CSE 5523: Lecture Notes 13
Gradient Descent Optimization

Contents

13.1 (Mini-batch) stochastic gradient descent 1
13.2 Adaptive optimization [Kingma and Ba, 2015] 1

We saw gradient descent minimize expected loss over models $\mathbf{W}$ of predictions $f_{\mathbf{W}}(\mathbf{x})$ using data $y, \mathbf{x}$:

$$\mathbf{W}^{(i)} = \mathbf{W}^{(i-1)} - \nabla_{\mathbf{W}^{(i-1)}} \frac{1}{N} \sum_{n \in \{1 \ldots N\}} \mathsf{L}_{\mathrm{NL}}(y_n,\ f_{\mathbf{W}^{(i-1)}}(\mathbf{x}_n))$$

(Gradient, using harp-shaped 'nabla' symbol ∇, is just the derivative in multiple dimensions.)

It can converge on a global optimum, but it can be slow.

13.1 (Mini-batch) stochastic gradient descent

One problem with simple 'batch' gradient descent is it only updates after predicting all data.

We can therefore speed optimization by updating after every 'mini-batch' of B examples:

$$\mathbf{W}^{(i)} = \mathbf{W}^{(i-1)} - \nabla_{\mathbf{W}^{(i-1)}} \frac{1}{B} \sum_{n \in \{iB-B+1 \ldots iB\}} \mathsf{L}_{\mathrm{NL}}(y_n,\ f_{\mathbf{W}^{(i-1)}}(\mathbf{x}_n))$$

(For simplicity, we define our example counter n to wrap around after reaching N.)

13.2 Adaptive optimization [Kingma and Ba, 2015]

Gradient descent can sometimes jump back and forth across long optima.

Optimization algorithms like ADAM attempt to mitigate this by using 'intertia'.

User parameters:

- β_M: inertia of old mean (recommended .9)
- β_V: inertia of old variance (recommended .999)
- α: learning rate (recommended .001)
- ϵ: avoid division by zero (recommended 10^{-8})

It estimates the mean $\mathbf{M}$ and variance $\mathbf{V}$ with each iteration i of gradient descent:

$$\mathbf{M}^{(0)} \stackrel{\text{def}}{=} \mathbf{0}$$

$$\mathbf{V}^{(0)} \stackrel{\text{def}}{=} \mathbf{0}$$

$$\mathbf{M}^{(i)} \stackrel{\text{def}}{=} \beta_M \, \mathbf{M}^{(i-1)} + (1 - \beta_M) \left(\nabla_{\mathbf{W}^{(i-1)}} \frac{1}{N} \sum_{n \in \{1 \ldots N\}} \mathsf{L}_{\text{NL}}(\, y_n, \, f_{\mathbf{W}^{(i-1)}}(\mathbf{x}_n) \,) \right)$$

$$\mathbf{V}^{(i)} \stackrel{\text{def}}{=} \beta_V \, \mathbf{V}^{(i-1)} + (1 - \beta_V) \left(\nabla_{\mathbf{W}^{(i-1)}} \frac{1}{N} \sum_{n \in \{1 \ldots N\}} \mathsf{L}_{\text{NL}}(\, y_n, \, f_{\mathbf{W}^{(i-1)}}(\mathbf{x}_n) \,) \right)^2$$

Adjust to counteract initial bias:

$$\hat{\mathbf{M}}^{(i)} \stackrel{\text{def}}{=} \frac{\mathbf{M}^{(i)}}{1 - (\beta_M)^i}$$

$$\hat{\mathbf{V}}^{(i)} \stackrel{\text{def}}{=} \frac{\mathbf{V}^{(i)}}{1 - (\beta_V)^i}$$

Use in update:

$$\mathbf{W}^{(i)} \stackrel{\text{def}}{=} \mathbf{W}^{(i-1)} - \frac{\alpha}{\sqrt{\hat{\mathbf{V}}^{(i)}} + \epsilon} \hat{\mathbf{M}}^{(i)}$$

References

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.