ResNet

f : neural network

$$x_{t+1} = x_t + f(x_t, \theta_t)$$

$t \in \{0, 1, \cdots, T\}$. (discrete)

$\Delta t \to 0$

$$\frac{dx(t)}{dt} = f(x(t), t, \theta)$$

ODE

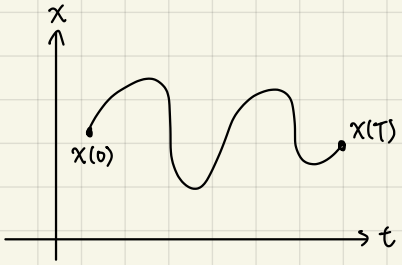
t : time

ODE : $x'(t) = f(t, x(t))$.

Euler's method

$\Rightarrow$ $x_{n+1} = x_n + hf(t_n, x(t_n))$, h : step size

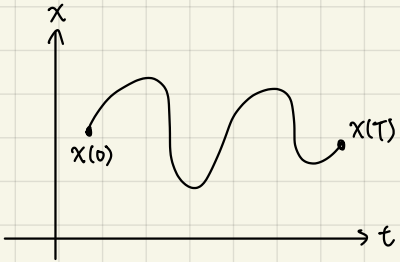
Define $x(0)$: input layer, $x(T)$: output layer, solution of ODE



neural network $\longleftrightarrow$ ordinary differential equations

image $\longrightarrow$ label

initial state $\longrightarrow$ output state



L : loss

$$L(x(t_1)) = L\left(x(t_0) + \int_{t_0}^{t_1} f(x(t), t, \theta)\, dt\right)$$

$\Downarrow$

find $\underset{\theta}{\operatorname{argmin}} \int_{t_0}^{t_1} f(x(t), t, \theta)\, dt$

instead of backwardpropagation

Lecture 2 simply extends the results from "Neural Ordinary Differential Equations"

We shall in the sequel take an unusual (but familiar to many researchers) point of view towards deep neural networks. Consider a controlled ordinary differential equation (ODE), i.e.

$$(*) \qquad dX_t = \sum_{i=1}^{d} V_i(X_{t-}) du^i(t),\ X_0 = x$$

where V_i are some smooth vector fields on $\mathbb{R}^m$ with all proper derivatives bounded. u is a finite variation curve with values in $\mathbb{R}^d$. For simplicity we shall always consider finitely many jumps of u in this lecture, and u being C^∞ between two jumps.

A solution of this equation is given by a fixed point of

$$X_t = x + \sum_{i=1}^{d} \int_0^t V_i(X_{s-}) du^i(s)$$

for $t \geq 0$. Notice that the integral is well defined due to the finite variation property.

In this case the Euler scheme, which approximates the solution of the ODE, is an iterative procedure where maps of the type

$$x \mapsto x + \sum_{i=1}^{d} V_i(x) \Delta u^i(t)$$

at point $x \in \mathbb{R}^N$ and $t \geq 0$ are concatenated. Obviously one can understand such finite concatenations as deep neural networks with at most $m + N$ neurons in each layers and depths being equal to the time discretization with respect to the ODE's initial value.

> We therefore call the map $x \mapsto X_t$ a deep neural network of infinite depths. Notice that for piecewise constant u we just obtain classical neural networks of finite depths.

For the third aspect consider a training data set $(x_l, y_l)_{l \in L}$, i.e. a subset of a graph of an $\mathbb{R}^m$-valued continuous function on some compact set. The cardinality of L should be seen as large but finite.

Consider the following ODE on $\mathbb{R}^{mL}$

$$dZ_t = \sum_{i=1}^{d} W_i(Z_{t-}) du^i(t)\,,$$

where

$$(W_i((x_l)))_l := V_i(x_l)$$

for $l \in L$ and $i = 1, \ldots, d$. We actually make a *particle system* of L particles moving precisely according to the dynamics of $(*)$. This system also has a unique solution for all times just as equation $(*)$.

Representability of the non-linear function on the training set amounts precisely to saying that, for given $t > 0$, the particle system Z can possibly reach at t any point (y_l) if starting at (x_l) for some choice of control u.

Chow's theorem

If the Lie brackets of (W_i) generate at the point (x_l) the whole space $\mathbb{R}^{mL}$, then any point (y_l) can be reached locally. Such systems are called exactly controllable.

The actual interpretation (which is another universal approximation theorem) is the following: a generic choice of vector fields V_i of the above controlled system can be realized by random neural networks and a certain number of controls $u^1, \ldots, u^d$ leads to an infinite depth neural network with $m + N$ nodes on each layer which can approximate any continuous function on a compact set.

Let us sum up the previous considerations:

1. We can understand deep neural networks as flows of controlled ordinary differential equations.
2. Backpropagation appears as the classical forward-backward calculation of first variations.
3. Variations with respect to the control can also be calculated in a forward-backward manner.
4. Since gradients can be efficiently calculated we can set up search algorithms stochastic gradient descent, simulated annealing, etc.