Analysis of Stochastic Gradient Descent

Ahmed Almostafa Gashgash
Technical Report

December 4th 2018

Abstract

Standard analysis of gradient descent does not distinguish between saddle points and local minima. In many problems of interest, saddle points are ubiquitous, and correspond to highly sub-optimal solutions. In this paper we present two algorithms that guarantee escaping saddle points and converging to a second order stationary point. It is assumed that saddle points are non-degenerate, and all second order stationary point are local minima. The first algorithm was the first work that provided global convergence guarantees for stochastic gradient descent. The number of iterations depended polynomially on the underlying dimension. The second algorithm extended this result and showed global convergence in a number of iterations that are independent of dimension up to poly-logarithmic terms.

1 Introduction

Gradient descent and its variants, such as stochastic gradient, are widely used first order iterative optimization algorithms for finding the minimum of an objective function. Due to their favorable computational properties, they have been very popular in machine learning tasks, especially in deep learning settings. A widely used method in deep learning named back-propagation is based on gradient descent. To find a local minima using gradient descent, the algorithm iteratively takes steps in the negative direction of the gradient until it converges to a first order stationary point, where the value of the gradient is zero. In convex settings, finding a local minima is equivalent to finding a global minima. Another favorable property of gradient descent is it is very efficient in high dimensional settings, since its convergence to a first order stationary points does not depend on the dimensionality of the data. In other words, for a L- Gradient Lipschitz function (see Definition 1) it requires $O(L(f(\mathbf{x}_0) - f^*)/\epsilon^2)$ iterations [Nesterov, 1998], where $\mathbf{x}_0$ is the starting point and f^* is the global minima of f, which is independent of the dimension d.

However, gradient descent is computationally expensive for large data sets and computing the gradient can be slow. Also, in non-convex settings, gradient descent could get stuck at a first order stationary point that is neither a local or global minima, such saddle points. For these reasons, stochastic gradient descent, which introduces small randomness to traditional gradient descent is more favorable. For non-convex settings, finding the global minima is NP-hard, however some recent work has shown that some problems of interest such as tenser decomposition [Ge, 2015], phase retrieval [Sun, 2016], and dictionary learning [Sun, 2016], it is sufficient to find a local minima. On the other hand, finding a saddle point is not as sufficient, and in many problems they correspond to a highly sub-optimal solution, such as in [Jain, 2015]. These saddle points are very common in high dimensional, non-convex settings, and form the main bottleneck in training neural networks [Dauphin 2014].

Traditional analysis of gradient descent does not distinguish between saddle points and local minima. Therefore the algorithm may get stuck at a saddle point, either asymptotically or for long enough to make the run time infeasible. However recently, researchers found that stochastic gradient descent is able to escape saddle points. In Ge et al. [2015], it was shoen that by adding noise to the gradient at every step, SGD was able to escape all saddle points in a number of iterations that depend polynomially on the dimension of the data, as long as the function satisfies the strict saddle property. In this paper we present this result and explain the proposed algorithm.

This result is still not efficient. The convergence polynomial dependence on the underlying dimension is sub-optimal compared to convergence to first order stationary points, where there is no dependence on convergence. In Jin et al. [2017], the authors aim to answer following question: Is it possible for gradient descent to escape all saddle points and converge to a minima in a number of iterations that are independent of dimension? They were successful in proposing a perturbed gradient descent algorithm that escaped all saddle points and converged close to a second order stationary point with only poly-logarithmic dependence on the dimension. Simialr to Ge et al. [2015], they added noise to the gradient only when a

perturbation condition was satisifed. In this paper we will present this algorithm, analyze it and provide a proof for it's correctness.

2 Preliminaries

In this paper we will use $\lambda_{min}(.)$ to denote the smallest eigenvalue, for vectors we will use $||.||$ to denote the l_2 -norm and spectral norms of matrices. We let $\mathbb{B}_{\mathbf{x}}(r)$ be a ball centered at $\mathbf{x}$ with radius r, its dimension will follow the context. For a function $f : \mathbb{R}^d \mapsto \mathbb{R}$, ∇f and $\nabla^2 f$ are its Gradient and Hessian matrices accordingly.

2.1 Gradient Descent

The Gradient Descent method aims to solve the following optimization problem:

$$x = \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} f(x) \tag{1}$$

Iteratively it does the following updates:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) \tag{2}$$

where η is the learning rate at each step t. The major problem of Gradient Descent, as discussed before, is that computing the gradient of the whole data set $\mathbf{x}$ is computationally expensive if the set is large. Also, the algorithm may converge to a stationary point that is neither a local or global minimum and get stuck. We therefore use a randomized version of Gradient Descent that is able to converge to the global minimum in the convex case, and is much more computationally efficient since it uses a smaller data batch size to compute the gradient. This is known as the Stochastic Gradient, similar to Gradient Descent but with added randomness as follows:

$$f(\mathbf{x}) = \mathbb{E}[g(\mathbf{x}] \tag{3}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla g(\mathbf{x}_t) \tag{4}$$

Before introducing how the Stochastic Gradient can escape saddle points, we first state some traditional theoretic definitions that are used in analyzing Gradient Descent algorithms on convex functions:

Definition 1: A differentiable function f is said to be L- gradient Lipschitz if:

$$\forall x_1, x_2 \quad ||\nabla f(x_1) - \nabla f(x_2)|| \leq L||x_1 - x_2||$$

This implies smoothness, which means that the gradient does not change rapidly with respect to the l_2-norm.

Definition 2: A twice differentiable function f is said to be ρ- Hessian Lipschitz :

$$\forall x_1, x_2 \quad ||\nabla^2 f(x_1) - \nabla^2 f(x_2)|| \leq \rho ||x_1 - x_2||$$

This is also implies that the Hessian cannot change dramatically with respect to the spectral norm.

Definition 3: For stationary points ($\nabla f(x) = 0$), a local maximum is when $\nabla^2 f(x) \prec 0$, a local minimum is when $\nabla^2 f(x) \succ 0$ and a saddle point is when $\nabla^2 f(x)$ has positive and negative eigenvalues.

We do not consider the degenerate case in which the Hessian has eigenvalues equal to zero. Instead we will deal with strict saddle points, where $\mathbf{x}$ is said to be a strict saddle point if it satisfies $\lambda_{min}(\nabla^2 f(\mathbf{x}) < 0$. Using the above definitions, we define second-order stationary points. These are very important in non-convex optimization, since if all saddle points are strict, then all second-order stationary points are local minima [Jin 2017].

Definition 4: For a ρ- Hessian Lipschitz function f, then $\mathbf{x}$ is a second-order stationary point if $||\nabla f(\mathbf{x})|| = 0$ and $\lambda_{min}(\nabla^2 f(\mathbf{x})) \geq 0$. Relaxing this we also say that $\mathbf{x}$ is a ϵ-second-order stationary point if:

$$||\nabla f(\mathbf{x})|| \leq \epsilon \qquad \text{and} \qquad \lambda_{min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\rho\epsilon}$$

This definition of ϵ-second-order stationary points in Jin [2017] follows the convention of Polyak [2006] and it reflects the natural relation between the gradient and the Hessian.

3 Escaping Saddle Points

Recently there has been many non-convex optimization convergence results for specific problems. In this paper we will introduce two results from Ge et al [2015] and Jin et al [2017],that guarantee convergence to a second-order stationary point without any use of Hessian information. These methods involve simple calculations involving only gradient operations. We will also provide a proof sketch of the main theorem in Jin et al [2017].

3.1 Noisy Stochastic Gradient

In his paper, Ge et al [2015] showed that adding noise to the stochastic gradient at every iteration guarantees convergence to a second-order stationary point in poly($\frac{d}{\epsilon}$) iterations, where the polynomial is of order at least 4 [Ge, 2015]. He identifies a property of a function f that guarantees stochastic gradient descent to converge to a local minimum efficiently, called the *strict saddle property*.

3.1.1 Strict Saddle Property:

A lot of objective functions in real world applications exhibit the strict saddle property. For example these are present in Orthogonal tensor decomposition [Ge et al 2015], deep residual networks [Kawaguchi et al 2016], Matrix completion [Ge et al 2016], Generalized phase retrieval problem [Sun et al 2016], Low rank matrix recovery [Bhojanapalli et al 2016]. In all these problems, arriving at a local minima is equally good as asrriving at a golbal minima. More formally we define the strict saddle property as follows [Jin 2017].:

Definition 5: A function f is (θ, γ, ζ) - strict saddle if for any $\mathbf{x}$, at ;east one of the following holds:

- $||\nabla f(\mathbf{x})|| \geq \theta$
- $\lambda_{min}(\nabla^2 f(\mathbf{x})) \leq -\gamma 0$
- $\mathbf{x}$ is ζ-close to χ^*

where χ^* is the set of local minima. In other words, this property states the presence of three regions in the $\mathbb{R}^d$ space. The first is the region where the gradient is large, the second is when the Hessian has a negative eigenvalue around the saddle point in which it can escape, and the last is where we are at a stationary point with no escape direction but are very close to a local minima.

In [Ge, 2015], a simple variant of stochastic gradient descent is introduced (see Algorithm 1). An extra noise term, sampled uniformly from a unit sphere, is added to the updates. Having noise in every direction allows the algorithm to explore the local neighborhood around a saddle point.

Algorithm 1 Noisy Stochastic Gradient Descent: NSGD($\mathbf{x}_0, \eta$)

1: $T \leftarrow \tilde{O}(\frac{1}{\eta^2})$
2: **for** `t=0,1,2,..., T-1` **do**
$noise \leftarrow \xi_t, \quad \xi_t \in$N(0,**I**)
$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) + noise)$

This work by Ge et al [2015] was one of the very first that proved global convergence to a second order stationary point using a gradient oracle, however it converges in at least $\Omega(d^4)$ iterations. To address this, Jin et al [2017] proposed a perturbed stochastic gradient descent that is able to converge to a local minima in a number of iterations that are almost dimension free, in other words, with only poly-log dependency on the dimension [Jin 2017]. In this next subsection, I will introduce the algorithm and the main theorem, and also provide a sketch of the theorems proof.

3.2 Perturbed Gradient Descent:

Jin et al. [2015] present the first sharp analysis showing that a perturbed gradient descent finds an approximate second order stationary point in at most polylog(d) iterations. Theorem 1 presents the detailed complexity of finding this point. This result matches well with the well known convergence rate of gradient descent to first order stationary points, up to polylog factors. The PGD algorith proposed (see Algorithm 2) follows a similar idea to the work introduced by Ge et al [2015], but only adding noise to the gradient if a perturbation condition holds. Also, under strict saddle conditions, as defined earlier, this convergence directly applies to finding a local minima.

Algorithm 2 Perturbed Gradient Descent: PGD($\mathbf{x}_0, \rho, L, \epsilon, c, \delta, \nabla_f$)

$\chi \leftarrow 3\max\{\log(\frac{dl\nabla_f}{c\epsilon^2\delta}), 4\}$, $\eta \leftarrow \frac{c}{L}$, $r \leftarrow \frac{\sqrt{c}}{\chi^2}.\frac{\epsilon}{L}$, $g_{thres} \leftarrow \frac{\sqrt{c}}{\chi^2}.\epsilon$, $f_{thres} \leftarrow \frac{c}{\chi^3}.\sqrt{\frac{\epsilon^3}{\rho}}$, $t_{thres} \leftarrow \frac{\chi}{c^2}.\frac{L}{\sqrt{\rho\epsilon}}$, $t_{noise} \leftarrow -t_{thres} - 1$,

for `t=0,1,2,...` **do**

  if $||\nabla f(\mathbf{x}_t)|| \leq g_{tres}$ and $t - t_{noise} > t_{tres}$ **then**

    $\mathbf{m}_t \leftarrow \mathbf{x}_t$, $t_{noise} \leftarrow t$,

    $\mathbf{x}_t \leftarrow \mathbf{m}_t + \xi_t$, ξ_t uniformly $\sim \mathbb{B}_0(r)$

  if $t - t_{noise} = t_{tres}$ and $f(\mathbf{x}_t) - f(\mathbf{m}_t) > -f_{thres}$ **then**

    return $\mathbf{m}_{tnoise}$

  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t)$

The Perturbed Gradient Descent algorithm above is based on a gradient descent with step size η. This algorithm only adds a small perturbation to the algorithm if the norm of the current gradient is small, hence $\leq g_{thres}$. This indicates that the most current iterate is near a saddle point. This perturbation is added at most once every t_{thres} iterations. This ensures that the dynamics are mostly of gradient descent. The perturbation added is uniformly sampled from a d-dimensional ball. Also, if the gradient does not decrease the value of the objective function by at least f_{thres}, then the algorithm outputs $\mathbf{m}_{tnoise}$. The following theorem and analysis guarantee that the algorithms output is necessarily close to a second order stationary point [Jin 2017]..

Theorem 1. Assume a function f that is L- gradient Lipschitz and ρ- Hessian Lipschitz, then there exists an absolute constant c_{max} such that, for any $\delta > 0$, $\epsilon \leq \frac{L^2}{\rho}$, $\Delta_f \geq f(\mathbf{x}_0) - f^*$, and constant $c \leq c_{max}$, PGD($\mathbf{x}_0$,L,ϵ,c,δ,Δ_f) outputs an ϵ- secondary order stationary point, with probability $1 - \delta$, and terminate in the following number of iterations [Jin 2017].:

$$O\left(\frac{L(f(\mathbf{x}_0) - f^*)}{\epsilon^2} log^4\left(\frac{dL\Delta_f}{\epsilon^2\delta}\right)\right)$$

This above result relies on characterizing the geometry around saddle points, these points constitute a thin band [Jin 2017].. Proving the above results goes as follows, a novel technique to bound the volume of this band is developed, as a result, it can be shown that after a random perturbation it is very unlikely to be in the band, therefore efficiently escaping saddle points.

3.2.1 Proof sketch of Theorem 1.

In order to prove Theorem 1, we need to show that PGD will not be stuck at a point which has a large gradient or near a saddle point. In the next lemma we prove that if current gradient is large, then we make progress in decreasing the objective function [Jin, 2017].

Lemma 1. Assume a function f that is L- gradient Lipschitz and ρ- Hessian Lipschitz, then for gradient descent with step-size $\eta < \frac{1}{L}$, we have:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{\eta}{2}||\nabla f(\mathbf{x}_t)||^2$$

Proof: By assuming the function is L- gradient Lipschitz and ρ- Hessian Lipschitz, we have:

$$\begin{aligned} f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^T(\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{L}{2}||\mathbf{x}_{t+1} - \mathbf{x}_t||^2 \\ &= f(\mathbf{x}_t) - \eta||\nabla f(\mathbf{x}_t)||^2 + \frac{\eta^2 L}{2}||\nabla f(\mathbf{x}_t)||^2 \\ &\leq f(\mathbf{x}_t) - \frac{\eta}{2}||\nabla f(\mathbf{x}_t)||^2 \end{aligned}$$

The next lemma says that if we are close to a saddle point, in other words, we are at a point where the gradient is small, but the Hessian has a large negative eigenvalue. This is the main part of our analysis, it shows that a perturbation followed by a small number of standard gradient descent steps can make the function value decrease with high probability.

Lemma 2. Assume a function f that is L- gradient Lipschitz and ρ- Hessian Lipschitz, then there exists an absolute constant c_{max} and any $c \leq c_{max}$, $\chi \geq 1$. Let η,r,g_{thres},t_{thres},f_{thres}, be calculated the same as in Algorithm 2, then if $\mathbf{m}_t$ satisfies:

$$||\nabla f(\mathbf{m}_t)|| \leq_{thres} \qquad and \qquad \lambda_{min}(\nabla^2 f(\mathbf{m}_t)) \leq -\sqrt{\rho\epsilon}$$

Let $\mathbf{x}_t = \mathbf{m}_t + \xi_t$ where ξ_t is sampled from a uniform distribution over $\mathbb{B}_0(r)$. Let $\mathbf{x}_{t+i}$ be the iterates of gradient descent from $\mathbf{x}_t$ with step-size η, then with a probability of at least $1 - \frac{dL}{\sqrt{\rho\epsilon}}e^{-\chi}$ we have:

$$f(\mathbf{x}_{t+t_{thres}}) - f(\mathbf{m}_t \leq -f_{thres}$$

A detailed proof of this lemma can be found in [Jin 2017]. We will use the above two lemmas to prove Theorem 1.

Theorem 1. Assume a function f that is L- gradient Lipschitz and ρ- Hessian Lipschitz, then there exists an absolute constant c_{max} such that, for any $\delta > 0$, $\epsilon \leq \frac{L^2}{\rho}$, $\Delta_f \geq f(\mathbf{x}_0) - f^*$, and constant $c \leq c_{max}$, PGD($\mathbf{x}_0$,L,ϵ,c,δ,Δ_f) outputs an ϵ- secondary order stationary point, with probability $1 - \delta$, and terminate in the following number of iterations [2]:

$$O\left(\frac{L(f(\mathbf{x}_0) - f^*)}{\epsilon^2} log^4\left(\frac{dL\Delta_f}{\epsilon^2\delta}\right)\right)$$

Proof: In this theorem we let $c_{max} = \min(C_{max}, \frac{1}{2})$, where C_{max} is the absolute allowed constant in Theorem 1. Also choose any constant $c \leq c_{max}$. In this proof we will achieve a point that satisfies :

$$||\nabla f(\mathbf{x})|| \leq_{thres} = \frac{\sqrt{c}}{\chi^2}.\epsilon \quad and \quad \lambda_{min}(\nabla^2 f(\mathbf{x})) \leq -\sqrt{\rho\epsilon} \tag{5}$$

Because $c \leq 1$ and≤ 1, we have $\frac{\sqrt{c}}{\chi^2} \leq 1$. This implies that any $\mathbf{x}$ satisfying Eq (5) is a ϵ-second order stationary point.

If we start with $\mathbf{x}_0$, then if $\mathbf{x}_0$ does not satisfy equation (5) then there are two possibilities only:

1. $||\nabla f(\mathbf{x})|| > g_{thres}$: In this case the algorithm will not add a perturbation. Using Lemma 1 we get:
$$f(\mathbf{x}_1) - f(\mathbf{x}_0) \leq -\frac{\eta}{2}.g_{thres}^2 = -\frac{c^2}{2\chi^4}.\frac{\epsilon^2}{L}$$

2. $||\nabla f(\mathbf{x})|| \leq g_{thres}$: In this case the algorithm will adds perturbation with radius r and then perform gradient descent for the net t_{thres} steps without adding a perturbation. If the termination condition is not met after this, then we must have:
$$f(\mathbf{x}_{thres}) - f(\mathbf{x}_0) \leq f_{thres} = -\frac{c}{\chi^3}.\sqrt{\frac{\epsilon^3}{\rho}}$$

   Which means on average, every step decreases the function value by :
$$\frac{f(\mathbf{x}_{thres}) - f(\mathbf{x}_0)}{t_{thres}} \leq -\frac{c^3}{\chi^4}.\frac{\epsilon^2}{L}$$

For case 1, we can repeat the same argument for $t = 1$ and for case 2 we can repeat the argument for $t = t_{thres}$. Therefore we can conclude that as long as Algorithm 2 has not yet terminated, every step, on avergae decreases the value of the objective function by at least $-\frac{c^3}{\chi^4}.\frac{\epsilon^2}{L}$. Also, we cannot decrease the function by more than the distance to the golbal minima, $f(\mathbf{x}_0) - f^*$, therefore Algorithm 2 will run for no longer than the following iterations:

$$\frac{f(\mathbf{x}_0) - f^*}{\frac{c^3}{\chi^4}.\frac{\epsilon^2}{L}} = O\left(\frac{L(f(\mathbf{x}_0) - f^*)}{\epsilon^2} log^4\left(\frac{dL\Delta_f}{\epsilon^2\delta}\right)\right)$$

We proved that the algorithm terminates, now we'll show that when it does, the point it finds is an ϵ- second order stationary point. We will show that every time a perturbation is added to the iterate $\mathbf{m}_t$, if $\lambda_{min}(\nabla^2 f(\mathbf{m}_t)) < -\sqrt{\rho\epsilon}$, then we get $f(\mathbf{x}_{t+t_{thres}}) - f(\mathbf{m}_t) \leq -f_{thres}.Thus, if the current point is not an \epsilon$- second order stationary point, the algorithm will not terminate.

In Algorithm 2, we immediatly know $||\nabla f(\mathbf{m}_t)|| \leq g_{thres}$ or a perturbation would not be added at time t. Therefore by Lemma 2, we know the probability of this event happening is at least $1 - \frac{dL}{\sqrt{\rho\epsilon}}e^{-\chi}$ each time. In one run of the entire algorithm, the number of times perturbations are added is at most:

$$\frac{1}{t_{thres}} \cdot \frac{\chi^4}{c^3} \cdot \frac{L(f(\mathbf{x}_0) - f^*)}{\epsilon^2} = \frac{\chi^3}{c} \cdot \frac{\sqrt{\rho\epsilon}(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \tag{6}$$

Lemma 2 is satisfies with high probability by union bound for all perturbations, which proves the correctness of the algorithm. This probability is at least:

$$1 - \frac{dL}{\sqrt{\rho\epsilon}}e^{-\chi} \cdot \frac{\chi^3}{c} \cdot \frac{\sqrt{\rho\epsilon}(f(\mathbf{x}_0) - f^*)}{\epsilon^2} = 1 - \frac{\chi^3 e^{-\chi}}{c} \cdot \frac{dL(f(\mathbf{x}_0) - f^*)}{c\epsilon^2}$$

From the algorithm, recall the our choice of $\chi \leftarrow 3\max\{\log(\frac{dl\nabla_f}{c\epsilon^2\delta}), 4\}$. Because $\chi \geq 12$, we have $\chi^3 e^{-\chi} \leq e^{-\frac{\chi}{3}}$, resulting in :

$$\frac{\chi^3 e^{-\chi}}{c} \cdot \frac{dL(f(\mathbf{x}_0) - f^*)}{c\epsilon^2} \leq e^{-\frac{\chi}{3}} \cdot \frac{dL(f(\mathbf{x}_0) - f^*)}{c\epsilon^2} \leq \delta$$

which ends the proof.

4 Conclusions

In this paper we present two algorithms that guarantee escaping saddle points and converging to a second order stationary point. It is assumed that saddle points are non-degenerate, and all second order stationary point are local minima. The first algorithm was the first work that provided global convergence guarantees for stochastic gradient descent. The number of iterations depended polynomially on the underlying dimension. The second algorithm extended this result and showed global convergence in a number of iterations that are independent of dimension up to poly-logarithmic terms.

References

Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Global optimality of local search for low rank matrix recovery. arXiv preprint arXiv:1605.07221, 2016.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Advances in Neural Information Processing Systems, pages 29332941, 2014.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle pointsonline stochastic gradient for tensor decomposition. In COLT, 2015.

Prateek Jain, Chi Jin, Sham M Kakade, and Praneeth Netrapalli. Computing matrix square-root via non convex local search. arXiv preprint arXiv:1507.05854, 2015.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michaek I. Jordan. How to escape saddle points efficiently. arXiv preprint arXiv:1703.00887, 2017.

Kenji Kawaguchi. Deep learning without poor local minima. In Advances In Neural Information Processing Systems, pages 586594, 2016.

Yu Nesterov. Introductory lectures on convex programming volume i: Basic course. Lecture notes, 1998.

Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. In Information Theory (ISIT), 2016 IEEE International Symposium on, pages 23792383. IEEE, 2016b.