Dokumentacja projektu z przedmiotu:

Sztuczne sieci neuronowe

Temat projektu:

Zastosowanie sieci neuronowej do gry w kółko i krzyżyk

Autorzy:

Andrzej Węgrzyn

Justyna Michalik

Paweł Noga

Marcin Komisarz

Data wykonania:

xxx

1. Wstęp
   1. Cel projektu

TODO

* 1. Zasady gry w kółko i krzyżyk

TODO

1. Rozwiązanie problemu
   1. Przygotowanie algorytmu generującego dane uczące

TODO

1. Test sieci
   1. Przygotowanie zestawu danych

Na podstawie wcześniej przedstawionego algorytmu wygenerowano zestaw danych uczących oraz testujących.

Należy przy tym zauważyć, że generowanie tychże zbiorów było bardzo czasochłonne i wygenerowanie zestawu składającego się z 5000 przykładowych plansz oraz ich kolejnych ruchów trwało kilka godzin, dlatego przygotowano następujące zbiory danych

* Rozmiar zbioru uczącego: **5000**
* Rozmiar zbioru testowego: **100**
  1. Test sieci

W ramach projektu przetestowano 2 rodzaje sieci:

* Sieć jednokierunkowa wielowarstwowa
* Sieć RBF

Kolejnym krokiem był dobór wyjścia sieci spośród 2 następujących koncepcji:

1. W warstwie wyjściowej 2 neurony. Wyjście neuronu nr 1 określa współrzędną punktu X na planszy, z kolei wyjście neuronu nr 2 określa współrzędną Y. (problem aproksymacji).

Np.

Dla takiej odpowiedzi neuronów X = 2, Y = 3.

1. W warstwie wyjściowej 5 neuronów. Indeks pobudzonego neuronu w warstwie wyjściowej określa współrzędną. W takim wariancie potrzebujemy 2 sieci, gdzie pierwsza daje odpowiedź dla współrzędnej X, natomiast druga dla współrzędnej Y. (problem klasyfikacji).

Np. sieć dla Wsp. X sieć dla WSP. Y

Dla takiej odpowiedzi neuronów X = 5, Y = 2.

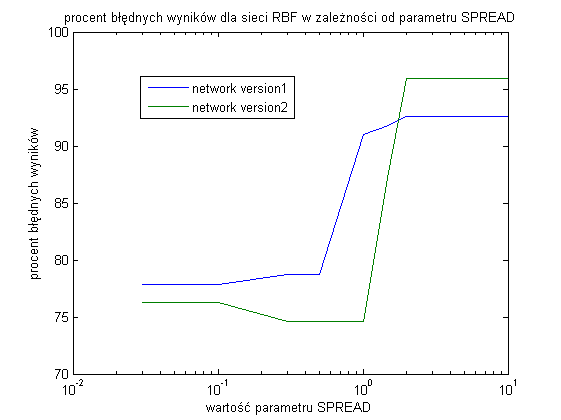
* + 1. Sieć RBF

**Test SPREAD**

Dla tego typu sieci, przede wszystkim należało przetestować różne wartości parametru SPREAD.

W tym przypadku zastosowano zestaw 2000 danych uczących dla przyspieszenia obliczeń oraz zestaw 100 danych testowych.

W wyniku symulacji otrzymano następujący wykres przedstawiający zależność błędu sieci od parametru SPREAD.



Jak widać, najbardziej optymalna wartość parametru SPREAD dla obu typów sieci RBF wynosi około 0.1.

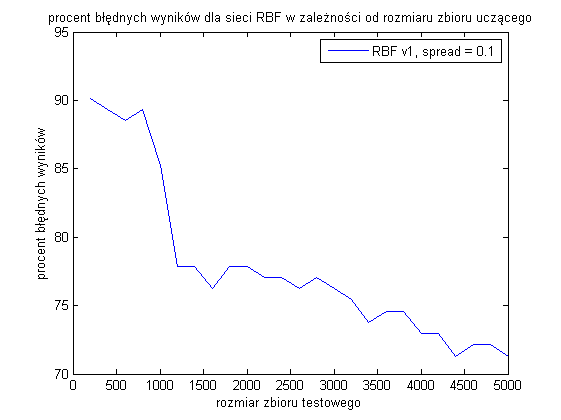
Oprócz tego widzimy nieznaczną przewagę sieci z wariantem konfiguracji warstwy wyjściowej nr 2.

**TEST LICZEBNOSCI ZBIORU UCZĄCEGO**

W tym kroku zbadano zależność błędu sieci od liczebności zbioru uczącego. Jako parametr SPREAD dla sieci zastosowano wartość równą 0.1.

Test przeprowadzono dla sieci z wariantem warstwy wyjściowej nr 2. (Test dla sieci z wariantem nr 1 pominięto ze względu na czas obliczeń, jako że dla obu sieci spodziewamy się podobnych wyników w tej kwestii).

W wyniku symulacji otrzymano następujący wykres:



3.2.1.1 Wykres dokładności sieci RBF od rozmiaru zbioru uczącego

Jak widać na podstawie wykresu błąd sieci maleje wraz ze wzrostem liczebności zbioru uczącego. Największą różnicę widać w przedziale 1000 a 1500, gdzie obserwujemy zmniejszenie błędu sieci o około 10%.

W przedziale 1500 – 5000 różnica wynosi około 5%.

Dodatkowo możemy wywnioskować, że kolejne zwiększanie rozmiaru zbioru uczącego przyniesie lepsze rezultaty, aczkolwiek ze względu na ograniczenia sprzętowe nie jesteśmy w stanie znacząco zwiększyć tego zbioru.

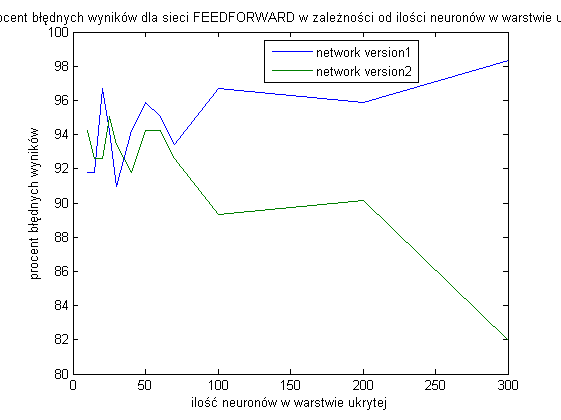
* + 1. Sieć Feedforward (jednokierunkowa wielowarstwowa).

**TEST LICZBY NEURONÓW W WARSTWIE UKRYTEJ**

W tego typu sieci należało przetestować różne liczby neuronów w warstwie ukrytej.

Zastosowano zbiór uczący równy 2000, oraz przetestowano 2 konfiguracje warstwy wyjściowej tak jak w poprzednim przykładzie.

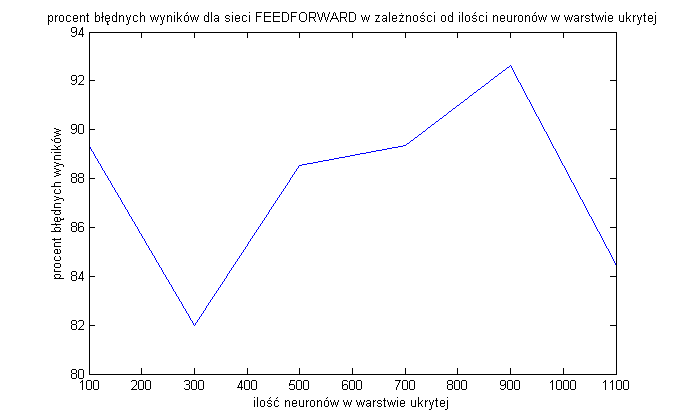
W wyniku symulacji otrzymano następujący wykres:



Jak widać na wykresie, zastosowanie większej liczby neuronów dało pozytywny skutek jedynie dla sieci z konfiguracją warstwy wyjściowej w wersji nr 2.

Różnica w zastosowaniu 50 neuronów a 300 neuronów wynosi około 10%. Mimo wszystko widzimy, że błąd sieci wciąż jest niezadowalający.

Na podstawie wykresu można również zauważyć, iż sieć w wariancie nr 2 daje coraz lepsze rezultaty dla większych ilości neuronów w warstwie ukrytej, dlatego postanowiono przetestować tą sieć z większą liczbą neuronów, co przedstawia z kolei następujący wykres:



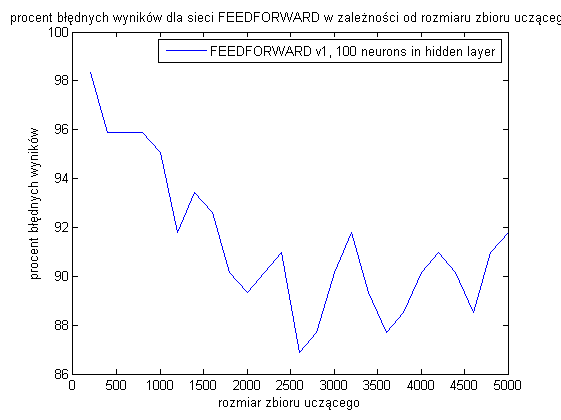
Analizując wykres widzimy jednak, że dokładność sieci nie zwiększa się w nieskończoność wraz z liczbą neuronów w warstwie ukrytej.

Można stwierdzić, że najbardziej optymalna liczba neuronów w warstwie ukrytej wynosi około 300.

Podsumowując test należy dodać, że stosowanie kolejnych warstw ukrytych z różną liczbą neuronów przynosiło jeszcze gorsze efekty.

**TEST LICZEBNOSCI ZBIORU UCZĄCEGO.**

Dla tego testu zastosowano sieć jednokierunkową z konfiguracją warstwy wyjściowej w wersji nr 2, która dała lepsze rezultaty w poprzednim teście, oraz liczbę neuronów w warstwie ukrytej ustalono na 100 (w celu przyspieszenia obliczeń).



Analizując wykres zależności dokładności sieci od rozmiaru zbioru uczącego można zauważyć, że zwiększanie zbioru daje pozytywny efekt jedynie do rozmiaru około 2500, natomiast dalsze zwiększanie tegoż zbioru o dziwo nie przynosi żadnych korzyści.

* 1. Podsumowanie testów sieci.

Podsumowując przeprowadzone testy dla sieci RBF oraz FeedForward możemy stwierdzić następujące fakty:

* Dla obu sieci lepszy rezultat przyniosło zastosowanie konfiguracji warstwy wyjściowej nr 2 (sieć skonfigurowana do rozpoznawania wzorców).
* Poniższa tabelka przedstawia zestawienie najlepszych wariantów obu sieci pod kątem dokładności

|  |  |  |  |
| --- | --- | --- | --- |
| Sieć | **RBF** | **FEEDFORWARD** | **WYBÓR LOSOWY** |
| % poprawnych odp. | ~ **30%** | ~ **20%** | **4%** |

W tabelce zawarto również wynik dla losowych wyborów. Dla planszy 5x5 prawdopodobieństwo wyboru właściwego pola wynosi 1/25 co daje nam 4%.

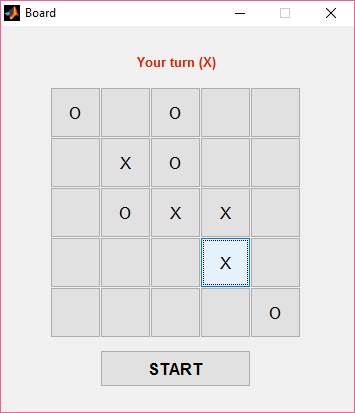
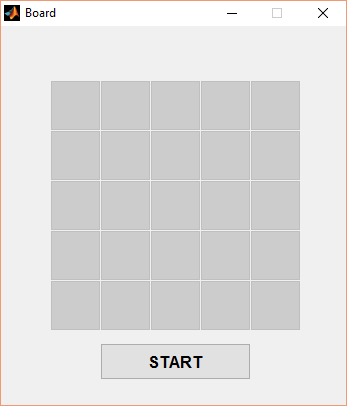
Oczywiście otrzymane rezultaty nie są zbytnio zadowalające.

Jak widzimy najlepszą dokładność możemy uzyskać poprzez zastosowanie sieci RBF. Dokładność dla tej sieci możemy nieco zwiększyć poprzez zwiększenie zbiory uczącego. Analizując otrzymany wykres (3.2.1.1) możemy domniemywać iż zwiększenie zbioru uczącego do 20 000 mogłoby dać w rezultacie dokładność około **40%**, aczkolwiek trzeba mieć świadomość ograniczeń wynikających z długiego czasu generowania takiej liczby danych oraz ograniczenia pamięci komputera

.

1. GUI

Interfejs graficzny jest bardzo prosty w użyciu co usprawnia użytkownikowi grę.

Jak widzimy na powyższych obrazkach, gra rozpoczyna się dopiero po naciśnięciu przycisku START, inaczej klikanie po tabeli nie daje żadnych rezultatów. Nie jest to jedyny przycisk w oknie. Cała tabela do gry złożona jest z przycisków. Dodatkowo, komunikat wyświetlany na czerwono sugeruje nam nasz ruch, bądź ruch komputera a także informuje o wygranej.

1. Wnioski

TODO

1. Bibliografia