

Konrad Jaworski 216782 216782@edu.p.lodz.pl
Jakub Plich 216866 216866@edu.p.lodz.pl

Zadanie 2.: Perceptron wielowarstwowy i klasyfikacja

1. Cel

Celem zadania było:

- zaimplementowanie oraz zbadanie działania perceptronu wielowarstwowego
- rozwiązywanie problemu klasyfikacji przy pomocy różnych klasyfikatorów

2. Wprowadzenie

2.1. Perceptron wielowarstwowy

Pierwsza część ćwiczenia przed zadaniem 2 zakłada zaimplementowanie perceptronu. Stworzona przez nas implementacja perceptronu jest uniwersalna i pozwala na skalowalność jego architektury. Jako metodę nauki wykorzystuje wsteczną propagację błędów. Perceptron posiada 2 tryby: nauki i testowania. W trybie nauki perceptron uczony jest metodą z nauczycielem. Sekwencja czynności podczas nauki wygląda następująco: wzorzec treningowy podawany jest na wejścia sieci, następnie odbywa się jego propagacja wprzód, 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. Program umożliwia dostosowanie wartości współczynników nauki i momentum. Czas nauki określony jest przez liczbę epok lub wielkość błędu.

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

W części badawczej ćwiczenia należało nauczyć perceptron autoasocjacji 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)).

Dodatkowo należało zbadać wpływ uwzględnienia obciążenia w neuronach nieliniowych na skuteczność nauki. W tym celu zbadane zostały wartości otrzymywane na wyjściach neuronów ukrytych po zakończeniu nauki (przy współczynnikach: nauki - 0,6 momentum - 0,0). Zbadana została również szybkość nauki perceptronu dla różnych wartości współczynników.

2.2. Problem klasyfikacji

Zadanie polegało na rozwiązaniu problemu klasyfikacji dwóch zbiorów danych przy pomocy dwóch klasyfikatorów. Do implementacji klasyfikatorów został wykorzystany wcześniej wykonany perceptron wielowarstwowy oraz implementacja biblioteczna algorytmu k najbliższych sąsiadów. Zbiór danych do treningu perceptronu stanowi jedną trzecią zbioru (wybierany jest co trzeci zestaw). Reszta danych stanowi zbiór do nauki perceptronu. Baza danych testowych dla klasyfikatora KNN stanowi 30 procent całego zbioru danych. Reszta czyli 70 procent zbioru stanowi zbiór do nauki.

3. Opis implementacji

Program implementujący perceptron oraz dodatkowy klasyfikator został napisany w języku Python. Wybór uzasadniony był dostępem do bibliotek znacząco ułatwiających wczytywanie i przetwarzanie danych oraz generowanie wykresów interpretujących wyniki. Ogólnie dostępne biblioteki zostały wykorzystane do implementacji drugiego klasyfikatora (algorytm k najbliższych sąsiadów).

3.1. Menu

Plik ten zawiera kod pozwalający stworzyć sieć o konkretnych parametrach:

- ilość warstw
- ilość neuronów na poszczególnej warstwie
- obecność biasu
- współczynnik nauki
- współczynnik momentum
- zbiór danych
- ilość epok
- porządana wielkość błęd

3.2. prepData

Plik zawiera zbiór funkcji przygotowujących dane do przetwarzania oraz dzielenia ich na zbiory do nauki i testów.

3.3. NeuralNetwork

Plik ten zawiera:

- definicję funkcji aktywującej
- definicję pochodnej funkcji aktywującej

- definicję klasy `NeuralNetwork` implementującej propagację w przód i w tył
- definicję klasy `NeuronLayer` implementującej funkcjonalność pojedynczej warstwy
- definicję funkcji `learn` implementującej proces nauki sieci
- definicję funkcji `test` implementującej proces testowania sieci

3.4. KNN

Plik zawiera implementację klasyfikatora opartego o algorytm k najbliższych sąsiadów z użyciem bibliotek.

4. Materiały i metody

Aby uruchomić program należy umieścić w katalogu z programem wybrane zbiory danych. W celu skorzystania z implementacji klasyfikatora opartego o perceptron należy uruchomić plik "Menu" po wcześniejszym dobraniu odpowiednich współczynników i zbioru danych. Aby skorzystać z klasyfikatora opartego o algorytm k najbliższych sąsiadów należy po wybraniu zbioru danych uruchomić plik "KNN".

5. Wyniki

5.1. Perceptron autoasocjacji

Do zadanego błędu - 0.001

5.1.1. Wpływ uwzględnienia obciążenia w neuronach nieliniowych na skuteczność nauki

Ilość iteracji: 10000