Generalization of a Parametric Learning Rule*

Samy Bengio Yoshua Bengio Jocelyn Cloutier Jan Gecsei

Université de Montréal, Département IRO
Case Postale 6128, Succ. "A", Montréal, QC, Canada, H3C 3J7
e-mail: bengio@iro.umontreal.ca

Abstract

In previous work ([4, 2, 1]) we discussed the subject of parametric learning rules for neural networks. In this article, we present a theoretical basis permitting to study the *generalization* property of a learning rule whose parameters are estimated from a set of learning tasks. By generalization, we mean the possibility of using the learning rule to learn solve new tasks. Finally, we describe simple experiments on two-dimensional categorization tasks and show how they corroborate the theoretical results.

1 Introduction

Learning mechanisms in neural networks are usually associated with changes in synaptic efficiency. In such models, synaptic learning rules control the variations of the parameters (synaptic weights) of the network. Researchers in neural networks have proposed learning rules based on mathematical principles (such as backpropagation) or biological analogy (such as Hebbian rules), but better learning rules may be needed to achieve human-like performance in many learning problems.

Chalmers proposed in [5] a method to find new learning rules using evolution mechanisms such as genetic algorithms. His method considers the learning rule as a parametric function with local inputs which is the same for all neurons in the network. He used genetic algorithms to find a learning rule for networks without hidden layers and found that among the class of rules he investigated, the *delta rule* was most often selected and performed best for linearly separable boolean problems.

Independently of Chalmers, we proposed a similar approach in [4] that extends this idea to networks with hidden layers and with the possibility to use any standard optimization methods (such as genetic algorithms, but also gradient descent and simulated annealing). In this paper, we give theoretical principles for the design of such parametric learning rules.

*A one-page extended abstract of this paper has been published in [3].

In section 2 we introduce the idea of parametric learning rules. Section 3 explains how the concept of *generalization* can be applied to learning rules. Finally section 4 shows how practical experiments corroborates theoretical results.

2 Parametric Learning Rules

We describe briefly in this section the basic idea of optimizing learning rules. More detailed treatment can be found in [4, 2, 1], but also in [5]. The principle is straightforward: we consider the learning rule as a parametric function and we optimize its parameters using an optimization algorithm such as gradient descent. In doing so, we make the following hypothesis:

- In a large neural network, there is only a limited number of different learning rules; therefore a given rule is used in a large number of neurons.
- There is a (possibly stochastic) dependency between the synaptic modification and information available locally (in the physical neighborhood of the synapse).
- This dependency can be approximated by a parametric function $f(x_1, x_2, ...x_n; \theta)$ where x_i are the arguments of the function and θ is a set of parameters.

Since the space of possible learning algorithms is very large, we propose to constrain it by considering only a subset of possible parametric functions for the rule. The form of the rule may be inspired by certain known synaptic mechanisms. Thus, the only input arguments considered (the x_i above) are local to the synapse, such as the presynaptic and postsynaptic activities, the synaptic weight, the activity of facilitatory (or modulatory) neurons, or the concentration of a chemical agent (see figure 1).

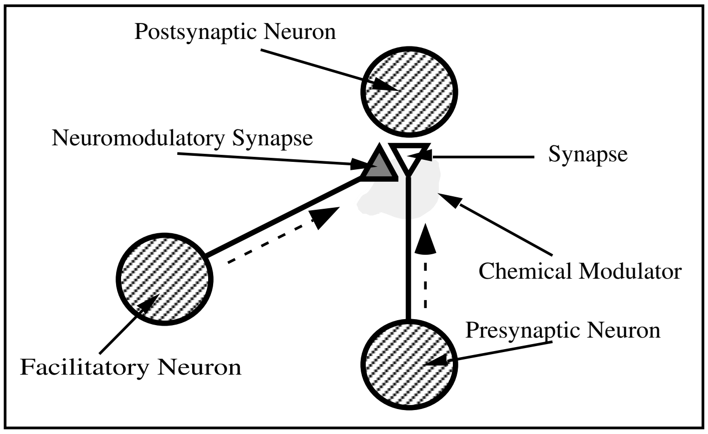

Figure 1: Elements found in the vicinity of a synapse, which can influence its efficacy.

This way of constraining the search space to be biologically plausible should not be perceived as an artificial constraint but rather as a way of limiting the search to a subspace in which solutions share some features with neurobiological learning mechanisms. We suppose that this will facilitate the search for new learning rules. Admittedly, our neuron models are very simplified with respect to what is currently known about the working of the brain; furthermore, many aspects of brain function are still unknown, particularly in what concerns learning and synaptic plasticity.

Let us denote by $w(i,j)$ the weight (efficacy) of a given synapse or set of synapses from neuron i to neuron j. Weight changes proceed according to the following equation:

$$\Delta w(i,j) = \Delta w(\text{local variables at synapse } i \rightarrow j; \theta) \tag{1}$$

The synaptic modification $\Delta w(i,j)$ of the efficacy of the connection from neuron i to neuron j is obtained with the function $\Delta w()$, using the values of local variables at the synapse, and those of a set of parameters, θ. These parameters will be tuned in order to improve the learning rule.

The idea is to try to find a set of parameters θ corresponding to a rule allowing a network to learn to solve different kinds of problems. For this, we can use standard optimization methods (such as gradient descent, genetic algorithms, or simulated annealing). The question addressed in the next section is whether or not we can find a rule that will be able to learn tasks not used to select the parameters θ.

3 Capacity of a Parametric Learning Rule

In order for a learning rule obtained through optimization to be useful, it must be successfully applicable in training networks for new tasks (i.e., tasks other than those used during optimization of the learning rule). This property of a learning rule is a form of *generalization*. We will see that this kind of generalization can be described using the same formalism used to derive the generalization property of learning systems, based on the notion of *capacity*.

3.1 Standard Notion of Capacity

The capacity of a learning system can be intuitively seen as a measure of the cardinality of the set of functions the system can learn. A quantity known as the Vapnik-Chervonenkis Dimension (VCdim) ([6]) is an example of such a measure. $F(\theta) : X \rightarrow Y$ represents a family of functions (parameterized by θ), with domain the set X and image the set Y. X represents the space of examples and Y an error measure.

The capacity h of the learning system $F(\theta)$ is related to generalization error ϵ and number of training examples N in the following way. For a fixed number of examples N, starting from $h = 0$ and increasing it, one finds generalization ϵ to improve (decrease) until a critical value of the capacity is reached. After this point, increasing h makes generalization deteriorate (ϵ increases). For a fixed capacity, increasing the number of training examples N improves generalization (ϵ asymptotes to a value that depends on h). The specific results of [6] are obtained in the worst-case, for any distribution of X.

3.2 Extension to Parametric Learning Rule

We can define the generalization property of a parametric learning rule as its ability to correctly *learn* new tasks, i.e., tasks not used for the optimization of the rule's parameters. The capacity of a parametric learning rule can thus be defined as a measure of the cardinality of the set of learning rules that can be realized. The VCdim, as defined in [6], applies to classes of functions from an arbitrary set to $\{0, 1\}$ (classification problems) or to the reals R (e.g, regression problems). It can be used to measure the capacity of a parametric learning rule if we appropriately define the domain of these functions as *tasks* (e.g. input/output specifications, which may be given for some classes of tasks by a function). The parametric class of functions $F(\theta)$ is defined as a set of functions that take as input a *task* specification and produce as output a performance measure (e.g., average error after training a given learning machine with the learning rule defined by the parameters θ). Here, the "rule generalization error" ϵ is defined as the average error of a learning system trained using the learning rule, where the average is taken over a class of tasks (which should not be confused with the average over examples of a task, as is usually the case).

We can draw several conclusions from this extension. For example, it becomes clear that the expected error of a learning rule over new tasks (ϵ) should decrease when increasing the number of tasks (N) used for learning the parameters θ. However, it could increase with the number of parameters and the capacity of the learning rule class if an insufficient number or variety of training tasks are used in the optimization. This justifies the use of *a-priori* knowledge in order to limit the capacity of the learning rule. It also appears more clearly that the learning rule will be more likely to generalize over tasks which are similar to those used for the optimization of the rule's parameters. In consequence, it is advantageous to use, for the optimization of the learning rule, tasks which are representative of those on which the learning rule will be ultimately applied.

4 Experiments

We performed experiments to verify the theory of capacity and generalization applied to parametric learning rules. In particular, we wanted to study the variation of N, h and the complexity of the tasks, over the learning rule's generalization property (ϵ). Moreover, we did these experiments using three different optimization methods, namely gradient descent, genetic algorithms and simulated annealing. Experiments were conducted in the following conditions:

- The tasks were two-dimensional classification problems. Some were linearly separable (L) while others where non-linearly separable (NL).
- Each task was learned with 800 training examples and tested with 200 examples. A task was said to be successfully learned when there were no classification error over the test set.
- We used a fully connected neural network with two input units, one hidden unit and one output unit. Furthermore, we added a backward path of modulator neurons to provide a measure of the error to each unit.

- We tried two different parametric learning rules. The first rule was defined using biological a-priori knowledge to constrain the number of parameters to 7 (we can find for instance in this rule a Hebbian mechanism):

$$\begin{aligned}\Delta w(i,j) &= \theta_0 + \theta_1\, y(i) + \theta_2\, x(j) + \theta_3\, y(mod(j)) + \\ &\quad \theta_4\, y(i)\, y(mod(j)) + \theta_5\, y(i)\, x(j) + \theta_6\, y(i)\, w(i,j) \qquad (2)\end{aligned}$$

  where $w(i,j)$ is the synaptic efficacy between neurons i and j, $x(j)$ is the activation potential of neuron j (postsynaptic potential), $y(i)$ is the output of neuron i (presynaptic activity), and $y(mod(j))$ is the output of a modulatory neuron influencing neuron j.

  The second rule had 16 parameters and was defined as follows:

$$\begin{aligned}\Delta w(i,j) &= \theta_0 + \theta_1\, y(i) + \theta_2\, x(j) + \theta_3\, y(mod(j)) + \theta_4\, w(i,j) + \theta_5\, y(i)\, x(j) + \\ &\quad \theta_6\, y(i)\, y(mod(j)) + \theta_7\, y(i)\, w(i,j) + \theta_8\, x(j)\, y(mod(j)) + \\ &\quad \theta_9\, x(i)\, w(i,j) + \theta_{10}\, y(mod(j))\, w(i,j) + \theta_{11}\, y(i)\, x(j)\, y(mod(j)) + \\ &\quad \theta_{12}\, y(i)\, x(j)\, w(i,j) + \theta_{13}\, y(i)\, y(mod(j))\, w(i,j) + \\ &\quad \theta_{14}\, x(j)\, y(mod(j))\, w(i,j) + \theta_{15}\, y(i)\, x(j)\, y(mod(j))\, w(i,j) \qquad (3)\end{aligned}$$

- A typical experiment was conducted as follows: We chose a parametric learning rule, an optimization method (genetic algorithms, gradient descent, or simulated annealing[1]), a number of tasks to optimize the rule (1 to 9), and a complexity for the tasks (linearly separable (L), or non-linearly separable (NL)). Then we optimized the rule for a fixed number of iterations, and finally, we tested the new rule over other tasks (that is, try to learn new tasks with their 800 training patterns and evaluate performance with a test over the other 200).

The first experiment verified that the number of tasks N used for the optimization had an influence on the rule's generalization performance. Figure 2 shows that for a given and fixed optimization method and capacity h, generalization error tends to decrease when N increases, as theory assumes.

The second experiment verified if the type of tasks used during optimization had an influence on the rule's generalization performance. Figure 3 illustrates the results. We can see that when the rule is optimized using linearly separable tasks, generalization error on both linearly and non linearly separable tasks stays high, whereas if we use non linearly separable tasks during rule optimization, generalization error decreases when the number of tasks increases.

In the third experiment (figure 4), we verified if the capacity of a parametric learning rule influences its generalization performance. Here, we compared both rules (with 7 and 16 parameters). If the number of tasks used for optimization is too small, the rule with the smallest capacity is better than the other, but the advantage tends to vanish when the number of tasks increases.

[1] In fact, it was not possible to find any good solution using gradient descent. It always fell into a local minima close to the initial set of parameters.

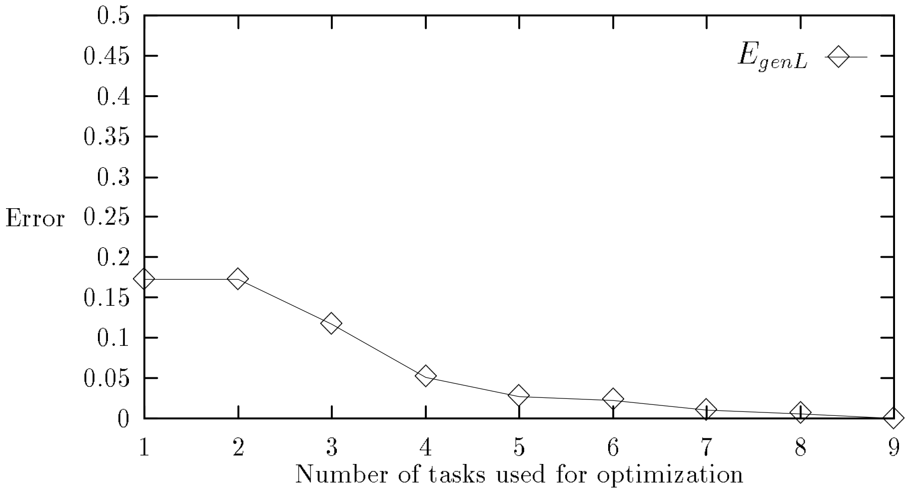

Figure 2: Evolution of generalization error (E_{genL}) with respect to the number of tasks used during optimization. In this example, we used **genetic algorithms**, a rule with **7 parameters**, and **linearly separable** tasks.

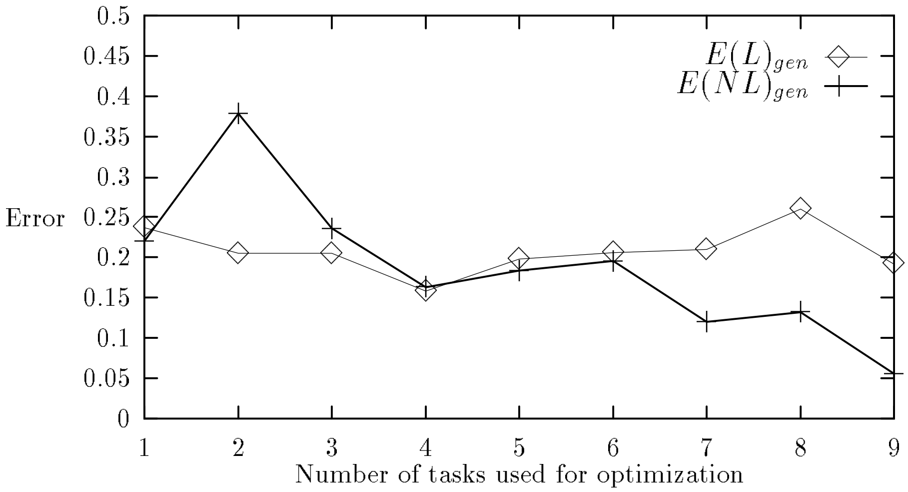

Figure 3: Evolution of generalization error with respect to the task difficulty used during optimization. Here, we used **genetic algorithms** and a rule with **7 parameters**.

Finally, in figure 5, we compare the use of two different optimization methods to find parameters of a learning rule: genetic algorithms and simulated annealing. As we can see, genetic algorithm seem generally better, especially when the number of tasks used for optimization is small.

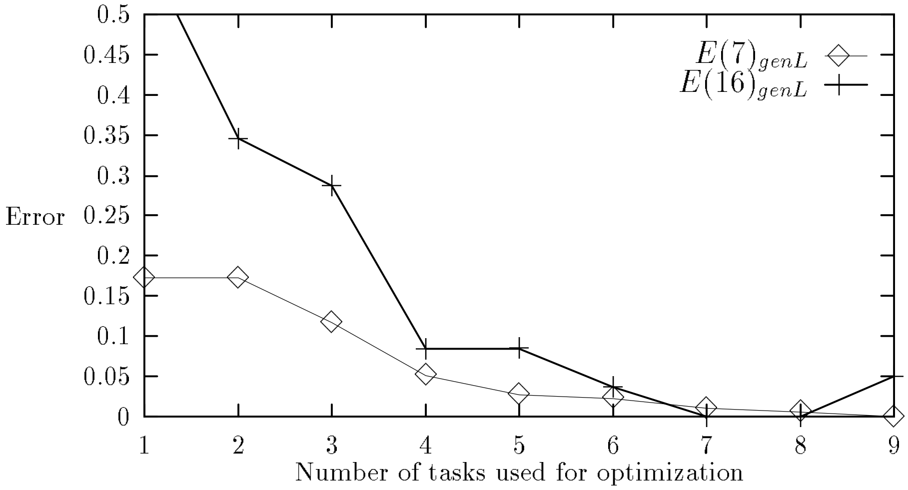

Figure 4: Evolution of generalization error with respect to capacity of the parametric learning rule. Here, we used **genetic algorithms** and tasks were **linearly separable**.

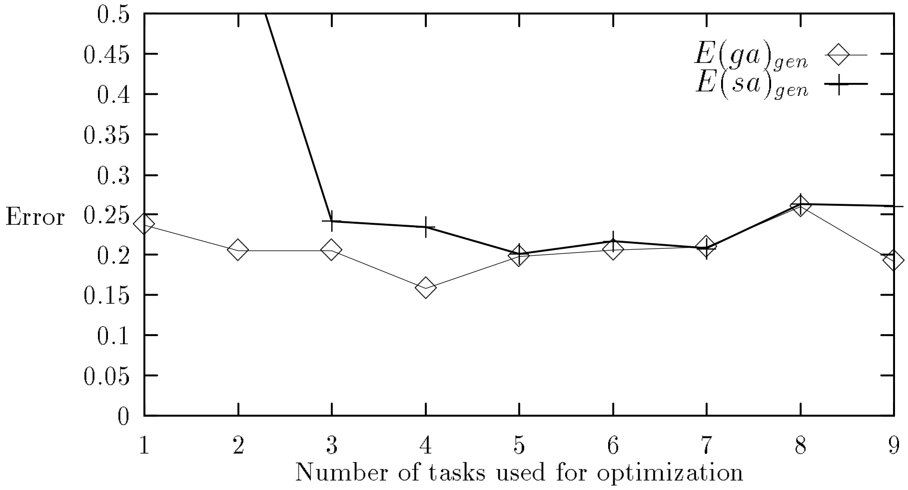

Figure 5: Evolution of generalization error with respect to the optimization method. Here, tasks were **linearly separable** and the rule had **7 parameters**. *ga* stands for **genetic algorithms**, and *sa* for **simulated annealing**.

5 Conclusion

In this article we have established the conceptual bases permitting to study the generalization properties of a learning rule whose parameters are trained on a certain number of tasks. To do so, we have introduced the notion of capacity of parametric learning rules.

Experiments show that it is possible to discover through optimization learning rules capable of solving a variety of simple problems. Using gradient descent, genetic algorithms and simulated annealing, we found learning rules for classical conditioning problems, classification problems, and boolean problems (see [2]). Moreover, the experimental results described in section 4 qualitatively agree with learning theory applied to parametric learning rules. Of course it has to be shown yet that the procedure is applicable to more complex tasks, which will probably require more complex learning rules and yield a more difficult optimization problem.

References

[1] S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei, *Aspects théoriques de l'optimisation d'une règle d'apprentissage*, in Actes de la conférence Neuro-Nimes 1992, Nimes, France, 1992.

[2] ——, *On the optimization of a synaptic learning rule*, in Conference on Optimality in Biological and Artificial Networks, Dallas, USA, 1992.

[3] ——, *Generalization of a parametric learning rule*, in ICANN '93: Proceedings of the International Conference on Artificial Neural Networks, Amsterdam, Nederlands, 1993.

[4] Y. Bengio and S. Bengio, *Learning a synaptic learning rule*, Tech. Rep. 751, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal (QC) Canada, 1990.

[5] D. Chalmers, *The evolution of learning: An experiment in genetic connectionism*, in Proceedings of the 1990 Connectionist Models Summer School, D. Touretzky, J. Elman, T. Sejnowski, and G. Hinton, eds., San Mateo, CA, USA, 1990, Morgan Kaufmann.

[6] V. N. Vapnik, *Estimation of Dependencies Based on Empirical Data*, Springer-Verlag, New-York, NY, USA, 1982.