Biologicznie Motywowane Metody Sztucznej Inteligencji

Raport projektu

Data oddania projektu: 26 czerwiec 2013

Temat ćwiczenia:

**Zaprojektować i zaimplementować program uczący prostą sieć neuronową metodą wstecznej propagacji błędu, który zostanie przetestowany na przykładzie systemu do rozpoznawania pewnych kategorii na podstawie opisu symbolicznego jego cech.**

**Skład sekcji:**

1. Marcin Kolny(marcin.kolny@gmail.com)

# Cel projektu

Celem projektu było stworzenie aplikacji umożliwiającej rozpoznawanie kształtów odczytanych z obrazów binarnych, przy użyciu sieci neuronowej. Należało zaimplementować zestaw klas reprezentujących sieć neuronową, a także metodę uczącą wstecznej propagacji błędów.

# Topologia sieci neuronowej

Sieć neuronowa, jakiej użyłem, ma 4 warstwy, o kolejnej liczbie neuronów: 3, 4, 5, 3. Topologię tę wyznaczyłem w następujący sposób:

* Najpierw sprawdziłem działanie sieci dla dwóch warstw(wejściowej oraz wyjściowej), która nie dawała zadowalających efektów.
* Następnie dołożyłem kolejną warstwę sieci(ukrytą), jednak również nie otrzymywałem dobrych rezultatów
* Próbowałem użyć sieci o większej liczbie warstw(8 ora 10), jednak i te warianty nie dawały wyników, jakie mogłyby mnie zadowolić.
* Wróciłem do mniejszej ilości warstw – użyłem dwóch warstw ukrytych, co okazało się poprawną topologią dla mojego problemu.

Ponadto przy każdym z powyższych punktów próbowałem różnych kombinacji liczby neuronów w poszczególnych warstwach.

## Liczba neuronów warstw zewnętrznych

Liczba neuronów warstw zewnętrznych jest podyktowana specyfiką zadania, do jakiego sieć została przygotowana. Kształty obrazów rozpoznawane są przy użyciu trzech parametrów: okrągłości, wypukłości, oraz kwadratowości. Są to więc wejścia sieci neuronowej.

Wyjścia sieci to natomiast kształt, jaki został wykryty przez sieć. Aktualnie sieć rozpoznaje trzy kształty: koło, kwadrat, oraz krzyż. W zależności od tego, na jakim wyjściu pojawi się wartość **1**, taki kształt został rozpoznany.

# Implementacja

## Język programowania

Cały projekt został napisany w języku C++.Użyłem też środowiska Microsoft Visual Studio 2010, jednak projekt powinien z powodzeniem skompilować się w dowolnym kompilatorze implementującym standard C++11.

## Dodatkowe biblioteki

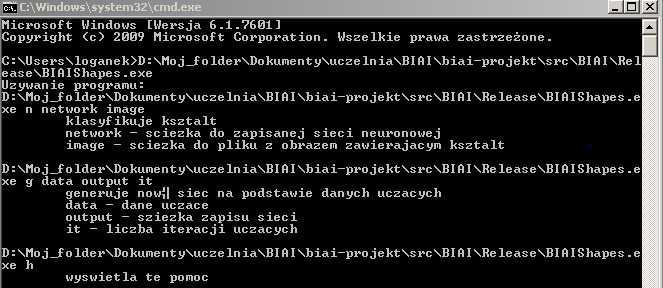
W projekcie skorzystałem z zewnętrznej biblioteki OpenCV w wersji 2.5. Użyłem jej do wyciągania wartości opisujących cechy charakterystyczne kształtów binarnych.

# Dane uczące

Dane uczące zostały wygenerowane przy użyciu programu napisanego w Adaptive Vision Studio - źródła tego programu oraz dane uczące zostały zamieszczone w załączniku(katalog avs-application).

# Interfejs użytkownika

W programie wykorzystano tekstowy interfejs użytkownika, podając do programu parametry wykonania. Poniżej zamieszczam zrzut ekranu pomocy programu.



# Testy programu

Na początku wygenerowałem sieć neuronową, na podstawie danych uczących. Następnie w edytorze obrazów rysowałem poszczególne kształty, i przesyłałem je do programu. Poniżej zamieszczam kilka rezultatów, jakie zwrócił program dla zadanych obrazów.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# Wnioski

Opisywany projekt nie jest moim pierwszym kontaktem z sieciami neuronowymi. Wcześniej implementowałem sieci neuronowe, jednak nigdy nie łączyłem tego z wizją komputerową. Podczas gdy poznawałem bibliotekę OpenCV, wymyśliłem kilka ciekawych zastosowań dla sieci neuronowych oraz wizji komputerowej.   
Sieć, którą zaimplementowałem, chciałbym wykorzystać między innymi na zawodach IMAV(<http://www.imav2013.org/>), których celem jest rozpoznawanie kształtów przy użyciu kamery zamieszczonej na obiekcie latającym.

# Załączniki

* Biai\_shapes.zip - archiwum zawierające wszystkie pliki źródłowe

# 