

Generování ručně psaného textu pomocí Clockwork-RNN

Jiří Slavotínek

Abstrakt—Práce pojednává o vytvoření modelu pro generování ručně psaného textu s využitím Clockwork rekurentní neuronové sítě. Výsledky jsou porovnávány s LSTM sítí. Ukazuje se, že na málo datech je model schopný se naučit stejné poznatky jako LSTM síť a to za čtvrtinový čas. Ovšem pro velká data, již výsledek nebyl tak povzbudivý a CWRNN nedosahovala tak dobrých výsledků.

I. ZADÁNÍ

Cílem této práce je vytvořit a otestovat schopnost učení Clockwork-RNN sítě na problému generování ručně psaného textu. Problém generování textu pomocí rekurentních neuronových sítí je diskutován v [1]. Výkonost CWRNN budu měřit oproti řešení problému za pomoci stacked LSTM sítě. Budu tak moci porovnat časy učení a výkonost sítě. Pro získání informací ohledně výsledků stacked LSTM sítě využívám implementaci dostupnou na [2].

II. ÚVOD

A. Data

Data použita pro daný experiment jsou získána z [3]. Data jsou jednotlivé řádky uložené v xml souborech, které popisují tahy v řádku. Řádek je reprezentován jako množina bodů, kde u každého bodu je zaznamenán čas, v kterém daný bod vznikl, pozice, na které se bod nachází, a dále pak označení, zda v daném bodě tah končí či nikoliv. Data bylo nutné přetransformovat do podoby popsané v [2], aby výsledky byly porovnatelné. Z dat byly odstraněny mezery mezi tahy přesahující velikost 50 pixelů. Tyto mezery byly nahrazeny mezerami o velikosti 50 pixelů. Dále také byly odstraněny řádky, které obsahovali méně než 300 bodů. Body byly následně seřazeny vzestupně podle času vzniku a u každého bodu byl uložen pouze jeho offset pozice oproti předchozímu bodu a příznak 0/1 podle toho, zda tah v daném bodě končí či nikoliv.

B. Popis sítě

Architektura sítě využitá pro tento problém je přesně popsána v [1]. První vrstva sítě předává vstup, sekvenci bodů, do rekurentní neuronové sítě. Výstup z této neuronové sítě je následně chápán jako směr rozdělení, kde je pomocí směsi bivarietního rozdělení předpovídána pravděpodobnost pozice následujícího bodu a pomocí směsi exponenciálního rozdělení je předpovídána pravděpodobnost ukončení tahu v následujícím bodě.

V mém případě jsem jako rekurentní síť volil Clockwork Recurrent Neural Network, popsanou v [4]. Její princip spočívá v tom, že síť používá klasické RNN buňky, které jsou ale rozděleny do modulů, kde každý modul je aktivní pouze, když

je jeho perioda shodná s pozicí aktuálního vstupního bodu ze sekvence. Dále moduly s vyšší periodou ovlivňují výstup modulů s nižší periodou, ale ne naopak. Náhled na CW-RNN síť si můžeme prohlédnout na 1.

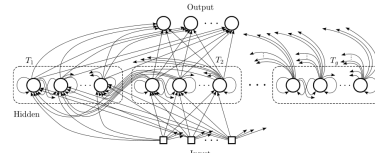


Fig. 1. Grafické znázornění struktury CWRNN sítě. Zdroj [4].

III. METODOLOGIE

Pro implementaci daného experimentu jsem využil open source framework TensorFlow [5]. Tensorflow je framework pro práci s rekurentními neuronovými sítěmi. Jeho základní verze neobsahuje implementaci CW-RNN, proto jsem využil implementaci dostupnou na https://github.com/aidangomez/models/tree/684883e4d4f310e59da109ba28cda8ba5f7785c9/clockwork_rnn. Tato implementace je ve stavu, že čeká na schválení commitu do TensorFlow. Příložené testy ovšem naznačují, že implementace je plně funkční.

Vstupem pro trénování neuronové sítě byla vždy sekvence dlouhá 300 bodů, kde se v každém bodě vytvářela směs rozdělení určující následující bod. Počet směsí generujících pozici následujícího bodu jsem zvolil 20 a pro předpovídání konce tahu jsem použil pouze 1 rozdělení. Hodnoty byly voleny shodně s hodnotami v [2]. Pro periody modulů jsem využíval mocniny dvou, tak jak je indikováno v původním článku o CW-RNN. Dále jsem také zkoušel některé vlastní hodnoty period, které jsou popsány v následující sekci.

IV. EXPERIMENTS

Síť jsem nejdříve porovnával pouze na 300 řádcích, kde jsem chtěl ověřit, že je daný problém schopna řešit stejně dobře jako LSTM pro daná data. Poté jsem testoval síť již pro celá data, konkrétně 12000 řádků.

Pro menší počet dat se ukázalo, že síť se dokáže naučit podobné věci jako LSTM síť. Na obrázcích 2 a 3 můžeme porovnat, výsledek z obou sítí. Obě sítě dávají obdobný výsledek, kde můžeme pozorovat základní dělení do slov a také naučení jednotlivých písmen. Písmena jsou ovšem silně kostrbatá a slova nevykazují žádnou podobu s anglickými slovy, na kterých byla síť učena. Obě sítě mají navíc problémy s určením pozice po konci tahu a často se tak stává, že

není viditelný jeden řádek. Při učení jsem ovšem pozoroval zajímavý fakt, že CWRNN se na stejnou trénovací chybu dostala přibližně za čtvrtinu času, než LSTM síť. To by naznačovalo potenciál této sítě pro danou úlohu, a proto byl vytvořen experiment na všech 12000 řádcích. Jako periody jsem volil mocniny dvou do 128, vyšší hodnoty podle mého názoru nemají smysl, jelikož délka vstupní sekvence je 300 bodů.

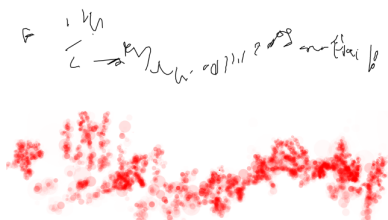


Fig. 2. CWRNN výstup, nahoře vygenerovaná sekvence, dole pravděpodobnost výskytu následujícího bodu.

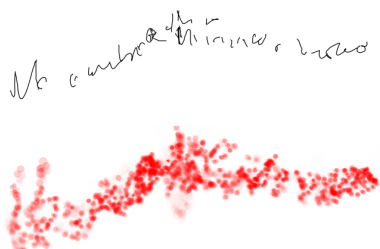


Fig. 3. LSTM výstup, nahoře vygenerovaná sekvence, dole pravděpodobnost výskytu následujícího bodu.

Pro všechna data se ovšem ukázalo, že síť nedosahuje stejných výsledků jako LSTM síť v [2]. Síť se narozdíl od sítě s méně daty, naučila udržet řádek, ovšem zhruba po 30 iteracích došlo k uváznutí v lokálním minimu a síť se již dále nezlepšovala. Síť jsem testoval s periodami mocnin 2, a také s vlastními periodami 1, 2, 4, 8, 16, 32, 64, 96, 128. Oba případy ovšem dopadly velmi podobně a náznaky častých slov jako "the" apod., diskutovány v [2], nebyly zaznamenány. Výsledek z této sítě si můžeme prohlédnout na obr. 4.

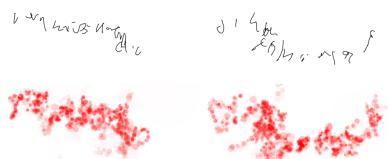


Fig. 4. CWRNN výstup při učení z 12000 řádků, nahoře vygenerovaná sekvence, dole pravděpodobnost výskytu následujícího bodu.

jsou velmi povzbuzivé výsledky, které mohou ušetřit výpočetní čas. Z ostatních publikovaných experimentů navíc také plyne, že CW-RNN je většinou dobrá náhrada za LSTM síť, s menšími nároky na výpočetní čas. Problém také může být v optimalizační metodě, která využívá Adam optimalizátor s postupným snižováním learning rate.

Zároveň by také bylo vhodné zrekonstruovat experiment pro LSTM síť, aby se ověřili prezentované výsledky.

Implementace experimentu a veškerá naměřená data jsou dostupná na <https://github.com/jirkaslavotinek/MI-ADM>.

REFERENCES

- [1] Graves A.: *Generating Sequences With Recurrent Neural Networks*, Url: <https://arxiv.org/abs/1308.0850>
- [2] Otoro: *Handwriting Generation Demo in TensorFlow*, Url: <http://blog.otoro.net/2015/12/12/handwriting-generation-demo-in-tensorflow/>
- [3] Liwicki, M. and Bunke, H.: *Handwriting Recognition of Whiteboard Notes. Proc. 12th Conference of the International Graphonomics Society*
- [4] Koutník J., Greff K., Gomez F., Schmidhuber J.: *A Clockwork RNN*, Url: <https://arxiv.org/pdf/1402.3511.pdf>
- [5] Google Research: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, Url: <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/45166.pdf>

V. ZÁVĚR

Neúspěch CWRNN pro všechna data přisuzuji špatnému nastavení parametrů jako jsou periody a počet směrů. Na malých datech byla ukázána schopnost sítě naučit se stejné poznatky jako stacked LSTM síť, ovšem za čtvrtinový čas. To