Summary: Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science

Mocanu D. C., Mocanu E., Stone P., Phuong H. N., Gibescu M., Liotta A.

Notes by Mauro Comi, maurocomi92@gmail.com

February 15, 2021

1 Important resources

Link to the paper: https://www.nature.com/articles/s41467-018-04316-3
Link to the implementation: https://github.com/dcmocanu/sparse-evolutionary-artificial-neural-networks

2 Domain-specific prerequisites

- Artificial Neural Networks
- Backpropagation
- Intuition on fully connected and sparse networks

3 Paper Notes

What

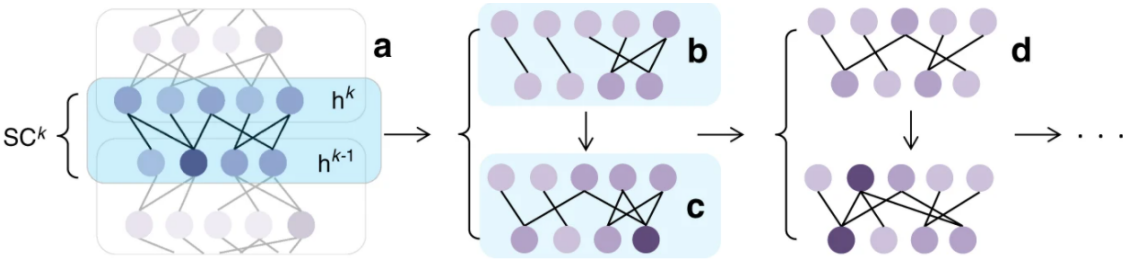
In this paper, the authors propose a novel methodology (sparse evolutionary traning, SET) to design and train artificial neural networks starting from a sparse topology. The algorithm is inspired by biological evolutionary processes. The training phase is based on an evolutionary algorithm that removes a fraction of weights (the ones that are closest to zero) and adds new random weights at each training epoch. This approach aims to combine traditional neuroevolution with deep learning to optimize the Artificial Neural Network (ANN) connections while keeping the search space relatively low. The paper shows the result of this approach on three ANN types: RBM, MLP, CNN.

Why

Biological neural networks have been demonstrated to have a sparse topology, rather than a dense one. This property is instrumental to learning efficiency, as well as scalability.

ANNs have not evolved to mimic this feature. Previous work pursues sparse topological connectivity only as an aftermath of the training phase: this yields benefits during the inference phase, but not during the training phase. NEAT (NeuroEvolution of Augmenting Topologies) optimizes both the weights and the network topology, thus yielding benefits during the training phase as well. However, the search space is very large and makes this approach not suitable for large applications.

How



An illustration of the SET procedure. For each sparse connected layer, SC^k (**a**), of an ANN at the end of a training epoch a fraction of the weights, the ones closest to zero, are removed (**b**). Then, new weighs are added randomly in the same amount as the ones previously removed (**c**). Further on, a new training epoch is performed (**d**), and the procedure to remove and add weights is repeated. The process continues for a finite number of training epochs, as usual in the ANNs training

Figure 1: SET training procedure. Image taken from the original paper (link in Section: Important resources)

To understand how connections are created, let's consider two consecutive layers: layer k and layer $k-1$. Every sparse connected layer has n neurons. Any neuron in layer k is connected to an arbitrary number of neurons belonging to the previous layer $k-1$. Given two neurons n_i and n_j belonging respectively to layer k and $k-1$, the probability p of their connection is calculated as follows:

$$p = \frac{\epsilon(n^k + n^{k-1})}{n^k n^{k-1}}$$

where ϵ is a parameter controlling the sparsity of connectivity. However, this randomly generated topology might not be well suited to the specific learning task at hand. For this reason, after every training epoch a fraction ζ of the smallest positive weights and of the largest negative weights of each layer is removed. Next, an amount of new connections - equal to the amount of weights previously removed - is randomly added. This way the number of connections remains constant during the training process. After the training ends, the topology remains fixed without any further weights addition or removal.

Results

This approach is applied to RBMs, CNNs and MLPs. In all these three cases, the performance of the sparse models is equal or better than their fully-connected counterparts, while only using 1% of the weights. Moreover, the resulting network appear more stable than the fully-connected ones, and less prone to overfitting. SET is capable of quadrati-

cally reducing the number of parameters of dense neural network layers at no decrease in accuracy.