18
Stochastic Gradient Descent Tricks

Léon Bottou

Microsoft Research, Redmond, WA
leon@bottou.org
http://leon.bottou.org

Abstract. Chapter 1 strongly advocates the *stochastic back-propagation* method to train neural networks. This is in fact an instance of a more general technique called *stochastic gradient descent* (SGD). This chapter provides background material, explains why SGD is a good learning algorithm when the training set is large, and provides useful recommendations.

18.1 Introduction

Chapter 1 strongly advocates the *stochastic back-propagation* method to train neural networks. This is in fact an instance of a more general technique called *stochastic gradient descent* (SGD). This chapter provides background material, explains why SGD is a good learning algorithm when the training set is large, and provides useful recommendations.

18.2 What Is Stochastic Gradient Descent?

Let us first consider a simple supervised learning setup. Each example z is a pair (x, y) composed of an arbitrary input x and a scalar output y. We consider a *loss function* $\ell(\hat{y}, y)$ that measures the cost of predicting $\hat{y}$ when the actual answer is y, and we choose a family $\mathcal{F}$ of functions $f_w(x)$ parametrized by a weight vector w. We seek the function $f \in \mathcal{F}$ that minimizes the loss $Q(z, w) = \ell(f_w(x), y)$ averaged on the examples. Although we would like to average over the unknown distribution $dP(z)$ that embodies the Laws of Nature, we must often settle for computing the average on a sample $z_1 \ldots z_n$.

$$E(f) = \int \ell(f(x), y)\, dP(z) \qquad E_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) \tag{18.1}$$

The *empirical risk* $E_n(f)$ measures the training set performance. The *expected risk* $E(f)$ measures the generalization performance, that is, the expected performance on future examples. The statistical learning theory [25] justifies minimizing the empirical risk instead of the expected risk when the chosen family $\mathcal{F}$ is sufficiently restrictive.

G. Montavon et al. (Eds.): NN: Tricks of the Trade, 2nd edn., LNCS 7700, pp. 421–436, 2012.

18.2.1 Gradient Descent

It has often been proposed (e.g., [18]) to minimize the empirical risk $E_n(f_w)$ using *gradient descent* (GD). Each iteration updates the weights w on the basis of the gradient of $E_n(f_w)$,

$$w_{t+1} \;=\; w_t - \gamma \frac{1}{n} \sum_{i=1}^{n} \nabla_w \, Q(z_i, w_t) \,, \tag{18.2}$$

where γ is an adequately chosen learning rate. Under sufficient regularity assumptions, when the initial estimate w_0 is close enough to the optimum, and when the learning rate γ is sufficiently small, this algorithm achieves *linear convergence* [6], that is, $-\log \rho \sim t$, where ρ represents the residual error.[1]

Much better optimization algorithms can be designed by replacing the scalar learning rate γ by a positive definite matrix Γ_t that approaches the inverse of the Hessian of the cost at the optimum:

$$w_{t+1} \;=\; w_t - \Gamma_t \frac{1}{n} \sum_{i=1}^{n} \nabla_w \, Q(z_i, w_t) \,. \tag{18.3}$$

This *second order gradient descent* (2GD) is a variant of the well known Newton algorithm. Under sufficiently optimistic regularity assumptions, and provided that w_0 is sufficiently close to the optimum, second order gradient descent achieves *quadratic convergence*. When the cost is quadratic and the scaling matrix Γ is exact, the algorithm reaches the optimum after a single iteration. Otherwise, assuming sufficient smoothness, we have $-\log\log \rho \sim t$.

18.2.2 Stochastic Gradient Descent

The *stochastic gradient descent* (SGD) algorithm is a drastic simplification. Instead of computing the gradient of $E_n(f_w)$ exactly, each iteration estimates this gradient on the basis of a single randomly picked example z_t:

$$w_{t+1} \;=\; w_t - \gamma_t \nabla_w \, Q(z_t, w_t) \,. \tag{18.4}$$

The stochastic process $\{ w_t, \; t=1, \dots \}$ depends on the examples randomly picked at each iteration. It is hoped that (18.4) behaves like its expectation (18.2) despite the noise introduced by this simplified procedure.

Since the stochastic algorithm does not need to remember which examples were visited during the previous iterations, it can process examples on the fly in a deployed system. In such a situation, the stochastic gradient descent directly optimizes the expected risk, since the examples are randomly drawn from the ground truth distribution.

[1] For mostly historical reasons, *linear convergence* means that the residual error asymptotically decreases exponentially, and *quadratic convergence* denotes an even faster asymptotic convergence. Both convergence rates are considerably faster than the SGD convergence rates discussed in section 18.2.3.