FeTa: Fast and Efficient Pruning of Deep Neural Networks.

Konstantinos Pitas  
Signal Processing Laboratory 2  
EPFL  
Switzerland  
Email: konstantinos.pitas@epfl.ch

Mike Davies  
IDCOM  
University of Edinburgh  
UK  
Email: Mike.Davies@ed.ac.uk

Pierre Vandergheynst  
Signal Processing Laboratory 2  
EPFL  
Switzerland  
Email: pierre.vandergheynst@epfl.ch

Abstract—Recent deep neural network architectures have a very large number of parameters and correspondingly large memory footprint. A simple method for reducing the number of parameters is hard thresholding of the deep neural network layer weights. Hard thresholding however in a large drop in classification accuracy, necessitating a fine-tuning of the pruned architecture. A number of recent works have focused on replacing hard thresholding with data-driven optimisation procedures, resulting in a smaller degradation in accuracy. We reformulate an existing layerwise pruning algorithm as a difference of convex functions optimisation problem. Given \( d_1 \) and \( d_2 \) the layer input and output dimensions the original formulation has space complexity of \( \mathcal{O}(d_1^2 d_2) \), by posing the problem as a difference of convex functions we provide a scalable solver with space complexity of \( \mathcal{O}(d_1 d_2) \) where typically \( d_2 \ll d_1 \). We also compare with other recently proposed algorithms and find that our algorithm is often orders of magnitude faster while simultaneously achieving better solutions in terms of the deep neural network accuracy.

I. INTRODUCTION

There has been a huge number of recent works aiming at DNN compression. Most methods do this by including a regularisation term during training. [1][2][3][4]. We are however interested in methods that prune fully connected layers after training as a data driven substitute to hard thresholding [5][6][7], as pretrained but uncompressed DNN models are now ubiquitous.

A. Net-Trim

We first make a detailed description of the Net-Trim algorithm [5]. Given a fully connected DNN weight matrix \( W \in \mathbb{R}^{d_1 \times d_2} \) and a set of input and output signals \( A \in \mathbb{R}^{d_1 \times m}, B \in \mathbb{R}^{d_2 \times m} \), and \( p(x) = \max(x, 0) \) Net-Trim aims to find a new weight matrix \( U \in \mathbb{R}^{d_1 \times d_2} \)

\[
\min_{U} \|U\|_1 \quad \text{subject to} \quad \|p(U^T A) - B\|_F^2 \leq \epsilon, \tag{1}
\]

where \( \epsilon \in \mathbb{R}^+ \) is a tolerance parameter. Net-Trim finds a sparse matrix \( U \) such that the output of the pruned nonlinear layer \( p(U^T A) \) stays close in terms of the Frobenius norm to the original unpruned output \( B \). The optimisation above is non-convex and the authors of Net-Trim propose to solve the following convex proxy. For \( U \in \mathbb{R}^{d_1 \times d_2}, A \in \mathbb{R}^{d_1 \times m}, B \in \mathbb{R}^{d_2 \times m}, \epsilon \in \mathbb{R}^+ \), and \( \Omega \subseteq \{1, \ldots, d_2\} \times \{1, \ldots, m\} \), Net-Trim minimizes

\[
\min_{U} \|U\|_1 \quad \text{subject to} \quad \frac{1}{\|U^T A - B\|_F^2} \leq \epsilon, \tag{2}
\]

which is solved using the ADMM approach.

B. FeTa

We introduce an objective, which differs in crucial ways. For \( U \in \mathbb{R}^{d_1 \times d_2}, a_j \in \mathbb{R}^{d_1}, b_j \in \mathbb{R}^{d_2}, \lambda \in \mathbb{R}^+ \) we optimise

\[
\min_{U} \frac{1}{m} \sum_{j \in S_m} \| p(U^T a_j) - b_j \|_2^2 + \lambda \Omega(U), \tag{3}
\]

where \( \lambda \) is the sparsity parameter. The term \( \| p(U^T a_j) - b_j \|_2^2 \) ensures that the nonlinear projection remains the same for training signals. The term \( \lambda \Omega(U) \) is any convex regulariser which imposes the desired structure on the weight matrix \( U \). For \( \Omega(U) = \lambda \|U\|_1 \) we recover an alternative formulation of Equation 1.

The objective in Equation 3 is non-convex. We show that the optimisation of this objective can be cast as a difference of convex functions (DC) problem [8] which can be optimised directly with efficient solvers. We assume just one training sample \( x \in \mathbb{R}^N \), for simplicity, with latent representations \( a \in \mathbb{R}^{d_1} \) and \( b \in \mathbb{R}^{d_2} \)

\[
\| p(U^T a) - b \|_2^2 + \lambda \Omega(U) = \sum_i \left[ \| p^2 (u_i^T a) + b_i^2 \| + \lambda \Omega(U) \right] + \sum_{b_i < 0} \| -2b_i \rho(u_i^T a) \| + \sum_{b_i \geq 0} \| -2b_i \rho(u_i^T a) \|. \tag{4}
\]

Notice that after the split the first term \( (b_i < 0) \) is convex while the second \( (b_i \geq 0) \) is concave. We note that \( b_i \geq 0 \) by definition of the ReLu and set

\[
g(U; x) = \sum_i \left[ \rho^2 (u_i^T a) + b_i^2 \right], \tag{5}
\]

\[h(U; x) = \sum_{b_i > 0} \left[ 2b_i \rho(u_i^T a) \right]. \tag{6}\]

Then by summing over all the samples we get

\[f(U) = \sum_j g(U; x_j) + \lambda \Omega(U) - \sum_j h(U; x_j) \tag{7}\]

which is a difference of convex functions. We name the corresponding algorithm FeTa (Fast and Efficient Trimming Algorithm).

II. EXPERIMENTS

We set \( \lambda \Omega(U) = \lambda \|U\|_1 \) and prune the fully connected layers of a number of DNN architectures using FeTa, Net-Trim[5], LOBS[6], Corenet[7]. We show the results in Table 1. We see that FeTa achieves the best solutions often orders of magnitude faster than other approaches. We also plot a detailed comparison of the objective function for Net-Trim and FeTa in Figure 1, as well as the peak memory consumption. We use as a baseline standard SGD + Hard Thresholding applied to equation (3). We see that often FeTa converges much faster than NetTrim while requiring significantly less memory. Finally we plot results using the low-rank promoting regulariser \( \lambda \Omega(U) = \lambda \|U\|_1 \), in Figure 2 and show that our algorithm obtains better results than simply applying SVD to the layer weights.
TABLE I: Accuracy decrease and total pruning time (hours (H), minutes (M), seconds (S)) for Mnist, Fashion Mnist and Cifar-10.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dense Mnist</th>
<th>Conv Mnist</th>
<th>Dense F-Mnist</th>
<th>Conv F-Mnist</th>
<th>Dense Cifar</th>
<th>Conv Cifar</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Decr</td>
<td>Time</td>
<td>Decr</td>
<td>Time</td>
<td>Decr</td>
</tr>
<tr>
<td>Thresholding</td>
<td>-</td>
<td>15%</td>
<td>-</td>
<td>1%</td>
<td>-</td>
<td>19%</td>
</tr>
<tr>
<td>Net-Trim</td>
<td>3M</td>
<td>10%</td>
<td>16M</td>
<td>1%</td>
<td>15M</td>
<td>7%</td>
</tr>
<tr>
<td>LOBS</td>
<td>3M</td>
<td>5%</td>
<td>3M</td>
<td>1%</td>
<td>3M</td>
<td>18%</td>
</tr>
<tr>
<td>CoreNet</td>
<td>1M</td>
<td>60%</td>
<td>3M</td>
<td>29%</td>
<td>42M</td>
<td>20%</td>
</tr>
<tr>
<td>FeTa</td>
<td>3M</td>
<td>10%</td>
<td>3M</td>
<td>1%</td>
<td>3M</td>
<td>7%</td>
</tr>
</tbody>
</table>

FeTa

<table>
<thead>
<tr>
<th>Thresholding</th>
<th>Time</th>
<th>Decr</th>
<th>Time</th>
<th>Decr</th>
<th>Time</th>
<th>Decr</th>
<th>Time</th>
<th>Decr</th>
<th>Time</th>
<th>Decr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net-Trim</td>
<td>3M</td>
<td>10%</td>
<td>16M</td>
<td>1%</td>
<td>15M</td>
<td>7%</td>
<td>24M</td>
<td>1%</td>
<td>13M</td>
<td>1%</td>
</tr>
<tr>
<td>LOBS</td>
<td>3M</td>
<td>5%</td>
<td>3M</td>
<td>1%</td>
<td>3M</td>
<td>18%</td>
<td>3H</td>
<td>5%</td>
<td>1H</td>
<td>10%</td>
</tr>
<tr>
<td>CoreNet</td>
<td>1M</td>
<td>60%</td>
<td>3M</td>
<td>29%</td>
<td>42M</td>
<td>20%</td>
<td>18M</td>
<td>25%</td>
<td>1H</td>
<td>11%</td>
</tr>
<tr>
<td>FeTa</td>
<td>3M</td>
<td>10%</td>
<td>3M</td>
<td>1%</td>
<td>3M</td>
<td>7%</td>
<td>7M</td>
<td>1%</td>
<td>5M</td>
<td>24M</td>
</tr>
</tbody>
</table>

Fig. 1: (a) Memory consumption for FeTa and NetTrim on synthetic data. Data fidelity (b) $\|\rho(U^T A) - B\|^2_F$ and sparsity (c) $\%$zeros for Layer 1 of the Dense Fashion Mnist architecture.

Fig. 2: a) We plot the histogram of values of a pruned layer after optimising with SGD + $l_1$ equation (3). We see that SGD + $l_1$ regularisation does not result in sparse weights, even after several epochs of training. It is therefore necessary to apply hard thresholding to obtain a sparse matrix. b)Mnist fully connected c)Cifar-10 fully connected: Feta provides better solutions compared to SVD for low rank approximations of the weight matrix. We have defined the compression ratio $CR = \#\text{remaining parameters}/\#\text{original parameters}$. The y-axis designates the accuracy of the pruned model.

REFERENCES