**basic NN**

**tensor，叶子节点**

每一个tensor都具有3个主要信息：数据域，梯度和计算图

tensor

数据域

梯度

计算图buffer

我们考虑一个简单的神经网络：

如果表示为tensor的计算图，则为：

点乘

点乘

不更新，不需要计算梯度的

称为叶子节点

需要计算梯度的，但由人为创建的

称为叶子节点

**梯度，反向传播**

由计算图我们可以得到：

的偏导数为：

由链式法则可知，在前向传播的过程中（也就是变量赋值过程），我们把每一层的偏导数相乘就可以得到叶子节点对于**目标变量**的偏导数。在前向传播中会生成计算图的buffer。当执行backward()时，默认情况下只有require\_grad的叶子节点的梯度会被记录，计算图会被销毁。

使用**目标变量**.backward()可以得到叶子tensor的梯度，即

如果我们把tensor 的每个分量的偏导数都按照顺序排好，可以得到一个同型的梯度tensor，它被记录在tensor的梯度之中。（把两组权重的雅克比矩阵排列起来）

以此类推。

中间变量的backward

从链式法则可知，如果我们已知中间变量tensor 关于目标的偏导数，就可以通过它乘以叶子节点关于的偏导数得到叶子节点关于目标变量的梯度。



对于中的每一组权重，我们首先可以得到其关于中间变量的雅克比矩阵，

我们可以看到，叶子节点的梯度就是这个中间变量的雅克比矩阵左乘中间变量的梯度矩阵。



**梯度累积**

每个tensor的梯度域，会根据每次的反向传播结果进行累计。可以通过zero\_grad清空。

**参数更新**

实现此简单网络的权重更新。

**线性回归模型**

完成《动手学深度学习》里的线性回归模型