Foundations of Machine Learning (CS 725)

FALL 2024

Lecture 15:

- Optimizers/Initialization

Instructor: Preethi Jyothi

Optimization Algorithms (I)

- "**SGD with Momentum**" weight update rule:

$$\mathbf{g}_t = \nabla_{\mathbf{w}} L(\mathbf{w}_{t-1})$$

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + (1-\beta)\nabla_{\mathbf{w}} L(\mathbf{w}_{t-1}) \qquad \text{for } 0 \leq \beta < 1$$

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \mathbf{v}_t$$

- Smooths parameter updates with exponentially decaying weights

Optimization Algorithms (II)

- "**RMSProp (Root Mean Squared Propagation)**" weight update rule:

$$\mathbf{s}_t = \gamma \mathbf{s}_{t-1} + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t$$

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \frac{\eta}{\sqrt{\mathbf{s}_t} + \epsilon} \odot \mathbf{g}_t$$

$$\mathbf{g}_t = \nabla_{\mathbf{w}} L(\mathbf{w}_{t-1})$$

Element-wise multiplication

- Need an adaptive learning rate that adapts to each dimension.

Optimization Algorithms (III)

- "**Adam**" weight update rule: Makes use of both momentum and adaptive learning rate

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + (1-\beta)\mathbf{g}_t$$

$$\mathbf{s}_t = \gamma \mathbf{s}_{t-1} + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t$$

$$\hat{\mathbf{s}}_t \leftarrow \frac{\mathbf{s}_t}{1-\gamma^t} \qquad \hat{\mathbf{v}}_t \leftarrow \frac{\mathbf{v}_t}{1-\beta^t}$$

Bias Correction

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \frac{\eta}{\sqrt{\hat{\mathbf{s}}_t} + \epsilon} \odot \hat{\mathbf{v}}_t$$

NN Weight Initialization

- Do not set all weights/biases to 0. No learning!
- Initialize weights to small random numbers, sampled randomly from a Gaussian or uniform distribution. Set the variance of the distribution as a hyperparameter.
- **Xavier (Glorot) Initialization [1]**: Popular scheme
  - Randomly sample from $\mathcal{N}(0,\sigma), \sigma = \sqrt{\dfrac{2}{\text{fan\_in} + \text{fan\_out}}}$ where fan_in is the number of input units and fan_out is the number of output units
  - Scaling is to make sure that the variance of input and output gradients (more or less) remains the same

[1] Glorot & Bengio, 2010, "Understanding the difficulty of training deep feedforward neural networks"
Explanation of Glorot initialization: https://andyljones.tumblr.com/post/110998971763/an-explanation-of-xavier-initialization