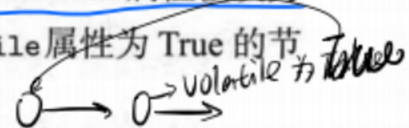


# core\_02\_01\_计算图

2019年12月27日 20:03

- autograd 根据用户对 variable 的操作构建计算图。对 variable 的操作抽象为 Function。
- 由用户创建的节点称为叶子节点，叶子节点的 grad\_fn 为 None。叶子节点中需要求导的 variable，具有 AccumulateGrad 标识，因其梯度是累加的。
- variable 默认是不需要求导的，即 requires\_grad 属性默认为 False。如果某一个节点 requires\_grad 被设置为 True，那么所有依赖它的节点 requires\_grad 都为 True。
- variable 的 volatile 属性默认为 False，如果某一个 variable 的 volatile 属性被设为 True，那么所有依赖它的节点 volatile 属性都为 True。volatile 属性为 True 的节点不会求导，volatile 的优先级比 requires\_grad 高。  

- 多次反向传播时，梯度是累加的。反向传播的中间缓存会被清空，为进行多次反向传播需指定 retain\_graph=True 来保存这些缓存。
- 非叶子节点的梯度计算完之后即被清空，可以使用 autograd.grad 或 hook 技术获取非叶子节点梯度的值。
- variable 的 grad 与 data 形状一致，应避免直接修改 variable.data，因为对 data 的直接操作无法利用 autograd 进行反向传播。
- 反向传播函数 backward 的参数 grad\_variables 可以看成链式求导的中间结果，如果是标量，可以省略，默认为 1。
- PyTorch 采用动态图设计，可以很方便地查看中间层的输出，动态地设计计算图结构。