



深度学习自动微分（一） 什么是自动微分？

嘉宾： 高仪宣



什么是自动微分?

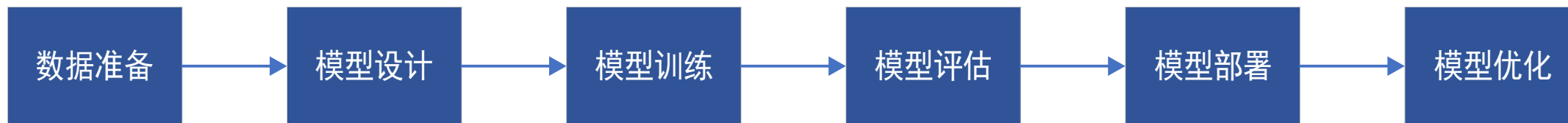
CONTENTS 目录

- 1、背景
- 2、常见的微分方式
- 3、自动微分的重要性
- 4、参考文献



什么是自动微分？

1、背景



深度学习的开发流程

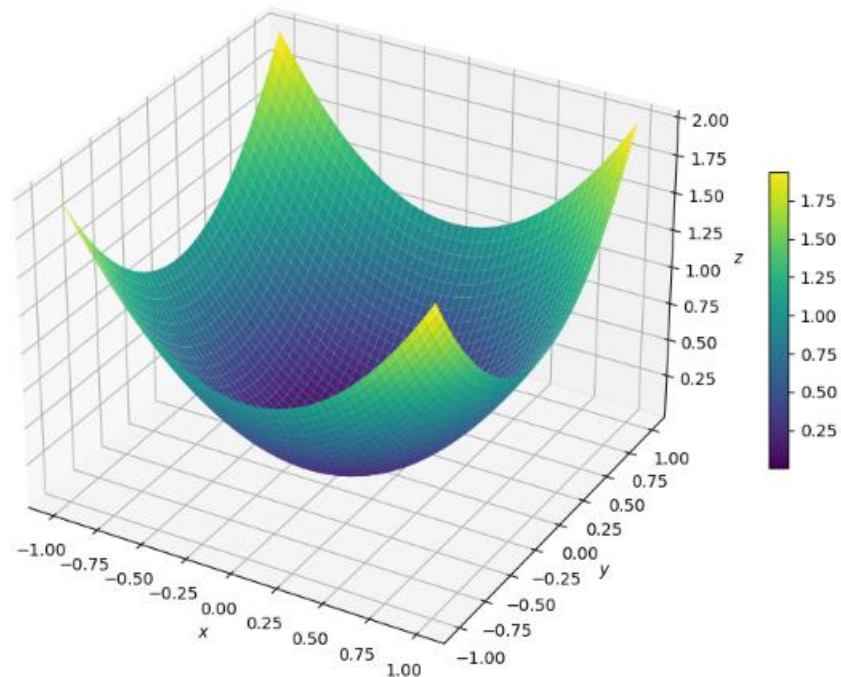
在模型训练过程中，通常会用损失函数来衡量模型的预测值与真实值之间的误差，我们的目标就是最小化这个损失函数，找到最优解。那么，如何最小化损失函数呢？



什么是自动微分？

在形式上，梯度是多元函数分别对各个自变量求偏导数，并把求得的偏导数组合形成的向量。在作用上，梯度向量方向就是该多元函数在给定点处上升最快的方向。

梯度可以反映函数的变化趋势，通过计算损失函数关于模型参数的梯度，可以找到使得损失函数最小化的参数值，进而优化模型，微分机制就是常用的求解模型梯度的方法。由此可以看出，微分机制对于深度学习而言具有至关重要的作用。



多元函数图像



什么是自动微分?

2、常见的微分方式



手动微分

利用手工求导，依据链式法则解出梯度公式并编写程序，带入自变量，求解梯度。



符号微分

利用求导规则对函数表达式进行自动计算，计算结果是导函数的表达式。带入自变量，求解梯度。



数值微分

利用导数的定义，通过有限差分近似方法求解梯度。



自动微分

是一种介于数值微分和符号微分之间的求导方法，采用类似有向图的方法来求解梯度。



什么是自动微分?

$$l_1 = x$$

$$l_{n+1} = l_n(l_n + 4)$$

$$f(x) = l_4 = x(x+4)(x+2)^2(x^2+4x+2)^2$$



f(x):

```
return x*(x+4)*((x+2)^2)*(x*x+4*x+2)^2
```



f(x):

```
v = x
for i = 1 to 3
    v = v*(v+4)
return v
```

closed-form表达式: 由初等函数经过有限次的初等运算复合而成的, 不包含编程语言中的循环、条件等结构的表达式。

函数编码流程



什么是自动微分?

2.1、手动微分

手动微分，是利用手工方式求解梯度的方法。

手动微分实现过程：

- 1) 写出目标函数的导函数表达式；
- 2) 依照目标函数的导函数表达式编写程序；
- 3) 将自变量带入到程序中，求出梯度结果。

优点：

- 1) 梯度结果数值精确。

缺点：

- 1) 手动计算导函数表达式工作量较大。
- 2) 通用性差，修改算法模型时需要重新手动计算导函数表达式。

$$f(x) = l_4 = x(x+4)(x+2)^2(x^2+4x+2)^2$$

手动微分

$$\begin{aligned} f'(x) = & (x+4)(x+2)^2(x^2+4x+2)^2 \\ & + x(x+2)^2(x^2+4x+2)^2 \\ & + 2x(x+4)(x+2)(x^2+4x+2)^2 \\ & + 2x(x+4)(x+2)^2(x^2+4x+2)(2x+4) \end{aligned}$$

编写程序

```
f'(x):  
    return (x+4)*((x+2)^2)*((x*x+4*x+2)^2)  
        + x*((x+2)^2)*((x*x+4*x+2)^2)  
        + 2*x*(x+4)*(x+2)*((x*x+4*x+2)^2)  
        + 2*x*(x+4)*((x+2)^2)*(x*x+4*x+2)*(2*x+4)
```

带入自变量

带入自变量，得到梯度结果

手动微分实现流程



什么是自动微分？

2.2、符号微分

符号微分，是利用求导法则制定表达式变换规则，然后对表达式进行自动计算、求解梯度的方法。

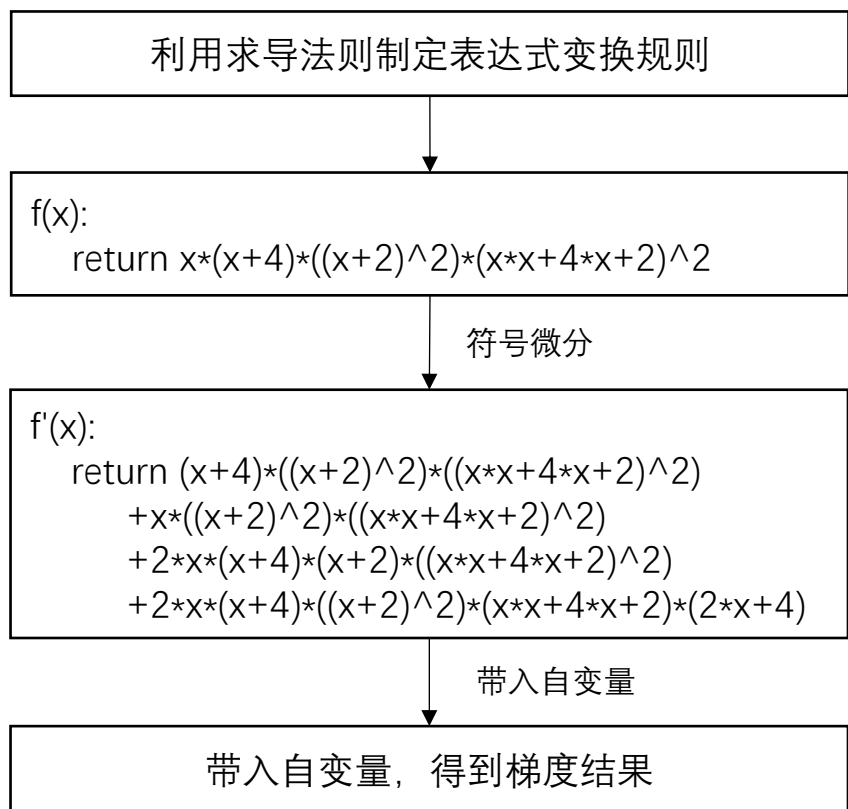
$$\frac{d}{dx}(f(x) + g(x)) \rightarrow \frac{d}{dx}f(x) + \frac{d}{dx}g(x)$$

$$\frac{d}{dx}(f(x)g(x)) \rightarrow \left(\frac{d}{dx}f(x)\right)g(x) + f(x)\left(\frac{d}{dx}g(x)\right)$$

求导法则

符号微分实现流程：

- 1) 利用求导法则制定表达式变换规则；
- 2) 将目标函数转换为closed-form表达式程序；
- 3) 依据表达式变换规则，对closed-form表达式程序进行自动计算，获得导函数表达式程序；
- 4) 将自变量带入到导函数表达式程序中，求出梯度结果。



符号微分实现流程



什么是自动微分?

优点:

- 1) 梯度结果数值精确。
- 2) 实现起来较为简单。

缺点:

- 1) 表达式必须是closed-form的。
- 2) 表达式较为复杂的时候, 结果可能会产生“表达式膨胀”问题。

n	l_n	$\frac{d}{dx} l_n$	$\frac{d}{dx} l_n$ (化简)
1	x	1	1
2	$x(x+4)$	$(x+4) + x$	$2x+4$
3	$x(x+4)(x+2)^2$	$(x+4)(x+2)^2$ $+x(x+2)^2$ $+2x(x+4)(x+2)$	$4(x^3+6x^2+10x+4)$
4	$x(x+4)(x+2)^2(x^2+4x+2)^2$	$(x+4)(x+2)^2(x^2+4x+2)^2$ $+x(x+2)^2(x^2+4x+2)^2$ $+2x(x+4)(x+2)(x^2+4x+2)^2$ $+2x(x+4)(x+2)^2(x^2+4x+2)(2x+4)$	$8(x^7+14x^6+78x^5+220x^4$ $+330x^3+252x^2+84x+8)$

“表达式膨胀”问题



什么是自动微分?

2.3、数值微分

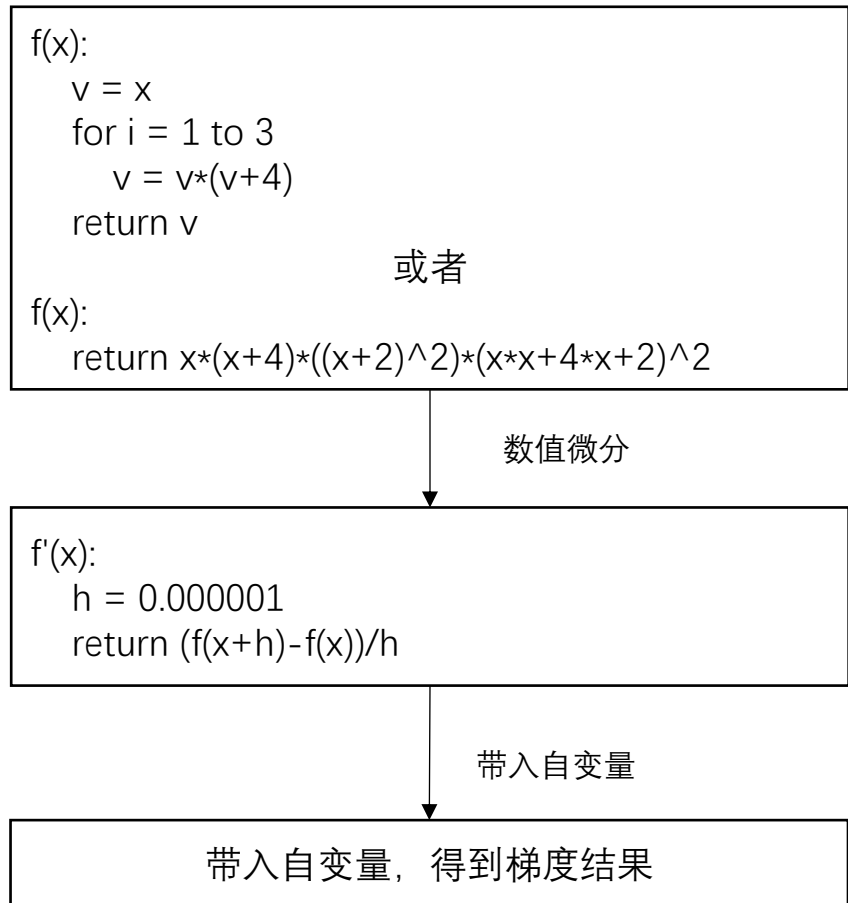
数值微分，是根据导数的原始定义，使用有限差分近似的方式来求解梯度的方法。

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + he_i) - f(x)}{h}, h > 0, e_i \text{ 为第 } i \text{ 个单位向量}$$

差分公式

数值微分实现流程：

- 1) 依照目标函数编写程序；
- 2) 选择合适的 h 值，依照差分公式编写程序；
- 3) 将自变量带入到差分公式程序中，求出梯度结果。



数值微分实现流程



什么是自动微分?

优点:

- 1) 几乎适用于所有情况, 除非不可导的点。
- 2) 对用户隐藏求解过程。
- 3) 实现起来较为简单。

缺点:

- 1) 梯度结果数值不精确, 会存在截断误差和舍入误差。
- 2) 计算复杂度高, 每计算一个参数的导数, 都要重新计算。
- 3) 需要选择合适的 h 值。

截断误差: 数值计算中 h 无法真正取零导致的近似误差。

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + he_i) - f(x)}{h}, h > 0, e_i \text{ 为第 } i \text{ 个单位向量}$$

差分公式

舍入误差: 计算过程中对小数位数的不断舍入会导致求导过程中的误差不断累积。



什么是自动微分？

2.4、自动微分

自动微分是介于数值微分和符号微分之间的方法，采用类似有向图的方式来求解梯度。

$$y = f(g(h(x)))$$

$$y' = (f(g(h(x))))' = f'(g(h(x)))g'(h(x))h'(x)$$

链式求导法则

自动微分实现流程：

- 1) 依照目标函数编写程序；
- 2) 利用表达式追踪机制记录每一次计算过程中的中间变量；
- 3) 利用链式求导法则组合各部分中间变量，生成梯度结果。

特点：

- 1) 所有数值计算都由有限的基本运算组成。
- 2) 基本运算的导数表达式是已知的。
- 3) 通过链式求导法则将数值计算各部分组合成整体。

表达式追踪 (evaluation trace)：表达式追踪是自动微分的基础，可以追踪数值计算过程的中间变量，表示中间变量之间的依赖关系。任何有关数值计算的逻辑最后都可以表达成一个基于数值的计算序列，该计算序列包含确定的输入、中间变量和输出。因为控制流不会直接改变数值变量的结果，所以自动微分也适用于使用了分支、循环、递归、函数调用的算法。



什么是自动微分?

```
f(x):  
  v = x  
  for i = 1 to 3  
    v = v*(v+4)  
  return v
```

或者

```
f(x):  
  return x*(x+4)*((x+2)^2)*(x*x+4*x+2)^2
```

自动微分

```
f'(x):  
  (v,dv)=(x,1)  
  for i = 1 to 3:  
    (v,dv) = (v*(v+4),2*v*dv+4*dv)  
  return (v,dv)
```

带入自变量

带入自变量, 得到梯度结果

自动微分实现流程

优点:

- 1) 梯度结果数值精确。
- 2) 对用户隐藏求解过程。
- 3) 无“表达式膨胀”问题。
- 4) 可以灵活结合编程语言的循环结构, 条件结构来使用。

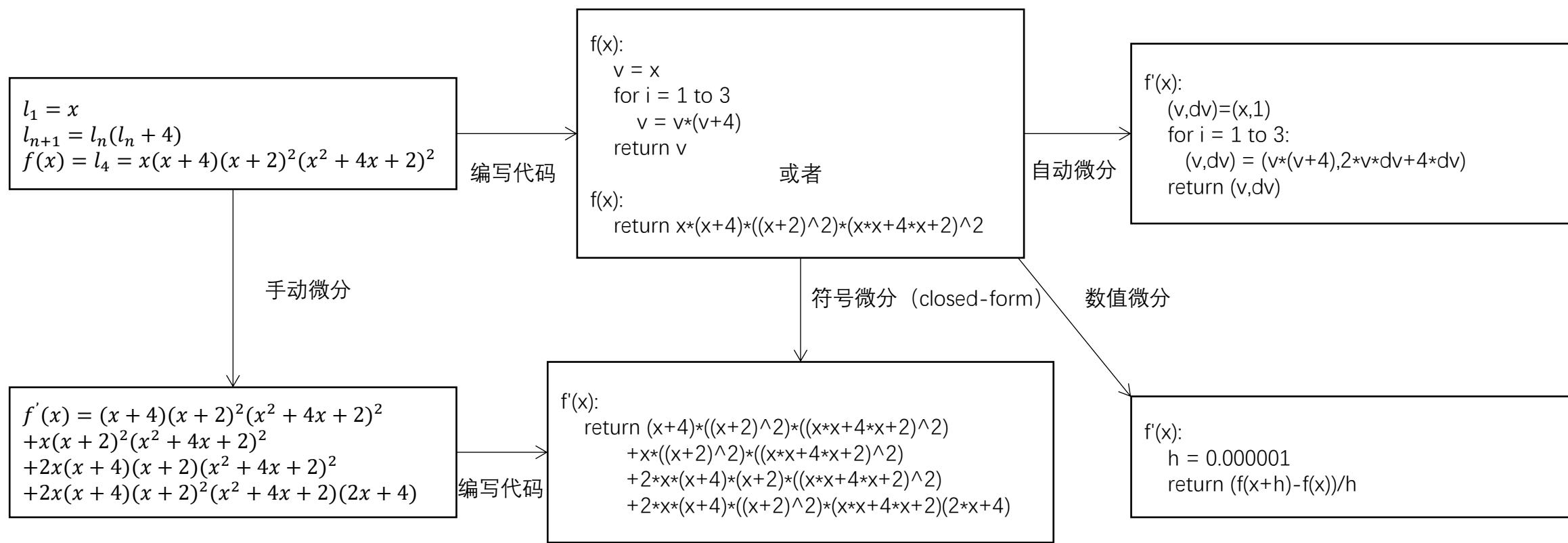
缺点:

- 1) 计算过程中需要存储一些中间求导结果, 因此会增加内存占用。



什么是自动微分？

2.5、总结



不同微分方式求解过程对比



什么是自动微分？

3、自动微分的重要性

自动微分作为求解梯度的方法，极大地促进了深度学习的发展，它使得用户摆脱手动推导公式并进行算法实现的繁琐操作，能够很方便地实现卷积、全连接、循环神经网络中梯度求解。目前，TensorFlow、PyTorch等各种主流深度学习框架都提供了自动微分功能。



部分主流深度学习框架



什么是自动微分？

4、参考文献

本期视频的参考文献如下:

- 1、Automatic Differentiation in Machine Learning a Survey
- 2、[【符号微分/数值微分/自动微分区别是什么？【自动微分】系列第二篇】](#)
- 3、[【自动微分原理】一文看懂AD原理](#)
- 4、[深度学习利器之自动微分](#)



感谢大家聆听