Subspace search: exact Newton's method, restricted to a linear subspace

August 11, 2017

Abstract

It is well known that Newton's method can lead to far better optimization performance than plain gradient descent. Unfortunately the memory and time required to compute and invert the Hessian, make Newton's method infeasable for the optimization of functions with many parameters, such as those used in deep learning applications. Instead of approximating the inverse Hessian, as is done is quasi-newton methods, we present a method which first reduces the parameter space of the function, and then applies Newton's method without approximation. We compare our method to several quasi-newton methods and other optimizers, on several common models from the deep learning literature. We show that our method converges faster and to lower loss than the alternatives, at the price of minimal overhead.

Line search is a commonly used algorithm to choose a step size in iterative optimization. Broadly it works as follows. Let $f(x)$ be a function over $\mathbb{R}^d$ we aim to minimize, and let x^n be our current candidate. Compute a direction in which we expect the minimium to lie (using either the negative gradient, or Newton's method). Minimize the function along this direction only, using ternary search, and let x^{n+1} be the location of this minimum.

They key insight is that reducing $f(x)$ to a function of one parameter allows us to use a much more powerful search method to find the optimum. It may only be the optimum in a one-dimensional subspace, but as a next step in the iteration, it makes a much better candidate than a step of arbitrary size in our chosen direction.

Our method is a generalization of this idea. Let $f(x)$ be a function over a very high-dimensional space (e.g. the loss of a ConvNet with millions of parameters). We can reasonably compute its gradient, and at most a diagonal approximation of the Hessian, but not much more. Instead of

approximating the Hessian, we first reduce the function to a lower parameter space. Let $\boldsymbol{W} \in \mathbb{R}^{d \times k}$ be a matrix with k basis vectors as its columns (with k many orders of magnitude smaller than d). The function $g(y) = f(\boldsymbol{W}y)$ with $y \in \mathbb{R}^k$ is the function f restricted to the linear subspace spanned by $\boldsymbol{W}$.

We then apply the exact Newton's method to $g(z)$. This requires inversion of the Hessian of $g(z)$, but this is only a $k \times k$ matrix, so if we choose k to be small enough, the computation required will be small.

The question remains how we should choose $\boldsymbol{W}$. In line search based on the gradient, the direction chosen is often characterized as the direction of steepest ascent. For our purposes, it is more instructive to take a slightly different perspective. The gradient gives us the direction of the optimum of the graph, if we approximate it with a tangent linear function. The direction provided by Newton's method, likewise, gives us the direction of the optimum if we approximate the graph locally with a quadratic function. Thus, we are looking for the k-dimensional linear subspace most likely to contain the optimum. Using only the gradient, there is not much more we can say than that the chosen subspace should contain the gradient direction. Since we will be applying Newton's method in the second step, we would prefer the subspace for which the second derivatives are largest.

Related work

- Deep Learning via Hessian-Free Optimization
- Krylov subspace
- subspace optimization

1 Method

- Compute the diagonal of the Hessian h
- Select the k standard basis vectors (i.e. one-hot-vectors) for which the corresponding value h_i is largest, plus the gradient. (ie. we restrict $f(x)$ to those dimensions in which the second derivative is greatest).
-

2 Results

3 Discussion