

Perceptron prosty oraz Adaline

Piotr Grzybowski

11 październik 2017

1 Opis problemu

Symulowanie działania bramek logicznych realizujących proste funkcje logiczne przy pomocy modeli matematycznych jako problem klasyfikacji binarnej.

Celem klasyfikacji binarnej jest *zaklasyfikowanie*, czyli przypisanie każdego elementu z danego zbioru do dwóch rozłącznych kategorii.

Rozważmy bramki logiczne, które mają dwa wejścia i jedno wyjście. Taka bramka działa jak klasyfikator binarny. Parze sygnałów wejściowych zostaje przypisana wartość zero lub jeden. (Przypisanie pary wejść do jednej z dwóch możliwych kategorii.)

W klasycznych bramkach logicznych wejścia jak i wartość realizowanej funkcji przyjmuje dyskretne wartości: zero lub jeden. Naszym zadaniem będzie zbudowanie, oraz wyuczenie modelu, który będzie poprawnie odzwierciedlał działanie funkcji także w przypadku gdy na wejściu pojawiają się wartości ciągle z pewnym odchyleniem ϵ . Przykład działania bramki logicznej *AND* przy $\epsilon = 0.05$. $(0.95 \text{ AND } 0.05) = 0$.

Zbiór danych za pomocą którego odbędzie się uczenie sieci neuronowej składać się będzie z uporządkowanych trójek (x_1, x_2, y) , gdzie x_1, x_2 to wartości sygnałów wejściowych do bramek logicznych, oraz y jako wartość konkretnej funkcji logicznej dla podanych wejść. (Klasa do której możemy przypisać daną parę sygnałów wejściowych.)

2 Proponowane rozwiązanie

Powyżej zaprezentowany problem zostanie rozwiązany przy użyciu prostej sieci neuronowej jako klasyfikatora binarnego. A dokładniej tylko pojedynczego neuronu w dwóch wersjach: pojedynczy perceptron prosty, oraz pojedyncza komórka Adaline.

2.1 Sieć neuronowa

Każdy z neuronów składa się z:

- Wektora wag o długości liczbie sygnałów wejściowych. Każdemu sygnałowi wejściowemu x_i , odpowiada dokładnie jedna kolejna waga w_i . Symbolicznie $W = [x_1, \dots, x_N]$, gdzie N to liczba sygnałów wejściowych do neuronu.
- Stałej zwanej "biasem". Liczba rzeczywista.
- Funkcji aktywacji, według której obliczana jest wartość wyjścia neuronów w sieci neuronowej.

Wartość wyjścia neuronu jest liczona w sposób następujący (g - funkcja aktywacji)

$$Output = g(x^T w + b) \quad (1)$$

Podczas rozwiązywania zadania zostaną zbadane dwa typy neuronów. Perceptron prosty oraz komórka Adaline. Ich struktura jest taka sama, jednakże różnią się procedurą uczenia. Sposoby te omówione zostaną poniżej.

2.2 Uczenie perceptronu prostego

Uczenie sieci realizowane jest poprzez aktualizację wag perceptronu dla każdego przypadku treningowego w każdej kolejnej iteracji uczenia.

$$w_i^{t+1} = w_i^t + \mu * \delta * x_i \quad (2)$$

gdzie:

- w_i^{t+1} wartość i -tej wagi w czasie $t + 1$.
- w_i^t - aktualna wartość w_i .
- μ - stała uczenia.
- x_i - wartość i -teg wygnału wejściowego.

W przypadku perceptronu prostego błąd δ definiuje się jako różnicę między wartością oczekiwaną a wartością funkcji aktywacji perceptronu:

$$\delta = E - P \quad (3)$$

gdzie:

- E - wartość oczekiwana,
- P - wartość predykowana

Ostatecznie procedura aktualizacji wag w każdym kroku będzie wyglądać następująco:

$$w_i^{t+1} = w_i^t + \mu * (E - P) * x_i \quad (4)$$

2.3 Uczenie komórki Adaline

Uczenie sieci realizowane jest poprzez aktualizację wag perceptronu dla każdego przypadku treningowego w każdej kolejnej iteracji uczenia w podobny sposób jak dla perceptronu prostego. Jednakże zamiast błędu dyskretnego δ będzie brany gradient błędu średniokwadratowego.

$$w_i^{t+1} = w_i^t + \mu * \nabla_{w_i} \delta * x_i \quad (5)$$

gdzie:

- w_i^{t+1} wartość i -tej wagi w czasie $t + 1$.
- w_i^t - aktualna wartość w_i .
- μ - stała uczenia.
- x_i - wartość i -teg wygnału wejściowego.

Różnicą będzie w postaci funkcji błędu. Skorzystamy tutaj z błędu średniokwadratowego. Wartością błędu będzie różnica wartości oczekiwanej oraz wartości pobudzenia neuronu. W przypadku perceptronu prostego do wyliczenia błędu aplikowana jeszcze była funkcja aktywacji do pobudzenia.

$$\delta_i = d_i - x^T w \quad (6)$$

$$\delta_i^2 = (E_i - \sum_{i=1}^N x_i * w_i)^2 \quad (7)$$

$$\frac{\partial \delta_i^2}{\partial w_i} = 2 * \delta_i * \frac{\partial \delta_i}{\partial w_i} \quad (8)$$

gdzie:

- E - wartość oczekiwana,
- P - wartość predykowana

3 Zbiór danych

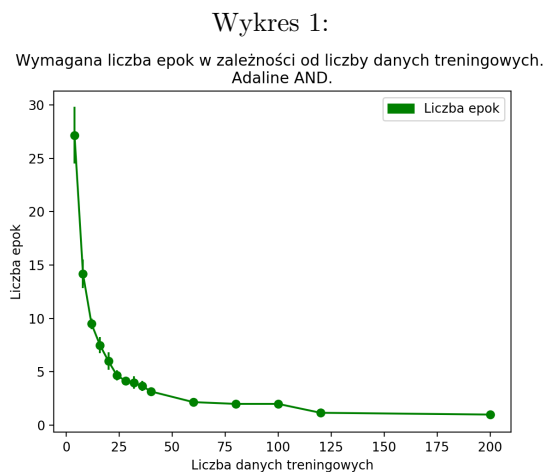
Do przygotowania zbioru danych został napisany generator, żeby w prosty sposób móc otrzymać zadaną liczbę przykładów w zbiorze. Zbiór generuje się w zależności od liczby przypadków oraz odchylenia epsilon. Odchylenie posłuży nam do zamodelowania logiki rozmytej. Przykładowo dla $\epsilon = 0.1$ i dla funkcji logicznej AND, pozytywne sygnały wejściowe przyjmą wartości z zakresu $[0.9, 1]$ a negatywne z zakresu $[0, 0.1]$.

4 Badania

Wyniki uczenia i jakości predykcji neuronów przy różnych parametrach o różnej wielkości zbioru danych.

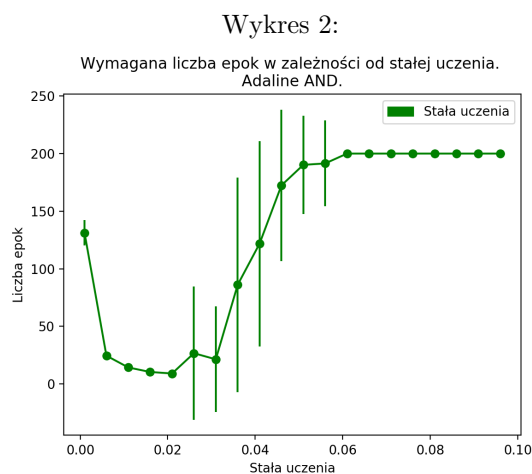
4.1 Adaline

Wymagana liczba epok do wyuczenia Adaline w zależności od wielkości zbioru uczącego na wykresie 1 poniżej:



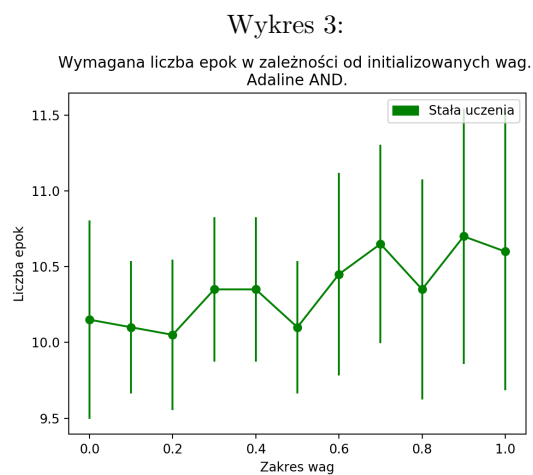
Na podstawie powyższego wykresu wyraźnie widać, że wraz ze wzrostem liczebności zbioru treningowego maleje liczba epok wymaganych do wyuczenia Adaline. Dzieje się tak dlatego, że podczas jednej epoki dostosowujemy wagi więcej razy.

Wymagana liczba epok do wyuczenia Adaline w zależności od stałej uczenia na wykresie 2 poniżej:



Na podstawie powyższego wykresu wyraźnie widać, że zależność liczby epok od kroku uczenia ma pewne minimum. Istnieje taki zakres wartości stałej uczenia w której Adaline wyucza się najszybciej. Oznacza to, że wartość kroku uczenia nie może być ani zbyt mała ani zbyt duża.

Wymagana liczba epok do wyuczenia Adaline w zależności od zakresu inicjalizowanych wag na wykresie 3 poniżej:

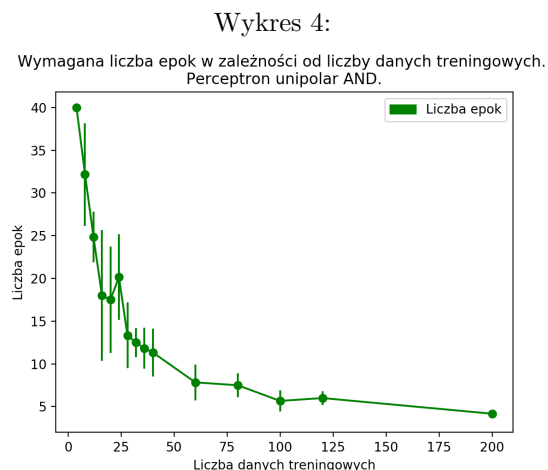


Zakres wag początkowych nie ma większego znaczenia w tym przypadku. Każdy z zakresów charakteryzuje się dużą wariancją i brakuje stabilnych wyników.

4.2 Perceptron

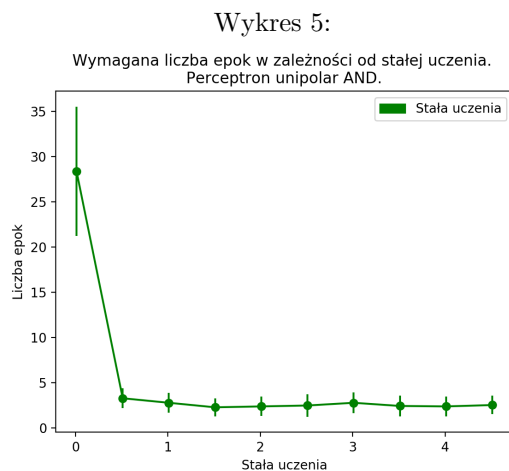
4.2.1 Funkcja aktywacji unipolarna

Wymagana liczba epok do wyuczenia perceptronu prostego dla funkcji aktywacji unipolarnej w zależności od wielkości zbioru uczącego na wykresie 4 poniżej:



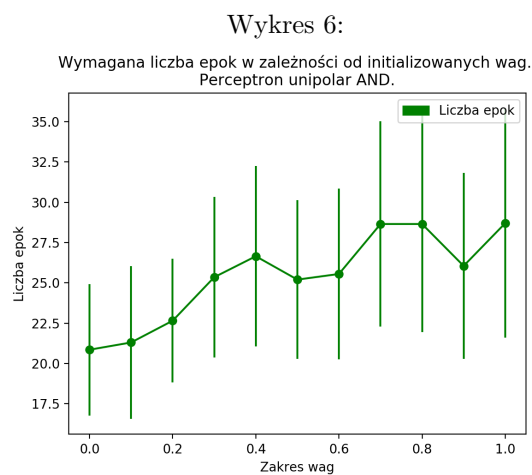
Wnioski do tego wykresu są analogiczne do wykresu 1. Jak w przypadku Adaline.

Wymagana liczba epok do wyuczenia perceptronu prostego dla funkcji aktywacji unipolarnej w zależności od wielkości zbioru uczącego na wykresie 5 poniżej:



Uczenie perceptronu nie ma nic wspólnego z algorytmem gradientu prostego. Nie ma więc znaczenia jak duża będzie stała uczenia. Wagi będą po prostu przeskalowane.

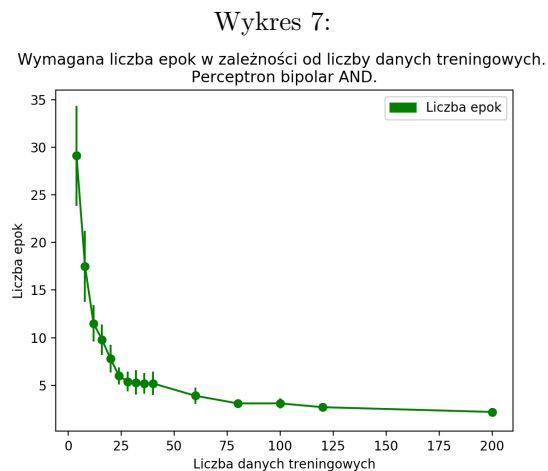
Wymagana liczba epok do wyuczenia perceptronu prostego dla funkcji aktywacji unipolarnej w zależności od wielkości zbioru uczącego na wykresie 6 poniżej:



Wyniki są obarczone dużą wiariancją. Dla bardzo małego zbioru treningowego można wywnioskować pewien trend rosnący, jednakże przy większym zbiorze treningowym, zakres inicjalizowanych wag nie wpływa na szybkość uczenia.

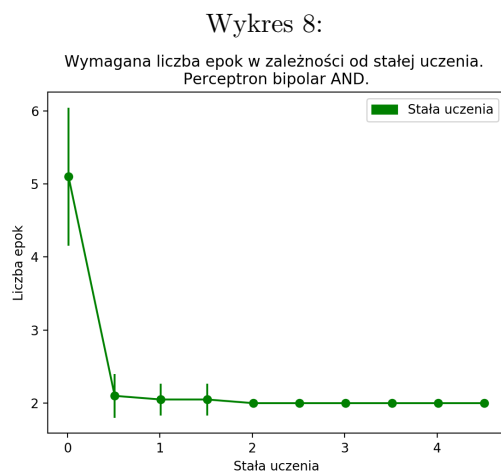
4.2.2 Funkcja aktywacji bipolarna

Wymagana liczba epok do wyuczenia perceptronu prostego dla funkcji aktywacji bipolarnej w zależności od wielkości zbioru uczącego na wykresie 7 poniżej:



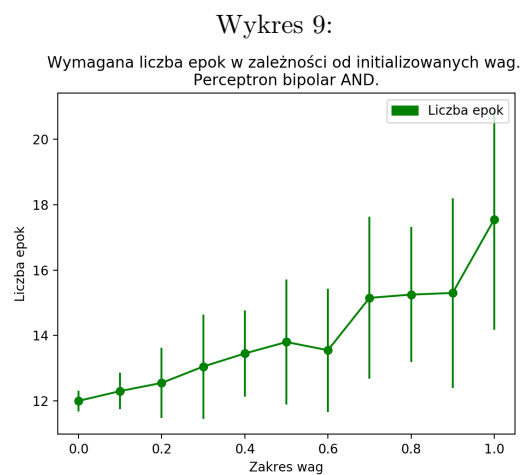
Wnioski do tego wykresu są analogiczne do wykresu 1. Jak w przypadku Adaline.

Wymagana liczba epok do wyuczenia Adaline w zależności od zakresu inicjalizowanych wag na wykresie 8 poniżej:



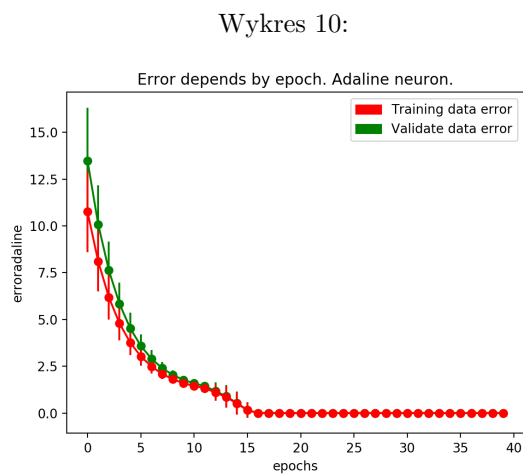
Uczenie perceptronu nie ma nic wspólnego z algorytmem gradientu prostego. Nie ma więc znaczenia jak duża będzie stała uczenia. Wagi będą po prostu przeskalowane.

Wymagana liczba epok do wyuczenia Adaline w zależności od zakresu initializowanych wag na wykresie 9 poniżej:



4.3 Przykładowy wykres błędu w trakcie uczenia

Wykres błędu w zależności od epoki na wykresie 10 poniżej:



5 Podsumowanie

Na podstawie przeprowadzonych badań w trakcie ćwiczenia zauważono, że w przypadku Adaline krok uczenia ma znaczenie w przeciwieństwie do perceptronu prostego. Dla Adaline wartość kroku uczenia nie może być ani zbyt mała ani zbyt duża. Gdy będzie zbyt mała będzie potrzeba zbyt wielu epok aby wyuczyć model, a w przypadku dużego kroku uczenia mamy do czynienia ze zjawiskiem eksplodującego gradientu. Dla perceptronu prostego stała uczenia nie może być zbyt mała z tego samego powodu, jednakże w przypadku jak będzie ona duża to nie wpływa ona na jakość uczenia.

Dla obydwóch modeli, liczba wymaganych epok maleje wraz z liczbą elementów w zbiorze treningowym.

Dla obydwóch modeli zakres wartości wag początkowych nie miał większego znaczenia w szybkości uczenia. Jedynie w przypadku bardzo małego zbioru treningowego, można zauważyć pewien trend. Dla bardzo małego zbioru treningowego, potrzebnych jest mniej epok gdy wagi są zbliżone do zera.

Obydwa modele nie radzą sobie z problemami nieseparowalnymi liniowo.