Optimization for neural networks

Toby Dylan Hocking

Supervised machine learning

- Goal is to learn a function $f(\mathbf{x}) = y$ where $\mathbf{x} \in \mathbb{R}^p$ is an input/feature vector and y is an output/label.
- $\mathbf{x} =$image of digit/clothing, $y \in \{0, \dots, 9\}$ (ten classes).
- $\mathbf{x} =$vector of word counts in email, $y \in \{1, 0\}$ (spam or not).

Optimization algorithms vs machine learning algorithms

- In the field of optimization, the ultimate goal is to minimize some function (which we can compute).
- In supervised machine learning our ultimate goal is to minimize prediction error on test set (which we can not compute).
- Instead we must assume train and test data are similar, and then minimize train error.
- Instead of directly minimizing the error function of interest (zero-one loss) our gradient descent algorithms attempt to minimize a differentiable surrogate loss (logistic / cross-entropy).
- But our goal has now been twice modified, so we need regularization in addition to optimization.

Basic gradient descent algorithm

- m is batch size.
- i is observation number in batch.
- $x^{(i)}, y^{(i)}$ are inputs/output.
- θ is full set of neural network parameters (weights + bias).
- Gradient computed via

$$g \leftarrow \frac{1}{m} \nabla_\theta \sum_i L[f(x^{(i)}; \theta), y^{(i)}].$$

Parameters updated for each iteration via

$$\theta \leftarrow \theta - \epsilon g.$$

Decreasing learning rate schedule

Parameters updated for each iteration via learning rate ϵ_k which decreases with the number of iterations k.

$$\theta \leftarrow \theta - \epsilon_k g.$$

To avoid overshooting minima, learning rate decays/decreases until iteration τ, after which it is held constant.

$$\epsilon_k = (1 - \alpha)\epsilon_0 + \alpha\epsilon_\tau.$$

- $\alpha = k/\tau$.
- $\tau \approx$ a few hundred epochs.
- $\epsilon_\tau/\epsilon_0 \approx 0.01$.

Momentum

- Accumulates an exponentially decaying moving average of past gradients.
- Depends on $\alpha \in [0, 1)$.
- For which value of α do we recover the usual gradient descent? (no momentum)
- Common values 0.5, 0.9, 0.99.

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

$$\theta \leftarrow \theta + v.$$