Zastosowanie sieci neuronowych do generowania tekstu

Maciej Szeptuch

Wrocław, 8 lutego 2015

1. Opis

Zastosowałem sieci neuronowe do generowania/dopełniania tekstu na podstawie podanego jako wejście prefiksu. Znalazłem, że jednym z lepszych jeśli nie najlepszym modelem do takich zastosowań są nawracające sieci neuronowe(recurrent neural networks) - przynajmniej w teorii.
W praktyce sprawiają one wiele problemów, przy wykorzystaniu SGD trzeba bardzo uważać na wszystkie parametry i współczynniki podczas nauczania, ponieważ gradient potrafi w nieoczekiwanych momentach eksplodować. W niektórych pracach jest też zastosowana metoda nazywana Hesian-Free optimization, jej główną zaletą jest to, że jest bardzo stabilna - w szczególności w porównaniu do SGD, a główną wadą to że jest nieporównywalnie wolna.

Próbowałem wykorzystać obie metody na danych z podkorpusu milionowego NKJP, konkretnie wyciąłem z niego wszystkie zdania (ok. 45000).

2. Dane

2.1. Znak po znaku

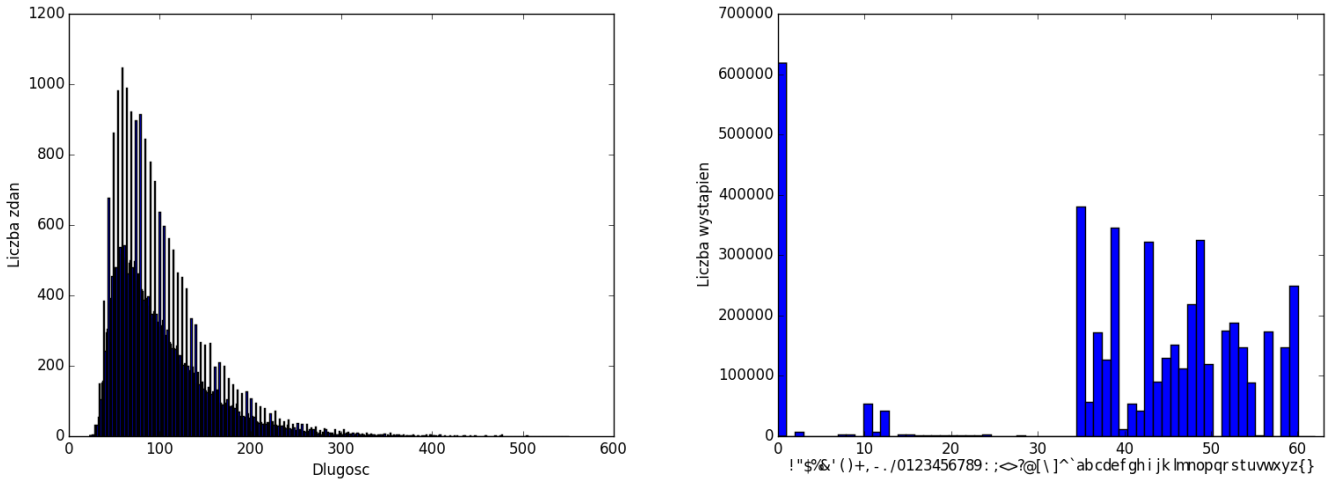
Trochę statystyk z danych:

- Liczba zdań: 45000
- Liczba różnych znaków: 63
- Długość zdań: od 23 do 551 znaków, średnio: 101 znaków

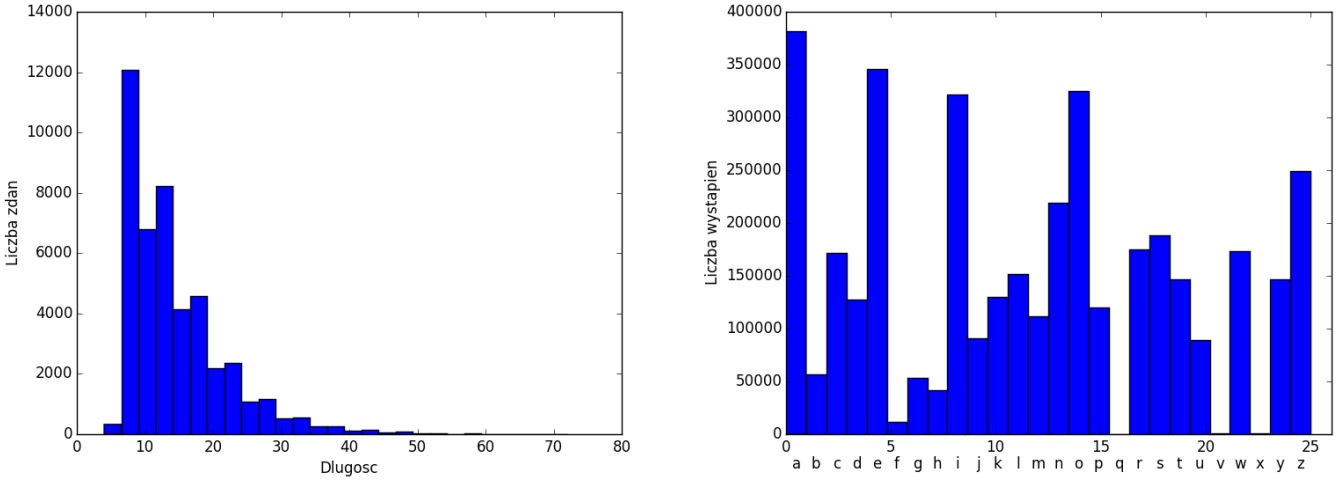
2.2. Podział na wyrazy

Trochę statystyk z oryginalnych danych:

- Liczba zdań: 45000
- Liczba słów: 100713
- Długość zdań: od 4 do 72 wyrazów, średnio: 14 wyrazów



Rysunek 1: rozkład na danych dla modelu bazującego na znakach



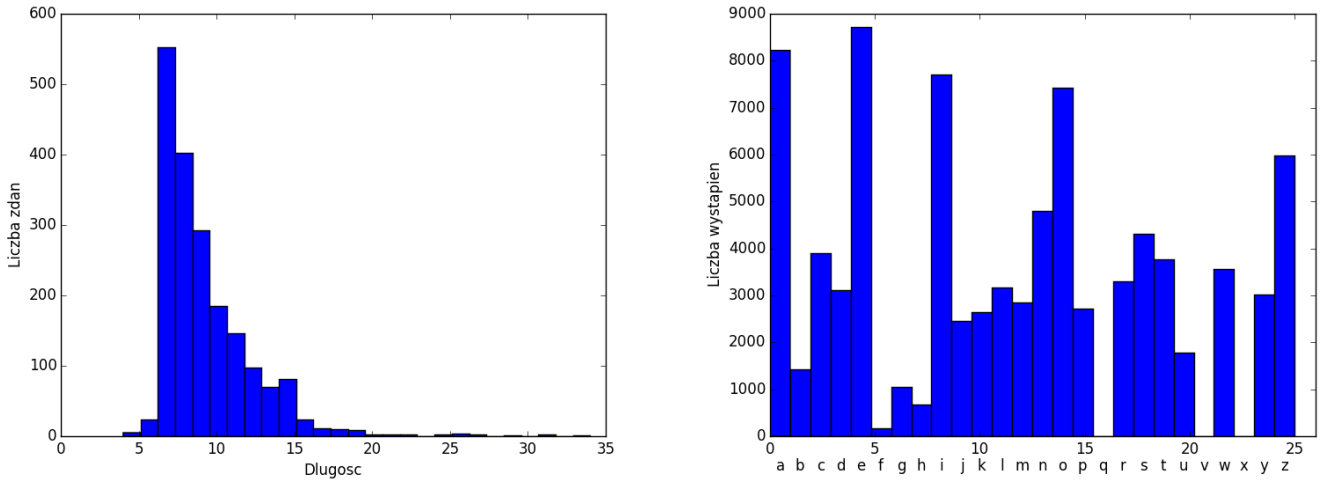
Rysunek 2: rozkład na danych dla modelu bazującego na słowach

Niestety z racji tego że to trochę za dużo różnych słów musiałem je przefiltrować żeby dało się na tym w sensownej pamięci i czasie uczyć. Wybrałem poprostu najpopularniejsze wyrazy i zdania które je zawierają. Trochę statystyk z przefiltrowanych danych:

- Liczba zdań: 1931
- Liczba różnych wyrazów: 4028
- Długość zdań: od 4 do 34 wyrazów, średnio: 9 znaków

3. Eksperymenty

Na wstępie muszę zaznaczyć, że wykorzystanie Hesian-Free optimization pomimo tego że jest stabilniejsze to przez to że potrzeba na nie dużo więcej czasu(=mocy obliczeniowej) to w tym wypadku nie dało lepszych wyników od zwyczajnego SGD.



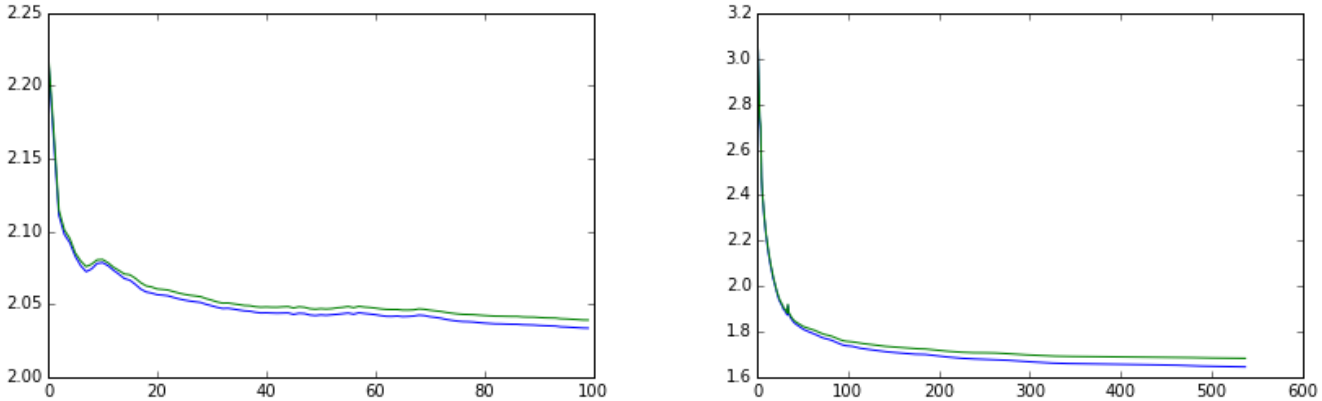
Rysunek 3: rozkład na przefiltrowanych danych dla modelu bazującego słowach

W związku z tym jedyne dane i wnioski są głównie na podstawie SGD, który przy dobrym doborze parametrów dawał szybko lepsze wyniki, i przy odrobinie szczęścia nie eksplodował.

Parametry do uczenia dobierałem robiąc przeszukiwanie binarne tak żeby znaleźć jak najlepszą szybkość uczenia i uniknąć eksplodowania gradientu. W praktyce jak miało się zepsuć to psuło się już gdzieś po maksymalnie kilkudziesięciu iteracjach na co na szczęście nie trzeba było czekać zbyt długo. Wypróbowałem dwa podejścia reprezentowania tekstu w sieci - znak po znaku oraz podział na wyrazy.

3.1. Znak po znaku

W przypadku pierwszego jest on o tyle ciekawy że pozwala on na tworzenie zupełnie nowych wyrazów, ale jak się zaraz okaże sprawia też że generowane sekwencje dla osiągniętych przeze mnie wyników nie mają większego sensu poza danymi treningowymi. Najlepszy wynik jaki udało mi się uzyskać to około 45% poprawności na danych testowych, co sprawiło że niektóre nowe tworzone wyrazy wyglądały sensownie.



Rysunek 4: nauka znak po znaku do poprawności 30% i 45%

Po kilkudziesięciu iteracjach: poprawność 30%

- to aoep solzi zdziysilzwo zie was palazy ppeatini p poeao zoe azi
  kutki wypadku odczuwac bedzie juz zawsze zlamana w kilku miejscas
- tozcin ze o pozpo pie zozzep e zwa zwo polrrio podtowa a h p
  prawa, jaka ma do niej, musi wiec nalezec do gatunku pospolitych s
- tere po inze za pooczami ptolzy pola z stwaepyasao apapiasz
  talymi towarzyszami krystyny skarbek byli: wysoki blondyn z wlosas
- teriezi zo i ao p po towznzi podezi pastazzneoe ziszi p powiaow
  tanislaw gomulka: - gospodarka polska jest calkiem mocna i kontros

Przykładowe generowane sekwencje(pogrubione dane wejściowe):
caly swiathesa a
ala ma kota a aa
nie mozna a a a

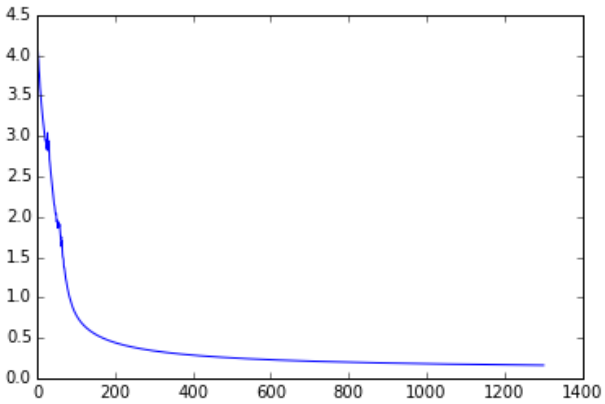
Po tysiącu iteracji poprawność 48%

- tutei pykadku bdrzycaj tidzie mez nacsze snne ani w soeku miessco
  kutki wypadku odczuwac bedzie juz zawsze zlamana w kilku miejscas
- trawa nak zagwoskaej josimsyec ciwezy do sarunku bowtrlitycz s
  prawa, jaka ma do niej, musi wiec nalezec do gatunku pospolitych s
- aana so arzyste i ooattyna iplrbu zyl systkienyondenglaniasy
  talymi towarzyszami krystyny skarbek byli: wysoki blondyn z wlosas
- aanislaw so e ki ” nlrcodarci polski pest pzlyoem mozny i control
  tanislaw gomulka: - gospodarka polska jest calkiem mocna i kontros

Przykładowe generowane sekwencje(pogrubione dane wejściowe):
caly swiatzlesoni i wuclino worn.cona. tokowacii ..lanni. lonui otalno
ala ma kotale narkowali .. woko.c. l nosc.wack. or..nacy ..nokaji ..r..
nie mozna a olatii o i.waniu molasa.. .at..ni i wascinenki . losci on

3.2. Podział na wyrazy

W tym przypadku nie wydzielałem zbioru testowego tylko starałem się jak najbardziej nauczyć sieć danych treningowych. Niestety z powodu większych danych (ok. 4000 wyrazów vs 63 litery) liczyło się trochę wolniej.



Rysunek 5: nauka po wyrazach do poprawności 70%

Po kilkuset iteracjach: poprawność 5%

- nie nie nie w w i i
  tak rzeczywiscie ci w niej do twarzy
- nie nie nie w w i
  to teraz jest to u nas
- nie nie
  wlasnie zaczal
- nie nie nie jest w w i
  tego czasu nie ma umowy o prace

Przykładowe generowane sekwencje(pogrubione dane wejściowe):
takich nie nie nie nie
nigdy nie nie nie w w
miec troche wiecej nie nie nie nie w i
uwazam ze nie nie nie nie

Po tysiącu iteracji: poprawność 30%

- bo pan jest wieku ty
  bo pan jest taki inny
- tego robic nie ma jednak o mam
  tego czasu nie ma umowy o prace
- dla kto z albo jest prawo drugie a kto prawo i raz
  wie kto z nas jest na pierwszym a kto na ostatnim miejscu
- dla czego nie wiedzial obrony jest chce w
  po prostu nie wiedzial gdzie jest jego dom

Przykładowe generowane sekwencje(pogrubione dane wejściowe):
takich dosc jak tez bylo wazne wiecej
nigdy nie tak nie wie z kobieta razem wczesniej
miec troche wiecej ze soba wszystko wie w nam gdyby kilka lat wszystko wie czym nam
uwazam ze ze lat ma kazdy do mi kilka

4. Wnioski

Wykorzystanie RNN do generowania/analizy tekstu wydaję się być całkiem sensownym pomysłem. Wydaje się że może to działać bardzo ładnie szczególnie jak zastosuje się optymalizacje HF, która niestety wymaga albo bardzo dużo czasu, albo większych mocy obliczeniowych. W moim przypadku osiągnięte z SGD wyniki nie są może świetne, ale całkiem zadowalające.