Optimization by Neural Networks

By G. Saravana Kumar and P K Kalra

Optimization refers to the art and science of allocating scarce resources to the best possible effect. Optimization techniques are called into play every day in question of industrial planning, allocation, scheduling, decision-making, etc. For example, how does a global petroleum refiner decide where to buy crude oil, where to ship it for processing, what products to convert it to, where to sell those products, and at what prices? A maximum-profit optimization model is used to solve this problem. How does an airline know how to route its planes and schedule its crews at minimum cost while meeting constraints on airplane flight hours between maintenance and maximum flight time for crews? New optimization techniques are arriving daily, often stimulated by fascinating insights from other fields. Genetic algorithms, for example, use an analogy to chromosome encoding and natural selection to evolve a good optimized solution. Artificial neural networks, on the other hand process information the way biological nervous systems, such as the brain, for optimal decision making. Certain characteristics of their architecture and the way they process information makes them superior to conventional techniques on certain class of optimization problems.

Pattern classification is one such domain in which neural networks have shown to perform better than conventional linear programming methods. The present article outlines the basic optimization formulation of a classification problem using a historical problem and later on shows how neural networks outperform conventional techniques for this class of optimization problem. A case study is further illustrated.

The diet problem is one of the first optimization problems to be studied back in the 1930’s and 40’s. It was first motivated by the Army’s desire to meet the nutritional requirements of the field while minimizing the cost. The goal of the diet problem is to find the cheapest combination of foods that will satisfy all the daily nutritional requirements of a person. We include constraints that regulate the number of calories and amounts of vitamins, minerals, fats, sodium and cholesterol in the diet. A verbal formulation of the problem is as following.

Minimize the "cost of the menu"
subject to the nutrition requirements:

- eat enough but not too much of Vitamin A
- eat enough but not too much Vitamin C

The optimization problem thus formulated can be classified as linear or nonlinear based on the type of objective and constraints. A very special case of great importance is where the objective function and the constraints are entirely linear; this is called Linear Programming (LP).

A typical optimization problem is a pattern classification problem in which the objective to classify or recognize a pattern. The examples are recognizing characters/signatures (Fig. 1a), classifying microstructure for recognizing composition of alloys (Fig. 1b), facial feature recognition to identify humans for secure access (Fig. 1c) and identifying cancerous cells for therapy (Fig. 1d). The applications are very many and the common task here is to distinguish between elements of two disjoint pattern sets. The solution to these classification problems were originally tried by conventional optimization techniques like linear programming. Consider a sub-class of the original diet problem, in which the interest is to classify any diet (combination of meat, carrots, etc.) as healthy and unhealthy. In a generic formulation consider a diet to
In most pattern classification applications we cannot assume that the samples are linearly separable. When the patterns are not separable by a hyperplane, we would still like to obtain a weight vector that classifies as many samples correctly as possible. It turns out that the number of errors (samples incorrectly classified) is not a linear function of the components of the weight vector. Thus trying to minimize the number of errors is not a LP problem and thus in a general pattern classification problem LPs do not give best classifying boundaries. Perceptrons were the first known neuron models and they yield a separating vector which is "reasonably good" solution in the inseparable case.

The perceptron is a single layer neural network whose weights and biases could be trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called the perceptron learning rule. The perceptron generated great interest due to its ability to generalize from its training vectors and work with randomly distributed connections. Perceptrons are especially suited for simple problems in pattern classification. Consider a perceptron network consisting of a single neuron connected to two inputs through a set of 2 weights, with an additional bias input. Fig. 2 shows a schematic. The perceptron is trained to respond to each input vector with a corresponding target output of either 0 or 1. The learning rule has been proven to converge on a
Considering the diet problem, the input vectors to the perceptron will be the quantity of meat and quantity of carrots and the output to be trained is 1 (for healthy diet) and 0 (for unhealthy diet). A set of such known vectors (training set) are presented to the network one after another. If the network’s output is correct, no change is made. Otherwise, the weights and biases are updated using the perceptron learning rule. An entire pass through all of the input training vectors is called an epoch. When such an entire pass of the training set has occurred without error, training is complete. At this time any input training vector may be presented to the network and it will respond with the correct output vector. If a vector P not in the training set is presented to the network, the network will tend to exhibit generalization by responding with an output similar to target vectors for input vectors close to the previously unseen input vector P. The diet classification problem (with overlapping healthy and unhealthy diet) as solved by a single perceptron is illustrated in Fig. 3. Consider now, the situation shown in Fig. 4. in pattern space. Two boundaries are needed to classify (and thus two neurons), thus making this problem a non-linearly separable classification problem.

In general we require an arbitrarily shaped decision surface. Multilayer perceptions (MLPs) are used to solve such kind of highly nonlinear classification problems. MLPs are feedforward neural networks trained with the standard backpropagation algorithm. They are supervised networks so they require a desired response to be trained. They learn how to transform input data into a desired response, so they are widely used for pattern classification. With one or two hidden layers, they can approximate virtually any input-output map. They have been shown to approximate the performance of optimal statistical classifiers in difficult problems. Also in many problems the explicit objective in terms of the variables cannot be written and thus only learning by example strategy can be used to solve such classification problems. Examples include detecting cancer cells, characters etc. Here images representing the classes are shown to the network and the network gets trained to recognize the classes. Neural networks have the following key behaviors

- interacting with noisy data
- massive parallelism
- fault tolerance
- adapting to circumstances that make them perform better than conventional computations in these class of problems
- where one can’t formulate an algorithmic solution.
- where one can get lots of examples of the behaviour we require.
where one need to pick out the structure from existing data.

An application study on Lymphoma diagnosis is illustrated. Lymphoma is a type of cancer that can occur when an error occurs in the way a lymphocyte is produced, resulting in an abnormal cell. The most common presentation of lymphoma is usually a painless swelling of lymph nodes that often occurs in the neck or under the arms. A common procedure for diagnosis is biopsy, in which a piece of tissue from an area of suspected cancer is removed from the body for examination under a microscope. The information provided by this tissue sample is crucial to diagnosing and treating lymphoma. The information provided by cell diagnosis can be numerous (running into several thousand parameters). Thus classifying a cell as cancerous/non-cancerous by computation is a challenging task. A total of 94 patterns of tissue data (each containing 4026 parameters) were obtained. A single data pertaining to Lymphoma tissue and that of a normal tissue is shown graphically in Fig. 5.

A MLP neural network trained by backpropagation with 4026 input neurons and two hidden layers of 20 and 10 neurons each and with a single output neuron to say 'Yes' or 'no' was designed to classify the tissue data as cancerous or normal. A part of the available data i.e. 72 sets were used for training and 22 were kept for testing the network for generality. After training for 1000 iterations the network predicted correctly on 91% of the test patterns (see Fig. 6).

Our current research is being focused on the adaptation of neurocomputing, fuzzy-logic, and evolutionary search, to overcome limitations encountered in the use of traditional optimization methods in problems of multidisciplinary analysis and design. The underlying challenges include reducing the high computational cost associated with repetitive analysis, adapting emergent optimization methods for large scale design problems involving a mix of continuous, discrete, and integer variables, and incorporating uncertainties and imprecise information into the design problem formulation.

Fig. 5: a) Data of a Lymphoma tissue. b) Data of a normal tissue.

Fig. 6: Classification performance by Neural networks.
Further Reading:

Acknowledgements: Mr. Vrijendra Singh was instrumental in simulating neural network algorithm for the case study and provided exciting results.

About the author: Dr. G. Saravana Kumar is a Research Associate in the Department of Mechanical Engineering. He obtained his PhD in Mechanical Engineering from IIT Kanpur, India. His research interests are Layered manufacturing, Shape optimization, Shape synthesis using point clouds, Bio-Modeling, Computational Geometry, and Soft Computing.

About the author: Professor Prem Kumar Kalra obtained his PhD in Electrical Engineering from University of Manitoba Canada. He is currently Professor of Electrical Engineering and the Dean of Resource Planning at IIT Kanpur, India. His research interests are Power systems, Expert Systems applications, HVDC transmission, Fuzzy logic and Neural Networks application and KARMAA (Knowledge Acquisition, Retention, Management, Assimilation & Application).
Now that you are at the very last article of this magazine, one thing must be clear: there is a wide gap between the theory and practice in the area of optimization. The theoretical articles seem to assume nice mathematical properties of functions, such as derivatives, smoothness, continuity, convexity etc., for predicting and guaranteeing the behavior of an algorithm. Since such properties do not often exist in real-world problems, these studies cannot take any serious real-world problems as a case study. On the other hand, the articles trying to solve practical optimization problems do not seem to care much about using any real theory. It is a fact that the gap exists. It exists not only in this magazine, but also in the optimization literature alike.

However, there are real reasons for the gap to exist. It is not a mere apathy between the two groups which has caused the gap, rather it is the fact that the mathematical rigor which is needed to handle commonplace matters in practical optimization, such as discreteness, non-convexity, non-smoothness, nonlinearity etc., is not suitable for practice. In another vein, a practical optimization study may not care much about finding the real optimum. A solution close to the optimum may be adequate for the purpose. Thus, all the theory which exist for arriving at the true optimum solution may not be true for finding a near-optimum solution. If one is interested in finding a near-optimum solution, there may exist a completely different theory and algorithm.

Another apparent reason is that there has not been a real effort to bridge the gap between the two extremes in any place at any point of time. Although an applicationist may not always be looking for an algorithm which has a convergence proof, it is our understanding that there exist many theories and properties of optimum or near-optimum solutions which can be put into practice with some effort. For example, the epsilon-approximation theory may be used to establish an upper bound on the proximity of the obtained solution from the true optimum. In another attempt, a theoretically-sound point-based search strategy can be used to replace the usual mutation operator in a GA.

Such efforts will not only make the application more reliable, but will also motivate more such synergies, thereby reducing the gap. At IIT Kanpur, we have started thinking in this direction and floated a course in the 2003-04 IInd semester on multiobjective optimization which was split 50-50 to theory and practice. Although students recognized the existence of a vivid gap between the two topics, we and students had experienced a number of possible research avenues, such as twigging an algorithm with a known theory and developing a theory for practice. Based on this experience, we are now thinking of research collaborations which should make our theoretical studies more meaningful and our application studies more foundational.

While we go for such ventures, we shall be looking for industry collaborations for solving real-world and challenging optimization problems. Such a venture will also require participation of many theorists and applicationists, possibly working together under a common roof in the form of a future ‘Centre of Applied Optimization’. Some such centres have recently come up in various parts of the world.

2. Institute for Optimization, Martin-Luther University, Halle, Germany, http://www.mathematik.uni-halle.de/institute/optimierung/
3. Center for Mathematical Modelling, University of Chile, Santiago, Chile, http://www.conicyt.cl/fondap/version-ingles/matematicas-ingles.html

With the vast resource and experience of faculty members at IIT Kanpur in the area of optimization demonstrated by the wide variety of articles in this magazine both in theory and in practice, there is no reason why we cannot establish a similar collaborative centre with the help of the Government, industries, and researchers.