

# Rozpoznawanie języka tekstu na podstawie częstotliwości liter

Krzysztof Kutt, Michał Nowak

**Abstrakt**—TODO: Tutaj leci abstrakt

**Słowa kluczowe**—sieć neuronowa, język, rozpoznawanie.

## I. WPROWADZENIE

**S**ZTUCZNE sieci neuronowe to bardzo uproszczone modele ludzkiego mózgu. Każda z sieci składa się z setek, albo tysięcy sztucznych neuronów, stworzonych na wzór naturalnego neuronu. Sztuczny neuron zbiera sygnały wejścia (poprzez 'dendryty'), dokonuje na nich odpowiedniej transformacji i zwraca odpowiednią wartość na wyjściu ('akson'). W zależności od budowy sieci (układu neuronów) oraz od wykorzystywanej przez neurony funkcji aktywacji, wyróżniane są różne rodzaje sieci [3].

### A. Rodzaje sieci

Najważniejsze typy sieci neuronowych to sieci jednokierunkowe, sieci rekurencyjne i sieci Kohonena [8].

1) *Sieci jednokierunkowe*: W tych sieciach neurony są ułożone warstwowo. Sygnały przechodzą od wejścia do wyjścia sieci, przez wszystkie jej warstwy, bez nawrotów (bez sprzężenia zwrotnego). Rozwiązują dosyć szeroką klasę problemów.

2) *Sieci rekurencyjne*: Sieci ze sprzężeniem zwrotnym. Połączenia między neuronami stanowią graf z cyklami. Stosowane są np do rozwiązywania problemów minimalizacji, przykładowo: problemu komiwojażera.

3) *Sieci Kohonena*: Sieć, która dopasowuje swoją strukturę przestrzenną do zbioru danych (mapa). Uczona bez nauczyciela.

### B. Zastosowania sieci

Sztuczne sieci neuronowe, ze względu na fakt podawania jedynie przybliżonych wyników, nie nadają się do obliczania dokładnych wartości. Za to, dzięki 'umiejętności' uczenia się, nadają się do rozwiązywania problemów słabo określonych, do których nie jesteśmy w stanie stworzyć algorytmów. Sieć w trakcie procesu uczenia się może wykryć zależności, których nie jesteśmy w stanie dostrzec.

Przykładowymi problemami, do których sieci neuronowe są odpowiednimi narzędziami, są: rozpoznawanie pisma, przetwarzanie obrazu (np w poszukiwaniu ukrytych jednostek wojskowych, czy podejrzanych pakunków na lotniskach), prognozowanie cen.

K. Kutt i M. Nowak, Katedra Automatyki, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Akademia Górniczo-Hutnicza, Kraków, Polska, e-mail: kkutt@student.agh.edu.pl, rial@student.agh.edu.pl

### C. Rozpoznawanie języka zadanego tekstu

W niniejszym artykule przedstawiony zostaje inny, ciekawy problem: rozpoznawanie języka zadanego tekstu.

Wg projektu "Ethnologue", na świecie istnieje prawie 7.5 tysiąca języków [4]. Narzędzie, które byłoby w stanie rozpoznać z jakim językiem mamy doczynienia, byłoby bardzo użyteczne - dzięki niemu, można np ustalić jakiego tłumacza potrzebujemy. Wizją przyszłości jest stworzenie automatycznego tłumacza, który najpierw rozpoznaje mowę naszego rozmówcy, następnie rozpoznaje język, tłumaczy go, a na samym końcu syntezuje mowę, byśmy mogli w naszym ojczystym języku usłyszeć słowa rozmówcy.

Do zadania tego można podejść na wiele sposobów [1].

1) *Alfabet*: Pierwszą przesłanką może być wykorzystywany alfabet. Wiele języków posiada charakterystyczne dla siebie znaki diakrytyczne, jak np język polski i 'ogonki'.

2) *Charakterystyczne wyrazy*: W przypadku takich samych alfabetów, decyzję o języku można podjąć, zwracając uwagę na charakterystyczne dla danego języka wyrazy, czy końcówki. Duże zestawienie takich prostych reguł znajduje się w serwisie Wikipedia [5].

3) *Częstotliwość liter*: Można również ustalić częstotliwość występowania poszczególnych liter. Już krótkie spojrzenie na listę zestawów do gry w Scrabble w różnych językach [7] (ilość i wartość danej litery zależy od częstotliwości jej występowania w danym języku), pozwala na stwierdzenie, że częstotliwości te są różne w różnych językach. Nie jest jednak możliwe proste określenie reguł rozpoznawania języka na tej podstawie. Dlatego idealnym narzędziem do operowania na takich danych jest sieć neuronowa.

### D. Rozpoznawanie języka na podstawie częstotliwości liter

Niniejszy artykuł przedstawia próbę rozpoznawania języka na podstawie częstotliwości liter. Celem uproszczenia budowy programu, zrezygnowano z analizy alfabetu i zdecydowano się na analizowanie tylko i wyłącznie częstotliwości 26 liter alfabetu łacińskiego.

1) *Liczba wejść i wyjść*: Sieć składa się z 26 wejść: na każdym podawana jest częstotliwość występowania odpowiedniej litery alfabetu łacińskiego w danym tekście. Liczba wyjść zależy od założonej ilości rozpoznawanych języków. W przygotowanej aplikacji założono obsługę 11 języków.

2) *Budowa sieci*: Ze względu na charakter zadania, czyli potrzebę wykrycia odpowiednich zależności w zestawie 26 sygnałów wejściowych, zdecydowano się na wykorzystanie sieci jednokierunkowej, złożonej z trzech warstw (dodatkowa

warstwa celem usprawnienia procesu uczenia sieci). Nie istnieje żaden algorytm wspomagający podejmowanie decyzji o ilości neuronów w warstwie ukrytej [2]. Decyzja ta jest zawsze podejmowana arbitralnie i następnie weryfikowana w czasie testów. Należy tylko pamiętać, aby nie była ani zbyt duża, ani zbyt mała. Zdecydowano się przyjąć, że będzie się ona składała z 10 neuronów.

3) *Funkcja aktywacji*: Każdy neuron wykorzystuje sigmoidalną funkcję aktywacji. Pozwala ona na lepsze różnicowanie zbliżonych sygnałów niż funkcja liniowa. W rozważanym problemie, pozwala na lepsze różnicowanie zbliżonych częstotliwości występowania liter.

4) *Uczenie sieci*: Do uczenia sieci wykorzystano algorytm Resilient Propagation. Jest to zmodyfikowana wersja podstawowego algorytmu uczenia z nauczycielem sieci wielowarstwowych jednokierunkowych, czyli algorytmu propagacji wstecznej. Zdecydowano się na ten algorytm, ponieważ jest on najbardziej efektywnym algorytmem uczenia dla prostych sieci neuronowych, takich jak nasza [6].

## II. SPECYFIKACJA WYMAGAŃ

**P**ODCZAS analizy problemu zidentyfikowano dwa najważniejsze przypadki użycia aplikacji: wczytanie zbioru uczącego i trenowanie sieci, oraz wczytanie zbioru testowego i egzaminowanie sieci.

Zdecydowano się również na obsługę dodatkowych przypadków użycia, które ułatwiają korzystanie z aplikacji: modyfikację zbioru uczącego / testowego (czyszczenie, dodawanie plików) i jego zapisywanie / wczytywanie do / z pliku, modyfikację zbioru obsługiwanych języków, zapis / odczyt wyuczonej sieci neuronowej do / z pliku.

## III. IMPLEMENTACJA

**A**PLIKACJA została napisana w języku Java (wersja 1.6). Wykorzystano środowisko Netbeans IDE oraz bibliotekę Encog Machine Learning Framework.

Program składa się z trzech głównych części: obsługi plików, obsługi sieci neuronowej oraz GUI.

### A. Obsługa plików

Aplikacja wykorzystuje trzy rodzaje plików. Wszystkie pliki przechowywane są w folderze data.

1) *Pliki z tekstami źródłowymi*: Pliki o rozszerzeniu \*.txt, zawierają tekst źródłowy. Każdy tekst znajduje się w osobnym pliku. Dla każdego pliku dokonywana jest analiza częstotliwości występowania liter, przy pomocy prostego modułu, stworzonego na potrzeby przedstawianej aplikacji.

2) *Pliki zestawów danych*: Plik o rozszerzeniu \*.txt, zawierający informacje o zestawie plików z tekstami źródłowymi (do nauki/testu). W pierwszej linii zawiera liczbę  $N$  plików tekstowych. W kolejnych  $N$  liniach zawiera informacje o każdym z plików w postaci par: JĘZYK ŚCIEŻKA, gdzie JĘZYK to oznaczenie literowe języka, w jakim jest napisany dany tekst, np. PL, EN, FR (może to być dowolne słowo), zaś ŚCIEŻKA to ścieżka dostępu do danego pliku (wymagane jest, by nie posiadała białych znaków).

3) *Pliki sieci neuronowych*:

### B. Obsługa sieci neuronowej

### C. GUI

## IV. TESTY

**P**RZEPROWADZONE testy. Wyniki. Informacje o częstotliwości liter, ale też o skuteczności sieci. Wykresy? Co nam wyszło, dla których języków działa, które okazały się podobne, etc

co nie wyszło w pracy (ma być krytyczne) problemy z implementacją, z siecią, z dokumentacją itd itp

## V. WNIOSKI

**P**ROJEKT jest fajny, bądź nie i jak go można dalej rozwijać, albo gdzie go można zastosować

## LITERATURA CYTOWANA

- [1] T. Smykowski, *Jak Google rozpoznaje język tekstu?*, <http://polishwords.com.pl/blog/2009/rozpoznawanie-jezyka-tekstu/>, (dostęp: 2012-05-31)
- [2] R. Tadeusiewicz, *Elementarne wprowadzenie do techniki sieci neuronowych z przykładowymi programami*, Warszawa, Polska: Akademicka Oficyna Wydawnicza PLJ, 1998.
- [3] R. Tadeusiewicz, *Sieci neuronowe*, wyd. 2, Warszawa, Polska: Akademicka Oficyna Wydawnicza RM, 1993.
- [4] *Ethnologue*, [http://www.ethnologue.com/language\\_index.asp](http://www.ethnologue.com/language_index.asp), (dostęp: 2012-05-31)
- [5] *Language recognition chart*, [http://en.wikipedia.org/wiki/Wikipedia:Language\\_recognition\\_chart](http://en.wikipedia.org/wiki/Wikipedia:Language_recognition_chart), (dostęp: 2012-05-31)
- [6] *Resilient Propagation*, [http://www.heatonresearch.com/wiki/Resilient\\_Propagation](http://www.heatonresearch.com/wiki/Resilient_Propagation), (dostęp: 2012-05-31)
- [7] *Scrabble letter distributions*, [http://en.wikipedia.org/wiki/Scrabble\\_letter\\_distributions](http://en.wikipedia.org/wiki/Scrabble_letter_distributions), (dostęp: 2012-05-31)
- [8] *Sieć neuronowa*, [http://pl.wikipedia.org/wiki/Sieć\\_neuronowa](http://pl.wikipedia.org/wiki/Sieć_neuronowa), (dostęp: 2012-05-31)