

Zadanie polega na implementacji i testowaniu sieci neuronowej typu MLP (ang. *Multi-layer Perceptron*).

Implementacja perceptronu ma mieć charakter uniwersalny, a zatem umożliwiającą łatwą skalowalność jego architektury. Oznacza to, że kod powinien gwarantować poprawność działania i nauki perceptronu bez względu na liczbę warstw i neuronów w poszczególnych warstwach, które to parametry powinny być określone w momencie tworzenia sieci. Neurony przetwarzające, czyli neurony ukryte i wyjściowe (ewentualne neurony wejściowe nie są zaliczane do neuronów przetwarzających) mają mieć charakter nieliniowy i wykorzystywać sigmoidalną funkcję aktywacji, której współczynnik nachylenia ma być równy 1. Program ma umożliwić określenie, czy podczas obliczania pobudzenia neuronu (sumy ważonej jego wejść) neurony przetwarzające mają uwzględniać wartość wejścia obciążającego (ang. *bias*), czy też nie. Wagi sieci, o ile nie jest ona wczytywana z pliku, mają być inicjalizowane w sposób pseudolosowy wartościami z niewielkiego przedziału otaczającego 0 (np. z przedziału $[-0,5; 0,5]$ lub $[-1; 1]$). Program ma umożliwić zachowanie sieci do pliku oraz wczytanie z pliku zapisanej w nim sieci. Ma także umożliwiać wczytanie zestawu wzorców z pliku.

Program ma zapewniać dwa tryby działania sieci: tryb nauki i tryb testowania.

W trybie nauki perceptron uczony jest metodą z nauczycielem. W każdej epoce prezentowane są sieci wszystkie wzorce treningowe, przy czym program powinien udostępniać możliwość wyboru pomiędzy niezmienną oraz losową kolejnością ich prezentacji. Sekwencja czynności, która zostaje wykonana dla pojedynczego wzorca, wygląda tu następująco: wzorec treningowy podawany jest na wejścia sieci, następnie odbywa się jego propagacja w przód, dalej na podstawie wartości odpowiedzi wygenerowanej przez sieć oraz wartości pożądanego wzorca odpowiedzi następuje wyznaczenie błędów, po czym propagowane są one wstecz, na koniec zaś ma miejsce wprowadzenie poprawek na wagi. Aktualizacja wag ma zatem charakter on-line (odbywa się po każdorazowej prezentacji wzorca). Program ma także umożliwiać określenie, czy w trakcie nauki ma być uwzględniany człon *momentum*, czy nie.

Czas trwania nauki powinien być determinowany albo zrealizowaniem wprowadzonej przez użytkownika liczby epok, albo osiągnięciem przez sieć podanego przez użytkownika poziomu błędu (należy jednak umożliwić tu zatrzymanie nauki w pewnym momencie, gdyby założony poziom błędu okazał się nieosiągalny), albo spełnieniem któregośkolwiek warunku z dwóch wymienionych.

Podczas nauki ma następować rejestrowanie do pliku wartości globalnego błędu popełnionego przez sieć w danej epoce, ale co określoną liczbę epok z pewnym założonym skokiem (np.: w każdej epoce, co 10, 20 lub 50 epokę). Skok może być z góry ustalony przez programistę lub wprowadzany przez użytkownika.

W trybie testowania wyznaczane są odpowiedzi sieci dla poszczególnych wzorców, natomiast nauka, a więc modyfikacja wag, nie zachodzi. Sekwencja czynności w tym

przypadku ogranicza się do trzech: wzorzec treningowy podawany jest na wejścia sieci, następnie odbywa się jego propagacja w przód, a na koniec na podstawie wartości odpowiedzi wygenerowanej przez sieć oraz wartości pożądanego wzorca odpowiedzi następuje wyznaczenie błędów.

Podczas testowania ma również następować rejestrowanie do pliku pewnych wielkości, a mianowicie: wzorca wejściowego, popełnionego przez sieć błędu dla całego wzorca, pożądanego wzorca odpowiedzi, błędów popełnionych na poszczególnych wyjściach sieci, wartości wyjściowych neuronów wyjściowych, wag neuronów wyjściowych, wartości wyjściowych neuronów ukrytych, wag neuronów ukrytych (w kolejności warstw od dalszych względem wejść sieci do bliższych). Ewentualnie program może umożliwiać użytkownikowi wybór tylko niektórych spośród nich do rejestracji w danym przebiegu.

Część badawcza polega na wykorzystaniu zaimplementowanej sieci do realizacji dwóch zadań:

1. Klasyfikacja zbioru Irysów (<https://archive.ics.uci.edu/ml/datasets/iris>);

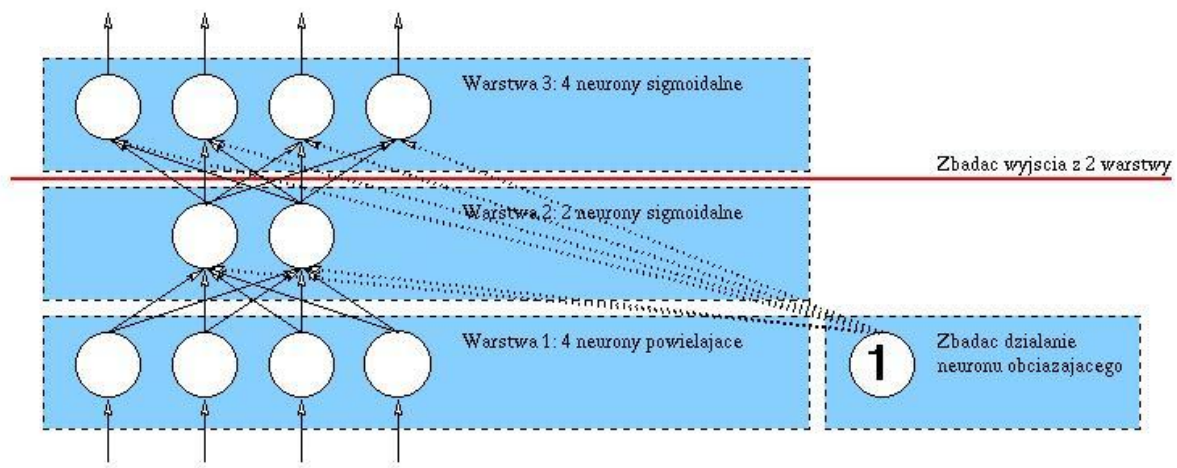
2. Autoasocjacja (sieć typu autoenkoder).

W pierwszym przypadku (Irysy) struktura sieci powinna być dobrana przez użytkownika. Należy zaproponować podział zbioru na część treningową i testową, tak, aby uzyskać dobre własności generalizacyjne sieci. Skuteczność nauki należy określić, obliczając:

- liczbę poprawnie sklasyfikowanych obiektów (łącznie i w rozbiciu na klasy);
- macierz pomyłek (ang. *confusion matrix*);
- wartości *precision*, *recall* i *F-measure*.

W drugim przypadku (autoenkoder) sieć powinna mieć następującą strukturę:

1. warstwa wejściowa (nieprzetwarzająca): 4 neurony;
2. warstwa ukryta (nieliniowa): 2 neurony;
3. warstwa wyjściowa (nieliniowa): 4 neurony.



Sieć tę należy nauczyć 4 poniższych wzorców (zapisanych tu w sposób - ((wejścia),(wyjścia))):

((1,0,0,0),(1,0,0,0)),
 ((0,1,0,0),(0,1,0,0)),
 ((0,0,1,0),(0,0,1,0)),
 ((0,0,0,1),(0,0,0,1)).

Należy zbadać wpływ uwzględnienia obciążenia w neuronach nieliniowych na skuteczność nauki tychże wzorców. W tym celu należy prowadzić naukę przy losowej kolejności podawania wzorców w każdej epoce i współczynniku nauki: 0,6, bez uwzględniania członu momentum. Należy wyjaśnić uzyskany wynik (wskazówka: zbadać wartości otrzymywane na wyjściach neuronów ukrytych po zakończeniu nauki).

Kolejne eksperymenty zastosować tylko do tej wersji perceptronu spośród dwóch powyższych (tj. z obciążeniem lub bez), dla której uzyskana została zbieżność w procesie nauki.

Zbadać szybkość uczenia perceptronu w zależności od uwzględnienia członu momentum, a także dla różnych wartości współczynnika uczenia i momentum.

Przetestować następujące kombinacje:

- współczynnik nauki - 0,9; współczynnik momentum - 0,0;
- współczynnik nauki - 0,6; współczynnik momentum - 0,0;
- współczynnik nauki - 0,2; współczynnik momentum - 0,0;
- współczynnik nauki - 0,9; współczynnik momentum - 0,6;
- współczynnik nauki - 0,2; współczynnik momentum - 0,9.

Dla zainteresowanych: podać matematyczne wyjaśnienie, dlaczego współczynnik momentum równy bądź większy od 1 może w ogólności prowadzić do rozbieżności podczas nauki.