

PyTorch实战

✓ 课程安排

- ✎ 纯实战，经典深度学习案例与项目
- ✎ 从零开始，详解其中每一步流程
- ✎ 通俗易懂，最接地气的方式进行讲解
- ✎ 提供所有数据与代码，追随热点持续更新



数据增强

✓ Data Augmentation

✎ 数据不够怎么办?

✎ 如何更高效利用数据?



迁移学习

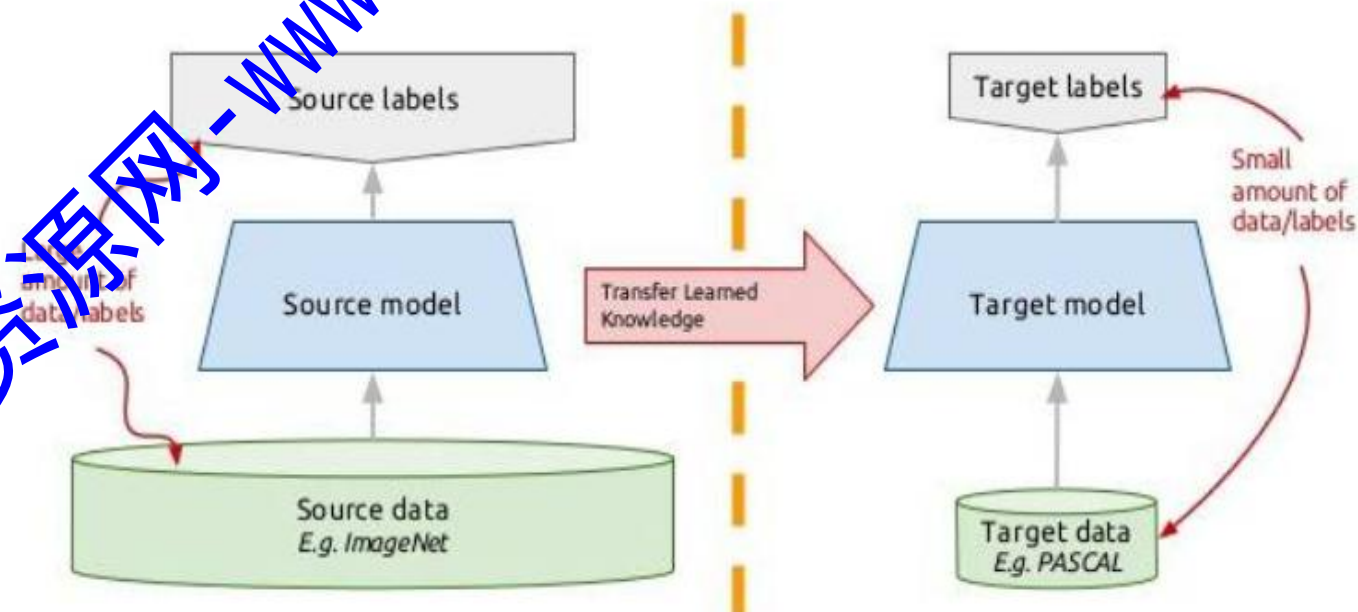
✓ 目标:



Source



Target



✓ 学习哪部分?

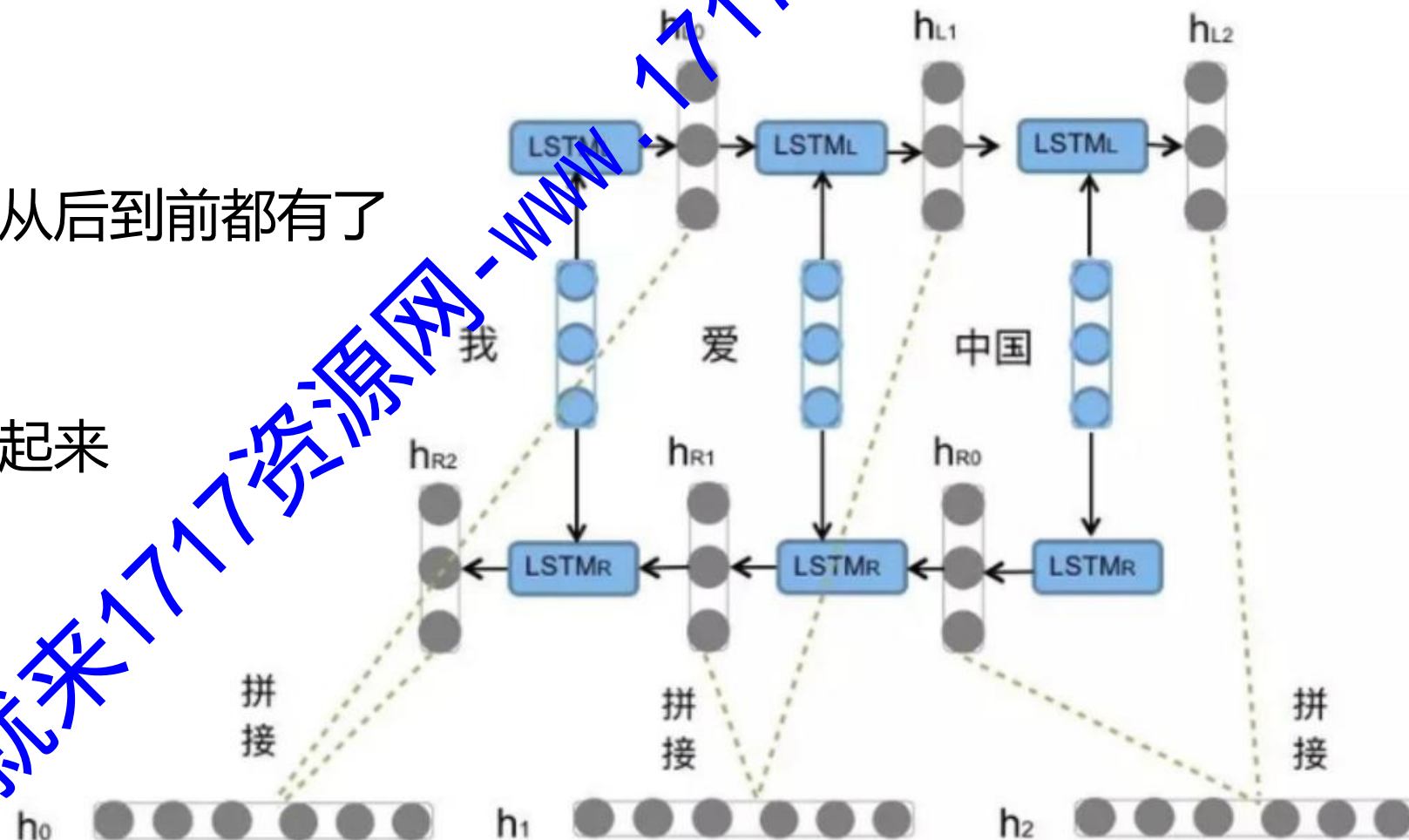


Text-LSTM

✓ BiLSTM

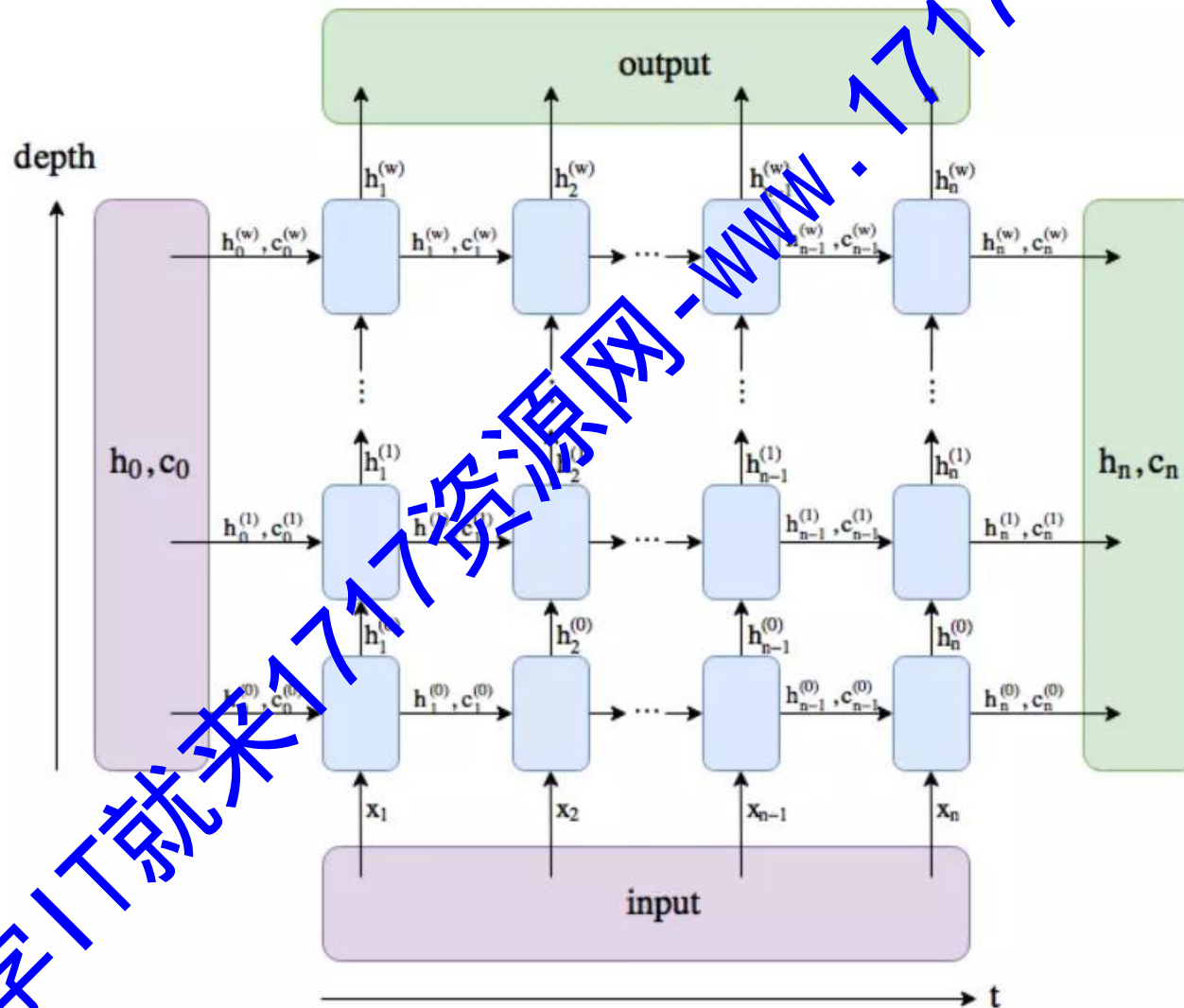
✎ 相当于从前到后和从后到前都有了

✎ 将得到的向量拼接起来



Text-LSTM

✓ 层的表示

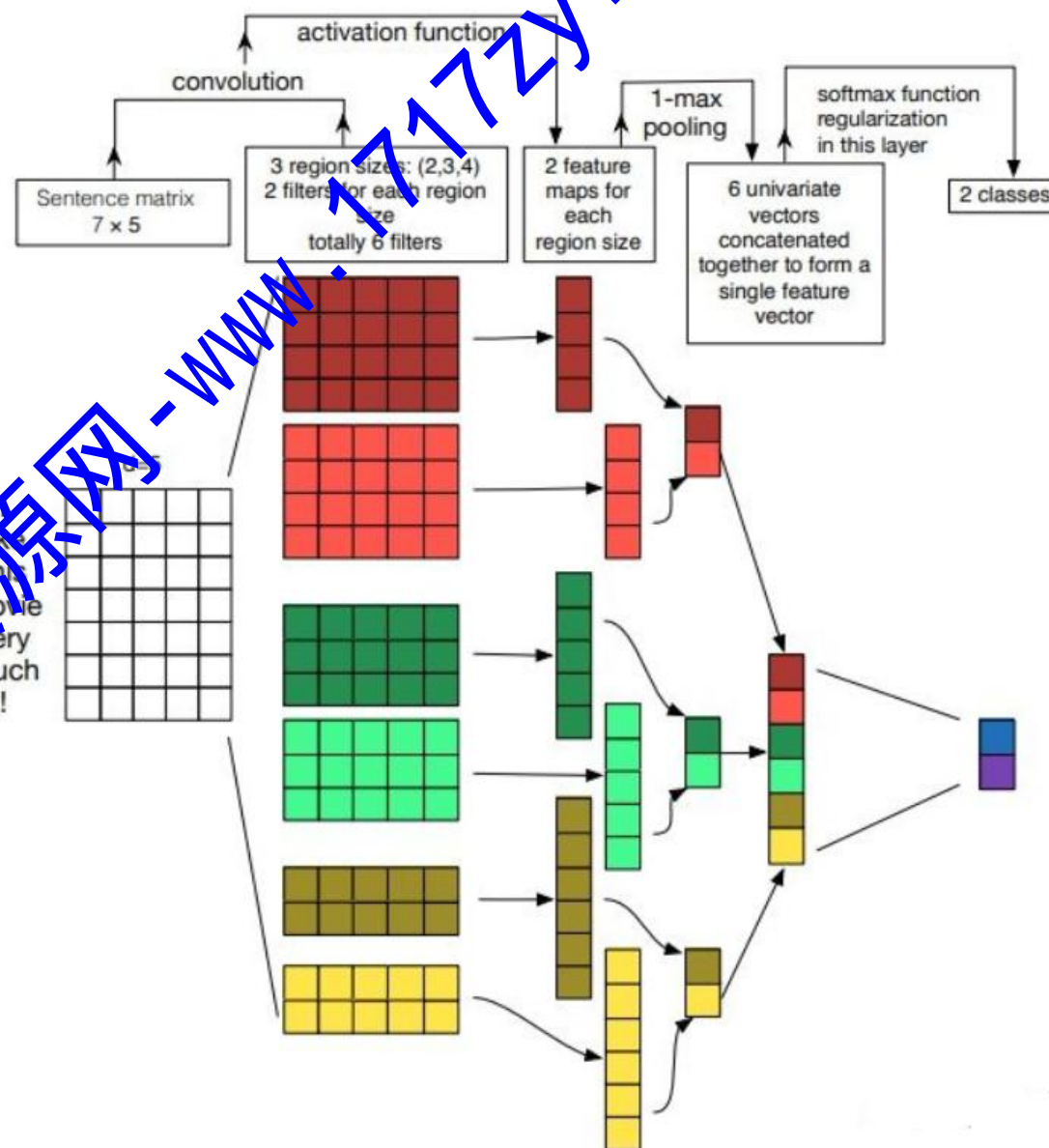


Text-CNN

✓ CNN应用于文本分类

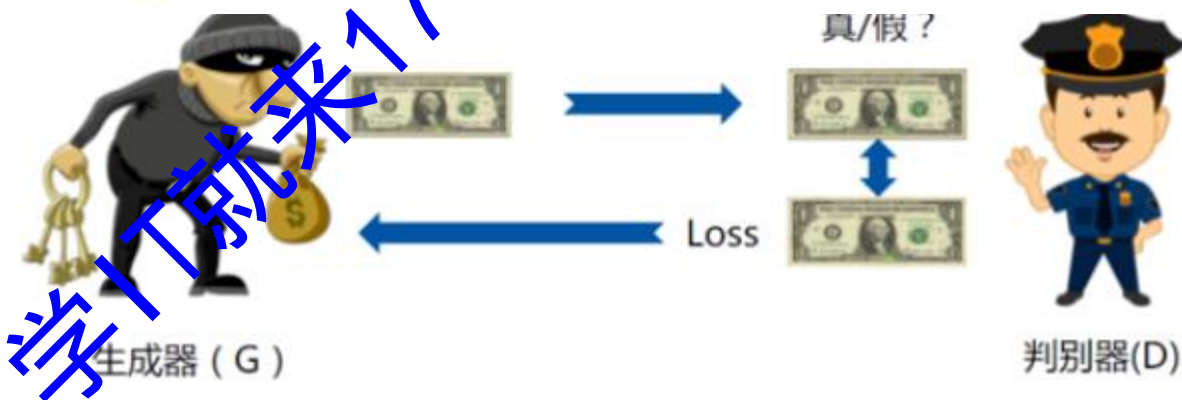
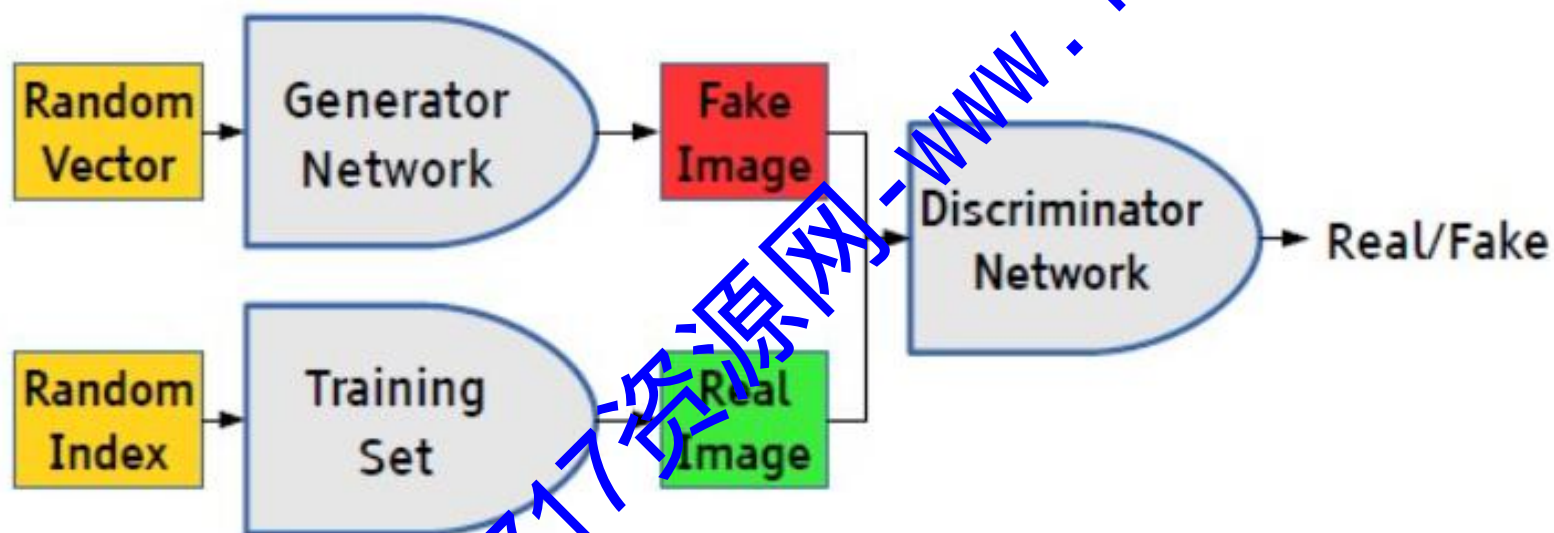
✎ 如何构造输入数据?

✎ 卷积如何使用?



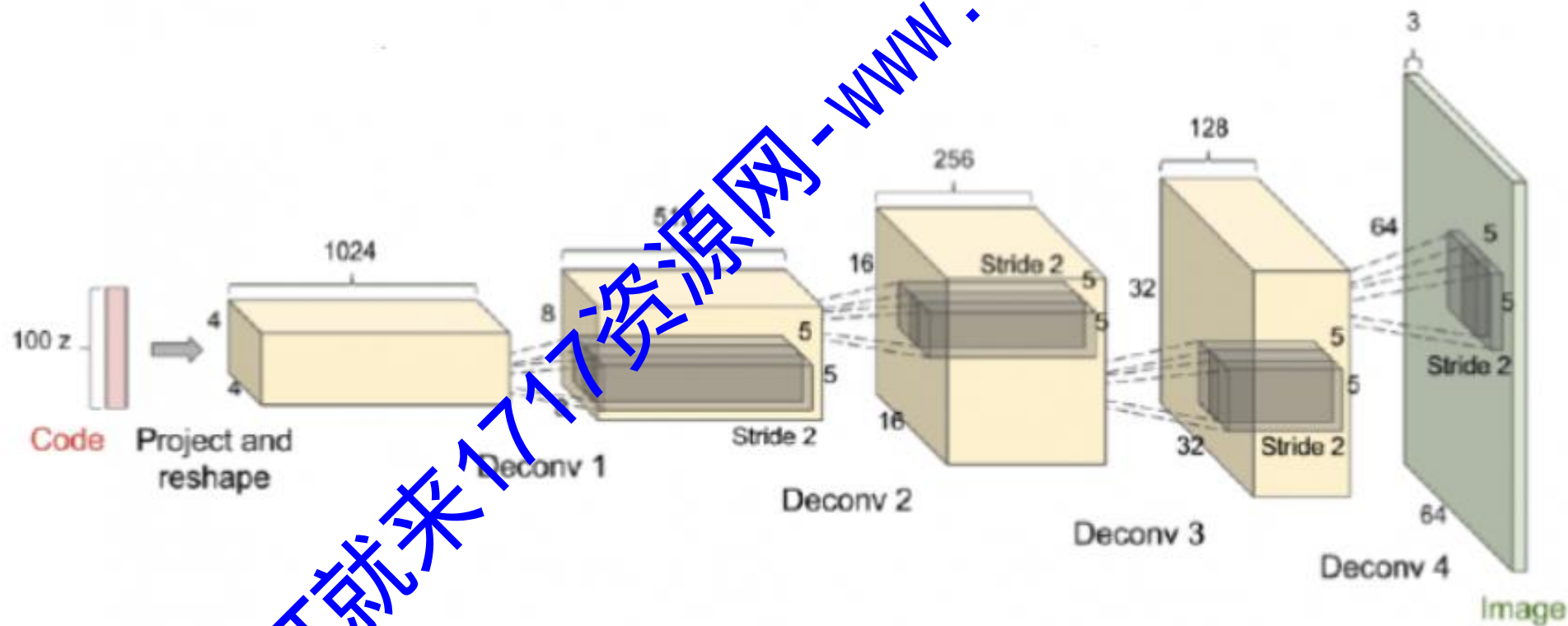
GAN

✓ 对抗生成网络



GAN

✓ DCGAN



GAN

✓ DCGAN

1	5	7	2
3	1	9	3
5	8	3	3
1	2	1	5

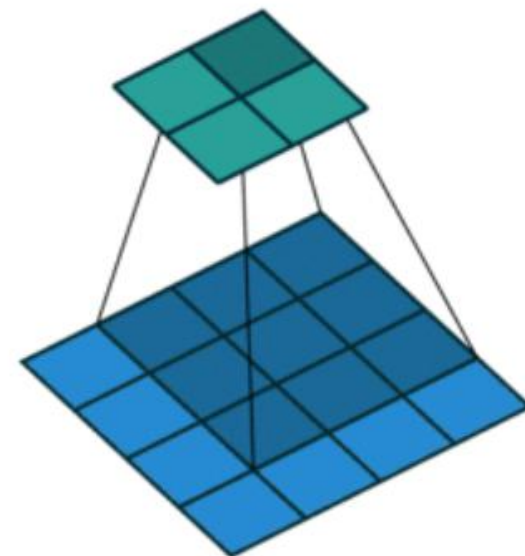
MaxPooling

5	9
8	5

7	6
5	2

UnSampling

7	7	6	6
7	7	6	6
5	5	2	2
5	5	2	2



CycleGan

✓ 这东西能达到什么效果呢?

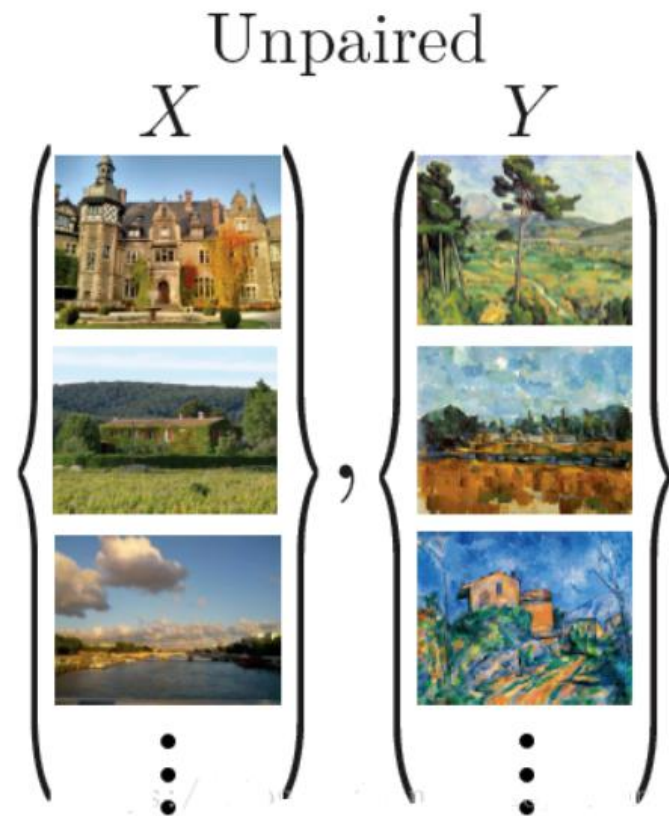


CycleGan

✓ 有什么特别的呢?

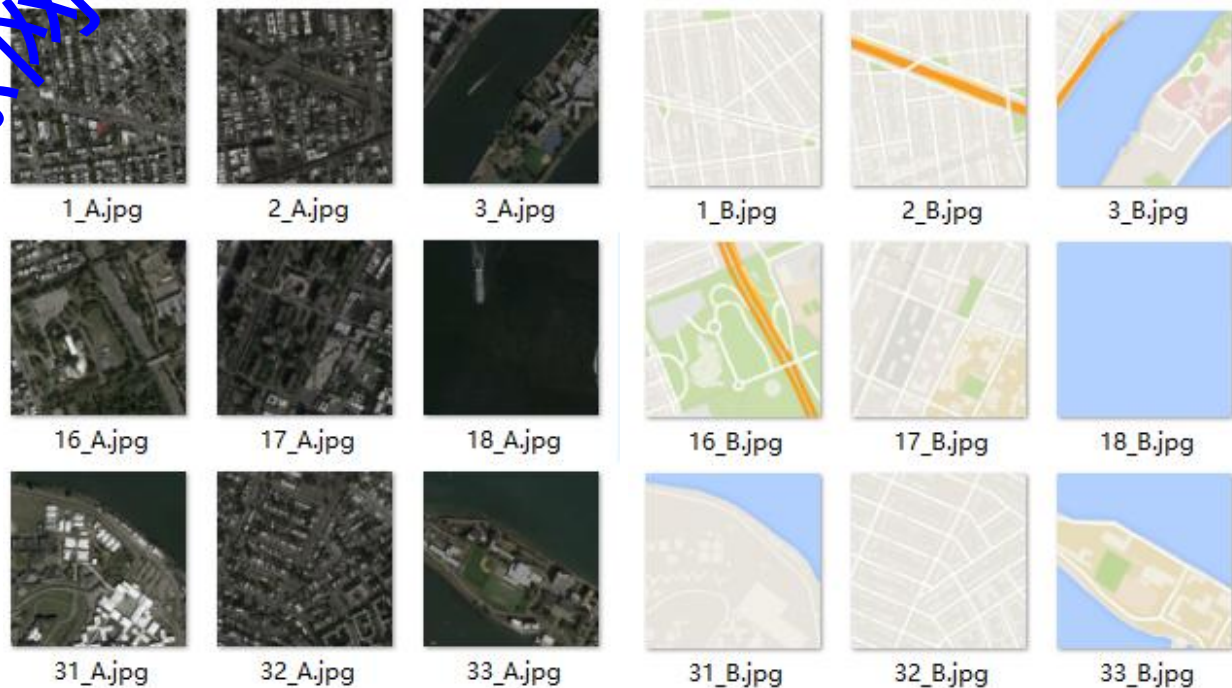
✎ 图像如何进行学习变化?

✎ 没有配对的图像能否训练呢?



✓ 输入数据长什么样子?

只需要两组图像数据集即可，无需指定对应关系，例如马和斑马：

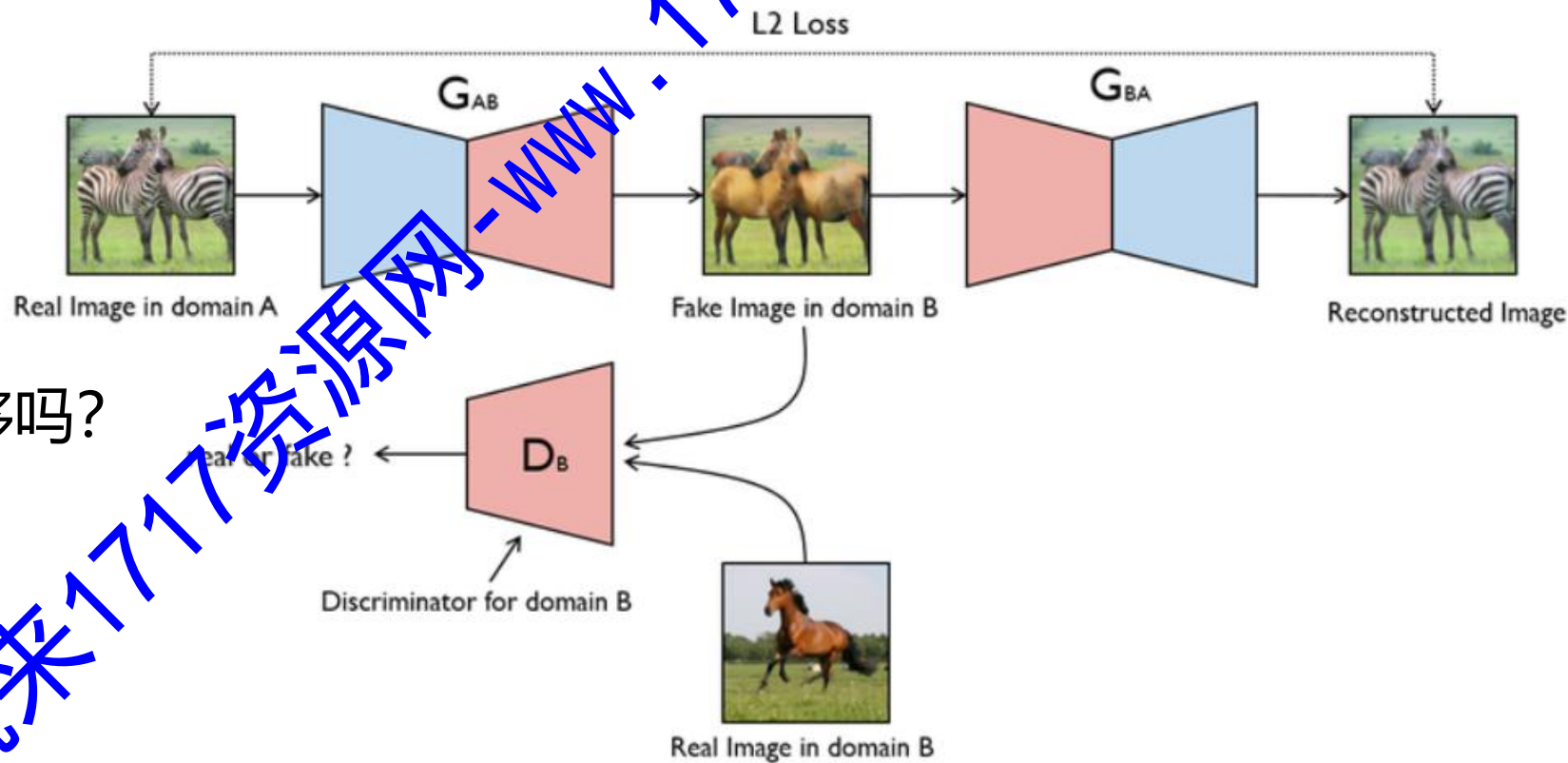


CycleGan

✓ 对抗网络如何进行学习呢?

✎ 需要2对G和D网络

✎ 一个单项的网络足够吗?



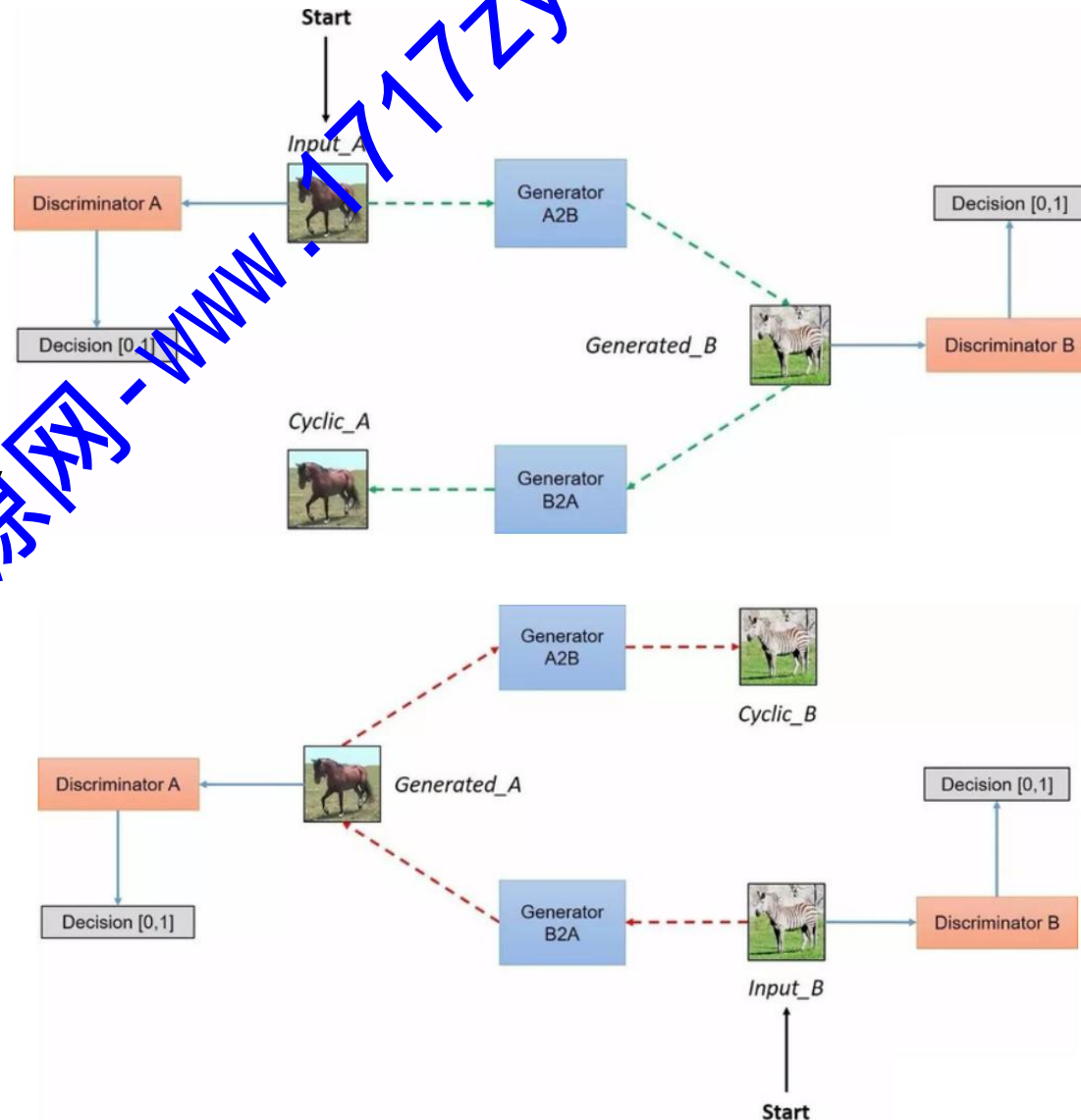
CycleGan

✓ 整体网络架构

✎ 涉及4种损失函数：

G网络, D网络, Cycle, Identity

✎ D网络有点特别, PatchGAN

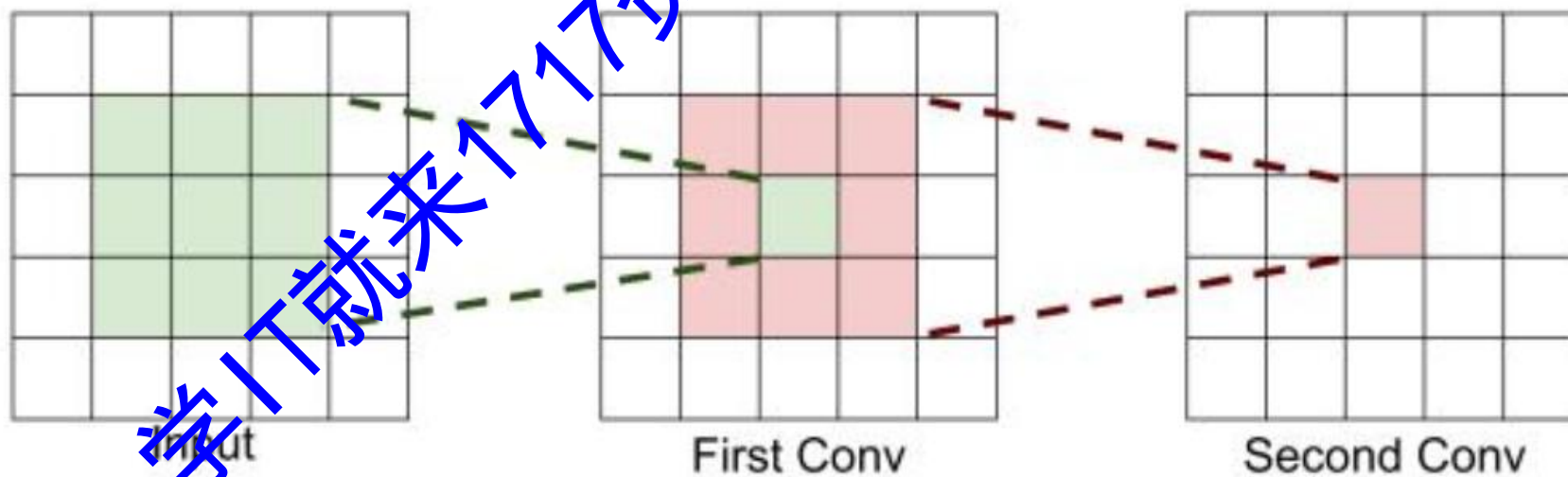


CycleGan

✓ PatchGAN的作用

✎ PatchGAN输出的是一个 $N \times N$ 的矩阵，需要基于感受野来计算损失

✎ 基于感受野在特征图上的预测结果，和标签（也需设置成 $N \times N$ ）计算损失



✓ OCR能做什么呢?

✎ 提取图像中的文字，并转换成文本形式，供后续NLP任务使用



出生日期 / Date of birth
21 AUG 1977
签发日期 / Date of issue
10 10月/OCT 2014

通行费

天津增值税电子普通发票

发票代码: 012001700112
发票号码: 06071016
开票日期: 2018年03月20日
校验码: 07674 22839 28753 80307

项目	名称	规格	单位	数量	单价	税率	税额
通行费	通行费	通行费	通行费	通行费	通行费	通行费	通行费
合计					¥12.95	3%	¥0.39

价税合计 (大写) 壹拾叁元叁角肆分 (小写) ¥13.34

开票人: 蔡红燕 复核: 蔡红燕 开票日期: 2018年03月20日

C:\Users\user\Desktop
9112011876126529X3_d20eba5eff7b459
eb1630fc5d654b3c4 (1).pdf
发票代码: 012001700112
发票号码: 06071016
开票日期: 2018年03月20日
购方企业名称: 天津鑫宇高速公路有限责
任公司
销方企业名称: 天津鑫宇高速公路有限责
任公司
销方纳税号: 9112011876126529X3
价税合计: 13.34
金额: 合计 ¥12.95
税额: ¥0.39
价税合计大写: 壹拾叁元叁角肆分
明细: 货物或应税劳务、服务名称=规格
型号=单位=数量=单价=金额=税率
=税额=备注: 汇总开具
税率:
校验码: 07674 22839 28753 80307
机器编号: 499099656151
票据类型: 增值税电子普通发票
购买方地址及电话: 烟台市莱山区杰瑞路
5号 0535-6766284
购买方开户行及帐号: 农行华苑软件大厦
支行200401040004762
销售方地址及电话: 天津市西青区南河镇
牛坨子村津晋高速公路津文收费站23981832
销售方开户行及帐号: 农行华苑软件大厦
支行200401040004762
收款人: 蔡红燕
复核: 蔡红燕
开票人: 王晶晶

✓ 如何从图像中提取文本呢?

✎ 检测到文字所在位置 (CTPN) :

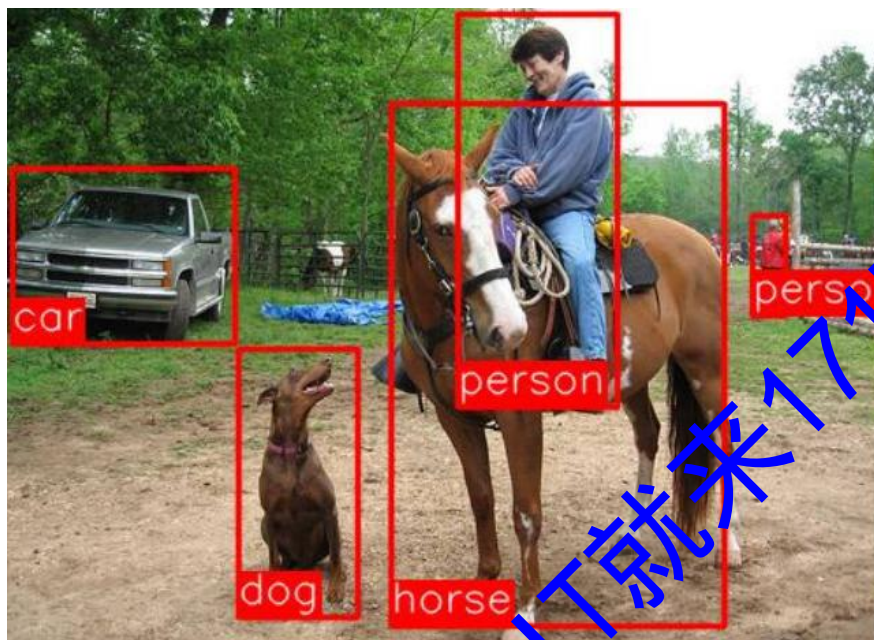


✎ 识别文本区域内容 (CRNN)



✓ CTPN (Connectionist Text Proposal Network) 算法:

✎ 文本检测本质上也属于物体检测，但是文本却跟常规的物体有较大区别:



✓ CTPN算法:

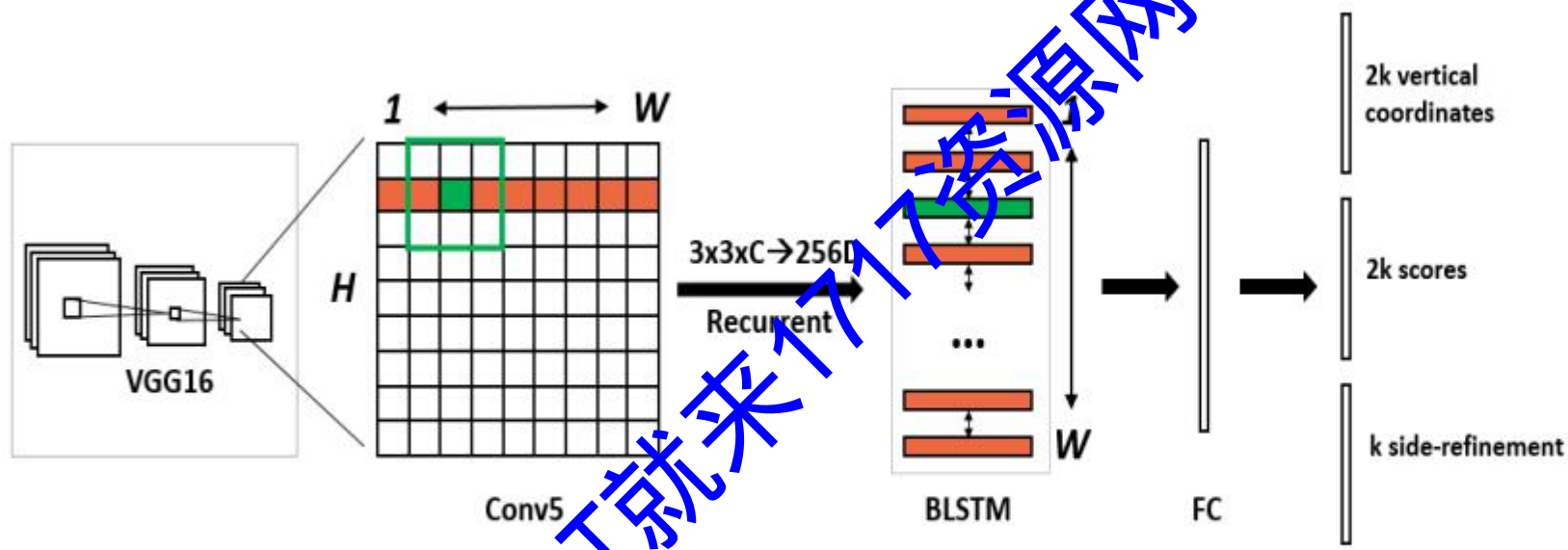
- ✎ 文本通常都是从左往右写的（水平），并且字之间的宽度都大致相同
- ✎ 固定宽度，来检测文本高度即可。但是如何应对变长序列呢？
- ✎ 本质上还是RPN方法（可参考faster-rcnn），可将检测到的框拼在一起！



OCR

✓ CTPN网络架构:

✎ VGG提取特征, BLSTM融入上下文信息, 基于RPN完成检测



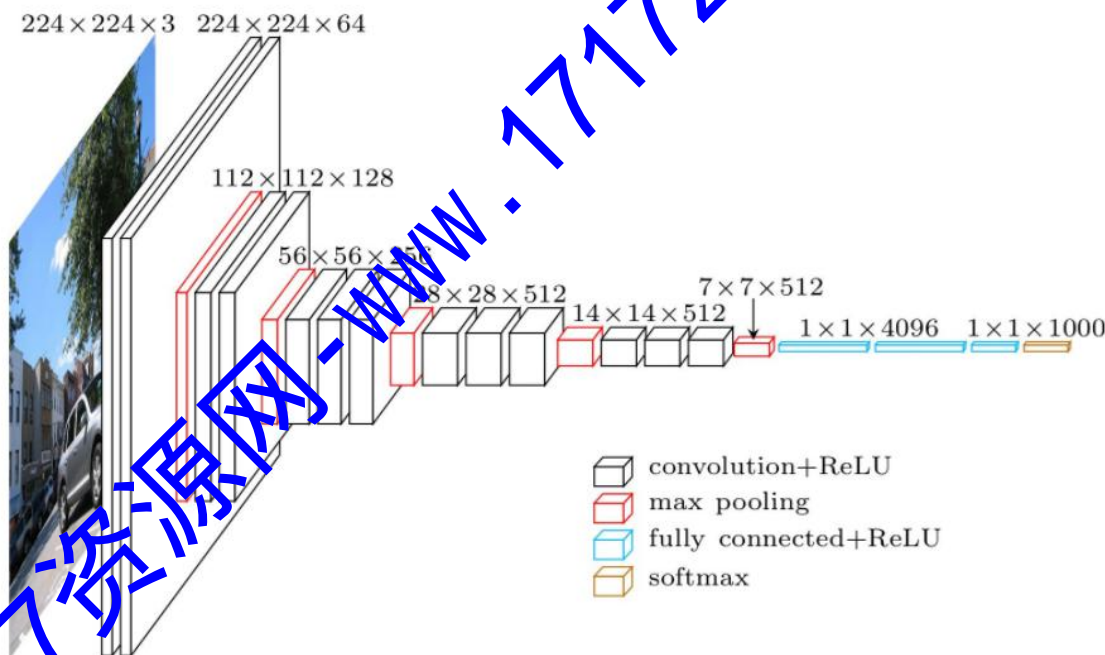
(a)



(b)

✓ CTPN网络架构:

✎ VGG特征提取模块:



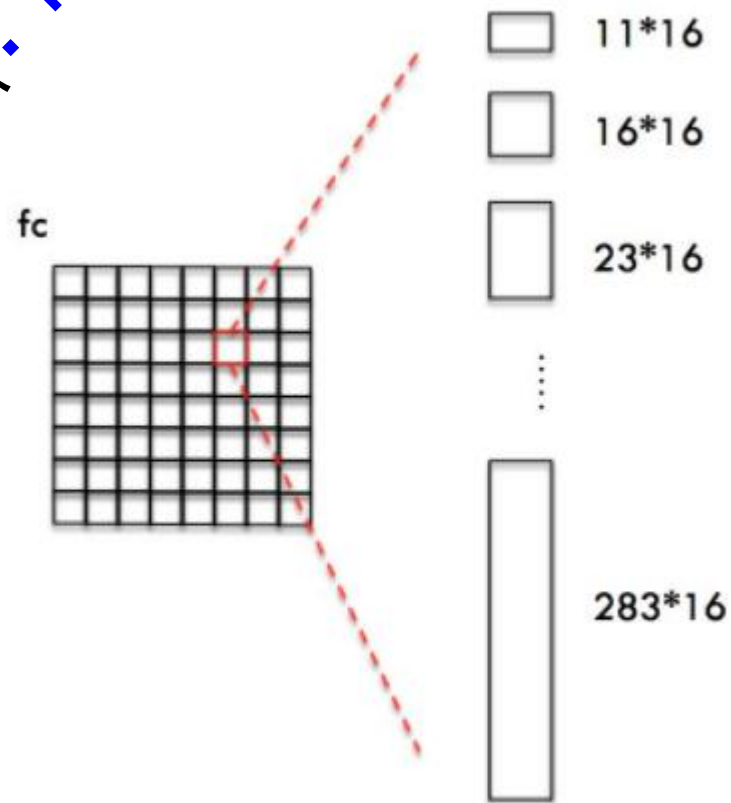
✎ 经过了4次池化操作,特征图中一个像素对应原始输入的16个像素

✓ CTPN网络:

✎ Anchor大小选择, 宽度固定, 长度选了10个

$\text{widths} = [16]$

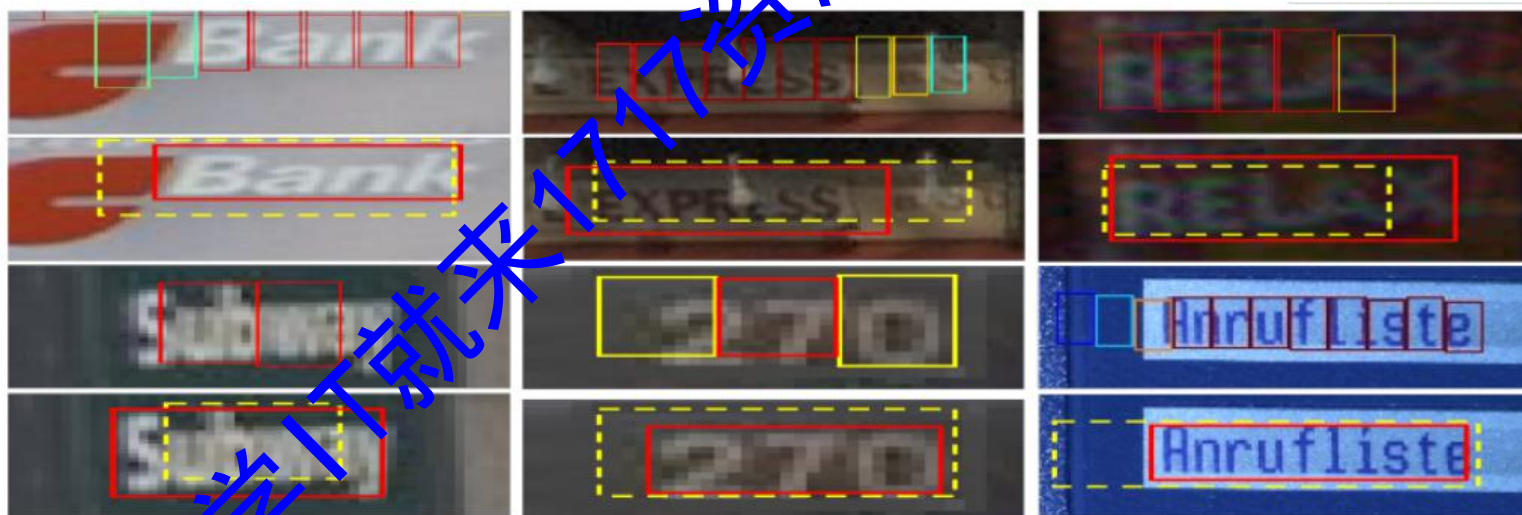
$\text{heights} = [11, 16, 23, 33, 48, 68, 97, 139, 198, 283]$



✓ CTPN网络架构:

✎ 输出结果包括了三部分: 2K得分, 2K回归, 1K边界调整

✎ 边界调整能使得文本检测框效果更好, 下列是调整后的结果:



✓ CTPN网络:

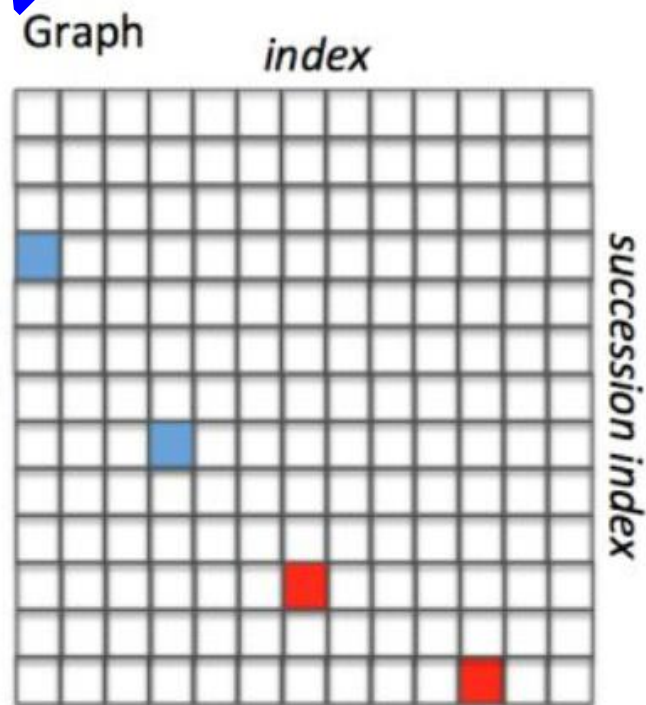
✎ 检测到每一个小块文本区域还需拼接成完整的文本区域:

✎ 规则, 分前向和后向两部分:

先前向走, 对于 X_i , 基于重合度 (0.7) 与位置距离 (50像素) 找到score值最大的 X_j , 接下来再反向走 (规则不变), 比较两次得分值大小来判断序列。



Build graph



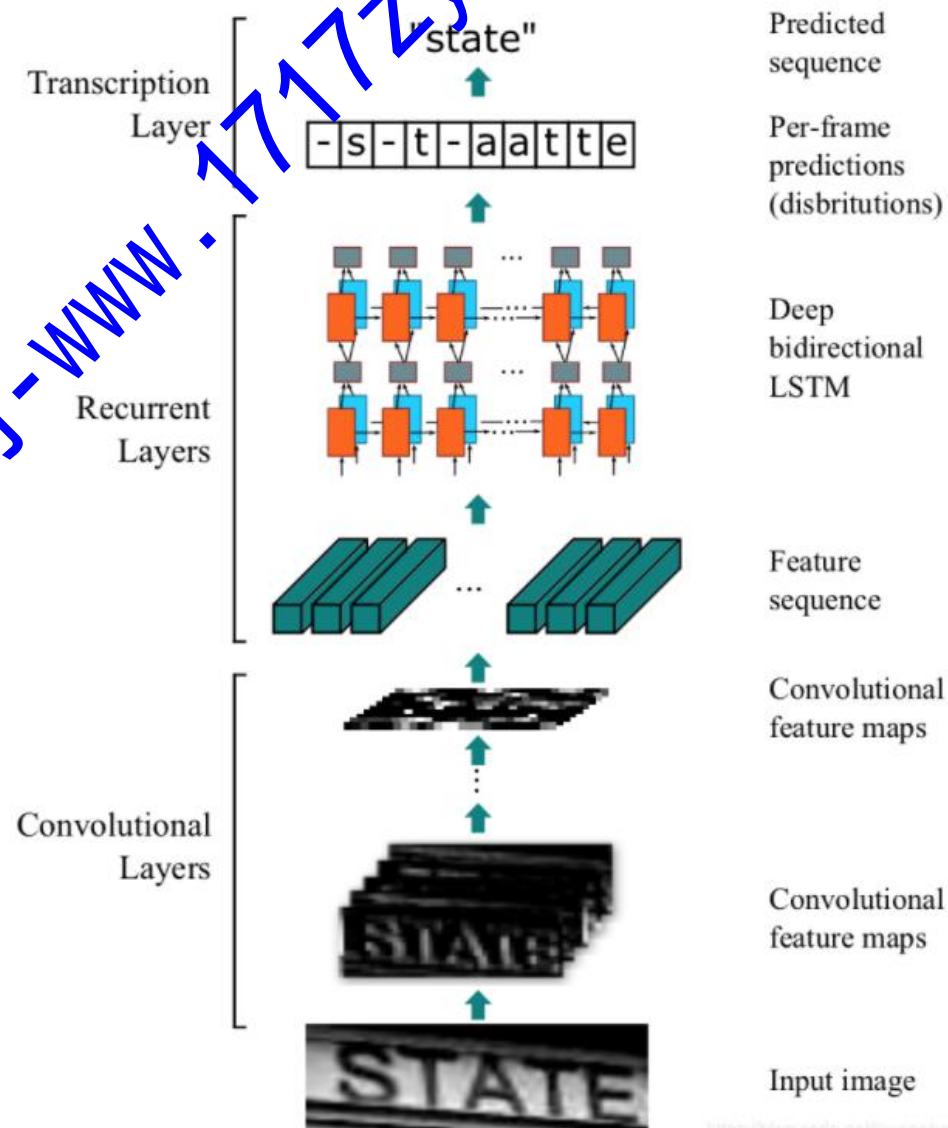
如果 $\text{score}_i \geq \text{score}_k$ 则这是一个长序列，反之则该序列被更长的序列所包含。

OCR

✓ CRNN算法:

✎ 一张图解释了网络架构:

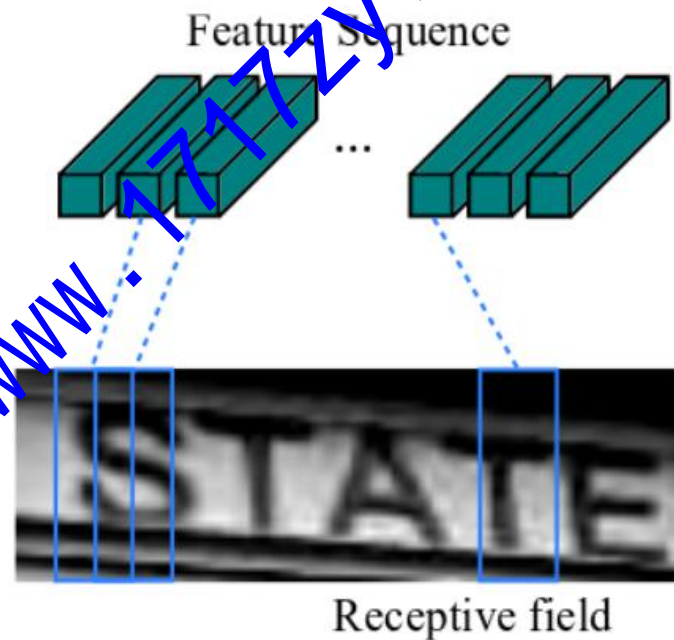
✎ 首先CNN进行特征提取, 接下来RNN进行序列特征提取, 最后得出预测结果即可。



OCR

✓ CRNN算法:

✎ 构建RNN的输入特征序列:



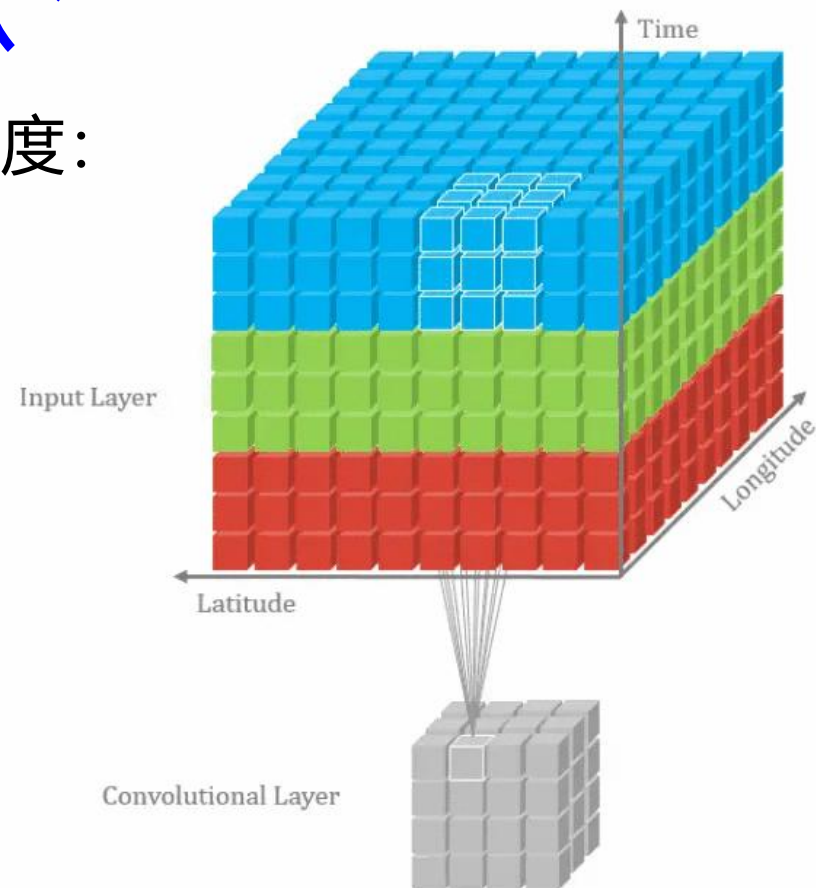
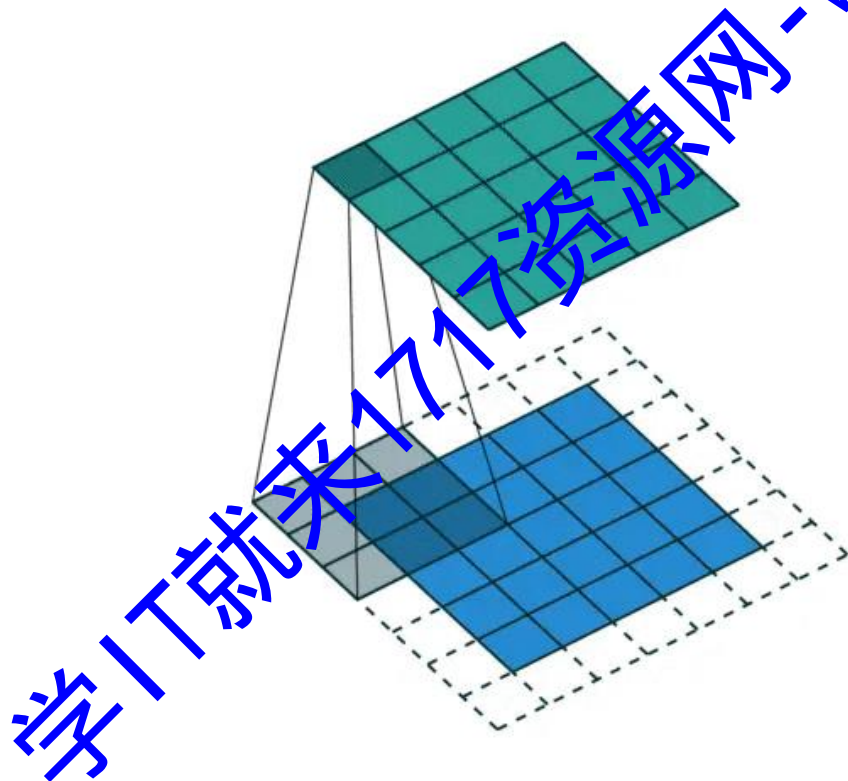
✎ 其中还涉及了CTC模块, 目的是对齐输入和输出结果:

h	h	e	ε	ε	l	l	l	ε	l	l	o
h	e	ε			l			ε	l	o	
h	e				l				l	o	
h	e	l	l	o							

视频分析3D卷积

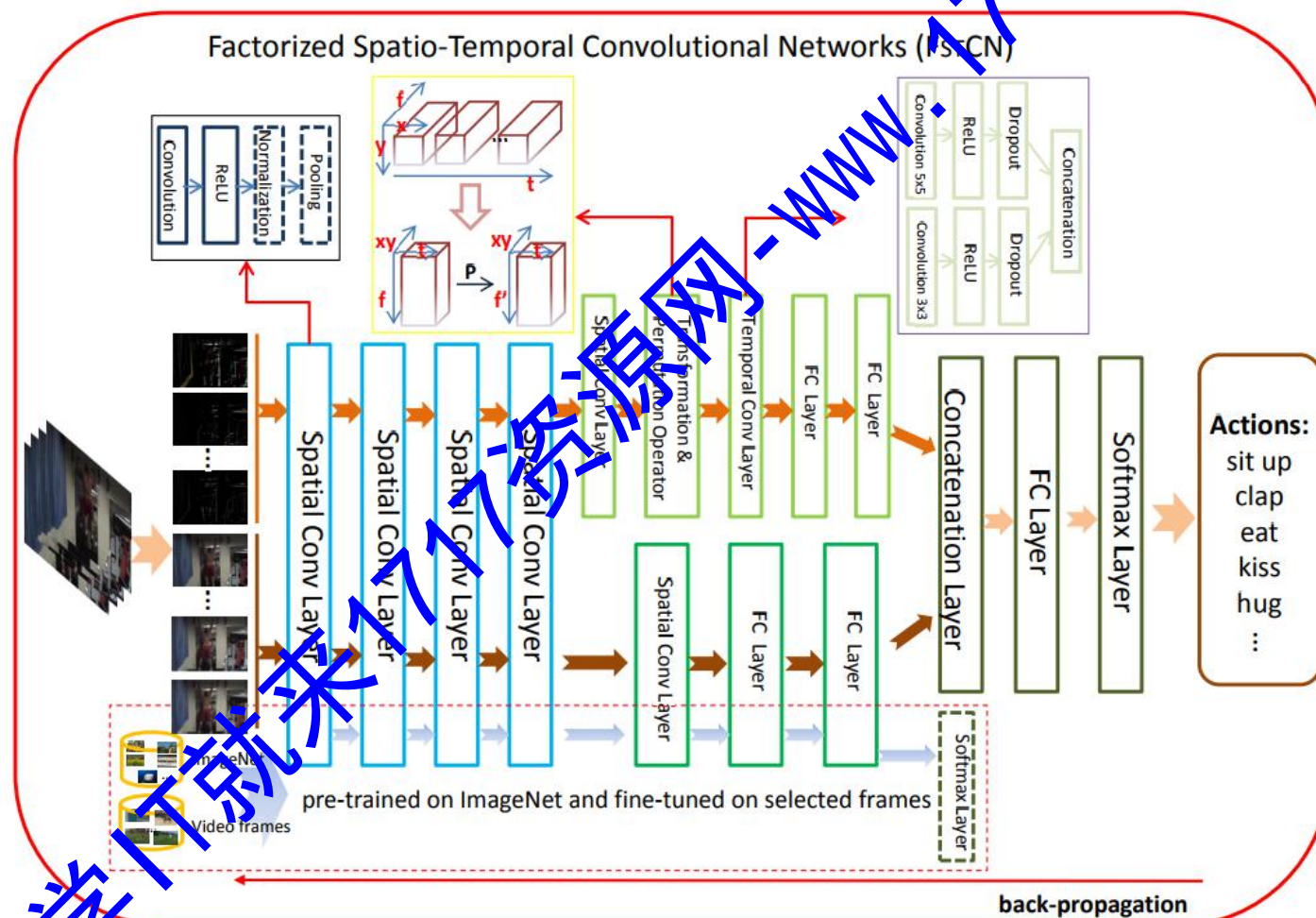
✓ 视频分析

✎ 3D卷积与2D卷积的区别，其中多了一个时间维度：



视频分析3D卷积

✓ C3D网络架构:



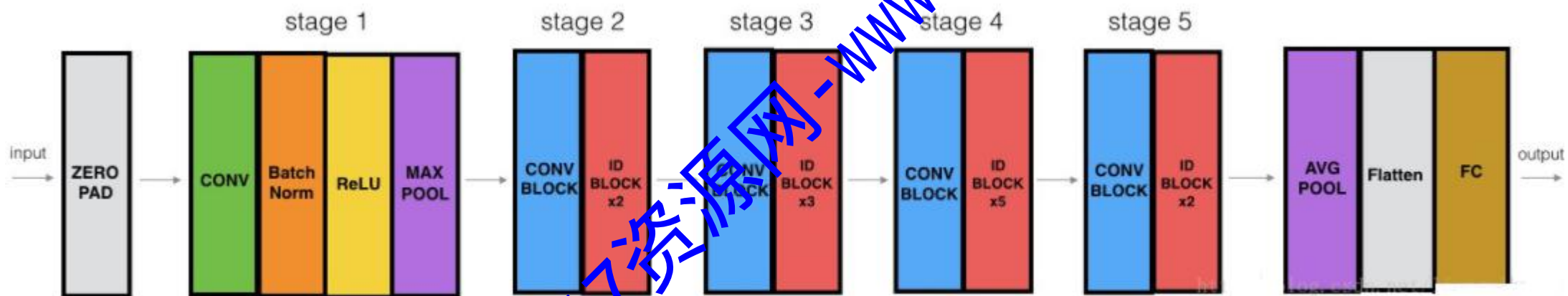
视频分析3D卷积

✓ UCF-101数据集:



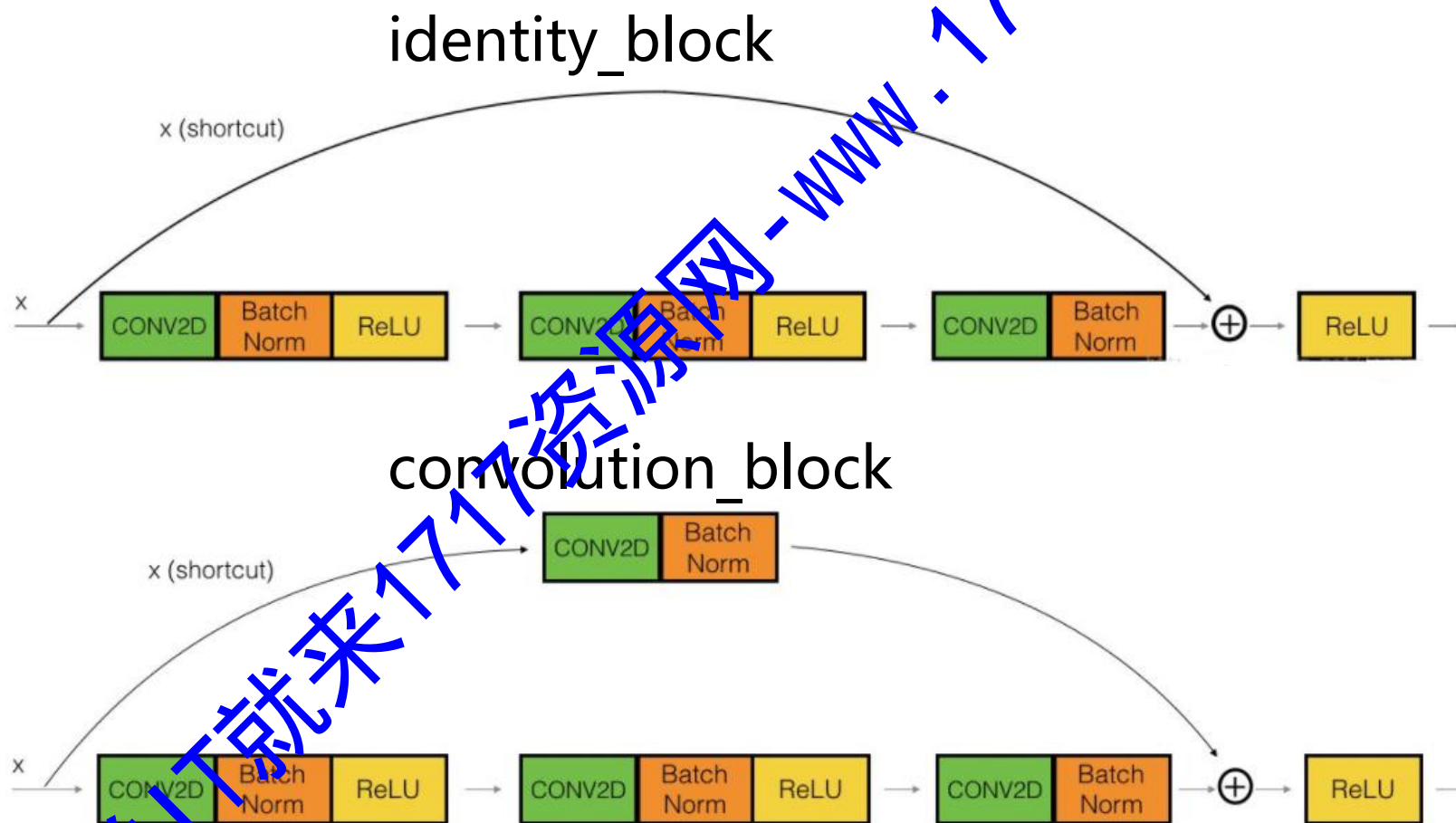
Resnet

✓ 网络结构

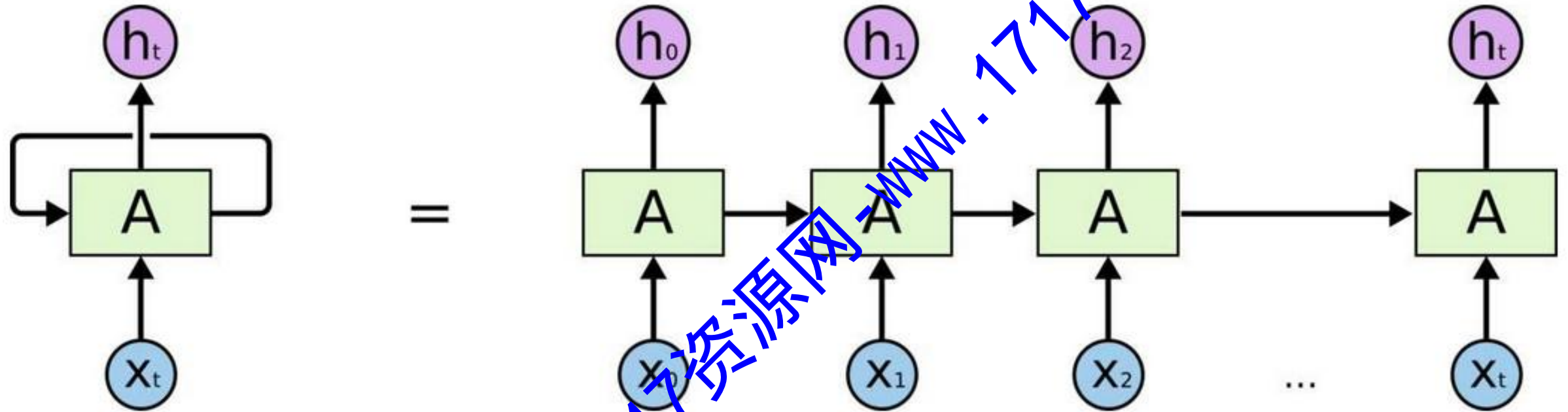


Resnet

✓ 网络结构



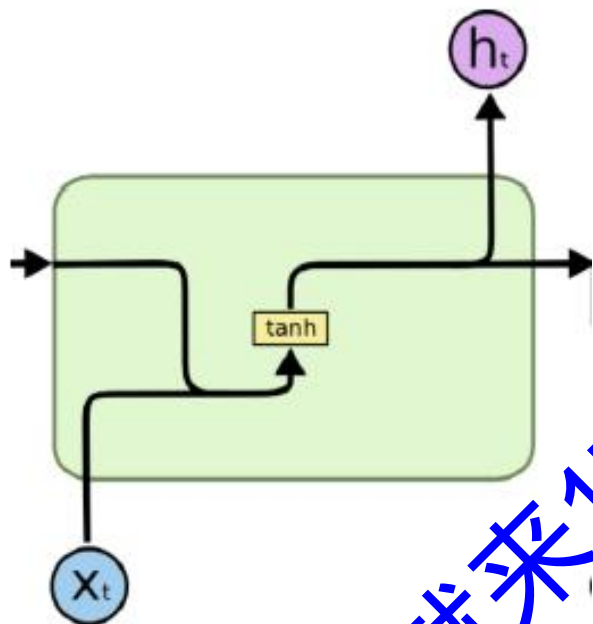
RNN



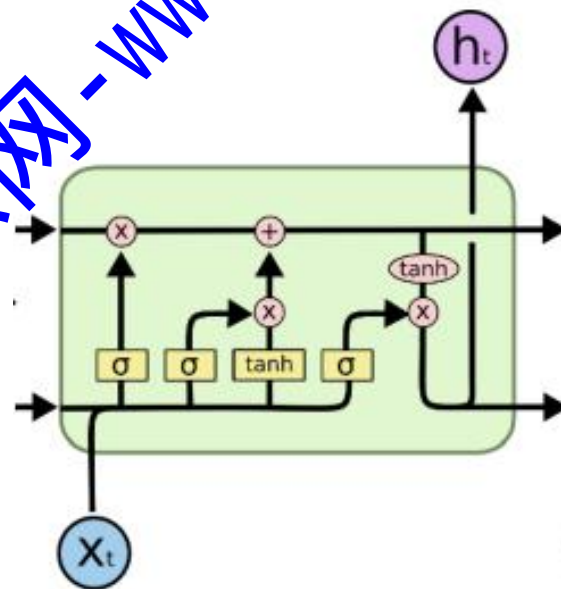
I am Chinese, I Love China

LSTM

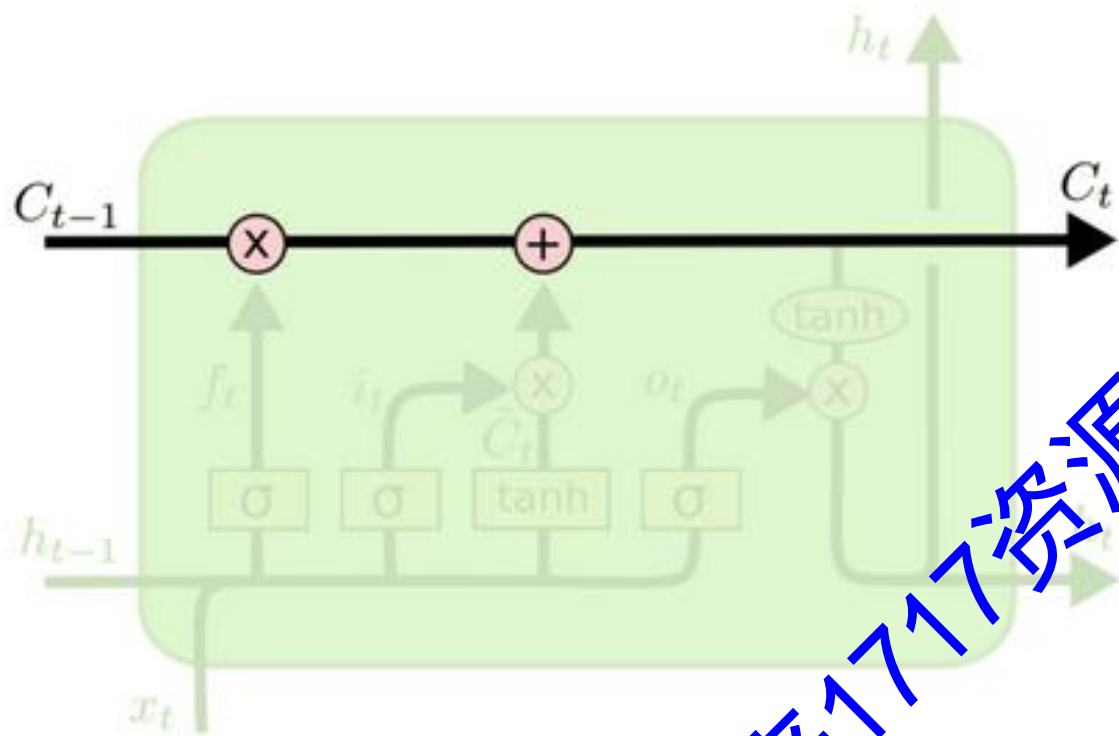
RNN



LSTM



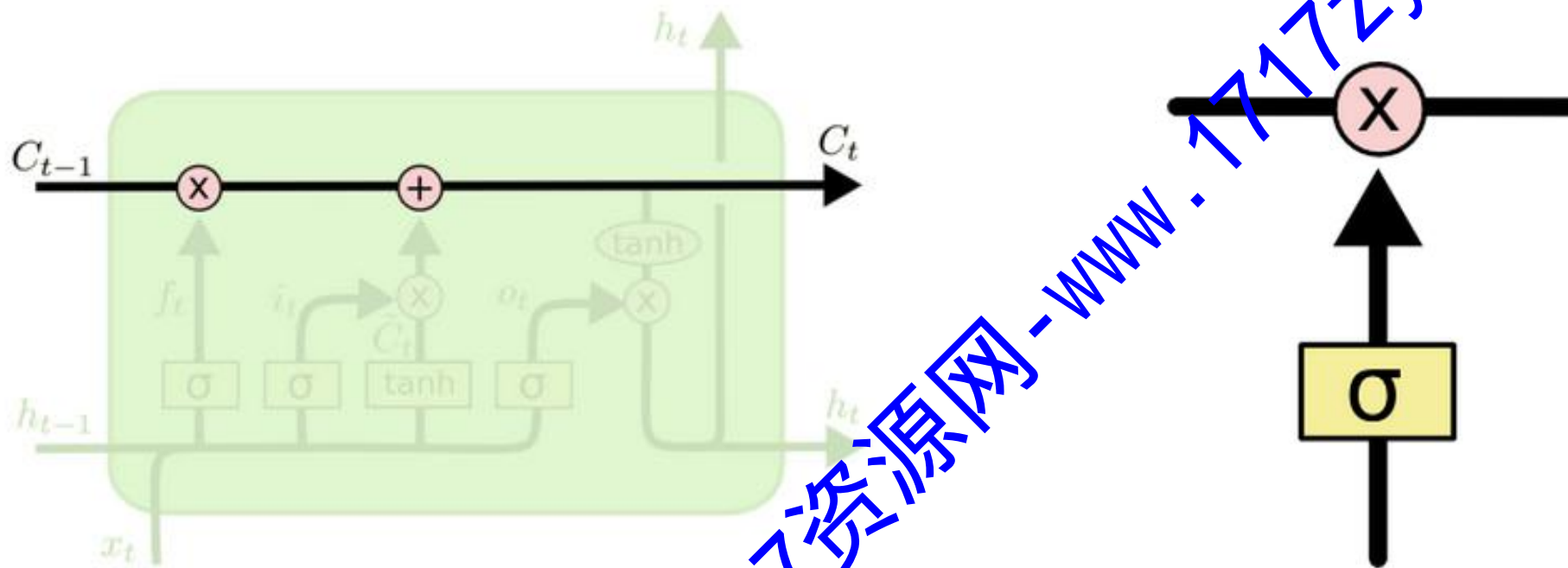
LSTM



C. 控制参数

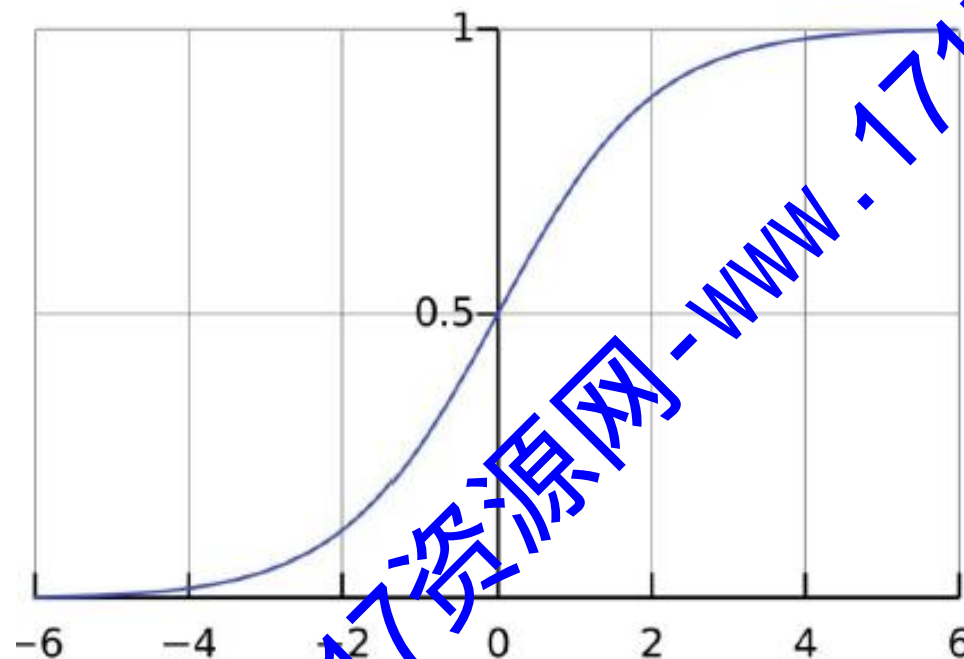
决定什么样的信息会被保留什么样的会被遗忘

LSTM



门是一种让信息选择式通过的方法
sigmoid 神经网络层和一乘法操作

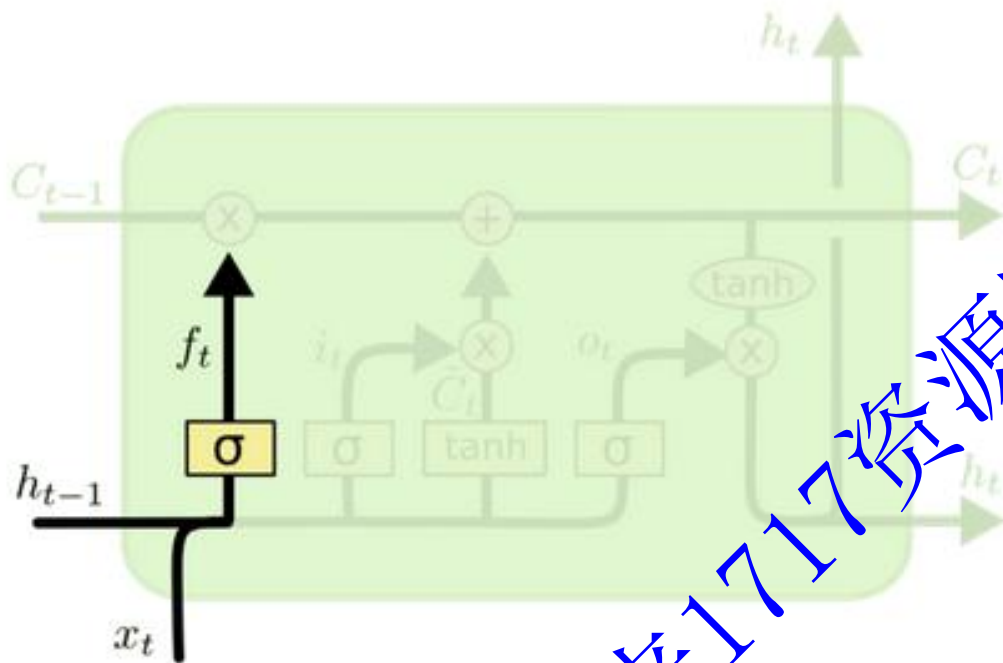
LSTM



Sigmoid函数

Sigmoid 层输出 0 到 1 之间的数值，描述每个部分有多少量可以通过。 0 代表“不许任何量通过”， 1 就指“允许任意量通过”！

LSTM

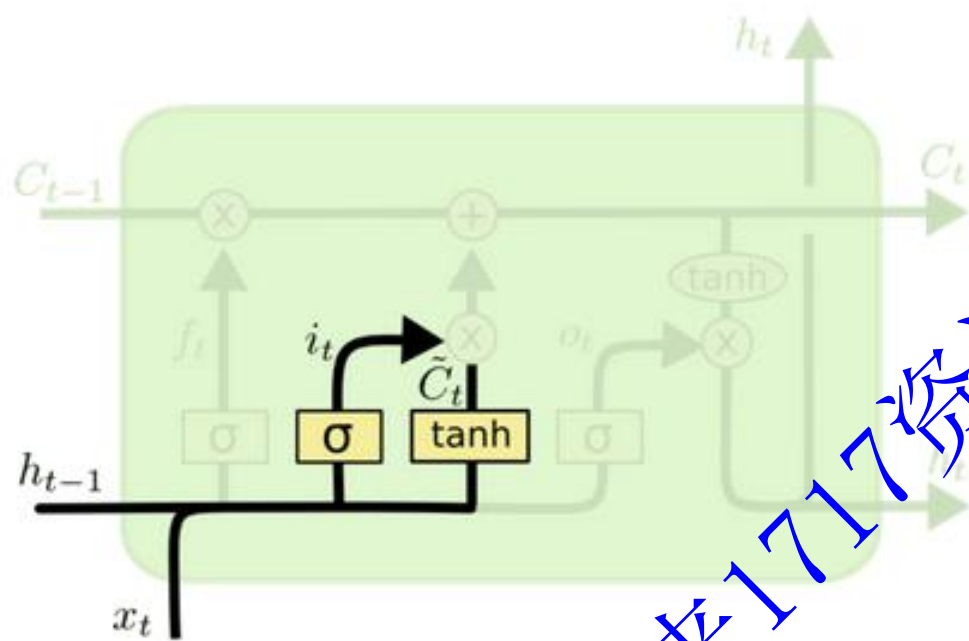


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

f_t 与 C_{t-1} 计算决定丢弃什么信息

决定丢弃信息

LSTM



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

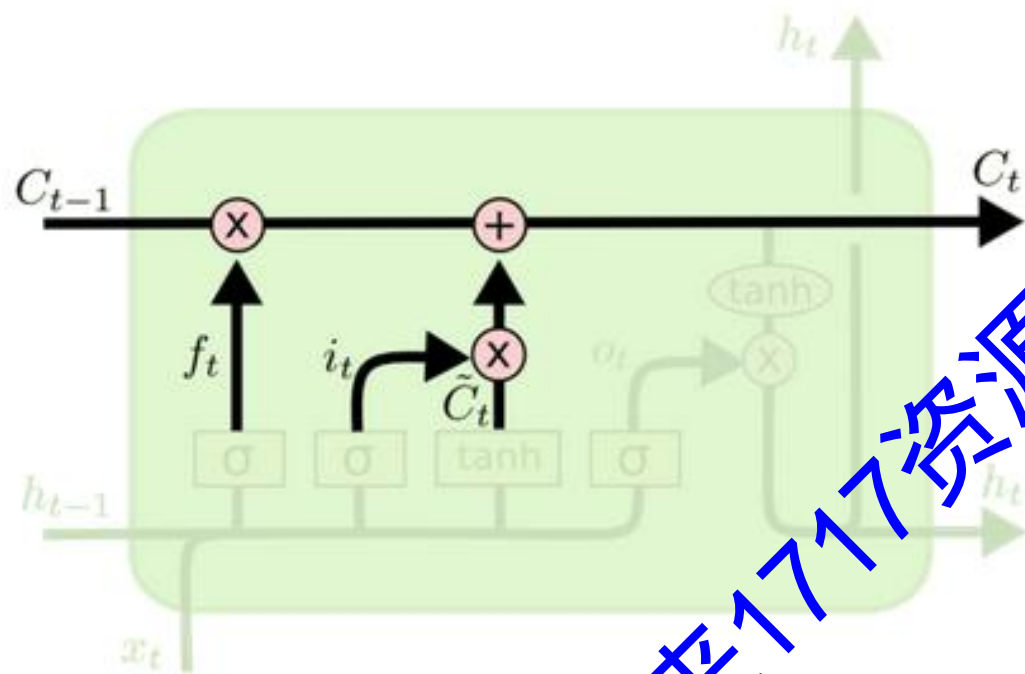
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

i_t 要保留下来的新信息

C_t 新数据形成的控制参数

确定更新的信息

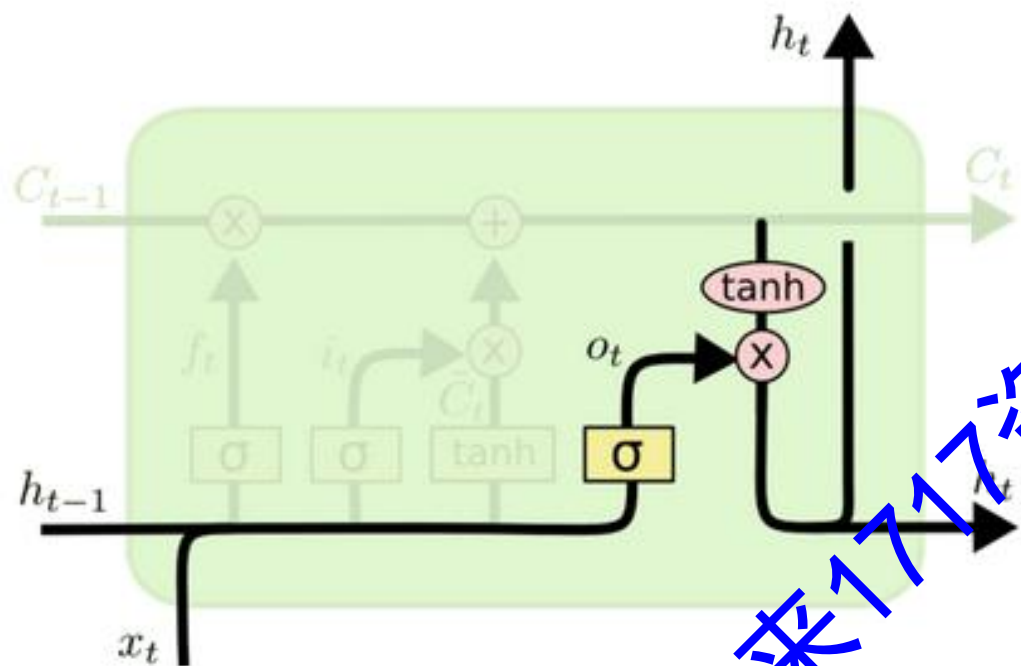
LSTM



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

更新细胞状态

LSTM



$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

利用新的控制参数产生输出

输出信息

LSTM

