Pierre-Yves Massé
"Online Optimisation of Time-Varying Systems"

Abstract:
Dynamical systems are a wide ranging framework which may model time varying settings, spanning from engineering (e.g., cars) to machine learning (e.g., recurrent neural networks), for instance. The correct behaviour of these systems is often dependent on the choice of a parameter (e.g., the gear ratio or the wheel in the case of cars, or the weights in the case of neural networks) which the user has to choose. Finding the best possible parameter is called optimising, or training, the system. Many real life issues require this training to occur online, with immediate processing of the inputs received by the system (e.g. the returns about the surroundings of the sensors of a car, or the successive frames of a video fed to a neural network).
We present a proof of convergence for classical online optimisation algorithms used to train these systems, such as the “Real Time Recurrent Learning” (RTRL) or “Truncated Backpropagation Through Time” (TBTT) algorithms. These algorithms avoid time consuming computations by storing information about the past, in the form of a time dependent tensor. However, the memory required to do so may be huge, preventing their use on even moderately large systems. The “No Back Track” (NBT) algorithm, and its implementation friendly “Unbiased Online Recurrent Optimisation” (UORO) variant are general principle algorithms which approximate the aforementioned tensor by a random, rank-one, unbiased tensor, thus decisively reducing the storage costs but preserving the crucial unbiasedness property allowing convergence. We prove that, with arbitrarily large propability, the NBT algorithm converges to the same local optimum as the RTRL or TBTT algorithms.
We might conclude by quickly presenting the “Learning the Learning Rate” (LLR) algorithm, which adapts online the step size of a gradient descent, by conducting a gradient descent on this very step. It thus reduces the sensitivity of the descent to the numerical choice of the step size, which is a well documented practical implementation issue.