

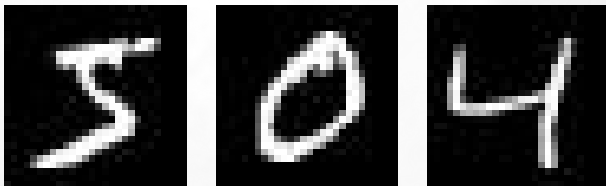
Sieć neuronowa - rozpoznawanie cyfr z bazy MNIST

Dawid Marszałkiewicz
Piotr Matuszak

2016-05-30

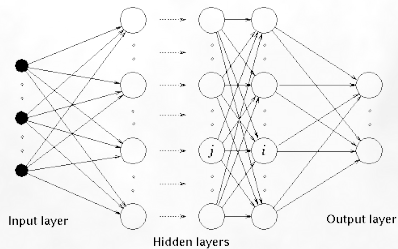
Założenia projektu

Stworzenie sieci neuronowej w języku Python, rozpoznającej cyfry



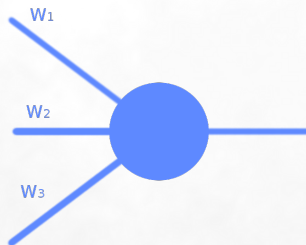
Rysunek: Trzy pierwsze cyfry w bazie treningowej MNIST

Sieć neuronowa



Rysunek: Przykładowa sieć neuronowa

Neuron sigmoidalny

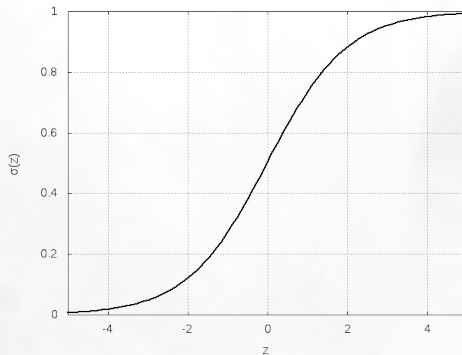


Neuron sigmoidalny

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma(z) \in [0, 1]$$

Neuron sigmoidalny



Rysunek: Wykres funkcji sigmoid

Propagacja do przodu

- Przemnóż wejście X przez wagę w_i - Z

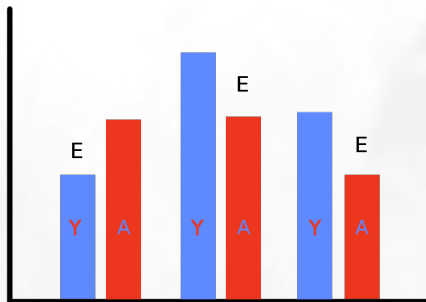
Propagacja do przodu

- Przemnóż wejście X przez wagę w_i - Z
- Zastosuj funkcję $\sigma(Z)$ - A

Propagacja do przodu

- Przemnóż wejście X przez wagę w_i - Z
- Zastosuj funkcję $\sigma(Z)$ - A
- A - wyjście neuronu

Błąd



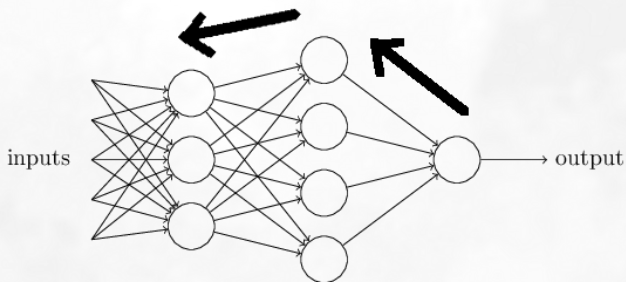
Rysunek: Definicja błędu

$$e = y - a$$

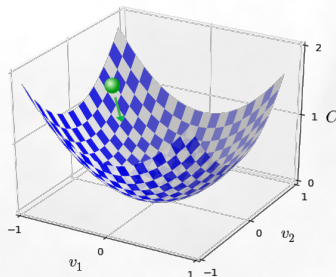
Funkcja kosztów

$$C(w) = \frac{1}{2n} \sum_x ||y(x) - a||^2$$

Wsteczna propagacja



Zmniejszanie błędu - Gradient Descend



Rysunek: Zasada działania metody Gradient Descend w dwóch wymiarach

Kolejność czynności

- Wprowadzić blok danych do nauki

Kolejność czynności

- Wprowadzić blok danych do nauki
- Dla każdego przykładu:

Kolejność czynności

- Wprowadzić blok danych do nauki
- Dla każdego przykładu:
 - Wykonać operację **propagacji do przodu**, obliczając wyjście kolejnych neuronów i podając je na wejścia neuronów po nich następujących (przejście po kolejnych warstwach).

Kolejność czynności

- Wprowadzić blok danych do nauki
- Dla każdego przykładu:
 - Wykonać operację **propagacji do przodu**, obliczając wyjście kolejnych neuronów i podając je na wejścia neuronów po nich następujących (przejście po kolejnych warstwach).
 - Obliczyć błąd na wyjściu z sieci

Kolejność czynności

- Wprowadzić blok danych do nauki
- Dla każdego przykładu:
 - Wykonać operację **propagacji do przodu**, obliczając wyjście kolejnych neuronów i podając je na wejścia neuronów po nich następujących (przejście po kolejnych warstwach).
 - Obliczyć błąd na wyjściu z sieci
 - Przeprowadzić **wsteczną propagację** błędu po sieci

Kolejność czynności

- Wprowadzić blok danych do nauki
- Dla każdego przykładu:
 - Wykonać operację **propagacji do przodu**, obliczając wyjście kolejnych neuronów i podając je na wejścia neuronów po nich następujących (przejście po kolejnych warstwach).
 - Obliczyć błąd na wyjściu z sieci
 - Przeprowadzić **wsteczną propagację** błędu po sieci
- Zaktualizować wagi połączeń sieci

Usprawnienie nauki - Stochastic Gradient Descend

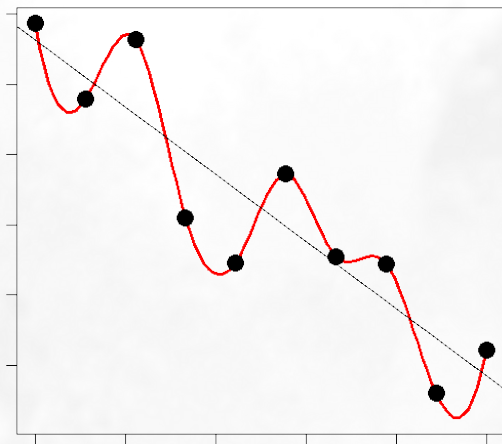
Polega na losowym wyborze z bazy treningowej n przykładów (*mini batch*) i korzystając z tylko części danych, uzyskiwać (dość dobrą) aproksymację gradientu funkcji kosztu.

Usprawnienie nauki - Stochastic Gradient Descend

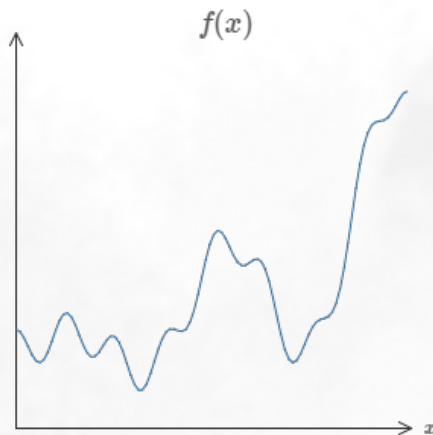
Polega na losowym wyborze z bazy treningowej n przykładów (*mini batch*) i korzystając z tylko części danych, uzyskiwać (dość dobrą) aproksymację gradientu funkcji kosztu.

Stosując analogię „schodzenia w dół doliny” w każdym kroku wybieramy *mniej-więcej* dobrą drogę w dół, nie musi być ona tą najlepszą.

Overfitting



Szukanie minimum globalnego



Przykład

Tabela: Dane empiryczne zebrane do nauki sieci - przykład

Wejście		Wyjście
X_1	X_2	Y
0.2	0.4	0.3
0.6	1.0	0.8
0.1	0.5	0.3

Przykład



Rysunek: Sieć neuronowa dla danych z przykładu