

Ćwiczenie nr 2

Klasyfikator neuronowy na bazie wielowarstwowego perceptronu

Część teoretyczna

Wykład 6: Sztuczne sieci neuronowe – klasyfikacja.

Zadania pomocnicze

Zapoznaj się z programem **nnd4db**, który pozwala ręcznie ustawiać granicę decyzyjną realizowaną przez neuron oraz z programem **nnd4pr**, w którym granica decyzyjna ustawiana jest przez neuron w procesie treningu. Punkty uczące należą do dwóch klas – czarnej i białej, możemy je przesuwac. W programie **nnd4pr** przycisk „Learn” uruchamia jeden krok treningu – prezentację na wejściu współrzędnych jednego punktu i adaptację wag, jeśli punkt ten jest źle zaklasyfikowany przez neuron. Przycisk „Train” uruchamia pięć kroków treningu. Przycisk „Random” nadaje wagom losowe wartości. Możemy wybrać opcję „Bias” – umożliwia to przesuwanie granicy decyzyjnej poprzez zmianę wagi b – lub „No bias” – wtedy granica decyzyjna kręci się wokół punktu (0, 0) (patrz program nnd2n1). Punkty źle klasyfikowane mają czerwoną otoczkę.

Załączam też skrypty (Przyk_*.m), w których oprogramowałem przykłady z [Żurada96].

Zadania do wykonania

Dany jest zbiór chorych A oraz zbiór zdrowych B. Każdy obiekt (człowiek) w tych zbiorach opisany jest dwiema cechami x_1 i x_2 (np. temperatura ciała, parametry krwi, występowanie lub brak określonych alleli genów itp.). Zaprojektuj klasyfikator neuronowy na bazie perceptronu do rozróżniania klas A i B.

Zbiory danych A i B generujemy samodzielnie za pomocą funkcji gener_zbiorow(), wprowadzając argumenty:

- nr_gr: nr sekcji np. 1.1, 1.2, ...
- wariant: 1 – zbiory A i B separowalne liniowo, 2 – separowalne nieliniowo, 3 - częściowo pokrywające się,
- lp: liczba punktów w zbiorach A i B.

Uwaga – w kodzie funkcji wprowadzamy aktualny rok kalendarzowy rk.

Funkcja zwraca zbiory A i B.

Zadanie realizujemy dla trzech zestawów danych wykorzystując program perceptron_kl:

1. zbiory A i B separowalne liniowo,
2. zbiory A i B separowalne nieliniowo,
3. zbiory A i B częściowo pokrywające się.

Dla każdego zestawu danych funkcji gener_zbiorow używamy dwukrotnie – do generacji zbiorów uczących A_trn i B_trn oraz testowych A_tst i B_tst, przyjmując $lp = 100$.

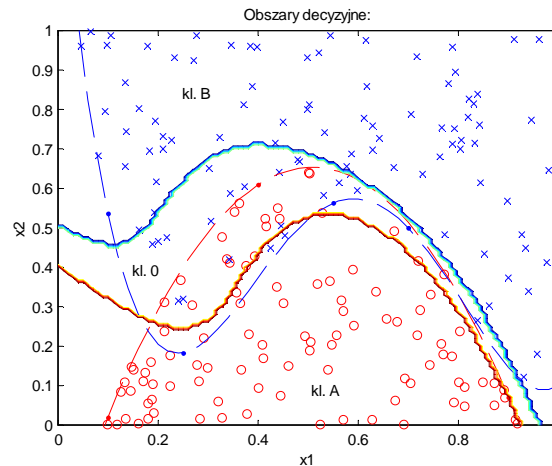
Z wykorzystaniem programu perceptron_kl dla każdego zestawu danych przeprowadź eksperymenty numeryczne zmieniając:

- a. liczbę neuronów w warstwie ukrytej ($l_{neur}=1, 2, 3, \dots$),
- b. postaci funkcji aktywacji (patrz poprzednie ćwiczenie),
- c. liczbę epok uczenia (20, 50, 100, ...),
- d. ewentualnie metody uczenia sieci (ang. training function).

Analogicznie do poprzedniego ćwiczenia w każdym punkcie a-d zastosuj wariant optymalny sieci z punktu wcześniejszego.

Opisz eksperymenty i wnioski z nich wypływające. W tabeli zamieść odsetki poprawnych, złych i nierozstrzygniętych klasyfikacji dla kilkunastu wybranych eksperymentów a-d i każdego zestawu danych 1–3.

Dla każdego zestawu danych i najlepszego twoim zdaniem wariantu sieci zamieść w sprawozdaniu wykresy obrazujące wygenerowane punkty oraz obszary decyzyjne utworzone przez sieci, tj. trzy wykresy postaci:



Następnie dla drugiego zestawu danych i najlepszego wariantu sieci zbadaj jak kształtuje się odsetek poprawnych, błędnych i nierozstrzygniętych klasyfikacji w zależności od liczności zbioru treningowego. W tym celu wygeneruj zbiory A_trn i B_trn o liczynościach 5, 10, 15, ..., 40, 60, 80, 100. Liczność zbiorów testowych A_tst i B_tst jest stała, przyjmij 500. Dla każdego zbioru przeprowadź trzykrotnie trening i test sieci. Zanotuj najlepsze wyniki i przedstaw na wykresie ww. zależności.

Zadania dodatkowe dla ambitnych

1. Oprogramuj przykład 3.2 z [Żurada96]. Wykonaj obliczenia i zobrazuj działanie dychotomizatora wykresami (rys. 3.12). Porównaj wyniki.
2. Oprogramuj przykład 3.3 z [Żurada96]. Wykonaj obliczenia i zobrazuj działanie klasyfikatora ciągłego wykresami (rys. 3.16). Porównaj wyniki.
3. Oprogramuj przykład 3.4 z [Żurada96]. Wykonaj obliczenia i zobrazuj działanie klasyfikatora ciągłego wykresem (rys. 3.18). Porównaj wyniki.
4. Oprogramuj przykład 4.3 z [Żurada96]. Wykonaj obliczenia i wykreśl linie decyzyjne opisane w tym przykładzie. Porównaj wyniki.

Przykładowe pytania zaliczeniowe

1. Narysuj model sztucznego neuronu.
2. Narysuj model sieci neuronowej użytej w ćwiczeniu.
3. Na czym polega klasyfikacja danych.
4. Narysuj i objaśnij wykres ze sprawozdania.
5. Funkcje aktywacji neuronów.
6. Wsteczna propagacja błędów.
7. Neuron jako klasyfikator.
8. Metody oceny klasyfikatora.

Do przygotowania na następne zajęcia

1. Zapoznać się z instrukcją do kolejnego ćwiczenia.
2. Zapoznać się z częścią teoretyczną do kolejnego ćwiczenia.
3. Wykonać zadania pomocnicze do kolejnego ćwiczenia.