**Pytorch**

in-place operation：以tensor.add\_(x)的附加下划线形式，表示直接修改该tensor而不是进行复制。

Tensor变量的requires\_grad的属性默认为False,若一个节点requires\_grad被设置为True，那么所有依赖它的节点的requires\_grad都为True。

is\_leaf表示没有被操作的tensor

All Tensors that have [requires\_grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.requires_grad) which is False will be leaf Tensors by convention.

For Tensors that have [requires\_grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.requires_grad) which is True, they will be leaf Tensors if they were created by the user. This means that they are not the result of an operation and so grad\_fn is None.

Only leaf Tensors will have their [grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.grad) populated during a call to [backward()](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.backward). To get [grad](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.grad) populated for non-leaf Tensors, you can use [retain\_grad()](https://pytorch.org/docs/stable/tensors.html#torch.Tensor.retain_grad).

不需要梯度的张量是叶子张量。（普通情况以及由多个叶子张量构成的张量，不需要梯度）

任何张量都需要考虑几个属性：数据域，计算图，梯度