

法律声明

本课件包括演示文稿、示例、代码、题库、视频和声音等内容，深度之眼和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。

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

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文



deepshare.net

深度之眼

池化、线性、激活函数层

导师：余老师

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

目录

1/ 池化层——Pooling Layer

2/ 线性层——Linear Layer

3/ 激活函数层——Activation Layer

池化层

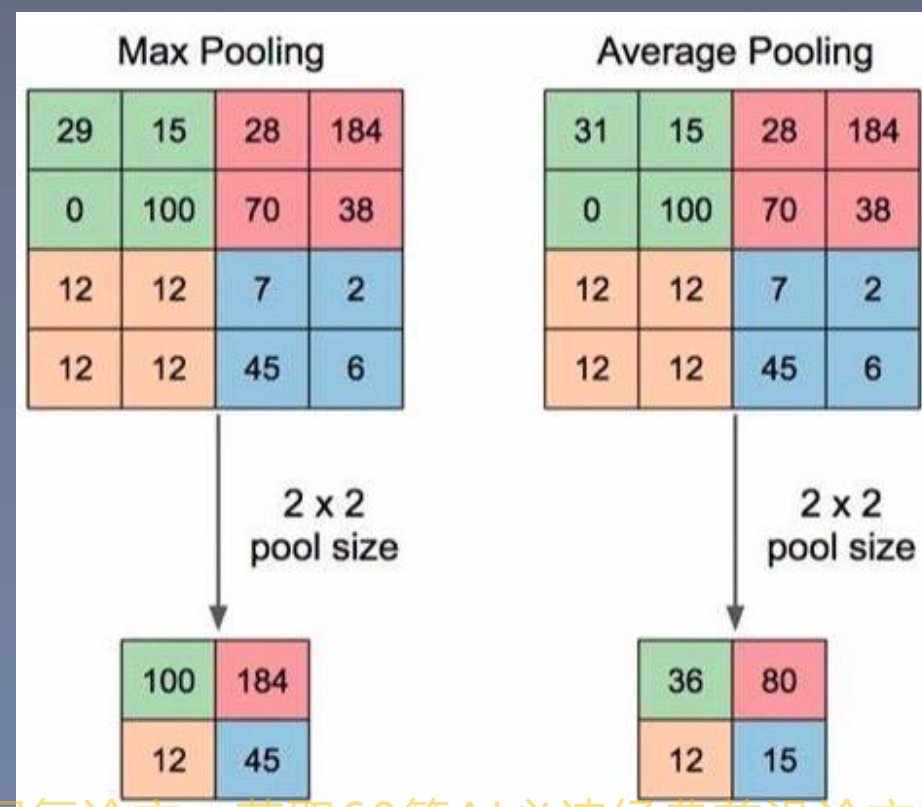
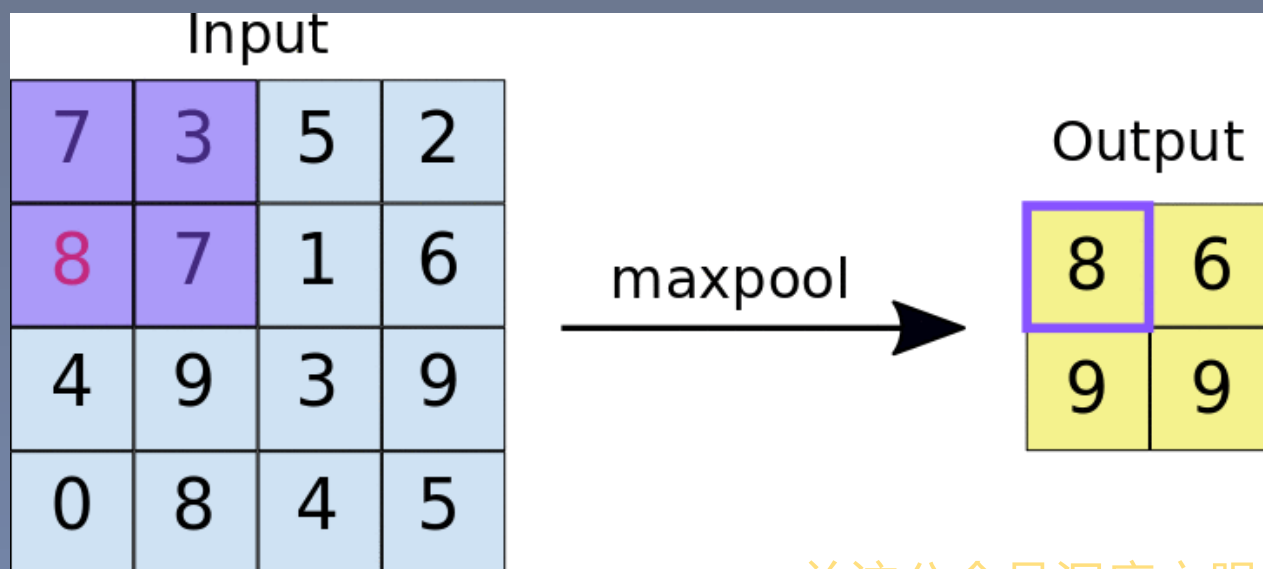
Pooling Layer

池化层

Pooling Layer

池化运算：对信号进行“**收集**”并“**总结**”，类似水池收集水资源，因而得名池化层

“收集”：多变少 “总结”：最大值/平均值



池化层

Pooling Layer

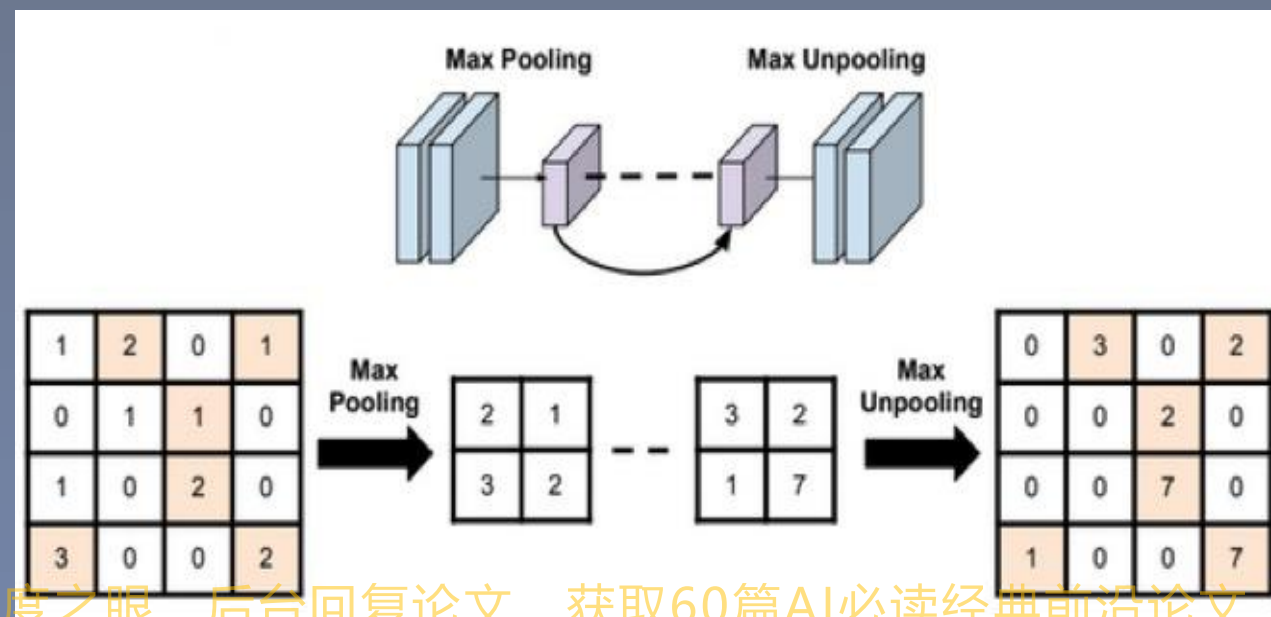
nn.MaxPool2d

功能：对二维信号（图像）进行最大值池化

主要参数：

- **kernel_size**：池化核尺寸
- **stride**：步长
- **padding**：填充个数
- **dilation**：池化核间隔大小
- **ceil_mode**：尺寸向上取整
- **return_indices**：记录池化像素索引

```
nn.MaxPool2d(kernel_size, stride=None,  
padding=0, dilation=1,  
return_indices=False,  
ceil_mode=False)
```



池化层

Pooling Layer



deepshare.net

深度之眼

nn.AvgPool2d

功能：对二维信号（图像）进行平均值池化

主要参数：

- **kernel_size**：池化核尺寸
- **stride**：步长
- **padding**：填充个数
- **ceil_mode**：尺寸向上取整
- **count_include_pad**：填充值用于计算
- **divisor_override**：除法因子

```
nn.AvgPool2d(kernel_size,  
              stride=None,  
              padding=0,  
              ceil_mode=False,  
              count_include_pad=True,  
              divisor_override=None)
```

池化层

Pooling Layer

nn.MaxUnpool2d

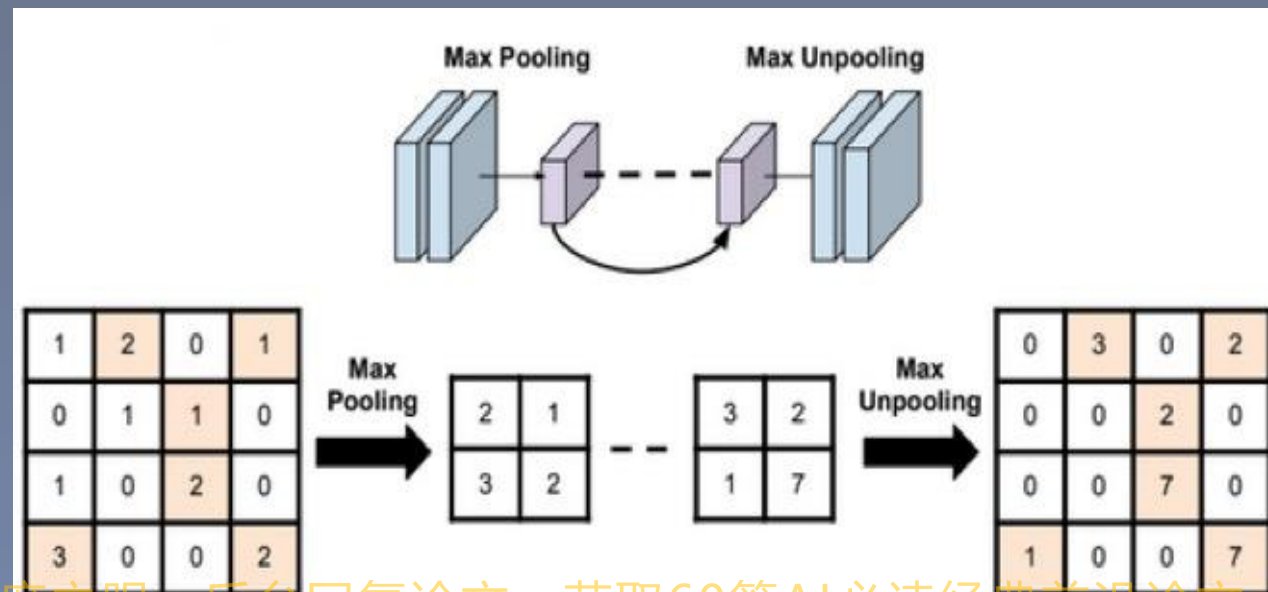
功能：对二维信号（图像）进行最大值池化上采样

主要参数：

- **kernel_size**：池化核尺寸
- **stride**：步长
- **padding**：填充个数

```
nn.MaxUnpool2d(kernel_size,  
                stride=None,  
                padding=0)
```

```
forward(self, input, indices, output_size=None)
```



线性层

Linear Layer

线性层又称全连接层，其每个神经元与上一层所有神经元相连

实现对前层的**线性组合**，**线性变换**

Input = [1, 2, 3]

shape = (1, 3)

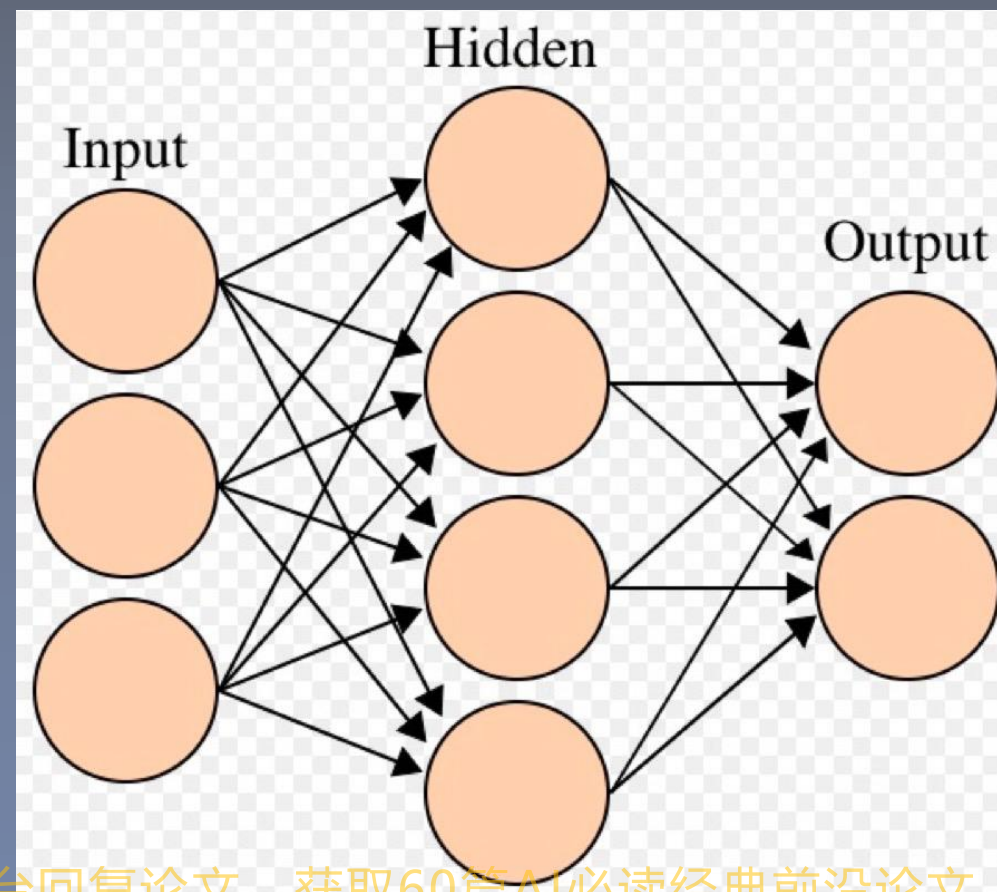
$$W_0 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

shape = (3, 4)

Hidden = Input * W_0

shape = (1, 4)

= [6, 12, 18, 24]



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

线性层

Linear Layer



deepshare.net

深度之眼

nn.Linear

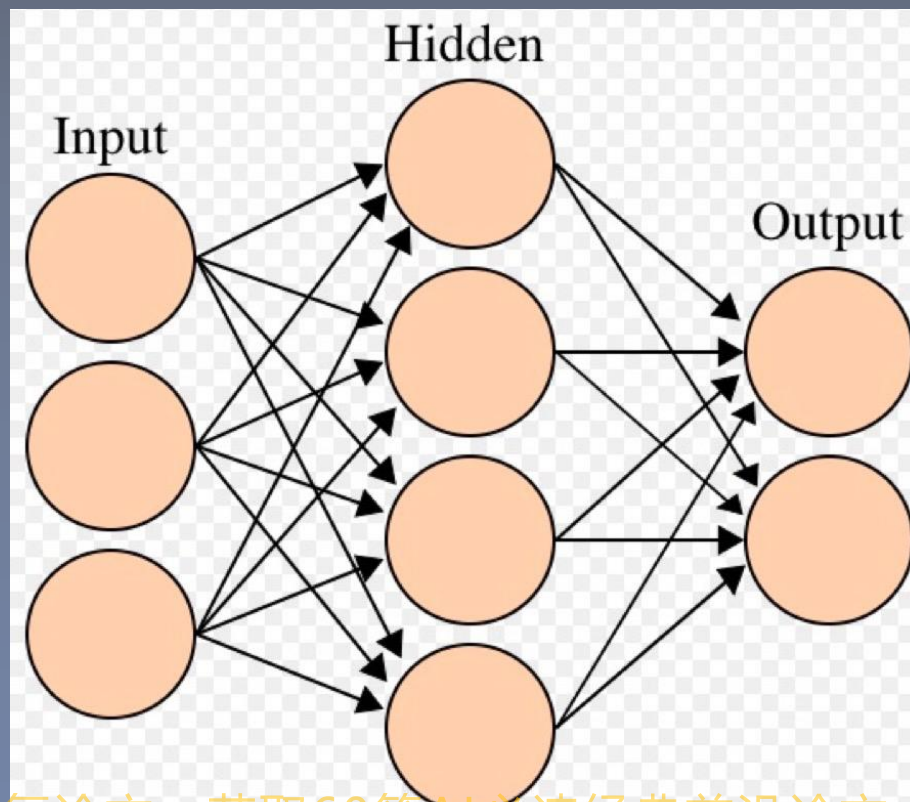
功能：对一维信号（向量）进行线性组合

主要参数：

- **in_features**：输入结点数
- **out_features**：输出结点数
- **bias**：是否需要偏置

计算公式： $y = xW^T + bias$

```
nn.Linear(in_features, out_features, bias=True)
```



激活函数层

Activation Layer

激活函数对特征进行非线性变换，赋予多层神经网络具有**深度**的意义

$$H_1 = X * W_1$$

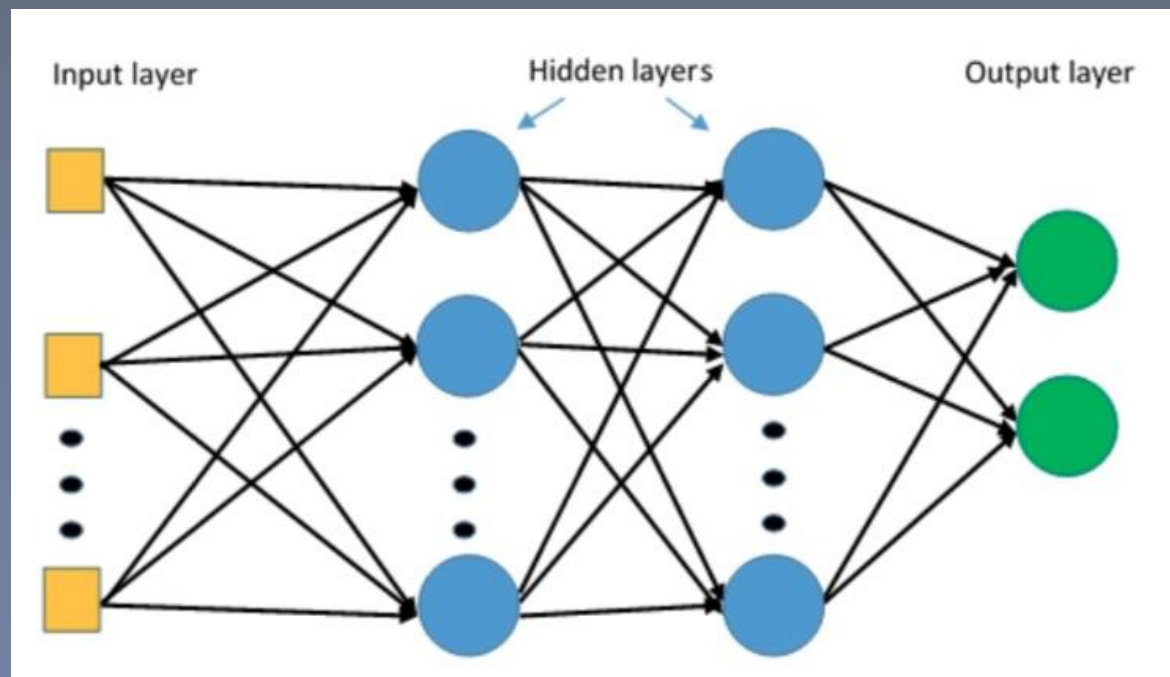
$$H_2 = H_1 * W_2$$

$$Output = H_2 * W_3$$

$$= H_1 * W_2 * W_3$$

$$= X * (W_1 * W_2 * W_3)$$

$$= X * W$$



X W_1 H_1 W_2 H_2 W_3 out
关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

激活函数层

Activation Layer



deepshare.net

深度之眼

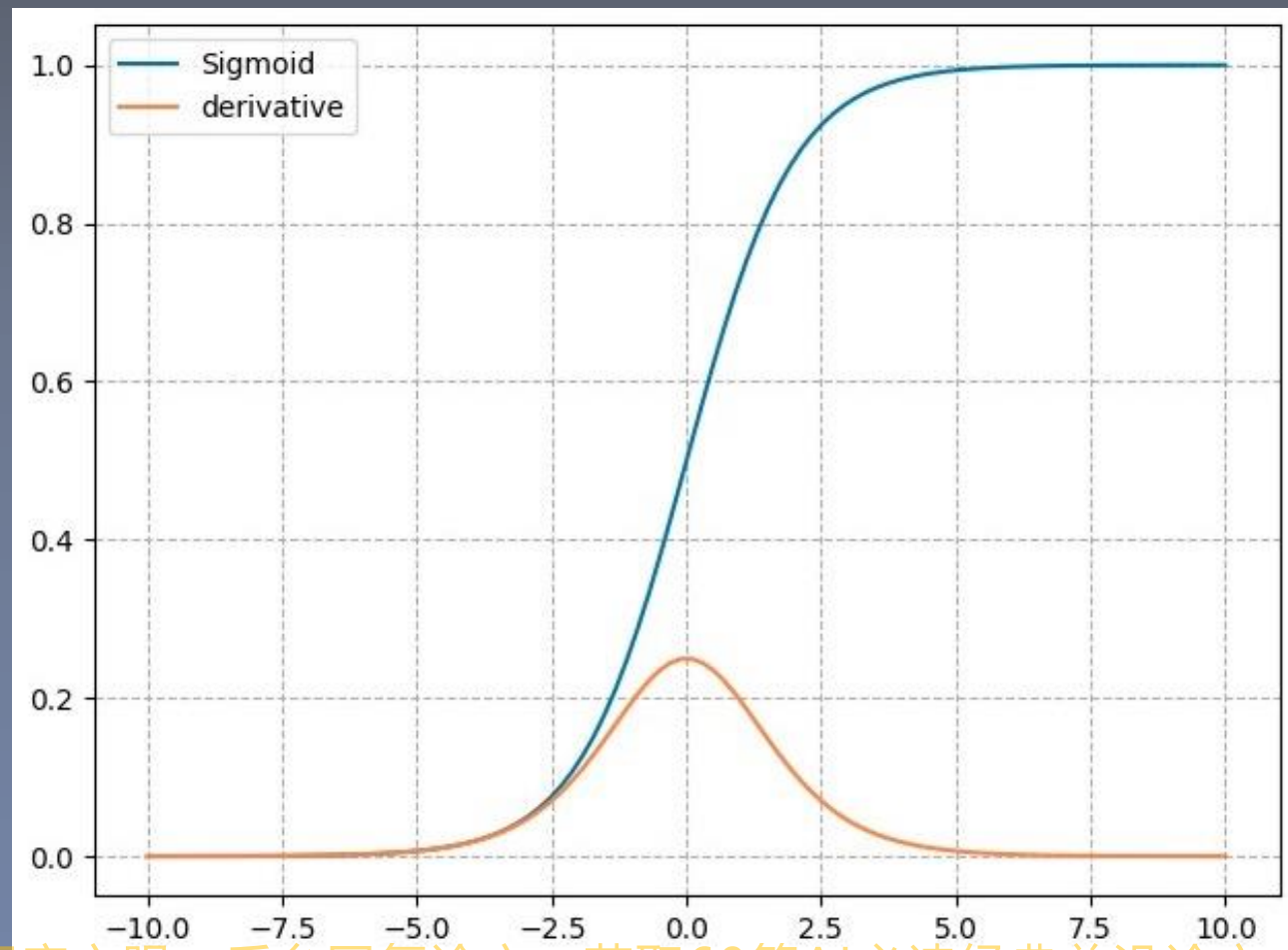
nn.Sigmoid

计算公式: $y = \frac{1}{1+e^{-x}}$

梯度公式: $y' = y * (1 - y)$

特性:

- 输出值在(0,1), 符合概率
- 导数范围是[0, 0.25], 易导致梯度消失
- 输出为非0均值, 破坏数据分布



关注公众号深度之眼, 后台回复论文, 获取60篇AI必读经典前沿论文

激活函数层

Activation Layer

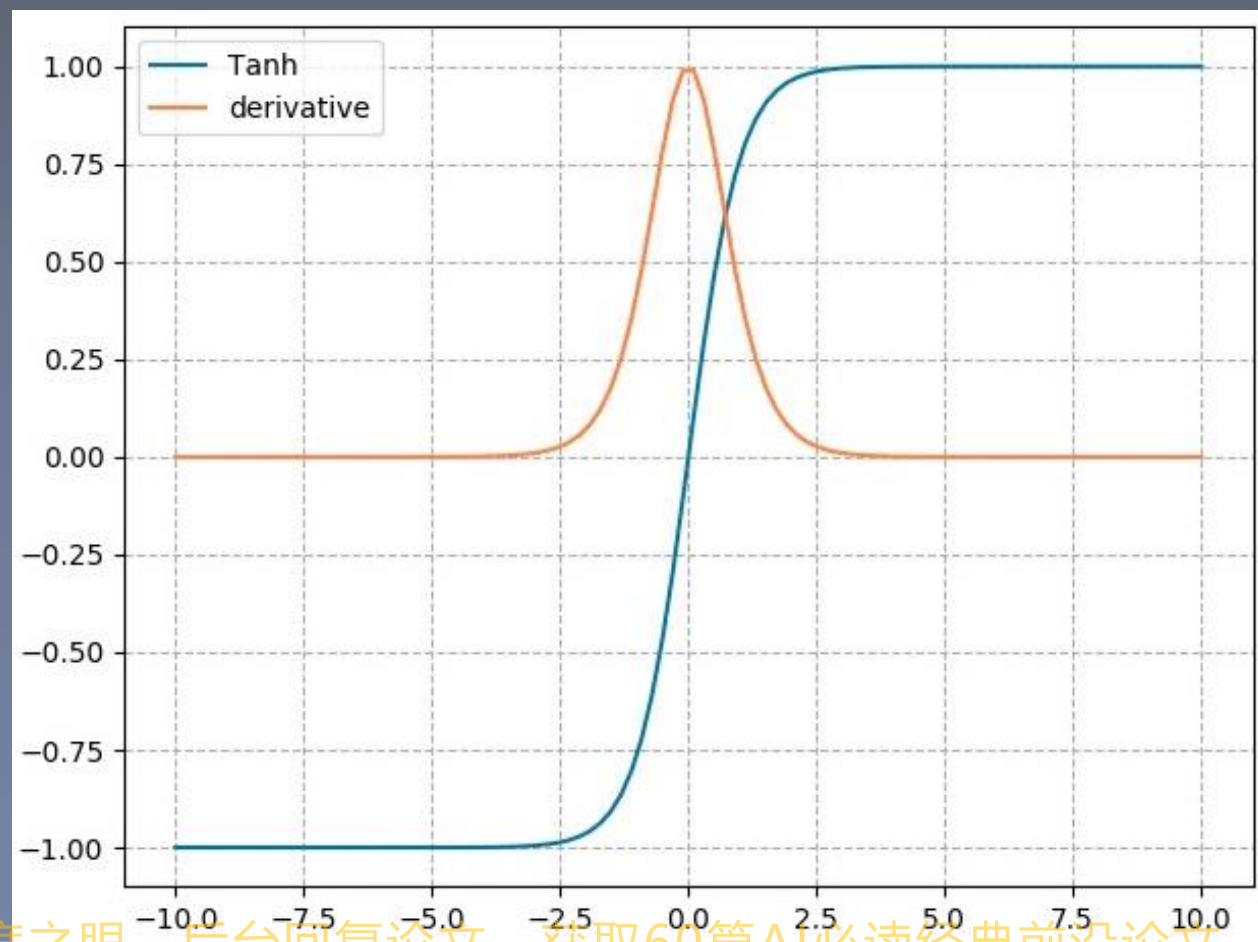
nn.tanh

计算公式: $y = \frac{\sin x}{\cos x} = \frac{e^x - e^{-x}}{e^+ + e^{-x}} = \frac{2}{1 + e^{-2x}} + 1$

梯度公式: $y' = 1 - y^2$

特性:

- 输出值在 $(-1, 1)$, 数据符合0均值
- 导数范围是 $(0, 1)$, 易导致梯度消失



激活函数层

Activation Layer

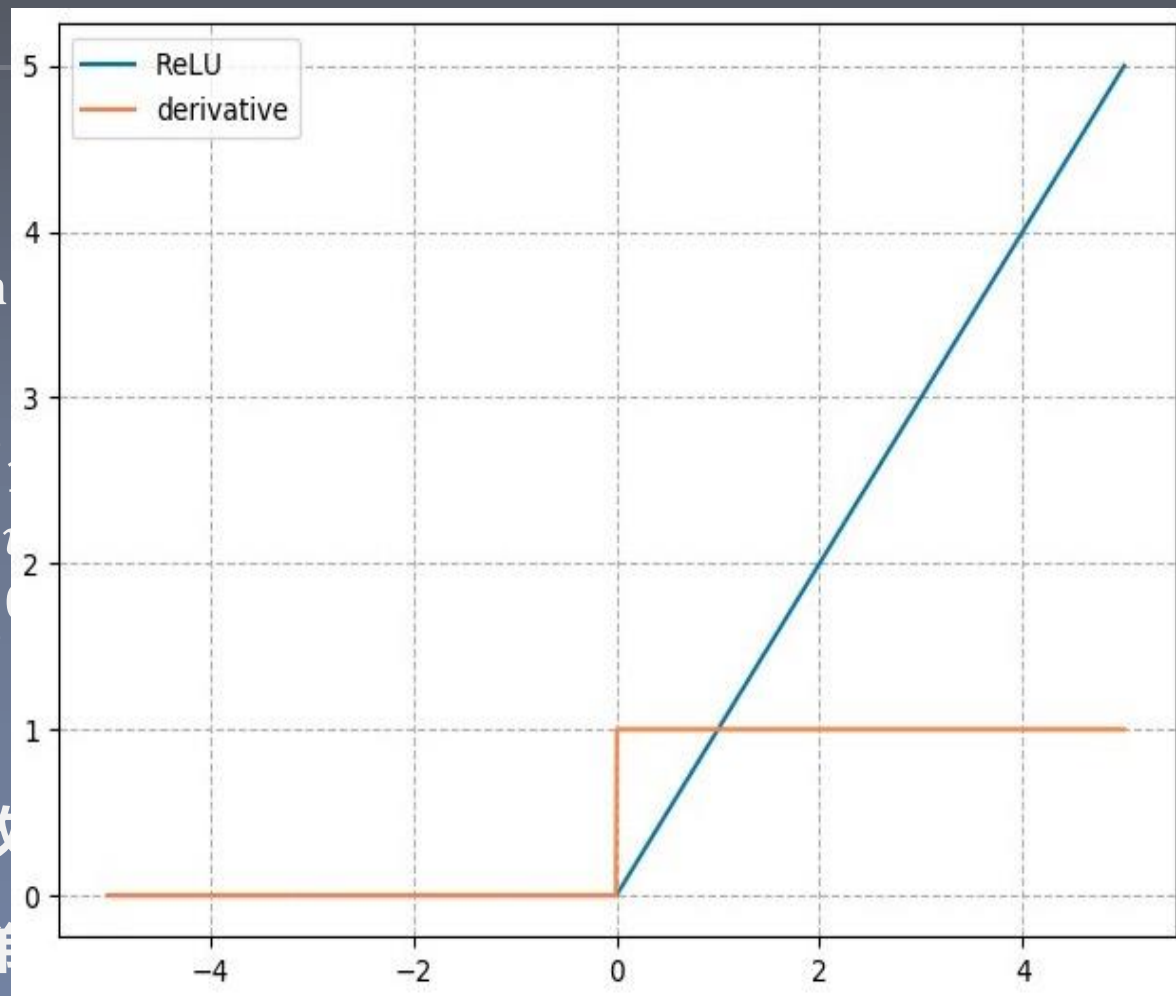
nn.ReLU

计算公式: $y = \max(0, x)$

梯度公式: $y' = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$

特性:

- 输出值均为正数
- 导数是1,缓解梯度爆炸



激活函数层

Activation Layer

nn.LeakyReLU

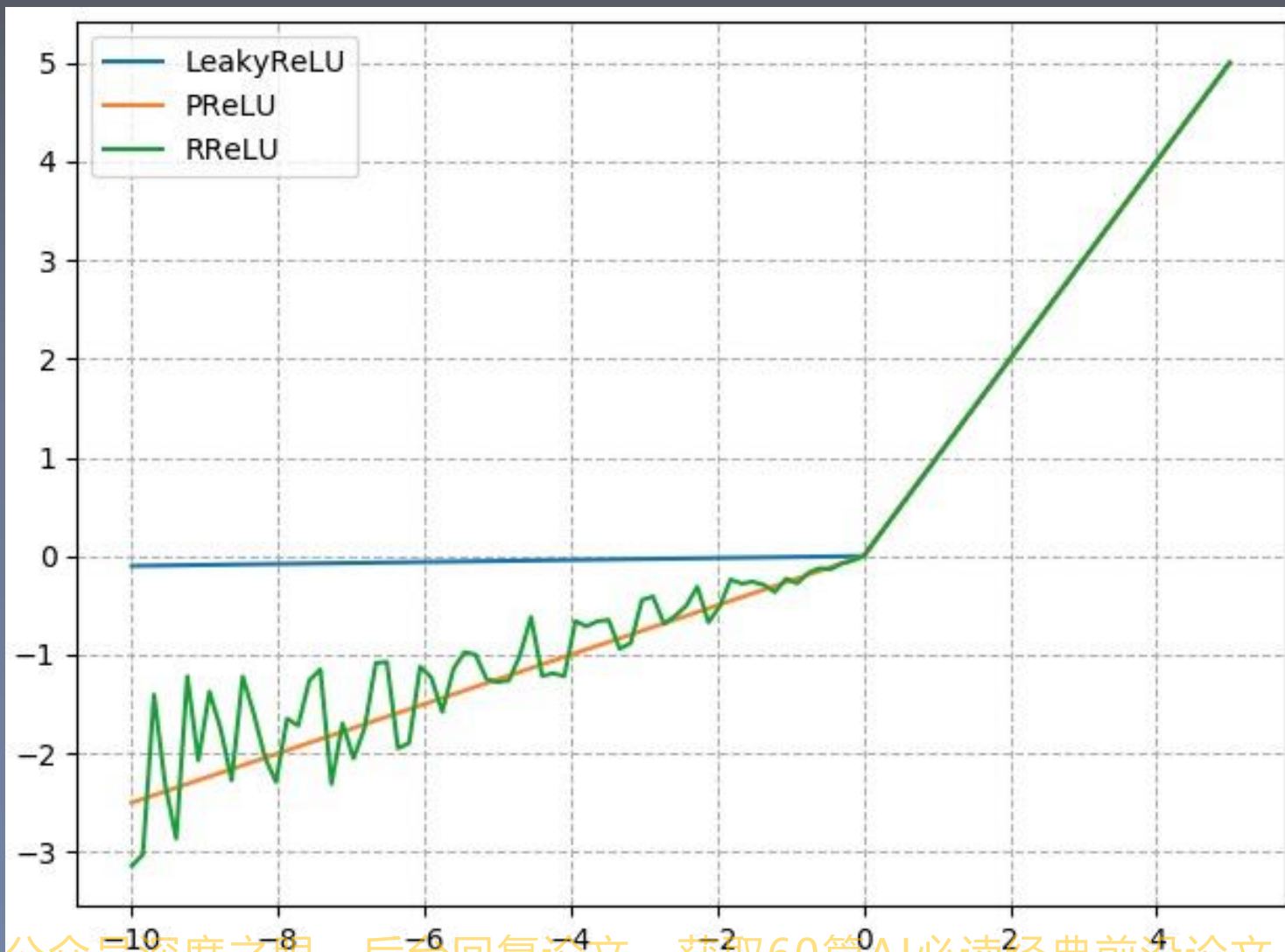
- **negative_slope**: 负半轴斜率

nn.PReLU

- **init**: 可学习斜率

nn.RReLU

- **lower**: 均匀分布下限
- **upper**: 均匀分布上限



—— 结 语 ——

在这次课程中，学习了nn模块中池化层，线性层
和激活函数层

在下次课程中，我们将会学习

网络层权值的初始化



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文



deepshare.net

深度之眼

联系我们:

电话: 18001992849

邮箱: service@deepshare.net

QQ: 2677693114



公众号



客服微信

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文