Deep Learning course

Session 8 – Optimizers

E. Francisco Roman-Rangel
edgar.roman@alumni.epfl.ch

CInC-UAEM. Cuernavaca, Mexico. September 22nd, 2018.

Outline

SGD limitations

- SGD (Stochastic gradient descent) slows down around ravines.
- SGD oscillates across the slopes of the ravine.
- Limited progress towards the local minimum.

Image 2: SGD without momentum

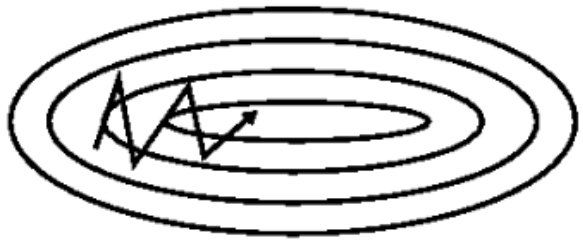


Image 3: SGD with momentum

Qian, 1999. "On the momentum term in gradient descent learning algorithms".

Momentum

$$\omega_t = \omega_{t-1} - \alpha z_t$$

$$z_t = \beta z_{t-1} + \nabla \mathcal{L}(\omega_{t-1})$$

- Helps accelerating SGD by keeping momentum.
- Adds a fraction of the previous gradient to the current one.
- Sort of exponential smoothing (moving average).
- Increases for parameters whose gradients keep directions.
- Reduces for parameters whose gradients change directions.
- $\beta = 0.9$.

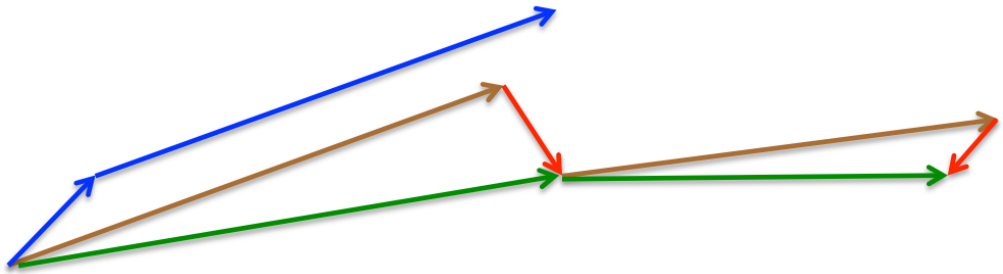
https://distill.pub/2017/momentum/

Nesterov

$$\omega_t = \omega_{t-1} - \alpha z_t$$

$$z_t = \beta z_{t-1} + \nabla \mathcal{L}(\omega_{t-1} - \beta z_{t-1})$$

- Gives a notion of where we go.
- Corrects high magnitude momentum.
- Allows slowing down before slopes.



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

Outline

Adagrad

- Adapts the learning rate to the parameters.
- Low learning rates for parameters of common features.
- High learning rates for parameters of uncommon features.
- Ideal for sparse data and word embeddings.

$$\omega_{t,i} = \omega_{t-1,i} - \frac{\eta}{\sqrt{\sum_{j=1}^{t-1} \nabla \mathcal{L}(\omega_{t,i})^2 + \epsilon}} \nabla \mathcal{L}(\omega_{t,i})$$

Duchi et al., 2011. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization".

Adadelta

- Avoids storing the history gradients.
- Restricts them to a window fixed length.

$$\omega_t = \omega_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \nabla \mathcal{L}(\omega_t)$$

where,

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma) \nabla \mathcal{L}(\omega_t)$$

Zeiler, 2012. "ADADELTA: An Adaptive Learning Rate Method".
RMSprop: a variant by Hinton (unpublished).

Adam

- Also adaptive for each parameter.
- Average of past squared gradients v_t (Adadelta).
- Average of past gradients m_t (Momentum).

$$\omega_t = \omega_{t-1} - \frac{\eta}{\sqrt{v_{t-1}} + \epsilon} m_{t-1}$$

where,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla \mathcal{L}(\omega_t)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \nabla \mathcal{L}(\omega_t)^2$$

Kingma & Ba, 2015. "Adam: a Method for Stochastic Optimization".
Other variants: AdaMax, Nadam, AMSGrad.

Outline

Animations.

To know more

Ruder, 2016. "An overview of gradient descent optimization algorithms".
https://arxiv.org/abs/1609.04747

Thank you.

Q&A