

Tensorflow 中的自动微分

反向传播的计算需要知道正向传播每一步的 Operator 和对应 operands 的值 since

$$\frac{dz}{dx} = \frac{dz}{dy}(y) \cdot \frac{dy}{dx}(x)$$

Diagram illustrating the chain rule with annotations: "operators" points to the derivative terms ($\frac{dz}{dy}$ and $\frac{dy}{dx}$), and "operands" points to the variables (y and x).

Tensorflow 使用 `GradientTape` 来记录运算过程中的操作符及操作数的值。

在 Tensorflow 中 `Tensor` 是 immutable 的, 故在函数中被认为是常量, 由于对变量求 Gradient 才有意义, 因此只可以对 `tf.Variable` 求导。

若需对 `Tensor` 求导则需显式地令 `GradientTape` 添加对其的 watch。

若 `GradientTape(persistent = True)` 则中间变量对原变量的导数会被 cache。

向量函数的求导

Tensorflow 对向量函数的求导处理与 Pytorch 不同。

$$f(x) = (f_1(x), \dots, f_n(x))$$

`tape.gradient` 计算其各分量导数之和: $\sum \frac{\partial f_i}{\partial x}$

而 `tape.jacobian` 计算雅可比矩阵。