Artificial Intelligence

Harrison B. Prosper

Department of Physics, Florida State University, Tallahassee, Florida, USA

Abstract. I summarize the ideas that emerged from the parallel sessions on Artificial Intelligence and I offer a critical assessment of what we learnt.

INTRODUCTION

In 1956, at a conference in Vermont organized by John McCarthy, the term "Artificial Intelligence," (AI to the cognoscenti) was coined. The grand vision of the pioneers was to create, in due course, machines that combined phenomenal computing power with human-like characteristics. In his classic science fiction novel "2001, A Space Odyssey" Arthur C. Clarke envisioned such an awesome machine, the HAL 9000. But, unfortunately for the crew of the Jupiter Mission, that machine, in spite of its enormous intellectual power, displayed the severest of human failings: it became murderously psychotic! Today, forty-five years after the brainstorming in Vermont, machines have indeed grown enormously more powerful. But the real success of AI, to date, has been not so much in making HAL-like machines but rather in solving difficult real-world problems.

Some of these problems are indeed extremely hard. An interesting example is the optimal scheduling of computer jobs in a distributed computing system, as described in the contribution by Newman and Legrand (1). Here, the term artificial intelligence has some resonance. The problem addressed by the authors is daunting: to build an autonomous system that schedules computer jobs in a a dynamic distributed computing environment that could contain millions of interacting components subject to thousands of time-dependent constraints. On the face of it, to have any chance of a near-optimal solution, this problem appears to demand a very high level of rapid and intelligent decision-making. The complexity is surely on a scale that is likely to overwhelm intelligence of the natural sort.

For the most part, the contributors to the parallel sessions on Artificial Intelligence had a more prosaic goal: the optimal analysis of multi-dimensional data in a more-or-less automatic manner. I shall focus on these contributions and refer the reader to the paper on job scheduling (1) and to the contribution by Berg (2) dealing with a generalization of the Clopper-Pearson method for constructing binomial confidence limits.

The remaining contributions can be usefully grouped into the following categories.

- **Classification**
  - Selecting W-pair events (DELPHI) (3)
  - Searching for single top events (DØ) (4)
  - Searching for a low-mass Higgs boson events (5)
  - Searching for instanton-induced processes (HERA) (6)
  - Electron/jet discrimination (ATLAS) (7)
  - Electron and τ-lepton identification (DØ) (8)
  - Rule induction (9)
- **Parameter Estimation**
  - Measuring the top quark mass (10)
  - Calorimeter energy estimation (ATLAS) (11)
- **Feature Recognition**
  - Vertex finding (ZEUS) (12)

Below I briefly consider each of these problem categories separately and end with a few conclusions.

CLASSIFICATION

The problem addressed by the contributors in this category is the optimal classification of objects, specifically, high energy events or particles. Each object is described by a feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ of N measured quantities x_i. The goal is to classify the object based on the value of its feature vector. The solution to this problem is easy to state: use the *Bayes decision rule*, namely, given $\mathbf{x}$ assign the object to the class k with the highest posterior probability

$$P(k|\mathbf{x}) = \frac{f(\mathbf{x}|k)p(k)}{\sum_k f(\mathbf{x}|k)p(k)}, \tag{1}$$

where $f(\mathbf{x}|k)$ and $p(k)$ are the density function and the prior probability of class k, respectively. This rule minimizes the probability of making incorrect classifications.

From the viewpoint of classification, the principal difference between the parallel session contributions is the algorithm used by the authors to approximate the posterior probability $P(k|\mathbf{x})$. Most authors (for well-founded reasons (13, 14, 15, 16)) approximated this probability using feed-forward neural networks (17). Koblitz (6), however, found it computationally more efficient to approximate this probability with a kernel density method that, in the neigborhood of a given point, counts the number of feature vectors of each class. The key feature of this method is an efficient (range searching) algorithm to determine which points lie within a given neighborhood.

Stepanov (9) noted the persistent perception, by some, of neural networks as "black boxes." In view of the considerable body of mathematical work that underpins neural networks and the great conceptual clarity that has ensued, I am inclined to conclude that this perception is a problem of sociology. Stepanov proposes to circumvent this (sociological) problem by using an interesting semi-automatic rule induction system that, in effect, approximates the posterior probability with a decision tree. The claimed advantage of this system is that a decision tree, constructed directly from the components of the feature vectors, provides an *explanation* for every classification it makes. This is true and could be important. For instance, a medical practitioner presumably has an interest in understanding how an expert system has arrived at a medical diagnosis. However, for the sorts of applications for which the rule induction system is targeted it is not clear what is gained by having a decision tree, rather than a single cut on a single discriminant.

Of course, it is often instructive to understand why an object was classified the way it was; why an electron and not a jet. But my claim is that one can gain an understanding of the "reasons" for a decision by examining how the single cut has distorted the distribution of each component or, for more insight, the distributions of different sub-sets of the components taken two or even three at a time. Where a rule induction system could be extremely valuable is, for example, in the alarm reporting system of a large and complex experiment such as DØ, where there is great interest in understanding why a particular class of fault was reported in response to a feature vector representing the active state of the experiment. A rule induction system is precisely what is needed in such an expert system.

It comes as no great surprise that every author reported, in some cases significant, improvement in classification performance relative to more traditional methods. For example, Chakraborty (8), who talked about electron and τ-lepton identification, reported a factor of 10 improvement, with respect to previous work by the DØ collaboration, in the correct classification of objects as background rather than as electrons, while achieving a classification probability of 90% for electrons.

One issue that was addressed by Seixas (7), in the context of electron/jet discrimination, is that of dimensionality reduction. All things being equal, it is advantageous to reduce the dimensionality of the feature vectors, if for no other reason than to decrease the computational burden. (For a counter example, see Dudko (4) who showed that even with moderately large dimensions it is still possible to make progress provided one proceeds with due caution.) The authors faced a severe computational problem: a feature vector of dimension 496, one dimension per calorimeter cell! However, by a judicious use of the method of Principal Component Analysis (PCA) (17) the 496 dimensions of the original feature vectors were reduced to as few as 9, while maintaining efficient electron/jet discrimination.

PARAMETER ESTIMATION

A remarkable property of feed-forward neural networks, which in the final analysis are nothing more than highly non-linear mathematical functions, is that they are universal approximators (16). In principle, they can be used to learn any mapping of the form $f : \Re^n \to \Re^m$. Another useful property, not specifically tied to neural networks, is their Bayesian interpretation (13, 14, 15). In the paper presented by Beri (10), explicit use was made of the fact that the outputs of feed-forward neural networks, when appropriately trained, approximate class posterior probabilities. The authors exploited the Bayesian interpretation to construct an algorithm to estimate the top quark mass using events in which a full kinematic reconstruction of the top quark 4-vectors is not possible.

In many situations, the chief concerns during the training phase of a neural network are simply to guard against over-training, that is, "over-fitting" and to ensure convergence. But, as Seixas (11) showed, the training strategy itself can be put to good use. The goal was to improve the estimation of the energy deposited in the ATLAS calorimeter by reducing the non-linearity in its response, while maintaining acceptable energy resolution. This was achieved by dividing the training into two phases. In the first, the network target output is set to the beam energy while during the second, the target output is set to a linear sum of the calorimeter cell energies. The use of a neural network, trained in this novel way, reduced the non-linearity by almost a factor of 4, and caused no significant degradation in the energy resolution.

FEATURE RECOGNITION

The application described by Etzion (12) provided a beautiful example of feature recognition that incorporates parameter estimation. Etzion described a hierarchy of neural networks that recognizes tracks in the ZEUS vertex detector. The inspiration for this architecture is the hierarchically processing of information in the primary visual system. The system studied consists of a "retina" of inputs followed by three layers of neural networks. The outputs of the neural networks in the first and second layers of the hierarchy provide the inputs to the subsequent network. The first layer finds line segments, the second combines segments into arcs and the last combines arcs into tracks. This application, in which data are compressed into more and more useful forms, is an excellent example of the possible advantage of trying to solve a problem by mapping it to simpler and simpler sub-problems, each solved by its own "expert."

CONCLUSIONS

All classification methods are approximations to the same problem: the calculation of class posterior probabilities, $P(k|\mathbf{x})$. Given a fixed and finite set of resources, time, computational, intellectual, it remains a matter of experimentation to decide which method of approximation is the best for a given application since experience suggests that no single method is uniformly the best over the set of all possible problems. Therefore, provided that they are mathematically well-founded, one should embrace them all.

That being said, however, it is nonetheless useful to have a *default* method that is known to work well in circumstances of general interest. It seems that neural networks are emerging as the default method for classification, parameter estimation, functional approximation and feature recognition. And for truly difficult problems, such as the problem of optimal job scheduling in a large and complex distributed system, the use of neural networks of one sort or another may prove to be the only feasible approach that stands any chance of succeeding.

It has been amply demonstrated at this workshop, in the many excellent contributions at the parallel sessions, that neural networks work well in practice. But, as is true of anything that one wishes to do well, it is necessary to exercise a fair measure of non-artificial intelligence to render that statement true!

REFERENCES

1. Newman, H.B., Legrand, I.C., these proceedings.
2. Berg, B.A., these proceedings.
3. Becks, K.-H., Buschmann, P., Drees, J., Müller, U., Wahlen, H., these proceedings.
4. Dudko, L., these proceedings.
5. Tentindo-Repond, S., Bhat, P.C, Prosper, H.B., these proceedings.
6. Carli, T., Koblitz, B., these proceedings.
7. Vassali, M.R., Seixas, J.M., these proceedings.
8. Chakraborty, D., these proceedings.
9. Stepanov, N., these proceedings.
10. Beri, S., Bhat, P.C., Kaur, R., Prosper, H.B., these proceedings.
11. da Silva, P.V.M, Seixas, J.M., Seixas, J., these proceedings.
12. Dror, G., Etzion, E., these proceedings.
13. Ruck, D.W. *et al.*, *IEEE Trans. Neural Networks* **1 (4)**, 296 (1990).
14. Wan, E.A., *IEEE Trans. Neural Networks* **1 (4)**, 303 (1990).
15. Blum, E.K., and Li, L.K., *Neural Networks* **4**, 511 (1991).
16. Hornik, K., Stinchcombe, M., and White, H., *Neural Networks* **2**, 359 (1989); Hornik, K., *Neural Networks* **6**, 1069 (1993);
17. Bishop, C.M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1998; Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Inc, Upper Saddle River, New Jersey, 1999.