Neural Networks

Ankur Mishra

1/22/2018, 2/6/2017

Contents

1 Optimization

1.1 Two strategies

1. Random Search
   Randomly look through weights and record the W that returns the lowest loss. In a nutshell this is guess and check, which pretty much sucks, but slightly better than baseline ($10\% < 15\%$).

2. Gradient Descent
   Computing the slope accross every single direction; derivative. In multiple dimensions, the gradient is a vector of partial derivatives. It can be done numerically and analytically.In general, always use analytic gradients, but check if it is right with numeric gradients; also known as a gradient check.

   (a) Numerically Gradient
   Computing it can be thought as taking really small steps and finding the slope (Difference In Losses/Distance Between Points). Evaluating numerically is approximate and very slow, so don't do it. They are just easy to write.

(b) Analytic Gradient
Taking Derivatives
These are exact and very fast, but tricky to implement.

1.2 Mini-Batch Gradient Descent

Using small sections of training set to compute gradient. This is faster and better for the overall network and also creates noise which is better for optimization. Full-Batch will just give a straight line.
Common Sizes: 32, 64, 128. Usually start with high learning rate and decays over time/epochs.

2 Backpropagation