

参数初始化 (Weight Initialization)

- 全零初始化：无法进行模型训练
- 随机初始化：使用小的随机数(高斯分布，零均值， $1e-2$ 标准差)初始化。
小网络可以，对于深度网络有问题。网络输出数据分布的方差会随着神经元的个数而改变。
- Xavier初始化
 - 为保证前向传播和反向传播时每一层的方差一致。根据每层的输入个数和输出个数来决定参数随机初始化的分布范围。

高斯分布的权重初始化为：高斯分布的随机数乘上

$$\frac{\sqrt{2}}{\sqrt{n_{in} + n_{out}}}$$

其中： n_{in} 和 n_{out} 分别表示该层输入和输出的参数（权重）个数。

注：原论文中是基于线性函数推导的，同时在tanh激活函数上有很好的效果，但不适用于ReLU激活函数。

- He参数初始化

对Xavier方法的改进，将ReLU非线性映射造成的影响考虑进参数初始化中。

高斯分布的权重初始化为：高斯分布的随机数乘上 $\frac{\sqrt{2}}{\sqrt{n_{in}}}$

均匀分布的权重初始化为：均匀分布的随机数乘上 $\frac{\sqrt{2}}{\sqrt{n_{in}}}$

小结

- 使用 ReLU (without BN) 激活函数时，最好选用 He 初始化方法，将参数初始化为服从高斯分布或者均匀分布的较小随机数
- 借助预训练模型中参数作为新任务参数初始化的方式也是一种简便易行且有效的模型参数初始化方法
- 一般情况下，偏置设为0.

YOLOv3中的权重初始化

全连接层：均匀分布的He权重初始化

```
//权重初始化
float scale = sqrt(2./inputs);
for(i = 0; i < outputs*inputs; ++i){
    l.weights[i] = scale*rand_uniform(-1, 1);
}

// 初始化所有偏置值为0
for(i = 0; i < outputs; ++i){
    l.biases[i] = 0;
}
```

卷积层：高斯分布的He权重初始化

```
float scale = sqrt(2./((size*size*c/l.groups)));

for(i = 0; i < l.nweights; ++i) l.weights[i] = scale*rand_normal();

l.biases = calloc(n, sizeof(float));
```