Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230709

The advance of neural ordinary differential ordinary differential equations

Haoxuan Li

College of science, Xi’an Jiaotong-liverpool University, Suzhou, 215000, China

Haoxuan.Li19@student.xjtlu.edu.cn

Abstract. Differential methods are widely used to describe complex continuous processes. The main idea of ordinary differential equations is to treat a specific type of neural network as a discrete equation. Therefore, the differential equation solver can be used to optimize the solution process of the neural network. Compared with the conventional neural network solution, the solution process of the neural ordinary differential equation has the advantages of high storage efficiency and adaptive calculation. This paper first gives a brief review of the residual network (ResNet) and the relationship of ResNet to neural ordinary differential equations. Besides, his paper list three advantages of neural ordinary differential equations compared with ResNet and introduce the class of Deep Neural Network (DNN) models that can be seen as numerical discretization of neural ordinary differential equations (N-ODEs). Furthermore, this paper analyzes a defect of neural ordinary differential equations that do not appear in the traditional deep neural network. Finally, this paper demonstrates how to analyze ResNet with neural ordinary differential equations and shows the main application of neural ordinary differential equations (Neural-ODEs).

Keywords: Neural-ODEs, ResNet, dynamic system, DNN.

1. Introduction

The brain is the seat of human analysis, association, memory, and logical thinking. It is a sophisticated biological network of billions of intricately linked neurons. Synaptic connections allow neurons to transmit information to one another, and as neurons become more and less connected as they learn, the learned information is stored in these connections. To give the machine a certain amount of intelligence, the artificial network model, made up of neurons that mimic the fundamental information processing and storage unit of human brain, exhibits intelligent features such as self-organization and self-learning. The neural network's calculating architecture and learning principles are modeled after biological brain networks. In a digital computer, the process that nerve cells receive stimulation from neighboring cells and produce corresponding output signals can be simulated by "linear weighted sum" and "function mapping". At the same time "optimization learning algorithm" realizes the process of adjusting the network structure and weights. Although the bionic intelligent computing model established in this way can't be completely equivalent and comparable to the biological neural network, it has achieved superior performance in some aspects. Neural networks can be classified as feed-forward or feedback networks depending on how their neurons link. Neural networks may be categorized as continuous and discrete networks, deterministic and random networks, static and dynamic networks, and more based on their network type. Neural networks are split into supervised learning and unsupervised learning by learning preferences. Currently, neural networks have been developed, and hundreds of models have been successfully applied in fields such as handwriting recognition, saliency detection, speech recognition, image recognition, pattern recognition, human-computer interaction, optimization algorithm, and deep learning [1].

One significant factor that influence the performance of models is the depth of the neural network. When we increase the number of layers, more complex feature patterns can be extracted by the network. Therefore, theoretically, the deeper the model is, the better results can be obtained. However, with the increase of layers of neural networks, the probability of gradient disappearance or explosion of deep neural networks will increase, and the Degradation problem of neural networks will become a challenge. As the depth or neural network increasethe, the performance of the network will not be better and even be worse. To solve this problem, HE et al. proposed ResNet. Unlike the common neural network, ResNet uses short-circuit structure in each two layers, to learn residual information. It has an important design principle that keeps the complexity of the network layer, i.e. when the feature map size is reduced by half, the number of feature maps will double. ResNet can support more than 100 layers of deep neural networks for training thanks to the residual learning mechanism [2].

In 2018, a unique deep neural network is proposed. Instead of inserting various types of neural network blocks or adjusting the parameters of neural networks, ordinary differential equations connect one certain layer in neural networks with a fixed time point t of the dynamic flow of ordinary differential equations. This brilliant map explores a field in the intersection of neural networks and dynamic systems, which is a potential candidate for explaining neural networks. Neural-ODEs are used as a model component to design new models to do irregular time series modeling, density estimation and supervised learning. These new models can adjust their evaluation strategies according to each input and automately adapt the requirement of accuracy.

This paper tries to answer three questions: What are the advantages of Neural differential equations compared to ResNet? Can all deep neural network models be seen as neural ordinary differential equations? Can neural ordinary differential equations be used to explain neural ordinary differential equations?

The structure of this paper is as follows: Section 2 gives a brief review of ResNet. Section 3 bridges ResNet to neural ordinary differential equations. Section 4 lists three advantages of neural ordinary differential equations compared with ResNet. Section 5 introduces: 1) the class of DNN models that can be in- interpreted by numerical discretization of neural ordinary differential equations. 2) analysis a defect of neural ordinary differential equations that do not appear in the traditional deep neural network. and demonstrates how to analyze ResNet with neural ordinary differential equations. 3) shows the main application of neural ordinary differential.

2. Residual network

ResNet is a milestone in CNN development, which solved degeneration in Neural Networks [3]. For traditional convolutional neural network, the flow from layer n to layer $n+1$ is

$$h_{n+1} = f(h_n) \tag{1}$$

where h_n denote the input vector of layer n and f is a linear or non-linear function, depending on the model. Although Neural network areas share a common sense that adding more layers can increase accuracy, these models experimentally found that the more layer is added, the less accurate they have, called degeneration. ResNet introduces an additional block called residual block to solve this problem. If the residual function $f(x) = 0$, the output is just x, i.e. input data. This structure insures that the next layer cannot be worse than this layer so in general a deep convolutional neural network cannot be worse than shallow neural netwok [4]. Then the flow from layer n to $n+1$ is:

$$h_{n+1} = h_n + f(h_n) \tag{2}$$

Normally f is a non-linear function which combined a linear weighted function and an activation function like ReLU. Note that in original ResNet, there is an activation function on h_{n+1} to get the output of layer $n+1$, but the author of ResNet then found that the form above is esaier to compute [4].

The ResNet model uses a VGG structure, or the technique of replacing a single large convolution kernel with a number of very small convolution kernels. This technique reduces the parameters of

ResNet model and increases the number of its nonlinear activation functions, which decline the calculation size of the ResNet model. The number of filters is constant for the convolution layer when both the input and output feature maps are the same size. The number of filters doubles when the feature map is halved in size, and the feature map's downsampling step is set to two. Convolution layers are mostly where ResNet models with various depths differ from one another. The unit blocks of the ResNet model with depth (i.e., convolution layer number more than or equal to 50) employ bottleneck design to shorten training time. In addition, a 1x1 convolution layer is put in front and behind the 3x3 convolution layer. The major goal is to make the feature graph smaller such that the 3x3 convolution layer has a minimal output and input size limitation. The bottleneck network building component can speed up the effectiveness of model training while also increasing model size and time complexity.

Setting the size of the batch training, the learning rate, the number of categories, and the weight attenuation rate are the primary super parameters that need to be specified when training the ResNet model. The direction of the ResNet model's decrease depends on the batch training size selection. The size of batch training should be suitably adjusted when the data set is large enough, which can significantly minimize the calculation volume. The batch training should be adjusted to a higher number to lessen the influence of noise data if the amount of data is little and there is noise data. The ResNet model performs best in terms of training time and convergence accuracy when batch training reaches a particular threshold.

3. Neural ordinary differential equations

Researchers frequently build deep neural networks for learning that are time-space dependent using residual networks while studying time series models. The cost of memory and parameter size will, however, unavoidably rise as the number of remaining blocks increases. As a result, this is unable to balance memory usage, making it exceedingly challenging to implement the time model and model performance in real-world applications (such as intelligent management of urban traffic flow and intelligent dispatch of industrialized hydropower flow). It is required to use a dynamic system model with low memory use, adaptive computing capabilities, and a balance between forecast speed and accuracy to overcome this challenge.

In the existing literature, Weinan et al. established the relationship between deep neural network and stochastic dynamic system earlier [5]. For instance, the ultra-deep neural network ResNets may be used to interpret the answer using the Euler technique, however the accuracy of doing so is not very great. The other type of neural ordinary differential equation put forth by Chenl et al. in 2018 uses an adjoint method rather than the conventional back propagation algorithm to calculate the gradient in a memory-efficient manner [6]. This method directly models the dynamics of the hidden state of the network under the condition of continuous time.

Neural-ODEs based on ordinary differential equations provides numerous improvements over earlier dynamic systems. 1) Neural-ODEs may lower the memory consumption cost of the model from $\mathrm{O(LN)}$ to $\mathrm{O(L)}$ by using the so-called adjoint approach, where L stands for the number of layers of ODE or ResNet in the model and N stands for the total number of ResNet subnetworks or time steps in the integration time of ODE. 2) Different models based on neural ordinary differential equations may be successfully applied to real-world applications by adapting the number of time steps in ODE to the trade-off between prediction accuracy and computation cost. 3) Because the parameters of an ODE network are reusable, the total number of parameters for the model may be significantly decreased (to around 1/6 of that of a residual network), which also decreases training time and improves resilience. 4) The neural ordinary differential equation network, which functions as a decoder in the generation model of time series and has strong extrapolation capabilities, may also be employed as a generation model. Because only the hidden state of the most recent time step needs to be stored and the intermediate value (activation function value) in the forward calculation process does not, the calculation complexity is dependent on the numerical method chosen, allowing for a trade-off between speed and accuracy.We will next go into great depth about how neural ordinary differential equations are implemented.

The idea bridging ResNet and ODEs is quite simple. Consider a general ODE:

$$\frac{\partial h}{\partial t} = f(h) \tag{3}$$

Computer usually solve it with numerical method. Euler method is a simple one:

$$h_{n+1} = h_n + \eta f(h) \tag{4}$$

Where η is step size. Let $\eta = 1$, this is the same with flow formula of ResNet. Therefore, ResNet can seen as Euler discretization of ODEs with unit step size. The output of layer n then become the state t_n with respect to function h [7, 8].

4. Advantage of neural ordinary differential

Less memory cost. Consider a simple CNN with a linear weight function, and sigmoid activates the function. The error contributed only depends on the activated input and output data. This implies that the memory cost of training the neural network would increase sharply with layer and neural increase. On the other hand, Neural-ODEs use the adjoint sensitivity method to shrink this cost, which is a linear function concerning the problem size [4]. Because the adjoint system not only describes the process of the initial dynamic system's process and the derivative state of each point in the reverse process through the chain rule, but it is also through the adjoint system that the model can get the initial state of the differential, and similarly, get a parameter of a function ("residual block" or Euler method discretization process) that describes the dynamic system.

Flexible control. Flexibility means a trade-off between efficiency and accuracy. To solve ODEs with numerical methods, the step size and tolerance can be chosen randomly. If high accuracy is required, then the step size can be set to be small, which leads to an increase in on-time cost. By contrast, one can set a large step size to find the result in a limited time, so the accuracy is relatively low [4].

Guide for improvement. Based on the idea of a numerical method for solving ODEs, Lu Y et al. proposed a new neural network block structure called linear multi-architecture [9]. LM-ResNet analyzes ResNet from the point of view of the numerical differential equation, improves the single-step method corresponding to the multi-step method, experimentally demonstrates higher accuracy than the original ResNet.

5. Discussion

Some DNN models. Precvious work already shown that ResNet is the discretization of ODEs, but how about another neural network? By now, several neural networks correspond to the numerical method of solving ODE, which includes PolyNet (corresponds to the backward Euler method), FractalNet (correspond to the Runge-Kutta method), and RevNet (corresponds to the forward Euler method). On the other hand, recurrent neural network [9]. Furthermore, in training data sets with Neural models, there is a simple class of function that cannot be reached that traditional DNN can. Therefore, despite N-ODEs having several advantages to training data, the trainer should carefully identify the type of function. There is an augmented neural network, inserting N-ODEs block into traditional DNN, similar to the ResNet block [10]. However, unfortunately, at present, there is not much work on combining neural networks based on attention mechanisms with neural ordinary differential equations. The representative work of the attention mechanism transformer has made breakthrough progress in many fields. However, residual learning is not used because it does not depend on the deep network structure. However, its large-scale parameters and the consumption of computing resources require solutions similar to neural ordinary differential equations.

Neural-ODEs as an analysis tool. ResNet is shown to be the forward Euler discretization of N-ODEs, but this is not enough to analyze ResNet by Neural-ODEs Although it can go from Neural-ODEs to ResNet with certain step size, it is hard to guarantee that the deep limit of ResNet is converged to Neural-ODEs. This is the work by Matthew T and Yves G, who prove the gamma-convergence of deep limit ResNet. Therefore, if the corresponding N-ODE is stable, ResNet is also stable [11].

Latent ODEs. A recurrent neural network is a well-known neural network processing data with time information. However, this requires assuming that the data are collected in equal duration, which is unrealistic in the real world. One neural network that produces continuous time flow is introduced to break this limitation: neural ordinary differential equations [12].

6. Conclusion

In conclusion, N-ODEs use less memory while training and can be controlled flexibly. In addition, the idea of N-ODEs can guide researchers to improve the original ResNet. However, despite there is the number of advantages. N-ODEs have just a few neural networks that can be interpreted this way, and it has a limited function class that can be reached. It is a combination of neural network and dynamic system, but it does not have the method to explain why neural network works so well. It can still be used as a tool to analyze the stability of ResNet. However. N-ODEs are now used primarily to model irregular time series as an extension of recurrent neural network, which have great potential in the financial market.

References

[1] Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. Computer Science Review, 40, 100379.

[2] Xue, Z., Yu, X., Liu, B., Tan, X., & Wei, X. (2021). HResNetAM: Hierarchical residual network with attention mechanism for hyperspectral image classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 3566-3580.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning. Image Recognition, 7.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[5] Weinan, E. (2017). A proposal on machine learning via dynamical systems. Communications in Mathematics and Statistics, 1(5), 1-11.

[6] Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. Advances in neural information processing systems, 31.

[7] Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. Advances in neural information processing systems, 31.

[8] Weinan, E. (2017). A proposal on machine learning via dynamical systems. Communications in Mathematics and Statistics, 1(5), 1-11.

[9] Lu, Y., Zhong, A., Li, Q., & Dong, B. (2018, July). Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In International Conference on Machine Learning (pp. 3276-3285). PMLR.

[10] Dupont, E., Doucet, A., & Teh, Y. W. (2019). Augmented neural odes. Advances in Neural Information Processing Systems, 32.

[11] Thorpe, M., & van Gennip, Y. (2018). Deep limits of residual neural networks. arXiv preprint arXiv:1810.11741.

[12] Rubanova, Y., Chen, R. T., & Duvenaud, D. (1907). Latent odes for irregularly-sampled time series (2019). arXiv preprint arXiv:1907.03907.