On Generalisation of Machine Learning with Neural-Evolutionary Computations

Rajeev Kumar
Department of Computer Science, and
Centre for Robotics & Intelligent Systems
Birla Institute of Technology & Science
Pilani - 333 031, India
rajeevk@bits-pilani.ac.in

Abstract

Generalisation is a non-trivial problem in machine learning and more so with neural networks which have the capabilities of inducing varying degrees of freedom. It is influenced by many factors in network design, such as network size, initial conditions, learning rate, weight decay factor, pruning algorithms, and many more. In spite of continuous research efforts we could not arrive at a practical solution which can offer a superior generalisation. In this paper, we present a novel approach for handling complex problems of machine learning. A multiobjective genetic algorithm is used for identifying (near-) optimal subspaces for hierarchical learning. This strategy of explicitly partitioning the data for subsequent mapping onto a hierarchical classifier is found both to reduce the learning complexity and the classification time. The classification performance of various algorithms is compared and it is argued that the neural modules are superior for learning the localised decision surfaces of such partitions and offer better generalisation.

1. Introduction

An essential attribute of an intelligent machine is its ability to *learn from examples* and make effective decisions when presented with *unseen* data. Among the many domains of machine learning the connectionist intelligence is the most commonly used paradigm of learning. During the process of learning from examples, the network approximates the functional relationship of the restricted domain covered by the training set but it is also expected to understand the wider sampling it has not seen of the parent function under both situations of interpolation and extrapolation. Optimising such a neural network architecture for supervised learning tasks is a crucial issue for improving generalisation capability.

Viewed in terms of bias and variance [1], we can say that for a good generalisation we need to control the effective complexity of the network for an optimum mix of both bias and variance. Analogously several methods have been proposed for controlling the network complexity: there are approaches where one starts with a relatively large network and prune out the least significant connections or derives them to insignificance. Similarly one can start with a small network and add units during the learning process with the goal of arriving at an optimal network. There are other dependencies as well *e.g.*, initial network conditions, learning rate, cross-validation, stopping criterion and curse of dimensionality [2].

There exists another dimension to the problem of generalisation, and that relates to scaling of connectionist models for solving arbitrarily complex problems. Scaling connectionist models to larger systems is a difficult problem because larger networks require increasing amounts of training time and data, and eventually the complexity of the optimisation task reaches computationally unmanageable proportions. One possible reason for this is that a global error-surface may have extensive constant regions and significant variations in local minima. Such situations are common in most real-function approximations - more so with high-dimensional pattern spaces - which often require very long learning times or result in unsuccessful training. Simply increasing the number of hidden units is a popular solution but may unjustifiably increase the number of free-parameters of the net architecture, which can lead to poor generalisation.

In the context of addressing complex learning domains, two basic approaches are emerging as possible solutions to the poor scalability of neural networks: ensemble based and modular systems. The family of ensemble-based approaches relies on combing predictions of multiple models, each of which is trained on the same database; in general, the emphasis is on improving the accuracy for a better generalisation and not on simplifying the function approximators [3].

On the other hand, the main characteristic of modularity is that the system can take advantage of function decomposition. The direct advantages of partitioning are that the scale of the computation at each stage is much less than a single unpartitioned computation, the problem is better constrained and solvable, and can be computed in parallel. Nonetheless decomposition has its own difficulties - partitions in the absence of *a priori* knowledge of the pattern space are not unique. This implies that the modularity can only be meaningful if an *implicit* or *explicit* way to sensibly decompose a complex function exists, and to allow the desired (sub-)function mapping to emerge from the learning of subtasks. For those problems where one has some prior knowledge of the pattern space and the decomposition into subtasks is explicit; this is a trivial task. In the absence of any prior knowledge of the pattern space, decomposition-through-competition has been demonstrated where the decomposition and the learning phases are combined [3].

In this work, we adopt a different line of research and partition the task in a *generic* manner using genetic algorithms as a pre-processor to neural domain. We argue that separating the task of decomposition from the regime of modular learning simplifies the overall architecture and this strategy of data-processing before its submission to a classifier considerably reduces the learning complexity. Additionally only those patterns which lie close to the decision boundaries possibly warrant multiple learning effort in order to improve the prediction accuracy, and the clusters which contain only one data class are implicitly labelled without ambiguity.

2. Generalisation: An Overview

A rule of thumb for obtaining good generalisation is to use the smallest network that fits the data. Unfortunately, it is not obvious what size is the best: a network that is not sufficiently complex is very sensitive to initial conditions and learning parameters, such a small network learns extremely fast but has a high probability in getting trapped in local minima and thus may fail to train, leading to underfitting. On the other hand, larger networks have more functional flexibility than small networks so are better able to fit the data. A network that is too large may fit the noise not just the signal and this leads to overfitting [1, 4]. Overfitting produces excessive variance whereas underfitting produces excessive bias in the output [1].

One major contributor to network complexity is the network-size and it is always desired to minimise the number of free parameters. Many studies have been carried out on selecting a proper size, nonetheless it remains an unresolved problem. Some theoretical studies have

established the upper-bounds on the number of hidden nodes; but *a priori* knowledge of the upper-bounds can neither provide a *practical* guess on the number of hidden-nodes required for mapping a training set involving a large number of samples nor minimise the free parameters. Some researchers also defined the theoretical lower-bounds based on the Vapnik-Chervonenkis (VC) Dimension assuming that the future test samples are drawn from the distribution of training-samples. Weigend [4] avoided overfitting if the net-size was guided by the eigen-value spectra. But there remains the heuristic how to decide the effective dimensionality or the number of parameters.

Another promising approach to avoiding under-/over-fitting and increasing flexibility of network learning is to start with a large, fully-connected network and through regularisation or pruning improve generalisation [2]. Other type of approaches are based on pruning out the least significant connections either by removing individual weights or by removing complete units, *e.g*. optimal brain damage/surgeon. Many researchers have also proposed correlation or some heuristic(s) based pruning/merging methods for model simplification. These approaches are found to be effective on some problem set or the other.

Early stopping monitors the errors on a validation set and halts learning when the error on validation set starts increasing. The objective of this approach is to stop training before the network starts fitting noise. The results of many researchers have provided strong evidence for the efficiency of stopped training. At the same time, it has shown that for finite validation set there is a *dispersion* of stopping points around the best stopping point, and this increases the expected generalisation error. Other obvious problems are: there is no guarantee that the validation curve passes through the optimal point, it may go up and down many times during training. The validation set is again a limited sampling and may/can not represent the universe. It also requires crucial decisions regarding selection and ratio of examples to be divided into training and validation set, and selection of what strategy to be followed: leave-one-out, cross-validation, bootstrapping, or bagging.

Nonetheless better generalisation is indispensable to the growth of non-parametric non-linear systems. In this connection, "no free-lunch" theorems have been proposed, *e.g*., [5] which establish that for any algorithms, any elevated performance over one class of problems is offset by performance over another class. However, it is unarguably accepted that the simpler the network the superior is the generalisation. This work facilitates data partitioning and network modularity in an effort to minimising the network and learning complexity which yield improved prediction accuracy and thus offer better generalisation.

3. Partitioning of Pattern Spaces

In the present work we have achieved this partitioning by dividing the pattern space into a set of hyperspherical regions, the data within each hypersphere being learned by individual networks which are then combined. We have employed hyperspheres since these are geometric primitives which require comparatively few defining parameters; the technique is general, however, and any closed geometric primitive can be used. We have solved the highly problem-dependent partitioning task using a multiobjective genetic algorithm to optimise: the number of hyperspherical partitions, and their location and their radii.

If we consider feature-partitioning as a mapping $\mathbb{P}$ from an N dimensional feature space to j subspaces of dimensionality $n_{j,}$ then this formulation is an N dimensional function decomposition into many n_j - dimensional sub-functions subject to meeting certain criteria, $Obj\_f_i(\mathbf{X})$. Since n_j represents (hopefully) a less complex domain, a classifier can approximate such a sub-domain with less effort; one of the measures of complexity we employ is the local *intrinsic* dimensionality within a hyperspherical partition.

4. Genetic Optimisation of Partitioning

In terms of complexity, the above partitioning problem is NP-complete and genetic algorithms have been shown to be highly effective for exploring NP-complete search spaces compared to exhaustive search. GAs yield near-optimal solutions rather than an exact solution but have the advantage of not needing prior knowledge of the pattern space; the number of partitions that emerges from genetic search can be guided solely by the optimisation criterion and does not need to be pre-determined by the user.

In this work we introduce the approach of using a multiobjective genetic algorithm to partition the pattern space into hyperspheres for subsequent mapping onto a hierarchical neural networks for subspace learning. In our technique, clusters are generated on the basis of 'fitness of purpose' - that is, they are explicitly optimised for their subsequent mapping onto the hierarchical classifier - rather than emerging as some implicit property of the clustering algorithm. Multiobjective genetic algorithm perform optimisation on a vector space of objectives - see [6] for a review - and are able to explore the NP-complete search space for a set of equivalent partitions for pattern space.

We have identified a set of seven independent objectives for pattern space and optimising learning effort - for details see [6]. All seven elements in the objective vector are distinct and competing as well as complementary to each other and ensure a fair distribution of potential solutions. Rather than using an *ad hoc* linear combination of seven objectives we have employed the notion of Pareto optimality in which the superiority of one solution over another is measured in terms of 'dominance' resulting in a Pareto-optimal set which lies on a surface in the objective 7-space. To represent sub-space partitions, our GA implementation uses variable length individuals where each sub-block encodes the hypersphere centre and radii.

5. Hierarchical Neural Learning

From the obtained set of (near-) optimal Pareto solutions, a small subset based on sub-ranking of objectives was picked for hierarchical learning. The *ANCHOR* connectionist architecture [7], which we have developed for integrating multiple heterogeneous classifiers, is particularly suitable for hierarchical learning of subspaces. In terms of modularity, *ANCHOR* is designed to integrate arbitrarily heterogeneous neural nets in hierarchical nesting or cascade along with non-neural processing modules. *ANCHOR* supports multiple instantiations of a network; the notion of modularity demands that different networks learn similar training patterns differently and thus different mappings. This property has implications for learning and to the situations where multiple instantiations acquire different net-topologies and connection-strengths, which is crucial for generalisation.

6. Results

We have partitioned a benchmark problem in land-use classification of multi-spectral satellite image data of thirty-six dimensions. The dataset description and classification results for various algorithms are given in Taylor *et al* [8] where the best classification accuracy was obtained with a *k*-NN classifier. Results for two hyperspheres produced some partitions which contained only members of one class and a roughly 50:50 split in the other partition. Most solutions however included a hypersphere containing around ten members of the other class. This latter situation is an unattractive partitioning since *within* one of the hyperspheres, one class has a very small prior, which would lead to difficulties in reliably training a neural network. Partitions based on four clusters produced mostly hyperspheres of a single class

together with hyperspheres of roughly 50:50 membership. Six partitions produced very similar results to the four partition case except that the aggregate overlap measure was increased and a few of the hyperspheres were degenerate in that they largely or wholly overlapped other hyperspheres. We draw the conclusion that six clusters is indeed too many for this particular problem.

7. Discussion & Conclusions

Looking at the composition of clusters across all the Pareto-optimal solutions, we observed that only one or two clusters in each solution need post-partitioning classification. This is exactly what we are aiming with this evolutionary-neural approach since the localised decision surfaces should possess considerably reduced complexity over the global decision surface. In this work we have attempted solutions of complex problems of high dimensionality, in a generic manner where there is no prior knowledge of pattern space which could guide clustering, using multiobjective genetic algorithms with hierarchical neural learning. In this work, we have geared our vector of objectives to mapping onto an ensemble of multi-layer feedforward neural networks but clearly any desired set of objectives could be employed to maximise the fitness for some other purpose.

The strategy adopted in this work also supports the concept of ensemble-based approaches. Ensemble based approaches rely on integrating multiple classifiers to improve prediction accuracy by repeatedly mapping the *whole* data set on multiple models. In our approach the clusters, which contain only one data class, do not require any further processing and are implicitly labelled without ambiguity. In a partitioned ensemble approach, we suggest that only those clusters where more than one class is represented need to be multiply mapped on suitable classifiers. Thus the principal advantage of our partitioning approach is that *only* those patterns, which lie near decision boundaries, warrant learning effort, possibly multiple efforts for enhanced accuracy. Thus this evolutionary-neural approach simplifies the functional mapping, enhances the accuracy and offers better generalisation.

8. References

[1] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma", *Neural Networks*, vol. 4, no. 1, pp. 123-1232, 1994.

[2] R. Reed, "Pruning algorithms - a survey", *IEEE Trans. Neural Networks*, vol. 4, no. 5, pp. 740-747, 1993.

[3] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm", *Neural Computation*, vol. 6, no. 2, pp. 181-214, 1994.

[4] A. S. Weigend, "On overfitting and the effective number of hidden units", in *Proc. 1993 Connectionist Models Summer School*, Hillsdale, NJ: Lawrence Erlbaum, 1994, pp. 335-342.

[5] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimisation", *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, 1997.

[6] R. Kumar and P. I. Rockett, "Multiobjective genetic algorithm partitioning for hierarchical learning of high dimensional pattern spaces: a Learning-follows-decomposition strategy", *IEEE Trans. Neural Networks*, vol. 9, no. 5, pp. 882-830, 1998.

[7] R. Kumar and P. I. Rockett, "*ANCHOR* - a connectionist architecture for hierarchical nesting of multiple heterogeneous neural nets", in *Proc AAAI Workshop 'Integrating Multiple Learning Models (IMLM 96)*', Menlo Park, CA: AAAI Press, 1996, pp. 59-65.

[8] C. C. Taylor *et al*, "Dataset description and results", in D. J. Spiegelhalter & C. C. Taylor, Eds., *Machine Learning, Neural and Statistical Classification*, London: Ellis Horwood, 1994.