SGD a jeho varianty

.1 SGD

V této úloze se zaměřím na vlastnosti SGD a jeho vylepšení. Úloha je rozdělena do menších pod otázek, abych pokryl smysluplněji a přehledněji větší záběr ohledně algoritmů založených na gradientním sestupu. Otázky jsem se snažil klást (pokud to šlo) tak, aby na sebe logicky navazovaly .

Pro optimalizaci a zrychlení učení se používá **stochastický gradientní sestup** (Stochastic gradient descent, SGD).

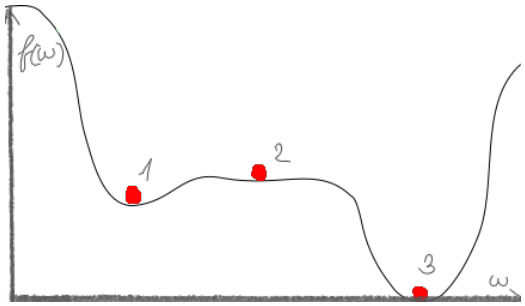
- Formulujte základní vztah této metody. Popište jednotlivé proměnné.

$$w_{k+1} = w_k - \alpha \cdot \frac{\partial f^T(w)}{\partial w}$$

w_{k+1}...*update vah*
w_k...*váhy*
α...*step size*
$\frac{\partial f^T(w)}{\partial w}$...*gradient funkce*

- Jaké nevýhody má tento iterační algoritmus? Nakreslete příklad a popište jej. Diskutujte problémy a nastiňte jejich možné řešení.

SGD se velmi jednoduše zasekne v lokálním minimu nebo na ploše s malým gradientem.



V prvním případě jsme pomocí SGD nalezli lokální minimum, které ale není optimální. Předejít uvíznutí v tomto bodě můžeme změnou step size nebo jinou vhodnou inicializací.

V druhém bodě jsme uvízli na ploše s malým gradientem. Algoritmus se v tomto bodě zpomalí nebo dokonce zasekne. Zvýšením step size zase můžeme "ulítnout"úplně mimo hledané řešení.

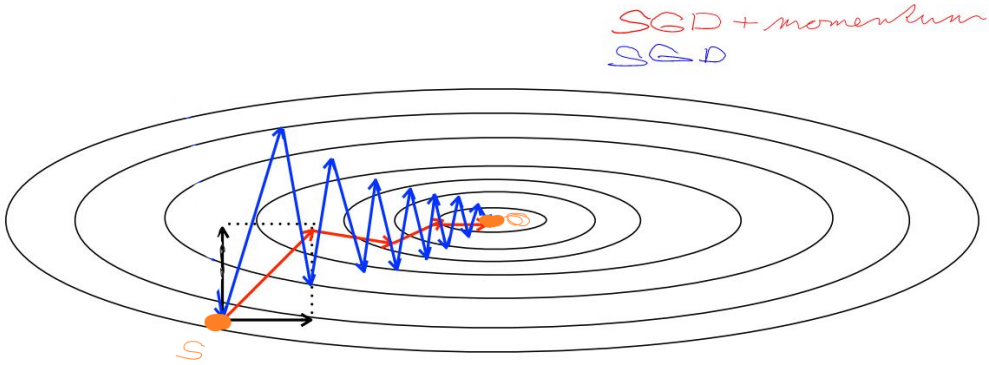
Optimální řešení, kterého vždy nemusíme dosáhnout.

.2 SGD + momentum

Vylepšením předchozího SGD algoritmu pomocí hybnosti dosáhneme lepších výsledků a tedy rychlejšího učení.

- Popište na předchozím příkladě nebo vymyslete nové případy, ve kterých dosáhneme zlepšení v problémových bodech.
V prvním případě při vhodné inicializaci za pomoci rychlosti "přejedeme"lokální minimum a dostaneme se do globálního minima.
V případě špatné inicializace na ploché části křivky opět zůstane algoritmus zaseknutý. Ovšem pokud bude inicializace vhodná, je možné to pomocí hybnosti opět "přejet".

- Máte zadanou funkci následujícím grafem. Je vyznačena startovní pozice S a optimální řešení O. Načrtněte jaké řešení byste očekávali pomocí SGD a SGD s hybností.



.3 AdaGrad

Adagrad je dalším vylepšením, které je založeno na základě gradientního hyperparametru.

- Jaký je hlavní rozdíl od předešlých algoritmů založených na gradientním sestupu?

  V průběhu tohoto algoritmu se přizpůsobují hodnoty learning rate vůči vstupním parametrům. Pro řídké parametry jsou aktualizace větší než pro husté, kdy jsou aktualizace learning rate menší.

- Jaké jsou výhody a nevýhody tohoto algoritmu? Hlavní výhodou AdaGradu je že eliminuje nutnost ladit learning rate. Většina implementací používá základní hodnotu 0.01 a nechává ji.

  Hlavní nevýhodou je akumulace kvadrátů gradientu v jmenovateli. V momentě, kdy jsou přírůstky stále pozitivní, suma se naakumuluje a poroste v průběhu testování. To způsobí, že se learning rate zmenší a nakonec se stane nekonečně malý, v tomto okamžiku již algoritmus není schopen se učit.

.4 Závěr

Proč si zasloužím více bodů? Protože jsem shrnul základy gradientních hyperparametrů a vytvořil jsem to ve formě, která se dá zadat jako test. Tím pádem to můžete příští rok použít při testu a v případě stížností to svést na to, že to dělali studenti.:D Zároveň jsem to jako správný student dělal s nejlepším vědomím a svědomím na poslední chvíli do ranních hodin a přesto jsem to neodbyl.

Zdroje:

[1] Gradient descent, how neural networks learn | Chapter 2, Deep learning - URL https://www.youtube.com/watch?v=IHZw
w

[2] 23. Accelerating Gradient Descent (Use Momentum) - URL https://www.youtube.com/watch?v=wrEcHhoJxjM

[3] Přednáškové materiály

[4] Du, S. S.; Lee, J. D.; Tian, Y.; aj.: Gradient Descent Learns One-hidden-layer CNN: Don't be Afraid of Spurious Local Minima. 2017. URL https://arxiv.org/pdf/1712.00779.pdf

[5] - Ruder, S.: An overview of gradient descent optimization algorithms. CoRR, ročník abs/1609.04747, 2016. URL http://arxiv.org/abs/1609.04747