Designing network architecture from differential equation view

Ge Yan

May 25, 2019

Abstract

Deep neural networks are powerful models used in many computer vision and natural language processing applications. By designing better network architecture, we can achieve better performance on given task. In this paper, we connect network design with numerical odinary differential equations. By simulating a popular numerical differential equations algorithnm, Adams, we propose a new architechture and test its performance on CIFAR-10 dataset.

1 Introduction

Deep neural networks are state of the art models in many aspects, including computer vision and natural language processing. However, training deep networks is a difficult and expensive task. To address this problem, many network structures are proposed to make training easier and imporve generalization of model.One of those models is Resnet proposed by He et al. [1]. They introduce skip connection between layers to avoid gradient vanishing ,allowing training of deeper network.Resnet is very effective and show great performance in diffirent computer vison tasks.

However, it is still a tough task to understand why skip connection helps training in Resnet. A lot of work has been done trying to explain it.E. [4] proposed a possible explaination by connecting neural networks with dynamical systems.He suggests that each module in Resnet can be regarded as one step on Euler discretization in solving differential equation.That is,

$$y_{n+1} = y_n + f(y_n, \theta_n).$$

This observation inspired many attempts to combine network design and numerical ordinary differential equation algorithmns. Lu et al. [5] suggests that many network ,such as Densenet [3], Polynet [11], can be viewed as different numerical discretization of differential equations. [5] combines neural networks with linear multistep method and proposed LM-Resnet.

In this paper, we construct a network simulating the famous numerical ODE algorithm, Adams. We experiment it on CIFAR-10 dataset and compared its performance between fixed coefficients between layers and learable coefficients.

2 Related works

Following the idea that recognizing Resnet module as one step in Euler discretization, Lu et al. [5] suggest that many variants of Resnet can be viewed as diffirent numerical scheme of ODE.They proposed LMnet, designing network by following formula:

$$y_{n+1} = (1 - k_n)y_n + k_n y_{n-1} + f(y_n, \theta_n).$$

Later, more attempts are made to design network architechture using ODE numerical method.Zhu et al. [6] suggest that Runge-Kutta method can be utilized in network designing.Runge-Kutta method is a famous numerical method in ODE solving.Generally, it can be written as [7]:

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i z_i$$

in which

$$z_i = f(t_n + c_i h, y_n + g \sum_{j=1}^{s} a_{ij} z_j).$$

3 Adams and Adamnet

In this paper, we design an architechture following Adams algorithm and making experiments with it.Adams is a famous numerical ODE algorithnm.Generally, its numerical scheme can be written as following formula:

$$y_{n+1} = y_n + \sum_{s=1}^{k} \beta_s f(y_{n-s}, t_{n-s}),$$

where k is a positive interger.Algorithmn following this formula is called k-step explicit Adams method.To construct a similar architechture in neural network, we use

$$y_{n+1} = y_n + \sum_{s=1}^{k} \beta_s f(\theta_{n-s}, y_{n-s})$$

where y_i is the output of ith layer, f_i represents the ith Resnet module [2],which can be written briefly as BatchNormalization(BN) [8]-relu-conv-BN-relu-conv.Other parts are the same as Resnet [1]

4 Experiment and results

We test two kinds of model: k=2 and k=3 on CIFAR-10 dataset with data augmentation in [9]. The network structure is mainly the same as pre-activate Resnet [2] except adding extra skip connections according to Adams method. Our training stategy is similar with [5]:Training by SGD with momentum of 0.9 and weight dacay of 0.0001.Batch size is 128.The whole training takes 160 epochs.Learning rate is initialized 0.1 and divided by 10 at epoch 80 and 120.Different with [5] , when choosing β_i ,we takes two diffirent stategy:

(A) Set β_i as trainable parameters and initialized it by random numble sampling from $\beta_1 : U(1, 1.1)(k = 2)$ and $\beta_1 : U(1, 1.1)$ $\beta_2 : U(-1, -0.9)(k = 3)$

(B) Fix β_i as constant given by Adams method:$\beta_1 = 1.5(k = 2)$ and $\beta_1 = 23/12, \beta_2 = -4/3(k = 3)$

We compares the results on CIFAR-10 with Resnet [1] and LMResnet [5]in Table 1.(A) stands for using learnable β_i and (B) stands for fixed β_i .

5 Conclusion and Discussion

In this paper, we design a network called Adamnet according to Adams method and compares its performance with Resnet [1] and LM-ResneLu2018BeyondFLNNt on CIFAR-10 dataset.From the results on Table-1, we can see that Adamnet performs slightly better when network is shallow and slightly worse when network is deep.We also observe an interesting phenomenon:fixed β_i usually makes the result slightly worse than trainable ones, which is consistent with out intuiation because it has less trainable parameters.However, it outperforms ones with trainable β_i when network depth is 20.

In the future , we will further investigate the link between neural network and dynamical system ,try to give a explaination to this phenomenon.

Model	Layers	Error
Resnet	20	8.75
Resnet	32	7.51
Resnet	44	7.17
Resnet	56	6.97
LMResnet	20	8.33
LMResnet	32	7.18
LMResnet	44	6.66
LMResnet	56	6.31
Adamnet(k=2)(A)	20	8.15
Adamnet(k=2)(A)	32	7.31
Adamnet(k=2)(A)	44	6.75
Adamnet(k=2)(A)	56	6.44
Adamnet(k=2)(B)	20	7.81
Adamnet(k=2)(B)	32	7.49
Adamnet(k=2)(B)	44	6.90
Adamnet(k=2)(B)	56	6.71
Adamnet(k=3)(A)	20	8.27
Adamnet(k=3)(A)	32	7.39
Adamnet(k=3)(A)	44	7.02
Adamnet(k=3)(A)	56	6.36
Adamnet(k=3)(B)	20	8.02
Adamnet(k=3)(B)	32	7.54
Adamnet(k=3)(B)	44	7.14
Adamnet(k=3)(B)	56	6.97

Table 1: Results on CIFAR-10 dataset

References

[1] He Kaiming et al. “Deep Residual Learning for Image Recognition.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)*: 770-778.

[2] He Kaiming et al. “Identity Mappings in Deep Residual Networks.” *ECCV (2016).*

[3] Huang, Gao et al. “Densely Connected Convolutional Networks.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)*: 2261-2269.

[4] Weinan ,E. "A Proposal on Machine Learning via Dynamical Systems." *Communications in Mathematics & Statistics 5.1(2017)*:1-11.

[5] Lu, Yiping et al. “Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations.” *ICLR (2018).*

[6] Mai Zhu et al. “Convolutional Neural Networks combined with Runge-Kutta Methods” *arxiv 1802.08831*

[7] Endre Sli et al. “An Introduction to Numerical Analysis” Cambridge University Press, 2003.

[8] Ioffe, Sergey and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” *ICML (2015).*

[9] Chen-Yu Lee et al. “Deeplysupervised nets” In Artificial Intelligence and Statistics, pp. 562–570, 2015.

[10] Chen, Tian Qi et al. “Neural Ordinary Differential Equations.” *NeurIPS (2018).*

[11] Zhang, Xingcheng et al. “PolyNet: A Pursuit of Structural Diversity in Very Deep Networks.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)*: 3900-3908.