Intro to Deep Learning
RECAP OF SESSION ON 7 JAN, 2019

Rauniyar & Reza

Questions:

What is RMSProp?

- speeds up Gradient Descent to perform better in the nonconvex setting
- Dampens the oscillations, but in a different way than momentum
- RMS prop also takes away the need to adjust learning rate, and does it automatically

[Goal is to increase w rate & decrease b rate]

On iteration k:

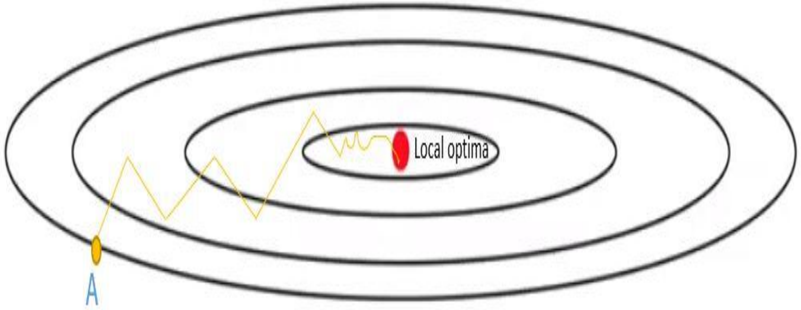
Compute dw and db on current mini-batch:

$Sdw = \beta . Sdw + (1-\beta)dw^2$ <- small

$Sdb = \beta . Sdb + (1-\beta)db^2$ <- large

$w = w - \alpha \, dw/\sqrt{Sdw}$

$b = b - \alpha \, db/\sqrt{Sdb}$



How does batch normalization works?

- Normalizes output of a previous activation layer by subtracting the batch mean & dividing by the batch standard deviation

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

What is AdaGrad?

Algorithm 8.4 The AdaGrad algorithm

Require: Global learning rate ϵ
Require: Initial parameter $\boldsymbol{\theta}$
Require: Small constant δ, perhaps 10^{-7}, for numerical stability
Initialize gradient accumulation variable $\boldsymbol{r} = \boldsymbol{0}$
while stopping criterion not met **do**
Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$.
Accumulate squared gradient: $\boldsymbol{r} \leftarrow \boldsymbol{r} + \boldsymbol{g} \odot \boldsymbol{g}$.
Compute update: $\Delta\boldsymbol{\theta} \leftarrow -\frac{\epsilon}{\delta + \sqrt{\boldsymbol{r}}} \odot \boldsymbol{g}$. (Division and square root applied element-wise)
Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$.
end while

- Parameters with the largest partial derivative of the loss have rapid decrease in their learning rate, & vice versa
- Net effect is greater progress in the more gently sloped directions

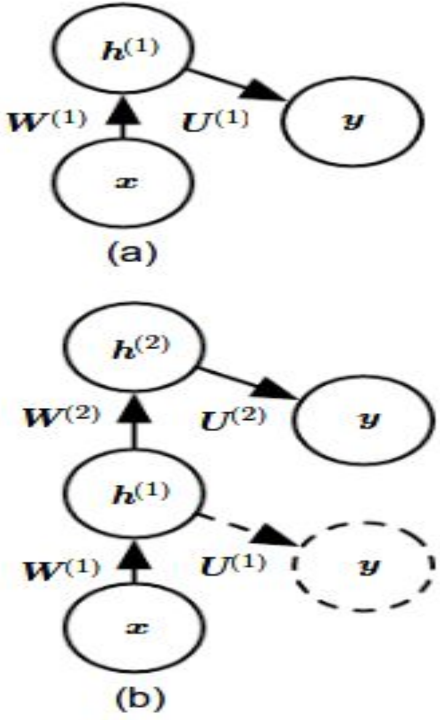
How does (greedy) supervised pre-training work?

- break a problem into many components, then solve for the optimal version of each component in isolation
- combines the optimized versions of the sub

  networks into a new model that solves the original

  problem

- can be computationally much cheaper



How does curriculum learning work?

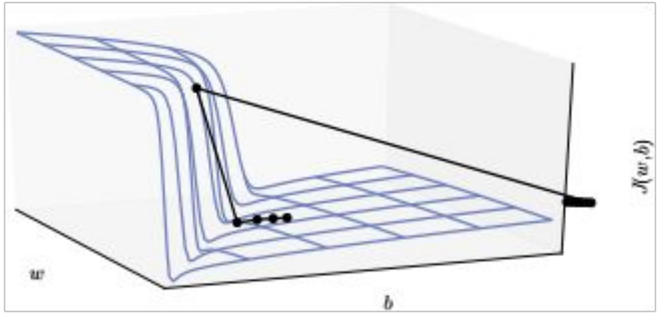
- Can be interpreted as a continuation method
- Learning from simple concept and progress to learning more complex concept

How does (block) coordinate descent work?

- Break optimization problem into separate optimization with respect to single variable.

Why are cliffs and long term dependencies challenging for optimization/learning?

- Facing with steep gradient region in loss function
- The gradient update step can move the parameters far
- Solution: Gradient clipping
- The gradient update only the optimal direction within small region



- Facing with computational graph with extremely deep
- Cause in Feedforward network and recurrent networks

$$\boldsymbol{W}^t = \left(\boldsymbol{V}\operatorname{diag}(\boldsymbol{\lambda})\boldsymbol{V}^{-1}\right)^t = \boldsymbol{V}\operatorname{diag}(\boldsymbol{\lambda})^t\boldsymbol{V}^{-1}.$$

What is stochastic gradient descent with momentum?

$$\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{f}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)}) \right),$$
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}.$$

What is Nesterov momentum and how is it different to standard momentum?

- The difference between is where the gradient is evaluated

$$\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \nabla_{\boldsymbol{\theta}} \left[\frac{1}{m} \sum_{i=1}^{m} L\left(\boldsymbol{f}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta} + \alpha \boldsymbol{v}), \boldsymbol{y}^{(i)} \right) \right],$$
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v},$$