Sequential learning in feedforward networks: proactive and retroactive interference minimization

Vicente Ruiz de Angulo and Carme Torras

Institut de Robòtica i Informàtica Industrial (CSIC-UPC)
Llorens i Artigas 4-6, 08028-Barcelona, Spain.
email{ruiz, torras}@iri.upc.es,
http://www-iri.upc.es

Abstract. We tackle the catastrophic interference problem with a formal approach. The problem is divided into two subproblems. The first arises when one tries to introduce some new information in a previously trained network, without distorting the stored information. The second is how to encode a set of patterns so as to preserve them when new information has to be stored. We suggest solutions to both subproblems without using local representations or retraining.

1 Introduction

The degradation in performance suffered by a neural network when a new pattern is introduced isolatedly is usually called *interference* or *catastrophic forgetting.* Its extent depends basically on two factors: the type of representation and the training scheme applied.

- Distributed representations are known to be prone to catastrophic forgetting. For the study of interference, we consider that an input is encoded distributedly if most of the network derivatives at the input with respect to the parameters have a significant magnitude [4]. Thus, each item is represented by several parameters and each parameter of the neural net supports partially several items. Local representations can avoid more easily the problem, but unfortunately they exploit memory resources much worse and also are reputed to interpolate less adequately (See [4] for more on this).
- The two basic training schemes are the introduction of the new item isolatedly and the join training of the new item with the old ones. The first one quickly learns the new item at the cost, in many cases, of catastrophic forgetting. The second requires a complex optimization process, especially with distributed representations.

The different previous works on interference can be placed in an imaginary plane whose axes refer to the locality of the representation and the extent of retraining with old items. For example, French's approach [7] is situated at the extreme of localized solutions. He uses sigmoidal units with a $[0, 1]$ range, saturating them and favoring the zero states. Hetherington and Seidenberg [8] suggest to retrain new and old items together after the introduction of the new pattern. This approach falls in the join training extreme in the retraining axis, where information about old items is used in a delayed mode. In [9], the new item is trained with a (fixed or, preferably, randomly changing) *subset* of the previously trained items. Thus, it can be situated in the middle of the retraining axis.

We have contributed to the field using neither local representations nor retraining. Theoretical reasons forbid a perfect solution to the interference problem under these conditions [3]. As a consequence, our goal has been limited to trying to palliate interference as much as possible. In this paper we will place in a common context the results we have obtained with different algorithms.

We consider that the interference problem must be handled in two stages separated by the arrival of the new item:

- A priori stage: encoding the old, known, items in the best way to prepare the network to minimize the impact of learning new items. We call this stage **proactive interference minimization**.
- A posteriori stage: learning the new items while minimizing the damages inflicted on the old items already stored in a given weight configuration. We call this stage **retroactive interference minimization**.

We will concretize the scenario in a set of items $1 \ldots N-1$ which are stored prior to the arrival of the new item N. Let $E_{1 \ldots p}$ denote the error in items $1, 2 \ldots p$. We will reflect the calculations of the a priori stage in the weight variable W and those of the a posteriori stage in the variable vector ΔW, so that the value of the weight vector after the introduction of the new item will be $W + \Delta W$. The minimization of $E_{1..N}(W)$ is approximately equivalent to:

$$\min_{W, \Delta W} \left[E_{1 \ldots N-1}(W) + \Delta E_{1 \ldots N-1}(\Delta W; W)\right]$$
$$\textit{subject to } E_N(W + \Delta W) = 0.$$

For clarity of explanation, we make all the derivations assuming a perfect encoding of the new item ($E_N(W + \Delta W) = 0$). However, the N-th item can also be introduced partially within this approach. We need to express $\Delta E(\Delta W + W)$ in a manageable way through a truncated Taylor series expansion. A linear model would be too rude (and unfeasible [3]). Since the computation of the Hessian requires very costly computations, we include only the diagonal terms. Thus, the most faithful problem formulation we can reasonably aspire to deal with is:

$$\min_{W, \Delta W} \left[E(W) + \sum_i \frac{\partial E}{\partial W_i} \Delta W_i + \frac{1}{2} \sum_i \frac{\partial^2 E}{\partial W_i^2} \Delta W_i^2\right]$$
$$\textit{subject to } E_N(W + \Delta W) = 0.$$

The problem is how to carry out this constrained minimization on W (interference prevention) and on ΔW (retroactive interference minimization).

2 Minimizing retroactive interference

2.1 Problem formulation

Here we consider W as a constant. Note that, by introducing the first $N-1$ items, we should have minimized E and, therefore, also the absolute value of its first derivatives. Thus we assume that the linear terms are zero (but see below), and our definitive formulation for the retroactive interference subproblem is:

$$\min_{\Delta W} \sum_i c_i \Delta W_i^2 \tag{1}$$
$$\textit{subject to } E_N(\Delta W) = 0.$$

This formulation, with different c_i assigments, was used in previous works by the authors [3] and others [2]. Among them, $c_i = 1$ is interesting for its cheap computational cost, and because it is intuitively appealing, since (1) calculates in this case the nearest solution for the new item in parameter space. One can wonder what the best c_i assignment is, outside the minimum of the old items. The answer is shown to be $c_i = \frac{\partial^2 E}{\partial W_i^2}$ [4], even in this general case.

2.2 Minimization method

The question remains of how to solve (1) efficiently. A usual way to tackle a constrained optimization problem is to linearly combine the cost function with the deviation of the constraint from zero, and then minimize this new error function. An appropriate scheduling biasing progressively the tradeoff between the cost function and the constraint satisfaction towards the latter is needed. More sophisticated algorithms from the theory of constrained optimization can also be applied, but they can be complex and computationally expensive.

In [3], an algorithm to solve (1) that exploits the structure of multilayer neural networks is developed. The drawbacks of solving a constrained minimization problem are here avoided through the transformation of retroactive interference into an unconstrained minimization problem. This transformation has also other advantages, like a number of variables much lower than in the original problem, and an always perfect encoding of the new item.

2.3 Experimentation

The aforementioned algorithm, called LMD, was applied in an experimental comparison of options for assigning c_i. One result is that indeed $c_i = \frac{\partial^2 E}{\partial W_i^2}$ is the best option in general, even outside the minimum. Another conclusion emerged: the difference between using $c_i = 1$ and $c_i = \frac{\partial^2 E}{\partial W_i^2}$ decreases with the number of items stored in the network. As the number of stored items increases, all weights tend to have similar, high-magnitude second derivatives and, thus, using either of the assignments tends to yield the same results. Experiments with backpropagation showed that, when the learning rate tends to zero, it approximates LMD with $c_i = 1$ and, thus being much more inefficient than LMD to obtain the same results.

2.4 Relation between retroactive interference minimization and pruning.

To palliate retrograde interference, one of the more important issues is the determination of the less significant parameters for the encoding of a number of stored memories. These are the parameters that should support most of the necessary changes to introduce new information. Instead, pruning detects the less profited parameters to eliminate them. This antithetic relation suggests that advances in pruning techniques can be incorporated into retroactive interference minimization algorithms.

3 Interference prevention

3.1 Problem formulation

Now, we have to minimize (1) in W, taking ΔW as an unknown constant. Before knowing the new item, we cannot assume any particular value for ΔW and, therefore, the ideal solution is that which, in average, best solves the problem, taking into consideration the distribution of ΔW:

$$\min_W \left[E(W) + \left\langle \frac{1}{2} \sum_i \Delta W_i^2 \frac{\partial^2 E}{\partial W_i^2} + \sum_i \Delta W_i \frac{\partial E}{\partial W_i} \right\rangle \right].$$

where $< \cdot >$ denotes expectation. Using results from [6], we get:

$$\min_W \left[E(W) + \frac{\sigma^2}{2} \sum_i \frac{\partial^2 E}{\partial W_i^2} \right], \tag{2}$$

where σ^2 is the variance of the ΔW distribution. Reassuringly, this expression can also be derived without supposing a particular shape for $\Delta E_{1 \ldots N-1}$ $(\Delta W; W)$, by formulating interference prevention as the search for a W such that, after being modified by the introduction of the new item, it would still be able to reproduce old ones. This can be expressed as minimizing $E(W)$ stably with respect to the random perturbations produced by the new unknown items:

$$\min_W \int E(W + \Delta W)\ \mathcal{P}(\Delta W)\ d\Delta W. \tag{3}$$

Using results in [6] again, this expresion can be made exactly equivalent to (2).

σ^2 could be approximately deduced from the expected error of future items. However, there is another important issue about the selection of σ^2 to be dealt with: the way in which the items $1 \ldots N-1$ are encoded determines the answer of the network to all possible inputs, i.e., generalization. This could be priorized over interference prevention in tuning σ^2. In conclusion, either by error or intentionally, the parameter σ^2 used in (2) could be significantly different from real weight variances. A detailed mathematical analysis [1] concludes that, if σ^2

is smaller than the real variance, the minimization (2) is always beneficial. The opposite case is also safe if the remaining error $E(W)$ in the minimum of (2) is not much higher than in the minimum of $E(W)$. $E(W)$ in the minimum of (2) has a sigmoidal shape when considered as a function of σ^2 and, therefore, in practice σ^2 can be increased until the error begins to grow quickly.

3.2 Minimization method

It is possible to minimize (2), or equivalently (3), by adding noise to the weights while minimizing $E(W)$, so that a sample of the gradient distribution of $\frac{\partial E}{\partial W_i}(W+\Delta W)$ is calculated in each iteration. However, it is extremely inefficient to sample the high-dimensional parameter space of a neural network, in order to obtain the averages in the optimization steps. In [6], a method based on the gradient of (2), especially adapted for feedforward networks, is developed. It has the advantage of being deterministic and much more stable. In addition, it is easily computable with an algorithm of the same order as the backpropagation of the gradient of $E(W)$. An important property of this algorithm is that it produces results that tend to be independent of the number of hidden units used [5].

3.3 Relation between interference prevention and generalization

We have assumed that the variance of ΔW, which is directly related to the expected error for the new item, does not change while performing the minimization (3), i.e., it is independent of W. But the error in the new items is also controlled by the selection of W. Thus, there exists an alternative way to prevent interference, namely improving generalization. The minimization (3) serves also this purpose, since the term $\frac{\sigma^2}{2}\sum_i \Delta W_i^2 \frac{\partial^2 E}{\partial W_i^2}$ is a regularizer that constrains the metwork function to be simple [5, 6].

3.4 Experimentation

The experimental results in [4] show that our approach to interference prevention alleviates interference while at the same time improving generalization. This is in contrast with other strategies for interference avoidance based on local representations (e.g., saturating the hidden units). Nevertheless, we must point out the limitations in the application of this approach: the algorithm minimizing (2) makes the network output insensitive to changes in the weights for the stored items, but this insensitiveness is transmitted or generalized to the rest of the input space. Because of this, it is also necessary to modify more the weights to introduce the new items, and the potential benefits of the strategy get limited.

4 Conclusions

The amount of information available clearly determines the nature of the interference problem. When the new pattern is known, it can be used as a constraint in the minimization of the damage inflicted by its introduction. When it is still not available, the problem is the minimization of the expectaction of the error increment that the possible new patterns can produce. For both problems we developed algorithms very well suited to feedforward networks. The experimental results were as good as could be expected, given the limitations on the solution derived theoretically.

References

1. Ruiz de Angulo V., 1996: Interferencia catastrófica en redes neuronales: soluciones y relación con otros problemas del conexionismo, Ph. D. Thesis, Universidad del Pais Vasco.
2. D.C. Park, M.A. El-Sharkawi, and R.J. Marks II, 1991: An adaptively trained neural network, *IEEE Transactions on Neural Networks* **2(3)**: 334-345.
3. Ruiz de Angulo V. and Torras C., 1995: On-line learning with minimal degradation in feedforward networks, *IEEE Transactions on Neural Networks* **6(3)**: 657-668.
4. Ruiz de Angulo V. and Torras C., 2000: A framework to deal with interference in connectionist systems. *AI Communications* **13(5)**: 259-274.
5. Ruiz de Angulo V. and Torras C., 2001: Architecture-independent approximation of functions, *Neural Computation* **13(4)**: 1119-1133.
6. Ruiz de Angulo V. and Torras C., 2002: A deterministic algorithm that emulates learning with random weights. *Neurocomputing*, In Press.
7. French R.M., 1994: Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference, in: *Proc. of the 16th Annual Conf. of the Cognitive Science Society*, Erlbaum, Hillsdale: 335-340.
8. Hetherington P.A. and Seidenberg M.S., 1989: Is there catastrophic interference in connectionist networks?, in: *Proc. of the Eleventh Annual Conf. of the Cognitive Science Society*, Erlbaum, Hillsdale, NJ: 26-33.
9. Robins A., Catastrophic forgetting, 1995: rehearsal and pseudorehersal, *Connection Science* **7(2)**: 123-146.