Rozpoznawanie mówcy za pomocą sieci neuronowych

Marcin Chwedczuk Albert Skłodowski

20 Grudzień 2011

Spis treści

1 Wstęp

Celem naszego projektu było rozpoznawanie mówcy na podstawie próbek jego głosu. Rolę klasyfikatora pełnił zestaw sieci neuronowych przy czym każda z sieci była nauczona rozpoznawania dokładnie jednej osoby, oraz odrzucania pozostałych. Do ekstrakcji cech głosu zostały wykorzystane współczynniki MFCC.

W poniższym dokumencie przedstawiamy wnioski płynące z pierwszych eksperymentów ze stworzoną przez nas aplikacją. Prezentujemy uzyskane wyniki, analizujemy wpływ parametrów takich jak współczynnik uczenia czy wielkość sieci neuronowej. Opisujemy również zmiany których wprowadzenia mogłoby znacznie polepszyć jakość rozpoznawania testowanego zbioru głosów.

2 Wstępne rezultaty

2.1 Zbiór uczący i walidacyjny

Zbiór uczący i walidacyjny tworzyły próbki głosu 10 osób (9 mężczyzn oraz 1 kobiety). Na każdą próbkę składało się sześć nagrań głosu danej osoby (dalej

numerowanych liczbami od 1 do 6). W każdym z tych sześciu nagrań osoby były proszone o wymówienie kolejno następującej listy słów:

1. tlen
2. kasza
3. żyzny
4. mini
5. ćma
6. ultra
7. krew
8. house
9. felicja
10. komputer

Zbiór uczący tworzyły próbki głosu o nieparzystych numerach (1, 3 i 5), natomiast próbki o numerach parzystych (2, 4 i 6) zostały wykorzystane jako zbiór walidacyjny.

2.2 Sposób nauki sieci neuronowych

Sieci neuronowe były uczone z następującym zestawem parametrów: $\eta = 0.21$ oraz $m = 0.1$, liczbę wykorzystywanych współczynników MFCC ustawiono na 12, współczynnik znajomości na 0.5 a liczbę ramek sygnału mowy na 12. Przedstawione wartości parametrów uzyskano empirycznie na drodze testowania aplikacji.

Podczas nauki pojedynczych sieci stosowano rozpad współczynników η oraz m zgodnie z wzorami:

$$\eta_i = \frac{\eta}{\sqrt{1+i}}$$

$$m_i = \frac{m}{\sqrt{1+i}}$$

gdzie i oznacza numer iteracji.

Maksymalna ilość iteracji wynosiła 160 tysięcy. Co tysiąc iteracji sprawdzano czy błąd na zbiorze walidacyjnym jest mniejszy od 0.04, gdy błąd okazywał się mniejszy nauka sieci była przerywana.

2.3 Sposób obliczania błędu

Jako miarę błędu wykorzystano błąd średniokwadratowy, przy czym przyjmowaliśmy że sieć powinna zwracać wartość 1 w przypadku głosu którego rozpoznawania była uczona, oraz 0 w pozostałych przypadkach.

Zarówno przy obliczaniu błędu jak i przy rozpoznawaniu stosowano próg wiarygodności sieci `THRESHOLD` którego wartość wynosiła 0.75. Oznacza to że jeżeli wartość wyjściowa sieci znajduje się w przedziale $[1 - 0.75 = 0.25; 0.75]$ to uznajemy że sieć nie jest pewna swojej klasyfikacji, i wartość wyjściowa sieci

nie jest uwzględniana zarówno przy liczeniu błędu jak i przy obliczaniu wyniku klasyfikacji.
See also AlgorithmsLogic.cs: AlgorithmsLogic.HardTest_Threshold

2.4 Sposób obliczania wyniku

Dla każdej z sieci liczoną średnią arytmetyczną wartości wyjść które NIE wpadały do przedziału $[0.25; 0.75]$.
See also AlgorithmsLogic.cs: AlgorithmsLogic.GetResults_AllMaxStrategy

2.5 Przykładowy przebieg nauki

Rysunek 2 przedstawia przebieg nauki dla testowanych sieci, zauważmy że w zależności od jakości zebranych próbek oraz brzmienia głosu czas nauki może się znacznie różnić. Szczególnie interesująca jest próbka 2i, w tym wypadku nauka wydaje się być nieskuteczna jest to spowodowane najprawdopodobniej „udziwnianiem głosu" podczas zbierania próbek uczących (w tym przypadku korzystne byłoby pobranie nowej próbki z bardziej naturalnym brzemieniem głosu).

W przypadku pozostałych próbek możemy zauważyć szybsze lub wolniejsze zmniejszanie się błędu w miarę nauki sieci. Rysunek 1 przedstawia końcowy błąd na zbiorze walidacyjnym.

Osoba	Średni błąd sieci
Marcin Chwedczuk	0.0453039763400456
Albert Skłodowski	0.0704494020192417
Artur Adamek	0.0441615361895799
Adrian Sroka	0.0551230965799408
Andrzej Legucki	0.0383946258685662
Janusz Paprzycki	0.121506072702558
Julian Zubek	0.0346518049400128
Tomek Sitarek	0.0395316621522022
Michał Okulewicz	0.192809687763438
Mai Hoa Pham	0.0398912916670119

Rysunek 1: Wyniki uzyskane podczas nauki sieci neuronowych

2.6 Odpowiedź nauczonych sieci

Rysunek 3 obrazuje jak zmienia się odpowiedz sieci neuronowych w zależności od wejściowego sygnału. Może zauważyć że w tym wypadku prawie wszystkie sieci poprawnie klasyfikują dany głos wejściowy.

2.7 Skuteczność działania aplikacji

Z powodu niedostępności koleżanek i kolegów musieliśmy ograniczyć się praktycznie do badania rozpoznawania własnych głosów. W tym przypadku skuteczność rozpoznawania wynosiła około 80 procent, a w tych przypadkach w których algorytm się mylił, właściwa osoba prawie zawsze znajdowała się w pierwszej trójce wytypowanych osób. Oczywiście zaraz po zaprezentowaniu poniższego dokumentu przystąpimy to testowania aplikacji na pełnym zbiorze osób.

3 Wnioski

3.1 Możliwe przyczyny niskiej skuteczności aplikacji

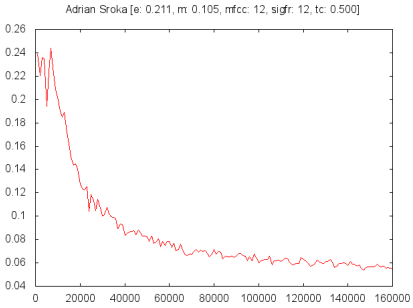
Prawdopodobnie najistotniejszym elementem naszego projektu okazała się część poświęcona wstępnemu przetwarzaniu sygnału dźwiękowego. Zgodnie z poczynionymi na samym początku założeniami, ekstrakcji cech charakterystycznych głosów poszczególnych osób dokonaliśmy za pomocą współczynników MFCC. To, na co zdecydowanie warto zwrócić uwagę, to znaczenie, jakie na wyniki ma jakość dostarczonych próbek dźwięku. Zbyt duże szumy, zakłócające właściwy głos, miały bardzo duży wpływ na obliczane wartości współczynników MFCC. Należało więc zadbać o to, by te szumy jak najlepiej odfiltrować, natomiast wzmocnić sygnał odpowiadający głosom poszczególnych osób.

Bardzo istotny okazał się sam proces zbierania próbek dźwięku, które miały służyć uczeniu sieci. Całkiem duże znaczenie miało tutaj to, by wszystkie próbki zbierać przy użyciu takiego samego mikrofonu i takiej samej karty dźwiękowej. Różnice w jakości sygnałów zbieranych za pomocą różnego sprzętu wprawdzie z pozoru nie były duże, ale powodowały trudności w dostrojeniu różnych parametrów algorytmów wstępnie przetwarzających sygnały.

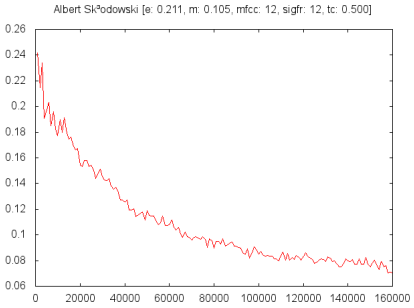
Warto również zauważyć, że nasze rozwiązanie jest podatne na próby oszustwa w postaci celowego manipulowania swoim głosem. Nasze algorytmy nie radzą sobie z rozpoznawaniem osób, które celowo zniekształcają swój głos. Sieć słabo rozpoznaje również osoby, które manipulowały swoim głosem w czasie nagrywania próbek służących jej uczeniu. Wszystko to wynika oczywiście z tego, że manipulując swoim głosem, znacząco wpływamy na wartości obliczanych współczynników MFCC. Po znalezieniu optymalnych wartości parametrów algorytmów wstępnie przetwarzających sygnał dźwiękowy, zajęliśmy się doborem odpowiednich parametrów samych sieci neuronowych.

3.2 Wpływ parametrów na działanie aplikacji

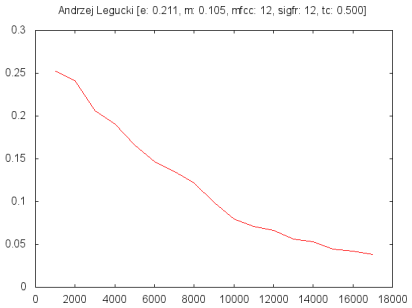
- Zwiększenie liczby ukrytych neuronów sieci neuronowych wpływa korzystnie na jakość klasyfikacji
- Liczba iteracji powyżej 100 tysięcy tylko w niewielkim stopniu polepsza jakość klasyfikacji
- Zwiększanie liczby ramek sygnału początkowo zwiększa a później zmniejsza efektywność klasyfikacji, z optimum w przedziale 6 do 12 ramek
- Ilość współczynników MFCC nie wpływa znacząco na jakość nauki jeżeli jest większa od 8
- Współczynnik znajomości powinien wynosić 0.5, w innych przypadkach przy stosowaniu biasu następuje przesunięcie w wartościach zwracanych przez sieć (to jest maksymalna wartość zwracana przez sieć jest dużo mniejsza od 1 dla współczynnika znajomości większego 0.5)
- Im dłuższa wypowiedz tym większa jest szansa prawidłowej klasyfikacji badanej osoby



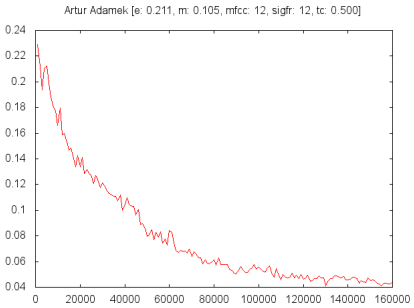
(a) Adrian Sroka



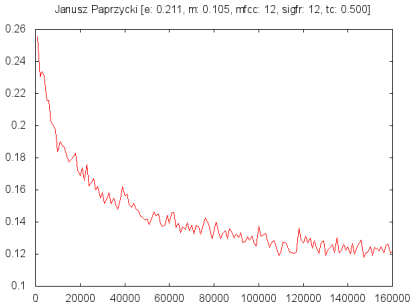
(b) Albert Skłodowski



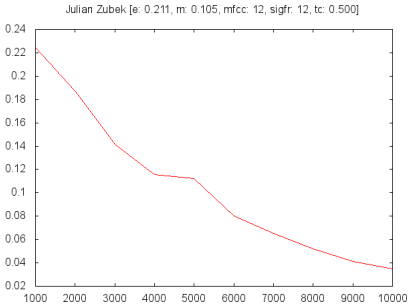
(c) Andrzej Legucki



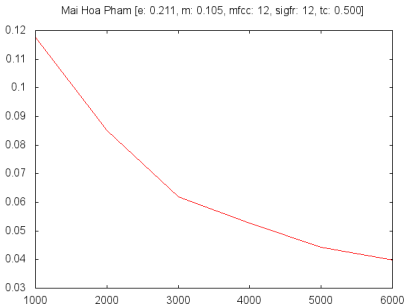
(d) Artur Adamek



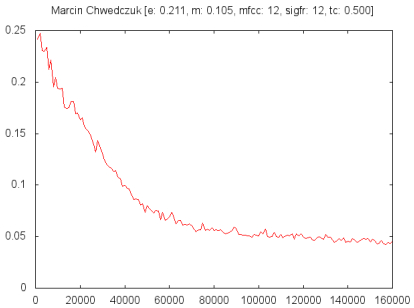
(e) Janusz Paprzycki



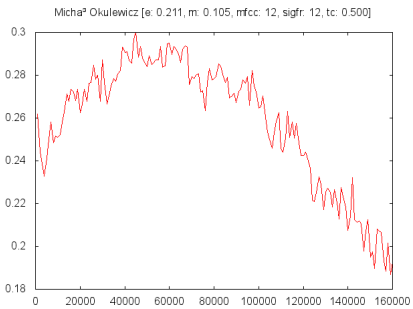
(f) Julian Zubek



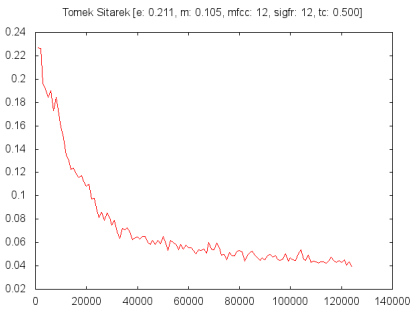
(g) Mai Hoa Pham



(h) Marcin Chwedczuk

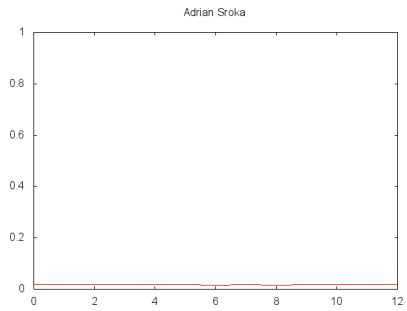


(i) Michał Okulewicz

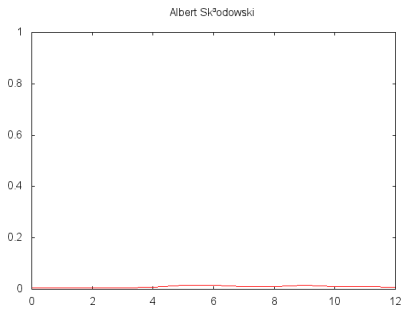


(j) Tomek Sitarek

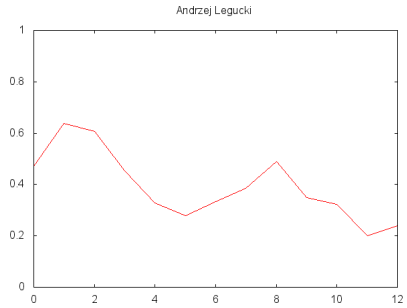
Rysunek 2: Wykresy przedstawiają wielkość błędu w zależności od liczby iteracji. Zauważmy że w przypadku niektórych sieci nauka została przerwana wcześniej, w takich wypadkach oś X została odpowiednio przeskalowana.



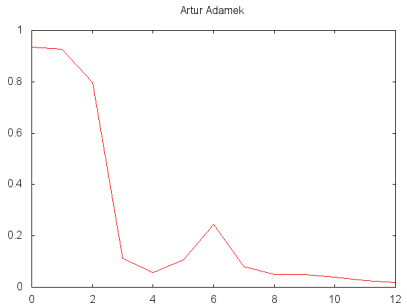
(a) Adrian Sroka



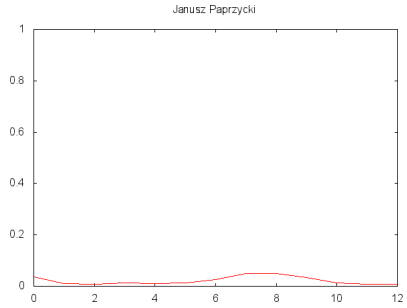
(b) Albert Skłodowski



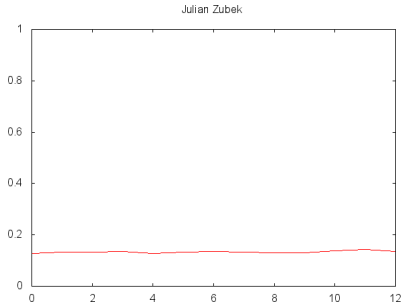
(c) Andrzej Legucki



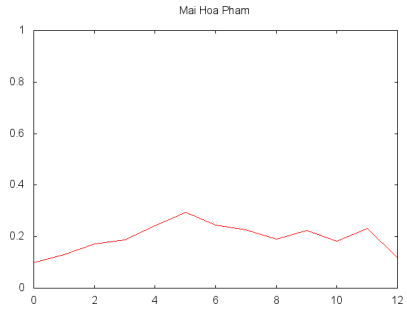
(d) Artur Adamek



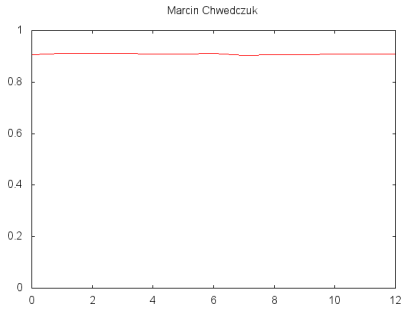
(e) Janusz Paprzycki



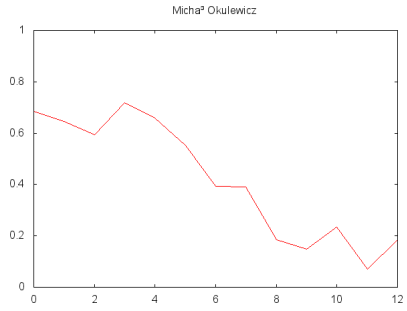
(f) Julian Zubek



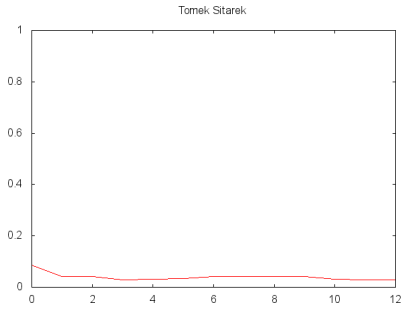
(g) Mai Hoa Pham



(h) Marcin Chwedczuk



(i) Michał Okulewicz



(j) Tomek Sitarek

Rysunek 3: Wykresy przedstawiają odpowiedzi poszczególnych sieci na frazę *Marcin Chwedczuk*. Oś X odpowiada początkowemu indeksowi zestawu 12 ramek MFCC