CS F425: Deep Learning

14 Optimizations For Neural NW

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 21, 2023 ON-CAMPUS Campus @ BITS-Pilani [Jan-May 2023]

http://ktiwari.in/dl

Training a Neural Network

Least Absolute Deviation (LAD): $\frac{1}{m}\sum_{i=1}^{m}|o_i - \hat{y}_i|$

Least Square Error (LSE): $\frac{1}{m}\sum_{i=1}^{m}(o_i - \hat{y}_i)^2$

Cross Entropy Loss (CEL): $-\sum_{i=1}^{c} o_i \log(\hat{y}_i)$

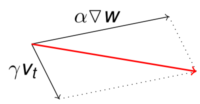
Training objective is to reduce ERROR

What we can try with multiple epochs?

- Gradient Descent (or batch GD) Cauchy 1847
- Stochastic Gradient Descent Update weights for every example
- Mini-Batch Gradient Descent
- Backpropogation Rumelhart *et al* 1986[1]

[1] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533-536.

Momentum Based Gradient Descent [2]



- While updating the weights a γ fraction of the previous update is added to the current

$$v_t = \gamma v_{t-1} + \alpha\nabla w_{t-1}$$

α being the learning rate

New weight would be $w_t = w_{t-1} - v_t$

Deep and recurrent neural networks were considered to be almost impossible to train using stochastic gradient descent with momentum. This paper shows that when stochastic gradient descent with momentum uses a well-designed random initialization and a particular type of slowly increasing schedule for the momentum parameter, it can train both DNNs and RNNs

[2] Sutskever, Ilya, et al. "On the importance of initialization and momentum in deep learning." International conference on machine learning. PMLR, 2013.

Nesterov Accelerated Gradient (NAG)

Momentum could overshoot the local minima

- NAG uses look ahead to decide if the learning pace shold be slowed to reach local minima to avoid oscillations.
- Pick a future weight

$$w_{fut} = w_{t-1} - \gamma.v_{t-1}$$

- See gradient there

$$v_t = \gamma.v_{t-1} + \alpha\nabla w_{fut}$$

$$w_t = w_{t-1} - v_t$$

- Nesterov accelerated gradient [3] is superior to momentum for conventional optimization [4]

[3] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In Doklady AN USSR, volume 269, pp. 543–547, 1983. Sutskever *et al* 2013

[4] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In International conference on machine learning, pp.1139–1147, 2013.

AdaGrad

- Uses adaptive learning rate
- Take different step size in different dimentions.
- Velocity

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

- Update weight using gradient only with different learning weight

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{v_t + \epsilon}}\nabla w$$

Every weight have now different learning rate.

The paper [5] presents a new family of subgradient methods that dynamically incorporate knowledge of the geometry of the data observed in earlier iterations to perform more informative gradient-based learning.

[5] Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of machine learning research 12.7 (2011).

RMSProp [6]

AdaGrad have an issue; v_t becomes large and tend to kill learning rate

- Let us try damped summation
- Moving average

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)(\nabla w)^2$$

β controls how much the current value is significant.

- Update weight using gradient only with different learning weight

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}}\nabla w$$

- RMS is root mean square

[6] Hinton, G., Srivastava, N., and Swersky, K. Lecture 6d - a separate, adaptive learning rate for each connection. Slides of Lecture Neural Networks for Machine Learning, 2012.

AdaDelta [7]

- May the slding window better suite
- Moving average over window

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)(\nabla w)^2$$

- $RMS[g]_t = \sqrt{E[g^2]_t + \epsilon}$
- g_t is gradient at time t

Algorithm 1 Computing ADADELTA update at time t
Require: Decay rate ρ, Constant ϵ
Require: Initial parameter x_1
1: Initialize accumulation variables $E[g^2]_0 = 0$, $E[\Delta x^2]_0 = 0$
2: **for** $t = 1 : T$ **do** %% Loop over # of updates
3: Compute Gradient: g_t
4: Accumulate Gradient: $E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho)g_t^2$
5: Compute Update: $\Delta x_t = -\frac{\text{RMS}[\Delta x]_{t-1}}{\text{RMS}[g]_t} g_t$
6: Accumulate Updates: $E[\Delta x^2]_t = \rho E[\Delta x^2]_{t-1} + (1-\rho)\Delta x_t^2$
7: Apply Update: $x_{t+1} = x_t + \Delta x_t$
8: **end for**

-
$$E[\triangle x^2]_t = \beta E[\triangle x^2]_{t-1} + (1-\beta)(\triangle x_t)^2$$

- Weight update

$$x_t = x_{t-1} + \triangle x_t$$

The method dynamically adapts over time using only first order information and has minimal computational overhead beyond vanilla stochastic gradient descent. The method requires no manual tuning of a learning rate and appears robust to noisy gradient information, different model architecture choices, various data modalities and selection of hyperparameters. We show promising results compared to other methods on the MNIST digit classification task using a single machine and on a large scale voice dataset in a distributed cluster environment.

[7] Zeiler, Matthew D. "Adadelta: an adaptive learning rate method." arXiv preprint arXiv:1212.5701 (2012).

Adeptive Moment Estimation (Adam) [8]

- Combining RMSProp and Momentum
- Decaying average of first and second moment

$$m_i = \beta_1 . m_{i-1} + (1-\beta_1).\nabla w \qquad v_i = \beta_2 . v_{i-1} + (1-\beta_2).(\nabla w)^2$$

- Corrected estimates are obtained by

$$\hat{m}_i = \frac{m_i}{1-\beta_1^t} \qquad \hat{v}_i = \frac{v_i}{1-\beta_2^t}$$

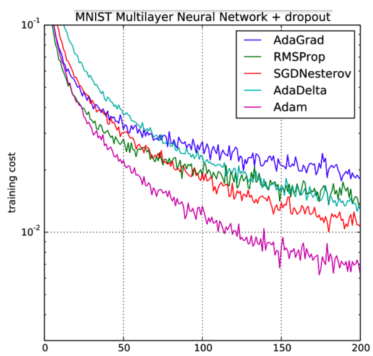
- New weights are obtained by

$$w_{i+1} = w_i - \eta \frac{\hat{m}_i}{\sqrt{\hat{v}_i} + \epsilon}$$

The method combines the advantages of two recently popular optimization methods: the ability of AdaGrad to deal with sparse gradients, and the ability of RMSProp to deal with non-stationary objectives.

[8] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

Adam [9]



Adam makes faster progress in terms of both the number of iterations and wall-clock time.

[9] Kingma, Diederik P and Ba, Jimmy, **Adam: A method for stochastic optimization**, ICLR 2015

Thank You!

Thank you very much for your attention!