Training Monotone Operator Equilibrium Network with Different Operator Splitting Methods

Zhuo Li
Key Lab of High Confidence Software Technologies
Peking University, China
lizhmq@pku.edu.cn

Abstract

Implicit-depth models such as Deep Equilibrium Networks have recently been attracting many researchers' interest, because they are shown to match or even exceed the performance of traditional deep networks while being much more memory efficient. Monotone Operator Equilibrium Networks (MON) further show that the process of finding the equilibrium point of an implicit network is equivalent to solving a form of monotone operator splitting problem, which admits efficient solvers with guaranteed, stable convergence. In this paper, we try several different monotone operator splitting algorithms and compare their performance on training MON on CIFAR10 dataset. We find that Peaceman-Rachford splitting is the best algorithm among the tested ones. Furthermore, we find that using LeakyReLU as activation function in MON leads to higher accuracy than ReLU. The code is availble at `https://github.com/Lizhmq/monotone_op_net`.

1 Introduction

Previous work (1; 4) has demonstrated the power of implicit-depth networks. In these neural networks, models are composed of infinite equivalent non-linear transformation layers, thus the forward computation is to find a fixed-point of the transformation function. Winston and Kolter (4) propose Monotone Operator Equilibrium Networks (MON) which convert the fixed-point problem to monotone operator splitting problem where efficient solvers are available and convergence is guaranteed.

Winston and Kolter (4) give the theoretical and practical foundations for training implicit-depth networks with monotone operator splitting methods. They have tried traditional operator splitting methods, such as forward-backward splitting and Peaceman-Rachford splitting. We replicate their work and further test the performance of Douglas-Rachford splitting algorithm and Proximal Decomposition algorithm (3). Douglas-Rachford splitting is a damped version of Peaceman-Rachford splitting and it always converges when $\mathbf{0} \in A(x) + B(x)$ has solution, while Peaceman-Rachford splitting needs C_A or C_B to be contraction. The main advantage of Proximal Decomposition Algorithm (PDA) is that it can easily be extended to inclusions with more than two operator (3). However, in MON (4), the implemented monotone operators are contractive and the problem is in two operators splitting formula. So Douglas-Rachford splitting method and PDA are not superior than Peaceman-Rachford splitting method in this implementation. In our experiments, Douglas-Rachford splitting method is faster than Proximal Decomposition. Peaceman-Rachford splitting is the fastest algoirhtm among the three methods. Furthermore, we find that using the LeakyReLU as the non-linear activation function leads to higher accuracy than ReLU.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

2 Monotone Operator Splitting for Fixed-Point Networks

Winston and Kolter (4) propose MON and test their performance on CIFAR10 dataset. In this paper, we replicate their work and try different operator splitting algorithms to train the implicit-depth network. In Section 2.1, we revisit the MON model proposed in (4). In Section 2.2, we will give a detailed description of our used operator splitting methods. In Section 2.3, we describe the activation functions we used as non-linear transformation.

2.1 Fixed-point networks as operator splitting

Considering the weight-tied, input-injected network in which $x \in \mathcal{R}^d$ denotes the input, and $z^k \in \mathcal{R}^n$ denotes the hidden units at layer k, given by the iteration

$$z^{k+1} = \sigma(Wz^k + Ux + b) \tag{1}$$

where $\sigma : \mathcal{R} \to \mathcal{R}$ is a nonlinearity applied elementwise, $W \in \mathcal{R}^{n \times n}$ are the hidden unit weights, $U \in \mathcal{R}^{n \times x}$ are the input-injection weights and $b \in \mathcal{R}^n$ is a bias term. The forward computation result of this implicit-depth network is a equilibrium point of Equation (1). According to MON (4), finding a fixed point of Equation (1) is equivalent to finding a zero point of the operator splitting problem $\mathbf{0} \in (A + B)(z^*)$ with the operators

$$A(z) = (I - W)(z) - (Ux + b), B = \partial f \tag{2}$$

and $\sigma(\cdot) = prox_f^1(\cdot)$ for some function f, where $prox_f^\alpha$ denotes the proximal operator

$$prox_f^\alpha(x) = \arg\min_z \frac{1}{2}\|x - z\|_2^2 + \alpha f(z). \tag{3}$$

Winston and Kolter (4) also provide the description of implementing a convolutional neural network with the above formula. Please refer to their paper for more details. In this work, we mainly focus on investigating the performance of different operator splitting methods on solving this problem.

2.2 Operator splitting methods

Given the MON formulation, we introduce how to solve the operator splitting problem in this section. A simple solution is to simply compute the forward iteration (Equation (1)) until convergence. However, it can be the case that the simple iteration does not converge. Instead, we can use the forward-backward iteration (Algorithm 1) which is guaranteed to converge when $\alpha \leq 2m/L^2$. Peaceman-Rachford algorithm (Algorithm 2) usually converges in fewer steps. Douglas-Rachford algorithm (Algorithm 3) is a dampled version of Peaceman-Rachford algorithm and it is more stable. However, in the case that both the two algorithms converges, Douglas-Rachford algorithm usually takes more steps. Another method we tried is Proximal Decomposition algorithm (PDA) (Algorithm 4). The main idea of PDA is that the two proximal steps used in Dougla-Rachford splitting can be performed in parallel. Readers can refer to the survey (3) for more details. We test the training efficiency of the above four operator splitting algorithms in our experiments.

Algorithm 1 Forward-backward iteration

$z := 0,\ err := 1$
repeat
  $z^+ := (1 - \alpha)z + \alpha(Wz + Ux + b)$
  $z^+ := \sigma(z^+)$
  $err := \frac{\|z^+ - z\|_2}{\|z^+\|_2}$
  $z := z^+$
until $err > \epsilon$
return z

Algorithm 2 Peaceman-Rachford iteration

$z := 0, err := 1; V := (I + \alpha(I - W))^{-1}$
repeat
  $u^{1/2} := 2z - u$
  $z^{1/2} := V(u^{1/2} + \alpha(Ux + b))$
  $u^+ := 2z^{1/2} - u^{1/2}$
  $z^+ = \sigma(u^+)$
  $err := \frac{\|z^+ - z\|_2}{\|z^+\|_2}$
  $z, u := z^+, u^+$
until $err > \epsilon$
return z

Algorithm 3 Douglas-Rachford iteration

$z := 0, err := 1; V := (I+\alpha(I-W))^{-1}$
repeat
  $u^{1/2} := 2z - u$
  $z^{1/2} := V(u^{1/2} + \alpha(Ux + b))$
  $u^+ := u + z^{1/2} - z$
  $z^+ = \sigma(u^+)$
  $err := \frac{\|z^+ - z\|_2}{\|z^+\|_2}$
  $z, u := z^+, u^+$
until $err > \epsilon$
return z

Algorithm 4 Proximal Decomposition algorithm

$x_1 := 0,\ y_1 := 0,\ x_2 := 0,\ y_2 := 0,\ err := 1;\ V := (I + \alpha(I - W))^{-1}$
repeat
  $z := x_1 + y_1$
  $x_1^{1/2} := V(x_1 + y_1 + \alpha(Ux + b))$
  $y_1^{1/2} := x_1 - x_1^{1/2} + y1$
  $x_2^{1/2} := \sigma(x_2 + y_2)$
  $y_2^{1/2} := x_2 - x_2^{1/2} + y_2$
  $x_1 := \frac{1}{2}(x_1^{1/2} + x_2^{1/2})$
  $x_2 := x_1$
  $y_1 := \frac{1}{2}(y_1^{1/2} + y_2^{1/2})$
  $y_2 := -y_1$
  $z^+ := x_1 + y_1$
  $err := \frac{\|z^+ - z\|_2}{\|z^+\|_2}$
  $z := z^+$
until $err > \epsilon$
return z

2.3 Activation function

As described in MON (4), the activation function σ can be any non-linear function which can be write as the proximal operator of a function f. In empirical studies, the performance of LeakyReLU is usually better than ReLU (5), and LeakyReLU can also be written as a proximal operator of simple expression (2). So we further test the accuracy of MON using LeakyReLU as activation function.

3 Experimental Results

We evaluate the training time and model accuracy on CIFAR10 dataset. The model configurations are shown in Table 1.

Efficiency We first test the training efficiency of the four operator splitting algorithms on the Multi-tier network without data augmentation. The training time of one epoch is listed in Table 2. (The four methods lead to similar model accuracy, so the results are not listed here.) From the table, we conclude that Peaceman-Rachford splitting algorithm is the best method for this problem. Douglas-Rachford algorithm and Proximal Decomposition algorithm are faster than forward-backward iteration method.

Table 1: Model hyperparameters

	CIFAR-10			
	Single conv	Multi-tier	Single conv lg.	Multi-tier lg.
Num. channels	81	(16,32,60)	200	(64,128,128)
Num. params	172,218	170,194	853,612	1,014546
Epochs	40	40	50	50
Initial lr	0.001	0.01	0.001	0.001
Lr schedule	step decay	step decay	1-cycle	1-cycle
Lr decay steps	25	10	-	-
Lr decay factor	10	10	-	-
Max learning rate	-	-	0.01	0.05
Data augmentation	-	-	✓	✓

Accuracy We test the accuracy of four models listed in Table 1 and compare the performance of using default ReLU activation and LeakyReLU activation. All the models are trained under the

Table 2: Training time per epoch of Multi-tier network

	Forward-Backward	Peaceman	Douglas	ProxDecomp
Time/s	202.97	**87.33**	129.57	140.82

same settings with Peaceman-Rachford splitting method. The results are shown in Table 3. The performance of models with LeakyReLU activation is better than those with ReLU activation. (In the originial paper, they train models with data augmentation for 65 epochs while we only train 50 epochs, so our results are a little lower than theirs.)

Table 3: Model accuracy (%)

Non-linearity	Single conv	Multi-tier	Single conv lg.	Multi-tier lg.
ReLU	74.10	72.14	81.40	88.62
LeakyReLU	**74.13**	**72.70**	**81.60**	**88.87**

4 Discussions

We intend to use different operator splitting methods to improve the training efficiency of MON. However, in our experiments we find that the original used Peaceman-Rachford algorithm is the fastest one. We further change the activation function to LeakyReLU and find it helps to improve the model accuracy a little. (Overall, well, I think this task is a little hard for me...)

To replicate our work, please refer to the attached GitHub repository address. The code is based on the original repository provided by Winston and Kolter (4).

References

[1] S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 688–699, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/01386bd6d8e091c2ab4c7c7de644d37b-Abstract.html.

[2] A. Bibi, B. Ghanem, V. Koltun, and R. Ranftl. Deep layers as stochastic solvers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryxxCiRqYX.

[3] A. Lenoir and P. Mahey. A survey on operator splitting and decomposition of convex programs. *RAIRO Oper. Res.*, 51(1):17–41, 2017. doi: 10.1051/ro/2015065. URL https://doi.org/10.1051/ro/2015065.

[4] E. Winston and J. Z. Kolter. Monotone operator equilibrium networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/798d1c2813cbdf8bcdb388db0e32d496-Abstract.html.

[5] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015. URL http://arxiv.org/abs/1505.00853.