White Paper: Design for Adaptation

Prof. G. Pottie
UCLA ECE Department
March, 2021

Nearly 400 years after the birth of Newton, there are no algorithms with guaranteed rapid convergence not using the intuition of his celebrated Method for finding minima: make a local linear approximation and adapt parameters to move down the slope. We have advanced to produce adaptive versions in the form of the LMS and RLS algorithms and found forms that are easily implemented in finite precision arithmetic, and have even adapted gradient descent to work with neural networks in the form of back-propagation. Many variants have been proposed, with continuous and block adaptations. We have explicitly used Newton's method to approximate non-convex problems as convex and so adapt in a set of (relatively computationally expensive) steps, albeit without guarantee of convergence, as for neural nets. We optimize in the dual domain, hoping it is less non-linear, or apply other transformations and adapt using gradient methods in those domains. We have used feedback in controls to linearize problems in the region of the solution and thus allow gradient methods to be used in highly non-linear problems. But at the heart is always the basic fact of Taylor's Theorem (the linear approximation is always locally good) combined with gradient descent. Non-linear adaptive methods exist (e.g., genetic algorithms) but they lack fast convergence properties; a broad set of heuristics for particular models lack both generality and guarantees, although they often work well enough. The obvious question for researchers in this space is then, given we have made such little progress since Newton in finding general methods with guaranteed convergence properties in non-convex problems, in spite of enormous incentives to find them, is there no realistic hope of a breakthrough?

Leaving aside the possibility that someone might actually come up with The Beyond Newton Algorithm^{TM}, we offer a suggestion on the path forward without such a wonder. The main principle of design is that if some aspect of a problem is obviously very hard, change the problem by working on other parts of the design space. Just such an approach has been wildly successful in communications systems design. Amplifiers are necessary for long-range communication, and they are all based on non-linear devices. If such raw devices were used in the transmission path, it would be impossible to adapt the receiver at high speed and in real time to reliably decode the bits. But yet we do so with relative ease, using utterly standard least squares adaptation that handles unstable oscillators, rapid channel fading, inter-symbol interference, inter-channel interference, and echoes, and have done so since the heyday of voice-band modems. How? The answer is that we engineered the overall channel at every level so that least squares methods would work. Amplifiers include internal feedback so that they are linearized, and amplifier chains are used to improve noise/amplification/linearity tradeoffs. The functions of frequency lock, phase lock, symbol lock, automatic gain control, and equalization are all separated, usually assisted with training sequences/signals, so that second order loops can be used (i.e., such that least squares will work). Even signals are designed to make them easily distinguishable with feasible receivers. This was partly accidental—radio systems were for many years implemented in purely analog form, and loop stability could not be easily engineered beyond second order systems. But it became embedded in training communication engineers. Then when channel coding was added, designers could be confident that the residual channel was quasi-stationary and Gaussian. This enables codes that approach Shannon capacity to be used, irrespective of whatever else is going on. Communication systems designers so love the Gaussian channel that even multiple access channels are largely transformed to be like them, via directed antennas and protocols that separate users into channels where the interference is well below the desired signal level. To be sure, many multi-user detection schemes have been proposed, but channel separation dominates in practice, even though this requires optimization of channel allocations, flow control, etc. It is with joy that designers greet the fact that the microwave band is very amenable to use of tight antenna beams, thus avoiding most of the non-linear optimizations associated with getting users to coexist on the same channel.

I have cited physical layer communications systems design to illustrate how far we can go by physically transforming a non-convex problem into a set of easily solved (approximately) convex problems, mainly because I have spent so many years in that research space and teaching a graduate course on the topic. But the situation is quite similar in control systems, where linear methods were pushed very hard for many years for very similar reasons. In both cases, extensive use is made of physical understanding of the systems (i.e., a causal model) in order to develop useful approximations. The question is how far we can extend this design mind-set to other domains, where the models may be less known and more complicated.

Let us consider data-driven adaptation. Right now, the dominant paradigm is to collect all the data and then adapt the neural net, i.e., fit the function. But we know that not all data is equal, nor are all classes equally hard to distinguish. For example, many phenomena follow a power law, where the probability distribution is consumed by a relatively small number of features or classes, but with heavy tails so that the number of features or classes can be very large. Could we not structure the adaptation of the classifier to take place in a sequence of steps? Small models (dealing with the small feature spaces) are easily trained with back-propagation. Transfer learning might then be used to train a slightly larger model. In effect we build out from sub-spaces to larger spaces in a sequential fashion. Recent work shows that using statistical invariance tests, one can rigorously test whether we have identified true causal models, rather than spurious causes. Effectively then we are easing the path for gradient methods to work by curating a sequence of data sets by exploitation of some structure in the mapping of features to classes. This is not without famous precedent. The scientific method effectively proceeds in this fashion, building out from simple models in a cycle of hypothesis, experimentation to collect data, and model refinement (often using least squares in some transform domain). Newton again! Why should not machine learning follow a similar multi-step cycle? Some additional machinery is needed: formulation of hypotheses requires counterfactual reasoning, and collecting data to advance some goal is a control problem. But there is great deal of research in each of these domains that can be brought to bear when we frame the learning problem not as one single event, but as a sequence of steps.

The other classical way the engineering discipline proceeds to cope with massive problems is to use hierarchy. The right set of abstractions has been hugely successful in managing both complexity and lack of tools that can handle the original problem. We are almost never by such means solving the exact original problem: approximations arise at many points, including in performance goals. But this may bring additional benefits, such as modularity that enables less effort in extending the model to new situations. For example, a hierarchy that corresponds to Pearl's three rungs on the ladder of causation might be a path towards a more general AI, so long as we are explicit in designing the system with its growth over some sequence of experimental cycles in mind. In another cut at the problem, we may have a set of modules at the lowest rung that correspond to different baskets of items to classify, with higher level reasoning to determine which produces the most likely outcomes to justify further processing. Or we may go back and forth between global views (converting features into some type of "image") and local views, perhaps with the aid of yet undiscovered compressive transforms designed to produce continuous spaces amenable to gradient search.

In summary, there appear to be many plausible paths to transforming problems into approximate forms amenable to some *iterative* combination of gradient methods, causal reasoning, device/system development, and experimental data collection. Such an approach is well within the mainstream of engineering design practice; perhaps the time has come to systematize it.