \in \text{Uni}(-1,1)$, pričom dokopy použijem $\#\{x_i\} = 1100$ dát. Každý z nich prejde vektorom $\mathbf{W}_{in}$ do rezervoáru $\mathbf{r}_i$ obsahujúcim 100 neurónov. Rezervoár si spomenie na svoj minulý stav a aktivuje sa nasledovným spôsobom:

$$\mathbf{r}_i = \tanh(\mathbf{W}_{\text{in}} x_i + \mathbf{W}_{\text{res}} \mathbf{r}_{i-1})$$

Matice $\mathbf{W}_{\text{in}}$, $\mathbf{W}_{\text{res}}$ inicializujem ako $\mathbf{W}_{\text{in}} \in 10^{-2} \cdot \text{Uni}(-1,1)$ a $\mathbf{W}_{\text{res}} \in \text{N}(0,1)$, pričom maticu $\mathbf{W}_{\text{res}}$ ešte po tomto kroku škálujem, aby jej najväčšia eigenvalue dosiahla hodnotu, aká sa v danej chvíli hodí. Preškálovanie matice číslom C spôsobí preškálovanie jej vlastných hodnôt číslom $1/C$ (dôkaz na jeden riadok). Najväčšia eigenvalue pre $\mathbf{W}_{\text{res}} \in \text{N}(0,1)$ sa pohybuje okolo hodnoty $\rho_1 \doteq 10$, takže pokiaľ na vstupe chcem $\mathbf{W}_{\text{res}}$ s $\rho_{\max} = \rho$, preškálujem maticu hodnotou ρ/ρ_1.

Na jednotlivom výstupe $\mathbf{y}_i$ pre daný x_i budeme v tejto práci od siete požadovať vektor s dĺžkou L, ktorý hovorí o histórii *správnych* výstupov $d_{i-1}, ..., d_{i-L}$. Na základe tohto budeme počítať chybu siete ako:

$$\text{Err} = \frac{1}{L\#} \sum_{l=1,i=1}^{L,\#} (d_{i-l} - y_i(l))^2$$

kde # je počet dát, na ktorých sieť testujeme.

IV. EXPERIMENT

Po vygenerovaní # = 1100 dát z homogénneho rozdelenia si z nich vyberám prvých 100, ktoré budú slúžiť na *prečistenie siete*. Získavam teda vektor $\mathbf{r}_{100}$, s ktorým sa môže začať tréning. Trénujem na ďalších 500 dátach, čím na konci získavam stav neurónov $\mathbf{r}_{500}$ a históriu stavov $\mathbf{R}^{500\times100}$. Z histórie stavov $\mathbf{R}$ analyticky vygenerujem takú maticu $\mathbf{W}_{\text{out}}$, ktorá by ju zobrazila do správnych výstupov $\mathbf{D}$:

$$\mathbf{W}_{\text{out}} = \mathbf{D}\mathbf{R}^{+}$$

Nakoniec testujem tak,že do siete vkladám posledných 500 dát, zozbieram novú históriu stavov $\mathbf{R}$ a pomocou výstupnej matice $\mathbf{W}_{\text{out}}$ konštruujem predpovede siete:

$$\mathbf{Y} = \mathbf{W}_{\text{out}}\mathbf{R}$$

Total memory capacity

Po každom takomto experimente (pre dané L) počítam celkovú kapacitu pamäte siete (TMC) ako:

$$\text{TMC} = \sum_{l=1}^{L} \frac{\text{cov}(\mathbf{Y}_l, \mathbf{D}_l)}{\text{var}(\mathbf{Y}_l)\text{var}(\mathbf{D}_l)}$$

Typický graf pre TMC v závislosti od ρ pre fixné L vyzerá ako na obrázku nižšie (FIG.2). Zobrazených je 10 epoch pre $L = 80$.

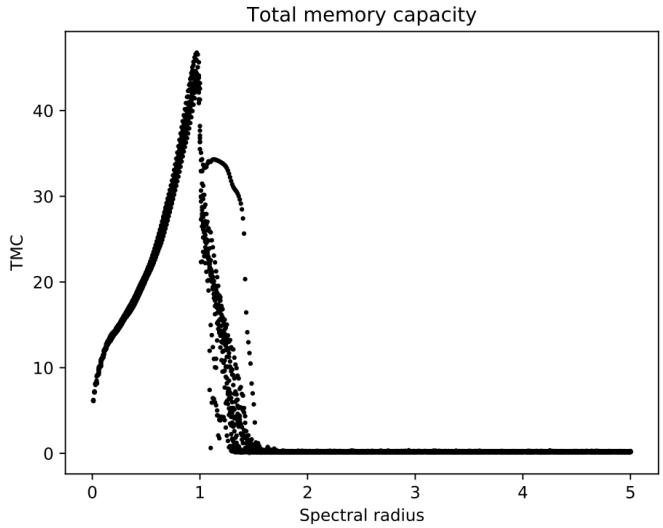

FIG. 2. Exemplárny graf správania TMC v závisloti od ρ.

Vidíme, že pre dostatočné veľké ρ padá TMC na nulu, teda schopnosť siete určovať minulosť pomocou pamäte je zanedbateľne nízka. Ďalšie grafy už nebudem vykresľovať pre zbytočne veľké eigenvalues, pretože povaha TMC v takej oblasti je zrejmá.

Seeking ideal L

Z prvého pocitu by sa mohlo zdať, že čím je väčšie L, tým lepšie (pretože sieť má šancu si toho viac zapamätať). Existuje však horné ohraničenie (dané zadaním) $L < 101$, kedže vzdialenejšia minulosť pred trénovacími dátami nie je známa. Nižšie na obrázku (FIG.3) je znázornené, ako vyzerá vývoj TMC pre rôzne hodnoty L .

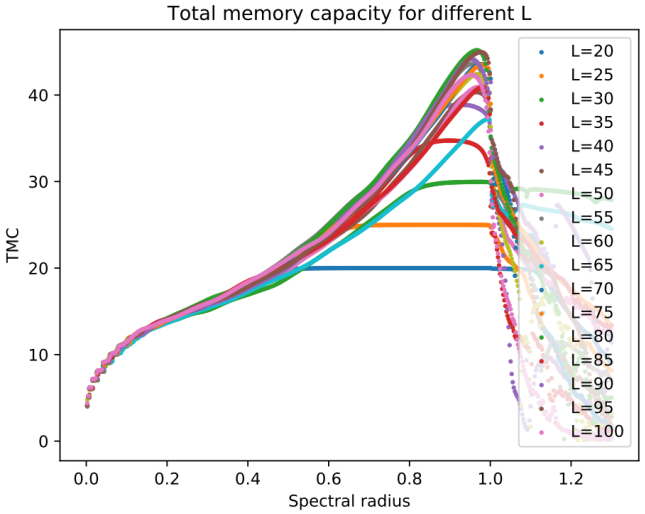

FIG. 3. Závislosť TMC(ρ) pre $L \in \{20, 25, ..., 95, 100\}$

Pri nízkych L systém dosiahne svoje TMC maximum príliš skoro a na nejakú dobu sa nechce od neho odlepiť. Pri vysokých L je evidentné, že maximum nastáva v oblasti $\rho \to 1$. Po prekročení tejto hodnoty nastáva prudký nekontrolovateľný pád.

Ked sa pozrieme na graf Erroru (FIG.4) pre rôzne L, nič nenaznačuje, ako by sme si mali vybrať to správne L:

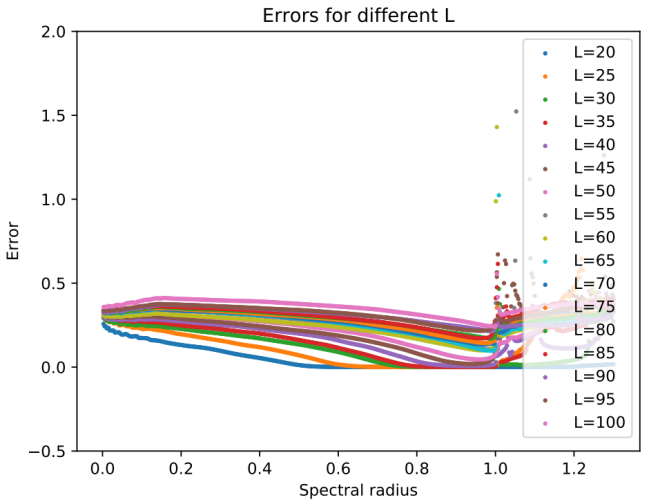

FIG. 4. Závislosť Erroru(ρ) pre $L \in \{20, 25, ..., 95, 100\}$

Zdá sa, že jediné dôveryhodné kritérium je výber

najväčšieho maxima, aké TMC kedy dosiahlo. Pre všetky ďalšie experimenty vyberám teda hodnotu $L = 95$. Na ďalšom obrázku (FIG.5) je možné vidieť opäť vývoj TMC(ρ). Okno do minulosti je už fixnuté na $L = 95$ a počet epoch vysoký= 500.

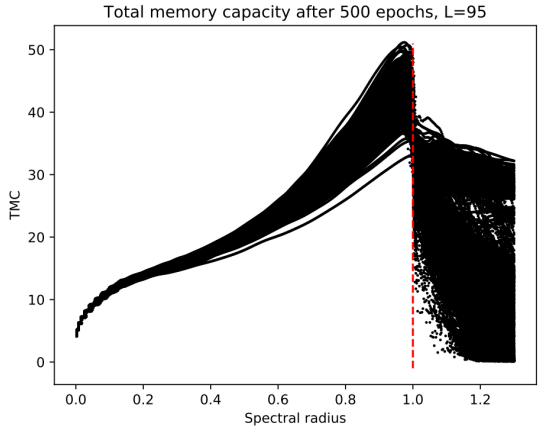

FIG. 5. Disperzia závislosti TMC(ρ) pre $L = 95$.

Pri hodnote $\rho = 1$ je jasne obrovský drop. Tento fenomén je v oblasti ML (ovzlášť pri rekurentných NN) veľmi známy, ide o *vanishing gradient problem*, kedy vďaka tanh(x) aktivačnej funkcii niektoré (error) výpočty klesajú exponenciálne k nule pre $\rho > 1$.

V. ADDITIONAL EXPERIMENTS

Logistic activation

Jednou zo zaujímavostí je zmena správania TMC(ρ), keď zameníme aktivačnú funkciu rezevoáru za sigmoidou. Krivka (FIG.6) dosahuje maximum zhruba pri $\rho \approx 7$ a to tiež s veľkou disperziou. Aj v tomto prípade je možné postrehnúť *vanishing gradient problem*, ale pokles TMC k nule nie je pre niektoré inicializácie okamžitý, ale deje sa až na rádovo väčších škálach ρ.

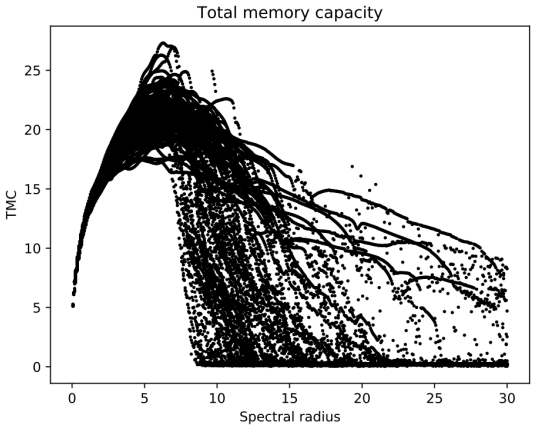

FIG. 6. Disperzia TMC(ρ) pri logistickej aktivačnej funkcii.

Reservoir sparsity

Druhou zaujímavosťou je teoretické vylepšenie výkonu siete vďaka odstráneniu niektorých synapsií v rezervoári. To znamená, že niektoré prvky matice $\mathbf{W}_{\text{res}}$ budú nulové. Na obrázku (FIG.7) možno vidieť, ako sa mení kvalita TMC pri znižovaní *sparsity* (definujem podľa Julie ako percento aktívnych nenulových prvkov v matici $\mathbf{W}_{\text{res}}$):

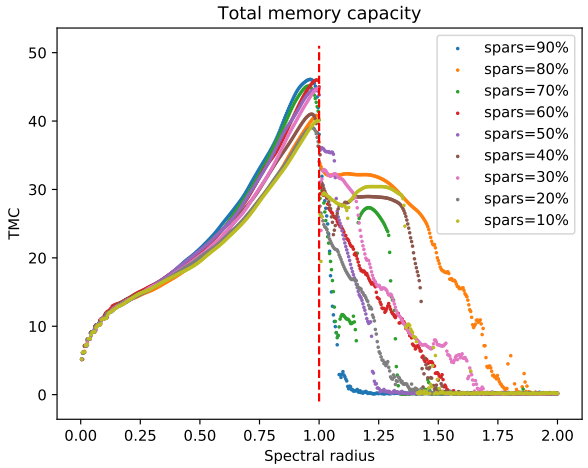

FIG. 7. TMC(ρ) pri $L = 95$ a sparsite $\{90\%, ...10\%\}$

Je jasne vidno, že globálny trend znižovania sparsity vyrába pokles kvality TMC. Pre názornosť prezentujem ešte jeden graf (FIG.8), kde porovnávam tri TMC krivky - 100%-nú, 10%-nú a 1%-nú sparsitu.

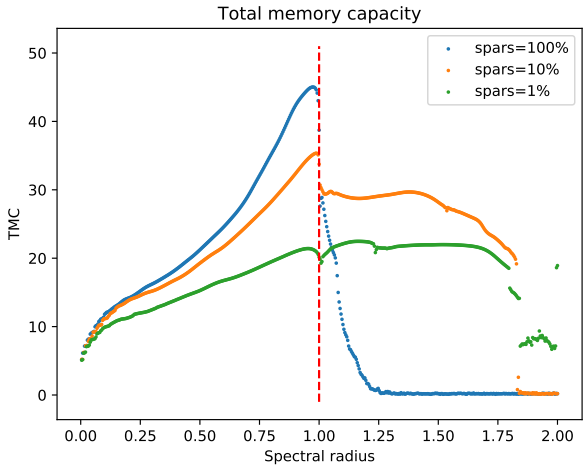

FIG. 8. 3 vybrané krivky pre extrémne hodnoty sprasity.

Zaujímavejšie by to ale mohlo byť v oblasti medzi 100% a 80%-ami, kde sa teoreticky môže nachádzať sprasitná oblasť, ktorá kvalitu TMC signifikantne zvyšuje. Tešiť sa predbežne však netreba, pretože veľká štatistika (FIG. 9) ukazuje, že stredná hodnota $\mathrm{TMC}_{\max}$ vôbec nezávisí na sparsite (v danom intervale):

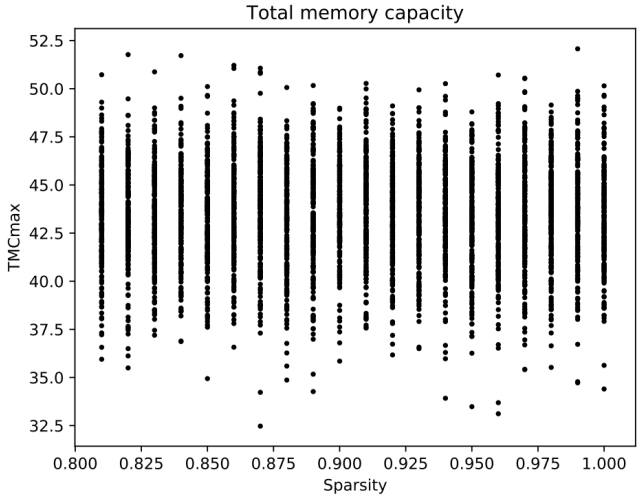

FIG. 9. Štatistika maxím TMC pre vysoké hodnoty sparsity.

MC plot

Posledná vec, ktorú v tomto projekte prezentujem, je vizualizácia príspevkov MC k TMC postupne pre každé $l = 1, 2, ..., L$. Takýto graf zároveň slúži aj ako kontrola toho, či bola správne vybratá hodnota L. Na grafe (FIG.10) vidno priebeh príspevkov k TMC v 100 epochách pre $\rho = 0.95$.

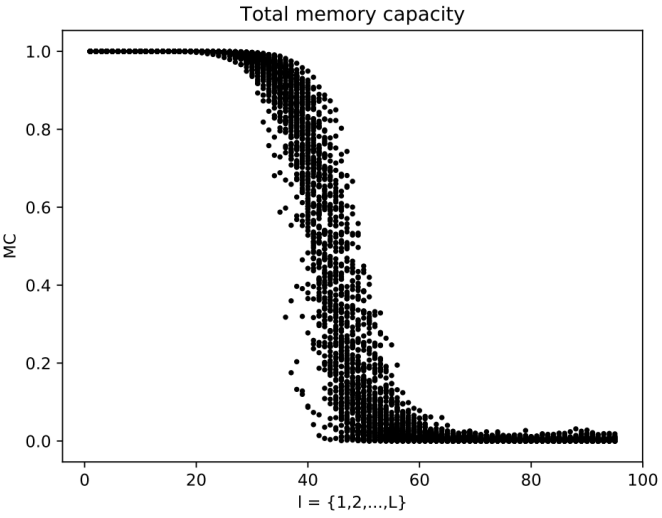

FIG. 10. Vývoj príspeckov k TMC pre $L = 95$, $\rho = 0.95$

VI. ZHRNUTIE A DISKUSIA

Záverom by som zhrnul a dodal niekoľko skutočností vyplývajúcich z predošlých analýz:

- ESN sa správa očakavateľným spôsobom, aj keď ja založená na omnoho jednoduchších predpokladoch, než pôvodne jej autori zamýšľali.

- Vďaka *vanishing gradient* efektu dosahuje sieť maximum TMC pri ρ blízkej (zľava) k 1. Disperzia maxima TMC vie byť pomerne veľká (až 25%ná odchýlka), čo môže byť spôsobené veľkým výberom L alebo malým počtom neurónov.

- Najdôveryhodnejší spôsob výberu hodnoty pre L je zrejme z grafu príspevkov MC. Z neho je vidno, že $L = 60$ by pôvodne úplne stačilo na všetky výpočty. Výber väčšieho $L = 95$ určite ale nie je chybou, pokiaľ je k dispozícii komptuačná sila.

- Ukázalo sa, že logistická aktivácia rezervoáru spôsobuje neistejšiu polohu maxima TMC a chaotickejší *vanishing gradient*. Čo je však dôležitejšie, hodnoty TMC sú výrazne menšie.

- Minimálne v mojom prípade je zrejmé, že sparsita váhovej matice rezervoáru vôbec neovyplvňuje (pokiaľ nie je výrazne nízka) výkon siete. Opäť to môže byť dôsledkom výberu vysokého L alebo nízkeho počtu neurónov.

VII. ZÁVEREČNÉ SLOVÁ A POĎAKOVANIE

Zo všetkých troch projektov som mal pocit, že pri nich treba veľa rozmýšľať, čo sa mne ako fyzikovi páčilo (alebo som na to zvyknutý). Keďže neurónové siete ako mocný tool nie je do hĺbky pochopená problematika, podobá sa práca s nimi na výskumnú činnosť, čo je minimálne v informatickej sfére netypická záležitosť.

Osobne mám pocit, že najviac som sa naučil práve pri projektoch, kedy som bol nepriamo nútený sa nad vecami naozaj zamýšľať. Týmto ďakujem hlavne Tomášovi, ktorý má (chválabohu) prepísané zadania projektov na svojej HTML, z ktorej človeku (väčšinou) bolo jasné, čo sa od neho chce. Prvý projekt bol požiadavkami asi prepísknutý, ale verím, že na základe toho sa nastavilo aj bodovanie. Druhý projekt (PCA) bol najzaujímavejší, ale didakticky nezvládnutý (iná definícia APEXu). Tretí bol fajn, k nemu nemám čo dodať.