Przegląd gradientowych metod optymalizacji

Przemysław Pobrotyn

Sigmoidal

Koło Uczenia Maszynowego MIM UW, 22.11.2017

sigmoidal

Plan prezentacji

Multi Layer Perceptron - przypomnienie

Batch/ mini-batch/ stochastic gradient descent

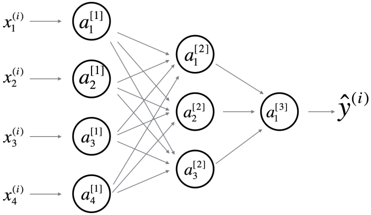
Wykładnicza średnia ważona (EWMA)

Popularne metody optymalizacji gradientowej

Notacja

- Superskrypty: $^{(i)}$ oznacza *i-ty* przykład treningowy, $^{[l]}$ oznacza *l-tą* warstwę sieci
- m: liczność zbioru treningowego
- n_x: liczba cech zbioru treningowego
- n_y: liczba klas do predykcji
- $n_h^{[l]}$: liczba neuronów (*hidden units*) w *l-tej* warstwie
- L: liczba warstw ukrytych w sieci
- $X \in \mathbb{R}^{n_x \times m}$: macierz danych, zbiór treningowy
- $Y \in \mathbb{R}^{n_y \times m}$: macierz etykiet
- $W^{[l]} \in \mathbb{R}^{n_h^{[l]} \times n_h^{[l-1]}}$: macierz wag *l-tej* warstwy
- $b^{[l]} \in \mathbb{R}^{n_h^{[l]}}$: *bias vector* *l-tej* warstwy

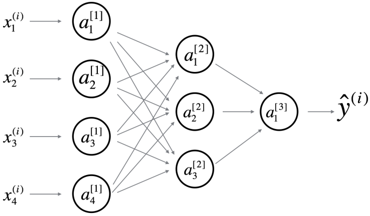
Multi Layer Perceptron - przypomnienie



Rysunek: deeplearning.ai

- $A^{[0]} = X$

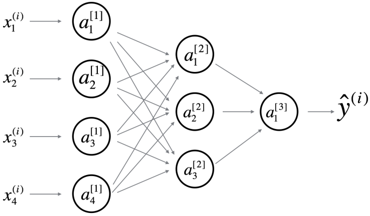
Multi Layer Perceptron - przypomnienie



Rysunek: deeplearning.ai

- $A^{[0]} = X$
- $A^{[l]} = g^{[l]}(W^{[l]}A^{[l-1]} + b^{[l]})$ dla l od 1 do $L - 1$

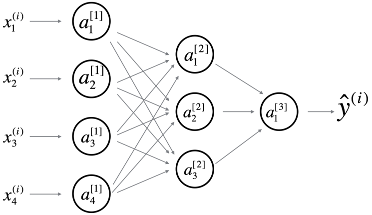
Multi Layer Perceptron - przypomnienie



Rysunek: deeplearning.ai

- $A^{[0]} = X$
- $A^{[l]} = g^{[l]}(W^{[l]}A^{[l-1]} + b^{[l]})$ dla l od 1 do $L-1$
- $\hat{Y} = g^{[L]}(W^{[L]}A^{[L-1]} + b^{[L]})$

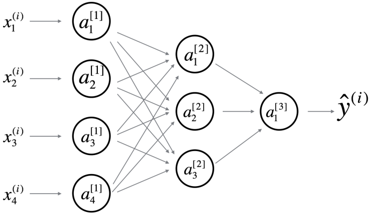
Multi Layer Perceptron - przypomnienie



Rysunek: deeplearning.ai

- $A^{[0]} = X$
- $A^{[l]} = g^{[l]}(W^{[l]}A^{[l-1]} + b^{[l]})$ dla l od 1 do $L-1$
- $\hat{Y} = g^{[L]}(W^{[L]}A^{[L-1]} + b^{[L]})$
- Gradienty $dW^{[l]}$ i $db^{[l]}$ obliczamy korzystając z *backprop*

Multi Layer Perceptron - przypomnienie



Rysunek: deeplearning.ai

- $A^{[0]} = X$
- $A^{[l]} = g^{[l]}(W^{[l]}A^{[l-1]} + b^{[l]})$ dla l od 1 do $L - 1$
- $\hat{Y} = g^{[L]}(W^{[L]}A^{[L-1]} + b^{[L]})$
- Gradienty $dW^{[l]}$ i $db^{[l]}$ obliczamy korzystając z *backprop*
- Akutalizacja parametrów:

$$W^{[l]} = W^{[l]} - \alpha dW^{[l]}.$$

$$b^{[l]} = b^{[l]} - \alpha db^{[l]}.$$

Batch/ mini-batch/ stochastic gradient descent

Trzy warianty spadku gradientu, w zależności od ilości danych wykorzystywanych przy jednej iteracji propagacji w przód.

Batch gradient descent

W jednej iteracji wykorzystujemy cały zbiór treningowy X.

Mini-batch gradient descent

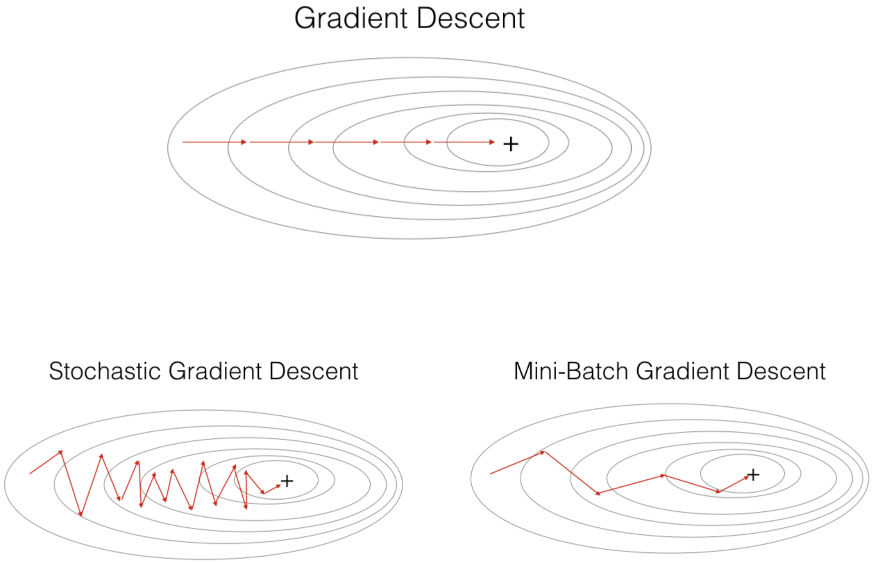
W jednej iteracji wykorzystujemy n elementowy podzbiór zbioru treningowego X.

Stochastic gradient descent

W jednej iteracji wykorzystujemy jeden element $x \in X$.

Epoka (*epoch*)

Jedno 'przejście' przez cały zbiór treningowy.



Rysunek: deeplearning.ai

Mini batch gradient descent - problemy

Mini batch gradient descent - problemy

- Wybór właściwego *learning rate* jest trudny

Mini batch gradient descent - problemy

- Wybór właściwego *learning rate* jest trudny
- Wybór *schedule* zmian *learning rate* jest trudny

Mini batch gradient descent - problemy

- Wybór właściwego *learning rate* jest trudny
- Wybór *schedule* zmian *learning rate* jest trudny
- Używamy jednego *learning* do wszystkich parametrów - cechy mogą mieć rożne częstotliwości występowania

Mini batch gradient descent - problemy

- Wybór właściwego *learning rate* jest trudny
- Wybór *schedule* zmian *learning rate* jest trudny
- Używamy jednego *learning* do wszystkich parametrów - cechy mogą mieć rożne częstotliwości występowania
- Chcemy uniknąć minimów lokalnych i *plateau's*

Mini batch gradient descent - problemy

- Wybór właściwego *learning rate* jest trudny
- Wybór *schedule* zmian *learning rate* jest trudny
- Używamy jednego *learning* do wszystkich parametrów - cechy mogą mieć rożne częstotliwości występowania
- Chcemy uniknąć minimów lokalnych i *plateau's*

Zaprezentujemy metody optymalizacyjne adresujące te problemy.

Wykładnicza średnia ważona (EWMA)

Dane są obserwacje $\theta_1, \theta_2, \ldots, \theta_n$.
Zdefiniujmy wykładniczą średnią ważoną (*exponentially weighted moving average*) jak następuje:

- $v_0 = 0$
- $v_t = \beta v_{t-1} + (1-\beta)\theta_t$ dla $t \in 1 \ldots n$.
- $\beta \in (0, 1)$

Bias correction

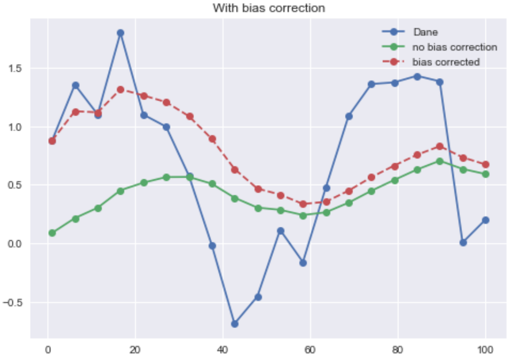
$$
\begin{aligned}
v_t = \beta v_{t-1} + (1-\beta)\theta_t = \\
= \beta(\beta v_{t-2} + (1-\beta)\theta_{t-1}) + (1-\beta)\theta_t = \\
= \beta^2 v_{t-2} + (1-\beta)\beta\theta_{t-1} + (1-\beta)\theta_t = \ldots = \\
= (1-\beta)(\beta^t\theta_0 + \beta^{t-1}\theta_1 + \ldots + \beta\theta_{t-1} + \theta_t)
\end{aligned}
$$

Zauważmy, że suma wag wynosi $1-\beta^t$ Wprowadźmy zatem *bias correction*:

$$v_{corrected} := v_t/(1-\beta^t), t \neq 0$$

Wtedy wagi sumują się do 1.

Bias correction



Prosty przykład: niech $\theta_1 = 10, \theta_2 = 10$
Wtedy EWMA ($\beta = 0.5$) bez *bias correction* wynosi
$(v_0 = 0), v_1 = 5, v_2 = 7.5$. Dodając *bias correction* dostajemy
$(v_0 = 0), v_1 = 10, v_2 = 10$

Interpretacja β

$$\lim_{\epsilon \to 0} (1 - \epsilon)^{\frac{1}{\epsilon}} = \frac{1}{e}$$

Niech $1 - \epsilon = \beta$, wtedy

$$\beta^{\frac{1}{1-\beta}} \approx \frac{1}{e} \approx 0.37$$

Kolejne wagi β^t dla $t > \frac{1}{1-\beta}$ są mniejsze niż 0.37 i maleją wykładniczo.

- Możemy zatem zinterpretować β jak współczynnik mówiący nam ile obserwacji bierzemy pod uwagę obliczając średnią kroczącą.
- Dla $\beta = 0.9$, średnia bierze pod uwagę ostatnie 10 obserwacji
- Dla $\beta = 0.999$, średnia bierze pod uwagę ostatnie 1000 obserwacji

Momentum

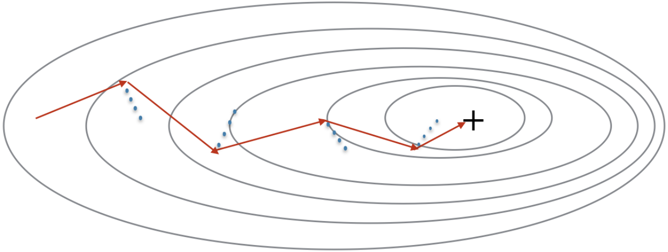
Dla i-tej iteracji:

Oblicz pochodną $d\theta$ funkcji kosztu na obecnym batchu;

$$v_\theta = \beta v_\theta + (1-\beta)d\theta$$

$$\theta = \theta - \alpha v_\theta$$

α i β to hiperparametry, dobrą domyślną wartością dla β jest 0.9.



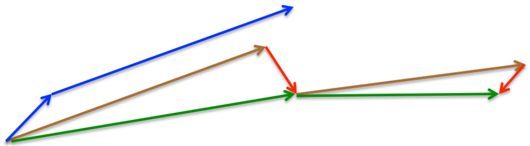
Qian, N. (1999). On the momentum term in gradient descent learning algorithms

Nestorov accelerated gradient

$$v_\theta = \beta v_\theta + \alpha \frac{\partial J(\theta - \beta v_\theta)}{\partial \theta}$$

$$\theta = \theta - v_\theta$$

Podobnie jak w *momentum*, dobrą domyślną wartością dla β jest 0.9.



Rysunek: G.Hinton, *Neural Networks for Machine Learning*, wykład 6c

Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence o(1/k2).

Adagrad

Dotychczas używaliśmy identycznego *learning rate* dla wszystkich parametrów. Metody adaptacyjne dostosowują *learning rate* dla każdego parametru.

Dla k-tej iteracji:

Dla i-tego parametru:

$$\theta_i = \theta_i - \frac{\alpha}{\sqrt{G_{i,i}} + \epsilon} d\theta_i$$

gdzie G to macierz diagonalna gdzie element (i, i) to suma kwadratów gradientów $d\theta_i$ obliczonych w iteracjach od 1 do k.

Problem: Gdy $k \to \infty$, $\frac{\alpha}{\sqrt{G_{i,i}}+\epsilon} \to 0$ ponieważ kwadraty gradientów są nieujemne.

Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.

RMSprop

Dla i-tej iteracji:

Oblicz pochodną $d\theta$ funkcji kosztu na obecnym batchu;

$$s_\theta = \beta s_\theta + (1-\beta)(d\theta)^2$$

$$\theta = \theta - \alpha \frac{d\theta}{\sqrt{s_\theta} + \epsilon}$$

α i β to hiperparametry, dobrą domyślną wartością dla β jest 0.9.
ϵ jest dodany dla stabilności numerycznej i zwykle wynosi 10^{-8}.

G.Hinton, Neural Networks for Machine Learning, wykład 6e

Adam

$$v_\theta = \beta_1 v_\theta + (1 - \beta_1) d\theta$$

$$v_{corrected} = \frac{v_\theta}{1 - (\beta_1)^t}$$

$$s_\theta = \beta_2 s_\theta + (1 - \beta_2)(d\theta)^2$$

$$s_{corrected} = \frac{s_\theta}{1 - (\beta_1)^t}$$

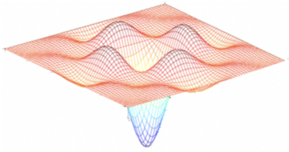
$$\theta = \theta - \alpha \frac{v_{corrected}}{\sqrt{s_{corrected}} + \epsilon}$$

Domyślne wartości: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

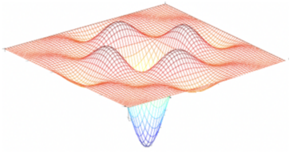
Kingma, D. P., Ba, J. L. (2015). Adam: a Method for Stochastic Optimization

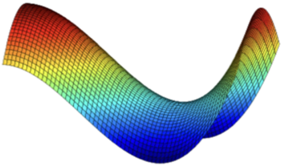
Problem minimów lokalnych

Problem minimów lokalnych

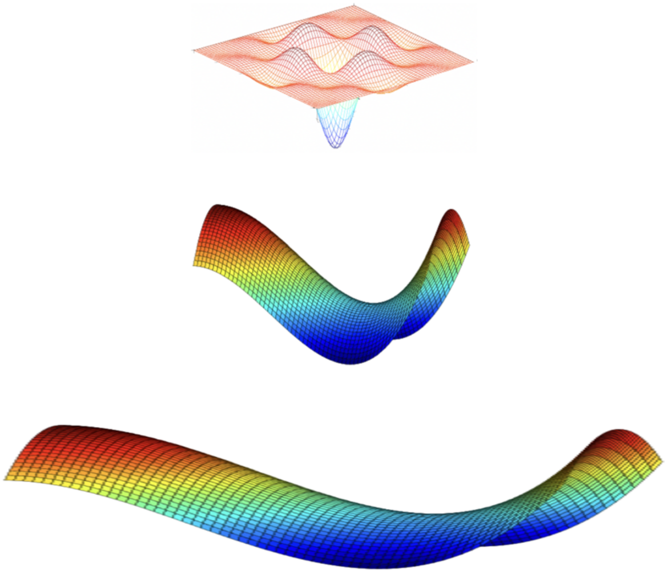


Problem minimów lokalnych





Problem minimów lokalnych



Rysunek: deeplearning.ai

Źródła

- http://ruder.io/optimizing-gradient-descent/
- Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization, Coursera course by deeplearning.ai
- http://cs231n.github.io/neural-networks-3/
- http://www.cs.toronto.edu/ tij-men/csc321/slides/lecture_slides_lec6.pdf

Dziękuję za uwagę!

sigmoidal