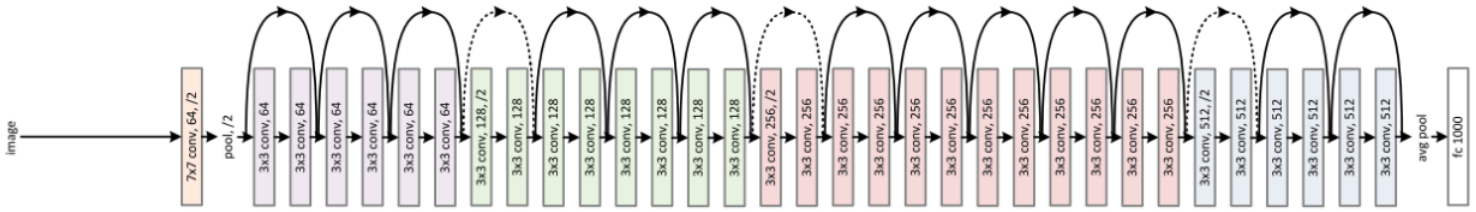
Содержание

1 Введение

В 2015 году, в работе Deep Residual Learning for Image Recognition была предложена архитектура, впоследствии получившая название Residual Network. Главная особенность такой архитектуры состоит в «пробрасывании» выходов предыдущих слоев на последующие: $z(t+1) = z(t) + f(z(t), \theta)$.



Такие «пробрасывания» (skip-connektions) позволяют обучать нейронные сети с большим количеством слоев без затуханий градиента, т.к процедура обратного рспространения ошибки проходит не только через нелинейные преобразования, но и через тождественные. Архитектура оказалась успешной и применяется во множестве прикладных задач.

2 Нейросетевые дифференциальные уравнения

Можно заметить, что система:

$$\begin{cases} z(t+1) = z(t) + f(z(t), \theta_t), \ t \in [0, \dots, T] \\ z(0) = \mathbf{x}, \ \mathbf{x} - \text{вход для нейронной сети} \end{cases}$$

Является схемой Эйлера с шагом $\Delta t = 1$ для решения дифференциального уравнения:

$$\begin{cases} \frac{dz(t)}{dt} = f(z(t), \theta, t), \ t \in [0, \dots, T] \\ z(0) = \mathbf{x}, \ \mathbf{x} - \text{вход для нейронной сети} \end{cases} \tag{1}$$

В статье Neural Ordinary Differential Equations было предложено рассматривать напрямую дифференциальное уравнение (1) и обучать параметры θ этого уравнения, представив $f(z(t), \theta, t)$ нейросетью, для оптимизации функции потерь $L(y) = L(z(T)) \to \min\limits_{\theta}$. Такая постановка задачи имеет ряд преимуществ:

- экономия памяти, т.к нет необходимости хранить значения активаций;
- возможность менять вчислительную сложность модели, варьируя масштаб сетки методов решения дифференциального уравнения(ODEsolver);
- меньшее количество параметров по сравнению с остаточными сетями (residual networks).

3 Обучение параметров дифференциального уравнения

Поиск $\frac{dL}{d\theta}$ напрямую, через операции ODEsolver'a приведет к большому расходу памяти. Поэтому было предложено перейти к сопряженной переменной $a(t) = \frac{\partial L}{\partial z(t)}$. Для такой переменной система будет выглядет следующим образом:

$$\begin{cases} \frac{da(t)}{dt} = -a(t) \frac{\partial}{\partial z(t)} f(z(t), t, \theta) \\ a(T) = \frac{\partial L}{\partial z(T)} \end{cases} \tag{2}$$

Доказательство:

$$\begin{aligned}
\frac{\partial L}{\partial z(t)} &= \frac{\partial L}{\partial z(t+\epsilon)}\frac{z(t+\epsilon)}{\partial z(t)} = a(t+\epsilon)\frac{z(t+\epsilon)}{\partial z(t)} \\
\frac{dz(t+\epsilon)}{dz(t)} &\approx \frac{d}{dz(t)}\big(z(t)+\epsilon f(z(t),t,\theta)\big) = 1+\epsilon\frac{\partial}{\partial z(t)}f(z(t),t,\theta) \\
a(t) &\approx a(t+\epsilon)\left(1+\epsilon\frac{\partial}{\partial z(t)}f(z(t),t,\theta)\right) \\
\frac{da(t)}{dt} &= \lim_{\epsilon\to 0}\frac{a(t+\epsilon)-a(t)}{\epsilon} = -\lim_{\epsilon\to 0}\frac{\epsilon a(t+\epsilon)\frac{\partial}{\partial z(t)}f(z(t),t,\theta)}{\epsilon} \\
&= -a(t)\frac{\partial}{\partial z(t)}f(z(t),t,\theta)\square
\end{aligned}$$

Переменную $a(t)$ можно интерпретировать как производную функции потерь по выходу слоя t остаточной сети с континуальным множеством слоев. Систему (2) можно получить и для дискретной сетки с шагом 1, т.е для остаточных сетей:

$$z(t+1) = z(t) + f(z(t),\theta_t)$$

$$a(t) = \frac{\partial L}{\partial z(t)} = \frac{\partial L}{\partial z(t+1)}\frac{\partial z(t+1)}{\partial z(t)} = a(t+1)\frac{\partial z(t+1)}{\partial z(t)} = a(t+1)\frac{\partial[z(t)+f(z(t),\theta_t)]}{\partial z(t)}$$

$$a(t+1)-a(t) = -a(t+1)\frac{\partial f(z(t),\theta_t)}{\partial z(t)}$$

Последнее выражение – разностное отношение при $\Delta t = 1$, приближающее производную $\frac{d}{dt}a(t)$.

Для поиска $\frac{dL}{d\theta}$ авторы предлагают рассмотреть расширенную систему:

$$\frac{d}{dt}\begin{bmatrix} z \\ \theta \\ t \end{bmatrix}(t) = f_{aug}\left([z,\theta,t]\right) = \begin{bmatrix} f(z(t),\theta,t) \\ \mathbf{0} \\ 1 \end{bmatrix}, \text{ т.к. } \frac{\partial\theta(t)}{\partial t} = \mathbf{0}, \qquad \frac{\partial t(t)}{\partial t} = 1$$

$$a_{aug} = \begin{bmatrix} a \\ a_\theta \\ a_t \end{bmatrix}, \qquad a_\theta(t) = \frac{dL}{d\theta(t)}, \qquad a_t(t) = \frac{dL}{dt(t)}$$

Втаком случае анологично доказательству (2) выводится:

$$\frac{da_{aug}(t)}{dt} = -\begin{bmatrix} a(t) & a_\theta & a_t \end{bmatrix}\cdot\frac{\partial f_{aug}}{\partial\left[z,\theta,t\right]}(t) = -\begin{bmatrix} a\frac{\partial f}{\partial z} & a\frac{\partial f}{\partial\theta} & a\frac{\partial f}{\partial t} \end{bmatrix}(t) \tag{3}$$

Тогда:

$$\frac{dL}{d\theta} = a_\theta(t_0) = a_\theta(T) - \int_T^{t_0} a(t)\frac{\partial f(z(t),t,\theta)}{\partial\theta}dt = \{a_\theta(T)=0\} = \int_{t_0}^{T} a(t)\frac{\partial f(z(t),t,\theta)}{\partial\theta}dt \tag{4}$$

Такой интеграл аналитически невозможно посчитать, однако можно оценить его значение численными методами.

4 Достоинства метода

Экономия памяти. Нет необходимости хранить значения активаций во время прямого прохода. Также память, требуемая для обновление весов модели, является функцией с ассимптотикой $O(1)$ от «глубины» сети.

Адаптивные вычислительные затраты.Возможно менять вчислительную сложность модели, варьируя масштаб сетки методов решения дифференциального уравнения(ODEsolver).

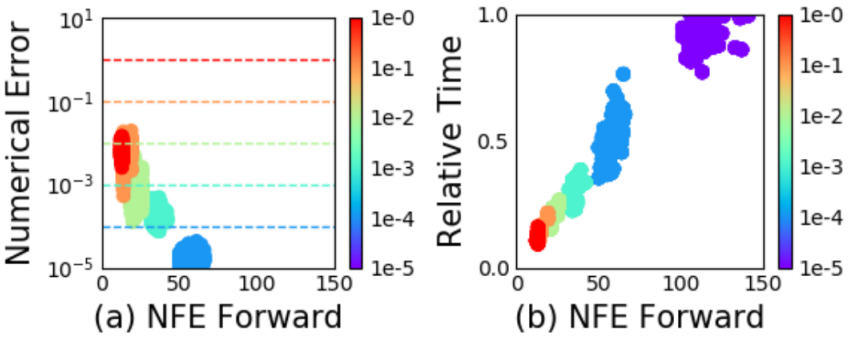


Рис. 1: Зависимость ошибки(слева) и времени(спарва) от NFE(number of function evaluations).

Также в такой постановке легко отойти от численно неустойчивого метода Эйлера к более продвинутым численным методам.

Еще одним преимуществом данного метода является **меньшее количество параметров** по сравнению с остаточными сетями (residual networks), т.к мы не требуем для каждого слоя новые параметры.

Ниже представлено сравнение архитектур на датасете MNIST: 1-слойная нейросети, остаточной сети и нейродифференциальных уравнений, обучаемых напрямую (RK-Net) и через сопряженную систему (ODE-Net).

	Test Error	# Params	Memory	Time
1-Layer MLP†	1.60%	0.24 M	-	-
ResNet	0.41%	0.60 M	$\mathcal{O}(L)$	$\mathcal{O}(L)$
RK-Net	0.47%	0.22 M	$\mathcal{O}(\tilde{L})$	$\mathcal{O}(\tilde{L})$
ODE-Net	0.42%	0.22 M	$\boldsymbol{\mathcal{O}(1)}$	$\mathcal{O}(\tilde{L})$

В файлах demo.ipynb и demo.html находится пример работы нейродифференциальных уравнений с разными способами обновления весов.