PSZT – Sieć neuronowa

1. Treść projektu

Temat: **Klasyfikacja punktów zadawanych przez użytkownika kliknięciami.**

Opis: Klikając w okienku lewym lub prawym klawiszem myszy użytkownik zadaje punkty należące do dwóch kategorii. Sieć neuronowa uczy się klasyfikować punkty na płaszczyźnie wg tych kategorii.

Algorytm: uczenie się on-line z zadanym parametrem kroku.

Struktura sieci: 1 warstwa ukryta z 20 neuronami lub 2 warstwy ukryte z 10-oma; należy też poeksperymentować z innymi strukturami.

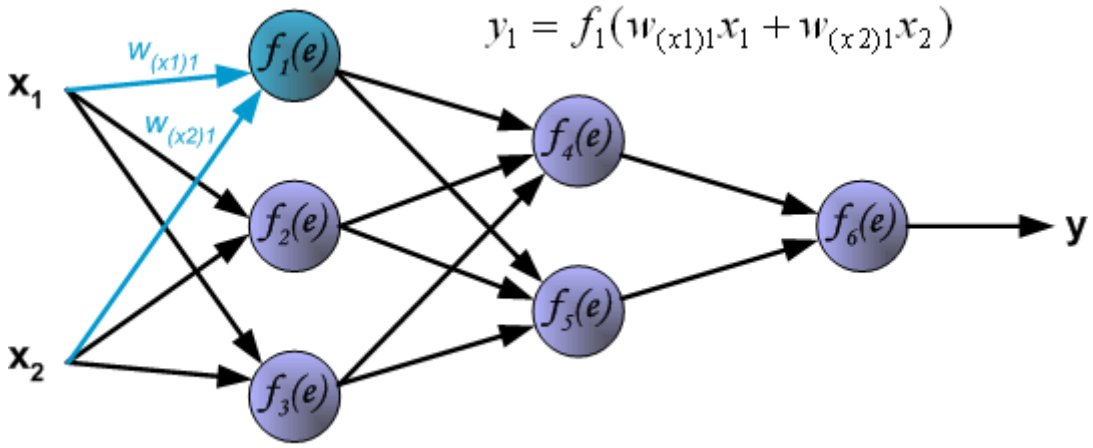
Interfejs: prosty interfejs graficzny umożliwiający definiowanie struktury sieci, wyklękiwanie punktów i demonstrujący klasyfikację wyznaczaną przez sieć.

Sieć została zaprojektowana w taki sposób, aby umożliwić stworzenie dowolnej struktury ukrytej. W konstruktorze sieci należy przekazać ilość warstw ukrytych, a odpowiedni parametr w klasie NeuronNetwork stwierdzi, ile neuronów przypada na każdą warstwę.

2. Algorytmy

Uczenie się sieci neuronowej zostało zrealizowane przy pomocy algorytmu propagacji wstecznej (ang. *backpropagation*) w trzech fazach.

- Faza 1 składa się z obliczenia wartości dla każdego z neuronów warstw ukrytych i warstwy wyjściowej (liczby neuronów w warstwach oraz liczba warstw jest przypadkowa):



gdzie:

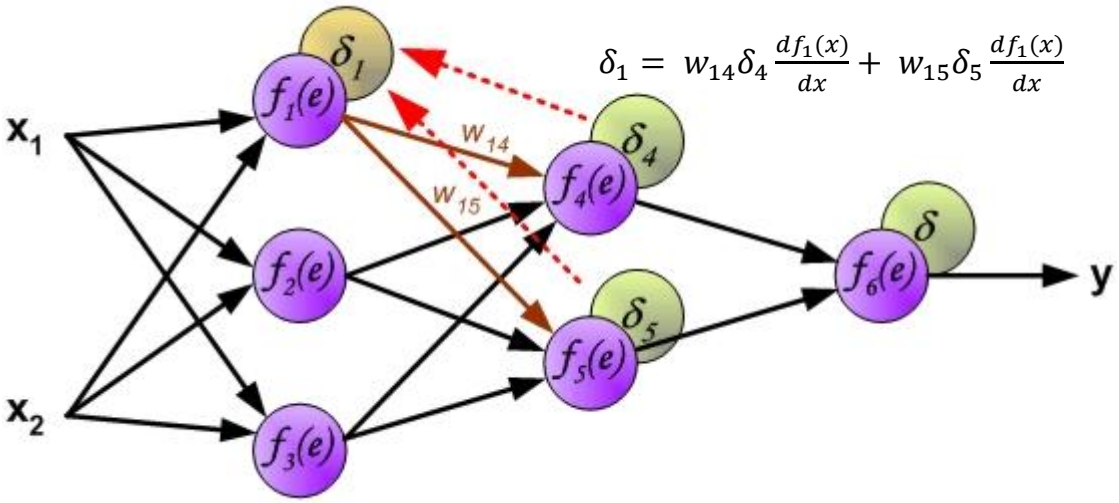
w – waga synapsy
x – obliczona wartość neuronu
f – funkcja aktywacji

Jako funkcję aktywacji w warstwach ukrytych przyjęto sigmoidę o wzorze:

$$f(x) = \frac{1}{1. + e^{-x}}$$

natomiast w warstwie wyjściowej funkcją aktywacji jest funkcja liniowa f(x) = x.

- Faza 2 polega na propagacji wstecznej błędu obliczenia , wzdłuż całej sieci:



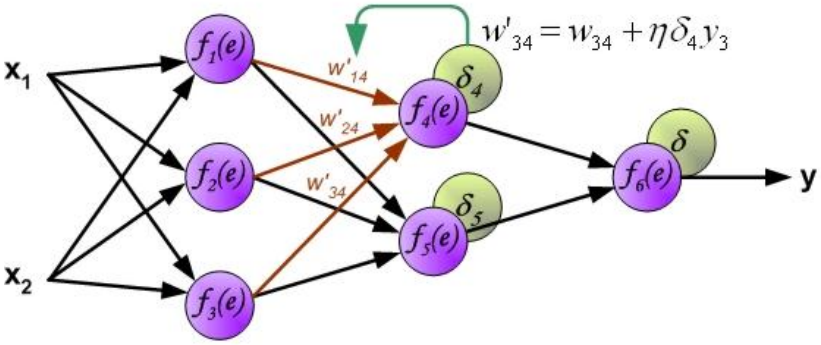
Przy czym dla warstwy zewnętrznej delta obliczana jest przez odjęcie od wartości oczekiwanej, wartości zwracanej przez neuron (y).

- Faza 3 polega na obliczeniu nowych wag połączeń, z uwzględnieniem wcześniej policzonych błędów (delt):

$$w'_{14} = w_{14} + \eta\delta_4 y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 y_2$$

$$w'_{34} = w_{34} + \eta\delta_4 y_3$$



3. Implementacja

W projekcie zaimplementowano następujące klasy:

Neuron – klasa reprezentująca pojedynczy neuron. W przypadku chęci zastąpienia funkcji aktywacji własną, należy dziedziczyć po klasie Neuron i przeciążyć metodę sigmoid(double x).

Synaps – klasa reprezentująca połączenie między dwoma neuronami. Z synapsą związana jest jej waga, która jest istotna przy przechodzeniu przez sieć.

NeuronNetwork – klasa reprezentująca perceptron, posiadająca jedną warstwę zewnętrzną i wewnętrzną oraz dowolną liczbę warstw ukrytych. W tej klasie zaimplementowano algorytm propagacji wstecznej. Cała komunikacja z siecią odbywa się przez metody udostępniane przez obiekt tej klasy.

Mainwindow – klasa implementująca interfejs graficzny programu

UniTest – klasa posiadająca zaawansowane testy pozwalające sprawdzić poprawność działania sieci.

Dokładne opisy klas, ich pól oraz metod można znaleźć w załączonej dokumentacji **JavaDoc**.

4. Przykłady użycia

Stworzone w celu ukazania możliwości sieci – nawet dla bardziej skomplikowanych kształtów i małej ilości danych wejściowych.