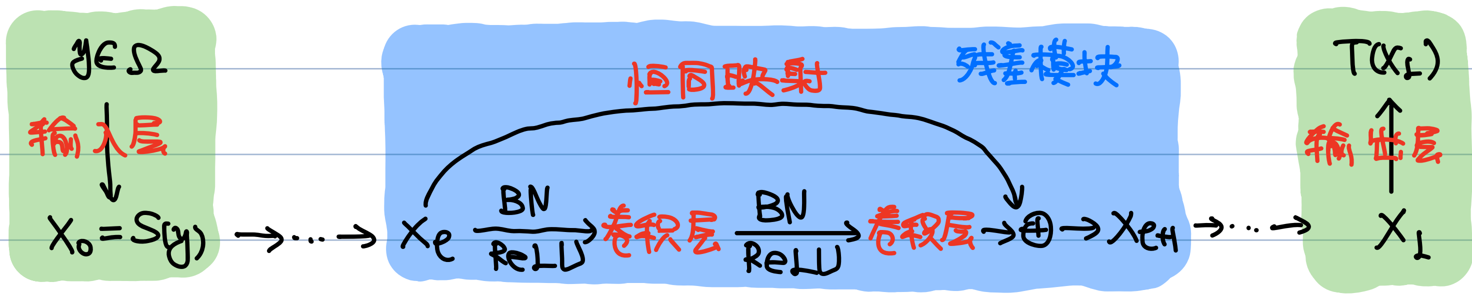
神经微分方程

孙琪

1. 残差网络

残差网络(ResNet)是深度学习中里程碑式的模型架构。不妨考虑分类学习问题，即给定输入数据及其标签 $\{y, h(y)\}_{y\in\Omega}$，残差网络的前向传播过程如下：



在得到神经网络对输入数据 $y\in\Omega$ 的分类预测 $T(x_L)$ 后，便可将其与真实标签 $h(y)$ 进行比较，从而得到损失函数（例如交叉熵）

$$\varphi(x_L) = \mathbb{E}_{y\in\Omega}\left[\| T(x_L) - h(y) \|\right]$$

不同于卷积层的简单堆叠，残差模块中引入了恒同映射

$$x_{\ell+1} = x_\ell + F(x_\ell, W_\ell), \quad 0\le \ell \le L-1$$

其中 W_ℓ 代表模型参数，从而有效解决了梯度爆炸的问题。

综上所述，用残差网络做分类任务的优化问题可写为

$$\underset{\{W_\ell\}_{\ell=0}^{L}}{\operatorname{argmin}} \left\{ \varphi(x_L) \;\middle|\; x_0 = S(y);\ x_{\ell+1} = x_\ell + F(x_\ell, W_\ell),\ 0\le \ell \le L-1 \right\}$$

其中模型参数的更新采用经典的反向传播算法，即

$$W_\ell \longleftarrow W_\ell - \eta \frac{\partial \varphi(x_L)}{\partial x_{\ell+1}} \frac{\partial x_{\ell+1}}{\partial W_\ell}, \quad 0\le \ell \le L-1.$$

其中 $\eta>0$ 代表学习率。

2. 神经微分方程

若将残差模块看作是常微分方程的前向 Euler 离散格式，则其相应的优化问题可写为：

$$\underset{w(t)}{\operatorname{argmin}} \left\{ \varphi(x(1)) \;\middle|\; x(0)=S(y);\ \frac{dx(t)}{dt}=f(t,x(t),w(t)),\ 0<t\le 1 \right\}$$

在科学计算领域，上述问题可写为更一般的形式

损失泛函

$$\text{minimize}\quad J[w(t)] = \Psi(x(t_1),t_1) + \int_{t_0}^{t_1} L(t,x(t),w(t))\,dt$$

$$\text{subject to}\quad x(t_0)=x_0\in\mathbb{R}^{m\times 1};\quad \frac{dx(t)}{dt}=f(t,x(t),w(t)),\quad t_0<t\le t_1$$

状态变量　控制变量

为了求解上述带约束的最优化问题，引入 Lagrange 乘子

伴随变量 ← $p(t)=[p_1(t),\ p_2(t),\ \cdots,\ p_m(t)]^T\in\mathbb{R}^{1\times m}$

将其改写为无约束的最优化问题，即增广 Lagrange 泛函

$$J_a[x,w,p] = \Psi(x(t_1),t_1) + \int_{t_0}^{t_1}\left[L(t,x,w) + p\left(f(t,x,w)-\dot{x}\right)\right]dt$$

（为了符号简便，省略记号"(t)"，并将对 t 求导记为"•"）

$$= \Psi(x(t_1),t_1) + \int_{t_0}^{t_1}\left[L+pf\right]dt - \int_{t_0}^{t_1} p\dot{x}\,dt$$

分部积分：$\int_{t_0}^{t_1} p\dot{x}\,dt = \int_{t_0}^{t_1} p\,dx = px\Big|_{t_0}^{t_1} - \int_{t_0}^{t_1}\dot{p}x\,dt$

$$= \Psi(x(t_1),t_1) + \int_{t_0}^{t_1}\left[L+pf+\dot{p}x\right]dt - p(t_1)x(t_1) + p(t_0)x(t_0)$$

记微分方程中控制变量微小扰动 δw 造成状态变量的变化为 δx，则对应增广 Lagrange 泛函的变化为

$$\delta J_a = \left[\left(\frac{\partial\Psi}{\partial x}-p\right)\delta x\right]_{t=t_1} + \int_{t_0}^{t_1}\left[\left(\frac{\partial L}{\partial x}+p\frac{\partial f}{\partial x}+\dot{p}\right)\delta x + \left(\frac{\partial L}{\partial w}+p\frac{\partial f}{\partial w}\right)\delta w\right]dt$$

其中 $x(t_0)$ 为固定的，因此 $\delta x|_{t=t_0}=0$. 若 Lagrange 乘子满足

$$\begin{cases} p(t_1)=\dfrac{\partial \Phi(x(t_1),t_1)}{\partial x}, \\ \dfrac{dp(t)}{dt}=-\dfrac{\partial L(t,x(t),w(t))}{\partial x}-p(t)\dfrac{\partial f(t,x(t),w(t))}{\partial x}, \quad t_0\le t<t_1, \end{cases}$$

则增广 Lagrange 泛函的变化为

$$\delta J_a=\int_{t_0}^{t_1}\left(\frac{\partial L}{\partial w}+p\frac{\partial f}{\partial w}\right)\delta w\, dt$$

因此 $w^*(t)$ 使得增广 Lagrange 泛函取得最小的必要条件为

$$\left[\frac{\partial L(t,x(t),w(t))}{\partial w}+p(t)\frac{\partial f(t,x(t),w(t))}{\partial w}\right]_{w(t)=w^*(t)}=0, \quad t_0\le t\le t_1.$$

神经微分方程的最优控制问题

特别的，考虑 $\Phi(x(t_1),t_1)=\varphi(x(t_1))$, $L(t,x(t),w(t))=0$. 则神经微分方程最优控制问题的解满足.

$$\begin{cases} x^*(0)=x_0;\ \dfrac{dx^*(t)}{dt}=f(t,x^*(t),w^*(t)),\ 0<t\le 1 & \text{[状态方程]} \\ p^*(1)=\dfrac{d\varphi(x^*(1))}{dx};\ \dfrac{dp^*(t)}{dt}=-p^*(t)\dfrac{\partial f(t,x^*(t),w^*(t))}{\partial x},\ 0\le t<1 & \text{[伴随方程]} \\ p^*(t)\dfrac{\partial f(t,x^*(t),w^*(t))}{\partial w}=0,\quad 0\le t\le 1 & \text{[控制方程]} \end{cases}$$

若改写为迭代格式，即为

$$\begin{cases} x(0)=x_0;\ dx(t)=f(t,x(t),w(t)),\ 0<t\le 1 & \text{[前向传播]} \\ p(1)=\dfrac{d\varphi(x(1))}{dx};\ dp(t)=-p(t)\dfrac{\partial f(t,x(t),w(t))}{\partial x},\ 0\le t<1 & \\ w(t)\leftarrow w(t)-\eta\, p(t)\dfrac{\partial f(t,x(t),w(t))}{\partial w},\ 0\le t\le 1 & \text{[反向传播]} \end{cases}$$

这与离散情形下 ResNet 的前向-反向传播完全一致。

3、应用与拓展

自然而然的，可以用神经微分方程来解决分类问题、生成模型等诸多应用问题，有着广泛的应用场景。特别的，由神经网络表达的右端项 $f(t, x(t), w(t))$ 拥有强大的拟合能力，可根据学习任务演化非常复杂的动力学行为（详情参见 Jupyter Notebook）.

当然，神经微分方程方法也存在有局限性，需要进一步改进以增强其表达能力，例如 Augmented Neural ODEs（详情参见 Jupyter Notebook）.