Neural networks 2

Blanca Cano Camarero

Universidad Autónoma de Madrid

December 16, 2022

Overview

Gradient descent

We start from a random w_0 and compute

$$w_{k+1} = w_k - \rho_k \nabla_w e(w_k) \tag{1}$$

Considerations:

- Subset or all the training data (*batch*,*mini-batch* or *online*).
- Learning rate tunning.

Which is the best size? (see Masters and Luschi (2018))

Learning rate

We want to ensure

$$e(w_{k+1}) < e(w_k) \tag{2}$$

Line minimization

$$\rho_k^* = argmin_\rho e(w_k - \rho_k \nabla_w e(w_k)) \tag{3}$$

The GD methods are called first order methods.

Netwon's Methods

(q generalization of the error function).

$$q(w) = aw^2 + bw + c \tag{4}$$

with $a > 0$.

$$0 = q'(w + \Delta w) = 2a(w + \Delta w) + b. \tag{5}$$

So $\Delta w = -\frac{2a+b}{2a}$.

$$w^* = w - \frac{2aw + b}{2a} = w - \frac{1}{q''(w)}q(w). \tag{6}$$

Newton's method

$$w_{k+1} = w^k - \rho_k \frac{1}{f''(w_k)} f'(w_k) \tag{7}$$

Why we maintain the LR. Considerations:

- $\frac{1}{f''(w_k)}$ acts as a self adjusting learning rate.
- Parabola sharp $a >> 0$ overstep in GD but $f''(w_k)$ is also big.

Multidimensional Newton's Method

Now $w \in \mathbb{R}^d$, the Taylor expansion of e at an optimum $w^* = w - \Delta$.

$$e(w) \approx e(w^*) + \nabla_w e(w^*)(w - w^*) + \frac{1}{2}(w - w^*)^t H(w^*)(w - w^*). \tag{8}$$

Since $\nabla_w e(w^*) = 0$ and using Taylor expansion again

$$\nabla e(w) \approx H(w^*)(w - w^*) \tag{9}$$

$$w^* \approx w - H(w^*)^{-1}\nabla_w e(w^*) = w - H(w^*)^{-1}H(w^*)(w - w^*). \tag{10}$$

Gauss Newton Matrix

In a general case

$$\nabla e(w) = E[\nabla f(x|w)(f(x|w) - y)] \tag{11}$$

$$\nabla^2 e(w) = E[\nabla^2 f(x|w)(f(x|w) - y)] + E[\nabla f(x|w)\nabla f(x|w)^t] \tag{12}$$

$w \approx w^*$, $f(x|w) \approx y$, $f(x|w) - y \approx 0$.

Gauss newton approximation

$$H_{(i,j)(p,q)}(w) = \left(\frac{\partial^2 e}{\partial w_{ij} \partial w_{pq}}(w) \right) \approx \left(E\left[\frac{\partial f}{\partial w_{pq}} \frac{\partial f}{\partial w_{ij}} \right] \right)_{(i,j)(p,q)} \tag{13}$$

- H is Fisher's information matrix.
- J is semidefinite invertible.
- (Not necessary invertible).

Often its diagonal is consider.

GN

Solve if no es invertible

Levenberg Marquardt

- Gradient descent away from w^*

Conjugate gradient methods

See wikipedia: Conjugate gradient methods (2023)

- Iterative algorithm, form sparse systems.
- Can be as an example of the Gram Schmidt orthonormalization.
- Keep the previous directions.

Conjugate gradient methods

Let f be quadratic function $f(x) = \frac{1}{2}x^T Hx - x^t b$ with H symmetric and positive definite.
Starting guess x_0, this means $p_0 = b - Ax_0$.
Let r_k be the residual $r_k = b - Ax_k$

$$p_k = r_k - \sum_{i<k} \frac{p_i^T Hr_l}{p_i Hp_i} p_i. \tag{14}$$

Following this direction the next optimal location is given by

$$x_{k+1} = x_k + \alpha p_k \tag{15}$$

with

$$\alpha = \frac{p_k^T (b - Hx_k)}{p_k^T Hp_k}. \tag{16}$$

Quasi Newton method

Methods to either find zeroes or local maxima and minima of function, as an alternative to Newton's method.
The can be used if the Jacobian or Hessian is unavailable or to expensive to compute at every iteration.
The hessian approximation B is chosen to satisfy

$$\nabla f(x + \Delta x) = \nabla f(x) + B\Delta x, \tag{17}$$

which is called the secant equation.
There are plenty iterative formulas see wikipedia: Quasi Newton method (2023).

Limited memory Broyden Fletcher Goldfarb Shannon

Using limited amount of computer memory.
wikipedia: Limited-memory BFGS (2023)

Momentum

Take previous state into consideration, give gradient descent a short-term memory.
$\Delta_k = w_k - w_{k-1}$

$$w_{k+1} = w_k - \rho_k \Delta_w e(w_k) + \mu_k \Delta_k \tag{18}$$

[Story about GD is a man walking down hill vs heavy ball down]
For further information see Works (2023)

Rewriting momentum

Defining:

$$\Delta_{k+1} = -\rho_w e(w_k) + \mu_k \Delta_k \tag{19}$$

Applying

$$w_{k+1} = w_k + \Delta_{k+1} \tag{20}$$

Nesterov's Accelerated Gradient

Variant of momentum
Defining:

$$\tilde{\Delta}_{k+1} = -\rho_w e(w_k + \mu_k \tilde{\Delta}_k) + \mu_k \tilde{\Delta}_k \tag{21}$$

Applying

$$w_{k+1} = w_k + \tilde{\Delta}_{k+1} \tag{22}$$

- In convex optimization it improves GD.
- Is often highly effective in Deep Network training.

Adam

From the article Kingma and Ba (2014).

- An algorithm for first-order gradient-based optimization of stochastic objective functions,
- based on adaptive estimates of lower-order moments,
- computationally efficient,
- has little memory requirements,
- is invariant to diagonal rescaling of the gradients,
- and is well suited for problems that are large in terms of data and/or parameters.
- The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients.
- The hyper-parameters have intuitive interpretations and typically require little tuning.

Adam explanation

Each step t Adam uses a new random mini- batch to

- Update exponential smoothen moments:
- compute estimator
- update weights

Adam: moments

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g \tag{23}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g^2 \tag{24}$$

where $v_0 = 0$ and g^2 indicates the elementwise square.

$$v_t = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} g_i^2 \tag{25}$$

Adam: Initialization bias correction

$$E[v_t] = E\left[(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}g_i^2\right] \tag{26}$$

$$= E[g_t^2](1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i} + \zeta \tag{27}$$

$$= E[g_t^2](1-\beta_2^t) + \zeta. \tag{28}$$

where $\zeta = 0$ is the true second moment $E[g_i^2]$ is stationary;

Adam: Initialization bias correction

Since we have

$$E[m_t] \approx (1 - \beta_1^t) E[g_t]; \tag{29}$$

$$E[m_t] \approx (1 - \beta_2^t) E[g_t^2]. \tag{30}$$

We compute the bias corrections

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}; \tag{31}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}. \tag{32}$$

Adam: weights update

$$W_t = W_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{33}$$

Understanding Adam

- Default values $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ usually work fine.
- Interpretation as a normalized GD.

Main points of the article

- 2006 deep neural network show their power with new initialization an training mechanisms.
- Logistic sigmoid activation is unsuited for deep networiks with random initialization because of its mean value, which can drive especially the top hidden layer into saturation.

Questions?

Why are deeper networks better than wider?
Intuitively approach piecewise polynomial functions.

References I

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks, 2018. URL https://arxiv.org/abs/1804.07612.

wikipedia: Conjugate gradient methods. Conjugate gradient methods, 2023. URL https://en.wikipedia.org/wiki/Conjugate_gradient_method.

wikipedia: Limited-memory BFGS. Limited-memory bfgs, 2023. URL https://en.wikipedia.org/wiki/Limited-memory_BFGS.

wikipedia: Quasi Newton method. Quasi newton method, 2023. URL https://en.wikipedia.org/wiki/Quasi-Newton_method.

Why Momentum Really Works. Why momentum really works, 2023. URL https://distill.pub/2017/momentum/.