法律声明

课程详情请咨询

- 微信公众号：深度之眼
- 客服微信号：deepshare0920

公众号

微 信

deepshare.net
深度之眼

计算图与动态图机制

导师：余老师

deepshare.net
深度之眼

目录

deepshare.net
深度之眼

计算图

Computational Graph

计算图

Computational Graph

计算图是用来描述运算的有向无环图

计算图有两个主要元素：结点（Node）和边（Edge）

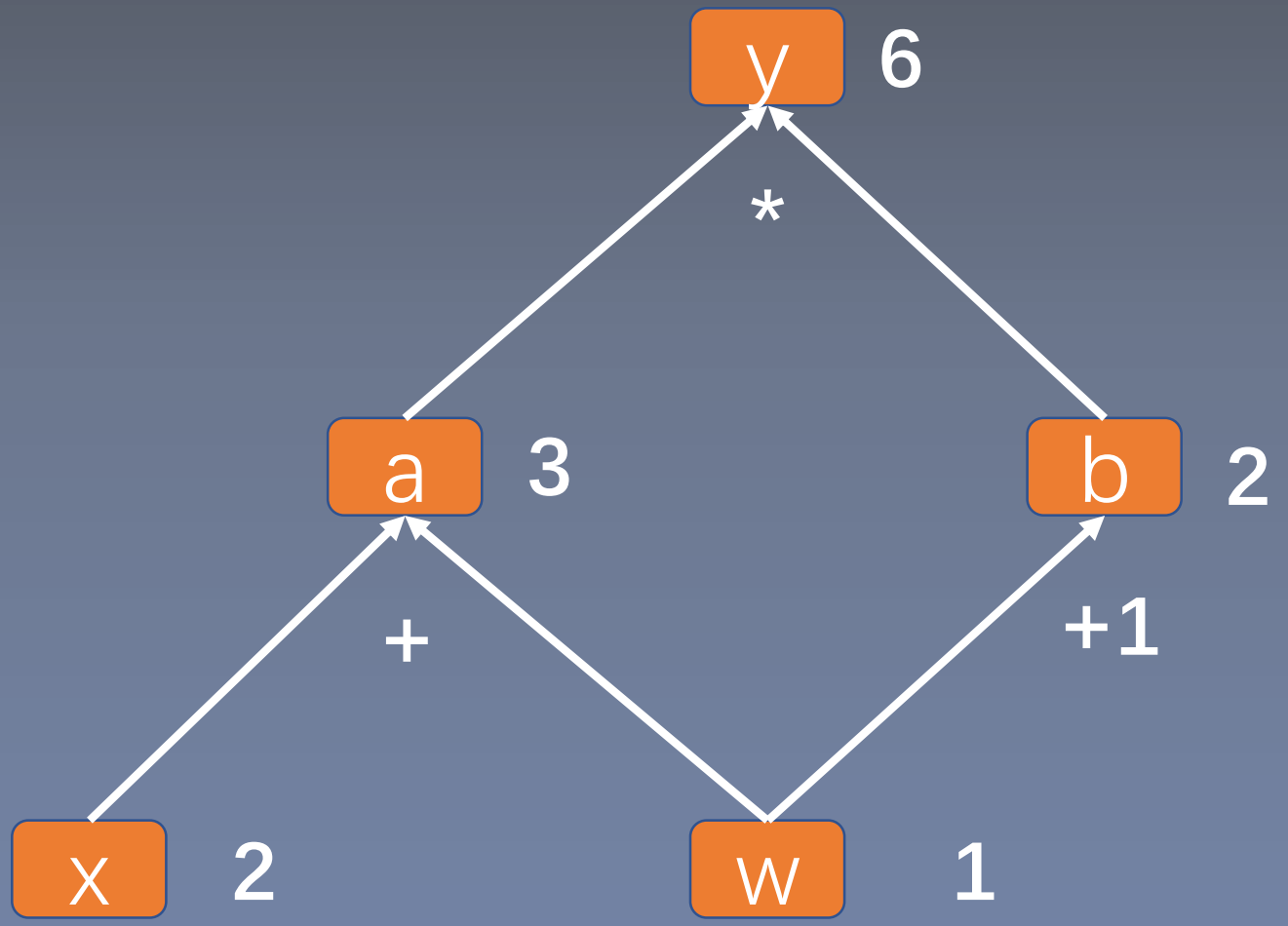
结点表示数据，如向量，矩阵，张量

边表示运算，如加减乘除卷积等

用计算图表示：y = (x+ w) * (w+1)

a = x + w　　b = w + 1

y = a * b



计算图

Computational Graph

计算图与梯度求导

y = (x+ w) * (w+1)

a = x + w　　b = w + 1

y = a * b

$$\frac{\partial y}{\partial w} = \frac{\partial y}{\partial a}\frac{\partial a}{\partial w} + \frac{\partial y}{\partial b}\frac{\partial b}{\partial w}$$

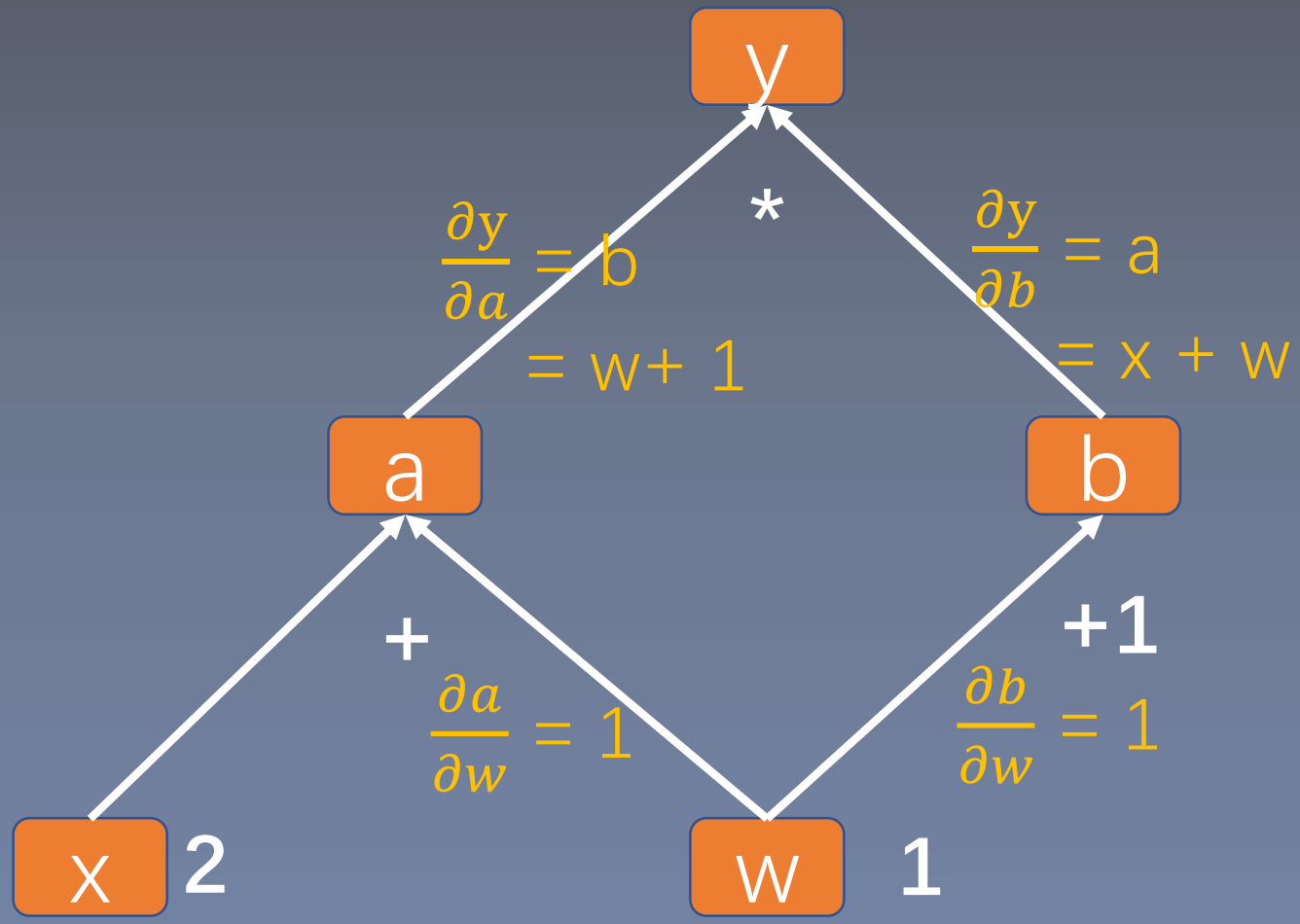
= b * 1 + a *1

= b + a

= (w + 1) + (x + w)

= 2*w + x + 1

= 2*1 + 2 + 1 = 5



计算图

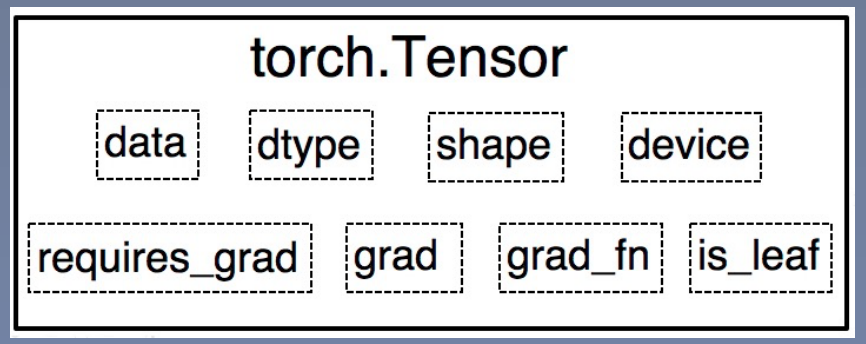
Computational Graph

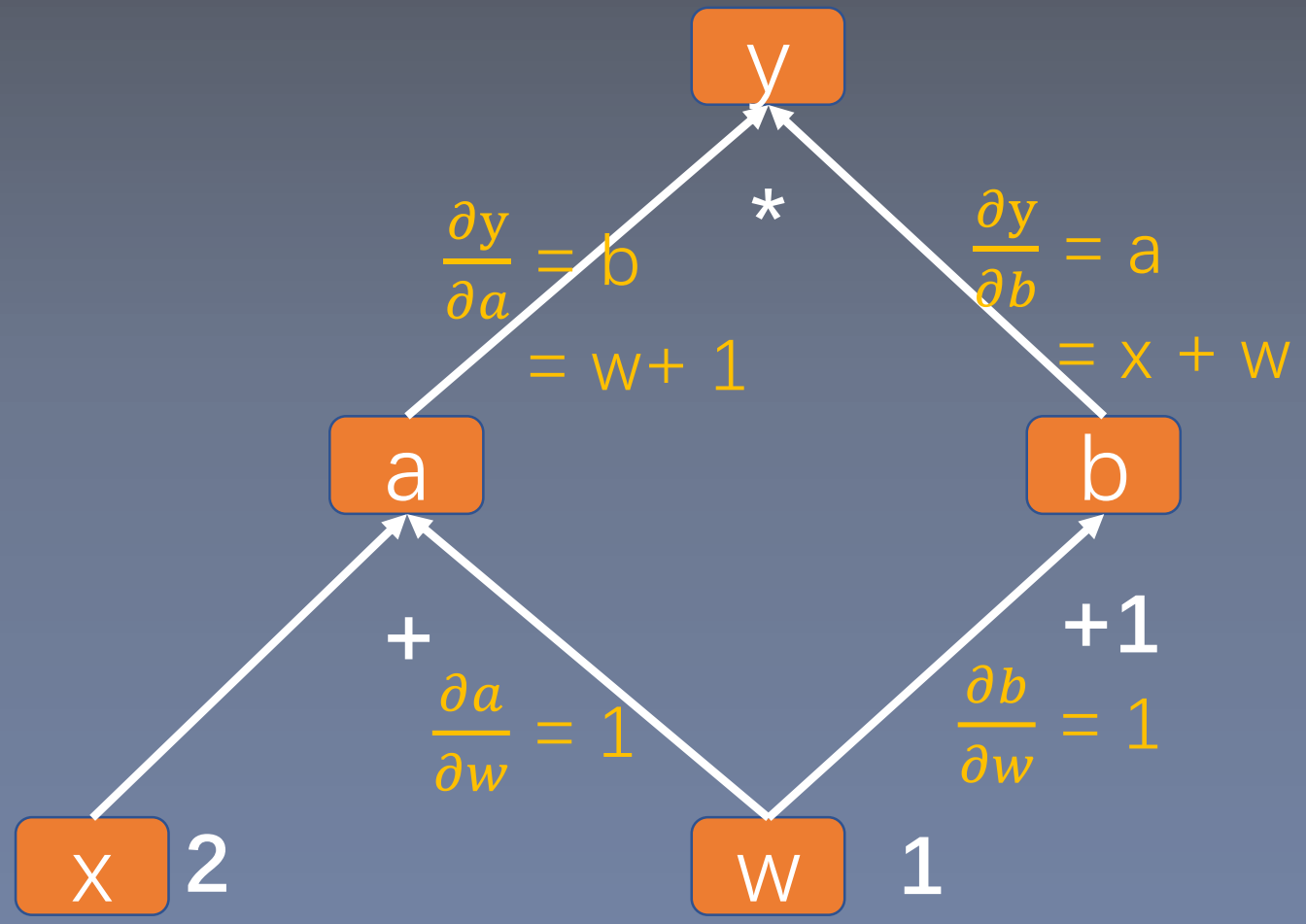
计算图与梯度求导

y = (x+ w) * (w+1)

叶子结点：用户创建的结点称为叶子结点，如X 与 W

is_leaf: 指示张量是否为叶子结点





计算图

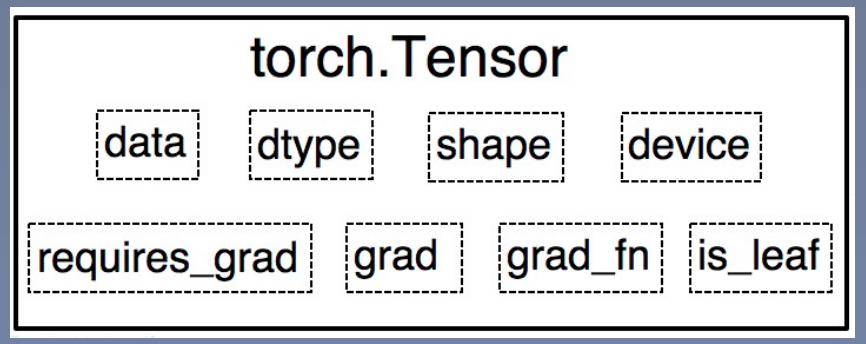
Computational Graph

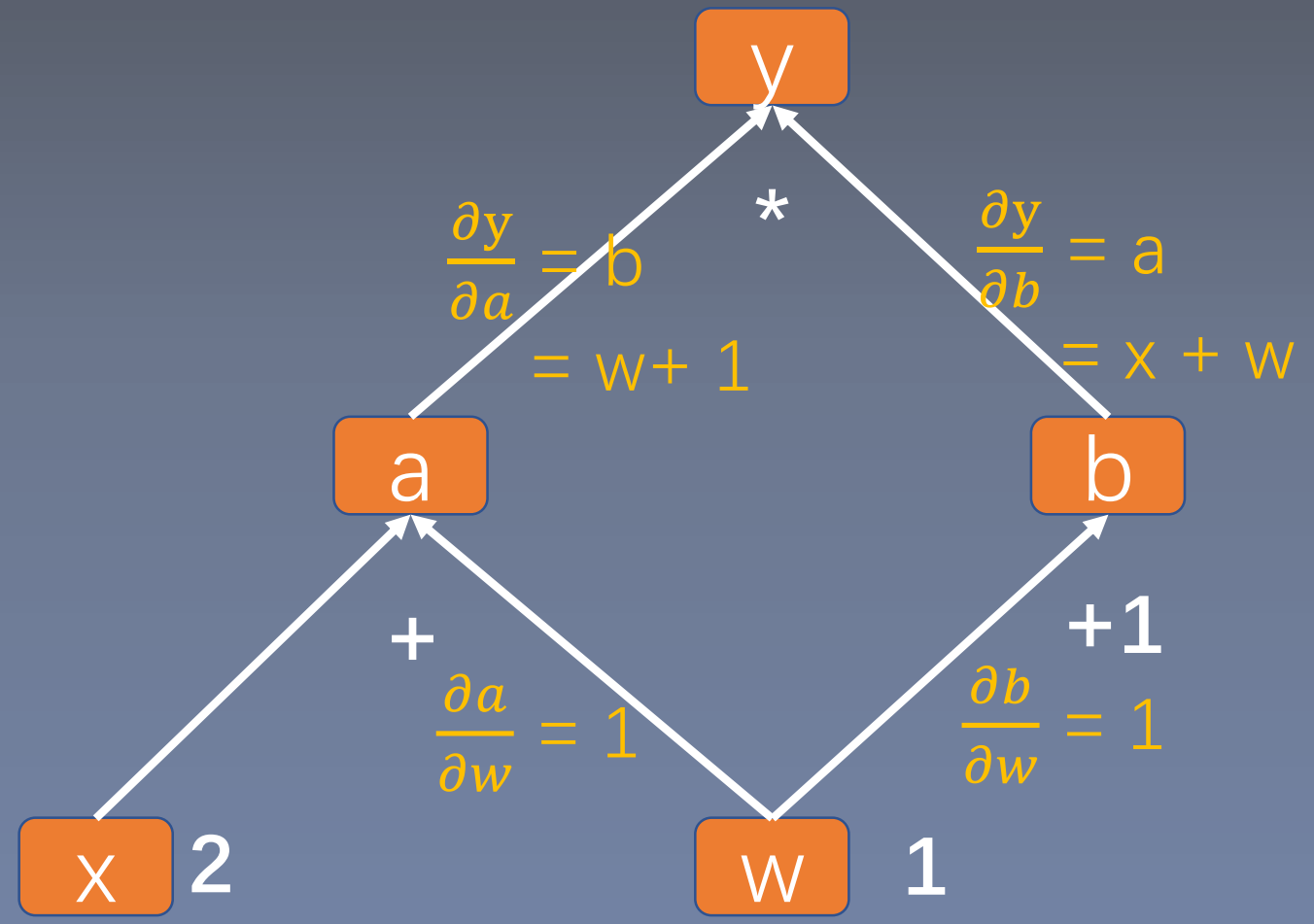
grad_fn: 记录创建该张量时所用的方法（函数）

y.grad_fn = <MulBackward0>

a.grad_fn = <AddBackward0>

b.grad_fn = <AddBackward0>





deepshare.net
深度之眼

动态图

Dynamic Graph

动态图vs 静态图

Dynamic VS Static Computational Graphs

deepshare.net
深度之眼

动态图

运算与搭建同时进行

灵活 易调节

静态图

先搭建图，后运算

高效 不灵活

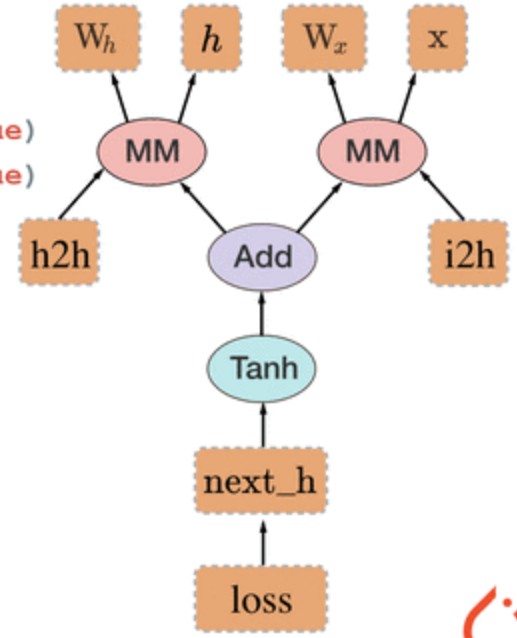
根据计算图搭建方式，可将计算图分为动态图和静态图

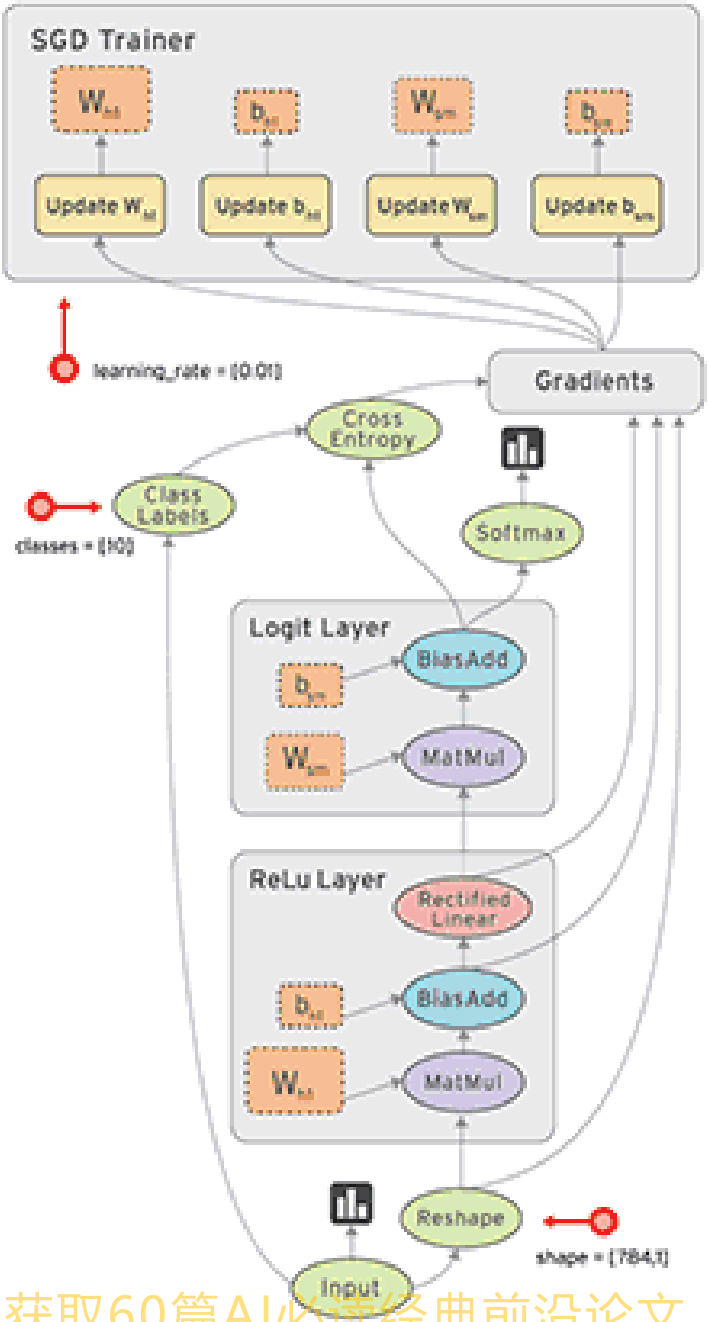
Back-propagation
uses the dynamically created graph

```
W_h = torch.randn(20, 20, requires_grad=True)
W_x = torch.randn(20, 10, requires_grad=True)
x = torch.randn(1, 10)
prev_h = torch.randn(1, 20)

h2h = torch.mm(W_h, prev_h.t())
i2h = torch.mm(W_x, x.t())
next_h = h2h + i2h
next_h = next_h.tanh()

loss = next_h.sum()
loss.backward()  # compute gradients!
```





动态图 PyTorch ↑

静态图 TensorFlow →

结 语

在这次课程中，学习了计算图与动态图机制

在下次课程中，我们将会学习PyTorch的

自动求导系统torch.autograd及

逻辑回归实现

deepshare.net

深度之眼

联系我们:

电话: 18001992849

邮箱: service@deepshare.net

Q Q: 2677693114

公众号

客服微信

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文