Learning sparse deep neural networks using efficient structured projections on convex constraints for green AI

Michel Barlaud 1, Antonin Chambolle 2 and Frédéric Guyard 3
(1) Univ. Côte d’Azur & CNRS, Sophia Antipolis, (2) CEREMADE, CNRS & Paris-Dauphine PSL (3) Orange Lab Sophia Antipolis

Abstract
Deep neural networks (DNN) have been applied recently to different domains and perform better than classical state-of-the-art methods. However the high level of performances of DNNs is most often obtained with networks containing millions of parameters and for which training requires substantial computational power. To deal with this computational issue, proximal regularization methods have been proposed in the literature but they are time consuming.

In this paper, we propose instead a constrained approach. We studied algorithms for different constraints: the classical ℓ1 unstructured constraint and structured constraints such as the ℓ1,l0 constraint (Group LASSO). We propose a new ℓ1,l1 structured constraint for which we provide a new projection algorithm. Finally, we used the recent “Lottery optimizer” replacing the threshold by our ℓ1,l1 projection. We demonstrate the effectiveness of this method with three popular datasets (MNIST, Fashion MNIST and CIFAR).

Results on MNIST with two Linear fully connected Networks
We used first a linear fully connected network (LFC4) with an input layer of d neurons, 4 hidden layers followed by a RELU activation function and a latent layer of dimension k. We used a linear fully connected network (LFC4) with an input layer of d neurons, 4 hidden layers followed by a RELU activation function and a latent layer of dimension k.

Results on Cifar10
The Cifar-10 data set is composed of 60,000 32x32 color images, 6,000 images per class, for classification in 10 classes. The training set is made up of 50,000 images, while the remaining 10,000 are used for the testing set. We use SimplerNet, a network composed of 13 blocks.

References

Classical Group LASSO structured constraint
Group LASSO was first introduced in [3]. The main idea is to enforce parameters of different classes to share common features. Group sparsity reduces so complexity by eliminating entire features. It consists in using the ℓ1 norm for the constraint on W, which is defined as follows. The rowwise ℓ2,1 norm of a d × k matrix W (where rows are denoted w_i, i = 1, ..., d) is

\[ ||W||_{2,1} = \sum_{d=1}^{d} ||w_i||_2. \]

A new ℓ1,l1 structured constraint
Unfortunately, the Group LASSO structured constraint algorithm [1] does not provide efficient sparsity. Thus we propose the following algorithm:

Algorithm 1 Projection on the ℓ1,l1 norm (proj_{1,l1}) is the projection on the ℓ1-ball of radius γ.

Input: \( W, t, \gamma \)
\[ L = \text{proj}_{1,l1}(W, t, \gamma) \]
for \( i = 1, \ldots, d \) do
\[ w_i = \text{proj}_{1,l1}(w_i, t, \gamma) \]
end
Output: \( W' \)

Analysis of the structured-ℓ1,l1 projection: While the algorithm above does not strictly speaking define a projection, its solution corresponds to solving a sort of bi-level projection and is easily shown to be obtained as a limit, as \( \epsilon \) goes to zero, of the minimizers of the convex problem:

\[ \min_{\sum_{d=1}^{d} \sum_{i=1}^{d} (|v_i| - t_i)^2 + \epsilon \sum_{i=1}^{d} \sum_{j=1}^{d} |v_{ij} - w_{ij}|^2} \]

In other words, \( (t, w) \in \mathbb{R}^d \times \mathbb{R}^{d \times d} \) can be seen as the projection of \( (\sum_{i=1}^{d} |v_i|, v) \) onto the convex set \( (t, w) \in [0, \infty)^d \times \mathbb{R}^{d \times d}, |v_{ij}| \leq t_i, \forall i \) in a degenerate (lexicographic) distance which infinitely favors the component over the variable.

Lottery optimizer
Following the work of Frankle and Carbin, who proposed an algorithm to find sparse sub-networks. We replaced their thresholding by our ℓ1,l1 projection and devised the following algorithm:

Algorithm 2 Projection on the ℓ1,l1 norm: Here \( \text{proj}_{1,l1}(W, \gamma) \) is the projection on the ℓ1,l1-ball of radius \( \gamma, \nabla L(W, M_d) \) is the masked gradient with binary mask \( M_d \), \( f \) is the ADAM optimizer and \( \gamma \) is the learning rate.

Input: \( W, \gamma > 0 \)
for \( i = 1, \ldots, N/\text{epochs} \) do
\[ V = f(W, \gamma, \nabla L(W)) \]
end for
Output: \( W' \)

Results and comparison of methods:
Results on MNIST with a convolutional Network
We selected the popular MNIST dataset containing 38 × 32 grey-scale images of handwritten digits of 10 classes (from 0 to 9). This dataset consists of a training set of 60,000 instances and a test set of 10,000 instances. We consider a neural network with two convolutional layers and two linear layers denoted as Net4.