

Design of Experiments by Committee of Neural Networks

Nicolas Gilardi and Abdelaziz Faraj

Division TIMA,

Institut Français du Pétrole

1 & 4 avenue Bois Préau

92500 Rueil-Malmaison, France

E-mail: nicolas.gilardi@ifp.fr ; abdelaziz.faraj@ifp.fr

Abstract—In this paper, we present a way of constructing Design of Experiments for neural networks models such as Multi-Layer Perceptron (MLP). We are trying to solve the problem of modeling a phenomenon with a minimum of measurements and almost no *a priori* knowledge. Our method is based on Query By Committee (QBC) which compares the predictions of various models on unsampled locations in order to select the most informative. We compare it to a random selection of samples.

I. INTRODUCTION

How to efficiently model a phenomenon when almost no information is available and measurement is expensive ? This is a common question in experimental sciences and industry. Formalised by Kiefer and Wolfowitz [5], optimal designs of experiments have contributed to solve this problem. However, this approach have long been restricted to models linear in their parameters, and with a well defined structure. MacKay [8] and Cohn [2] have proposed some solutions to apply optimal design of experiments to neural networks. The main idea was to use a linear approximation of the neural network model to get back to a “classical” situation. It was then possible to iteratively converge to an optimal design, for example using D-Optimality [3][9]. Unfortunately, the optimal structure of the network has to be known *a priori*. In addition, the first order approximation can be far from the model’s behaviour.

The approach that we propose tries to overcome the constraint on the model’s structure. To reach this, we are using the Query By Committee method, developed by Seung *et al.* for classification problems [10], and extended to regression tasks by Krogh and Vedelsby [6]. We also consider that no *a priori* information is available on the problem, except its experimental domain and the fact that it can be modeled using a Multi-Layer Perceptron. We also suppose that we have an idea of what is an “efficient” model in this context.

First, we will describe the QBC method and the way we are using it. Then, we will describe the comparison process, i.e. the experimental protocol, the two methods in “competition”: random and QBC, and the application data. Finally, we will present the results in term of training set size and associated testing error. We will conclude on the advantages of our method and its possible improvements.

II. QUERY BY COMMITTEE

In the field of Machine Learning, it is widely accepted that a good training set has to be composed of a large set of measurements, randomly and uniformly taken from the input space (i.e. the experimental domain). This is the easiest way to expect a good representation of the output’s distribution. However, when measurement has a great cost, such a training set is difficult to obtain. This is the reason why Machine Learning people have created the domain of Active Learning, in order to adapt the optimal design of experiment ideas to learning algorithms.

Since then, many methods have been proposed (see [4] for a review), among which the Query By Committee [10].

A. Main Idea

One is looking for modeling an unknown function:

$$f : \mathbb{R}^p \mapsto \mathbb{R}$$

for which one has a set of n measurements

$$\mathcal{Z} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$$

which is the training set. The objective is to maximise the information available in this set by adding new data in it. By doing this, one is expecting to get a good estimation of f using a minimum of measurements. The new data points have to be chosen without *a priori* knowledge on their measure. It is thus necessary to find other criteria.

The solution proposed by QBC is to use the training set \mathcal{Z} to construct m models $\hat{f}^{(1)}, \dots, \hat{f}^{(m)}$. They will form the *committee*. These models need to have initial differences in order to explore the various pieces of information owned by the training set. Once constructed using a classical training procedure (like in [1]), each model can be used to estimate the output of any location of the experimental domain \mathcal{X} .

One is then going to present possible new points to the m models of the committee. For each point $\mathbf{x} \in \mathcal{X}$, one will get m estimations of the measure: $\hat{f}^{(1)}(\mathbf{x}), \dots, \hat{f}^{(m)}(\mathbf{x})$. If all the models are predicting more or less the same value for $f(\mathbf{x})$, this means that adding this point to \mathcal{Z} will not have much influence on the training process of the committee. On the other hand, if the models can not agree on this estimation,

this means that \mathbf{x} lies in a region of \mathcal{X} which is not efficiently described by the training set. Measuring $y = f(\mathbf{x})$ and adding this new datum to \mathcal{Z} should thus significantly improve knowledge about the phenomenon.

The main idea of QBC is to compare the predictions done by the members of the committee. Points causing the highest disagreement are added to the training set. The disagreement $D(\mathbf{x})$ can be calculated in various ways. The most intuitive has been proposed in [6]. It is related to the variance of the estimations:

$$D(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - \bar{y})^2 \quad (1)$$

where \bar{y} is the mean of the estimations $\hat{y}^{(i)}$ from the committee. The higher the variance, the greater the information brought by the point.

B. Implementation

When the models of the committee are MLPs, the disagreement surface defined by D over \mathcal{X} is smooth. It is thus possible to choose the new points by looking for local maxima. To do so, one is randomly generating points, uniformly distributed over \mathcal{X} . They will be the starting locations of various optimisation procedures trying to maximise $D(\mathbf{x})$. The new points to be added in the training set will be chosen inside the set of local maxima. The decision to keep a point or not is related to the number of points one wish to add to the training set. Considering a minimum distance between two measurement locations is also important and points too close to each other have to be merged.

The complete QBC procedure is the following:

- 1) Define the size of the committee m , and the minimum distance d to consider between points.
- 2) Construct m optimal MLP models of the training set. The initialisation of the MLPs is random.
- 3) Find the local maxima of the disagreement function $D(\mathbf{x})$ defined in equation 1.
- 4) Merge the points whose separation distance is smaller than d .
- 5) Take the points with the highest disagreement and return the requested number of them as the design of experiment.

The models of the committee are built in a usual way[1][7]. For example, one can tune the number of hidden units for capacity control. In this case, the optimal number can be chosen by k-fold cross-validation on the training set. Once found, the optimal model is built using the whole training set.

An important consequence of training all the committee models independently is that they may have different optimal number of hidden units. There is no *a priori* on the model's structure.

III. EXPERIMENTS

A. Description of the Method

Our aim in this article is to built a training set as small as possible in order to reach an acceptable precision quality. We

will illustrate this by comparing QBC to a random selection of new training data.

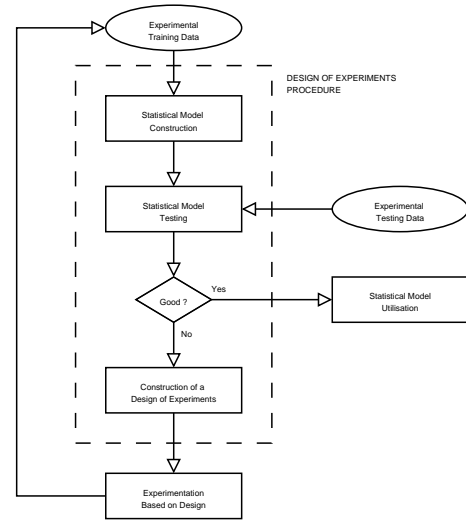


Fig. 1. General procedure for the construction of an optimal statistical model using design of experiments.

The two approaches will follow the general procedure described in figure 1:

- 1) Build an experimental testing set.
- 2) Build an initial experimental training set.
- 3) Construct an optimal statistical model on the training set.
- 4) Estimate generalisation error of the model on the testing set.
- 5) If generalisation performance is good enough, return the optimal model and stop the procedure. Else, proceed to the construction of a design of experiments.
- 6) Measure the data proposed by the design of experiments and add them to the training set.
- 7) Return to 3 until the maximum number of allowed experiments have been reached.

In this paper, the statistical model is a Multi-Layer Perceptron whose number of hidden units is determined by k-fold cross-validation on the training set. The optimal model is then built using the whole training set.

At each iteration, the design of experiments will contain a fixed number of locations to measure. These locations are selected either randomly or using the QBC procedure described in section II-B.

B. Application

The application we propose in this paper is a meta-modeling problem. One have a complex physical model which is very costly to run. The idea is to approximate it with a MLP model in order to gain in calculation time, and not to loose too much in precision. The number of input variables as been limited to 3, and there is only one output to predict.

Actually, we will not generate our data from the original physical model, but from an interpolation spline modeling

a 3000 experiments database. The database itself has been generated from the physical model.

Such a modeling problem is intractable using a classical design of experiment approach. The main reason is that our data are noiseless. Classical design of experiment methods need uncertainty in the data. A possible solution might be to add artificial noise to the measures. This is not necessary for QBC.

We are fixing our total number of experiments for being less or equal to 140 points. We are generating a testing set of 40 points in order to have a good evaluation of the generalisation error. Our maximum number of training data is thus 100. The initial training set is limited to 30 points. Each design of experiments will propose 5 new points to measure and add to the training set. The choice of this value is mainly empirical. On low dimensional data the usual number of “interesting” local maxima of the disagreement function randomly ranges from 1 to 10. By imposing to take 5 new points, we are taking the risk of introducing less information than available, but we are insuring that the algorithm will really progress at each iteration (which would not be the case if too few points were added).

The procedure will stop either if the generalisation error falls below 0.11^1 or if we reach the maximum number of experiments.

In order to evaluate the robustness of the method to various initial data sets, one will generate 10 different training sets. The testing set will remain the same. Both methods, i.e. random sampling and QBC, will start with the same initial training set.

Concerning the size of the committee, the rule of thumb is of course “bigger is better”. However, the bigger the committee, the more time it takes to build it. As a consequence, only 10 MLPs will be trained at each iteration in our experiments.

C. Results

Figures 2a and 2b show the evolution of the testing error while increasing the size of the training set. The figures represent the results for the 10 initial training sets. This testing error results from the construction of the optimal statistical model on each upgraded training set, as described in the general procedure of section III-A.

First, it is interesting to notice that for any given initial training set, QBC designs allowed to reach the minimum requested testing error. On the other hand, random designs succeeded 6 time over 10. However, the most important result is that QBC designs seems to converge faster to an acceptable solution than random designs. QBC proposed 9 time a final design in less than 100 training data, versus only 3 time for random sampling. It is thus possible to conclude that, on this data set, QBC designs extracted significantly more information than random designs.

QBC error curves also appear to be smoother than the random sampling one. This is an indication that QBC designs

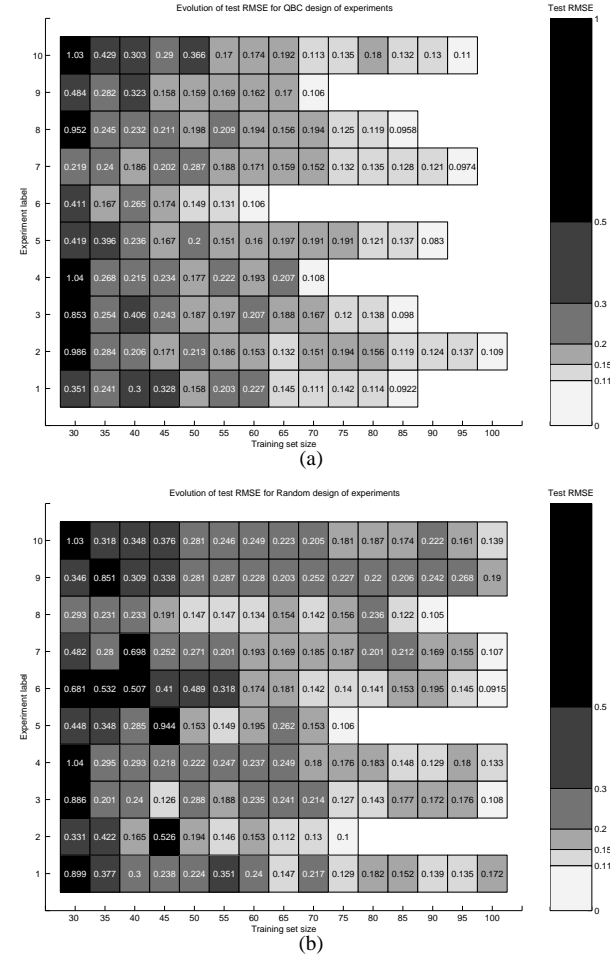


Fig. 2. Evolution of the root mean squared error on the testing data for QBC designs (a) and Random designs (b). Rows indicate the label of the original training set used (from 1 to 10). Columns indicate the training set size (from 30 to 100 points). Exact RMSE values appear inside each cell.

are producing somehow “robust” training set. However, the convergence speed isn’t robust at all. Depending on the initial training set, the QBC design ranges from 60 to 100 points, which is quite large. In our experiments, initial training data were chosen randomly. Further investigations are needed to understand which kind of training set should be given to the algorithm in order to speed up the convergence.

IV. CONCLUSIONS

After presenting the main idea behind QBC, we have shown that this method can be used to create efficient designs of experiments for MLP. The most interesting aspects of this method are that no *a priori* information about the data or the model structure is needed and that it leads to significantly smaller training data sets than a random sampling for a given generalisation performance. The main drawback is its dependency to the initial training set which can nearly double the size of the final design. The growth in performance compared to a random sampling is also smaller than expected. Finally, the number of “measurement campaigns” can be quite large.

¹this value is 5% of the standard deviation of the experimental data base

As a consequence, this method have to be improved. Various directions can be explored, in particular for de definition of the disagreement function. Modifying it might increase performances, both in term of number of data and in prediction. The various hyper-parameters of the method (size of the committee, tolerance on the minimum distance) also have to be studied more deeply in order to find a relationship between them and the number of new data points to be chosen. Last but not least, the problem of the initial training set have to be investigated very carefully.

ACKNOWLEDGMENT

The authors would like to thank the Neuropex Consortium members for the many interesting discussions on design of experiments which motivated this work.

REFERENCES

- [1] C. Bishop, "Neural Networks for Pattern Recognition", Clarendon Press, Oxford, 1995.
- [2] D. Cohn, "Neural Networks Exploration Using Optimal Experiment Design", *Advances in Neural Information Processing Systems* 6, 1994.
- [3] J.P. Gauchi, "Plans d'Expériences Optimaux pour Modèles de Régression Non Linéaire", *Plans d'Expériences, Applications à l'Entreprise* Chap. 8, Dreesbeke, Fine and Saporta éditeurs, Paris, 1997.
- [4] M. Hasenjäger and H. Ritter, "Active Learning in Neural Networks", *New Learning Techniques in Computational Intelligence Paradigms*, L. Jain editor, CRC Press, 2000.
- [5] J. Kiefer and J. Wolfowitz, "Optimum Designs in Regression Problems", *Annals of Mathematical Statistics*, 1959.
- [6] A. Krogh and J. Vedelsby, "Neural Networks Ensembles, Cross Validation, and Active Learning", *Advances in Neural Information Processing Systems* 7, Cambridge MA, 1995.
- [7] Y. LeCun, L. Bottou, G.B. Orr and K.R. Miller, "Efficient BackProp", *Neural Networks: tricks of the trade*, Springer, 1998.
- [8] D. MacKay, "Information-based Objective Functions for Active Data Selection", *Neural Computation* 4, 1992.
- [9] R. Myers and D. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 2nd Edition, Wiley, 2002.
- [10] H. Seung, M. Opper, and H. Sompolinsky, "Query By Committee", *Proceedings of the Fifth Workshop on Computational Learning Theory*, San Mateo CA, 1992.