1 Gradient Descent and Its Extensions

Below is an introductory overview of Gradient Descent and three key extensions (rozszerzeń): Momentum, Adagrad, and Adam. The goal is to minimize a scalar function $f(\mathbf{w})$ with respect(względem) to the parameter vector $\mathbf{w}$.

1.1 What is Gradient Descent and its main problem?

Goal: Find $\mathbf{w}$ that minimizes $f(\mathbf{w})$.

Update Rule:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla f(\mathbf{w}_t)$$

where

- $\nabla f(\mathbf{w}_t)$ is the gradient of f evaluated at $\mathbf{w}_t$,
- $\alpha > 0$ is the learning rate.

Issue: Vanilla (standardowy) Gradient Descent can oscillate along steep (stromych) directions or converge slowly when different parameters have different scales.

1.2 What is Momentum?

Motivation: Reduce oscillations and speed up convergence (przyśpieszenie zbieżności) by "remembering" previous gradients. We introduce (wprowadzamy) a velocity term (człon prędkości) v_t that accumulates gradients via an exponential decay (zaniku wykładniczego).

Update Equations:

1. **Velocity Update:**
$$v_t = \beta v_{t-1} + \alpha \nabla f(w_{t-1})$$
2. **Parameter Update:**
$$w_{t+1} = w_t - v_t$$
where $\beta \in [0, 1)$ is the momentum coefficient (współczynnik) (sometimes denoted (oznacza) γ in some references(źródłach)). A larger β applies more smoothing (wygładzenie) from previous steps.

$$\begin{aligned}
v_{t+1} &= \beta v_t + \alpha \nabla f(w_t) \qquad 0 < \beta < 1 \\
w_{t+1} &= w_t - v_t \\
v_t &= \beta v_{t-1} + \alpha \nabla f(w_{t-1}) \\
v_{t+1} &= \beta(\beta v_{t-1} + \alpha \nabla f(w_{t-1})) + \alpha \nabla f(w_t) \\
v_{t+1} &= \beta^2 v_{t-1} + \alpha\beta \nabla f(w_{t-1}) + \alpha \nabla f(w_t) \\
v_{t+1} &= \beta^n v_{t-n+1} + \alpha \sum_{i=0}^{n} \beta^i \nabla f(w_{t-i})
\end{aligned}$$

1.3 What is Adagrad?

Motivation: Adapt(dostosować) the step size for each parameter dimension (wymiary parametru). Parameters that have received large gradients in the past get smaller future updates, while parameters with smaller historical gradients get comparatively (względnie) larger updates.

Key Idea: Accumulate the sum of squared gradients in a vector G_t.

1. **Accumulator Update:**

$$G_t = G_{t-1} + (\nabla f(w_t))^2$$

(the square here is applied (obliczany) element-wise (element po elemencie)).

2. **Parameter Update:**

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{G_t + \varepsilon}} \nabla f(w_t)$$

$$G_t = \sum_{i=0}^{t} (\nabla f(w_i))^2$$

where ε is a small constant (e.g. 10^{-8}) to avoid division by zero.

This per-parameter (dla każdego parametru z osobna) scaling is especially effective when some features are sparse (rzadkie) or have different magnitudes (wielkości).

1.4 Adam

Motivation: Combines (łączy) the benefits of Momentum and Adagrad:

- **Momentum:** Keep (zachowuje) an exponential moving average (średnią) of gradients (the *first moment*).
- **Adagrad-like adaptation:** Keep an exponential moving average of squared gradients (the *second moment*).

Update Equations:

1. **First Moment (mean of gradients):**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla f(w_t)$$

2. **Second Moment:**

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla f(w_t))^2$$

3. **Bias Correction:**

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \qquad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

4. **Parameter Update:**

$$w_{t+1} = w_t - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

where $\beta_1, \beta_2 \in [0, 1)$ are decay rates (współczynnik zaniku)(typically $\beta_1 = 0.9$ and $\beta_2 = 0.999$), controlling how quickly old information is discarded (wygaszana), and ε (typically 10^{-8}) ensures numerical stability.

Bias Correction (korekcji) Explanation: The moving averages m_t and v_t are initialized with zeros, causing them to be biased (przesunięte) toward zero, especially during the early steps of training. The bias correction terms adjust (warunki dopasowują się) for this initialization bias by scaling the estimates. As t increases, the correction becomes less significant since $(1 - \beta^t) \to 1$ as $t \to \infty$. This correction ensures more accurate (dokładny) adaptive learning rates, particularly in the early stages of optimization.

agh-stochastic-ml

Recommended Reading

1. Kingma, Diederik P., and Jimmy Ba. *Adam: A method for stochastic optimization*.

2. Ruder, Sebastian. *An overview of gradient descent optimization algorithms*.

3. Sutskever, Ilya, et al. *On the importance of initialization and momentum in deep learning*.

4. *Adam — PyTorch 2.6 documentation*.

5. AdamW: Loshchilov, Ilya, and Frank Hutter. *Decoupled weight decay regularization*.