Notes on Neural ODEs

Xavier Garcia

April 15, 2019

The purpose of these notes is to understand some of the details in "Neural Ordinary Differential Equations".

1 Introduction to Neural ODEs

For the purposes of these notes, we will only consider neural networks that come in blocks, where the input and out of every block have the same dimension. Moreover, we assume that every block has a residual connection to the one before it and they all share the same weights. In particular, if f denotes the function which takes input $z_t \in \mathbb{R}^n$ from the tth block, and sends it to the $(t+1)$-th, then we can write the output z_{t+1} as

$$z_{t+1} = z_t + f(z_t, \theta) \tag{1}$$

where θ are the parameters of the block. We can give a different interpretation of this operation through classical mechanics. Let's consider z_0 as the position of a particle in space and z_t to be its position in space after t units of time. We can think of $f(z_0, \theta)$ as a velocity vector which we can add to our position z_0 to compute the next position. Therefore, z_1, following (1), would be the new position of our particle at z_0 after 1 unit of time following the velocity vector $f(z_0, \theta)$. Suppose, however, that instead of following this vector for 1 unit of time, we instead followed it for ϵ units of time, for some $\epsilon > 0$. With this representation, we can compute the new position $z_{t+\epsilon}$ from its last position z_t by the formula:

$$z_{t+\epsilon} = z_t + \epsilon f(z_t, \theta)$$

In particular, if we take $\epsilon \to 0$, we obtain:

$$\frac{dz_t}{dt} = \lim_{\epsilon \to 0} \frac{z_{t+\epsilon} - z_t}{\epsilon} = \lim_{\epsilon \to 0} f(z_t, \theta) = f(z_t, \theta)$$

For notational clarity, I will write $\dot{z} := \frac{dz_t}{dt}$ and hide the subscript whenever the context is obvious. In particular, I'll write $z := z_t :=:= z_t(x) := z(t, \theta, x)$ where x denotes the initial condition i.e. $z_0 = x$. If we assume that our network has T blocks, then in the limit, we can predict the output of a point x as the value of z_T, where z solves the ordinary differential equation:

$$\dot{z}_t = f(z, t, \theta)$$
$$z_0 = x$$

2 The Adjoint Method

In this section, we'll discuss how to effectively train Neural ODEs. We assume we have some loss function $L : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, where $L(z_T, y)$ computes the error from approximating y by z_T, where y is the desired output for input $z_0 := x$. Following the paper's notation, I'll write $L(z_T, y)$ as $L(z)$, hiding away the dependance on y and treating it as a function of a single variable. If we can compute $\frac{d}{d\theta}L(z)$, then we can perform gradient descent to figure out the best value of θ. A naive computation of the gradient would require computing $\frac{\partial z}{\partial \theta}$. This is problematic since this means we would need to differentiate with respect to operations within our ODE solver. To avoid this problem, we need to find an expression for the gradient of $L(z)$ that doesn't involve this term. We can solve this problem by invoking the adjoint method which I will describe below.

We begin by first introducing three auxilliary functions. First, we can rewrite $L(z)$ as follows:

$$L(z) = \int_0^T l(z, t, \theta)dt + L(z_0, y)$$

where we define $l(z, t, \theta)$ by the following equation:

$$l(z, t, \theta) := \frac{d}{dt}L(z_t, y) = \nabla L^\dagger(z_t)\dot{z}_t = \nabla L^\dagger(z_t)f(z_t, t, \theta)$$

where we use the $\dagger$ notation to denote tranposition. The second auxilliary function is $\lambda : \mathbb{R}_+ \to \mathbb{R}^n$ which we will define explicitly later on. Our final auxilliary function will be $\mathcal{L}$, which we define as follows:

$$\mathcal{L}(z, \dot{z}, \theta) := \int_0^T l(z, t, \theta)dt + \int_0^T \lambda^\dagger(\dot{z} - f(z, t, \theta)dt. \tag{2}$$

Notice if z satisfies the ODE from before then

$$\nabla_\theta \mathcal{L} = \nabla_\theta L$$

for any function λ. This gives us the freedom to tinker with λ and for the right choice of λ, we can remove the dependence on $\frac{\partial z}{\partial \theta}$ by the following theorem:

Theorem 2.1. *Suppose that function a_t satisfies the following ODE:*

$$\dot{a}_t = -a_t^\dagger \frac{\partial f}{\partial z}$$
$$a_T = \nabla_z L(z_T)$$

We can write $\nabla_\theta L(z)$ *as*

$$\nabla_\theta L(z) = \int_0^T a_t^\dagger \frac{\partial f}{\partial \theta} dt$$

Proof. We begin by differentiating $\mathcal{L}$.

$$\frac{d\mathcal{L}}{d\theta} = \int_0^T \nabla_z l^\dagger \frac{\partial z}{\partial \theta} + \nabla_\theta l \, dt + \int_0^T \lambda^\dagger \frac{\partial \dot{z}}{\partial \theta} - \lambda^\dagger \frac{\partial f}{\partial z}\frac{\partial z}{\partial \theta} - \lambda^\dagger \frac{\partial f}{\partial \theta} dt$$

We first collect all terms with a factor $\frac{\partial z}{\partial \theta}$. Unfortunately, one these has a time derivative. To fix that, we use integration by parts:

$$\int_0^T \lambda^\dagger \frac{\partial \dot{z}}{\partial \theta} dt = \lambda_T^\dagger \frac{\partial z_T}{\partial \theta} - \lambda_0^\dagger \frac{\partial z_0}{\partial \theta} - \int_0^T \dot{\lambda}^\dagger \frac{\partial z_t}{\partial \theta} dt$$

Since z_0 doesn't depend on θ, the second term on the right vanishes. Next, we collect terms with a factor of $\frac{\partial z}{\partial \theta}$ and rewrite our gradient:

$$\frac{d\mathcal{L}}{d\theta} = \int_0^T \nabla_\theta l \, dt + \int_0^T \left(\nabla_z l - \dot{\lambda}^\dagger - \lambda^\dagger \frac{\partial f}{\partial z} \right) \frac{\partial z}{\partial \theta} dt - \int_0^T \lambda^\dagger \frac{\partial f}{\partial \theta} dt + \lambda_T^\dagger \frac{\partial z_T}{\partial \theta} \quad (3)$$

Since our choice of λ is flexible, we choose it so that it satisfies the following ODE:

$$\begin{aligned} \dot{\lambda}_t^\dagger &= \nabla_z L - \lambda^\dagger \frac{\partial f}{\partial z} \\ \lambda_T^\dagger &= 0 \end{aligned}$$

This choice of λ makes the second integral the right of Equation 3 vanish. The last term also vanishes due to the initial conditions of this ODE. Next, we write out $\frac{\partial l}{\partial \theta}$ as follows:

$$\nabla_\theta l = \nabla_z L^\dagger \frac{\partial f}{\partial \theta}$$

By factoring $\frac{\partial f}{\partial \theta}$ from the remaining terms, we obtain:

$$\nabla_\theta \mathcal{L} = \int_0^T \left(\nabla_z L^\dagger - \lambda^\dagger \right) \frac{\partial f}{\partial \theta} dt$$

We conclude the proof by noting that if we define $a := \nabla_z L - \lambda$, then a satisfies the ODE from the statement of the theorem. By uniqueness of solutions to ODEs, we obtain the desired result.

□

3 Continuous-time normalizing flows

The advantages of Neural ODEs extend beyond traditional supervised learning tasks. One natural place to employ neural ODEs is normalizing flows. In this setting, we attempt to model the distribution of some random variable X by viewing it as the image of standard Gaussian $Z \sim \mathcal{N}(0, I)$ under a bijective smooth function F. Traditionally, we view f as being a composition of several simpler functions, $f_1, ..., f_m$. We can approach this problem in a similar fashion to the supervised task by forcing all the f_i to share the same weights and thus they are equivalent to a single function f. We can then endow f with the velocity vector interpretation from before and thus model the distribution X as the distribution z_T where z solves the ODE

$$\begin{aligned} \dot{z} &= f(z, t) \\ z_0 &= Z \end{aligned}$$

In order for this to be as useful as the standard normalizing flow, we need to be able to compute the density p_{z_T}. At this point, the standard approach would require computing the determinant of the Jacobian of $z_T(Z)$ as a function of Z. This is a costly operation. Instead, we can proceed as before and find a suitable differentiable equation which could reduce the cost of the computation.

Theorem 3.1. *Suppose $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is Lipschitz in the first coordinate, and continuous in the second one. Define $z_t(x)$ as the solution of the following differential equation:*

$$\begin{aligned} \dot{z} &= f(z, t) \\ z_0 &= x \end{aligned}$$

Suppose that Z is some random variable with density p_{z_0}. Then $z_t(Z)$ has a density p_{z_t} and the quantity $\log p_{z_t}(z_t(x))$ satisfies the following ODE:

$$\frac{d \log p_{z_t}(z_t(x))}{dt} = -\mathrm{Tr} \frac{df}{dz}(z_t(x), t)$$

where Tr *is the trace operator.*

Proof. By the change of variable formula, we can write the quantity $\log p_{z_t}(z_t(x))$ as

$$\log p_{z_t}(z_t(x)) = \log p_{z_0}(x) + \log \left| \det \frac{dz_t^{-1}}{dx}(z_t(x)) \right|.$$

where z_t^{-1} is the inverse of $z_t(x)$ as a function of x. By Jacobi's formula for differentiating determinants, we can compute the time derivative of $\log p_{z_t}(z_t(x))$ and obtain the following formula:

$$\frac{d}{dt} \log p_{z_t}(z_t(x)) = \mathrm{Tr} \left(\frac{\mathrm{adj}(A)}{\det A} \dot{A} \right)$$

where $A = \frac{dz_t^{-1}}{dx}(z_t(x))$ and $\text{adj}(A)$ is the adjoint matrix of A. Next, we make three useful observations. We first point out that the first term within the trace operator is actually the inverse of A:

$$\frac{\text{adj}(A)}{\det A} = A^{-1}$$

Next, notice that it is a little awkward to be differentiating the inverse z_t^{-1} as opposed to z_t. It is convenient to note that

$$\frac{dz_t^{-1}}{dx}(z_t(x)) = \left(\frac{dz_t}{dx}(x)\right)^{-1}.$$

In particular, this allows us to compute the time derivative as follows:

$$\begin{aligned}\dot{A} &= -\left(\frac{dz_t}{dx}(x)\right)^{-1} \cdot \frac{d\dot{z}_t}{dx}(x) \cdot \left(\frac{dz_t}{dx}(x)\right)^{-1} \\ &= -A \cdot \frac{d\dot{z}_t}{dx}(x) \cdot A\end{aligned}$$

The last observation we'll need is that the quantity $\frac{dz_t(x)}{dx}$ also satisfies an ODE, obtained by differentiating the ODE for z_t with respect to x:

$$\begin{aligned}\frac{d\dot{z}_t}{dx} &= \frac{\partial f}{\partial z}(z_t(x), t)\frac{dz_t}{dx} \\ \frac{dz_0}{dx} &= I\end{aligned}$$

Finally, by using all three observations, we can proceed as follows:

$$\begin{aligned}\frac{d}{dt}\log p_{z_t}(z_t(x)) &= \text{Tr}\left(\frac{\text{adj}(A)}{\det A}\dot{A}\right) \\ &= -\text{Tr}\left(A^{-1} \cdot A \cdot \frac{d\dot{z}_t}{dx} \cdot A\right) \text{(by observation 1 and 2)} \\ &= -\text{Tr}\left(\frac{\partial f}{\partial z}A^{-1}A\right) \text{(by observation 3 and since } \frac{dz_t}{dx} = A^{-1}\text{)} \\ &= -\text{Tr}\frac{\partial f}{\partial z}(z_t(x), t)\end{aligned}$$

□