Homework 3

Joseph Camacho

July 11, 2022

Adam: A Method for Stochastic Optimization

In "Adam: A Method for Stochastic Optimization," Kingma & Ba propose a method for stochastic optimization using exponentially decaying weighted averages of the first and second moments of the gradient. It's quite similar to gradient descent, but works much better for stochastic optimization, in which the evaluation function changes over time. The weighted averages are calculated as follows:

1. Initialize $m_0 = v_0 = \vec{0}$, where m_t is the first moment and v_t is the second moment.
2. Initialize constants $\beta_1, \beta_2 \in [0, 1)$. The authors recommend $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
3. At time step t, let g_t be the gradient. Update

$$\begin{aligned} m_t &= \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \\ v_t &= \beta_1 \cdot v_{t-1} + (1 - \beta_1) \cdot g_t^2 \end{aligned}$$

4. These estimates are biased downwards because of the initializations to 0, so fix them by dividing by $1 - \beta_1^t$ or $1 - \beta_2^t$.
5. Take a step in the negative $\hat{m}_t / \sqrt{\hat{v}_t}$ direction, where $\hat{m}_t$ and $\hat{v}_t$ are the unbiased estimates of the first and second moments.

Adam is comparable to other top stochastic optimizers like SGD and AdaBoost. It is guaranteed to find the local minimum for convex online learning problems.

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Early neural networks were rife with overfitting. They modelled their training data well, but failed catastrophically to generalize to unseen new data. One method, called "early stopping", to avoid overfitting is to split the dataset into "train" and "validation" sets, and train on using the "train" data until the network fails to improve on the "validation" data. In their paper, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Srivastava, etc. propose an elegant rival to early stopping: dropout.

Dropout consists of randomly dropping out nodes of the neural network while training. When a node is dropped, all connections to other nodes in the network are cut. It's equivalent to setting the activation of that neuron to 0. At evaluation time, each node is multiplied by 1 - {the probability it was dropped while training}. This causes the expected value of the node to stay the same. Dropout's advantages are that it is fast, extremely simple to implement, and very good at stopping overfitting.

Dropout's robustness relies on the fact that it is, in effect, approximating an exponentially large set of models that share the same nodes. Training on a large collection of models makes dropout much better at generalizing, since it is unlikely that all of them will overfit in the same ways.

The authors of "Dropout" stated that crossing over in sexual reproduction inspired dropout. Crossing over creates robustness in individual genes because they can't rely on a large network of other genes to compensate for their quirks; any individual gene must rely on itself or a small number of other genes. Dropout similarly enforces each node's self-reliance because there are no guarantees that other nodes will be there for it.