Notes: A Continuous Model of Neural Networks
Part I: Residual Networks

Zhen Li * Zuoqiang Shi †

August 22, 2017

Abstract

In this series of notes, we try to model neural networks as discretizations of continuous flows on the space of data, which can be called flow model. The idea comes from an observation of their similarity in mathematical structures. This conceptual analogy has not been proven useful yet, but it seems interesting to explore.

In this part, we start with a linear transport equation (with nonlinear transport velocity field) and obtain a class of residual type neural networks. If the transport velocity field has a special form, the obtained network is found similar to the original ResNet [He et al. 2016]. This neural network can be regarded as a discretization of the continuous flow defined by the transport flow.

In the end, a summary of the correspondence between neural networks and transport equations is presented, followed by some general discussions.

1 Transport Equation

Consider the following terminal value problem (TVP) for linear transport equation:

\[
\begin{aligned}
\partial_t u + v(t, x) \cdot \nabla u &= 0, \quad x \in \mathbb{R}^d, t \in [0, T] \\
u(T, x) &= f(x), \quad x \in \mathbb{R}^d.
\end{aligned}
\] (1)

Here \(v\) is a \(\mathbb{R}^d\)-valued function, called the transport velocity field. It can be chosen in different ways. We will consider firstly the general form, then a special type:

\[v(t, x) = W^{(2)}(t) a \left( W^{(1)}(t) + b^{(1)}(t) \right) + b^{(2)}(t),\] (2)

where \(W^{(1)}(t), W^{(2)}(t) \in \mathbb{R}^{d \times d}, b^{(1)}(t), b^{(2)}(t) \in \mathbb{R}^d\). The activation \(a\) is a \(\mathbb{R}^d\)-valued nonlinear function, which is Lipschitz continuous.
It is well known that the solution of equation (1) is transported along characteristics. The characteristics are defined as solutions of the initial value problems (IVP) of the ODE:

\[
\begin{align*}
\dot{x} &= v(t, x), \quad t \in [0, T] \\
x(0) &= x_0,
\end{align*}
\] (3)

where \(x_0 \in \mathbb{R}^d\). Along the solution curve \(x = q(t)\), it is easy to verify that

\[
\frac{d}{dt} u(t, q(t)) = (\partial_t u(t, x) + \dot{q}(t) \cdot \nabla u(t, x))_{x=q(t)}
\] (4)

\[
= (\partial_t u(t, x) + v(t, q(t)) \cdot \nabla u(t, x))_{x=q(t)} = 0.
\] (5)

So \(u\) remains unchanged along the curve. See Figure 1 for a conceptual illustration. Therefore

\[
u(0, x_0) = u(t, q(t)) = u(T, q(T)) = f(q(T)).\] (6)

Figure 1: Illustration of characteristics. Here \(x, u(t, x) \in \mathbb{R}\).

## 2 Method of Characteristics

In this part we will use the method of characteristics to solve (1). In order to make the following approximations reasonable, we assume that the change of \(v(t, x)\) with \(t\) and \(x\) is regular enough. Especially, we assume that the solution of (1) and (3) exist for the posed conditions, and they are regular enough.

Let \(\{t_k\}_{k=0}^L\) with \(t_0 = 0\) and \(t_L = T\) be a partition of \([0, T] \subset \mathbb{R}\) such that for any \(k = 1, \ldots, L\), \(s_k = t_k - t_{k-1}\) is small enough. Let \(x = q(t)\) be a characteristic of the transport equation (1), i.e. the solution of (3), and denote \(x_k = q(t_k)\). Denote \(V_k(x) = v(t_k, x)\) and \(u_k(x) = u(t_k, x)\) for any \(x \in \mathbb{R}\). See Figure 2 for a illustration of the discretization.
Near time $t_k$, the ODE (3) is approximately
\[ \dot{x} = V_k(x_k) \approx V_k(x_{k-1}). \] (7)

Use Euler method to integrate this ODE from $t_{k-1}$ to $t_k$, we get
\[ x_k \approx x_{k-1} + \int_{t_{k-1}}^{t_k} V_k(x_{k-1})dt \]
\[ \approx x_{k-1} + s_k V_k(x_{k-1}) \]
\[ = (\text{id} + s_k V_k)(x_{k-1}). \] (9)
\[ \approx x_{k-1} + s_k V_k(x_{k-1}) \]
\[ = (\text{id} + s_k V_k)(x_{k-1}). \] (10)

Therefore
\[ x_L = (\text{id} + s_L V_L)(x_{L-1}) \]
\[ = (\text{id} + s_L V_L) \circ \cdots \circ (\text{id} + s_1 V_1)(x_0) \] (11)
\[ = (\text{id} + s_L V_L) \circ \cdots \circ (\text{id} + s_1 V_1)(x_0) \]
\[ = (\text{id} + s_L V_L) \circ \cdots \circ (\text{id} + s_1 V_1)(x_0) \] (12)

If the terminal value function of $u$ is given as $u_L = f$, we might be able to use (12) to get the initial value $u_0$. According to (6),
\[ u_0(x_0) = u_L(x_L) = f \circ (\text{id} + s_L V_L) \circ \cdots \circ (\text{id} + s_1 V_1)(x_0) \] (13)

3 Neural Network Representation

3.1 General form

The discrete solution (13) of the terminal value problem of transport equation (1) is valid for any $x_0 \in \mathbb{R}^d$. Its basic structure is shown in Figure 3. This structure reminds us of the ResNet (He et al., 2016), but it is merely a formal one. In order to see the actual structure, we need to specify the definition of $V_k$'s.

Figure 3: Basic structure of a general ResNet. Notice that $\{skV_k\}_{k=1}^L$ are generally nonlinear functions of the input.
3.2 A Special Type

In order to get a ResNet with explicit 2-layer block, consider the special type of transport velocity field given by (2). Denote
\begin{align*}
W_k^{(1)} &= W^{(1)}(t_k), \quad b_k^{(1)} = b^{(1)}(t_k), \\
W_k^{(2)} &= W^{(2)}(t_k), \quad b_k^{(2)} = b^{(2)}(t_k), \\
W_k &= s_k W_k^{(2)}, \quad \tilde{b}_k^{(2)} = s_k b_k^{(2)}.
\end{align*}
(14)  \tag{15}  \tag{16}
By using the method of characteristics as before, we can get
\begin{equation}
x_k = x_{k-1} + \overline{W}_k^{(2)} a \left( W_k^{(1)} x_{k-1} + b_k^{(1)} \right) + \overline{b}_k^{(2)}.
\end{equation}
(17)

It generates a 2-layer ResNet block, which is much more like the original ResNet. Figure 4 illustrates its basic structure.

![Basic structure of the 2-layer ResNet block.]

At a first glance, it appears that directly define the transport velocity as (2) is not natural. But it is actually reasonable. The inner weights $W^{(1)}$ and $b^{(1)}$ are used to adjust the distribution of the transport velocity field according to the location of input. Notice that the activation $a$ is usually non-negative, or even restricted to $[0, 1]$. Thus the outer weights $W^{(2)}$ and $b^{(2)}$ are needed to adjust the direction and magnitude of the transport velocity. Both inner weights and outer weights are necessary ingredients of the velocity field.

3.3 Discussions

The ResNets obtained here are special. Firstly, as we can see in (10) and (17), due to the time step $s_k$, the residual term can be made sufficiently small comparing with the leading term $x_k$. This is a necessary condition for the ResNet to be modeled by transport equation.

Secondly, the weights of the ResNet changes slowly from block to block. More specifically, the weights on the same positions of adjacent ResNet blocks should be close to each other, because they are assumed to be discretizations of continuous functions of time. For example, $W_k^{(1)}$ is close to $W_{k-1}^{(1)}$, $W_k^{(2)}$ is close to $W_{k-1}^{(2)}$, and so on.

\footnote{One concern is that the layers of this ResNet are of the same width. This may not be a series problem in theory, because they can always be embedded in to the same high dimensional space.}
3.4 Summary

Clearly, there is a correspondence between the ResNets (10) (17) and the transport equation (1). This correspondence also exists in plain nets (neural networks without residual shortcuts). It is summarized in Table 1 and illustrated in Figure 5.

Table 1: Correspondence between neural network and transport equation

| Neural Network | Transport Equation |
|---------------|-------------------|
| layer $k$     | time $t_k$        |
| weights $\overline{W}_k$, bias $\overline{b}_k$, activation $a$ | transport velocity field $v(t, x)$ |
| output function $f$ | terminal value function $u(T, \cdot) = f$ |
| prediction map $F$ | initial value map $u(0, \cdot)$ |
| label $y$ | initial value $u(0, X) = F(X)$ |
| feedforward | solving IVP of characteristic equation |
| prediction | solving TVP of transport equation |
| supervised learning | solving inverse problem |

Figure 5: Correspondence between neural network and transport equation.

4 Discussions

**Claim:** The discussions here are not intended to provide solution to any problem in theory and applications of neural networks, nor be able to answer any concern. They only present our ideas about some of candidate directions that could be explored.
1. In the viewpoint of solving terminal value problem of transport equation, we can also consider other numerical methods intentionally designed for PDEs, besides of the method of characteristics. For example, we can increase regularity of solutions by adding dissipative terms to the transport equation.

2. The training of neural networks could be considered as solving inverse problem of transport equation. It means that both initial value (on samples) and terminal value are posed. The task is to find a transport velocity field (depending on time) that transports the initial value to the terminal value. It will be made clearer in the rest parts of this series of notes.

3. The correspondence provides one way to partially explain why deep is good for neural networks. Usually, in order to learn a good prediction map, the dataset needs to be distorted significantly. But the distortion provided by each layer is very limited. So it needs more layers to accomplish the total distortion. In the language of differential equations, deep means fine partition and small time step size. It allows the discretization to be more smooth and regular, such that each layer makes only a small progress. At the same time, the initial and terminal conditions can get a better match through the solution, hence the learnt prediction map is more accurate and can generalize better.

Acknowledgement

Zhen Li would like to show his gratitude to the support of professors Yuan Yao and Yang Wang from the Department of Mathematics, HKUST.

References

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on.