Convergence of a Relaxed Variable Splitting Method for Learning Sparse Neural Networks via ℓ_1, ℓ_0, and Transformed-ℓ_1 Penalties

Thu Dinh and Jack Xin

Department of Mathematics,
University of California, Irvine, CA 92697, USA.
{t.dinh, jack.xin}@uci.edu

Abstract. Sparsification of neural networks is one of the effective complexity reduction methods to improve efficiency and generalizability. We consider the problem of learning a one hidden layer convolutional neural network with ReLU activation function via gradient descent under sparsity promoting penalties. It is known that when the input data is Gaussian distributed, no-overlap networks (without penalties) in regression problems with ground truth can be learned in polynomial time at high probability. We propose a relaxed variable splitting method integrating thresholding and gradient descent to overcome the non-smoothness in the loss function. The sparsity in network weight is realized during the optimization (training) process. We prove that under ℓ_1, ℓ_0, and transformed-ℓ_1 penalties, no-overlap networks can be learned with high probability, and the iterative weights converge to a global limit which is a transformation of the true weight under a novel thresholding operation. Numerical experiments confirm theoretical findings, and compare the accuracy and sparsity trade-off among the penalties.

Keywords: regularization, sparsification, non-convex optimization

1 Introduction

Deep neural networks (DNN) have achieved state-of-the-art performance on many machine learning tasks such as speech recognition (Hinton et al., 2012 [8]), computer vision (Krizhevsky et al., 2016 [10]), and natural language processing (Dauphin et al., 2016 [3]). Training such networks is a problem of minimizing a high-dimensional non-convex and non-smooth objective function, and is often solved by simple first-order methods such as stochastic gradient descent. Nevertheless, the success of neural network training remains to be understood from a theoretical perspective. Progress has been made in simplified model problems. Shamir (2016) showed learning a simple one-layer fully connected neural network is hard for some specific input distributions [20]. Recently, several works (Tian, 2017 [22]; Brutzkus & Globerson, 2017 [1]) focused on the geometric properties of loss functions, which is made possible by assuming that the input data distribution is Gaussian. They showed that stochastic gradient descent (SGD) with random or zero initialization is able to train a no-overlap neural network in polynomial time.

Another notable issue is that DNNs contain millions of parameters and lots of redundancies, potentially causing over-fitting and poor generalization [26] besides spending unnecessary computational resources. One way to reduce complexity is to sparsify the network weights using an empirical technique called pruning [11] so that the non-essential ones are zeroed out with minimal loss of performance [7,24,14]. Recently a surrogate ℓ_0 regularization approach based on a continuous relaxation of Bernoulli random variables in the distribution sense is introduced with encouraging results on small size image data sets [12]. This motivated our work here to study deterministic regularization of ℓ_0 via its Moreau envelope and related ℓ_1 penalties in a one hidden layer convolutional neural network model [1].
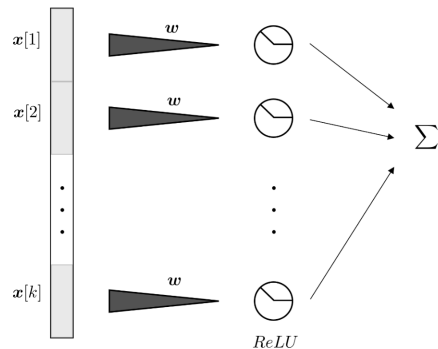

Fig. 1. The architecture of a no-overlap neural network

Our contribution: We propose a new method to sparsify DNNs called the Relaxed Variable Splitting Method (RVSM), and prove its convergence on a simple one-layer network (Figure 1). Consider the population loss:

$$f(\boldsymbol{w}) := \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}} \left[(L(\boldsymbol{x}; \boldsymbol{w}) - L(\boldsymbol{x}; \boldsymbol{w}^*))^2 \right]. \tag{1}$$

where $L(\boldsymbol{x}, \boldsymbol{w})$ is the output of the network with input $\boldsymbol{x}$ and weight $\boldsymbol{w}$ in the hidden layer. We assume there exists a ground truth $\boldsymbol{w}^*$. Consider sparsifying the network by minimizing the Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}) = f(\boldsymbol{w}) + \|\boldsymbol{w}\|_1 \tag{2}$$

where the ℓ_1 penalty can be changed to ℓ_0 or Transformed-ℓ_1 penalty [15,27]. Empirical experiments show that our method also works on deeper networks, since the sparsification on each layer happens independently of each other.

The rest of the paper is organized as follows. In Section 2, we briefly overview related mathematical results in the study of neural networks and complexity reduction. Preliminaries are in section 3. In Section 4, we state and discuss the main results. The proofs of the main results are in Section 5, and numerical experiments are in Section 6.

2   Related Work

In recent years, significant progress has been made in the study of convergence in neural network training. From a theoretical point of view, optimizing (training) neural network is a non-convex non-smooth optimization problem, which is mainly solved by (stochastic) gradient descent. Stochastic gradient descent methods were first proposed by Robins and Monro in 1951 [18]. Rumelhart et al. introduced the popular back-propagation algorithm in 1986 [19]. Since then, many well-known SGD methods with adaptive learning rates were proposed and applied in practice, such as the Polyak momentum [16], AdaGrad [6], RMSProp [23], Adam [9], and AMSGrad [17].

The behavior of gradient descent methods in neural networks is better understood when the input has *Gaussian* distribution. In 2017, Tian showed the population gradient descent can recover the true weight vector with random initialization for one-layer one-neuron model [22]. Brutzkus & Globerson (2017) showed that a convolution filter with non-overlapping input can be learned in polynomial time [1]. Du et al. showed (stochastic) gradient descent with random initialization can learn the convolutional filter in polynomial time and the convergence rate depends on the smoothness of the input distribution and the closeness of patches [4]. Du et al. also analyzed the polynomial convergence guarantee of randomly initialized gradient descent algorithm for learning a one-hidden-layer convolutional neural network [5]. Non-SGD methods for deep learning were also studied in the recent years. Taylor et al. proposed the Alternating Direction Method of Multipliers (ADMM) to transform a fully-connected neural network into an equality-constrained problem to solve [21]. A similar algorithm to the one introduced in this paper was discussed in [13]. There are a few notable differences. First, their parameter ϱ (respectively our parameter β) is large (resp. small). Secondly, the update on $\boldsymbol{w}$ in our paper does not have the form of an argmin update, but rather a gradient descent step. Lastly, their analysis does not apply to ReLU neural networks, and the checking step will be costly and impractical for large networks. In this paper, we will show that having β small is essential in showing descent of the Lagrangian, angle, and giving a strong error bound on the limit point. We became aware of [13] lately after our work was mostly done.

3   Preliminaries

3.1   The One-layer Non-overlap Network

In this paper, the input feature $\boldsymbol{x} \in \mathbb{R}^n$ is i.i.d. Gaussian random vector with zero mean and unit variance. Let $\mathcal{G}$ denote this distribution. We assume that there exists a ground truth $\boldsymbol{w}^*$ by which the training data is generated. The population risk is then:

$$f(\boldsymbol{w}) = \mathbb{E}_{\mathcal{G}}[(L(\boldsymbol{x}; \boldsymbol{w}) - L(\boldsymbol{x}; \boldsymbol{w}^*))^2]. \tag{3}$$

We define

$$g(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{E}_{\mathcal{G}}[\sigma(\boldsymbol{u} \cdot \boldsymbol{x})\sigma(\boldsymbol{v} \cdot \boldsymbol{x})]. \tag{4}$$

Then:

Lemma 1 *[1,2] Assume $\boldsymbol{x} \in \mathbb{R}^d$ is a vector where the entries are i.i.d. Gaussian random variables with mean 0 and variance 1. Given $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$, denote by $\theta_{\boldsymbol{u},\boldsymbol{v}}$ the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$. Then*

$$g(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{2\pi} \|\boldsymbol{u}\| \|\boldsymbol{v}\| \left(\sin \theta_{\boldsymbol{u},\boldsymbol{v}} + (\pi - \theta_{\boldsymbol{u},\boldsymbol{v}}) \cos \theta_{\boldsymbol{u},\boldsymbol{v}} \right).$$

For the no-overlap network, the loss function is simplified to:

$$f(\boldsymbol{w}) = \frac{1}{k^2} \left[a(\|\boldsymbol{w}\|^2 + \|\boldsymbol{w}^*\|^2) - 2kg(\boldsymbol{w}, \boldsymbol{w}^*) - 2b\|\boldsymbol{w}\| \|\boldsymbol{w}^*\| \right]. \tag{5}$$

where $b = \frac{k^2 - k}{2\pi}$ and $a = b + \frac{k}{2}$.

3.2   The Relaxed Variables Splitting Method

Let $\eta > 0$ denote the step size. Consider a simple gradient descent update:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \nabla f(\boldsymbol{w}^t). \tag{6}$$

It was shown [1] that the one-layer non-overlap network can be learned with high probability and in polynomial time. We seek to improve sparsity in the limit weight while also maintain good accuracy. A classical method to accomplish this task is to introduce ℓ_1 regularization to the population loss function, and the modified gradient update rule. Consider the minimization problem:

$$l(\boldsymbol{w}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1. \tag{7}$$

for some $\lambda > 0$. We propose a new approach to solve this minimization problem, called the Relaxed Variable Splitting Method (RVSM). We first convert (7) into an equation of two variables

$$l(\boldsymbol{w}, \boldsymbol{u}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{u}\|_1.$$

and consider the augmented Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{u}\|_1 + \frac{\beta}{2} \|\boldsymbol{w} - \boldsymbol{u}\|^2. \tag{8}$$

Let $S_{\lambda/\beta}(\boldsymbol{w}) := sgn(\boldsymbol{w})(|\boldsymbol{w}| - \lambda/\beta)\chi_{\{|\boldsymbol{w}| > \lambda/\beta\}}$ be the soft thresholding operator. The RSVM is:

Algorithm 1 RVSM

  Input: $\eta, \beta, \lambda, max_{epoch}, max_{batch}$
  Initialization: $\boldsymbol{w}^0$
  Define: $\boldsymbol{u}^0 = S_{\lambda/\beta}(\boldsymbol{w}^0)$
  for $t = 0, 1, 2, ..., max_{epoch}$ **do**
    for $batch = 1, 2, ..., max_{batch}$ **do**
      $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \nabla f(\boldsymbol{w}^t) - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^t)$
      $\boldsymbol{u}^{t+1} \leftarrow \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}) = S_{\lambda/\beta}(\boldsymbol{w}^t)$
    end for
  end for

3.3   Comparison with ADMM

A well-known, modern method to solve the minimization problem (7) is the Alternating Direction Method of Multipliers (or ADMM). In ADMM, we consider the Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}, \boldsymbol{z}) = f(\boldsymbol{w}) + \lambda \|\boldsymbol{u}\|_1 + \langle \boldsymbol{z}, \boldsymbol{w} - \boldsymbol{u} \rangle + \frac{\beta}{2} \|\boldsymbol{w} - \boldsymbol{u}\|^2. \tag{9}$$

and apply the updates:

$$\begin{cases} \boldsymbol{w}^{t+1} \leftarrow \arg\min_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}^t, \boldsymbol{z}^t) \\ \boldsymbol{u}^{t+1} \leftarrow \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}, \boldsymbol{z}^t) \\ \boldsymbol{z}^{t+1} \leftarrow \boldsymbol{z}^t + \beta(\boldsymbol{w}^{t+1} - \boldsymbol{u}^{t+1}) \end{cases} \tag{10}$$

Although widely used in practice, the ADMM method has several drawbacks when it comes to regularizing deep neural networks: First, the loss function f is often non-convex and only differentiable in some very specific regions, thus the current theory of optimizations does not apply [25]. Secondly, the update

$$\boldsymbol{w}^{t+1} \leftarrow \arg\min_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}, \boldsymbol{z}^t)$$

is not applicable in practice on DNN, as it requires one to know fully how $f(\boldsymbol{w})$ be-haves. In most ADMM adaptations on DNN, this step is replaced by a simple gradient descent. Lastly, the Lagrange multiplier $\boldsymbol{z}^t$ tends to reduce the sparsity of the limit of $\boldsymbol{u}^t$, as it seeks to close the gap between $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$. In contrast, the RVSM method re-solves all these difficulties presented by ADMM. First, we will show that in a one-layer non-overlap network, the iterations will keep $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$ in a nice region, where one can guarantee Lipschitz gradient property for $f(\boldsymbol{w})$. Secondly, the update of $\boldsymbol{w}^t$ is not an $\arg\min$ update, but rather a gradient descent iteration itself, so our theory does not de-viate from practice. Lastly, without the Lagrange multiplier term $\boldsymbol{z}^t$, there will be a gap between $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$ at the limit. The $\boldsymbol{u}^t$ is much more sparse than in the case of ADMM, and numerical results showed that $f(\boldsymbol{w}^t)$ and $f(\boldsymbol{u}^t)$ behave very similarly on deep net-works. An intuitive explanation for this is that when the dimension of $\boldsymbol{w}^t$ is high, most of its components that will be pruned off to get $\boldsymbol{u}^t$ have very small magnitudes, and are often the redundant weights.

In short, the RVSM method is easier to implement (no need to keep track of the variable $\boldsymbol{z}^t$), can greatly increase sparsity in the weight variable $\boldsymbol{u}^t$, while also main-taining the same performance as ADMM. Moreover, RVSM has convergence guarantee and limit characterization as stated below.

4   Main Results

Before we state our main results, the following Lemma is needed to establish the exis-tence of a Lipschitz constant L:

Lemma 2 *(Lipschitz gradient)*
There exists a global constant L such that the iterations of Algorithm 1 satisfy

$$\|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^{t+1})\| \le L \|\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\|, \quad \forall t. \tag{11}$$

An important consequence of Lemma 2 is: for all t, the iterations of Algorithm 1 satisfy:

$$f(\boldsymbol{w}^{t+1}) - f(\boldsymbol{w}^t) \leq \langle \nabla f(\boldsymbol{w}^t), \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2.$$

Theorem 1. *Suppose the initialization of the RVSM Algorithm satisfies:*
(i) Step size η is small so that $\eta \leq \frac{1}{\beta+L}$;
(ii) Initial angle $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, for some $\delta > 0$;
(iii) Parameters k, β, λ are such that $k \geq 2$, $\beta \leq \frac{\delta \sin \delta}{k\pi}$, and $\frac{\lambda}{\beta} < \frac{1}{\sqrt{d}}$.
Then the Lagrangian $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ decreases monotonically; and $(\boldsymbol{w}^t, \boldsymbol{u}^t)$ converges sub-sequentially to a limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$, with $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$, such that:
(i) $0 \in \partial_{\boldsymbol{u}}\mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$ and $\nabla_{\boldsymbol{w}}\mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}) = 0$;
(ii) $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t) = O(\epsilon)$ in $O(1/\epsilon^2)$ iterations;
(iii) The limit point $\bar{\boldsymbol{w}}$ is close to the ground truth $\boldsymbol{w}^*$ in the sense that $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$ and $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\| = O(\beta)$.

The full proof of Theorem 1 is given in the next section. Here we overview the key steps. First, we show that the iterations of Algorithm 1 will eventually bring $\boldsymbol{w}^t$ to within a closed annulus D of width $2M$ around the sphere centered at origin with radius $\|\boldsymbol{w}^*\|$. In other words, there exists a T such that for all $t \geq T, \|\boldsymbol{w}^t\| \in [\|\boldsymbol{w}^*\| - M, \|\boldsymbol{w}^*\| + M]$, for some $0 < M < \|\boldsymbol{w}^*\|$. Then with no loss of generality, we can assume that $\boldsymbol{w}^t$ is in this closed strip, for all t.

Next, for the region D of the iterations, we will show there exists a global constant L such that the Lipschitz gradient property in Lemma 2 holds.

Finally, the Lipschitz gradient property allows us to show the descent of angle θ^t and Lagrangian $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$. The conditions on η, β, λ are used to show $\theta^{t+1} \leq \theta^t$; and an analysis of the limit point gives the bound on $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$ and $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$. From the descent property of $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$, classical results from optimization [1] can be used to show that after $T = O\left(\frac{1}{\epsilon^2}\right)$ iterations, we have $\nabla_{\boldsymbol{w}}\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t) = O(\epsilon)$, for some $t \in (0, T]$. This leads to the desired convergence rate and finishes the proof.

It should be noted that without the condition on β being small, one may not guarantee monotonicity of θ^t. However, it still can be shown that $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ decreases and thus the iteration will converge to some limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. In this case, the limit point may not be near the ground truth $\boldsymbol{w}^*$; i.e. we may not have $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$. Furthermore, the bound on $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$ will also be weaker.

Corollary 1. *Suppose the initialization of the RVSM Algorithm satisfies Theorem 1, then the $\bar{\boldsymbol{w}}$ equation below holds:*

$$\boldsymbol{w}^* = \frac{k\pi}{\pi - \theta}\beta(\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}})) + C\bar{\boldsymbol{w}}, \tag{12}$$

where $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$, constant $C \in (0, \frac{1}{1-2k\lambda\sqrt{d}})$. Since component-wise, $\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}})$ has the same sign as $\bar{\boldsymbol{w}}$, the ground truth $\boldsymbol{w}^*$ is an expansion of $C\bar{\boldsymbol{w}}$, or equivalently $\bar{\boldsymbol{w}}$ is (up to scalar multiple) a shrinkage of $\boldsymbol{w}^*$.

The proofs of Theorem 1 and Corollary 1.1 do not require convexity of the regularization term $\lambda\|\boldsymbol{u}\|_1$, hence extend to other sparse penalties such as ℓ_0 and transformed ℓ_1 penalty [27]. We have:

Corollary 2. *Under the conditions of Theorem 1 however with the l_1 penalty replaced by an ℓ_0 or transformed-ℓ_1 penalty, the RVSM Algorithm converges sub-sequentially to a limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$ satisfying $\nabla_{\boldsymbol{w}}\mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}) = 0$. The Lagrangian and angle θ^t also decrease monotonically, with the limit angle satisfying $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$. Here $\bar{\boldsymbol{u}}$ is a thresholding of $\bar{\boldsymbol{w}}$, and equation (12) holds with $S_{\lambda/\beta}(\cdot)$ replaced by the thresholding operator of the corresponding penalty.*

5   Proof of Main Results

The following Lemmas will be needed to prove Theorem 1:

Lemma 3 *(Properties of the gradient, [1])*
For the loss function $f(\boldsymbol{w})$ of equation (5), the following holds:
1. $f(\boldsymbol{w})$ is differentiable if and only if $\boldsymbol{w} \neq 0$.
2. For $k > 1$, $f(\boldsymbol{w})$ has three critical points:
(a) A local maximum at $\boldsymbol{w} = 0$.
(b) A unique global minimum at $\boldsymbol{w} = \boldsymbol{w}^*$.
(c) A degenerate saddle point at $\boldsymbol{w} = -\left(\frac{k^2-k}{k^2+(\pi-1)k}\right)\boldsymbol{w}^*$.
For $k = 1, w = 0$ is not a local maximum and the unique global minimum $\boldsymbol{w}^*$ is the only differentiable critical point.
Given $\theta := \theta(\boldsymbol{w}, \boldsymbol{w}^*)$, the gradient of f can be expressed as

$$\nabla f(\boldsymbol{w}) = \frac{1}{k^2}\left[\left(k + \frac{k^2-k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\sin\theta - \frac{k^2-k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\right)\boldsymbol{w} - \frac{k}{\pi}(\pi-\theta)\boldsymbol{w}^*\right]. \tag{13}$$

Lemma 4 *(Lipschitz gradient with co-planar assumption, [1])*
Assume $\|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \geq M$, $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}^*$ are on the same two dimensional half-plane defined by $\boldsymbol{w}^*$, then

$$\|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\| \leq L\|\boldsymbol{w}_1 - w_2\|$$

for $L = 1 + \frac{3\|\boldsymbol{w}^*\|}{M}$.

Lemma 5 *For $k \geq 1$, there exists constants $M_k, T > 0$ such that for all $t \geq T$, the iterations of Algorithm 1 satisfy:*

$$\|\boldsymbol{w}^t\| \in [\|\boldsymbol{w}^*\| - M_k, \|\boldsymbol{w}^*\| + M_k]. \tag{14}$$

where $M_k < \|\boldsymbol{w}^*\|$, and $M_k \to 0$ as $k \to \infty$.

From Lemma 5, WLOG, we will assume that $T = 0$.

Lemma 6 *(Descent of $\mathcal{L}_\beta$ due to $\boldsymbol{w}$ update)*
For η small such that $\eta \leq \frac{1}{\beta+L}$, we have

$$\mathcal{L}_\beta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t).$$

5.1   Proof of Lemma 2

By Algorithm 1 and Lemma 5, $\|\boldsymbol{w}^t\| \geq \|\boldsymbol{w}^*\| - M > 0$, for all t, and $\boldsymbol{w}^{t+1}$ is in some closed neighborhood of $\boldsymbol{w}^t$. We consider the subspace spanned by $\boldsymbol{w}^t, \boldsymbol{w}^{t+1}$, and $\boldsymbol{w}^*$, this reduces the problem to a 3-dimensional space.

Consider the plane formed by $\boldsymbol{w}^t$ and $\boldsymbol{w}^*$. Let $\boldsymbol{v}^{t+1}$ be the point on this plane, closest to $\boldsymbol{w}^t$, such that $\|\boldsymbol{w}^{t+1}\| = \|\boldsymbol{v}^{t+1}\|$ and $\theta(\boldsymbol{w}^{t+1}, \boldsymbol{w}^*) = \theta(\boldsymbol{v}^{t+1}, \boldsymbol{w}^*)$. In other words, $\boldsymbol{v}^{t+1}$ is the intersection of the plane formed by $\boldsymbol{w}^t, \boldsymbol{w}^*$ and the cone with tip at zero, side length $\|\boldsymbol{w}^{t+1}\|$, and main axis $\boldsymbol{w}^*$ (See Figure 2). Then

Fig. 2. Geometry of the update of $\boldsymbol{w}^t$ and the corresponding $\boldsymbol{w}^{t+1}, \boldsymbol{v}^{t+1}$.

$$\|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^{t+1})\|$$
$$\leq \|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{v}^{t+1})\| + \|\nabla f(\boldsymbol{v}^{t+1}) - \nabla f(\boldsymbol{w}^{t+1})\|$$
$$\leq L_1 \|\boldsymbol{w}^t - \boldsymbol{v}^{t+1}\| + L_2 \|\boldsymbol{v}^{t+1} - \boldsymbol{w}^{t+1}\| \tag{15}$$

for some constants L_1, L_2. The first term is bounded since $\boldsymbol{w}^t, \boldsymbol{v}^{t+1}, \boldsymbol{w}^*$ are co-planar by construction, and Lemma 4 applies. The second term is bounded by applying Equation 13 with $\|\boldsymbol{w}^{t+1}\| = \|\boldsymbol{v}^{t+1}\|$ and $\theta(\boldsymbol{w}^{t+1}, \boldsymbol{w}^*) = \theta(\boldsymbol{v}^{t+1}, \boldsymbol{w}^*)$. It remains to show there exists a constant $L_3 > 0$ such that

$$\|\boldsymbol{w}^t - \boldsymbol{v}^{t+1}\| + \|\boldsymbol{v}^{t+1} - \boldsymbol{w}^{t+1}\| \leq L_3 \|\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\|$$

Let A, B, C be the tips of $\boldsymbol{w}^t, \boldsymbol{v}^{t+1}, \boldsymbol{w}^{t+1}$, respectively. Let P be the point on $\boldsymbol{w}^*$ that is at the base of the cone (so P is the center of the circle with B, C on the arc). We will show there exists a constant L_3 such that

$$|AB| + |BC| \leq L_3 |AC| \tag{16}$$

<u>Case 1:</u> A, B, P are collinear: By looking at the cross-section of the plane formed by AB, AC, it can be seen that AC is not the smallest edge in $\triangle ABC$. Thus there exists some L_3 such that Equation 16 holds.

<u>Case 2</u>: A, B, P are not collinear: Translate B, C, P to B', C', P' such that A, B', P' are collinear and $BB', CC', PP' /\!/ \boldsymbol{w}^*$. Then by Case 1:

$$|AB'| + |B'C'| \le L_3 |AC'|$$

and AC' is not the smallest edge in $\triangle AB'C'$. By back-translating B', C' to B, C, it can be seen that AC is again not the smallest edge in $\triangle ABC$. This implies

$$|AB| + |BC| \le L_4 |AC|$$

for some constant L_4. Thus Equation 16 is proved. Combining with Equation 15, Lemma 2 is proved.

5.2 Proof of Lemma 5

First we show that if $\|\boldsymbol{w}^t\| < \|\boldsymbol{w}^*\|$, then the update of Algorithm 1 will satisfy $\|\boldsymbol{w}^{t+1}\| > \|\boldsymbol{w}^t\|$. By Lemma 3,

$$\nabla f(\boldsymbol{w}) = \frac{1}{k^2}\left[\left(k + \frac{k^2 - k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\sin\theta - \frac{k^2 - k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\right)\boldsymbol{w} - \frac{k}{\pi}(\pi - \theta)\boldsymbol{w}^*\right]$$
$$= \frac{1}{k^2}(C_1\boldsymbol{w} - C_2\boldsymbol{w}^*)$$

so the update of $\boldsymbol{w}^t$ reads

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta\frac{C_1^t + \beta k^2}{k^2}\boldsymbol{w}^t + \eta\frac{C_2^t}{k^2}\boldsymbol{w}^* + \eta\beta\boldsymbol{u}^{t+1},$$

where $C_2^t > 0$. Since $\boldsymbol{u}^{t+1} = S_{\lambda/\beta}(\boldsymbol{w}^t)$, the term $\eta\beta\boldsymbol{u}^{t+1}$ will increase the norm of $\boldsymbol{w}^t$. For the remaining terms,

$$C_1^t = k + \frac{k^2 - k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\sin\theta - \frac{k^2 - k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$$
$$\le k + \frac{k^2 - k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right)$$

When $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$ is large, C_1^t is negative. The update will increase the norm of $\|\boldsymbol{w}^t\|$ if $C_1^t + \beta k^2 \le 0$ and

$$\left\|\frac{C_1^t + \beta k^2}{k^2}\boldsymbol{w}^t\right\| > \left\|\frac{C_2^t}{k^2}\boldsymbol{w}^*\right\|$$

This condition is satisfied when

$$-\left[k + \frac{k^2 - k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right) + \beta k^2\right] > \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$$

When $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} > 1$, the LHS is $O(k^2)$, while the RHS is $O(k)$. Thus there exists some M_k such that $\boldsymbol{w}^t$ will eventually stay in the region $\|\boldsymbol{w}^t\| \ge \|\boldsymbol{w}^*\| - M_k$. Moreover, as

$k \to \infty$, we have $M_k \to 0$.

Next, when $\|\boldsymbol{w}^t\| > \|\boldsymbol{w}^*\|$, the update of $\boldsymbol{w}^t$ reads

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \frac{C_1^t}{k^2}\boldsymbol{w}^t + \eta \frac{C_2^t}{k^2}\boldsymbol{w}^* - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$$

the last term decreases the norm of $\boldsymbol{w}^t$. In this case, C_1^t is positive and

$$C_1^t \geq \frac{k\pi - k}{\pi} + \frac{k^2 - k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right)$$

The update will decrease the norm of $\boldsymbol{w}^t$ if

$$\frac{k\pi - k}{\pi} + \frac{k^2 - k}{\pi}\left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right) > \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$$

which holds when $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} < 1$, and the Lemma is proved.

5.3   Proof of Lemma 6

Proof. By the update of $\boldsymbol{u}^t$, $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$. For the update of $\boldsymbol{w}^t$, notice that

$$\nabla f(\boldsymbol{w}^t) = \frac{1}{\eta}\left(\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\right) - \beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$$

Then for a fixed $\boldsymbol{u} := \boldsymbol{u}^{t+1}$, we have

$$\begin{aligned}
&\mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}) - \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u})\\
=& f(\boldsymbol{w}^{t+1}) - f(\boldsymbol{w}^t) + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right)\\
\leq& \langle\nabla f(\boldsymbol{w}^t), \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2\\
&+ \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right)\\
=& \frac{1}{\eta}\langle\boldsymbol{w}^t - \boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \beta\langle\boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle\\
&+ \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right)\\
=& \frac{1}{\eta}\langle\boldsymbol{w}^t - \boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \left(\frac{L}{2} + \frac{\beta}{2}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2\\
&+ \frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \frac{\beta}{2}\|\boldsymbol{w}^t - \boldsymbol{u}\|^2\\
&- \beta\langle\boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2\\
=& \left(\frac{L}{2} + \frac{\beta}{2} - \frac{1}{\eta}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2
\end{aligned}$$

Therefore, if η is small so that $\eta \leq \frac{2}{\beta + L}$, the update on $\boldsymbol{w}$ will decrease $\mathcal{L}_\beta$.

5.4   Proof of Theorem 1

We will first show that if $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, then $\theta(\boldsymbol{w}^t, \boldsymbol{w}^*) \leq \pi - \delta$, for all t. We will show $\theta(\boldsymbol{w}^1, \boldsymbol{w}^*) \leq \pi - \delta$, the statement is then followed by induction. To this end, by the update of $\boldsymbol{w}^t$, one has

$$= C\boldsymbol{w}^0 + \eta \frac{\pi - \theta(\boldsymbol{w}^0, \boldsymbol{w}^*)}{k\pi} \boldsymbol{w}^* + \eta\beta\boldsymbol{u}^1$$

for some constant $C > 0$. Since $\boldsymbol{u}^1 = S_{\lambda/\beta}(\boldsymbol{w}^0), \theta(\boldsymbol{u}^1, \boldsymbol{w}^0) \leq \frac{\pi}{2}$. Notice that the sum of the first two terms on the RHS brings the vector closer to $\boldsymbol{w}^*$, while the last term may behave unexpectedly. Consider the worst case scenario: $\boldsymbol{w}^0, \boldsymbol{w}^*, \boldsymbol{u}^1$ are co-planar with $\theta(\boldsymbol{u}^1, \boldsymbol{w}^0) = \frac{\pi}{2}$, and $\boldsymbol{w}^*, \boldsymbol{u}^1$ are on two sides of $\boldsymbol{w}^0$ (See Figure 3). We need

Fig. 3. Worst case of the update on $\boldsymbol{w}^t$

$\frac{\delta}{k\pi}\boldsymbol{w}^* + \beta\boldsymbol{u}^1$ to be in region I. This condition is satisfied when β is small such that

$$\sin\delta \geq \frac{\beta\|\boldsymbol{u}^1\|}{\frac{\delta}{k\pi}\|\boldsymbol{w}^*\|} = \frac{k\pi\beta\|\boldsymbol{u}^1\|}{\delta}$$

since $\|\boldsymbol{u}^1\| \leq 1$, it is sufficient to have $\beta \leq \frac{\delta\sin\delta}{k\pi}$.

Next, consider the limit of the RVSM algorithm. Since $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ is non-negative, by Lemma 6, $\mathcal{L}_\beta$ converges to some limit $\mathcal{L}$. This implies $(\boldsymbol{w}^t, \boldsymbol{u}^t)$ converges to some stationary point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. By Lemma 3 and the update of $\boldsymbol{w}^t$, we have

$$\bar{\boldsymbol{w}} = c_1\bar{\boldsymbol{w}} + \eta c_2\boldsymbol{w}^* + \eta\beta\bar{\boldsymbol{u}} \tag{17}$$

for some constant $c_1 > 0, c_2 \geq 0$, where $c_2 = \frac{\pi-\theta}{k\pi}$, with $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$, and $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$. If $c_2 = 0$, then we must have $\bar{\boldsymbol{w}} /\!\!/ \bar{\boldsymbol{u}}$. But since $\bar{\boldsymbol{u}} = S_{\lambda/\beta}$, this implies all non-zero components of $\bar{\boldsymbol{w}}$ are either equal in magnitude, or all have magnitude smaller than $\frac{\lambda}{\beta}$. The latter case is not possible when $\frac{\lambda}{\beta} < \frac{1}{\sqrt{d}}$. Furthermore, $c_2 = 0$ when $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) = \pi$ or 0. We have shown that $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) \leq \pi - \delta$, thus $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) = 0$. Thus, $\bar{\boldsymbol{w}} = \boldsymbol{w}^*$, and all non-zero components of $\boldsymbol{w}^*$ are equal in magnitude. This has

probability zero if we assume $\boldsymbol{w}^*$ is initiated uniformly on the unit circle. Hence we will assume that almost surely, $c_2 > 0$. Expression (17) implies

$$c_2 \boldsymbol{w}^* + \beta \bar{\boldsymbol{u}} /\!/ \bar{\boldsymbol{w}} \tag{18}$$

Expression (18) implies $\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}},$ and $\boldsymbol{w}^*$ are co-planar. Let $\gamma := \theta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. From expression (18), and the assumption that $\|\boldsymbol{w}^*\| = 1$, we have

$$(\langle c_2 \boldsymbol{w}^* + \beta \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}} \rangle)^2 = \|c_2 \boldsymbol{w}^* + \beta \bar{\boldsymbol{u}}\|^2 \|\bar{\boldsymbol{w}}\|^2$$

or

$$\|\bar{\boldsymbol{w}}\|^2 (c_2^2 \cos^2 \theta + 2 c_2 \beta \|\bar{\boldsymbol{u}}\| \cos \theta \cos \gamma + \beta^2 \|\bar{\boldsymbol{u}}\|^2 \cos^2 \gamma)$$
$$= \|\bar{\boldsymbol{w}}\|^2 (c_2^2 + 2 c_2 \beta \|\bar{\boldsymbol{u}}\| \cos(\theta + \gamma) + \beta^2 \|\bar{\boldsymbol{u}}\|^2)$$

This reduces to

$$c_2^2 \sin^2 \theta - 2 c_2 \beta \|\bar{\boldsymbol{u}}\| \sin \theta \sin \gamma + \beta^2 \|\bar{\boldsymbol{u}}\|^2 \sin^2 \gamma = 0,$$

which implies $\frac{\pi - \theta}{k\pi} \sin \theta = \beta \|\bar{\boldsymbol{u}}\| \sin \gamma$. By the initialization $\beta \leq \frac{\delta \sin \delta}{k\pi}$, we have $\frac{\pi - \theta}{k\pi} \sin \theta < \frac{\delta}{k\pi} \sin \delta$. This implies $\theta < \delta$.

Finally, the limit point satisfies $\|\nabla f(\bar{\boldsymbol{w}}) + \beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\| = 0$. By the initialization requirement, we have $\|\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\| < \beta \leq \frac{\delta \sin \delta}{k\pi}$. This implies $\|\nabla f(\bar{\boldsymbol{w}})\| \leq \frac{\delta \sin \delta}{k\pi}$. By the Lipschitz gardient property in Lemma 2 and critical points property in Lemma 3, $\bar{\boldsymbol{w}}$ must be close to $\boldsymbol{w}^*$. In other words, $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$ is comparable to the chord length of the circle of radius $\|\boldsymbol{w}^*\|$ and angle θ:

$$\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\| = O\left(2 \sin\left(\frac{\theta}{2}\right)\right) = O(\sin \theta)$$
$$= O\left(\frac{k\pi \beta \|\bar{\boldsymbol{u}}\| \sin \gamma}{\pi - \theta}\right) = O(k\beta \sin \gamma).$$

6   Numerical Experiments

First, we experiment RVSM with VGG-16 on the CIFAR10 data set. Table 1 shows the result of RVSM under different penalties. The parameters used are $\lambda = 1.e - 5, \beta = 1.e - 2,$ and $a = 1$ for $T\ell_1$ penalty. It can be seen that RVSM can maintain very good accuracy while also promotes good sparsity in the trained network. Between the penalties, ℓ_0 gives the best sparsity, ℓ_1 the best accuracy, and $T\ell_1$ gives a middle ground between ℓ_0 and ℓ_1. Since the only difference between these parameters is in the pruning threshold, in practice, one may simply stick to ℓ_0 regularization and just fine-tune the hyper-parameters.

Secondly, we experiment our method on ResNet18 and the CIFAR10 data set. The results are displayed in Table 2. The base model was trained on 200 epochs using standard SGD method with initial learning rate 0.1, which decays by a factor of 10 at the 80th, 120th, and 160th epochs. For the RVSM method, we use ℓ_0 regularization and

set $\lambda = 1.e\text{-}6$, $\beta = 8.e\text{-}2$. For ADMM, we set the pruning threshold to be 60% and $\rho = 1.e\text{-}2$. The ADMM method implemented here is per [28], an "empirical variation" of the true ADMM (Eq. 10). In particular, the $\arg\min$ update of $\boldsymbol{w}^t$ is replaced by a gradient descent step. Such "modified" ADMM is commonly used in practice on DNN.

It can be seen in Table 2 that RVSM runs quite effectively on the benchmark deep network, promote much better sparsity than ADMM (93.70% vs. 47.08%), and has slightly better performance. The sparsity here is the percentage of zero components over all network weights.

Table 1. Sparsity and accuracy of RVSM under different penalties on VGG-16 on CIFAR10.

Penalty	Accuracy	Sparsity
Base model	93.82	0
ℓ_1	93.7	35.68
Tℓ_1	93.07	63.34
ℓ_0	92.54	86.89

Table 2. Comparison between ADMM and RVSM (ℓ_0) for ResNet18 training on the CIFAR10 dataset.

ResNet18	Accuracy	Sparsity
SGD	95.07	0
ADMM	94.84	47.08
RVSM (ℓ_0)	94.89	93.70

7   Conclusion

We proved the global convergence of RVSM to sparsify a convolutional ReLU network on a regression problem and analyzed the sparsity of the limiting weight vector as well as its error estimate from the ground truth (i.e. the global minimum). The proof used geometric argument to establish angle and Lagrangian descent properties of the iterations thereby overcame the non-existence of gradient at the origin of the loss function. Our experimental results provided additional support for the effectiveness of RVSM via ℓ_0, ℓ_1 and Tℓ_1 penalties on standard deep networks and CIFAR-10 image data. In future work, we plan to extend RVSM theory to multi-layer network and structured (channel/filter/etc.) pruning.

8   Acknowledgments

References

1. Brutzkus, A., Globerson, A.: Globally optimal gradient descent for a convnet with gaussian inputs (2017). ArXiv preprint 1702.07966
2. Cho, Y., Saul, L.K.: Kernel methods for deep learning. In Advances in neural information processing systems pp. 342–350 (2009)
3. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. (2016). ArXiv preprint 1612.08083
4. Du, S., Lee, J., Tian, Y.: When is a convolutional filter easy to learn? (2017). ArXiv 1709.06129
5. Du, S., Lee, J., Tian, Y., Poczos, B., Singh, A.: Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. In: International Conference on Machine Learning (ICML) (2018)
6. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**, 2121–2159 (2011)
7. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding (2015). ArXiv preprint 1510.00149
8. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine **29**(6), 82–97 (2012)
9. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. (2014). ArXiv preprint 1412.6980
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems pp. 1097–1105 (2012)
11. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. NIPS **2**, 598–605 (1989)
12. Louizos, C., Welling, M., Kingma, D.: Learning sparse neural networks through ℓ_0 regularization (2018). ArXiv preprint 1712.01312v2
13. Lu, Z., Zhang, Y.: Penalty decomposition methods for rank minimization. Optimization Methods and Software **30**(3), 531558 (2014). DOI 10.1080/10556788.2014.936438
14. Molchanov, D., Ashukha, A., Vetrov, D.: Variational dropout sparsifies deep neural networks (2017). ArXiv preprint 1701.05369
15. Nikolova, M.: Local strong homogeneity of a regularized estimator. SIAM Journal on Applied Mathematics **61**(2), 633–658 (2000)
16. Polyak, B.: Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics **4**(5), 1–17 (1964)
17. Reddi, S., Kale, S., Kumar, S.: On the convergence of adam and beyond. In: International Conference on Learning Representations (2018)
18. Robbins, H., Monro, S.: A stochastic approximation method. Annals of Mathematical Statistics **22**, 400–407 (1951)
19. Rumelhart, D., Hinton, G., Williams, R.: Learning representations by back-propagating errors. Nature **323**, 533536 (1986)
20. Shamir, O.: Distribution-specific hardness of learning neural networks (2016). ArXiv preprint 1609.01037
21. Taylor, G., Burmeister, R., Xu, Z., Singh, B., Patel, A., Goldstein, T.: Training neural networks without gradients: A scalable admm approach. In: International Conference on Machine Learning, pp. 2722–2731 (2016)
22. Tian, Y.: An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis (2017). ArXiv preprint 1703.00560

23. Tieleman, T., Hinton, G.: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. Tech. rep., Technical report (2017)

24. Ullrich, K., Meeds, E., Welling, M.: Soft weight-sharing for neural network compression. ICLR (2017)

25. Wang, Y., Zeng, J., Yin, W.: Global Convergence of ADMM in Nonconvex Nonsmooth Optimization. Journal of Scientific Computing, online (2018). DOI 10.1007/s10915-018-0757-z

26. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization (2016). ArXiv preprint 1611.03530

27. Zhang, S., Xin, J.: Minimization of transformed l_1 penalty: Closed form representation and iterative thresholding algorithms. Communications in Mathematical Sciences **15**(2), 511–537 (2017). DOI 10.4310/cms.2017.v15.n2.a9

28. Zhang, T., Ye, S., Zhang, K., Tang, J., Wen, W., Fardad, M., Wang, Y.: A systematic dnn weight pruning framework using alternating direction method of multipliers. arXiv preprint 1804.03294 (2018). URL https://arxiv.org/abs/1804.03294