Selfless Sequential Learning

Rahaf Aljundi
KU Leuven

Marcus Rohrbach
Facebook AI Research

Tinne Tuytelaars
KU Leuven

Abstract

Sequential learning studies the problem of learning tasks in a sequence with access restricted to only the data of the current task. In this paper we look at a scenario with a fixed model capacity, and postulate that the learning process should not be selfish, i.e. it should account for future tasks to be added and thus leave enough capacity for them. To achieve *Selfless Sequential Learning* we study different regularization strategies and activation functions that could lead to less interference between the different tasks. We find that learning a sparse representation is more beneficial for sequential learning than encouraging parameter sparsity. In particular, we propose a novel regularizer, that encourages representation sparsity by means of local neural inhibition. It results in few active neurons which in turn leaves more free neurons to be utilized by upcoming tasks. We combine our novel regularizer with state-of-the-art sequential learning methods that penalize changes to important previously learned parts of the network and show consistent performance improvement.

1 Introduction

Sequential learning, also referred to as continual, incremental, or lifelong learning, studies the problem of learning a sequence of tasks, one at a time, without access to the training data of previous tasks, yet avoiding catastrophic interference with the previously learned tasks [4]. This scenario not only has many practical benefits, but also resembles more closely how the mammalian brain learns tasks over time. However, while the mammalian brain is composed of billions of neurons, at any given time, information is represented by only a few active neurons resulting in a sparsity of 90-95% [9]. In neural biology, *lateral inhibition* describes the ability of an activated neuron to reduce the activity of its weaker neighbors. This creates a powerful decorrelated and compact representation with minimum interference between different input patterns in the brain [12]. On the contrary, artificial neural networks typically learn dense representations that are highly entangled and sensitive to changes in the input [2].

Based on these observations, we advocate for the use of sparse models, leading to **Selfless Sequential Learning**, i.e. leaving capacity for future tasks. In this paper we postulate, and confirm experimentally, that a sparse and decorrelated representation is preferable over parameter sparsity in a sequential learning scenario. There are two arguments for this: first, a sparse representation is less sensitive to new and different patterns (such as data from new tasks) and second, the training procedure of the new tasks can use the free neurons leading to less interference with the previous tasks, hence reducing forgetting. In contrast, when the effective parameters are spread among different neurons, changing the ineffective ones would change the function of their corresponding neurons and hence interfere with previous tasks.

Given that we need sparse activations, we propose a new regularizer that exhibits a behavior similar to the lateral inhibition in biological neurons. The main idea is to penalize neurons which are active at the same time, leading to more sparsity. However, complex tasks may actually require multiple active neurons in a layer at the same time to learn a strong representation. Therefore, our regularizer, **Sparse coding through Local Neural Inhibition** (`SLNI`), only penalizes neurons locally. Furthermore, we don't want inhibition to affect previously learned tasks, even if later tasks use neurons from earlier

Preprint. Work in progress.

tasks. An important component of SLNI is thus to discount inhibition involving neurons which have high *neuron importance* – a new concept that we introduce in analogy to parameter importance. When combined with a state-of-the-art important parameters preservation method [1], our regularizer consistently improves the lifelong learning performance.

2 Selfless Sequential Learning

One of the main challenges in single model sequential learning is to avoid catastrophic forgetting of previous tasks as a result of learning new tasks. In order to prevent catastrophic forgetting, importance weight based methods such as EWC [7] or MAS [1] introduce an importance weight Ω_k for each parameter θ_k in the network. While these methods differ in how to estimate the important parameters, all of them penalize changes to important parameters when learning a new task T_n using L_2 penalty: $T_n: \quad \min_\theta \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}(y_m, f(x_m,\theta^n)) + \lambda_\Omega \sum_k \Omega_k(\theta_k^n - \theta_k^{n-1})^2$ where θ_k^{n-1} are the optimal parameters that are learned so far, i.e. before the current task. x_m is an input and y_m its desired output of the network. λ_Ω is a trade-off parameter between the new task objective and the changes on the important parameters, i.e. the amount of forgetting.

In this work we introduce an additional regularizer R_{SSL} which encourages sparsity in the activations $H_l = \{h_i^m\}$ for each layer l.

$$T_n: \quad \min_\theta \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}(y_m, f(x_m,\theta^n)) + \lambda_\Omega \sum_k \Omega_k(\theta_k^n - \theta_k^{n-1})^2 + \lambda_{\text{SSL}}\sum_l R_{\text{SSL}}(H_l) \tag{1}$$

λ_{SSL} and λ_Ω are trade-off parameters that control the contribution of each term. When training the first task ($n = 1$), $\Omega_k = 0$. For sparsity in parameters we instead regularize the parameters $R_{\text{SLL}}(\theta_k)$.

To promote sparsity in the representation, we can identify, in the literature [5], the use of L_1 norm on the activations (since minimizing the L_0 norm is an NP hard problem). However, L_1 norm imposes an equal penalty on all the active neurons leading to small activation magnitude across the network. On the other hand, learning a decorrelated representation has been explored before with the goal of reducing overfitting. This is usually done by minimizing the Frobenius norm of the covariance matrix corrected by the diagonal as in [3]. Such a penalty results in a decorrelated representation but with activations that are mostly close to a non zero mean value. In the case of ReLU activation function, merging the objectives of sparse and decorrelated representation can be achieved by the following objective:

$$R(H_l) = \frac{1}{M}\sum_{i,j}\sum_m h_i^m h_j^m, \quad i \neq j \tag{2}$$

$$\frac{\partial R(H_l)}{\partial h_i^m} = \frac{1}{M}\sum_{j\neq i} h_j^m \tag{3}$$

$H_l = \{h_i^m\}$ are the activations for a set of inputs $X = \{x_m\}$ where $i, j \in 1, .., N$ with N the number of neurons in the hidden layer. Eq. 3 evaluates the derivative of the presented regularizer w.r.t. the activation, each active neuron receives a penalty from every other active neuron that corresponds to that other neuron's activation magnitude. In other words, if a neuron fires, with a high activation value, for a given example, it will suppress firing of other neurons for that same example. Hence, this results in a decorrelated sparse representation.

The loss imposed by this objective will only be zero when there is at most one active neuron per example. This seems to be too harsh for complex tasks that need a richer representation. Thus, we suggest to relax the objective by imposing a spatial weighting to the correlation penalty. In other words, an active neuron penalizes mostly its close neighbours and this effect vanishes for neurons further away. Instead of uniformly penalizing all the correlated neurons, we weight the correlation penalty between two neurons with locations i and j using a Gaussian weighting. This gives

$$R_{\widehat{\text{SLNI}}}(H_l) = \frac{1}{M}\sum_{i,j} e^{-\frac{(i-j)^2}{2\sigma^2}}\sum_m h_i^m h_j^m, \quad i \neq j \tag{4}$$

As such, each active neuron inhibits its neighbours, introducing a locality in the network inspired by biological neurons. σ^2 is a hyper parameter representing the scale at which neurons can affect each other. Our regularizer inhibits locally the active neurons leading to a sparse coding, hence the name Sparse coding through Local Neural Inhibition (SLNI).

Neuron importance for discounting inhibition. In a learning sequence with tasks with completely different input patterns, the active neurons of the previous tasks will not be activated given the new

tasks input patterns. However, when the new tasks are of similar or shared patterns, neurons used for previous tasks will be active. In that case, our penalty would discourage other neurons from being active and encourage the new task to adapt the already active neurons instead. This would interfere with the previous tasks and could increase forgetting. To avoid such interference, we add a weight factor taking into account the importance of the neurons with respect to the previous tasks. To estimate the importance of the neurons, we use as a measure the sensitivity of the learned objective to their changes. This is approximated by the gradients of the loss w.r.t. the neurons outputs evaluated at each data point. To get an importance value, we then accumulate the magnitude of the gradients over the given data points obtaining importance weight α_i for neuron n_i:

$$\alpha_i = \frac{1}{M}\sum_{m=1}^{M} \mid\mid g_i(x_m) \mid\mid \quad , \quad g_i(x_m) = \frac{\partial(\mathcal{L}(y_m, f(x_m, \theta^t)))}{\partial n_i^m} \tag{5}$$

where n_i^m is the output of neuron n_i for a given input example x_m, and θ^t are the parameters after learning task t. Then, we can weight our regularizer as follows:

$$R_{\texttt{SLNI}}(H_l) = \frac{1}{M}\sum_{i,j} e^{-(\alpha_i+\alpha_j)} e^{-\frac{(i-j)^2}{2\sigma^2}} \sum_m h_i^m h_j^m, \quad i \neq j \tag{6}$$

which can be read as: if an important neuron for a previous task is active given an input pattern from the current task, it will not suppress the other neurons from being active neither be affected by other active neurons. The final objective for training is given in Eq. 1, setting $R_{SSL} := R_{\text{SLNI}}$.

3 Experiments

3.1 An in-depth comparison of regularizers and activation functions in Sequential Learning

We study possible regularization techniques that could lead to less interference between the different tasks either by enforcing sparsity or decorrelation. Additionally, we examine the use of activation functions that encourage competition between active neurons. We use the MNIST dataset as a first task in a sequence of 5 tasks, where we randomly permute all the input pixels for tasks 2 to 5. The goal is to classify MNIST digits from all the different permutations. We study the following methods:

Representation Based methods:

- `L1-Rep`: to promote representational sparsity, an L_1 penalty on the activations is used.
- `Decov` [3] aims at reducing overfitting by decorrelating neuron activations.

Activation functions:

- `Maxout` [6]: only the neuron with maximum activation in each block is forwarded to the next layer.
- `LWTA` [11]: similar to the Maxout but the non-maximum activations maintain their connections.
- `ReLU` [5]: used as a baseline here and indicated in later experiments as `No-Reg`.

Parameters based regularizers:

- `OrthReg` [10]: decorrelates the weight vectors by minimizing the cosine of the angle between them.
- `L2-WD` [8]: controls the complexity of the learned function by minimizing the weights magnitude.
- `L1-Param`: L_1 penalty on the parameters to encourage a solution with sparse parameters.

Results: Figure 1 presents, for the studied methods, the test accuracy on each task at the end of the sequence. Clearly, in all the different tasks, the representational regularizers show a superior performance over the other studied techniques. More specifically, the `L1-Rep` improvement over `L1-Param` indicates the advantage of encouraging representational sparsity over parameters sparsity. For the regularizers applied to the parameters, `L2-WD` and `L1-Param` do not exhibit a clear trend and do not systematically show an improvement over the use of the different activation functions only. While `OrthReg` shows a consistent good performance, it is lower than what can be achieved by the representational regularizers. Regarding the activation functions, `Maxout` and `LWTA` achieve a slightly higher performance than `ReLU` but this remains moderate as they need a fixed window size and special architecture design. Our regularizer `SLNI` achieves among the top accuracies in all the tasks indicating its ability to achieve less interference between tasks and to gain more flexibility.

3.2 10 Task sequences on CIFAR-100 and Tiny Imagenet

We study the case of learning different categories of one dataset. For this we split the CIFAR-100 and the Tiny ImageNet dataset into ten tasks each, resulting in 10 and 20 categories per task, respectively. We compare the top competing methods from Sec. 3.1, `L1-Rep`, `DeCov` and our `SLNI`, and `No-Reg` as a baseline (`ReLU` in previous experiment). Figure 2 shows the performance on each of the ten tasks at the end of each sequence. For both datasets, `SLNI` performs overall best. `L1-Rep` and `DeCov` again

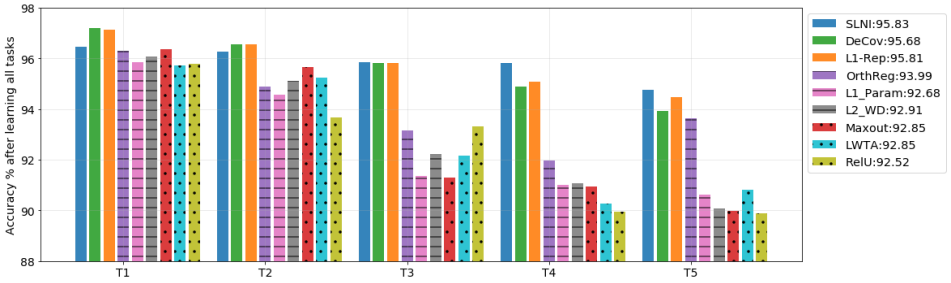

Figure 1: Comparison of different regularization techniques on 5 permuted MNIST sequence of tasks. Representation based regularizers are solid bars, bars with lines represent parameters regularizers, dotted bars represent activation functions. Average test accuracy over all tasks is given in the legend.

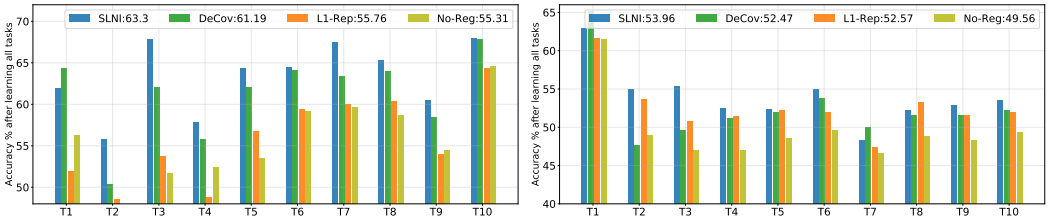

Figure 2: Comparison of different regularization techniques on a sequence of ten tasks from (a) Cifar split and (b) Tiny ImageNet split. The legend shows average test accuracy over all tasks at the end of the sequence.

improve over the non regularized case `No-Reg`. These results confirm our proposal on the importance of sparsity and decorrelation in sequential learning. For more details, please refer to the arxiv version of this paper.

4 Conclusion

In this paper we study the problem of sequential learning using a fixed model capacity. We argue that when learning a task one should be selfless and account for upcoming tasks. We show that in a sequential learning scenario, sparsity should be enforced in the learned representation rather than in the parameters. Our proposed regularizer based on neural inhibition systematically results in performance improvement across the different sets of tasks studied in this paper.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

[2] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[3] M Cogswell, F Ahmed, R Girshick, L Zitnick, and D Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv:1511.06068*, 2015.

[4] Robert M French. Catastrophic forgetting in connectionist networks. *TCS*, 1999.

[5] X Glorot, A Bordes, and Y Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011.

[6] I J Goodfellow, D Warde-Farley, M Mirza, A Courville, and Y Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[7] J Kirkpatrick, R Pascanu, N Rabinowitz, J Veness, G Desjardins, A A Rusu, K Milan, J Quan, T Ramalho, A Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *arXiv:1612.00796*, 2016.

[8] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

[9] Peter Lennie. The cost of cortical computation. *Current biology*, 13(6):493–497, 2003.

[10] P Rodríguez, J Gonzalez, G Cucurull, J M Gonfaus, and X Roca. Regularizing cnns with locally constrained decorrelations. *arXiv preprint arXiv:1611.01967*, 2016.

[11] R K Srivastava, J Masci, S Kazerounian, Fa Gomez, and J Schmidhuber. Compete to compute. In *NIPS*. 2013.

[12] Yuguo Yu, Michele Migliore, Michael L Hines, and Gordon M Shepherd. Sparse coding and lateral inhibition arising from balanced and unbalanced dendrodendritic excitation and inhibition. *Journal of Neuroscience*, 34(41):13701–13713, 2014.