deepshare.net
深度之眼

# 法律声明



**课程详情请咨询**

- 微信公众号：深度之眼
- 客服微信号：deepshare0920

公众号

微 信

deepshare.net
深度之眼

# 权值初始化

导师：余老师

deepshare.net
深度之眼

# 目录

deepshare.net
深度之眼

# 梯度消失与爆炸

Gradient Vanishing and Exploding

# 梯度消失与爆炸

## Gradient Vanishing and Exploding



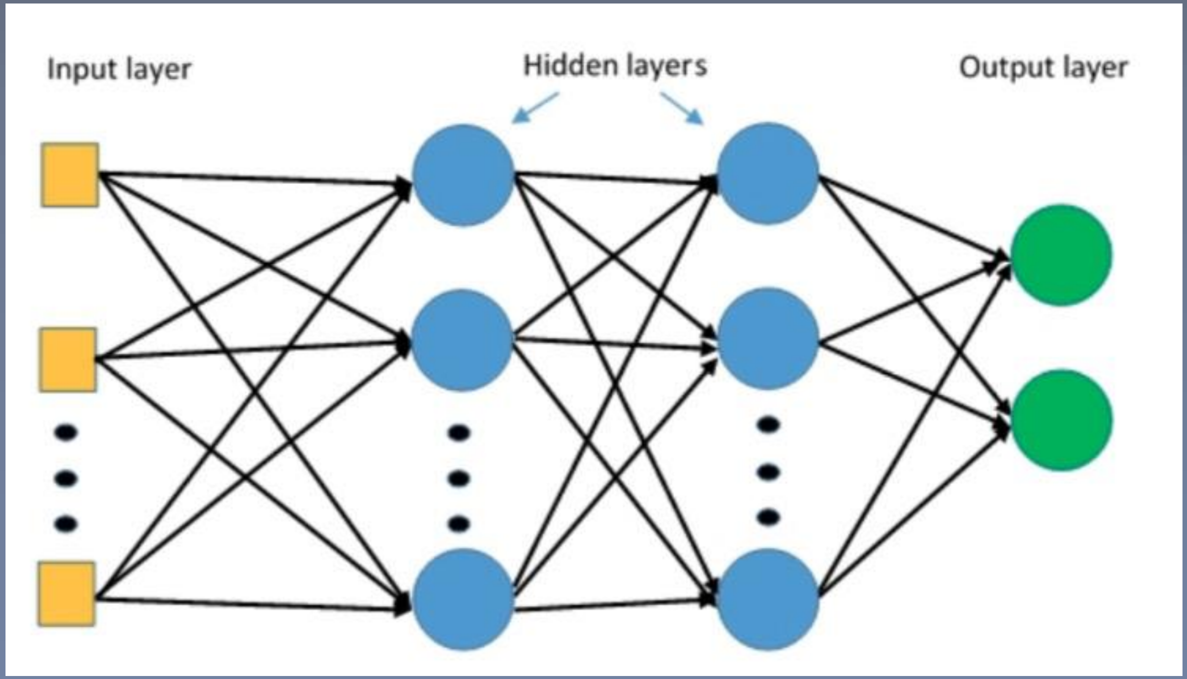
$$H_2 = H_1 * W_2$$

$$\Delta W_2 = \frac{\partial Loss}{\partial W_2} = \frac{\partial Loss}{\partial out} * \frac{\partial out}{\partial H_2} * \frac{\partial H_2}{\partial w_2}$$

$$= \frac{\partial Loss}{\partial out} * \frac{\partial out}{\partial H_2} * H_1$$

梯度消失： $H_1 \to 0 \quad \Rightarrow \quad \Delta W_2 \to 0$

梯度爆炸： $H_1 \to \infty \quad \Rightarrow \quad \Delta W_2 \to \infty$




$X \quad W_1 \quad H_1 \quad W_2 \quad H_2 \quad W_3 \quad out$

# 梯度消失与爆炸

Gradient Vanishing and Exploding



1. $\mathrm{E}(X * Y) = E(X) * E(Y)$

2. $\mathrm{D}(X) = E\left(\mathrm{X}^2\right) - [E(X)]^2$

3. $\mathrm{D}(X + Y) = D(X) + D(Y)$

1.2.3 $\Rightarrow$ $\mathrm{D}(\mathrm{X}*\mathrm{Y}) = \mathrm{D}(\mathrm{X})*\mathrm{D}(\mathrm{Y}) + \mathrm{D}(\mathrm{X})*[E(Y)]^2 + \mathrm{D}(\mathrm{Y})*[E(X)]^2$

若E(X)=0，E(Y)=0

$\mathrm{D}(\mathrm{X}*\mathrm{Y}) = \mathrm{D}(\mathrm{X})*\mathrm{D}(\mathrm{Y})$

# 梯度消失与爆炸

Gradient Vanishing and Exploding



$$\mathrm{H}_{11} = \sum_{i=0}^{n} X_i * W_{1i}$$

D(X*Y) = D(X)*D(Y)

$$\mathbf{D}(\mathrm{H}_{11}) = \sum_{i=0}^{n} D(X_i) * D(W_{1i})$$

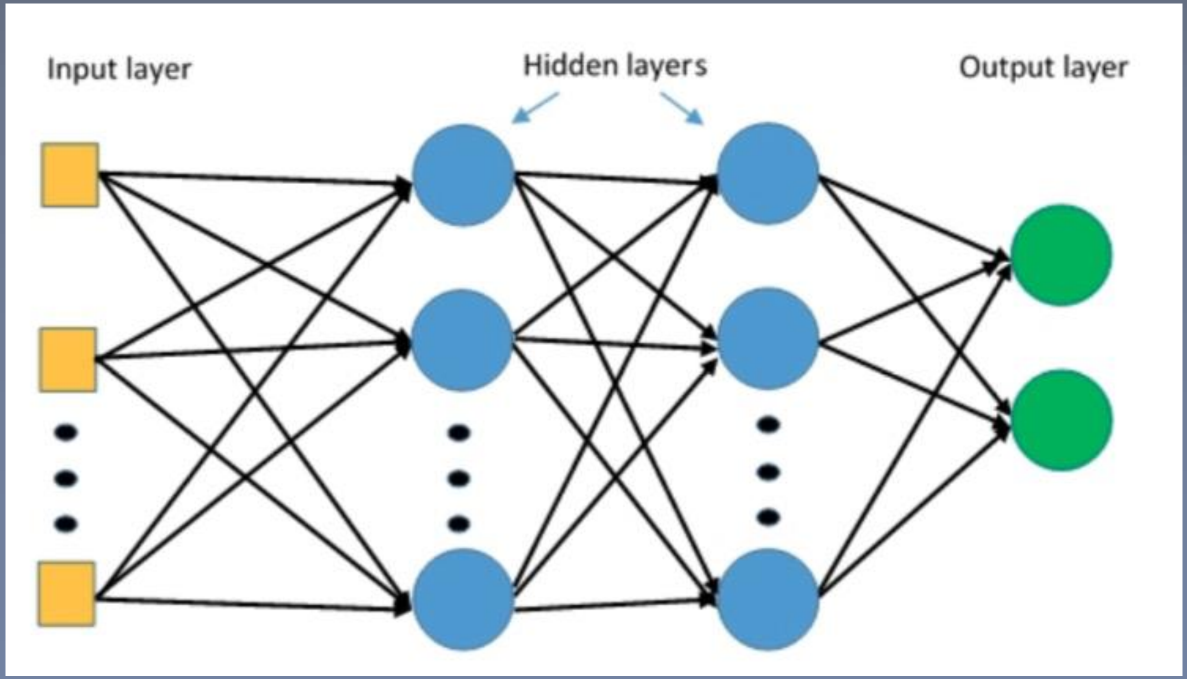
$$= \mathrm{n} * (1 * 1)$$

$$= \mathrm{n}$$

$$\mathrm{std}(\mathrm{H}_{11}) = \sqrt{\mathbf{D}(\mathrm{H}_{11})} = \sqrt{n}$$

$$\mathbf{D}(\mathrm{H}_1) = n * D(X) * D(W) = 1$$

$$D(W) = \frac{1}{n} \Rightarrow \mathrm{std}(W) = \sqrt{\frac{1}{n}}$$




$X \quad W_1 \quad H_1 \quad W_2 \quad H_2 \quad W_3 \quad out$

# Xavier初始化

Xavier Initialization



**方差一致性**：保持数据尺度维持在恰当范围，通常方差为1

激活函数：饱和函数，如Sigmoid，Tanh

$$n_i * D(W) = 1$$

$$n_{i+1} * D(W) = 1$$

$$\Rightarrow D(W) = \frac{2}{n_i + n_{i+1}}$$

$$W \backsim U[-a,\ a]$$

$$D(W) = \frac{(-a-a)^2}{12} = \frac{(2a)^2}{12} = \frac{a^2}{3}$$

$$\frac{2}{n_i + n_{i+1}} = \frac{a^2}{3} \Rightarrow a = \frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}$$

$$\Rightarrow W \backsim U\left[-\frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}, \frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}\right]$$

参考文献：《Understanding the difficulty of training deep feedforward neural networks》

# Kaiming初始化

Kaiming Initialization

deepshare.net 深度之眼

**方差一致性**：保持数据尺度维持在恰当范围，通常方差为1

激活函数：ReLU及其变种

$$D(W) = \frac{2}{n_i}$$

$$D(W) = \frac{2}{(1+a^2) * n_i}$$

$$std(W) = \sqrt{\frac{2}{(1+a^2) * n_i}}$$

参考文献：《Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification》

# 十种初始化方法

Initialization Methods



1. Xavier均匀分布
2. Xavier正态分布
3. Kaiming均匀分布
4. Kaiming正态分布
5. 均匀分布
6. 正态分布
7. 常数分布
8. 正交矩阵初始化
9. 单位矩阵初始化
10. 稀疏矩阵初始化

```
nn.init.calculate_gain(nonlinearity,  param=None)
```

## nn.init.calculate_gain

主要功能：计算激活函数的**方差变化尺度**

主要参数

- **nonlinearity**: 激活函数名称
- **param**: 激活函数的参数，如Leaky ReLU的negative_slop

https://pytorch.org/docs/stable/nn.init.html

# 结 语

在这次课程中，学习了权值初始化方法的准则——方差一致性原则，以及Xavier和Kaiming

权值初始化方法

在下次课程中，我们将会学习

**损失函数**

deepshare.net
深度之眼

联系我们:
电话：18001992849
邮箱：service@deepshare.net
Q Q： 2677693114

公众号

客服微信