1 Notation

t $\in [1, T]$ - time step/number of iterations/examples/minibatches

d - dimensionality of the parameters vector

ϵ - small constant for numerical stability

$\Theta_t \in \mathbb{R}^d$ - parameter vector at time **t**, of dimensionality **d**

$L_t(\Theta)$ - loss function at time **t** (e.g. different minibatch in each timestep)

$\eta \in (0, 1]$ - learning rate, step size

∇ - **nabla**, not delta

$\odot$ - a dot in a circle, elementwise product (Hadamard product)

$g_t = \nabla L_t(\Theta_t$ - gradient in time **t**

$g_t^2 = g_t \odot g_t$ - 'notation abuse'

$\rho \in [0, 1); \mu \in [0, 1)$ - some coefficients

initialization - assume initialization with **zeros** unless stated otherwise

2 SGD

Stadnard formulation

$$\Theta_{t+1} = \Theta_t - \eta \nabla L_t(\Theta_t) \tag{1}$$

Ordinary Momentum

$$v_{t+1} = \mu v_t - \eta \nabla L_t(\Theta_t); \mu \in [0, 1) \tag{2}$$

$$\Theta_{t+1} = \Theta_t + v_{t+1}$$

Nesterov Accelerated Gradient (NAG)

$$v_{t+1} = \mu v_t - \eta \nabla L_t(\Theta_t + \mu v_t); \mu \in [0, 1) \tag{3}$$

$$\Theta_{t+1} = \Theta_t + v_{t+1}$$

3 Dimension-wise Adaptive Methods

Adagrad[1]

$$G_t = \sum_{i=1}^{t} g_t^2 \tag{4}$$

$$\Theta_{t+1} = \Theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

RMSProp[2]

$$G_t = 0.9 * G_{t-1} + 0.1 * g_t^2 = 0.1 \sum_{i=1}^{t} 0.9^{T-i} g_t^2 \tag{5}$$

$$\Theta_{t+1} = \Theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

Adadelta[3]

$$G_t = \rho G_{t-1} + (1-\rho) g_t^2 = (1-\rho) \sum_{i=1}^{t} \rho^{t-i} g_t^2 \tag{6}$$

$$\overline{\Delta}_{t-1} = \rho \overline{\Delta}_{t-2} + (1-\rho) \Delta\Theta_{t-1}^2$$

$$\Delta\Theta_t = \frac{\sqrt{\overline{\Delta}_{t-1} + \epsilon}}{\sqrt{G_t + \epsilon}} \odot g_t$$

$$\Theta_{t+1} = \Theta_t + \Delta\Theta_t$$

3.1 Worth reading perhaps

- Adam[4]
- vSGD maybe [5]
- CoCob [6]
- a blogpost about momentum

References

[1] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report UCB/EECS-2010-24, EECS Department, University of California, Berkeley, Mar 2010.

[2] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[3] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[5] Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 343–351, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[6] Francesco Orabona and Tatiana Tommasi. Backprop without learning rates through coin betting. *CoRR*, abs/1705.07795, 2017.