# 梯度消失和梯度爆炸

## 背景

在神经网络中，我们通常使用正向传播来计算每一层的激励值以得到预测结果，然后我们使用反向传播来修正权重，而修正权重最关键的值便是梯度。在这个过程中，就容易出现梯度消失和梯度爆炸现象，即梯度值变得非常小或者非常大，从而影响神经网络的训练。

## 梯度消失和爆炸的原因

以全链接神经网络为例，在反向传播过程中，根据链式求导法则

对末尾的权值求梯度为

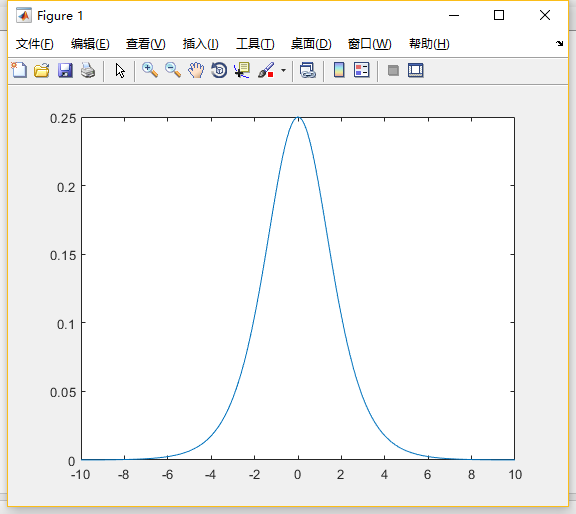
对求梯度可以表示为

以此类推

可以看出，求的梯度中会存在激活函数求导的累乘。

如果激活函数的值小于1，随着反向传播的进行，乘积值会越来越小，使得靠前的权值更新的幅度非常小。

以为例，其导数图像如下，



导数恒小于等于0.25，那么在反向传播的过程中非常容易出现梯度消失，反之则会有梯度爆炸现象。

的两端也被称为饱和区，意为两端的梯度过于平缓，使得权值无法得到有效的更新。

因为，在链式求导的过程中，也存在权值的累乘。如果初始权值设置的过大，那么在反向传播的过程中，梯度就会快速增长，靠前的层的权值变化更快，靠后的变化更慢，这个效果就会越来越明显，反之亦然。

一般来说梯度消失更加常见一些。

## 解决方法简述

#### 使用正则或者对梯度剪切

正则方法原本用于抑制过拟合，其本质是将权值作为惩罚项加入函数中，以抑制权值增长。所以此方法可以用来抑制梯度爆炸。

常用方法为正则（取权值的绝对值）和正则（取权值的平方）。

梯度剪切则更为简单粗暴，直接设定一个权重的阈值，可以防止梯度爆炸。

#### 使用ReLU等激活函数

如果激活函数的导数为1，那么就不存在链式求导中，导数的累乘所引发的梯度问题了。

#### Batch Normalization

//TODO

#### 残差结构

因为捷径的存在，反向传播时可以直接对靠后的层传播梯度，避免了梯度消失过快的现象。