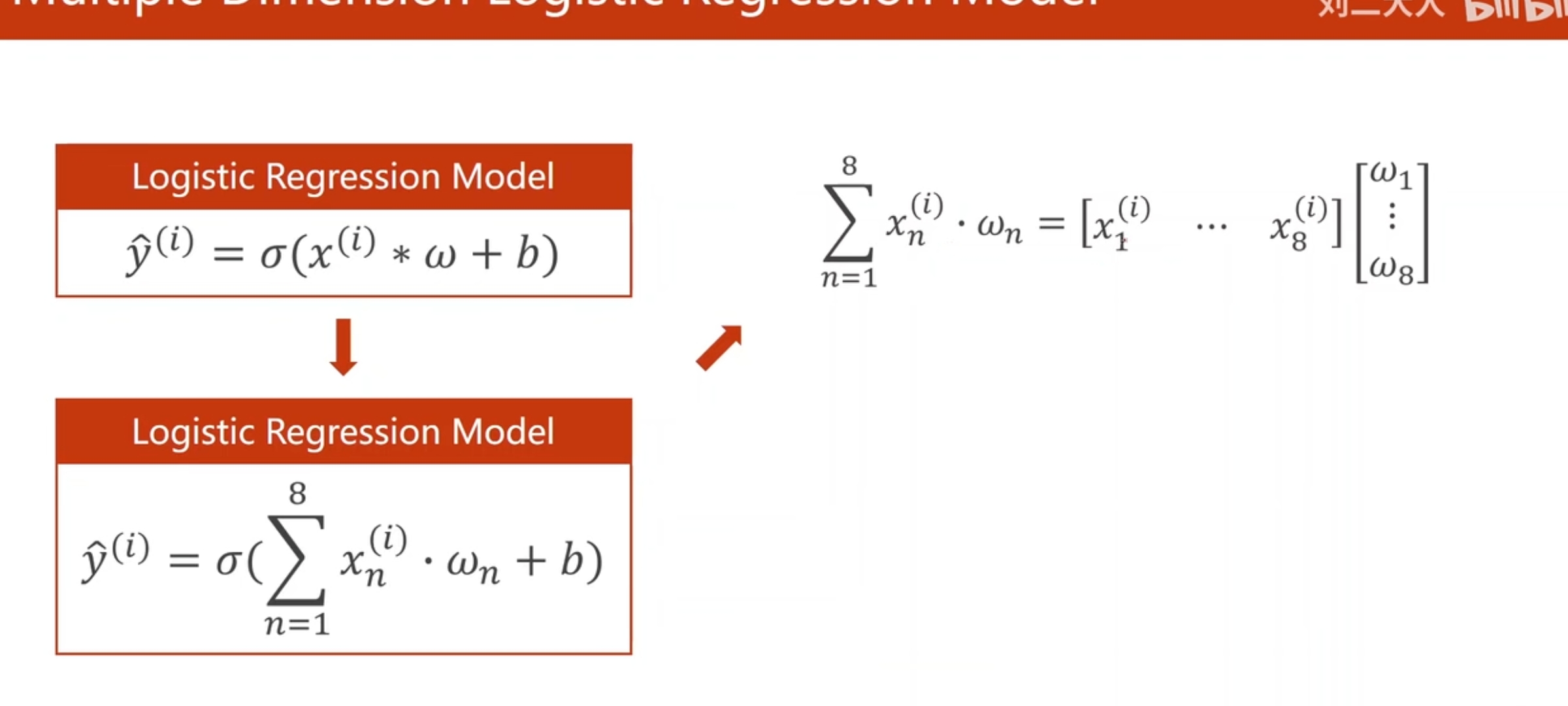
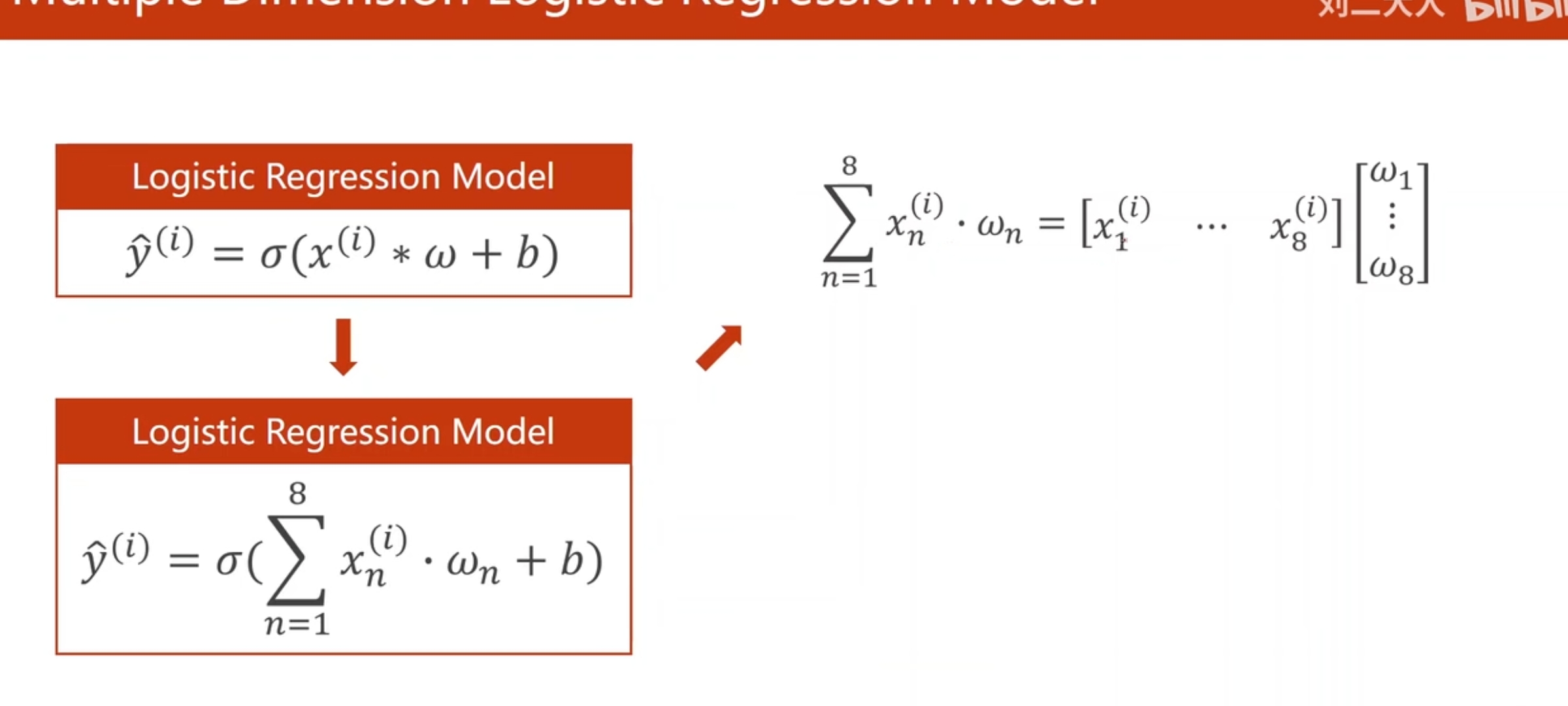
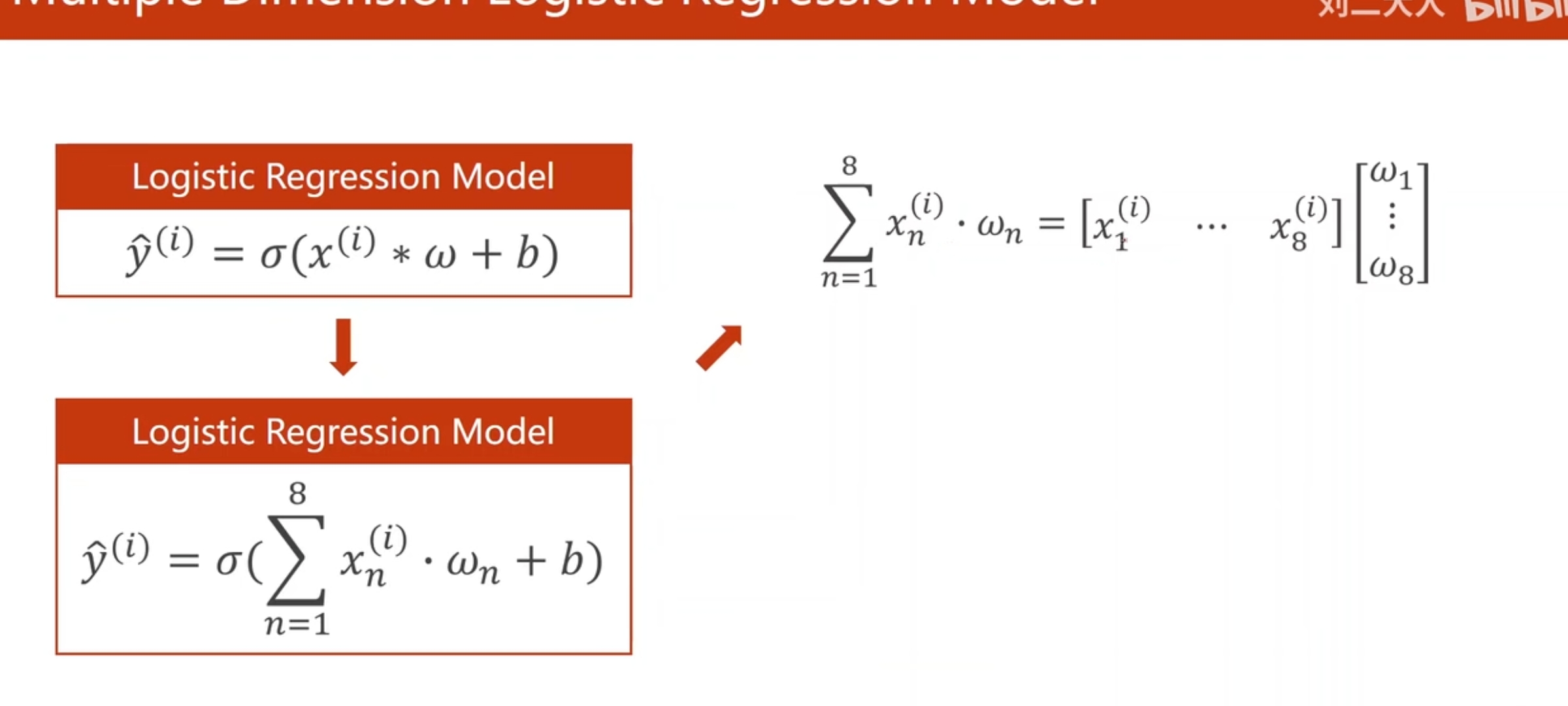
处理多维特征的输入



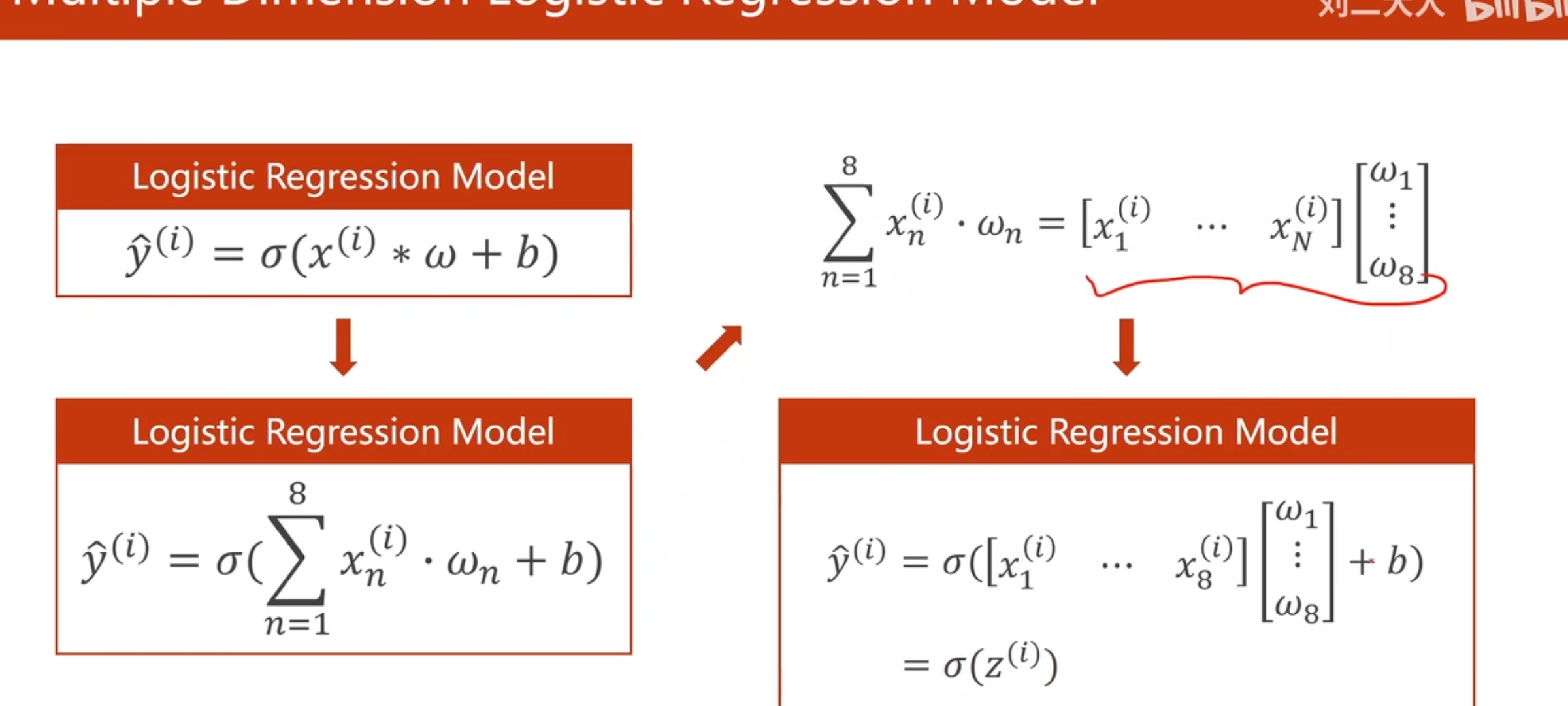
↓



↓



↓



self.linear = torch.nn.Linear(8,1)

注：输入维数维8，输出维数为1

矩阵是空间变换的函数

学习能力太强会把样本的噪声也学会了，会造成真实数据的误差

编程开发的核心能力：1.读文档 2.基本架构理念

解决过程一样：

1. 准备数据集
2. 设计模型
3. 建立损失函数，并且进行优化
4. 训练循环

torch.Tensor()是python类，更明确地说，是默认张量类型torch.FloatTensor()的别名，torch.Tensor([1,2])会调用Tensor类的构造函数\_\_init\_\_，生成单精度浮点类型的张量。

Batch方法可以利用向量计算的方式最大化地提升向量运算的速度

Mni-Batch里面的一些概念：

Epoch:对于训练集的一次前向和一次反向传播

Batch-size:前向和反向传递的训练集的数目

Ieration：使用Batch-size的轮数

Shuffle:打乱数据集里的排列顺序

什么是Python魔法方法

魔法方法是指Python内部已经包含的，被双下划线所包围的方法，这些方法在进行特定的操作时会自动被调用，它们是Python面向对象下智慧的结晶。初学者掌握Python的魔法方法也就变得尤为重要了。

为什么要使用Python魔法方法

使用Python的魔法方法可以使Python的自由度变得更高，当不需要重写时魔法方法也可以在规定的默认情况下生效，在需要重写时也可以让使用者根据自己的需求来重写部分方法来达到自己的期待。而且众所周知Python是支持面向对象的语言Python的基本魔法方法就使得Python在面对对象方面做得更好。

\_\_init\_\_(self[, ...]) 构造方法，初始化类的时候被调用

\_\_getitem\_\_(self, key) 定义获取容器中指定元素的行为，相当于 self[key]

\_\_len\_\_(self) 定义当被 len() 调用时的行为（一般返回容器类的长度）

shape函数是numpy.core.fromnumeric中的函数，它的功能是读取矩阵的长度，比如shape[0]就是读取矩阵第一维度也就是第一列的的长度。

enumerate()说明

enumerate()是python的内置函数

enumerate在字典上是枚举、列举的意思

对于一个可迭代的（iterable）/可遍历的对象（如列表、字符串），enumerate将其组成一个索引序列，利用它可以同时获得索引和值

enumerate多用于在for循环中得到计数

enumerate还可以接收第二个参数，用于指定索引起始值

同一组数据用两种方法实践出来的效果

Batch\_size

test\_pred = 2.8163191018393263e-05

infact\_pred = 0.003064409364014864

特点：计算速度慢但是精度高

All\_size

test\_pred = 6.451273685767695e-21

infact\_pred = 0.003064409364014864

特点：运算速度快但是精度低

Softmax Layer

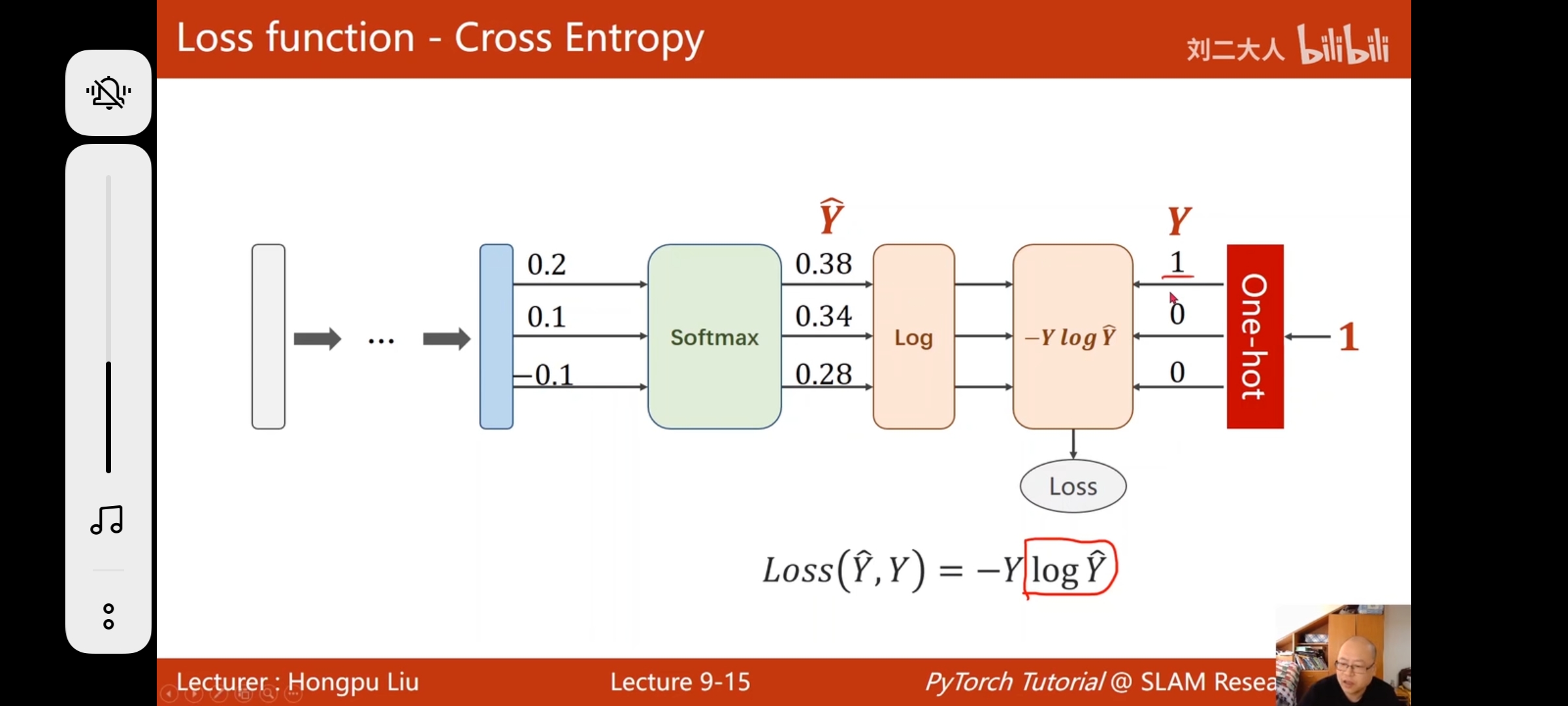
形式： and 

假设时最后一个线性层的输出，Softmax function为

(这个等式非常契合上述的两个要求)

损失函数：





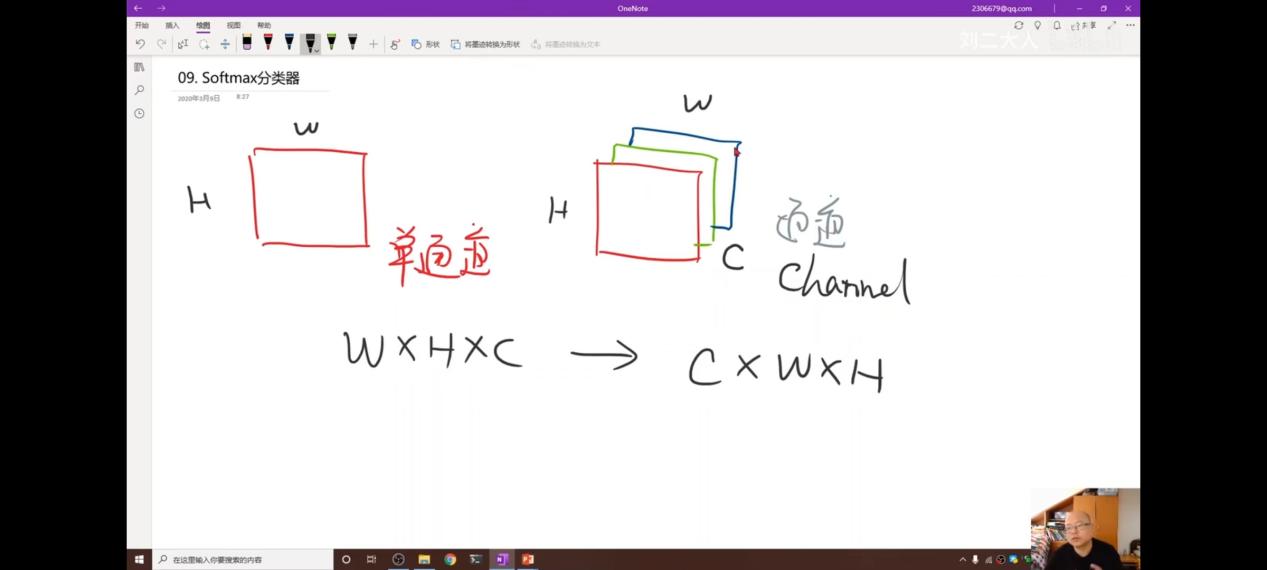
神经网络最后一层不要用激活函数直接交给交叉熵

nll\_loss()与CrossEntropyLoss()损失函数计算的关系为CrossEntropyLoss()等于对输入数据先做softmax，再做log处理，再加nll\_loss()操作。

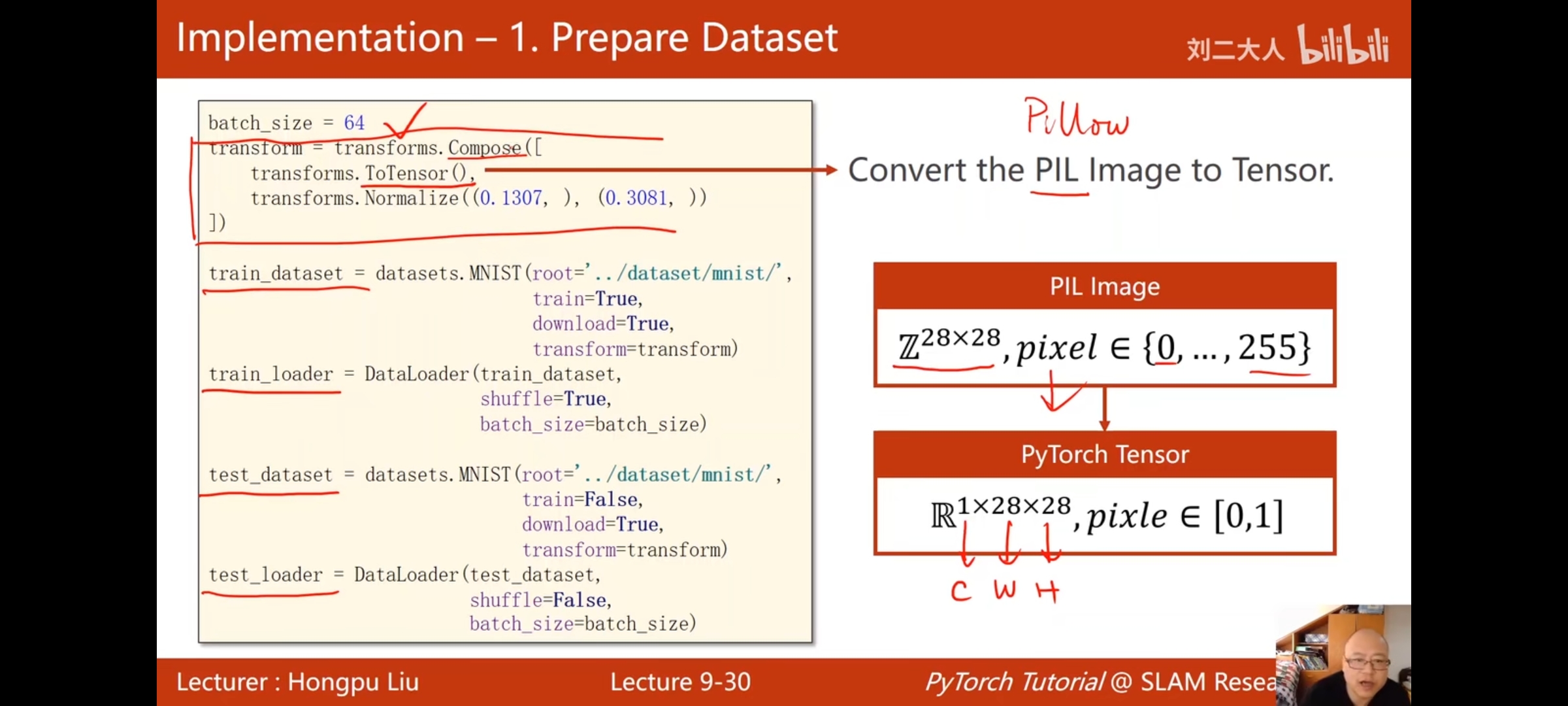
1.NLLLoss 的输入是一个对数概率向量和一个目标标签，它不会为我们计算对数概率.，适合网络的最后一层是log\_softmax损失函数，只是直接用概率向量与目标标签进行数乘运算。有时候调用预训练模型修改网络最后的全连接层的时候会把最后一层的输出改变，这时候就不适合用CrossEntropyLoss()作为损失函数；CrossEntropyLoss()与NLLLoss()相同，唯一的不同是它为我们去做 softmax。

2.CrossEntropyLoss():交叉熵损失函数，交叉熵描述了两个概率分布之间的距离，当交叉熵越小说明二者之间越接近。

图像的构成说明 图像由三层构成(red,green,blue)层数又称为通道数，H为图像高度，W为图像宽度，C为通道数



Transforms.ToTensor()将一个PIL图像转变为张量



在torch里面，view函数相当于numpy的reshape。

在函数的参数中经常可以看到-1例如x.view(-1, 4)

这里-1表示一个不确定的数，就是你如果不确定你想要reshape成几行，但是你很肯定要reshape成4列，那不确定的地方就可以写成-1

例如一个长度的16向量x，

x.view(-1, 4)等价于x.view(4, 4)

x.view(-1, 2)等价于x.view(8，2)