Abstract: Memory, bias, and variance reduction

Matthew J. Holland and **Kazushi Ikeda**
Graduate School of Information Science
Nara Institute of Science and Technology

Background

Training neural networks with many free parameters $\boldsymbol{w} \in \mathbb{R}^d$ and numerous instances $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathcal{Z}$ is a central task in machine learning applications, including modern speech recognition and machine translation systems. To do this successfully requires algorithms for reliable statistical inference, as well as efficient implementations of these procedures. Our interest is *learning efficiency*: achieving the best generalization with the least computational resources (time, samples, etc.). In this work, we explore the possibility of improving learning efficiency through a strategic use of memory and robust, potentially biased estimators of underlying task parameters.

To ensure performance generalizes off-sample, in training we typically use a loss $l(\boldsymbol{w}; \boldsymbol{z}) \geq 0$, to be minimized in $\boldsymbol{w}$, over the random draw of $\boldsymbol{z} \sim \mu$. Estimating μ is difficult, but the risk $R(\boldsymbol{w}) = \mathbf{E}_{\mu}\, l(\boldsymbol{w}; \boldsymbol{z})$, makes for a more practicable objective function (albeit still unknown). A general strategy is the *approximate* gradient update

$$\boldsymbol{w}_{(t+1)} = \boldsymbol{w}_{(t)} - \alpha_{(t)} \widehat{\boldsymbol{g}}_{(t)}$$

where $\widehat{\boldsymbol{g}}_{(t)}$ is a sample-based estimate of $\nabla R(\boldsymbol{w}_{(t)})$. When n and d are very large, using the entire sample to build each $\widehat{\boldsymbol{g}}_{(t)}$ can incur a prohibitive cost. An extreme cost-saving tactic is to randomly choose $i(t) \in [n]$ and set $\widehat{\boldsymbol{g}}_{(t)}$ to $\nabla l_{i(t)}(\boldsymbol{w}_{(t)}) = \nabla l(\boldsymbol{w}_{(t)}; \boldsymbol{z}_{i(t)})$. Unfortunately, this approximation of ∇R is so poor that many iterations are typically required for convergence. A simple and useful "variance reduction" tactic has been proposed in the finite-sum optimization context (Johnson and Zhang 2013). One simply shifts the single-point estimators using a correction term

$$\widehat{\boldsymbol{g}}_{(t)} = \nabla l_{i(t)}(\boldsymbol{w}_{(t)}) - \Delta_{(t)},$$

where $\Delta_{(t)} = \nabla l_{i(t)}(\widetilde{\boldsymbol{w}}) - \widetilde{\boldsymbol{g}}$. Here $\widetilde{\boldsymbol{w}}$ is a reference vector computed periodically in an outer loop, and $\widetilde{\boldsymbol{g}} = n^{-1} \sum_{i=1}^{n} \nabla l(\widetilde{\boldsymbol{w}}; \boldsymbol{z}_i)$, the full-sample estimate at $\widetilde{\boldsymbol{w}}$.

A natural interpretation is that the learner uses its recent memory to identify "errant" observations, and correct them in a direct manner. Pursuing this memory analogy further, the quality and nature of memories assuredly impacts learning in humans, and the same should be true here. How do corrections based on different stored memories impact learning? Does more reliable observations speed up the process? We consider some simple examples here.

Memory-based corrections

If n is very large, one expects the approximation $\widetilde{\boldsymbol{g}} \approx R'$ to be accurate. Running with this, at each iteration we can compute

$$\boldsymbol{\beta}_{(t)} = \nabla l_{i(t)}(\widetilde{\boldsymbol{w}}) / \widetilde{\boldsymbol{g}},$$

with division carried out element-wise. Assuming the point $i(t)$ has similar idiosyncracies in terms of deviation from the mean at $\widetilde{\boldsymbol{w}}$ as at $\widehat{\boldsymbol{w}}_{(t)}$, then one can adjust as

$$\widehat{\boldsymbol{g}}_{(t)} = \nabla l_{i(t)}(\boldsymbol{w}_{(t)}) / \boldsymbol{\beta}_{(t)}.$$

This represents a qualitatively distinct memory, namely the relative size, rather than explicit differences. Countless similar examples of other types can naturally be explored.

Closely related: what if the learner's observations themselves improve? Efficient learners make use of task-relevant features; using $\nabla l_{i(t)}(\boldsymbol{w}_{(t)})$ as an estimate of the risk gradient is too naive to be plausible. While the utility of using mini-batch estimates has been well-studied (Lin and Rosasco 2016; Jain et al. 2016), more reliable estimates can in principle be created. One example is

$$\widehat{\boldsymbol{\theta}}_{(t)}(\boldsymbol{w}) \leftarrow \arg\min_{\boldsymbol{\theta}} \frac{1}{|D_{(t)}|} \sum_{i \in D_{(t)}} \rho\left(\nabla l_i(\boldsymbol{w}) - \boldsymbol{\theta}\right)$$

with the optimization carried out element-wise. Here ρ is a slow-growing, convex, even function, say $\rho(u) = \log(\cosh(u))$. While this estimate may be biased, it is easily computed using raw observations, and truncates errant observations. While this alone is appealing (Holland and Ikeda 2017), of chief interest here is how this first layer of robustness interacts with memory-based corrections. The most direct way to examine this is to plug in these new observations:

$$\widehat{\boldsymbol{g}}_{(t)} = \widehat{\boldsymbol{\theta}}_{(t)}(\widehat{\boldsymbol{w}}_{(t)}) / \boldsymbol{\beta}_{(t)}$$

for the relative size tactic, and

$$\Delta_{(t)} = \widehat{\boldsymbol{\theta}}_{(t)}(\widetilde{\boldsymbol{w}}) - \widetilde{\boldsymbol{g}}$$
$$\widehat{\boldsymbol{g}}_{(t)} = \widehat{\boldsymbol{\theta}}_{(t)}(\widehat{\boldsymbol{w}}_{(t)}) - \Delta_{(t)}$$

for the difference-based tactic. In addition to some theoretical groundwork, we provide empirical analysis of how strategic memory use and robust observations impact learning efficiency under diverse task conditions.