

# 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，小象学院拥有完全知识产权的权利；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。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将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：大数据分析挖掘

■ 新浪微博：ChinaHadoop



# 卷积神经网络—基础篇

---

主讲人： 李伟

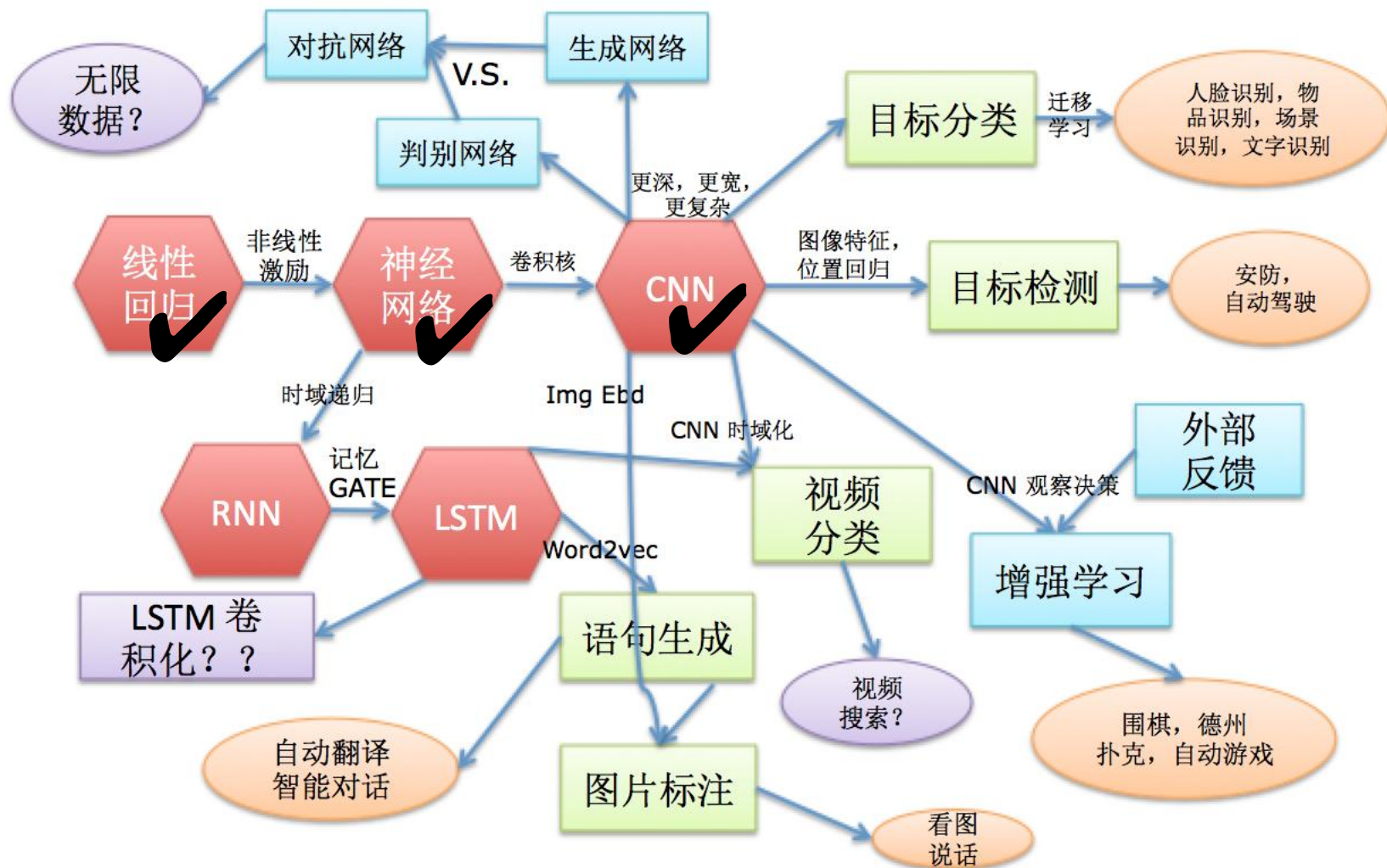
纽约城市大学博士

主要研究深度学习，计算机视觉，人脸计算  
多篇重要研究文章作者，重要会议期刊审稿人

微博ID: weightlee03 (相关资料分享)

GitHub ID: wiibrew (课程代码发布)

# 结构



# 提纲

---

- 1. 链式反向梯度传导
- 2. 卷积神经网络-卷积层
- 3. 卷积神经网络-功能层
- 4. 实例：卷积神经网络MNIST分类

# 期待目标

---

- 1. 清楚神经网络优化原理，掌握反向传播计算
- 2. 掌握卷积神经网络卷积层的结构特点，关键参数，层间的连接方式。
- 3. 了解不同卷积神经网络功能层的作用，会进行简单的卷积神经网络结构设计。
- 4. 能够运行Tensorflow 卷积神经网络MNIST

# 提纲

---

- 1. 链式反向梯度传导
- 2. 卷积神经网络-卷积层
- 3. 卷积神经网络-功能层
- 4. 实例：卷积神经网络MNIST分类

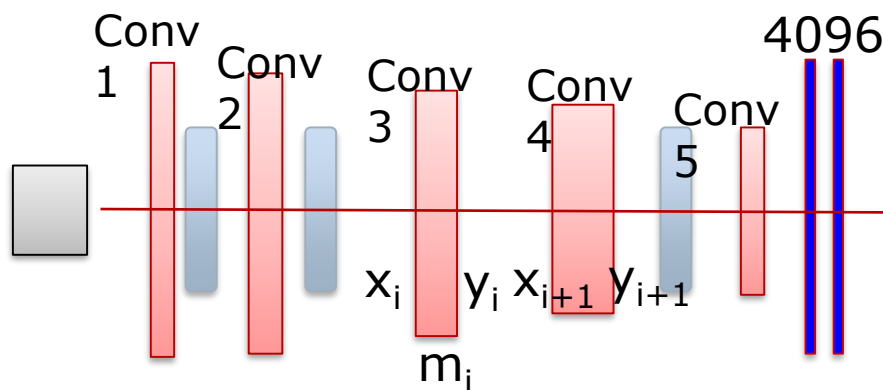
# 链式反向梯度传导

## □ 1. 链式法则的计算

$$y = f(x), z = g(y)$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x}$$

## □ 2. 神经网络中链式法则

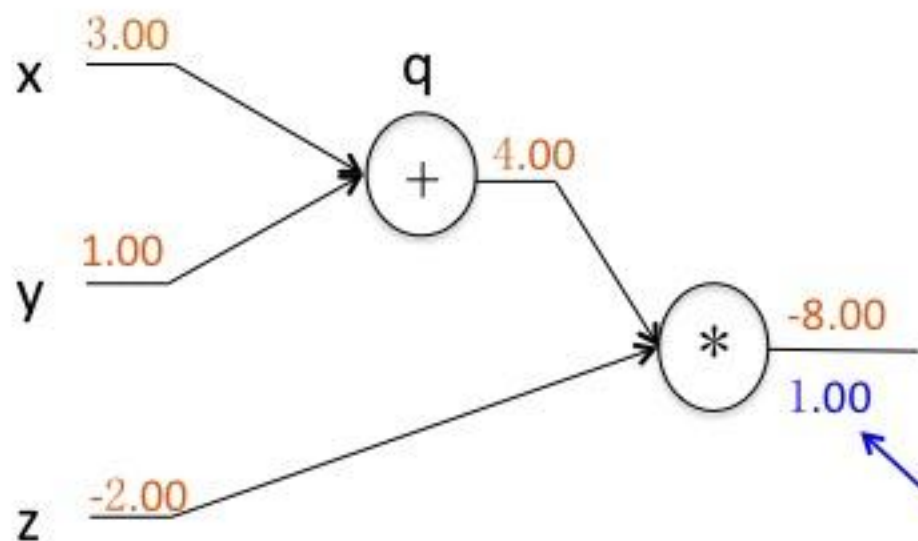


$$\frac{\partial \text{Loss}}{\partial m_i} = \frac{\partial \text{Loss}}{\partial y_i} \cdot \frac{\partial y_i}{\partial m_i} = \Delta y_i \cdot \frac{\partial y_i}{\partial m_i}$$
$$\Delta y_i = \Delta x_{i+1} \frac{\partial x_{i+1}}{\partial y_i} = \Delta y_{i+1} \cdot \frac{\partial y_{i+1}}{\partial x_{i+1}} \cdot \frac{\partial x_{i+1}}{\partial y_i}$$

链式法则:  $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x}$

$$f(x, y, z) = (x + y) * z$$

目标:  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial z} = q, \frac{\partial f}{\partial q} = z$$

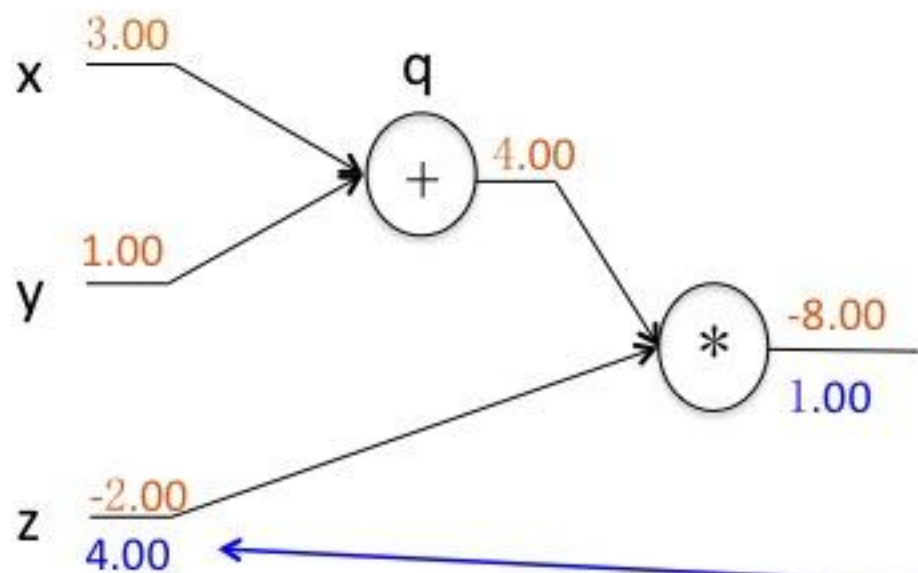
$$\frac{\partial f}{\partial f} = 1$$



链式法则:  $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x}$

$$f(x, y, z) = (x + y) * z$$

目标:  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

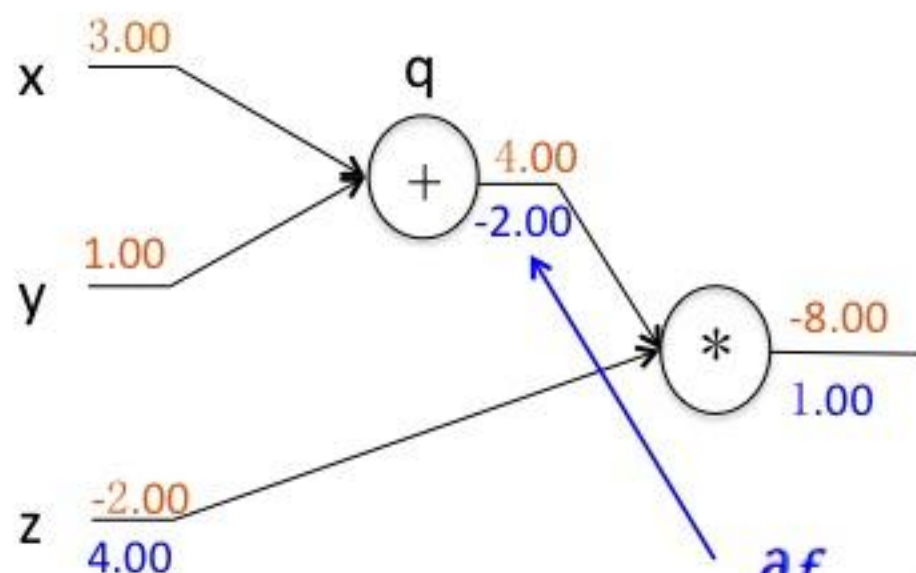
$$f = qz \quad \frac{\partial f}{\partial z} = q, \frac{\partial f}{\partial q} = z$$

$$\frac{\partial f}{\partial z} = q = 4$$

链式法则:  $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x}$

$$f(x, y, z) = (x + y) * z$$

目标:  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

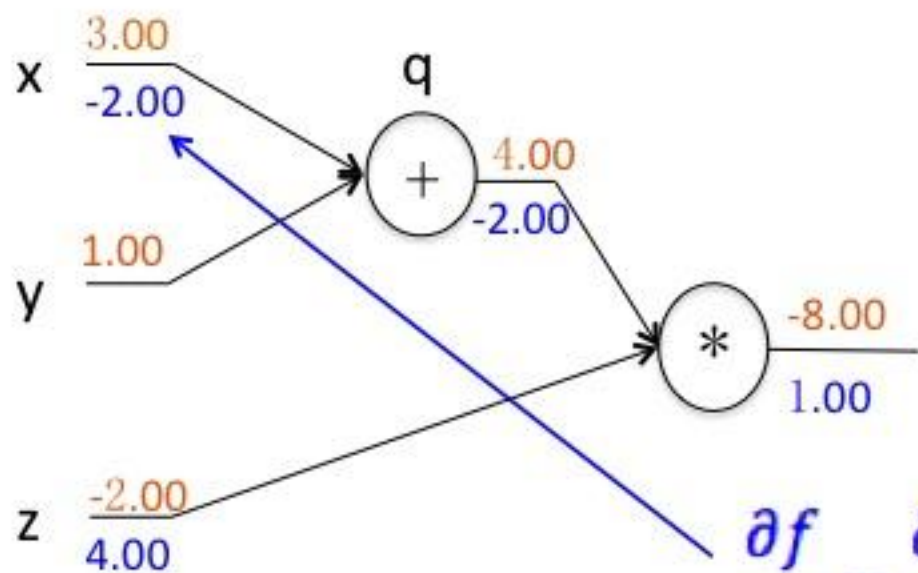
$$f = qz \quad \frac{\partial f}{\partial z} = q, \frac{\partial f}{\partial q} = z$$

$$\frac{\partial f}{\partial q} = z = -2$$

链式法则:  $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x}$

$$f(x, y, z) = (x + y) * z$$

目标:  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

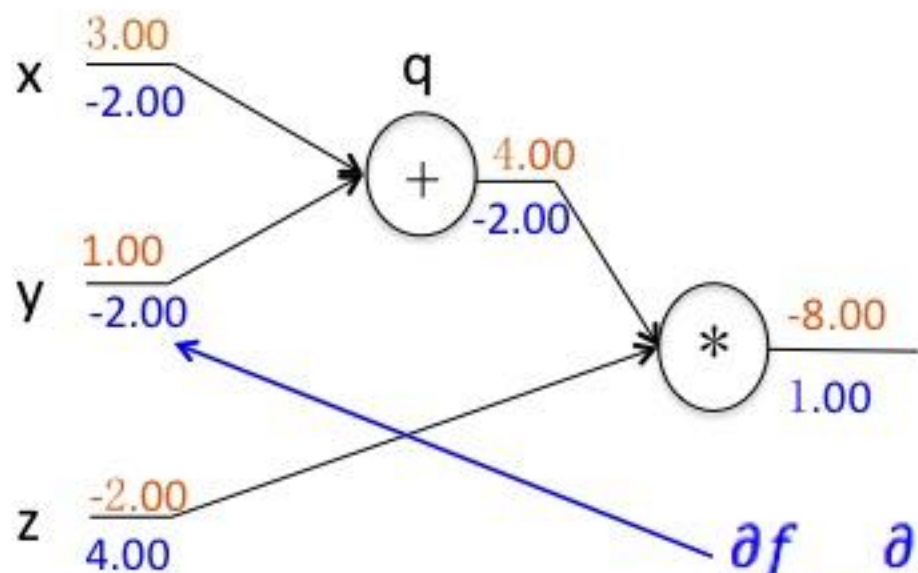
$$f = qz \quad \frac{\partial f}{\partial z} = q, \frac{\partial f}{\partial q} = z$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x} = -2 * 1$$

链式法则:  $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x}$

$$f(x, y, z) = (x + y) * z$$

目标:  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

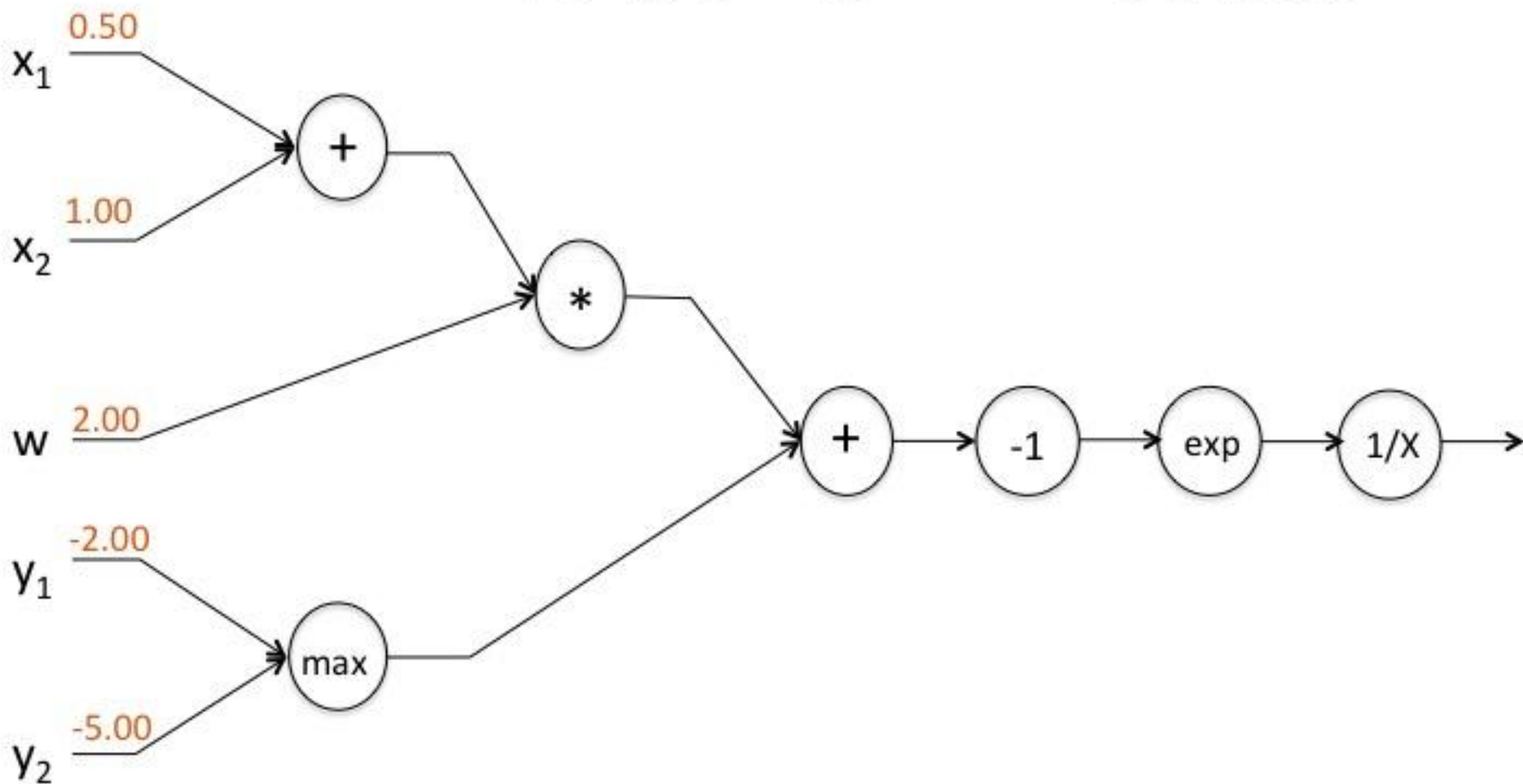


$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

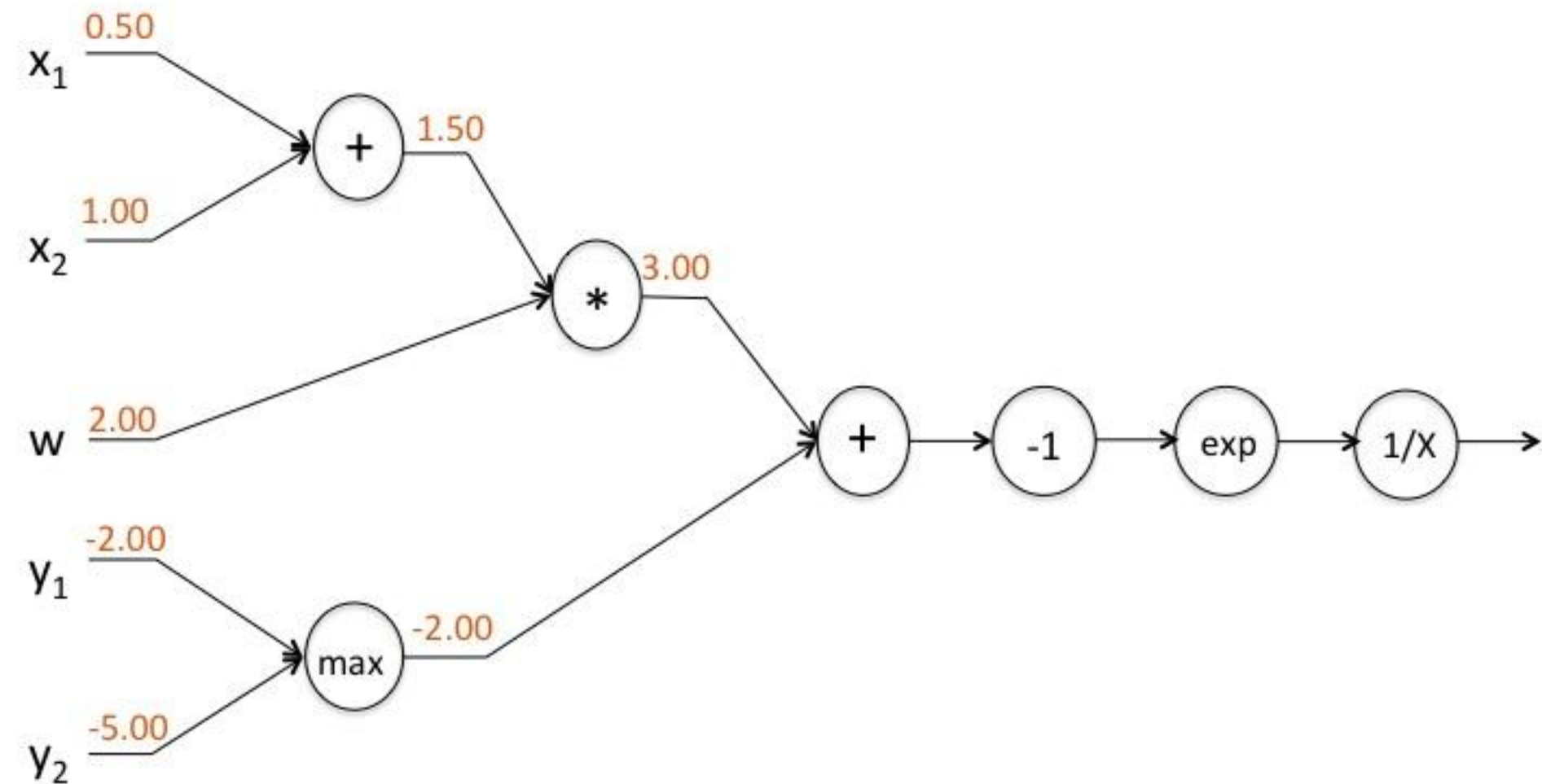
$$f = qz \quad \frac{\partial f}{\partial z} = q, \frac{\partial f}{\partial q} = z$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial y} = -2 * 1$$

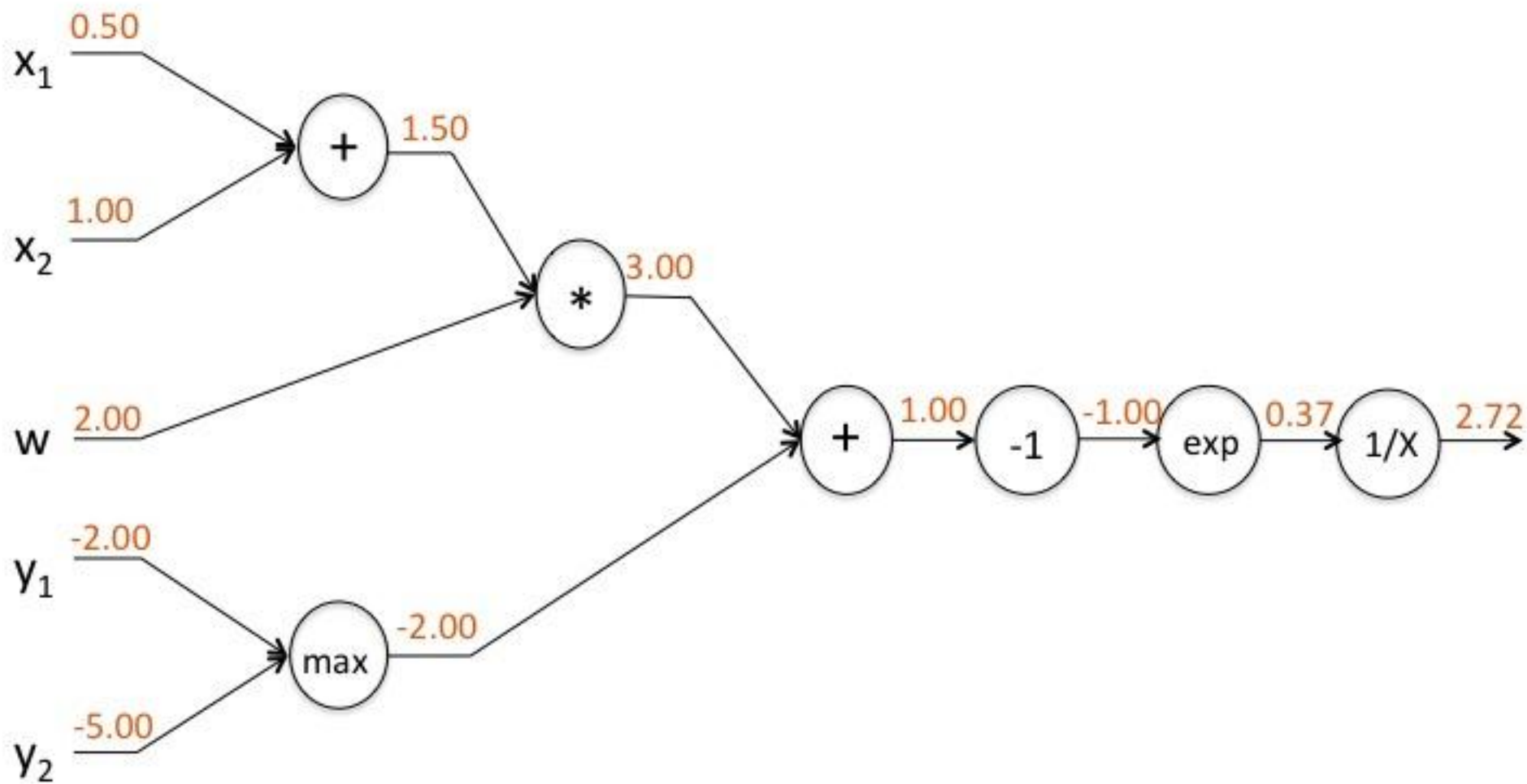
$$f(x, y, w) = \frac{1}{\exp\{-(x_1 + x_2)w + \max(y_1, y_2)\}}$$



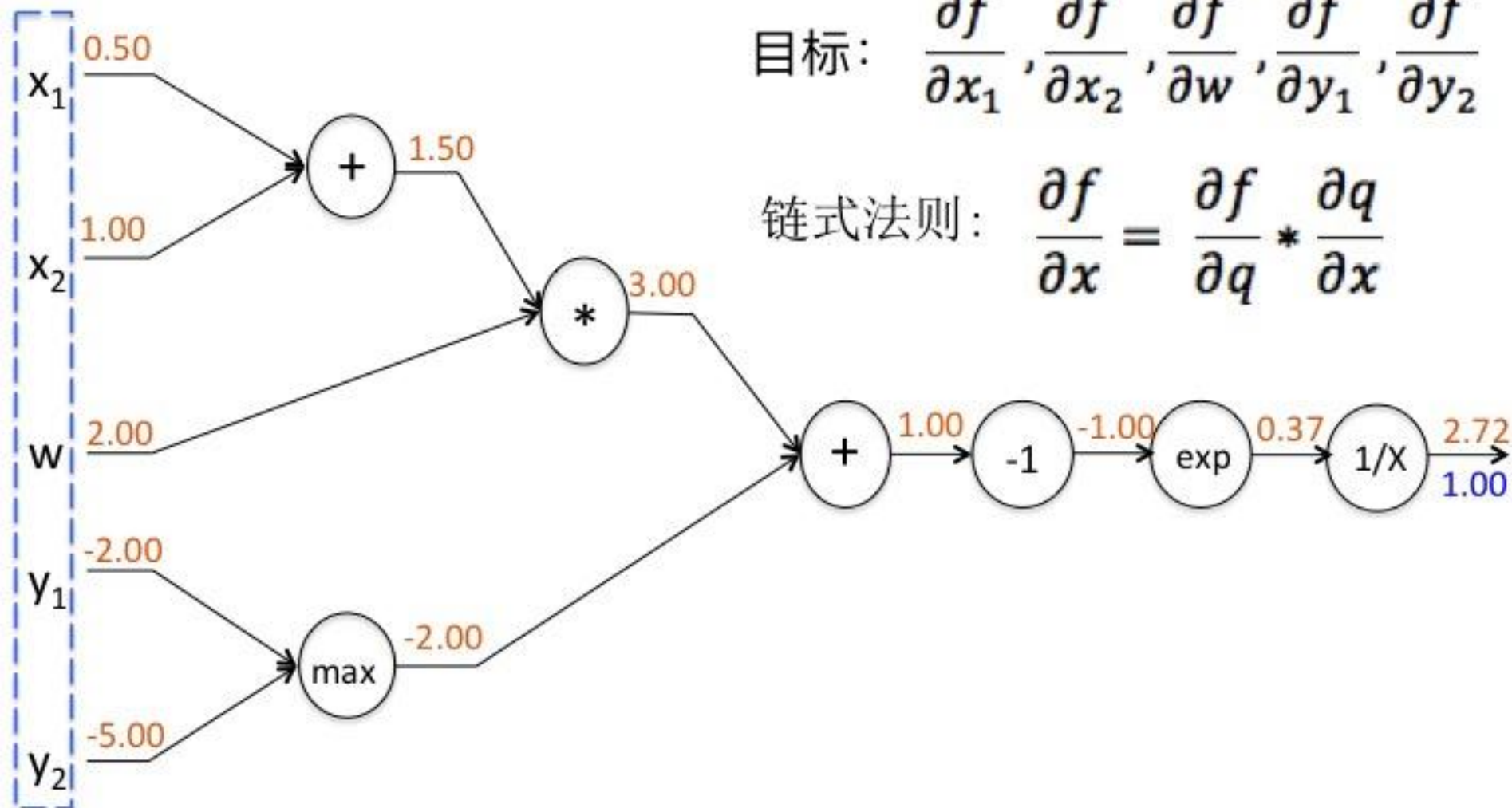
## Forward





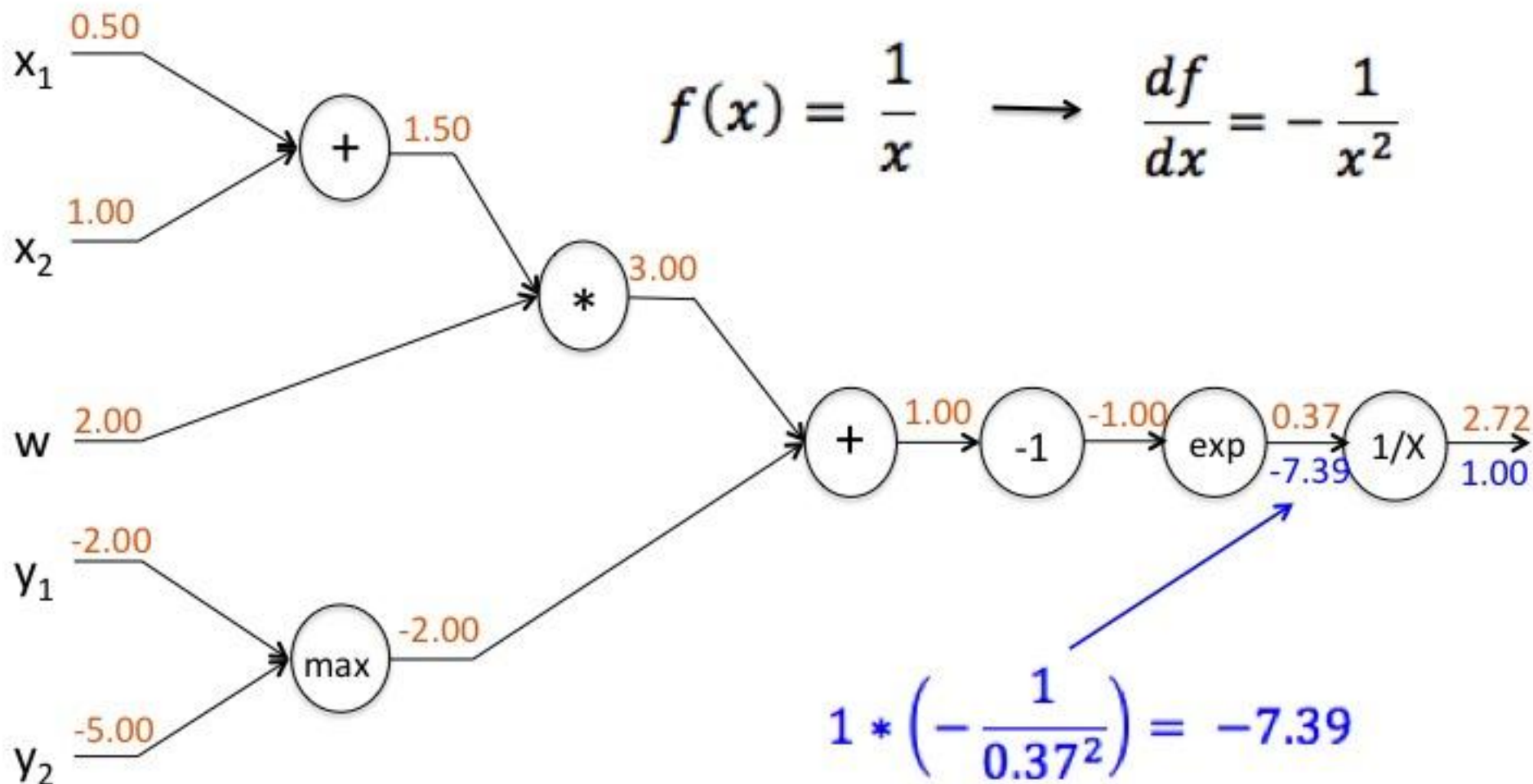


Backward:

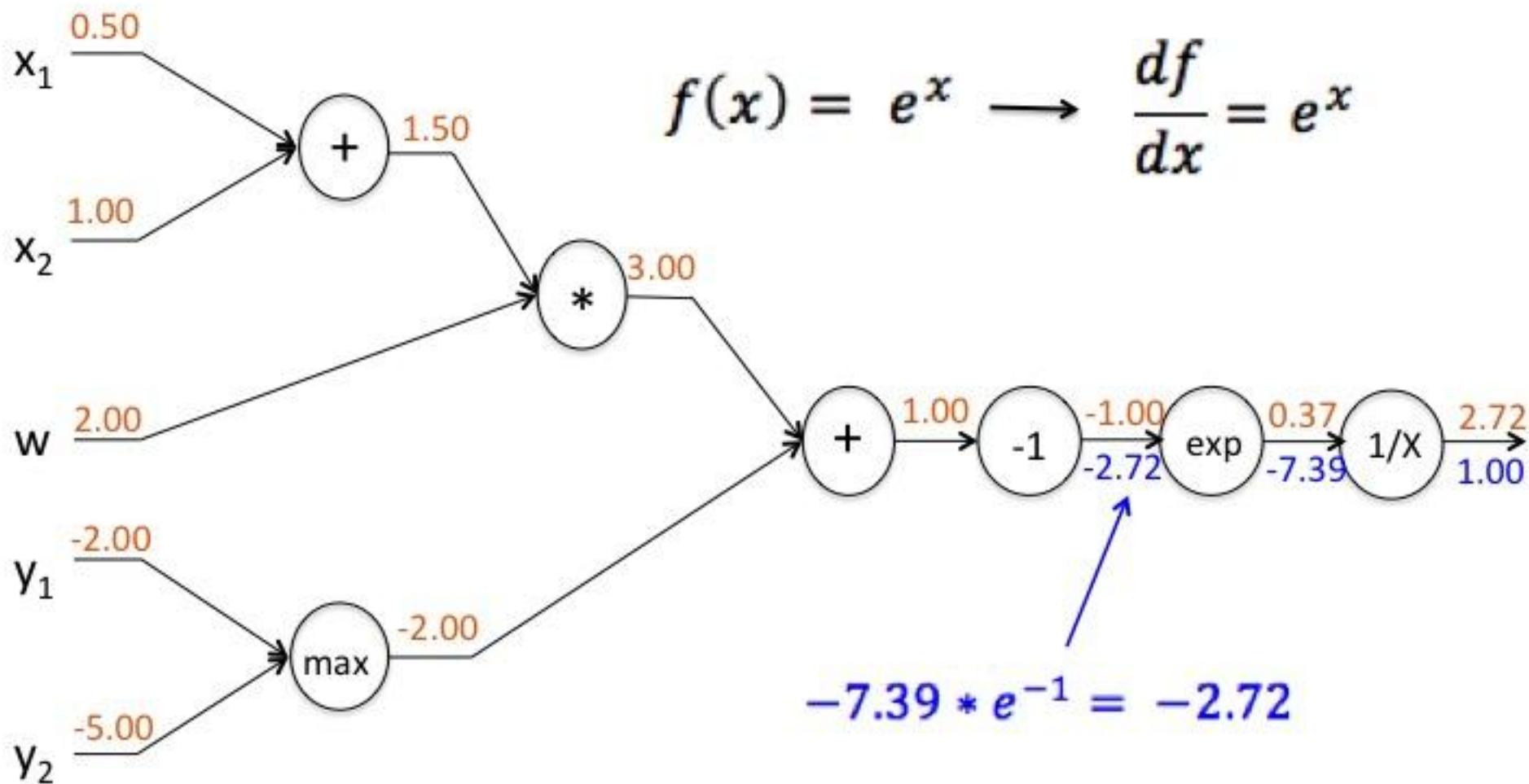




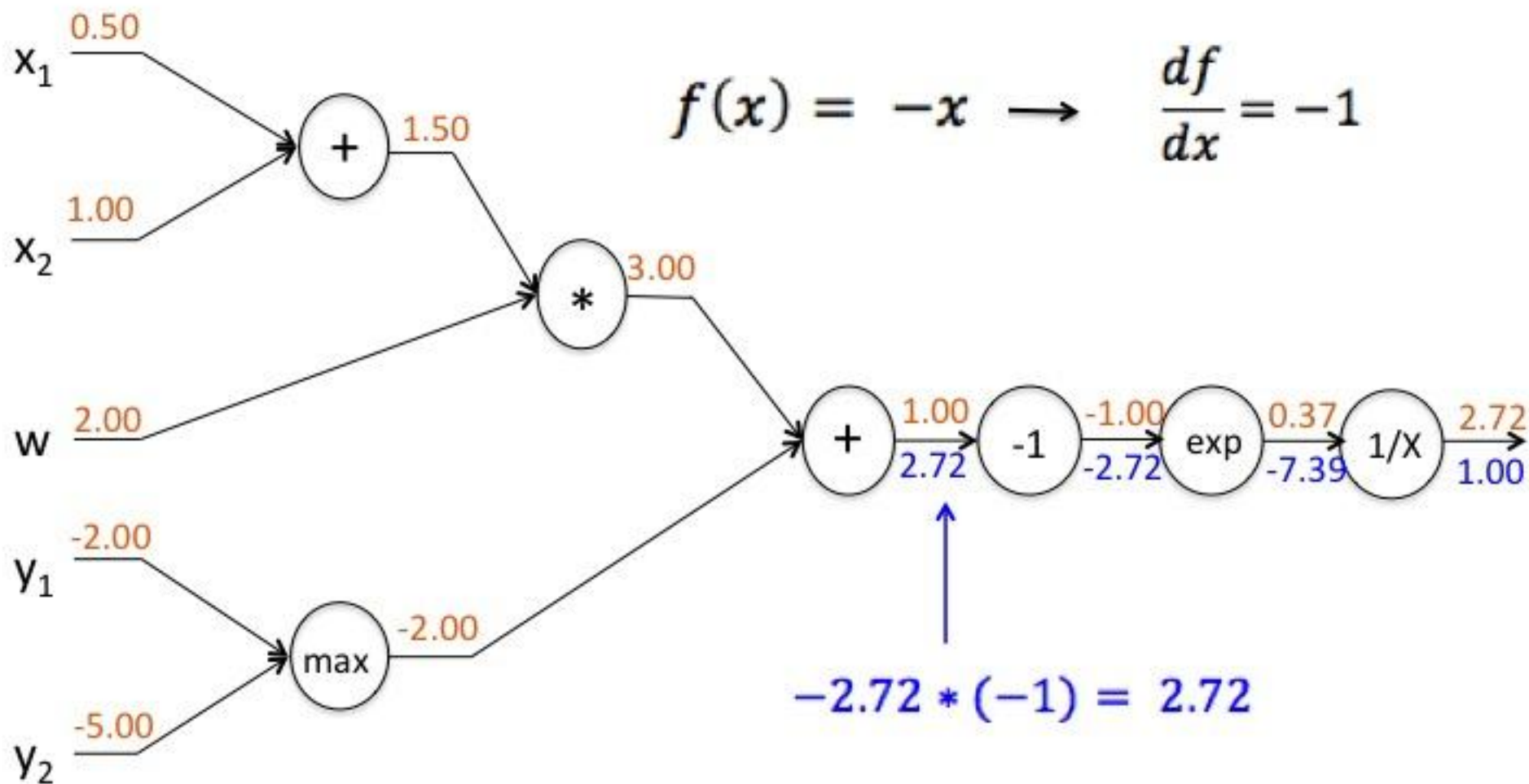
Backward :



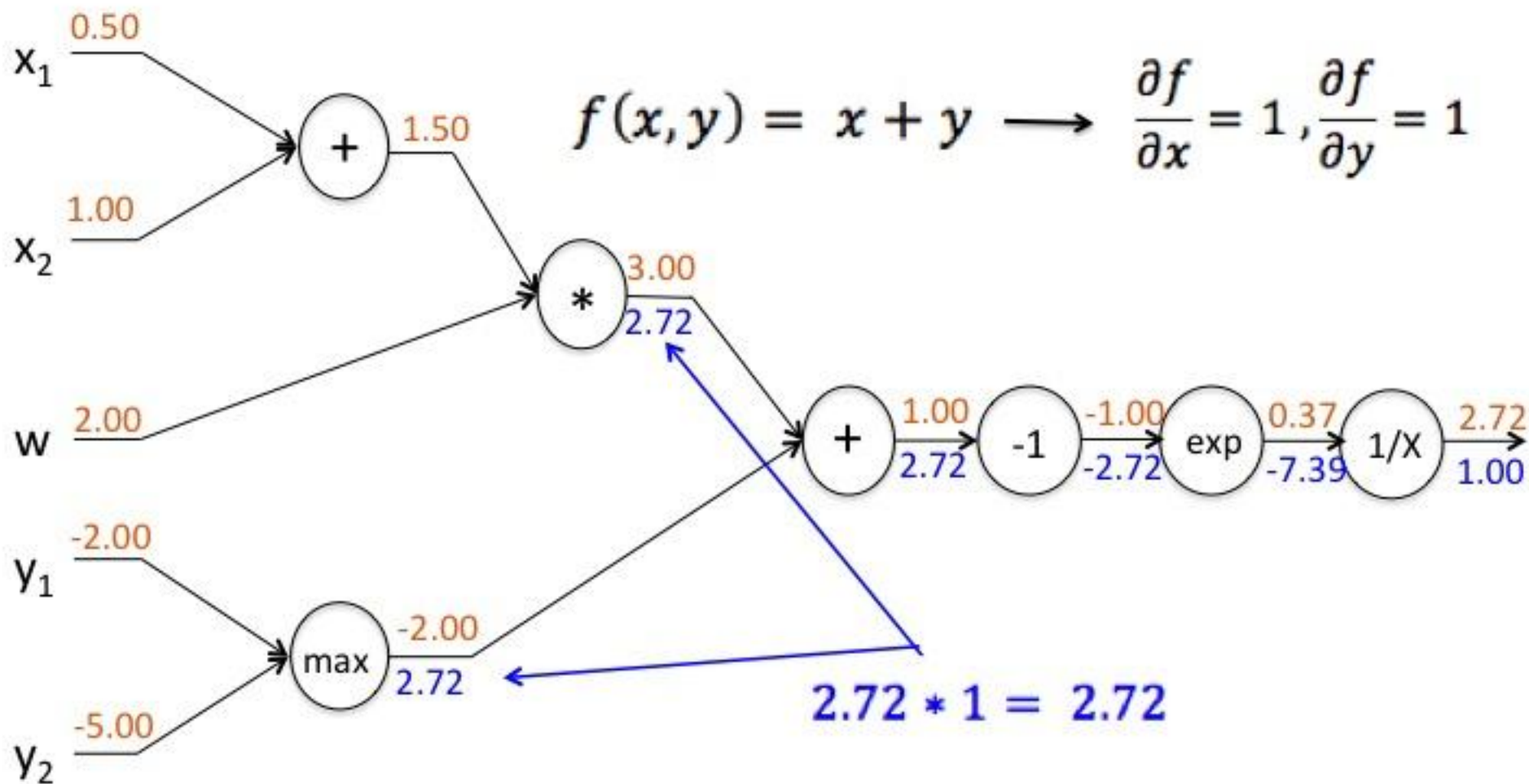
Backward:



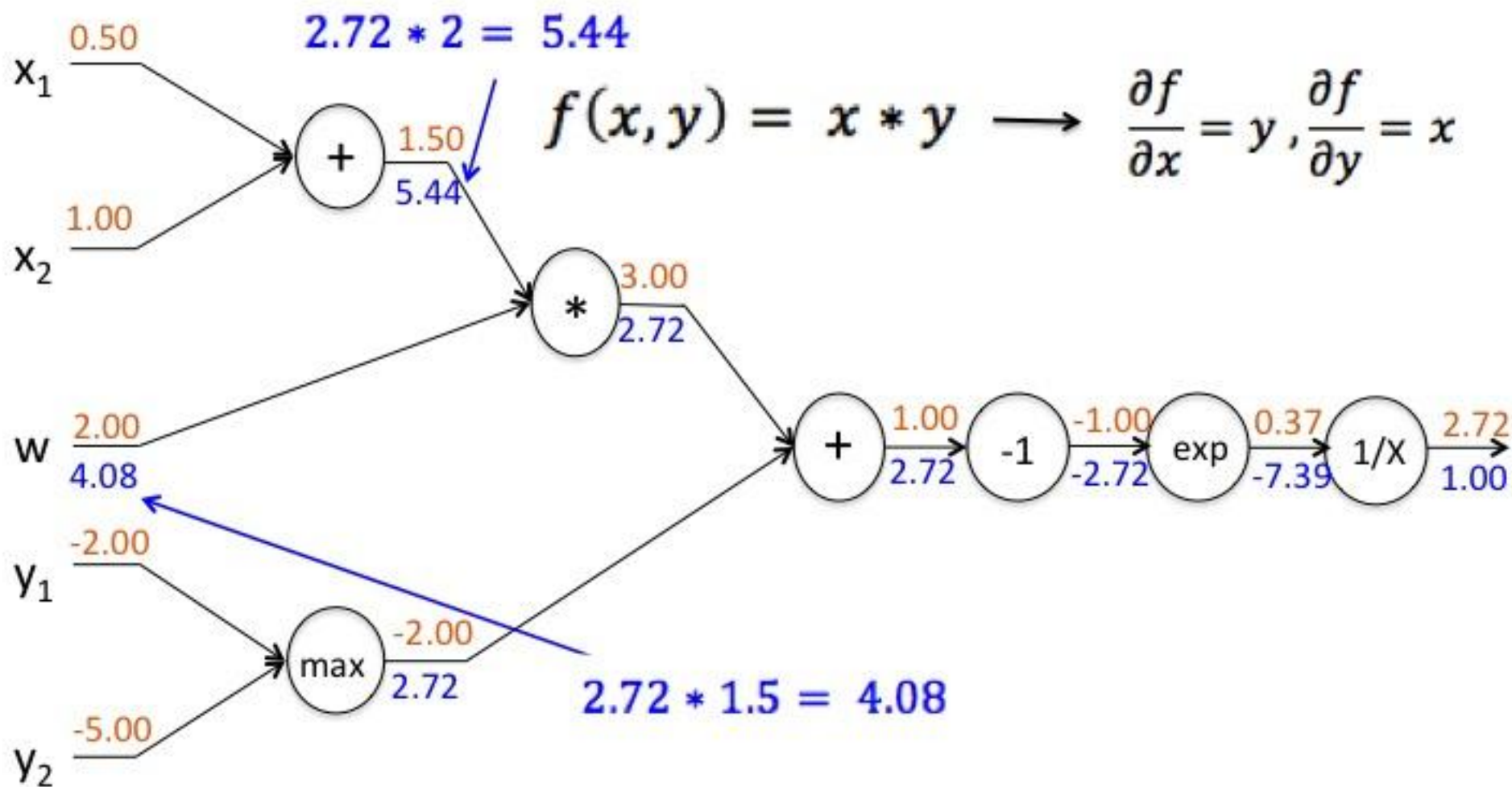
Backward:



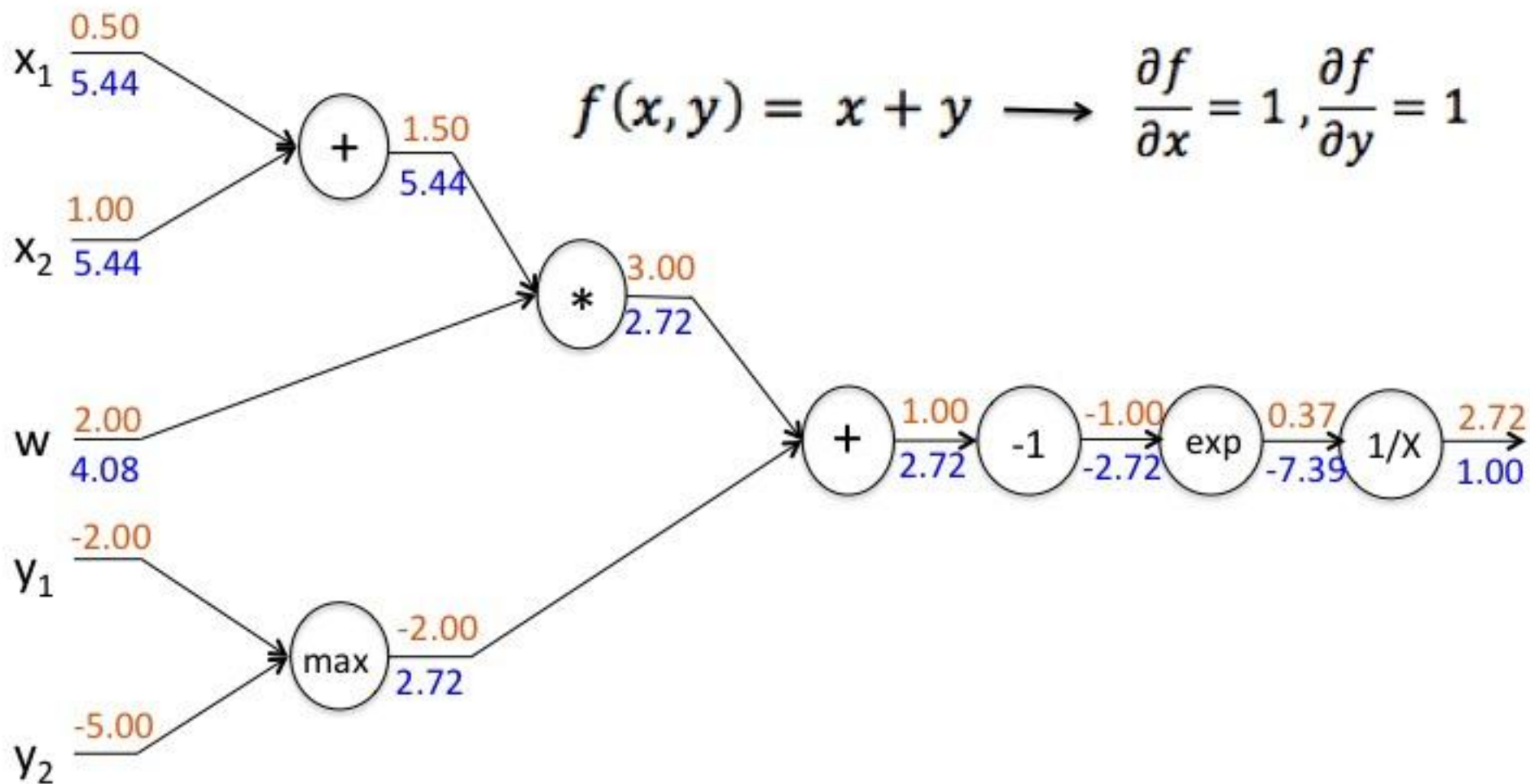
Backward:



Backward:

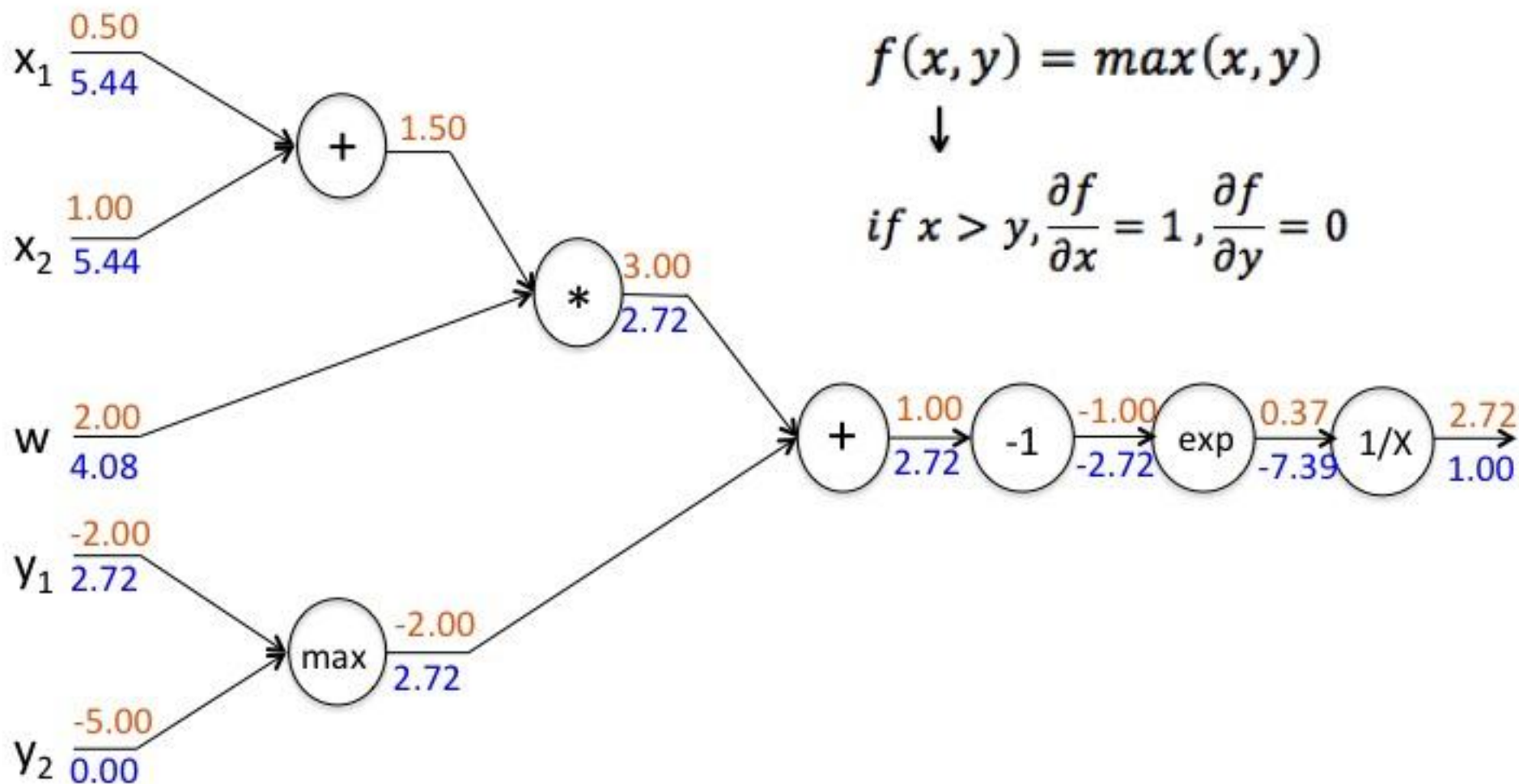


Backward:





Backward:



# 神经网络中链式法则

---

## □ 计算顺序

从loss向输入传播

## □ 导数存储

每层的导数 ( $\Delta y, \Delta x$ ) 结果进行存储，用于下一层导数的计算



# 提纲

---

- 1. 链式反向梯度传导
- 2. 卷积神经网络-卷积层
- 3. 卷积神经网络-功能层
- 4. 实例：卷积神经网络MNIST分类

# 卷积神经网络—卷积层

---

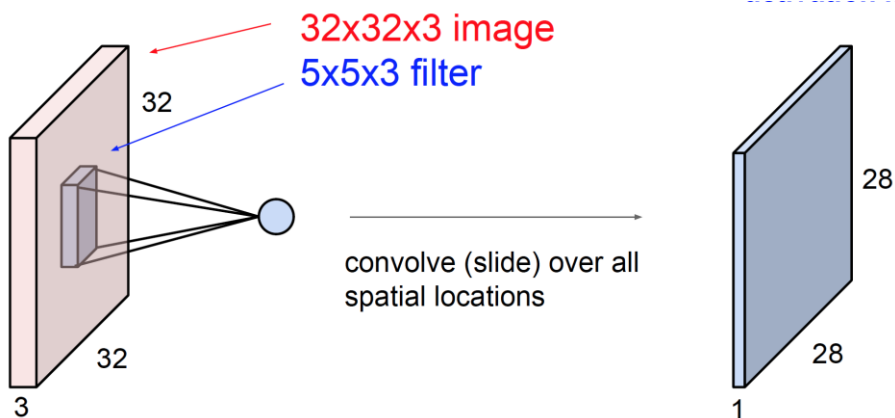
- 1. 什么是卷积层
- 2. 有什么组成
- 3. 受什么影响，有何特点
- 4. 怎样工作

# 卷积神经网络—卷积层

## □ 卷积层

### (Convolutional Layer)

1. 卷积神经网络基本结构
2. 由多个卷积核组合形成
3. 每个卷积核同输入数据卷积运算，形成新的特征“图”

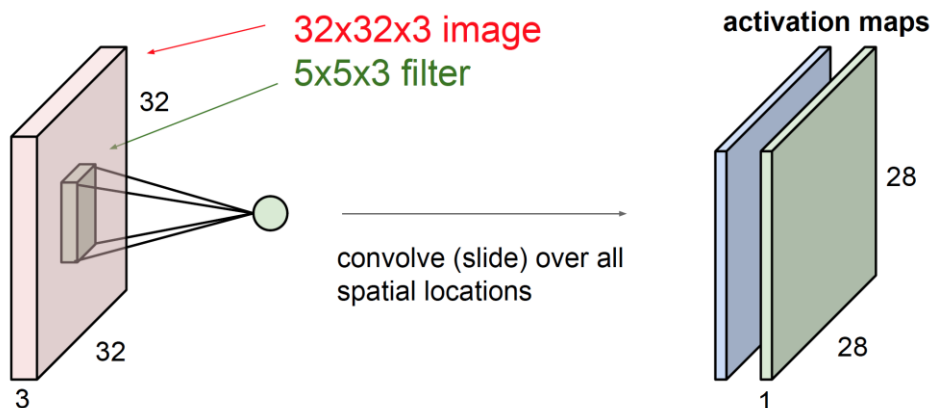


# 卷积神经网络—卷积层

## □ 卷积层

### (Convolutional Layer)

1. 卷积神经网络基本结构
2. 由多个卷积核组合形成
3. 每个卷积核同输入数据卷积运算，形成新的特征“图”

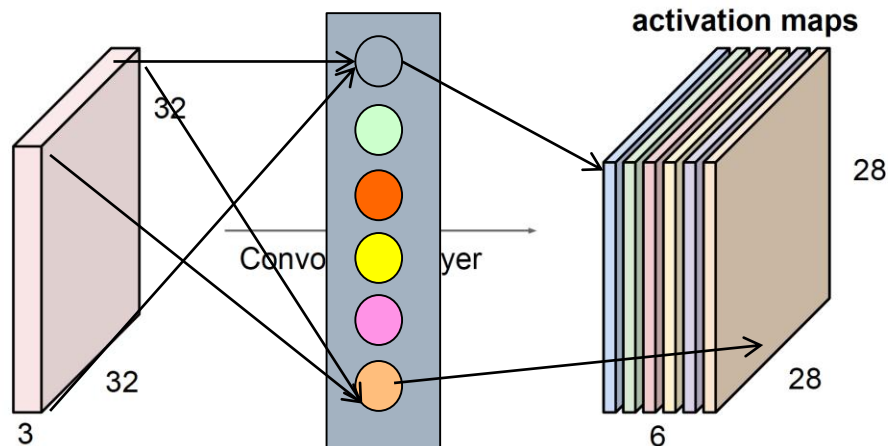


# 卷积神经网络—卷积层

## □ 卷积层

### (Convolutional Layer)

1. 卷积神经网络基本结构
2. 由多个卷积核组合形成
3. 每个卷积核同输入数据卷积运算，形成新的特征“图”



# 卷积神经网络—卷积层

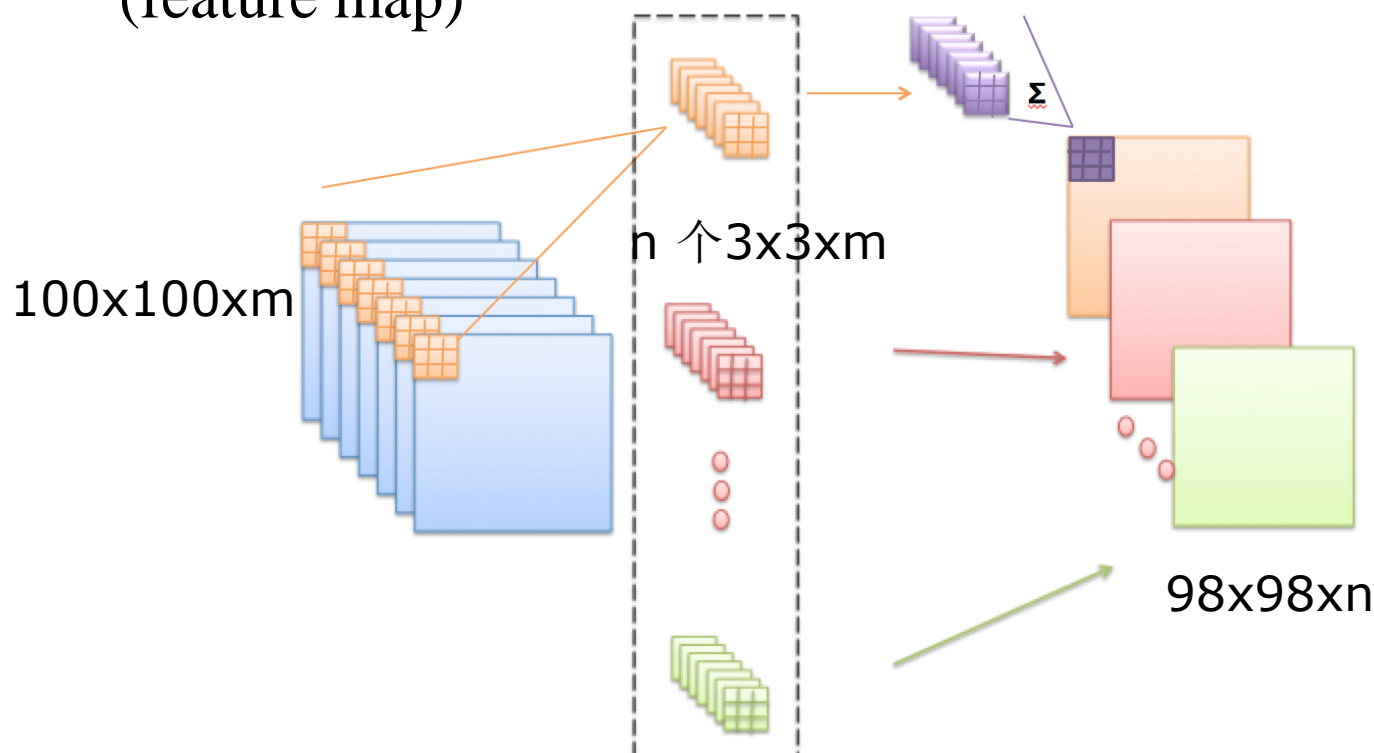
---

## □ 卷积核(Convolutional kernel)

1. 同输入数据进行计算的二维（一维，三维）算子
2. 大小（size）用户定义，深度输入数据定义
3. 卷积核“矩阵”值：卷积神经网络的参数
4. 卷积核初值随机生成，通过反向传播更新

# 卷积神经网络—卷积层

□ 卷积核组合方式：卷积层(conv layer)－特征图(feature map)



这些数据是怎样确定的？

# 卷积神经网络—卷积层

---

## □ 卷积层—关键参数

卷积核大小:

1. 奇偶选择
2. 大小选择
3. 厚度确定?
4. 覆盖范围



# 卷积神经网络—卷积层

---

## □ 卷积层—关键参数

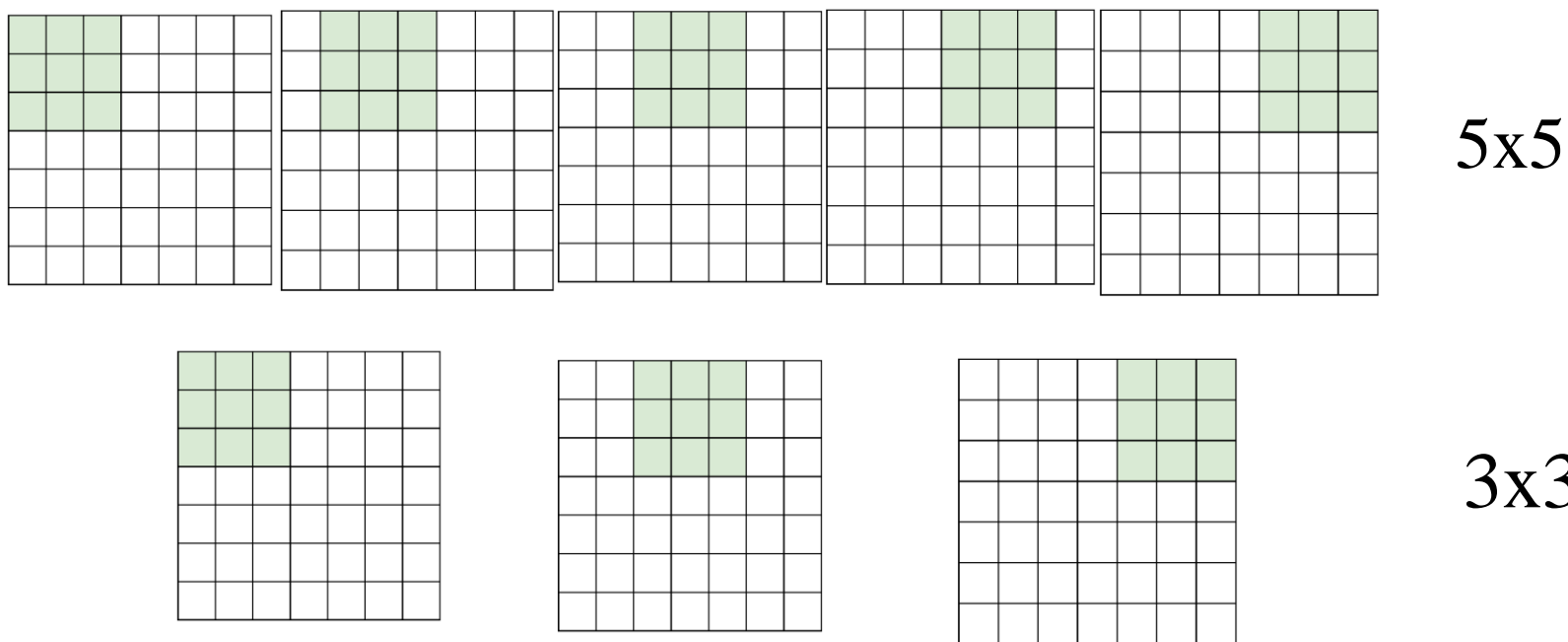
卷积核大小:

1. 奇偶选择: 一般奇数, 满足对称
2. 大小选择: 根据输入数据, 根据图像特征
3. 厚度确定? 与输入数据一致
4. 覆盖范围 一般覆盖全部输入, 特殊情况  
覆盖局部区域

# 卷积神经网络—卷积层

## □ 卷积层—关键参数

步长(stride): 对输入特征图的扫描间隔  
对输出特征图的影响



# 卷积神经网络—卷积层

## □ 卷积层—关键参数

**边界扩充(pad):** 在卷积计算过程中, 为了允许边界上的数据也能作为中心参与卷积运算, 将边界假装延伸

**扩充目的:** 确保卷积后特征图  
尺度一致

**确定方法:** 卷积核的宽度 $2i + 1$ ,  
添加pad宽度为 $i$

0	0	0	0	0	0			
0								
0								
0								
0								

# 卷积神经网络—卷积层

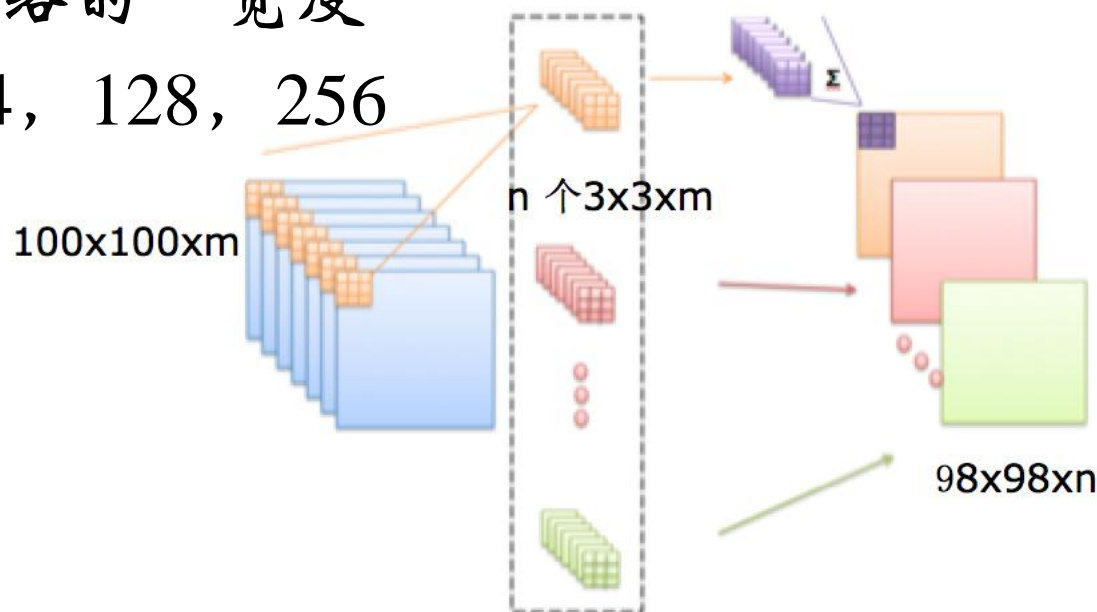
## □ 卷积层—关键参数

卷积核数目(kernel number)

卷积神经网络的“宽度”

常见参数 64, 128, 256

Why?



# 卷积神经网络—卷积层

## □ 卷积层—关键参数

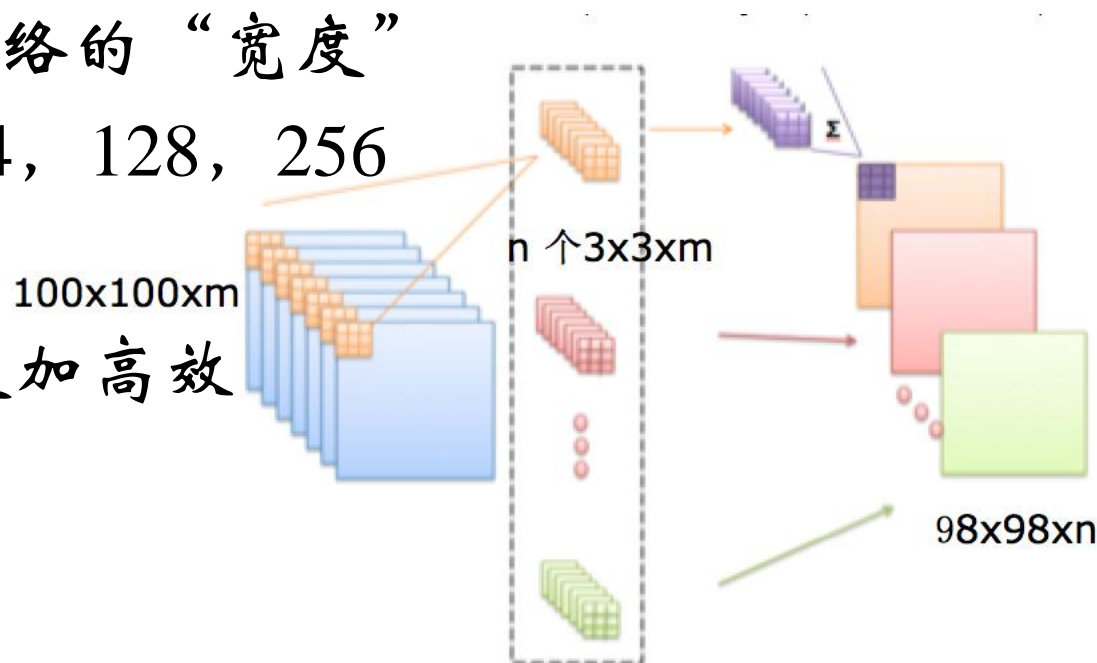
卷积核数目(kernel number)

卷积神经网络的“宽度”

常见参数 64, 128, 256

Why?

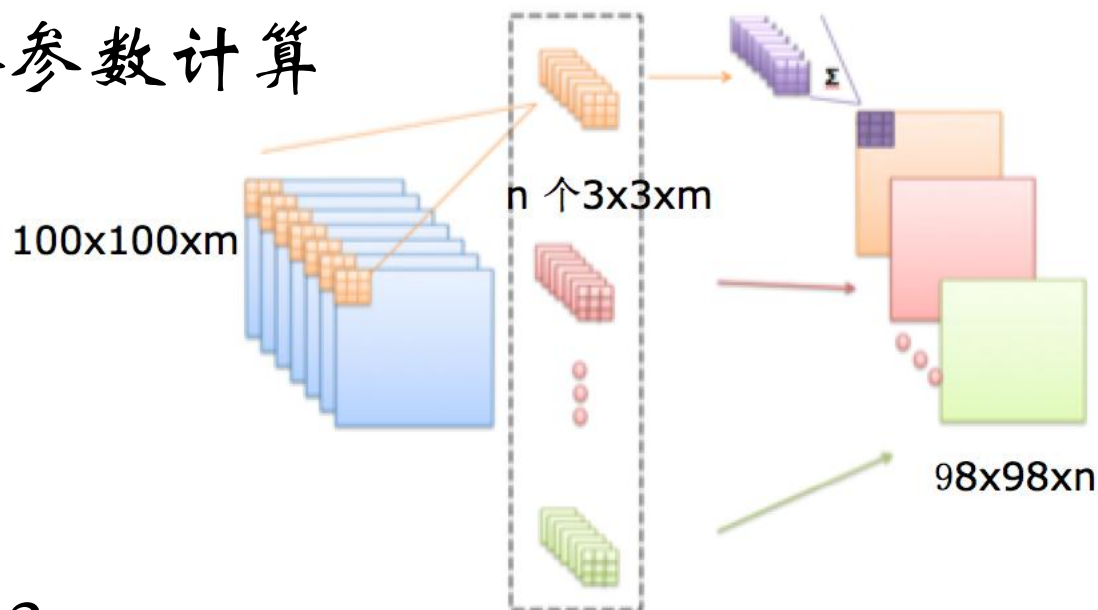
GPU 并行更加高效



# 卷积神经网络—卷积层

## □ 卷积层 → 卷积网络

### 卷积网络参数计算



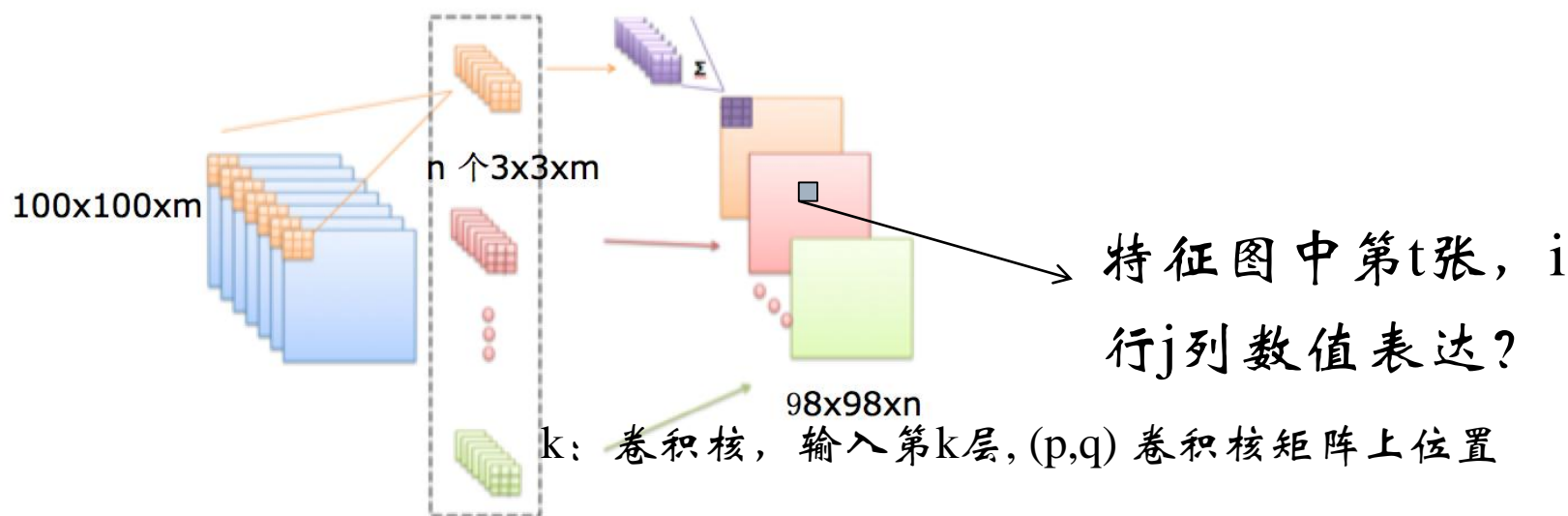
Num:  $3 \times 3 \times m \times n$

与传统神经网络相比（参数/计算量）更多还是更少？

# 卷积神经网络—卷积层

## □ 卷积层 -> 卷积网络

### 正向传播 (Forward Propagation)

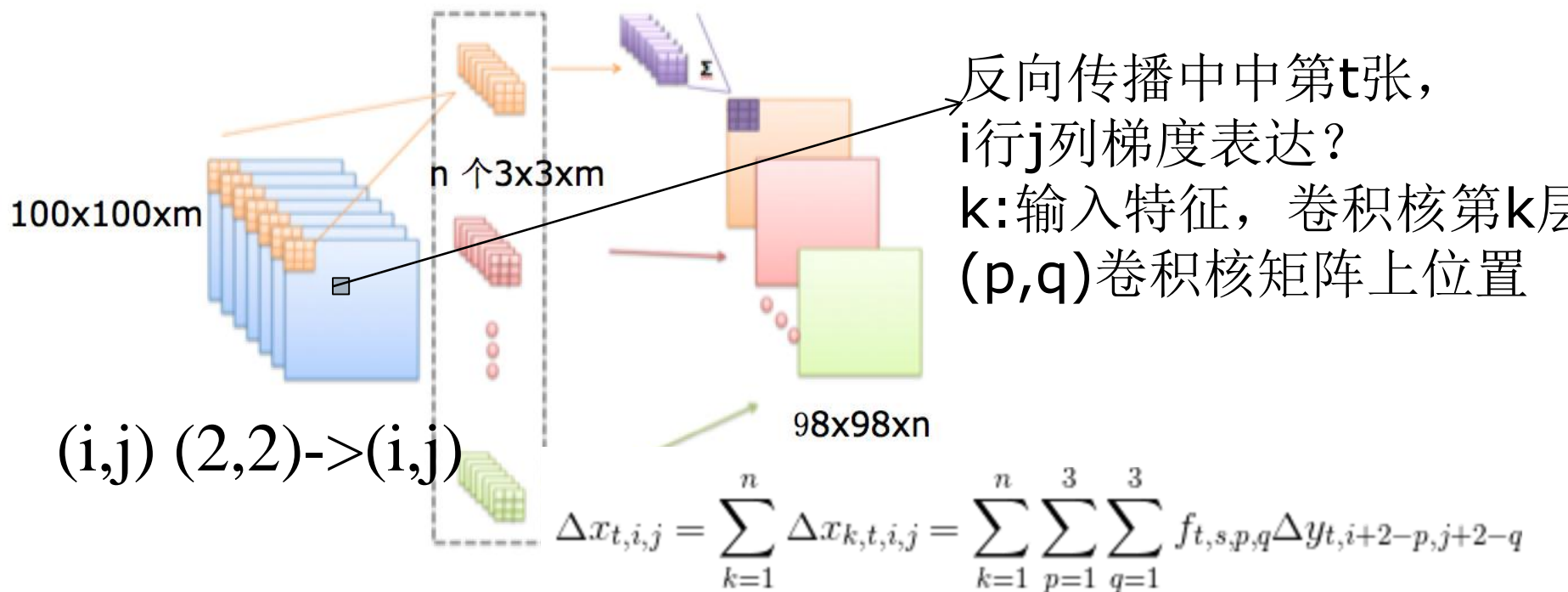


$$y_{t,i,j} = \sum_{k=1}^m z_{k,i,j} = \sum_{k=1}^m f_{t,k} * x_{k,i,j} = \sum_{k=1}^m \sum_{p=1}^3 \sum_{q=1}^3 f_{t,k,p,q} \cdot x_{k,i-2+p,j-2+q}$$

# 卷积神经网络—卷积层

## □ 卷积层 -> 卷积网络

### 反向传播 (Back Propagation)



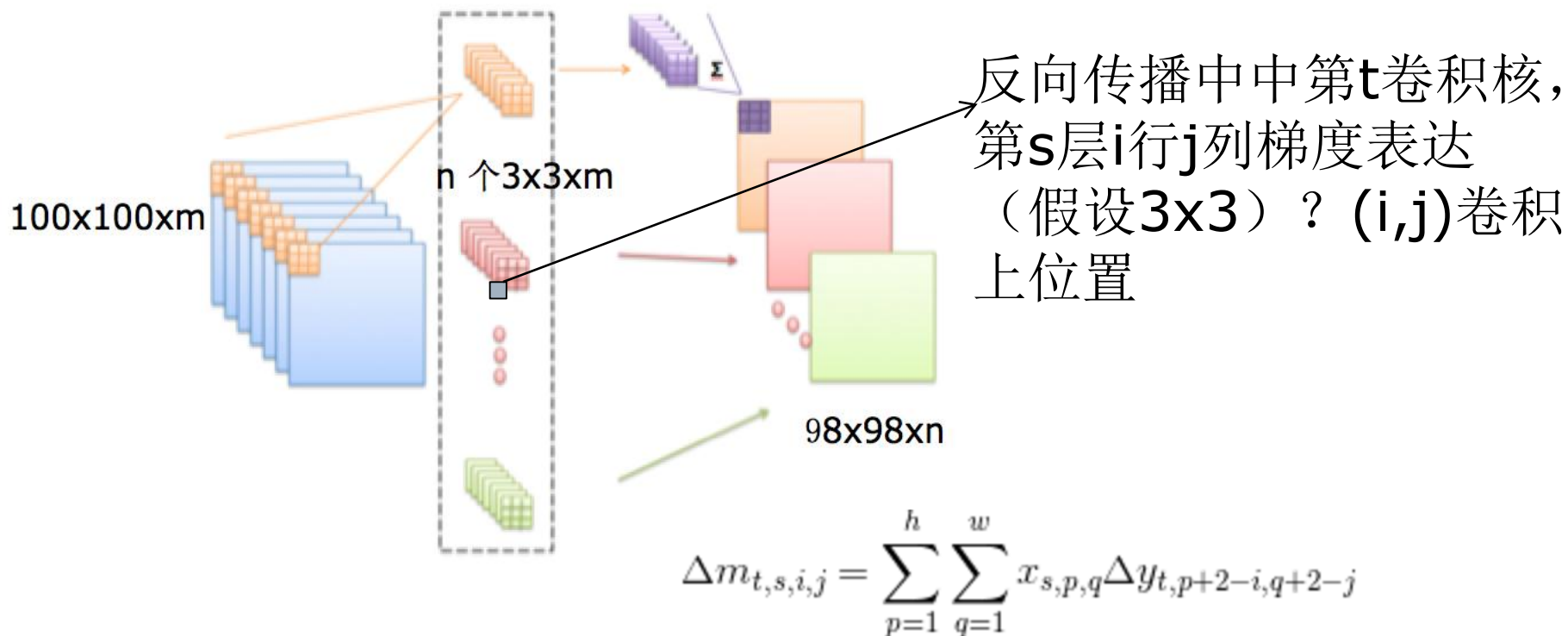
( $i,j$ ) (1,1)->(i+1,j+1)?



# 卷积神经网络—卷积层

## □ 卷积层 -> 卷积网络

### 反向传播 (Back Propagation)



# 卷积神经网络—卷积层

---

## □ 总结

- 卷积核是卷积组成单位
- 卷积核数目，卷积运算的步长，pad对卷积层输出的影响
- 卷积层参数数目计算
- 正向／反向卷积运算表达式

# 提纲

---

- 1. 链式反向梯度传导
- 2. 卷积神经网络-卷积层
- 3. 卷积神经网络-功能层
- 4. 实例：卷积神经网络MNIST分类

# 卷积神经网络—功能层

---

卷积神经网络需要哪些额外功能？

# 卷积神经网络—功能层

---

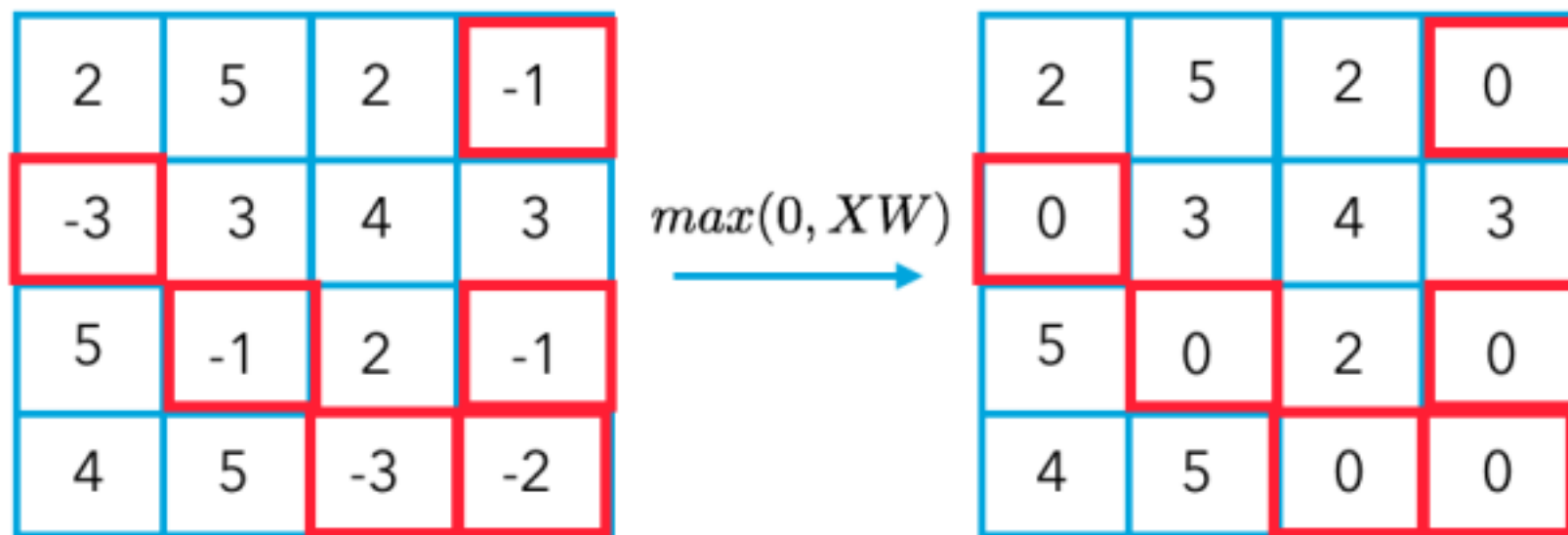
## 卷积神经网络需要哪些额外功能？

- ☐ 非线性激励：卷积是线性运算，增加非线性描述能力
- ☐ 降维：特征图稀疏，减少数据运算量，保持精度
- ☐ 归一化：特征的scale保持一致
- ☐ 区域分割：不同区域进行独立学习
- ☐ 区域融合：对分开的区域合并，方便信息融合
- ☐ 增维：增加图片生成或探测任务中空间信息

# 卷积神经网络—功能层

## □ 非线性激励层(Non-linear activation layer)

ReLU函数:

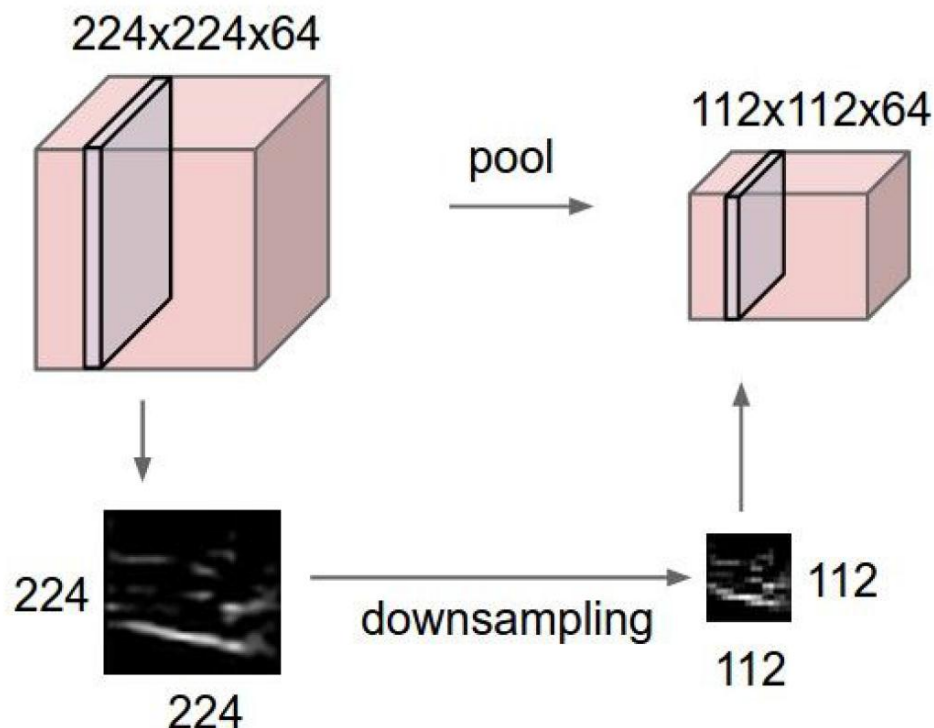


# 卷积神经网络—功能层

## □ 池化层(Pooling layer)

1. 数据降维，方便计算，  
存储 (max, ave)

2. 池化过程中，每张  
特征图单独降维



# 卷积神经网络—功能层

## □ 归一化层(Normalization Layer)

### Batch Normalization (BN)

批量归一化:

1. 原因: 特征数

Scale不一致

2. 好处: 加速训练, 提高精度

**Input:** Values of  $x$  over a mini-batch:  $\mathcal{B} = \{x_1 \dots x_m\}$ ;  
Parameters to be learned:  $\gamma, \beta$

**Output:**  $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$



# 卷积神经网络—功能层

## □ 归一化层(Normalization Layer)

近邻归一化(Local Response Normalization):

$$x_i = \frac{x_i}{(k + (\alpha \sum_j x_j^2))^{\beta}}$$

与BN的区别:

1. BN依据mini batch的数据, 近邻归一仅需要自身
2. BN训练中有学习参数,

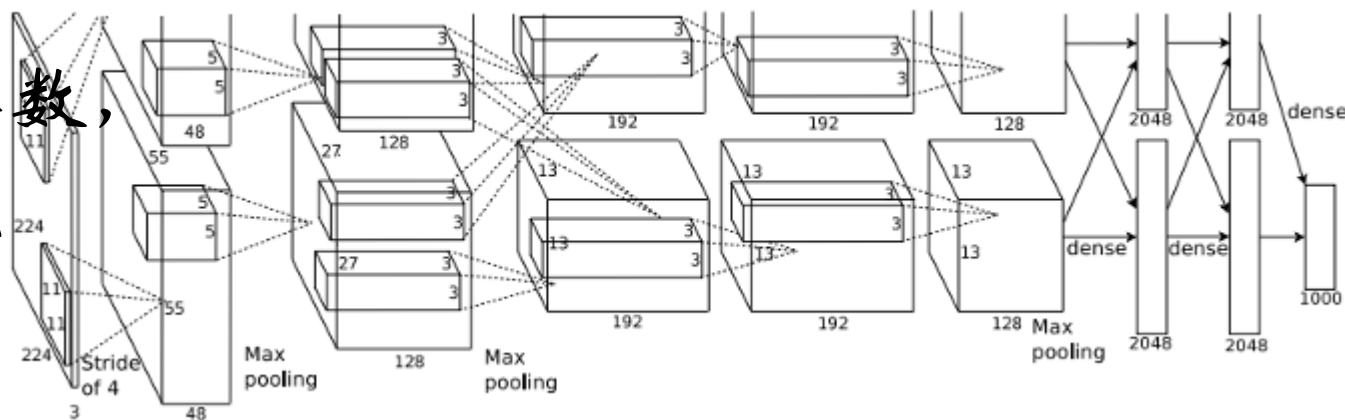
# 卷积神经网络—功能层

## □ 切分层(Slice Layer)

在某些应用中，希望独立对某些区域单独学习

好处：

学习多套参数，  
更强的特征  
描述能力



# 卷积神经网络—功能层

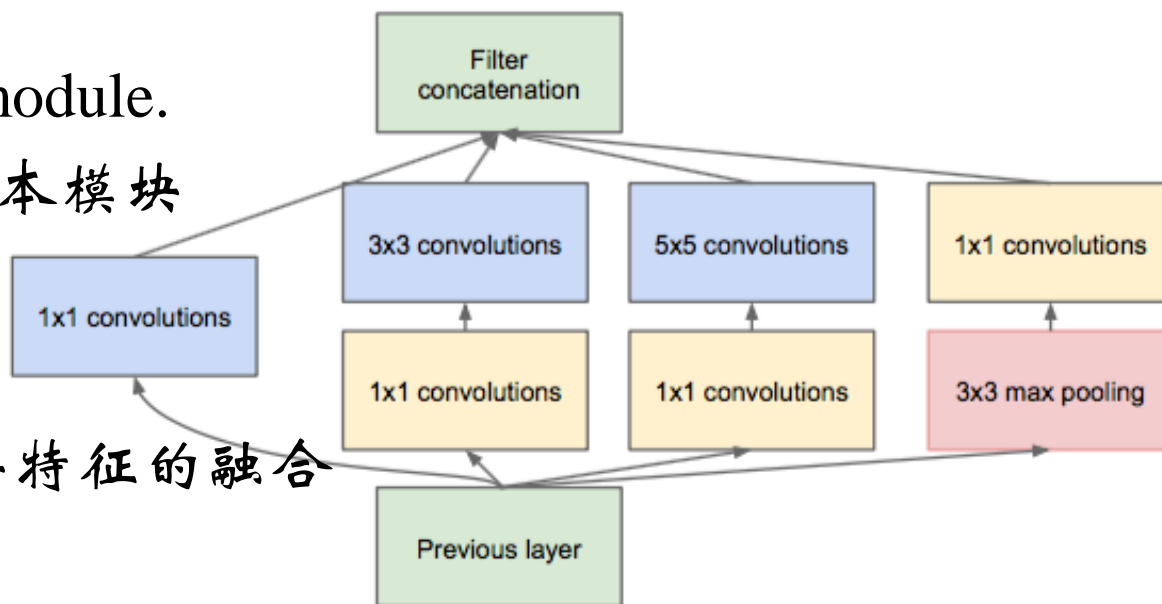
## □ 融合层(Merge Layer)

对独立进行特征学习的分支进行融合，构建高效而精简的特征组合

Google Inception module.

GoogleLeNet 的基本模块

用多种分辨率对目标特征进行学习  
之后进行多分辨率特征的融合



# 卷积神经网络—功能层

## □ 融合层(Merge Layer)

级连—concatenation

不同输入网络特征简单叠加

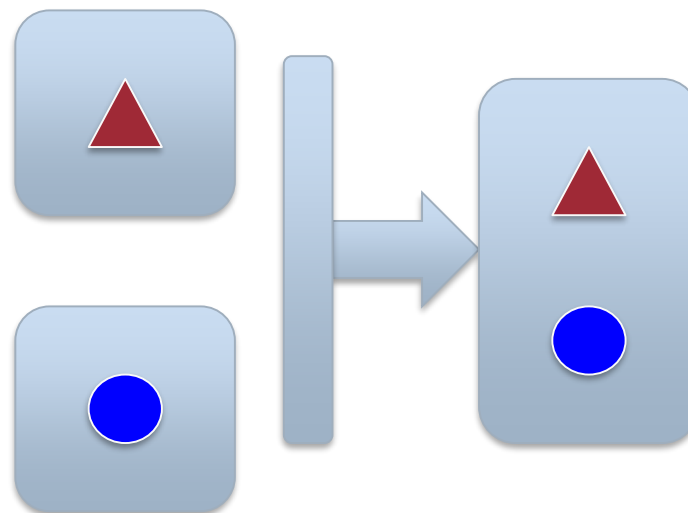
如：1x100 1-D vector+1x100

1-D vector->1x200

或：100x100x3 图片 + 100x

100x1 深度图

100x100x4 RGBD

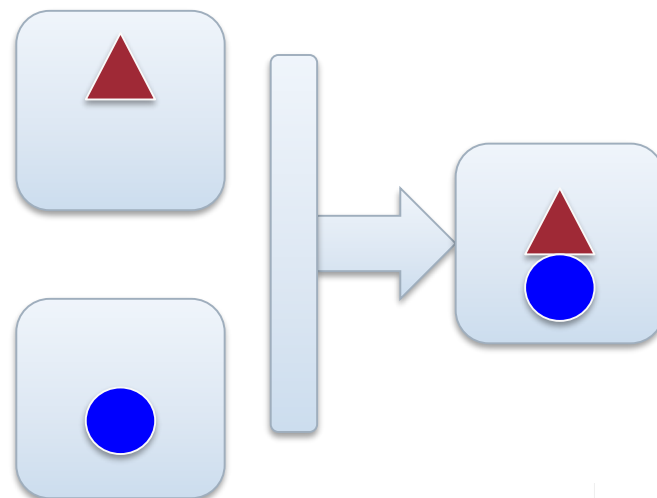


# 卷积神经网络—功能层

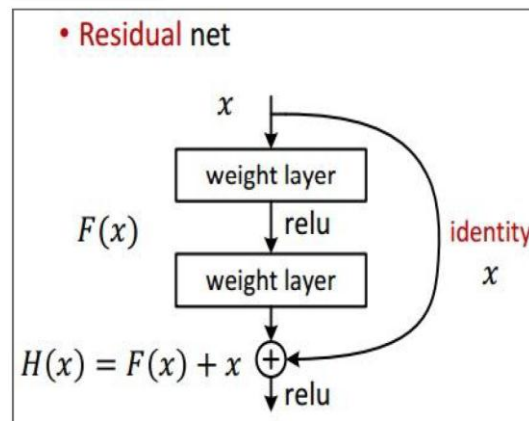
## □ 融合层(Merge Layer)

合并—运算融合

形状一致的特征层，通过  
(+, -, x, max, conv)  
运算，形成形状相同的输出



ResNet的融合:



# 卷积神经网络—功能层

---

## □ 总结

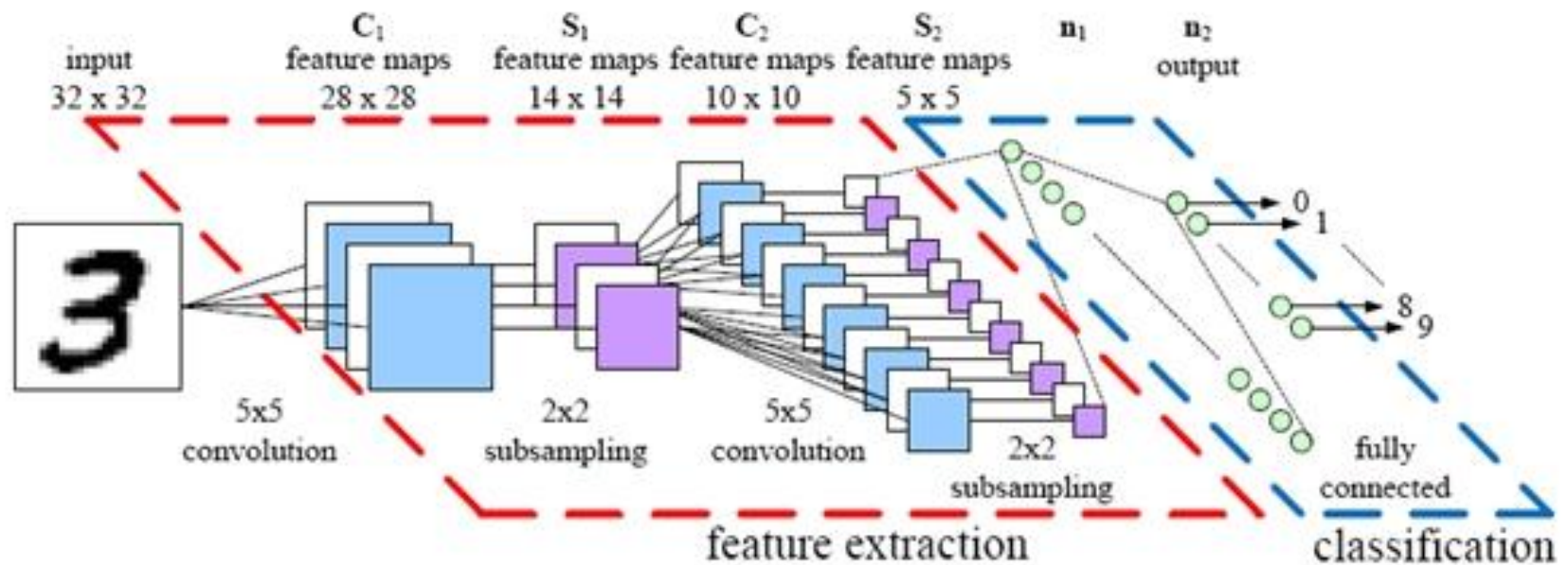
- 非线性激励：卷积是线性运算，增加非线性描述能力
- 降维：特征图稀疏，减少数据运算量，保持精度
- 归一化：特征的scale保持一致
- 区域分割：不同区域进行独立学习
- 区域融合：对分开的区域合并，方便信息融合
- 增维：增加图片生成或探测任务中空间信息

# 提纲

---

- 1. 链式反向梯度传导
- 2. 卷积神经网络-卷积层
- 3. 卷积神经网络-功能层
- 4. 实例：卷积神经网络MNIST分类

# 卷积神经网络MNIST分类





# 总结

---

- 1. 链式反向梯度传导
- 2. 卷积神经网络-卷积层
- 3. 卷积神经网络-功能层
- 4. 实例：卷积神经网络MNIST分类

# 期待目标

---

- 1. 清楚神经网络优化原理，掌握反向传播计算
- 2. 掌握卷积神经网络卷积层的结构特点，关键参数，层间的连接方式。
- 3. 了解不同卷积神经网络功能层的作用，会进行简单的卷积神经网络结构设计。
- 4. 能够运行Tensorflow 卷积神经网络MNIST
- 下节预告：高级卷积网络分析，准备好GPU

# 总结

---

□ 有问题请到课后交流区

□ 问题答疑：<http://www.xxwenda.com/>

■ 可邀请老师或者其他人回答问题

□ 课堂QQ群438285995，微信群

□ 讲师微博：weightlee03，每周不定期分享DL资料

□ GitHub ID：wiibrew（课程代码发布）