Sieci neuronowe – projekt nr 2

Klasyfikator osób na podstawie próbek głosu

Konspekt

Marcin Chwedczuk
Albert Skłodowski

Wstęp

Celem projektu jest stworzenie systemu identyfikacji osób na podstawie próbek ich głosu, bazującego na sieciach neuronowych. Zagadnienie leży w obszarze zainteresowań bardzo wielu naukowców na całym świecie, o czym świadczą liczne poświęcone mu artykuły i książki naukowe. Bezpośrednią przyczyną takiego stanu rzeczy są bez wątpienia bardzo liczne potencjalne sposoby wykorzystania takiej technologii. Mogłaby ona posłużyć m.in. w celu kontroli dostępu do różnego rodzaju usług, np. obsługi kont bankowych przez telefon, dostępu do poufnych danych, sterowania urządzeń za pomocą głosu jedynie przez osoby do tego uprawnione itp.

Można wyróżnić co najmniej dwa podejścia do rozpoznawania głosu. Pierwszy z nich polega na rozpoznawaniu słów pochodzących z ograniczonego ich zbioru, drugi natomiast dotyczy rozpoznawania wypowiedzi dowolnej, czyli swobodnej, z nieograniczonego zbioru słów. W naszej pracy postanowiliśmy skupić się na tym pierwszym przypadku. Oczywiście, jeśli uzyskane przez nas wyniki okażą się zadowalające, będą one stanowiły świetną podstawę do przeprowadzenia szerszych badań uwzględniających również rozpoznawanie osób na podstawie ich dowolnych wypowiedzi.

Ekstrakcja danych

Bardzo istotnym etapem działania projektowanego systemu jest zamiana próbek głosu na dane, które mają posłużyć jako wejście sieci neuronowej. Należy w tym miejscu użyć narzędzia, które pozwoli na wyekstrahowanie charakterystycznych cech głosu poszczególnych osób. Równie ważne jest ustalenie początku i końca poszczególnych wypowiedzi.

Ustaliliśmy, że ekstrakcji cech charakterystycznych głosu dokonamy za pomocą współczynników **MFCC**(ang. Mel-Frequency Cepstral Coefficients). Dokładny opis algorytmu obliczania współczynników znajduje się np. w [3], sam algorytm polega na wykonaniu następujących kroków:

1. [Opcjonalne] Wzmocnienie wysokich częstotliwości
2. Podział sygnału na ramki o czasie trwania 20 – 30 milisekund. Początki kolejnych ramek powinny być przesunięte względem siebie o 10 – 20 milisekund
3. Zastosowanie okna Hamminga do wygaszenia amplitudy sygnału na końcach ramki

4. Wykonanie transformaty Furiera algorytmem FFT, oznaczmy j-te wyjście transformaty jako fft_j

5. Oznaczmy przez e_j moduł liczby zespolonej fft_j, wektor współczynników $[e_j]$ będziemy traktować jako sygnał wejściowy dla kolejnych kroków algorytmu

6. Filtracja sygnału $[e_j]$ za pomocą banku filtrów trójkątnych rozłożonych zgodnie ze skalą Mel, niech t_k oznacza logarytm z energii sygnału skumulowanej na wyjściu filtra o indeksie k

7. Wykonanie dyskretnej transformaty cosinusowej (DCT) na współczynnikach t_k . Wyjście transformaty stanowią współczynniki MFCC

8. [Opcjonalne] Obliczenie pierwszej i drugiej pochodnej po czasie współczynników MFCC, czyli tzw. współczynników delta oraz delta-delta.

Warto zaznaczyć że zazwyczaj stosuje się od 20 do 24 filtrów trójkątnych (z punktu 6 algorytmu), oraz że wykorzystuje się jedynie 8 – 16 pierwszych współczynników MFCC uzyskanych za pomocą DCT, przy czym czasami pomija się współczynnik pierwszy reprezentujący energię sygnału.

Rozpoznania początku wypowiedzi można dokonać śledząc obliczane w kroku 5 algorytmu współczynniki e_j. Dokładniej przyjmiemy że dana ramka zawiera sygnał mowy jeżeli $\sum_j e_j \geqslant P$, P jest tutaj współczynnikiem określającym jaka musi być minimalna moc sygnału abyśmy uznali go za początek wypowiedzi.

Dane uczące i testowe

Danymi wejściowymi końcowej aplikacji będą pliki WAVE. Aby uniknąć zniekształcenia sygnału mowy, częstotliwość próbkowania będzie wynosić 44 100Hz przy 16 bitach przeznaczonych na próbkę. Sygnał będzie zapisywany w wersji MONO.

Nagrania wypowiedzi dokonamy za pomocą darmowego programu Audacity w wersji 1.2.6, przy wykorzystaniu mikrofonów komputerowych. Próbki po nagraniu nie będą przetwarzane w żaden dodatkowy sposób, ale samo nagrywanie odbywać się będzie w miejscu nie narażonym na hałas.

Każdy z plików będzie zawierał nagraną bardzo krótką wypowiedź (prawdopodobnie pojedyncze słowo) danej osoby. Nagramy pięć wypowiedzi każdej osoby, każdą z nich sześć razy. Tak uzyskane dane zostaną podzielone na dane służące do uczenia sieci oraz na dane służące do jej testowania. Każda wypowiedź znajdzie się zarówno w danych uczących, jak też testowych (odpowiednio cztery nagrania każdej wypowiedzi posłużą do uczenia, a pozostałe dwa do testowania sieci).

Zgodnie z wymaganiami postawionymi przez prowadzących przedmiot, sieć zostanie nauczona ok. czterdziestu osób (w tym co najmniej dwudziestu studentów naszego wydziału). Postaramy się zdobyć nagrania wszystkich osób wchodzących w skład naszej grupy projektowej.

Sieć neuronowa

Zdecydowaliśmy, że siecią, której użyjemy w projekcie, będzie perceptron wielowarstwowy. Decyzję tę motywujemy tym, że właśnie ten rodzaj sieci był zdecydowanie najczęściej wykorzystywany przez autorów artykułów naukowych poświęconych technikom rozpoznawania mówców. Perceptron

wielowarstwowy jest bardzo uniwersalnym narzędziem i wydaje się być dobrym rozwiązaniem przy przetwarzaniu wektorów parametrów cepstralnych.

Niech N oznacza liczbę osób które powinna rozpoznawać tworzona przez nas aplikacja. Każda z tych N osób będzie reprezentowana przez sieć neuronową net_j która rozpoznaje osobę j-tą, natomiast odrzuca pozostałe osoby.

Wejście sieci net_j stanowią wektory MFCC oraz opcjonalnie współczynniki delta z S kolejnych ramek sygnału mowy (S, oraz liczba współczynników MFCC będzie ustalona eksperymentalnie). Wyjście sieci stanowi pojedyncza liczba z przedziału [0, 1] która reprezentować będzie prawdopodobieństwo tego że podana wypowiedz należała do j-tej osoby.

Ze względu na wysoki stopień trudności zadania każda sieć net_j będzie siecią trójwarstwową, o pełnych połączeniach pomiędzy poszczególnymi warstwami. Liczba neuronów w warstwach ukrytych będzie zależała w pewien sposób od wielkości wejścia, ale będzie od niego znacznie mniejsza. Na przykład przy wykorzystaniu 8 współczynników MFCC bez współczynników delta, oraz przyjęciu S równego 10 mamy wejście składające się z 80 liczb rzeczywistych. Dla tego przykładu pierwsza warstwa ukryta może posiadać około 10 neuronów, druga warstwa ukryta 5 neuronów, dodatkowo sieć posiada 1 neuron wyjściowy.

Zastosowaną funkcją aktywacji będzie funkcja sigmoidalna. Natomiast do nauki sieci zostanie wykorzystany klasyczny algorytm wstecznej propagacji błędu. Współczynnik szybkości nauki oraz momentu dla algorytmu backprop zostaną dobrane eksperymentalnie. Dodatkowo będziemy wykorzystywali *współczynnik znajomości T* będący liczbą z przedziału [0, 1]. Poszczególne sieci będą uczone według następującego algorytmu:

1. Niech L_j reprezentuje zbiór próbek uczących dla j-tego mówcy, T – współczynnik znajomości, K – numer osoby którą uczymy się rozpoznawać
2. Stwórz nową sieć neuronową net_K
3. Powtarzaj aż do uzyskania odpowiednio małego błędu na zbiorze testowym kroki 4 - 6
4. Niech r będzie liczbą wylosowaną z rozkładem jednostajnym z przedziału [0, 1]
5. Jeżeli $r < T$ to nauczaj sieć net_K na losowo wybranej próbce ze zbioru L_K, jako oczekiwaną odpowiedź sieci przyjmij 1
6. W przeciwnym przypadku losowo wybierz mówcę różnego od K oznaczmy go L, nauczaj sieć net_K na losowo wybranej próbce ze zbioru L_L, jako oczekiwaną odpowiedź sieci przyjmij 0

Współczynnik T powinien zawierać się w przedziale [0.2, 0.6]

Opis aplikacji

Stworzona przez nas aplikacja umożliwi identyfikację mówcy. Aby zacząć korzystać z aplikacji użytkownik będzie musiał podać dane poszczególnych osób, na pojedyncza encje danych będą składały się pliki WAVE zawierające próbki głosu, dane identyfikujące osobę (np. imię i nazwisko) oraz opcjonalnie zdjęcie.

Po dodaniu wystarczającej liczby osób użytkownik będzie mógł uruchomić proces uczenia, opcjonalnie korygując jego parametry. Użytkownik będzie miał możliwość zmiany stałej uczącej i momentu algorytmu wstecznej propagacji błędów, oraz wcześniej omawianych parametrów takich jak ilość współczynników MFCC czy współczynnika znajomości. Będzie istniała również możliwość ustawienia liczby neuronów warstw ukrytych sieci, oraz maksymalnej liczby cykli nauki.

Po zakończeniu nauki aplikacja wyświetli krótkie podsumowanie zawierające między innymi wartości błędów poszczególnych sieci na zbiorze testowym.

Po wyświetleniu podsumowania nauki aplikacja przejdzie w tryb rozpoznawania, w tym trybie aplikacja będzie nasłuchiwać w czasie rzeczywistym wejścia karty dźwiękowej. I natychmiast po zebraniu odpowiedniej liczby użytecznych (tj. zawierających mowę) ramek przystąpi do identyfikacji osoby mówiącej i prezentacji wyników. Użytkownik wybierając odpowiednią opcję będzie mógł powrócić do trybu rozpoznawania.

Wynikiem identyfikacji będzie lista zawierająca tożsamość N najbardziej prawdopodobnych mówców, wyświetlona w postaci danych identyfikacyjnych lub zdjęć (jeśli zostały dostarczone).

Uwagi techniczne

Aplikacja zostanie napisana w języku C# w wersji 4.0, a więc do poprawnego działania będzie wymagała zainstalowania w systemie użytkownika biblioteki Microsoft .NET Framework w wersji co najmniej 4.0.

Do przetwarzania plików WAVE wykorzystamy bibliotekę NAudio [5], do wykonywania obliczeń numerycznych np. FFT zastosujemy bibliotekę ALGLIB [6]. Sieć neuronowa zostanie zaimplementowana za pomocą biblioteki Encog [7].

Cele

1. Dokonać klasyfikacji czterdziestu mówców ze skutecznością powyżej 80%
2. Zbadać skuteczność rozpoznawania w zależności od ilości współczynników MFCC, współczynników delta oraz architektury sieci neuronowej; Rozważyć zastosowanie innych metod pozwalających na ekstrakcję cech w przypadku osiągnięcia złych wyników podczas stosowania MFCC
3. Ewentualnie określić przyczyny nie uzyskania zadowalających wyników

Bibliografia

1. Brian J. Love, Jennifer Vining, Xuening Sun: *Automatic speaker recognition using neural networks*.
2. Lindasalwa Muda, Mumtaj Begam,I. Elamvazuthi: *Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques*.
3. Todor Ganchev, Nikos Fakotakis, George Kokkinakis: *Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task*
4. Audacity, http://audacity.sourceforge.net/

5. NAudio, http://naudio.codeplex.com/

6. ALGLIB, http://www.alglib.net/

7. Encog, http://www.heatonresearch.com/encog