Adjoint Method in Neural ODE and Continuous Backpropagation

Boeva Galina, group M05-304a

October 11, 2024

1 Introduction

In recent years, Neural Ordinary Differential Equations (Neural ODEs) have become a popular tool in the field of machine learning and artificial intelligence. They allow for the modeling of continuous processes and provide new opportunities for training deep neural networks. The Adjoint Method and Continuous Backpropagation play a crucial role in the optimization and training of such models. In this essay, we will explore these methods, their mathematical descriptions, and their interconnections, as well as their impact on the training process of Neural ODEs.

2 Mathematical Description and Connections

2.1 Neural Ordinary Differential Equations (Neural ODEs)

Neural ODEs represent a model where the state of the system is described by an ordinary differential equation. Let $\mathbf{z}(t)$ be the state of the system at time t, and $f_\theta(\mathbf{z}(t), t)$ be a function depending on parameters θ. Then, the dynamics of the system are described by the following equation:

$$\frac{d\mathbf{z}(t)}{dt} = f_\theta(\mathbf{z}(t), t)$$

To solve this equation, numerical integration methods such as the Runge-Kutta method are used.

2.2 Adjoint Method

The Adjoint Method is used for the efficient computation of gradients of the loss function in Neural ODEs. The main idea of the method is to introduce an adjoint equation that describes the dynamics of the gradients. Let $L(\mathbf{z}(T))$ be the loss function depending on the final state of the system $\mathbf{z}(T)$. Then, the adjoint equation is given by:

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^\top \frac{\partial f_\theta(\mathbf{z}(t), t)}{\partial \mathbf{z}}$$

where $\mathbf{a}(t)$ is the adjoint state, and $\frac{\partial f_\theta(\mathbf{z}(t),t)}{\partial \mathbf{z}}$ is the Jacobian of the function f_θ with respect to the state $\mathbf{z}$.

The gradients of the parameters θ can be computed as follows:

$$\frac{dL}{d\theta} = \int_0^T \mathbf{a}(t)^\top \frac{\partial f_\theta(\mathbf{z}(t),t)}{\partial \theta} dt$$

2.3 Derivation of the Adjoint Method

To derive the Adjoint Method, consider the loss function $L(\mathbf{z}(T))$. We want to find the gradient of this function with respect to the parameters θ. Let $\mathbf{z}(t)$ be the solution to the equation $\frac{d\mathbf{z}(t)}{dt} = f_\theta(\mathbf{z}(t),t)$. Then:

$$\frac{dL}{d\theta} = \frac{\partial L}{\partial \mathbf{z}(T)} \frac{\partial \mathbf{z}(T)}{\partial \theta}$$

To compute $\frac{\partial \mathbf{z}(T)}{\partial \theta}$, we use the chain rule:

$$\frac{\partial \mathbf{z}(T)}{\partial \theta} = \int_0^T \frac{\partial \mathbf{z}(t)}{\partial \theta} \frac{\partial f_\theta(\mathbf{z}(t),t)}{\partial \theta} dt$$

Now, introduce the adjoint state $\mathbf{a}(t)$ that satisfies the equation:

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^\top \frac{\partial f_\theta(\mathbf{z}(t),t)}{\partial \mathbf{z}}$$

with the initial condition $\mathbf{a}(T) = \frac{\partial L}{\partial \mathbf{z}(T)}$.

Then, the gradient of the loss function with respect to the parameters θ can be expressed as:

$$\frac{dL}{d\theta} = \int_0^T \mathbf{a}(t)^\top \frac{\partial f_\theta(\mathbf{z}(t),t)}{\partial \theta} dt$$

2.4 Continuous Backpropagation

Continuous Backpropagation is an extension of the classical backpropagation algorithm for continuous systems. In the context of Neural ODEs, Continuous Backpropagation can be described as follows. Let $L(t)$ be the loss function at time t, and $\theta(t)$ be the vector of parameters of the neural network. Then, the dynamics of the parameter updates are described by the differential equation:

$$\frac{d\theta(t)}{dt} = -\eta \nabla_\theta L(t)$$

where η is the learning rate, and $\nabla_\theta L(t)$ is the gradient of the loss function with respect to the parameters θ.

2.5 Example of Continuous Backpropagation on a More Complex Function

Consider a more complex loss function $L(\theta) = \frac{1}{2}\theta^4$. Let $\theta(t)$ be the parameter we want to minimize. Then, the dynamics of the parameter updates are described by the following equation:

$$\frac{d\theta(t)}{dt} = -\eta \nabla_\theta L(t) = -\eta \cdot 2\theta(t)^3$$

The solution to this equation is:

$$\theta(t) = \theta(0) \left(1 + 6\eta\theta(0)^2 t\right)^{-\frac{1}{2}}$$

where $\theta(0)$ is the initial value of the parameter. It can be seen that the parameter θ decreases over time, corresponding to the minimization of the loss function.

3 Interconnections Between Methods

The Adjoint Method and Continuous Backpropagation are closely related. Both methods use the gradients of the loss function to update the parameters, but they do so in different contexts. The Adjoint Method focuses on computing gradients through the introduction of an adjoint equation, while Continuous Backpropagation views the training process as a continuous differential equation.

The interconnection between these methods can be seen in the fact that both use the gradient of the loss function to update the parameters. In the Adjoint Method, the gradient is used to determine the direction of search, while in Continuous Backpropagation, the gradient is used to determine the rate of change of the parameters.

4 Question: Is Time T Real Time?

In the context of Neural ODEs and optimization methods such as the Adjoint Method and Continuous Backpropagation, time T is not necessarily real time. Instead, T represents a parameter that defines the length of the integration interval or the time horizon over which the system dynamics are considered. This parameter can be chosen arbitrarily depending on the task and may not correspond to real time.

For example, in tasks such as time series prediction or physical process modeling, time T may correspond to real time. However, in other tasks such as image classification or text processing, time T may be an abstract parameter that defines the depth or complexity of the model.

Thus, time T in Neural ODEs and related optimization methods is a flexible parameter that can be tuned depending on the specific task and does not necessarily correspond to real time.

5 Conclusion

The Adjoint Method and Continuous Backpropagation are two powerful tools for the optimization and training of Neural ODEs. Both methods use the gradient of the loss function to update the parameters, but they do so in different contexts: the Adjoint Method focuses on computing gradients through the introduction of an adjoint equation, while Continuous Backpropagation views the training process as a continuous differential equation. Understanding and utilizing these methods allows for significant improvements in the efficiency and accuracy of machine learning models based on Neural ODEs.