Learning the Structure of Deep Architectures Using $\ell_1$ Regularization

Praveen Kulkarni$^1$
Praveen.Kulkarni@technicolor.com
Joaquin Zepeda$^1$
Joaquin.Zepeda@technicolor.com
Frederic Jurie$^2$
frederic.jurie@unicaen.fr
Patrick Pérez$^1$
Patrick.Perez@technicolor.com
Louis Chevallier$^3$
Louis.Chevallier@technicolor.com

$^1$975 avenue des Champs Blancs,
CS 17616, 35576 Cesson Sévigné, France
http://www.technicolor.com

$^2$University of Caen Basse-Normandie,
CNRS UMR 6072, ENSICAEN, France http://www.unicaen.fr

Our proposed approach is illustrated in Fig. 1. The architecture we consider consists of a sequence of fully-connected layers, with a diagonal matrix between them. We present a method that automatically selects the size of the weight matrices inside fully-connected layers indexed by $j = 1,\ldots,J$. Our approach relies on a regularization penalty term consisting of the $\ell_1$ norm of the diagonal entries of diagonal matrices inserted between the fully-connected layers. Using such a penalty term forces the diagonal matrices to be sparse, accordingly selecting the effective number of rows and columns in the weights matrices of adjacent layers. We present a simple algorithm to solve the proposed formulation and demonstrate it experimentally on a standard image classification benchmark.

Figure 1: Proposed deep processing pipeline. Given an image representation, e.g., the output of the convolutional part of a pre-trained state-of-art DNN, $J$ fully connected layers, each involving a diagonal matrix that controls its effective dimensions, are jointly learned with final linear SVM classifiers.

In the above expression, we have used (i) $l(x)$ to denote the hinge loss, given by $\max(0,1-x)$; (ii) $||D||_2$ to denote the trace norm, given by $\sum_{i,j} |D_{ij}|$ for the case of diagonal $D$; and (iii) $||M||_F^2$ to denote the squared Frobenius norm $\sum_{i,j} M_{ij}^2$.

To illustrate the motivation behind this learning objective, we note first that the terms inside the summation over $k$ in (3) are recognizable as an SVM objective for class $k$, where the scalar $C$ is the SVM regularization parameter. The feature vectors used within this SVM objective are given by $f_j \circ \ldots \circ f_1(x_i)$, which depends on $(\{M^j, b^j, D^j\})_{j=1}^J$. Hence we are learning the classifiers jointly with the feature extractor used to represent the input images.

The two regularization terms comprised of summations over $j$ in (3) have two important purposes. First is to keep the SVM terms from decreasing indefinitely. A second important purpose is to automatically select the shapes of the weights matrices $M^j$ and $b^j$ and $\ell_1$ norms applied to diagonal matrices such as $D^j$ are sparsity inducing norms.

In order to minimize (3), we will employ a block-coordinate SGD approach. We evaluate our method on Pascal VOC 2007 dataset and compare it against various state-of-the-art algorithms. Our learning algorithm is governed by three important terms: the penalty weights $\mu$ and $\delta$ and the number of training epochs $T$. The number of training epochs is determined using the validation set. Further, we evaluate the performance of our method as a function of the number of layers $J$ in the architecture.

In Fig. 2, we plot both the sparsity for all layers and the corresponding test and validation mAPs when varying the penalty weight $\delta$. Note that increasing $\delta$ drastically increases the number of zero diagonal entries in the architecture while only slightly affecting the classification performance.

