UVA CS 6316: Machine Learning : 2019 Fall
Course Project: Deep2Reproduce @
https://github.com/qiyanjun/deep2reproduce/tree/master/2019Fall

ESCAPING SADDLES USING STOCHASTIC GRADIENT DESCENT

Reproduced by:
Archana Narayanan, Kamya mehul Desai, Daniel Choi, Yuancheng Lin

12/04/2019

http://proceedings.mlr.press/v80/daneshmand18a.html

Motivation

- Optimization of non-convex functions pose challenges due to the presence of Saddle Points and Suboptimal Local Minima.
- Existing work has proved convergence of GD by injecting explicit, isotropic noise to make GD escape saddle points.
- Since Isotropic noise exhibits variance in all the directions, the first order and second order time complexities become dimension dependent.

Background

- **Reaching a 1st order stationary point** :
  - ➢ First order stationary point can be reached much faster by SGD compared to GD.

- **Reaching a 2nd order stationary point**
  - ➢ Existing 1st order techniques add isotropic noise with a known variance.
  - ➢ GD methods are unlikely to get stuck, but if they do, adding noise allows them to escape saddle points *(Lee et al., 2016)*.

- **Using Curvature Information**
  - ➢ Since negative curvature signals potential descent directions, a 2nd order method can be applied to exploit this curvature direction to escape saddle points.
  - ➢ Does not guarantee global convergence and is locally attracted by saddle points and local maxima *(Dauphin et al., 2014)*.

Related Work

- Convergence of SGD with additional noise was analysed *(Ge et al., 2015)* but no prior work demonstrated SGD Convergence without explicitly adding noise.

- Using Curvature information, convergence to a second-order stationary point (Conn et al., 2000) was derived & has been shown to achieve the optimal worst case iteration bound *(Polyak, 2006)*.

- Sub-sampling the Hessian can reduce the dependence on n by using various sampling schemes *(Kohler & Lucchi, 2017; Xu et al., 2017)*.

- It was shown that noisy gradient updates act as a noisy Power method allowing to find a negative curvature direction using only first-order information *(Xu & Yang, 2017) and (Allen-Zhu & Li, 2017)*

Claim / Target Task

Claim :

- Injection of explicit, isotropic noise usually applied to make GD escape saddle points can successfully be replaced by a simple SGD step.
- Derive the first convergence rate for plain SGD to a second-order stationary point in a number of iterations that is independent of the problem dimension.

Target Task :

- Analyse the convergence of PGD and SGD for optimizing non convex function under a new assumption
- Correlated Negative Curvature (CNC) - requiring stochastic noise to exhibit variance along the directions of negative curvature. Thereby, removing dependency on dimensionality.

WHY Claim

- In almost all previous methods of escaping the saddle point involving adding noise to the GD process.
- This will add difficulties when dealing with a large dataset with high dimensionalities

Method	Perturbation	Noise	Opt. strategy
(1) Cubic Reg.	-	-	2nd-order
(2) PGD	$\triangle \mathbf{w} = \xi$	$\xi \sim B_r^d(0)$	1st-order
(3) NGD	$\triangle \mathbf{w} = -\eta_{\mathbf{w}} \nabla f(\mathbf{w}) + \xi$	$\xi \sim N(0, I)$	1st-order
(4) PSGD	$\triangle \mathbf{w} = -\nabla f_{\mathbf{z}_i}(\mathbf{w}) + \xi$	$\xi \sim N(0, I)$	stochastic 1st
(5) SGLD	$\triangle \mathbf{w} = -\eta_{\mathbf{w}} \nabla f_{\mathbf{z}_i}(\mathbf{w}) + \xi$	$\xi \sim N(0, I)$	stochastic 1st

(1) [Nesterov and Polyak, 2006]
(2) [Jin et al., 2017]
(3) [Levy, 2016]
(4) [Ge et al., 2015]
(5) [Zhang et al., 2017]

- A simple SGD step that can escape the saddle point will be extremely helpful for dataset with high dimensionalities.

Proposed Solution

1. **CNC PGD (Perturbed GD):**
   - The Gradient Descent under the CNC assumption (perturbed with SGD steps) performs better than GD perturbed with isotropic noise.

1. **CNC SGD:**
   - SGD without perturbations, escapes the saddle point faster due to the intrinsic noise generated in each iteration due to sampling.

Implementation

Replace the Perturbation(a) with a Stochastic Gradient Descent Step(b)

$$w_t = w_t + \zeta$$

$$w_t = w_t - r \nabla f_{z_i}(w_t)$$

Theory was proven by showing that given a stochastic gradient that meets the CNC assumption, replacing the isotropic noise with intrinsic noise from a SGD step returns a secondary stationary point with probability proportional to the number of steps

Data Summary

- MNIST dataset with 70,000 samples and 30 parameters was used
- 28x28 pixel black and white images of handwritten digits
- The training data consists of 60000 training samples and the testing data is 10000 samples

Experimental Results

1. SGD, GD, ISO-PGD and CNC-PGD are initialized close to a saddle-point with Gaussian input data and sigmoid loss.

1. Results show that the Stochastic Gradient Descent finds a negative curvature faster to escape the saddle points.

1. CNC SGD performs slightly better than CNC GD

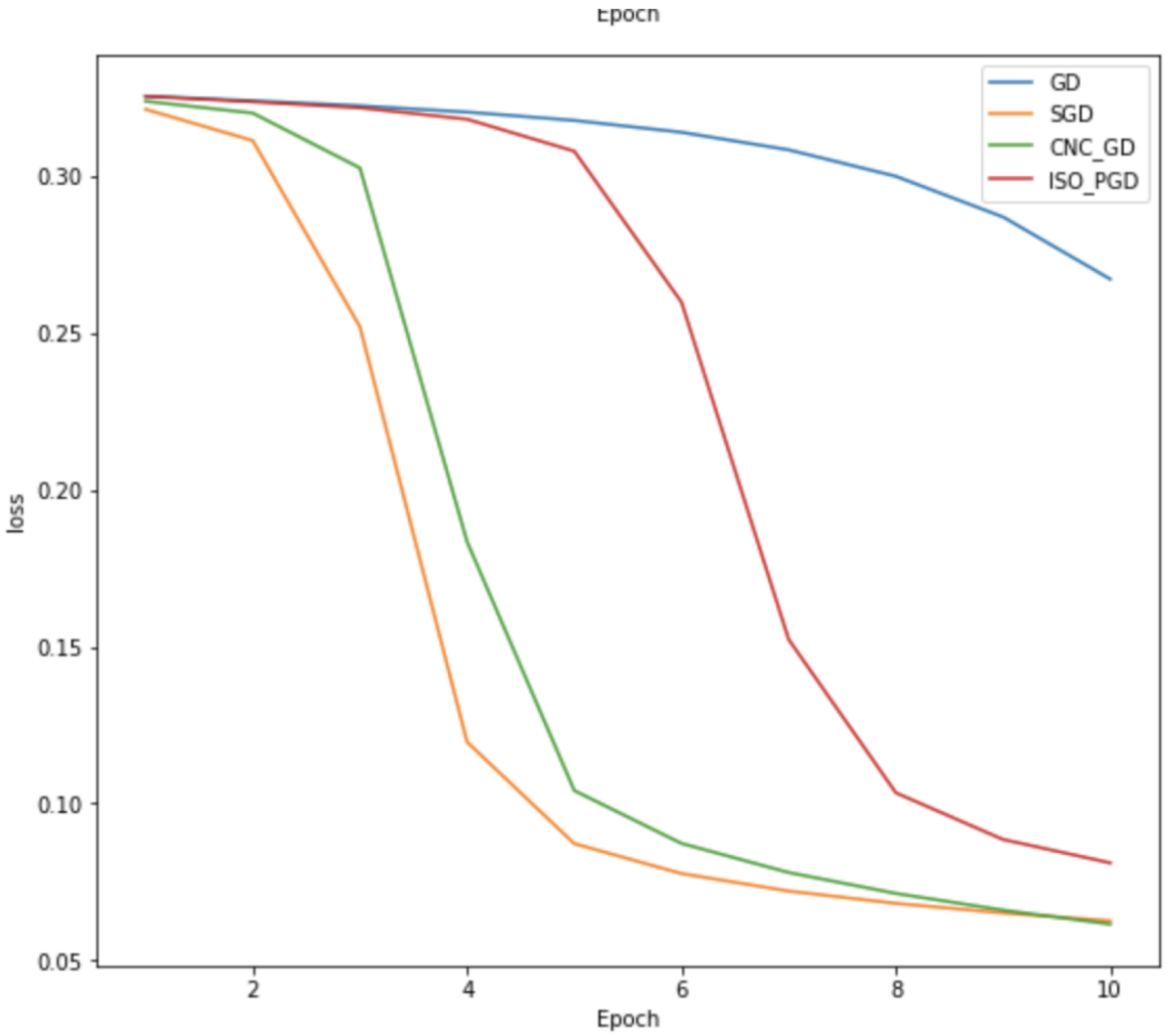
1. ISO PGD contains isotropic noise to escape the saddle point within finite iterations

Experimental Analysis

Based on the above Experimental Results, following are the implications:

1. Wide and deep neural networks can be trained by using Stochastic Gradient Descent while escaping the saddle points.
2. The bounds established through Lemma 4 can be used to train the Neural Networks. The regularization of optimization methods rely on the SGD since, the directions of large curvature correspond to the principal components of the data for many ML models.

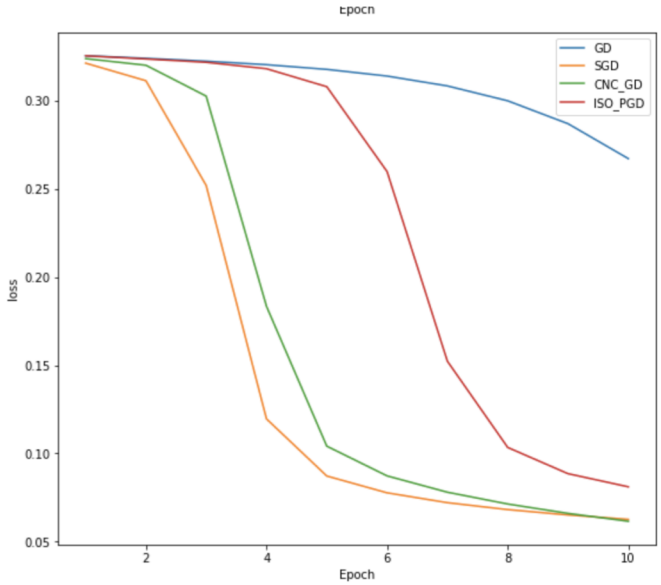
Reproduction Results

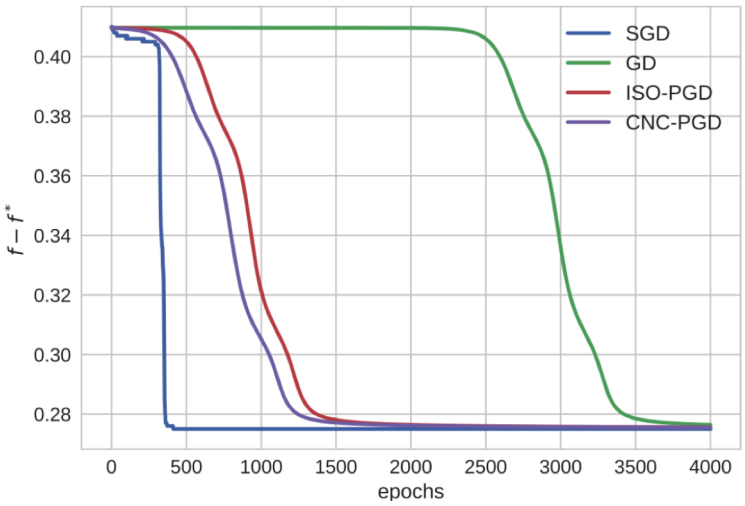


Reproduction

- Implemented algorithm of SGD, GD, CNC-PGD and ISO-PGD on a Keras library
  - Used a Feed-forward CNN

- Used the MNIST dataset





Conclusion and Future Work

- Convergence of PGD and SGD is analyzed for optimizing non-convex functions which makes use of stochastic noise to exhibit variations along the most negative curvature.
- CNC - GD and CNC SGD produce better results than GD with isotropic noise.
- The future works suggested in this paper is regularization and optimization of stochastic gradient methods.

Tasks for each team member

Daniel Choi

- code implementation
- Slides - Proposed Solution, Implementation, Data Summary

Kamya mehul Desai

- code implementation
- Slides -Experimental result, Experimental Analysis, Conclusion and Future Work

Yuancheng Lin

- code implementation
- Slides - Claim, Target Task, WHY Claim

Archana Narayanan

- code implementation
- Slides - Motivation, Background and Related Work

References

- https://arxiv.org/abs/1803.05999v2
- https://vimeo.com/312282771