

Zadanie nr 1 - Pojedynczy neuron i reguła Delta

Inteligentne przetwarzanie danych

Piotr Grzelak, 207549 Bartosz Makowski, 213565

02.04.2017 r.

1 Cel zadania

Celem zadania było zaimplementowanie pojedynczego neuronu liniowego wraz z algorytmem uczenia - regułą delta.

2 Wstęp teoretyczny

2.1 Neuron liniowy

Neuron liniowy jest elementem obliczeniowym, który można opisać za pomocą wektora wag \mathbf{w} . Dla zadanego wektora wejściowego \mathbf{x} oblicza on jako swoją odpowiedź następującą funkcję:

$$f(x) = \sum_{i=1}^n x_i w_i$$

gdzie n oznacza ilość wymiarów wektorów \mathbf{x} i \mathbf{w} , a x_i i w_i oznaczają i -te współrzędne owych wektorów.

2.2 Reguła delta

W celu wyznaczenia wektora \mathbf{w} neuron poddaje się procesowi treningu. W tym celu należy przygotować zbiór par (\mathbf{x}, z) gdzie \mathbf{x} jest wektorem wejściowym, a z jest oczekiwaną odpowiedzią neuronu na podany wektor wejściowy. Jest to tzw. zbiór treningowy. Proces treningu polega na znalezieniu takich wartości wag neuronu, dla których będzie on obliczał wartość najbardziej zbliżoną do oczekiwanej odpowiedzi dla danego wektora wejściowego. Niech:

- N - liczba wag neuronu
- M - ilość elementów w zbiorze treningowym
- K - liczba epok treningu (ilość iteracji jaką będzie wykonywany algorytm)
- $\Omega = \{(\mathbf{x}^m, z^m) \in \mathbb{R}^N \times \mathbb{R}, m = 1, \dots, M\}$ - zbiór M przykładów treningowych gdzie para: (\mathbf{x}^m, z^m) będzie oznaczała m -ty przykład treningowy
- η - współczynnik nauki

Algorytm ma następującą postać:

1. Przypisz wagom neuronu losowe wartości z wybranego wcześniej przedziału: $[\omega_1, \omega_2]$. Przypisz $k = 1$, oznaczające numer aktualnej epoki treningu.
2. Jeśli $k > K$ zakończ algorytm, w przeciwnym razie kontynuuj
3. Przypisz $m = 1$ oznaczające numer prezentowanego przykładu treningowego.
4. Sprawdź czy $m > M$. Jeśli tak to przejdź do , w przeciwnym razie kontynuuj
5. Dla wektora wejściowego $\mathbf{x}^m = [x_1^m, \dots, x_n^m]$ z obecnie przetwarzanego przykładu treningowego, oblicz wyjście neuronu według wzoru:

$$y = \sum_{i=1}^N x_i^m w_i$$

6. Oblicz różnicę między otrzymanym wyjściem neuronu, a oczekiwaną wartością z^m z przykładu treningowego: $\delta = y - z^m$
7. Zaktualizuj wartości wag neuronu według wzoru:

$$w_i^k = w_i^{(k-1)} + \eta \delta x_i^m$$

gdzie: w_i^k oznacza nową wartość i -tej wagi, a $w_i^{(k-1)}$ jej dotychczasową wartość

8. Zwiększ wartość m o 1 i przejdź do kroku 4
9. Zwiększ wartość k o 1 i przejdź do kroku 2

3 Eksperymenty i wyniki

Eksperymenty przeprowadzono wykorzystując zbiór treningowy opisany równaniem:

$$z^m = \alpha_1 x_1^m + \alpha_2 x_2^m + \dots + \alpha_N x_N^m + \epsilon_m$$

gdzie:

- z^m - oczekiwana odpowiedź w m -tym przykładzie treningowym

- x_1^m, \dots, x_N^m - współrzędne wektora wejściowego w m -tym przykładzie treningowym
- $\alpha_1, \dots, \alpha_N$ - współczynniki wygenerowane losowo z przedziału $[-2; 2]$
- ϵ_m - współczynnik wygenerowany losowo z przedziału $[-1, 5; 1, 5]$ dla m -tego przykładu treningowego

Wartości współrzędnych każdego z wektorów wejściowych wylosowano z przedziału $[0; 10]$.

Dla każdego eksperymentu obliczono błąd średniokwadratowy odpowiedzi dawanych przez neuron względem oczekiwanych wartości. Błąd ten wyraża się wzorem:

$$MSE = \frac{1}{M} \sum_{m=1}^M (z^m - y^m)^2$$

gdzie y^m oznacza odpowiedź neuronu na m -ty przykład wejściowy.

Początkowe wagi neuronu losowano z przedziału $[-1; 1]$.

Testom poddano neuron posiadający 10 wejść.

Program zaimplementowano w języku Java.

3.1 Eksperyment nr 1

Eksperyment przeprowadzono dla następujących parametrów:

- rozmiar zbioru treningowego: $M = 20$
- współczynnik nauki: $\eta = 0,001$
- liczba epok treningu: $K = 100$

Poniższa tabela zawiera odpowiedzi neuronu \mathbf{y} dla przykładów treningowych zestawione z odpowiedziami oczekiwanymi \mathbf{z} .

Odpowiedź neuronu y	Oczekiwana odpowiedź z
4,932	4,353
7,830	10,300
6,825	5,988
9,237	9,463
8,322	7,038
14,277	14,076
11,481	11,834
11,066	12,621
11,698	11,377
2,113	2,688
7,106	7,734
6,828	6,512
4,934	5,132
12,448	12,645
10,046	8,623
8,234	8,600
5,421	3,946
3,070	2,690
4,364	5,397

Błąd średniokwadratowy: $MSE = 0,946$

3.2 Eksperyment nr 2

Eksperyment przeprowadzono dla następujących parametrów:

- rozmiar zbioru treningowego: $M = 20$
- współczynnik nauki: $\eta = 0,01$
- liczba epok treningu: $K = 100$

Poniższa tabela zawiera odpowiedzi neuronu **y** dla przykładów treningowych zestawione z odpowiedziami oczekiwanymi **z**.

Odpowiedź neuronu y	Oczekiwana odpowiedź z
NaN	5,132
NaN	10,300
NaN	2,688
NaN	11,834
NaN	8,623
NaN	11,377
NaN	5,988
NaN	9,463
NaN	14,076
NaN	7,734
NaN	2,690
NaN	12,645
NaN	4,353
NaN	6,512
NaN	7,038
NaN	12,621
NaN	3,946
NaN	5,397
NaN	8,600

Błąd średniokwadratowy: $MSE = \text{NaN}$

Program jako wartości wag ustawił NaN (Not a number) co oznacza, że proces uczenia nie był zbieżny.

3.3 Eksperyment nr 3

Eksperyment przeprowadzono dla następujących parametrów:

- rozmiar zbioru treningowego: $M = 20$
- współczynnik nauki: $\eta = 0,001$
- liczba epok treningu: $K = 1000$

Poniższa tabela zawiera odpowiedzi neuronu **y** dla przykładów treningowych zestawione z odpowiedziami oczekiwanymi **z**.

Odpowiedź neuronu y	Oczekiwana odpowiedź z
9,415	9,463
2,268	2,688
10,163	8,623
11,996	11,377
12,548	12,645
11,725	11,834
5,207	5,132
8,355	8,600
14,400	14,076
7,261	7,734
3,203	2,69
7,062	6,512
5,078	4,353
5,525	3,946
11,188	12,621
7,902	10,300
6,864	5,988
8,414	7,038
4,552	5,397

Błąd średniokwadratowy: $MSE = 0,953$

3.4 Eksperyment nr 4

Eksperyment przeprowadzono dla następujących parametrów:

- rozmiar zbioru treningowego: $M = 20$
- współczynnik nauki: $\eta = 0,001$
- liczba epok treningu: $K = 10$

Poniższa tabela zawiera odpowiedzi neuronu **y** dla przykładów treningowych zestawione z odpowiedziami oczekiwanymi **z**.

Odpowiedź neuronu y	Oczekiwana odpowiedź z
5,271	2,690
11,534	12,621
8,104	6,512
12,019	11,377
9,378	7,734
7,766	3,946
12,663	14,076
11,456	9,463
7,615	5,988
8,744	10,300
7,832	8,600
13,59	12,645
4,519	5,132
4,593	4,353
12,319	11,834
9,688	8,623
3,362	5,397
7,966	7,038
2,469	2,688

Błąd średniokwadratowy: $MSE = 2,497$

3.5 Eksperyment nr 5

Eksperyment przeprowadzono dla następujących parametrów:

- rozmiar zbioru treningowego: $M = 10$
- współczynnik nauki: $\eta = 0,001$
- liczba epok treningu: $K = 100$

Poniższa tabela zawiera odpowiedzi neuronu **y** dla przykładów treningowych zestawione z odpowiedziami oczekiwanymi **z**.

Odpowiedź neuronu y	Oczekiwana odpowiedź z
7,233	6,512
8,749	8,600
4,719	5,132
4,428	4,353
8,324	8,623
4,253	3,946
2,603	2,688
7,898	7,734
11,674	11,834

Błąd średniokwadratowy: $MSE = 0,096$

4 Wnioski

- Bardzo ważne jest dobranie odpowiedniej wartości współczynnika nauki. Zbyt duża wartość może spowodować, że proces uczenia neuronu nie będzie zbieżny, co pokazał eksperyment 2.
- Zwiększenie liczby epok treningu wpływa pozytywnie na proces treningu neuronu. Pokazały to eksperymenty 1 i 4. Błąd średniokwadratowy w eksperymencie 1 był mniejszy od tego w eksperymencie 4. Warto jednak zauważyć, że wzrost jakości rezultatów treningu w stosunku do liczby epok nie jest liniowy. Od pewnego momentu zwiększenie liczby iteracji nie przynosi już korzyści. Pokazał to eksperyment 3, w którym nie osiągnięto lepszych wyników niż w eksperymencie 1 mimo większej liczby epok.
- Ciekawe rezultaty przyniósł eksperyment 5 w zestawieniu z eksperymentem 1. Zmniejszenie rozmiaru treningowego przyczyniło się do uzyskania dużo lepszych rezultatów treningu. Otrzymany błąd średniokwadratowy był o rząd wielkości mniejszy. Przyczyną takiego stanu rzeczy może być to, że mniejszy zbiór treningowy posiada mniejszą wariancję co sprawia, że dopasowanie wartości wag minimalizujących błąd popełniany przez neuron jest dużo łatwiejszym zadaniem. Należy jednak mieć wątpliwości czy neuron nauczony na takim zbiorze będzie w stanie dawać lepsze odpowiedzi na przykładach spoza zbioru treningowego.

Literatura

- [1] Materiały udostępnione na stronie przedmiotu, na platformie Wikamp.