

Evolutionary Search for Interesting Behavior of Neural Network Ensembles

Pavel Kordík, Jan Saidl, Miroslav Šnorek

Abstract— Very useful outcome of a neural network model is that relationship of input and output variables can be plotted revealing some potentially interesting information about a modeled system. However this approach is not often used because there are several problems appearing from a closer look. At first there is a problem with the "curse of dimensionality", secondly the problem of model credibility arises when system state space is not fully covered by training data. There are also problems with irrelevant input variables, with the time needed to find some useful plot in multidimensional state space, etc. This paper shows that all these problems can be successfully overcome using modern techniques of evolutionary computation and ensemble modeling. The result of our research is an application that is able to automatically locate interesting plots of system behavior.

I. INTRODUCTION

Models based on neural networks are often called "black box" models. The reason for that is that the information gained by training is hidden inside the network - compressed into weights of neurons. It is impossible to interpret the information stored in hundreds of weights. That is why neural networks are often treated unreliable. Recently there are attempts to open the black box by visualizing values of weights and topology in a clever way [1], but this approach is constrained to a limited number of problems and neural network topologies. We offer fundamentally different approach to open the black box of neural networks. Our approach is applicable to a wide spectrum of inductive models in general.

A. Plots of model behavior

Very useful visualization technique is to plot the output sensitivity of a model to the change on one particular input¹. This plot can reveal the relationship of selected input and the output variable of the model in conditions given by values of other inputs staying constant. Figure 1 illustrates simple example, how this plot can be made using any black-box model in general. If the curve would be constant the feature x_2 would be irrelevant in our conditions (no output reaction to the change of the feature).

B. Ensembling: what do we mean by "interesting behavior"?

Real-world data usually does not cover whole state space of features (we do not have measurements for all possible

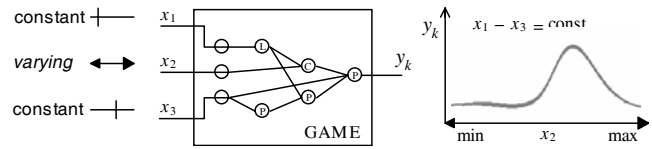


Fig. 1. While x_2 is varying from min to max and other inputs stay constant, the output of the model shows sensitivity of the model to variable x_2 in the configuration $x_1, x_3 = const$.

combinations of input variables). Features are seldom independent and the correlated data are normally distributed around some cluster centers. The rest of the input state space is empty.

When we use such training data to build a model, responses of model would be random in areas, where training data are absent. The problem is that we cannot simply find the border of model credibility. If we limit the credibility just to areas of training data presence, we cannot deal with irrelevant features. Therefore we use the Ensemble Techniques [3]. Ensemble models is a group of diverse models build on similar training data [6]. In our research we use several techniques (Bagging, Niching Genetic Algorithm, heterogeneous neurons, etc.) to promote diversity that is particularly important in our problem. Ensemble models helps us to find out areas where models are credible and their behavior is "interesting".

When you look at the Figure 2, you can see, how we defined

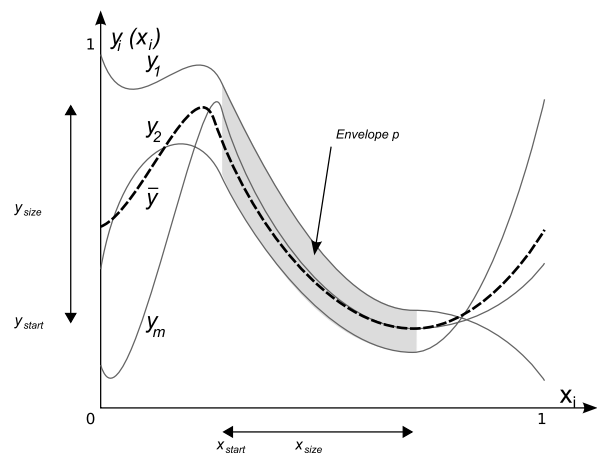


Fig. 2. Interesting behavior of models

the term "interesting". Each curve represents the output of one ensemble model y_i when a feature x_i is changed. The more rapid change we observe ($y_{size} \rightarrow max$), the more

Pavel Kordík, Jan Saidl, Miroslav Šnorek are from the Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic (emails: {kordikp, snorek}@fel.cvut.cz, jan@saidl.info).

¹Terms input variable, feature and input are equivalent

interesting behaviour is displayed. The second criterion for the importance of projection is the credibility of models. We need to look for rapid changes of output variable just within the areas, where models are credible. This can be achieved by simple assumption. The random (different) output of ensemble models signify that we are outside the area of credible behavior. The equal output means that all models converged to single (credible) value that is based on training data. The second criterion can be therefore computed from dispersion of ensemble model outputs - the envelope p should be minimized:

$$p = \sum_{j=x_{start}}^{x_{start}+x_{size}} \left(\arg \max_{0 < i \leq m} (y_i(j)) - \arg \min_{0 < i \leq m} (y_i(j)) \right) \quad (1)$$

The third criterion $x_{size} \rightarrow \max$ helps us to privilege bigger areas of interesting behavior.

II. EVOLUTIONARY SEARCH ON SIMPLE SYNTHETIC DATA

The optimization problem how to find optimal values of constant features (x_j where $j \in (1, n)$, $j \neq i$ and n is number of features), the start ($x_{i-start}$) and the size (x_{i-size}) of the area can be solved by evolutionary computation. We performed an experiment with a synthetic data to find the best properties of the search method: genetic algorithm [9]. The synthetic data we generated are depicted in the Figure 3. In the left part of the Figure 3 you can see the training data

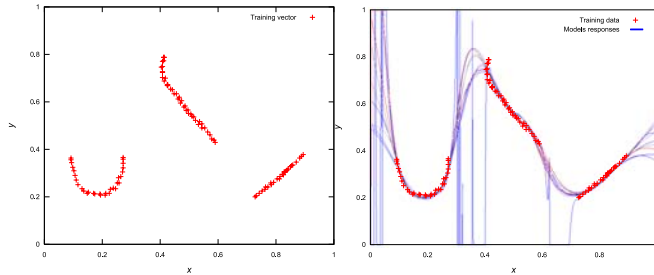


Fig. 3. Synthetic training data and ensemble models approximating it.

used to generate the ensemble of GAME (Group of Adaptive Models Evolution, see [2]) models. The outputs of ensemble models (curves) are in the right part of the figure, together with the training data.

To be able to use genetic algorithm to find areas of interesting behavior, we have to construct the genotype of individuals, genetic operators and the fitness function. In this simple problem, models have just one input x . The genotype will therefore contain just two genes (x_{start}, x_{size}). As genetic operators we employed the commonly used mutation and crossover for real valued genes - with the constrain, that $x_{start} + x_{size} < 1$ for normalized features. The fitness function is derived from the three criteria of interesting projection discussed above. To find suitable fitness function, we had been experimenting with several equations.

The best properties, as you can see on the Figure 4, has

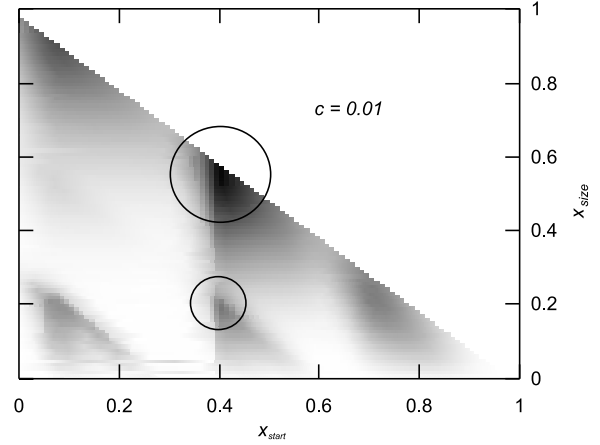


Fig. 4. Fitness function for all possible individuals. Darker background signifies higher value of fitness function.

the fitness function of the following type:

$$fitness = y_{size} * \frac{1}{p} * x_{size},$$

where

$$y_{size} = \arg \max (\bar{y}(t)) - \arg \min (\bar{y}(t)), t \in (x_{start}, x_{size}).$$

The term $\bar{y}(t)$ is in fact Simple Ensemble of models [5] defined as $\bar{y}(t) = \frac{1}{m} \sum_{i=1}^m y_i(t)$ for $t \in (0, 1)$. This term represents the first criterion in the fitness function. The second criterion (term p) can be computed according to Equation 1 and the third criterion (x_{size}) is directly encoded in the genotype.

We started genetic algorithm and after 50 generations all individuals had the same genes ($x_{start} = 0.4, x_{size} = 0.5$). As you can check on the Figure 4, the global maximum of the fitness function was found. The individual with such properties is also shown on the Figure 5.

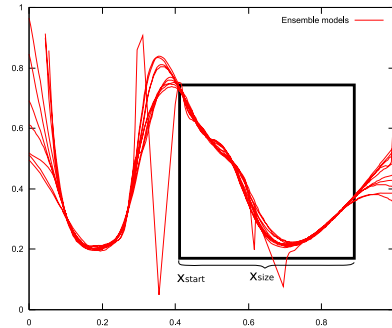


Fig. 5. The individual with the highest fitness dominated the population of genetic algorithm.

A. Experiments with diversity

We demonstrated that using genetic algorithm and well chosen fitness function, we are able to locate the area of the most interesting behavior according to definition in subsection I-B of this paper.

We also want to locate other interesting areas, that are not the best, but carry also very important information about input-output relationship. For this reason we needed to promote diversity in the population of genetic algorithm. Once the population preserves also suboptimal solutions, other interesting areas can be found.

The technique allowing us to preserve diversity in genetic algorithm is called niching [7]. To be able to say if individuals are diverse, the distance operator has to be introduced. We used so called genotypic distance of individuals that can be computed as $d(\vec{x}, \vec{z}) = \sum_{i=0}^n (x_i - z_i)^2$, where \vec{x} and \vec{z} are genes of two individuals. Figure 6 shows this distance (diversity) in the matrix for all individuals in the population of standard genetic algorithm. The darker is a

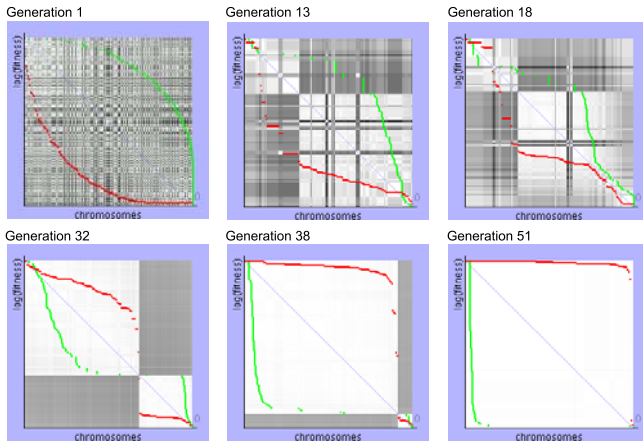


Fig. 6. Diversity in population for the standard genetic algorithm.

point in the matrix, the more diverse are individuals with indexes corresponding to particular row and column of the matrix. You can observe that after 50 generations, there is no diversity among individuals - all are equal to the one depicted on the Figure 5.

Then we used niching technique in the genetic algorithm. We employed Deterministic Crowding that has many advantages over other niching methods [7]. The results can be observed

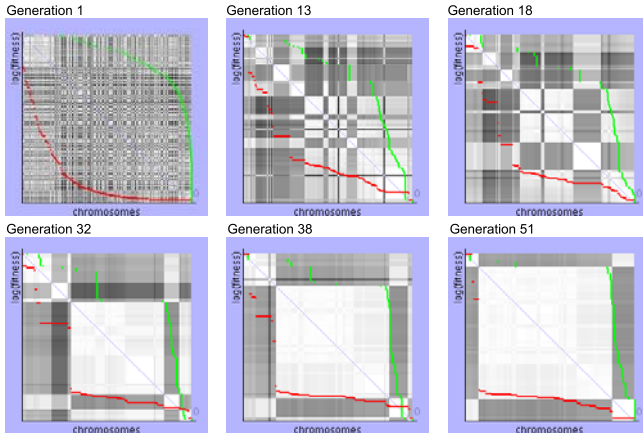


Fig. 7. Diversity in population for the niching genetic algorithm (Deterministic Crowding employed).

on the Figure 7. After 50 generations there are still three different individuals in the population. One represents the optimal solution, two other represents suboptimal solutions that are apparent also in the Figure 4. The properties of all three individuals are shown on the Figure 8.

It is necessary to say, that niching does not preserve subopti-

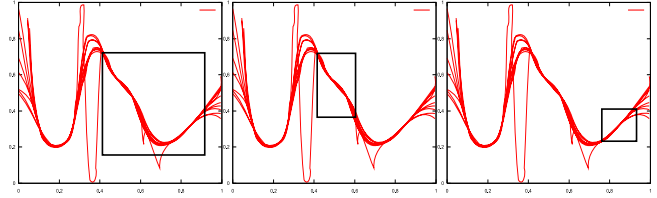


Fig. 8. Three solutions found by the niching genetic algorithm.

mal solution in the population forever. It is important to stop the algorithm before diversity of individuals drops to zero. The niching is particularly important for more complex problems, where it prevents the algorithm from early convergence and also maintains stable subpopulations.

III. STUDY WITH MORE COMPLEX SYNTHETIC DATA

To prove that our algorithm works also for more dimensional data, we prepared synthetic data set with three input variables. We generated vectors just in four clusters

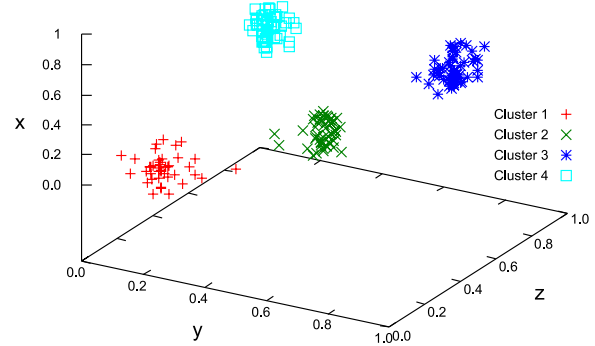


Fig. 9. Input vectors used for generating training data are concentrated in clusters.

(see Figure 9) to emulate real world data with partial definition of the input state space. Total 160 training vectors $\vec{t} = (x_1, x_2, x_3, f(x_1, x_2, x_3))$ were generated, where $x_1, x_2, x_3 \in (0, 1)$ and

$$f(x_1, x_2, x_3) = \frac{\sin(2\pi x_1) + (2x_2 - 1)^2 + x_3 + 1}{4}.$$

Training data were used again to generate ensemble of GAME models. After that we executed the genetic search for interesting behavior again. For this experiment, we had to run the algorithm for three times - for each feature separately. For the first feature (x_1) the genotype of individuals was $x_2, x_3, x_1\text{-start}, x_1\text{-size}$, for other two features the genotype was similar. The Figure 10 shows the best individuals from the population after ten and fifty generations for each feature. We can observe dramatic improvement in the quality of

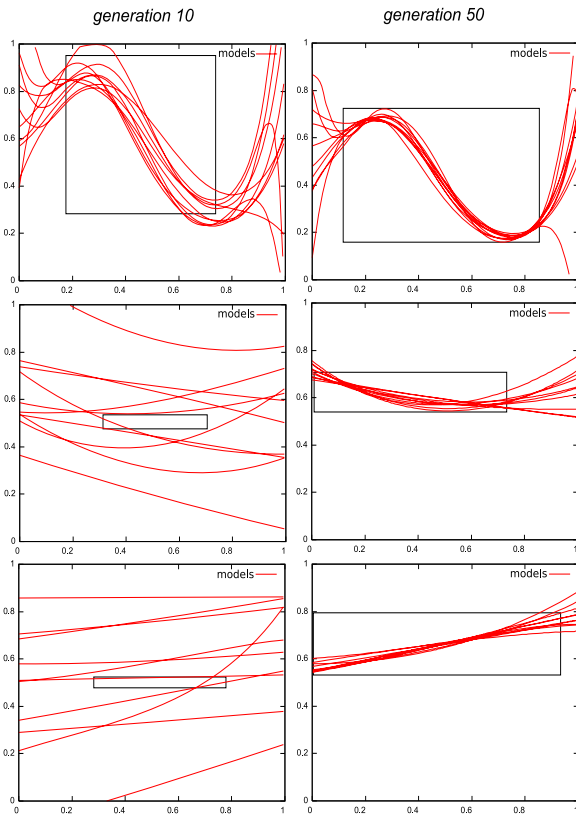


Fig. 10. The best individuals after ten and fifty generations for features x_1, x_2, x_3 .

projection. The resulting plots are much more interesting than the best plots in the generation 10.

IV. EXPERIMENTS WITH REAL WORLD DATA

The "Building data set" is frequently used for benchmarking modeling methods [4]. It consists of more than three thousand measurements inside the building (hot water (*wbhw*), cold water (*wbc*), energy consumption (*wbe*)) for the specific weather conditions outside (temperature (*temp*), humidity of the air (*humid*), solar radiation (*solar*), wind strength (*wind*)). We excluded the information about time of measurement. On this data set we ensemble of GAME models for each output (*wbc*, *wbhw*, *wbe*). Then we executed the genetic algorithm four times (4 features) for each output. The most important genes of best individuals found in 12 runs are summarised in the Table I.

In the Figure 11 there are projections with higher fitness (the most interesting) showing the relationship of the feature *temp* and output variables *wbc*, *wbhw* and *wbe*. We found out, that the values of fitness found for each feature significantly correlate with the significance of feature, we derived by means of other technique [2], and can be therefore used also for the feature ranking [8].

V. CONCLUSIONS

In this paper we presented the technique that can be used for automatic extraction of interesting plots showing input-

TABLE I

CHROMOSOMES OF THE BEST INDIVIDUALS AND THEIR FITNESS.

Output	<i>temp</i>	<i>humid</i>	<i>solar</i>	<i>wind</i>	Fitness
wbcw	-	0.2	0.3	0.12	17.01
wbe	-	0.43	0.09	0.12	3.96
wbhw	-	0.23	0.23	0.57	38.29
wbcw	0.74	-	0.47	0.17	8.93
wbe	0.71	-	0.08	0.74	1.8
wbhw	0.84	-	0.23	0.58	2.75
wbcw	0.57	0.33	-	0.2	2.51
wbe	0.57	0.21	-	0.41	4.96
wbhw	0.71	0.67	-	0.6	1.33
wbcw	0.66	0.26	0.92	-	2.58
wbe	0.85	0.45	0.395	-	5.18
wbhw	0.6	0.36	0.31	-	1.96

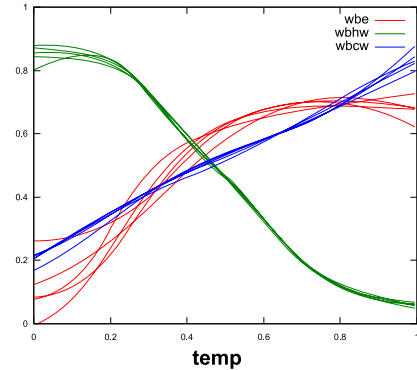


Fig. 11. Three most interesting plots showing relationship of the feature *temp* and outputs *wbc* (fitness 17.1), *wbhw* (fitness 38.29) and *wbe* (fitness 3.96).

output relationship of a system modeled by ensemble of neural networks. This technique can make data analysis much more effective. It also minimizes the need to interact with neural network and reveals some useful knowledge hidden inside. All experiments were performed using the application GAME, that can be downloaded from [10] so you can try, if it also works on your data.

REFERENCES

- [1] Fan-Yin Tzeng, Kwan-Liu Ma, *Opening the Black Box - Data Driven Visualization of Neural Networks*, Proceedings of IEEE Visualization '05 Conference, Minneapolis, October 23-28, 2005.
- [2] Kordík, *Group of Adaptive Models Evolution*, Technical Report DCSE-DTP-2005-07, CTU Prague 2005.
- [3] L.K. Hansen, P. Salamon, *Neural network ensembles*, IEEE Trans. Pattern Anal. Machine Intelligence 12 (10) (1990).
- [4] Prechelt, L., *A Set of Neural Network Benchmark Problems and Rules*, Technical Report 21/94, Karlsruhe, Germany 1994.
- [5] Zhi-Hua Zhou, Jianxin Wu, Wei Tang, *Ensembling neural networks: Many could be better than all*, Artificial Intelligence 137 (2002).
- [6] Gavin Brown, *Diversity in Neural Network Ensembles*, Ph.D. thesis, The University of Birmingham, January 2004
- [7] Mahfoud. S. W., *A Comparison of Parallel and Sequential Niching Methods*, Proceedings of the Sixth International Conference on Genetic Algorithms, p. 136-143, 1995.
- [8] Piramuthu, S., *Evaluating feature selection methods for learning in data mining applications*, European Journal of Operational Research 156, p. 483-494, 2004.
- [9] Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [10] The application GAME, <http://neuron.felk.cvut.cz/game/index.html>