Sprawozdanie 3 - Metody Inteligencji Obliczeniowej

Algorytmy Genetyczne

Krzysztof Kostecki

**Wprowadzenie**

Algorytmy genetyczny jest metodą rozwiązywania problemów optymalizacyjnych która bazuje na naturalnej selekcji, procesie który możemy spotkać w przyrodzie. Algorytm w kolejnych iteracjach modyfikuje populacje a potem porównuje z funkcją dopasowania (ang. Fitness Function). Różnice pomiędzy algorytmem genetycznym i klasycznym podejściem do rozwiązania problemu optymalizacji są podsumowane w poniższej tabeli.

|  |  |
| --- | --- |
| **Klasyczny algorytm** | **Algorytm Genetyczny** |
| Generuje pojedynczy punkt podczas każdej iteracji. Kolejne punkty zbliżają nas do optymalnego rozwiązania | Generuje populacje punktów podczas każdej iteracji. Najlepszy osobnik z populacji zbliża nas do optymalnego rozwiązania. |
| Wybiera następny punkt w sekwencji poprzez deterministyczne obliczenia | Kolejne populacje są generowane przy wykorzystaniu generatora liczb pseudolosowych. Algorytm niedeterministyczny. |

Wymagane cechy funkcji aktywacji to:

* ciągłe przejście pomiędzy swoją wartością maksymalną a minimalną (np. 0 - 1)
* łatwa do obliczenia i ciągła pochodna
* możliwość wprowadzenia do argumentu parametru do ustalania kształtu krzywej

Najczęściej stosowaną jest funkcja sigmoidalna zwana też krzywą logistyczną ( przyjmuje ona wartości pomiędzy 0 a 1 ). Inne używane funkcje nieliniowe to:

* tangens hiperboliczny (wartości pomiędzy - 1 a 1)
* sinusoida i cosinusoida (ograniczone do odcinków odpowiednio 1,5 Pi do 2,5 Pi oraz Pi do 2 Pi przyjmując wartości pomiędzy 0 a 1)

Rozróżniamy kilka rodzajów neuronów:

* warstwy wejściowej
* warstw ukrytych
* warstwy wyjściowej

Pojemność informacyjna pojedynczego neuronu nie jest duża. Powiększenie pojemności i zwiększenie zdolności przetwarzania uzyskuje się poprzez odpowiednie połączenie wielu neuronów. Powstaje wówczas sieć neuronowa.

**Mała ilość dostępnych przykładów**

Podczas uczenia stosujemy podział na zbiór uczący i testujący gdzie ten drugi jest zazwyczaj znacznie większy, pozwala nam to ocenić jak sieć została nauczona. Pomocne zwłaszcza w przypadku przeuczenia sieci, taka sieć reagowała by perfekcyjnie na wektor uczący ale podawała by błędne dane dla wektora testującego. W sytuacji kiedy mamy mało dostępnych przykładów stosujemy metody pozwalające na podział dostępnego zbioru. Najczęściej stosowaną jest metoda k-krotnej walidacji krzyżowej (kroswalidacji). Polega ona na losowym podziale początkowego zbioru przykładów na k możliwie równe wzajemnie niezależne części S1, S2 ... Sk.

k-1 części stanowi zbiór treningowy, k-ta część stanowi zbiór testowy- klasyfikator konstruujemy k-krotnie(otrzymujemy k-klasyfikatorów). Inną popularną metodą szacowania klasyfikatora dla mało licznego zbioru przykładów jest n-krotna walidacja krzyżowa(ang. Leave-one-out cross validation) lub wielokrotne próbkowanie(ang. Bootstraping)

**Wyniki**

W realizacji ćwiczenia ponownie wykorzystywaliśmy oprogramowanie MATLAB oraz dostępny tam pakiet Neural Network Toolbox.

Zadanie polegało na aproksyksymacji funkcji



Przy wykorzystaniu wielowarstwowej sieci neuronowej. Uczenie takiej sieci polega na przekazywaniu jej par wartości wejściowych *(x,y)* oraz oczekiwanej dla nich wartości *z.*

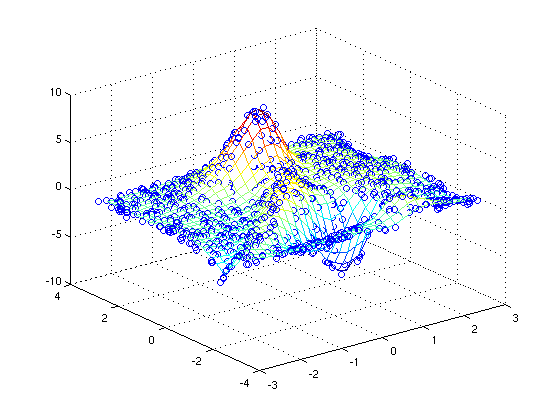
Adptacja wag odbywa się na podstawie różnicy pomiędzy aktualnym wartości z na wyjściu a oczekiwanej.

Efektywność otrzymanej sieci należało zbadać na wygenerowanym zbiorze testującym o liczebności 10 razy większej niż zbiór uczący.

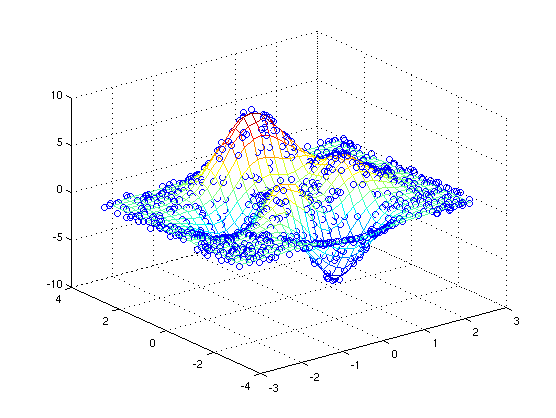
Podczas eksperymentu testowane były różne wariantów sieci. Zmiennymi parametrami były ilość warstw oraz ilość neuronów w danej warstwie. Każda konfiguracja została poddana uczeniu 3 razy, następnie została pomijana największa wartość błędu średnio-kwadratowego natomiast z pozostałych wartości została wyciągana średnia. Taka metodologia badania była wymuszona przez losowość wag początkowych. Przez co w niektórych przypadkach proces uczenia zatrzymywał się w minimum lokalnym więc wynik nie był miarodajny.

Najlepsze rezultaty zostały uzyskane dla sieci o dwóch warstwach ukrytych z ilością neuronów 6 dla pierwszej i 3 dla drugiej. Wyniki widoczne na wykresach poniżej.

Błąd średnio-kwadratowy dla tej konfiguracji wyniósł *1.002*



Obraz 1: Wynik aproksymacji dokonanej przez sieć dla danych testowych



Obraz 2: Wykres dla oczekiwanych wyników dla danych testowych

W drugim zadaniu należało dla gotowego zbioru danych dokonać klasyfikacji. Sieć wymagała 3 wyjść. Dla sieci o układzie (6,3) i funkcji aktywacji hardlim uzyskano błąd średnio-kwadratowy na poziomie 1,223.

**Wnioski**

W wyniku testowania różnych konfiguracji sieci neuronowych uzyskaliśmy wynik aproksymacji który jest bardzo zbliżony do oczekiwanych danych. Co jest warte zauważenia zwiększenie liczby neuronów w warstwach nie sprawiło że nasza aproksymacja była bardziej dokładna, natomiast sam proces uczenia trwał znacznie dłużej. Mieliśmy do czynienia z typowym przykładem przeuczenia sieci która źle reagowała na dane testowe.