Fast algorithms for learning deep neural networks

Miguel A. Carreira-Perpiñán **Weiran Wang**
EECS, University of California, Merced

With the increase in computation power and data availability in recent times, machine learning and statistics have seen an enormous development and widespread application in areas such as computer vision, computational biology and others. A focus of current research are deep neural nets: nested functions consisting of a hierarchy of layers of thousands of weights and nonlinear, hidden units. These hidden units encode hierarchical, distributed features that are useful for automatic discovery of patterns in complex, high-dimensional data such as images or speech, for applications such as classification, regression or dimensionality reduction.

However, as was already observed in the early 1990s, deep nets are very difficult to train because of the ill-conditioned nature of their objective function, in turn caused by its deep level of nesting and the use of squashing nonlinearities. This results in methods such as stochastic gradient descent and nonlinear conjugate gradients taking tiny steps towards a minimum. The optimization proceeds so slowly that in many cases the researcher stops training after a predetermined number of iterations, so that the weights obtained may be far from a minimum. Second-order methods still move slowly and have limited applicability because of the large size of the Hessian. The relative merit of different methods is unclear at the moment. Finally, all these methods require several user parameters that are difficult to set and whose effect on the convergence is crucial: learning rate and its decay, momentum coefficient, minibatch size, etc. [1]. In summary, although there has been some recent progress in optimizing deep nets, this has been limited to (1) the use of parallel computers and GPUs, (2) heuristics to obtain a good initialization such as pretraining [2], and (3) hand-crafted implementations of standard optimization methods. At present, training neural nets remains an art, and for large datasets it requires hours or days even with parallel computers. This makes it hard to select the best architecture for a given task, and the amount of hand-tuning involved means the results are difficult to replicate by others.

We propose a framework for deep net optimization, that we call *method of auxiliary coordinates (MAC)*, that directly address the ill-conditioning problem, based on an idea of introducing auxiliary variables. This replaces the original problem involving a deeply nested function with an equivalent, constrained problem involving a different function in an augmented space, but much better conditioned and without nesting. Thus, it directly addresses the fundamental problem of deep net training, rather than use existing optimization methods with the ill-conditioned objective function. This constrained problem can be solved in a number of ways. We have currently explored the simplest option, based on the quadratic-penalty strategy. Here, we define a sequence of unconstrained subproblems where the constraints act as penalties of increasing weight. Convergence to a minimum of the deep net can be proved under mild assumptions. Optimizing each subproblem jointly over all the weights and variables can be done very efficiently by alternating steps that minimize over the net weights and over the auxiliary variables, respectively. Each such step decouples into a large number of small, independent minimizations, each of which can be easily solved, and without backpropagating any gradients. The result is that a single iteration of this algorithm, while costlier than a simple gradient iteration, achieves a far larger decrease in the objective function, leading to faster convergence overall.

Another fundamental advantage of our approach is that it affords a trivial parallelization with fine granularity and very small overhead, because of the large number of independent subproblems that arise during the optimization. This is in contrast to existing methods, such as stochastic gradient descent or conjugate gradients, where parallelization is harder to implement and has much higher overhead.

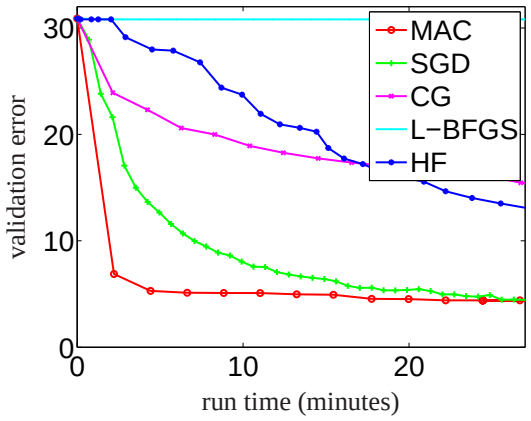

Figure 1: Experiments with the USPS dataset of handwritten digit images using a deep net with 5 layers of hidden units. We plot the validation error as a function of run time, for a random initialization of the weights. Methods: method of auxiliary coordinates (MAC), stochastic gradient descent (SGD), conjugate gradients (CG), L-BFGS, Hessian-free (HF). Markers shown every epoch (SGD), every 100 iterations (CG, L-BFGS) or every 1 iteration (MAC, HF).

As seen in fig. 1, our results using deep autoencoders (of up to 13 layers of sigmoidal units) with handwritten digit images and with no parallel computation already show that our approach needs extremely few iterations to take us very close to a minimum of the nested problem, and although each iteration is relatively costly, we achieve great speedups in total runtime over standard techniques such as stochastic gradient descent [3], conjugate gradients, L-BFGS [4] and Hessian-free methods [5].

References

[1] Genevieve B. Orr and Klaus-Robert Müller, editors. *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.

[2] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 28 2006.

[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, November 1998.

[4] Quoc Le, Jiquan Ngiam, Adam Coates, Abhik Lahiri, Bobby Prochnow, and Andrew Ng. On optimization methods for deep learning. In Lise Getoor and Tobias Scheffer, editors, *Proc. of the 28th Int. Conf. Machine Learning (ICML 2011)*, pages 265–272, Bellevue, WA, June 28 – July 2 2011.

[5] James Martens. Deep learning via Hessian-free optimization. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proc. of the 27th Int. Conf. Machine Learning (ICML 2010)*, pages 735–742, Haifa, Israel, June 21–25 2010.