

猫狗大战与算式识别

2019-04-07

主讲内容

猫狗大战

算式识别

猫狗大战

1 项目介绍

2 数据预处理

3 解决方法

项目介绍

问题的本质

计算视觉任务

目标检测
语义分割
目标追踪
OCR文字识别
图像分类 (猫狗大战)

分类是计算机视觉中最为常见的问题，图像也是一种数据，和大家前面机器学习分类的项目一样，同样可以使用前面的机器学习算法来做，比如SVM，RF，KNN，Boosting等，可以把这些传统机器学习方法和深度学习方法的结果进行对比，就会发现为啥现在是深度学习的天下

猫狗大战 -- 项目介绍



项目介绍

问题的本质

计算视觉任务

目标检测
语义分割
目标追踪

图像分类 (猫狗大战)

数据来源

Kaggle (Kaggle是一个数据分析的竞赛平台)

网址 : <https://www.kaggle.com/competitions>

获取数据步骤



注册

登录

参加比赛

下载数据

猫狗大战 -- 项目介绍

General

InClass


Sort by

Grouped


All Categories

Search competitions


2 Entered Competitions




LANL Earthquake Prediction
Can you predict upcoming laboratory earthquakes?
Research · 2 months to go · 🧠 earth sciences, physics, signal processing



\$50,000
1182/2313




Histopathologic Cancer Detection
Identify metastatic tissue in histopathologic scans of lymph node sections
Playground · 6 days ago · 🧠 research, oncology and cancer, medicine




Knowledge
1,157 teams


22 Active Competitions




Two Sigma: Using News to Predict Stock Movements
Use news analytics to predict stock price performance
Featured · Kernels Competition · 3 months to go · 🧠 news agencies, time series, finance, money




\$100,000
2,927 teams




Jigsaw Unintended Bias in Toxicity Classification
Detect toxicity across a diverse range of conversations
Featured · Kernels Competition · 3 months to go · 🧠 biases, nlp, text data




\$65,000
379 teams



Santander Customer Transaction Prediction
Can you identify who will make a transaction?
Featured · 5 days to go · 🧠 banking, tabular data, binary classification



\$65,000
8,652 teams



Dogs vs. Cats Redux: Kernels Edition

Distinguish images of dogs from cats

1,314 teams · 2 years ago

Overview

Data

Kernels

Discussion

Leaderboard

Rules

Team

My Submissions

Late Submission

Data Description

The train folder contains 25,000 images of dogs and cats. Each image in this folder has the label as part of the filename. The test folder contains 12,500 images, named according to a numeric id. For each image in the test set, you should predict a probability that the image is a dog (1 = dog, 0 = cat).

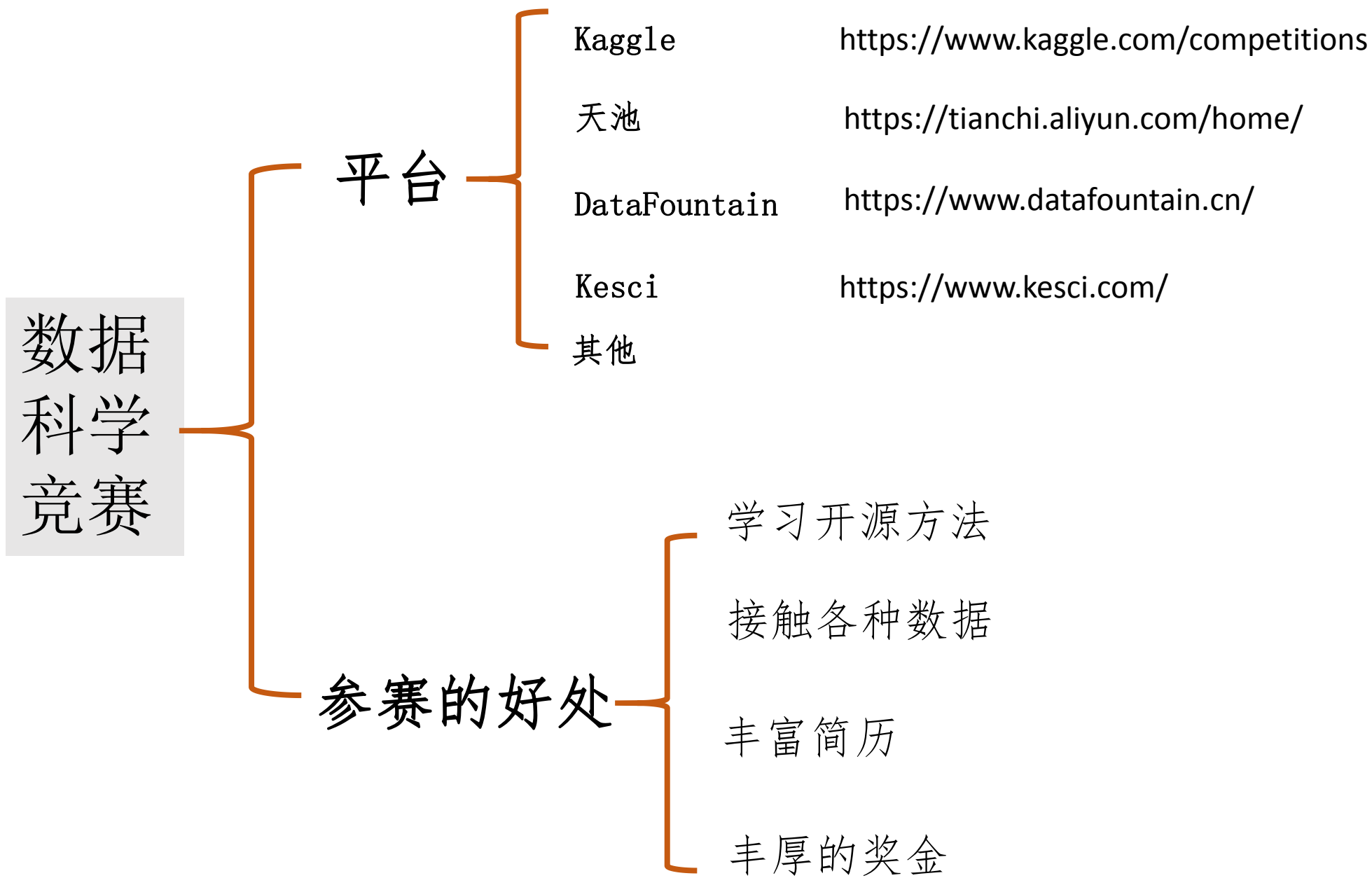
猫狗大战 -- 项目介绍



#	Team Name	Kernel	Team Members	Score	Entries	Last
1	Cocostarcu			0.03302	29	2y
2	guangsha			0.03305	34	2y
3	malr87			0.03483	89	2y
4	Bojan Tunguz			0.03507	435	2y
5	DeepBrain			0.03518	56	2y
6	lefant			0.03580	84	2y
7	matview			0.03778	40	2y
8	Bancroftway Systems [Andy ...			0.03804	41	2y
9	Arvinder Chopra			0.03805	5	2y
10	Ranjeeta			0.03807	5	2y
11	Adarsh Tadimari			0.03838	12	2y
12	Yehya Abouelnaga			0.03882	50	2y
13	HeshamEraqi			0.03889	53	2y
14	HMen			0.03928	12	2y

但是因为该比赛结束，现在提交结果不会先排行榜显示，可以对照排行榜，计算排名

猫狗大战 -- 项目介绍



猫狗大战 -- 项目介绍



[Competitions](#)

[Datasets](#)

[Kernels](#)

[Discussion](#)

[Learn](#)

[...](#)

[首页](#)

[全部赛事](#)

[数据集](#)

[社区](#)

[new](#)

[专家库](#)

[企业办赛](#)

[个人主页](#)

[帮助](#)

[登录](#)

[注册](#)

[K-Lab](#)

[项目](#)

[数据集](#)

[比赛](#)

[任务](#)

[首页](#)

[天池大赛](#)

[天池实验室](#)

[数据集](#)

[技术圈](#)

[En/中](#)

[Models](#)

[课程](#)

[讨论区](#)

[anagege](#)

[退出](#)

[DC竞赛](#)

[DC学院](#)

[DC直聘](#)

[AI童学](#)

[帮助](#)

[English](#)

[HOME](#)

[PROGRAM](#)

[ATTENDING](#)

[NEWS & CALLS](#)

[KDD CUP](#)

[SPONSORS](#)

[ORGANIZING COMMITTEE](#)

猫狗大战 -- 项目介绍



项目介绍

问题的本质

计算视觉任务

目标检测

语义分割

目标追踪

图像分类 (猫狗大战)

数据来源

Kaggle (Kaggle是一个数据分析的竞赛平台)

网址：<https://www.kaggle.com/competitions>

获取数据步骤

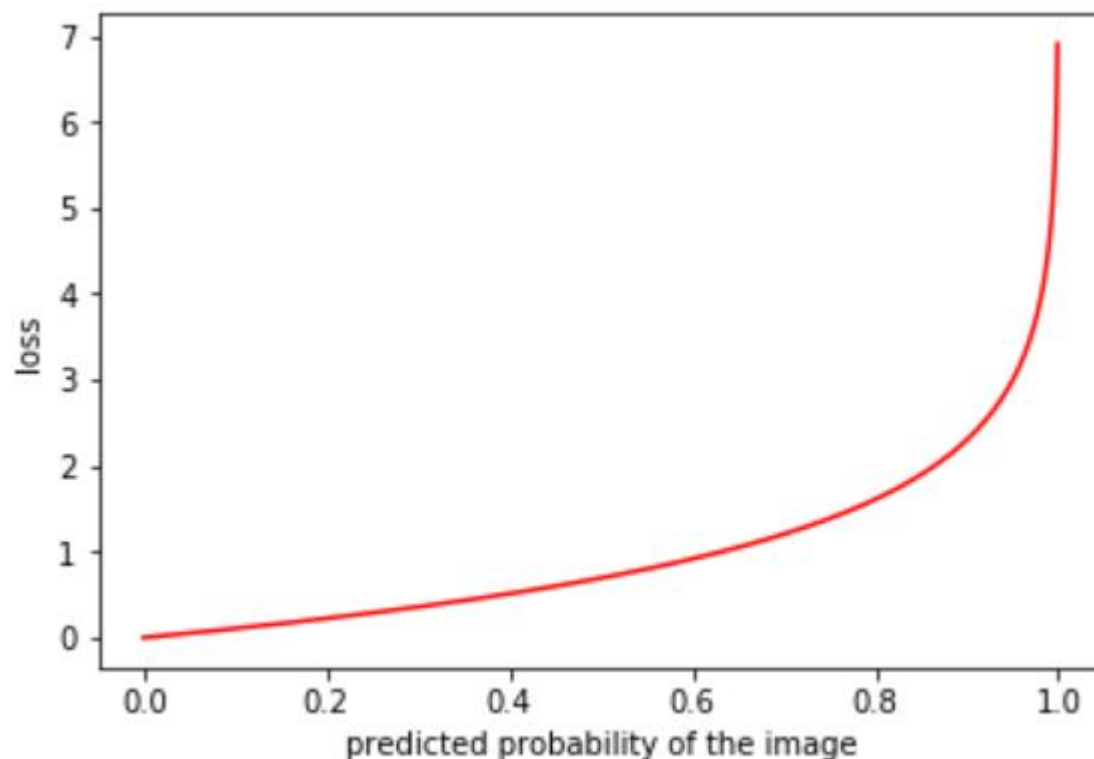
评价方式

$$\text{LogLoss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

$$\text{LogLoss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where

- n is the number of images in the test set
- \hat{y}_i is the predicted probability of the image being a dog
- y_i is 1 if the image is a dog, 0 if cat
- $\log()$ is the natural (base e) logarithm



将预测结果固定在
[0.005,0.995]区间之间，
因为如果预测正确，预测
为0.995和1，0.005和0
损失相差不大，但是预测
错误，相差很大，可以
从图像两端看出

trick:clip(0.005,0.995)

评价方式

猫狗大战 -- 数据预处理

数据预处理

数据认识

训练数据

猫 12500 (0)

测试数据

12500 张图片

猫狗大战 -- 数据预处理

数据预处理

数据认识

数据可视化探索

训练数据

测试数据

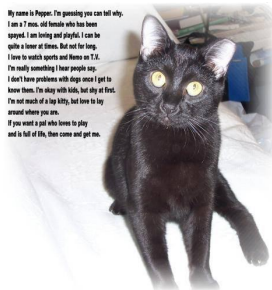
猫 12500 (0)

12500 张图片

猫狗大战 -- 数据预处理



Just waiting for my forever family...



My name is Pepsi. The guessing you can tell why.
I am a 7 year old female who has been
adopted. I am loving and playful. I can be
quite a lover at times. But not for long.
I love to watch sports and movies on TV.
I'm really something I hear people say.
I don't have problems with dogs since I get to
know them. The dog with kids, but only at first.
I'm not much of a dog lover, but have to say
around where you are.
If you want a pet who loves to play
and is full of life, then come and get me.



数据预处理

数据认识

训练数据

猫 12500 (0)

测试数据

12500 张图片

数据可视化探索

- 大多数照片猫和狗占据整个图片，但部分较小
- 猫和狗背景各种各样，存在干扰
- 猫和狗背景图像大小不一样
- 猫和狗在图片中的位置不定，大多数居中

猫狗大战 -- 数据预处理

数据预处理

数据认识

训练数据

猫 12500 (0)

测试数据

12500 张图片

数据可视化探索

大多数照片猫和狗占据整个图片，但部分较小

猫和狗背景各种各样，存在干扰

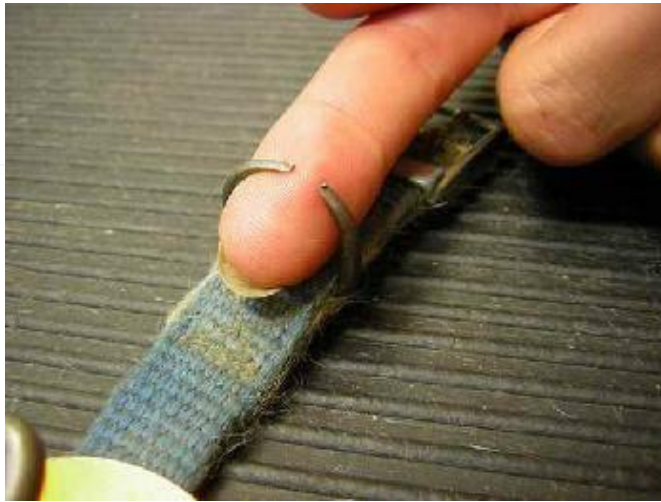
猫和狗背景图像大小不一样

猫和狗在图片中的位置不定，大多数居中

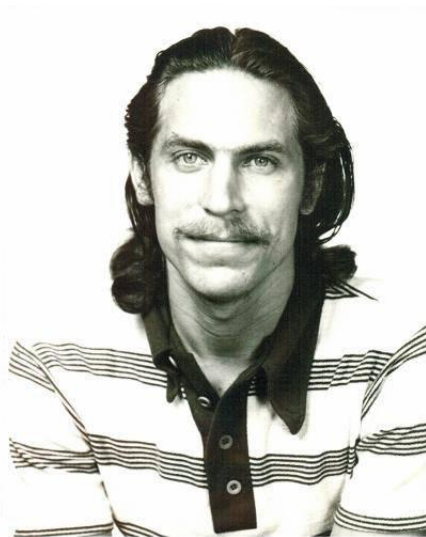
数据异常处理

与前面机器学习项目一样，图像也是数据，也需要异常处理

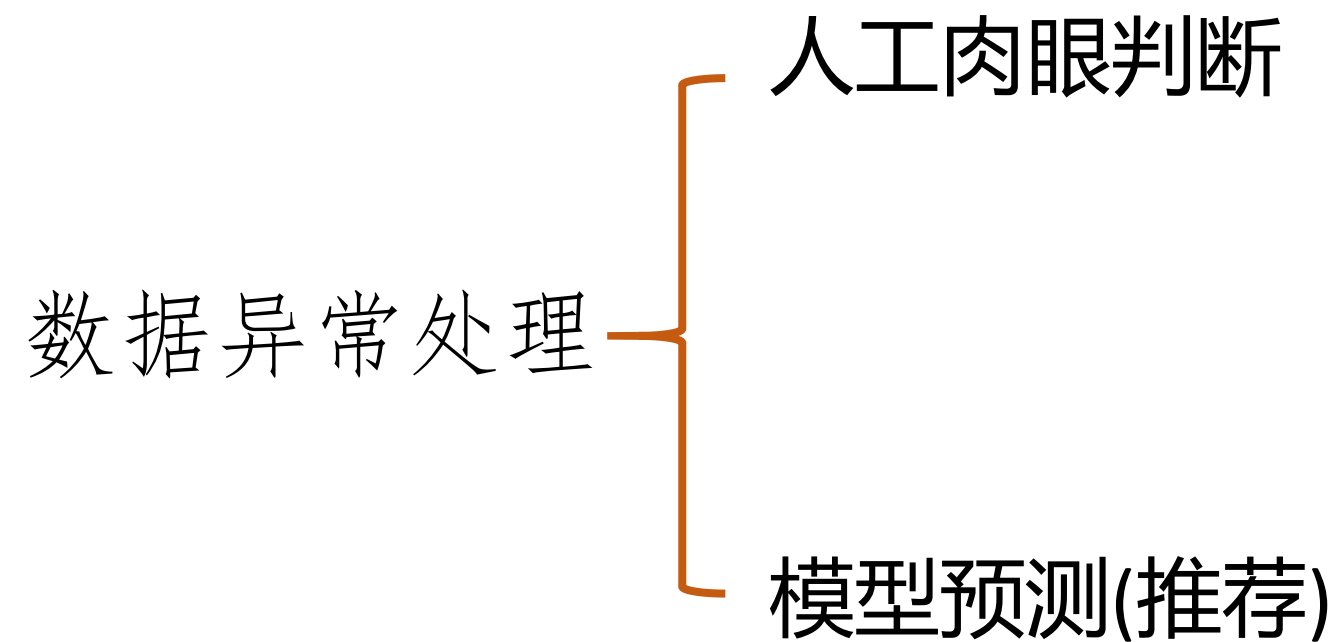
猫狗大战 -- 数据预处理



达学城
ACITY



The Caring Containment Professionals.™



模型预测

原理

使用预训练模型，就是在Imagenet上训练好的模型，比如VGG，Inception系列，ResNet系列，这些模型训练的数据是ImageNet，ImageNet共1000个类别，其中包括狗的118种，猫共8种，另外，这些训练好的模型均开源，Keras,Tensorflow等框架model zoo均有，所以可以直接用这些模型预测，预测选取top-N，选择top-N预测非猫与非狗的图像

参考链接:

<https://zhuanlan.zhihu.com/p/34068451>

ImageNet各类别介绍:

<https://blog.csdn.net/zhangjunbob/article/details/53258524>

猫狗大战 -- 数据预处理

数据预处理

数据认识

训练数据

猫 12500 (0)

测试数据

12500 张图片

数据可视化探索

大多数照片猫和狗占据整个图片，但部分较小

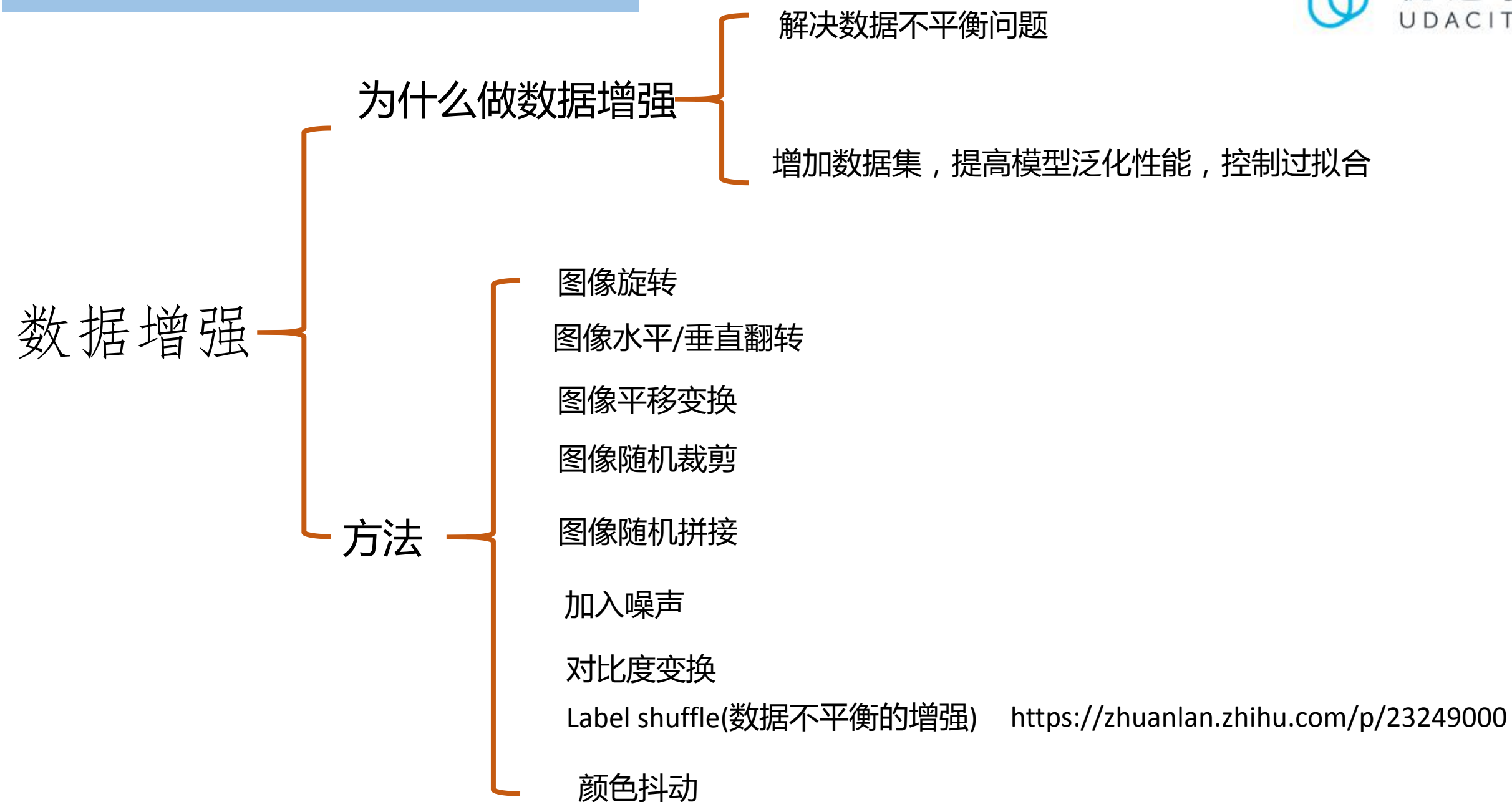
猫和狗背景各种各样，存在干扰

猫和狗背景图像大小不一样

猫和狗在图片中的位置不定，大多数居中

数据异常处理

数据增强



解决方法

传统机器学习方法（抛弃）

SVM

KNN

等等

卷积神经网络(CNN)

自己搭建CNN网络

Fine - Tune(推荐)

Fine-Tune

原理

因为很多网络，VGG，Resnet系列，Inception系列，都是在ImageNet上训练好的，如果自己完整去训练一个Resnet网络是十分困难的，首先是训练时间很长、计算能力要求高，从头训练也很困难。所以常用预训练模型，在这些模型上微调模型

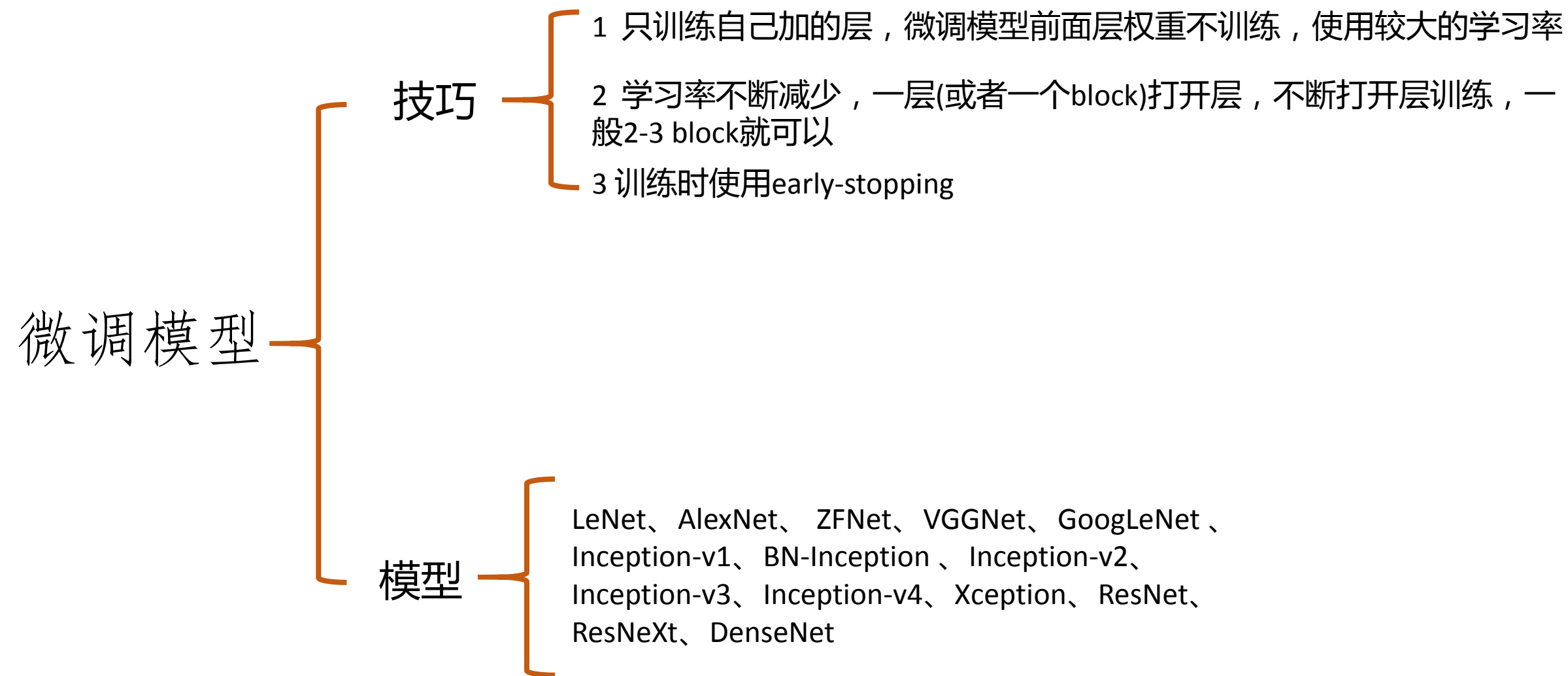
方法

把预训练模型当做特征提取器

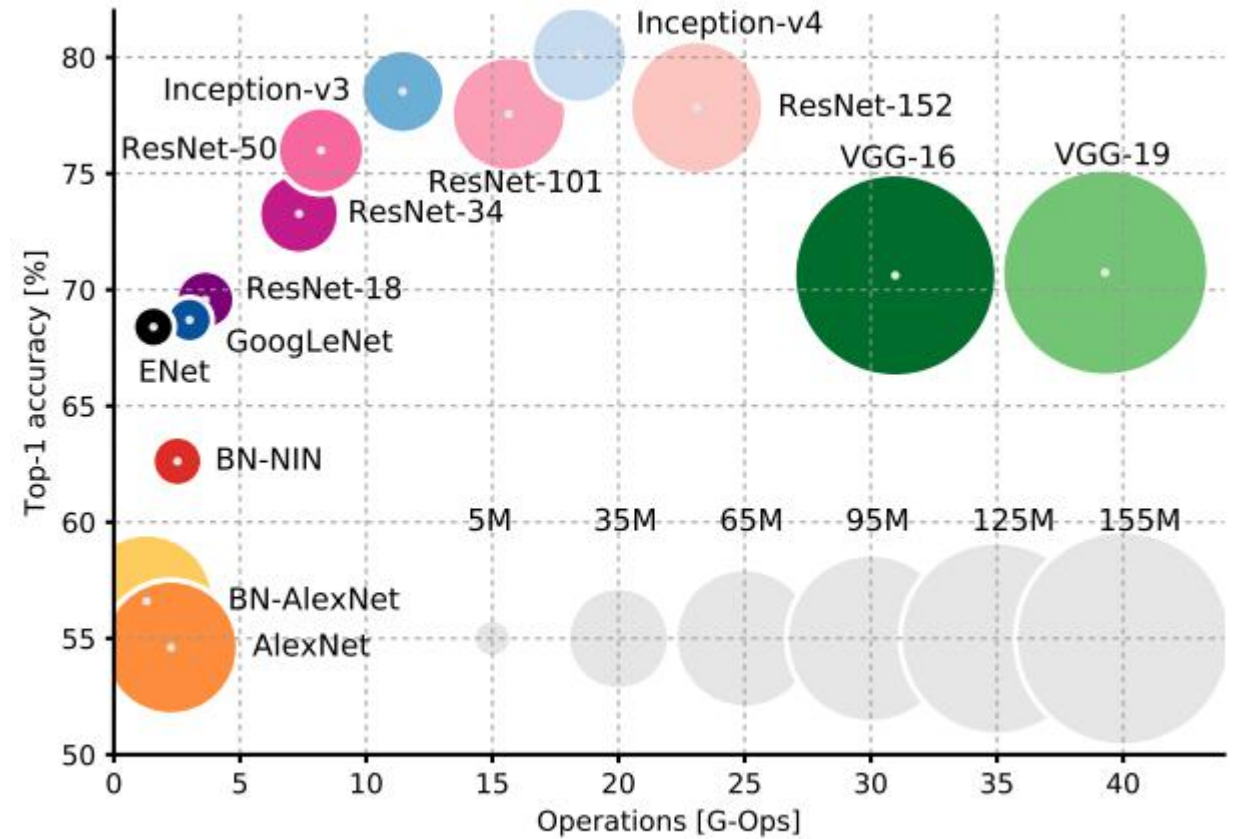
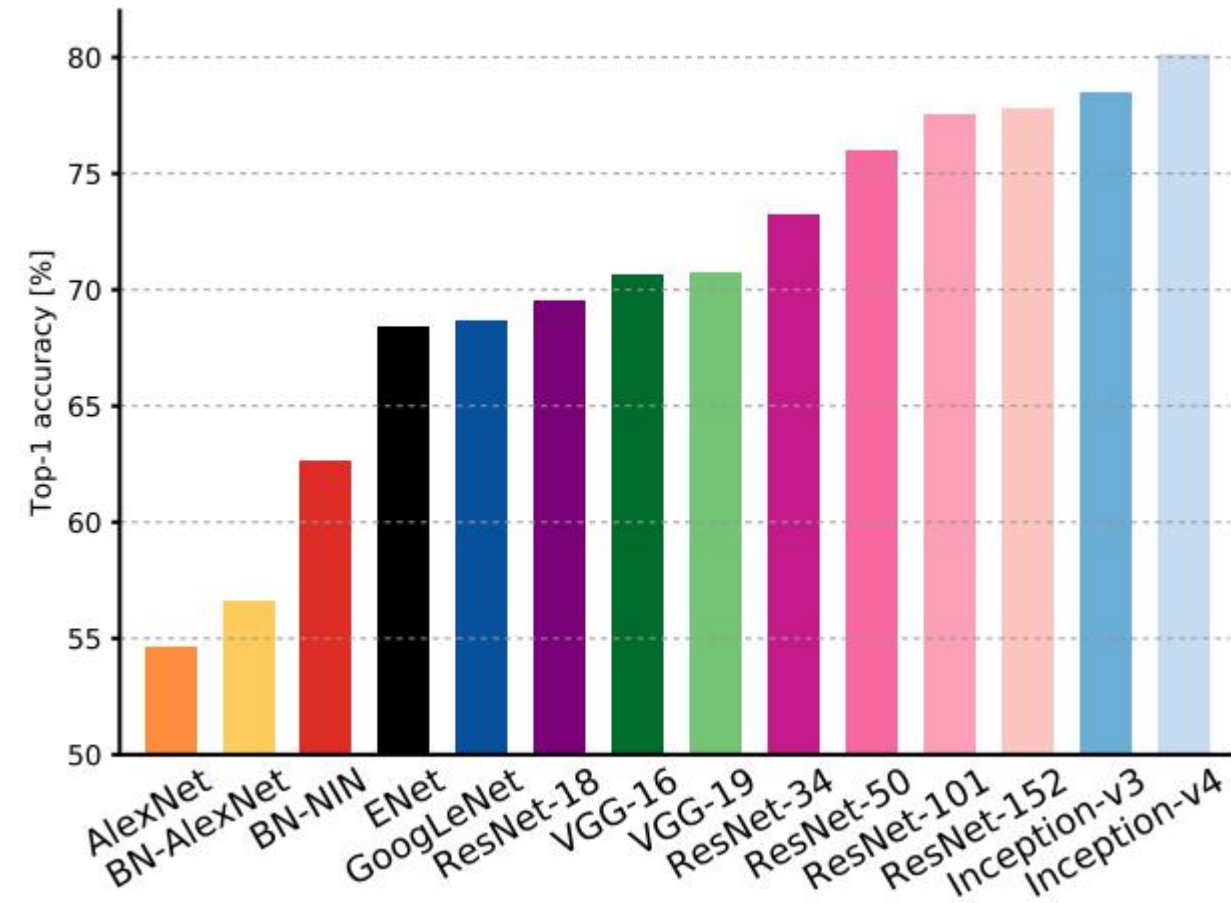
直接使用模型卷积层部分，把卷积层部分的特征导出来，使用softmax或者SVM训练模型用于分类

微调模型

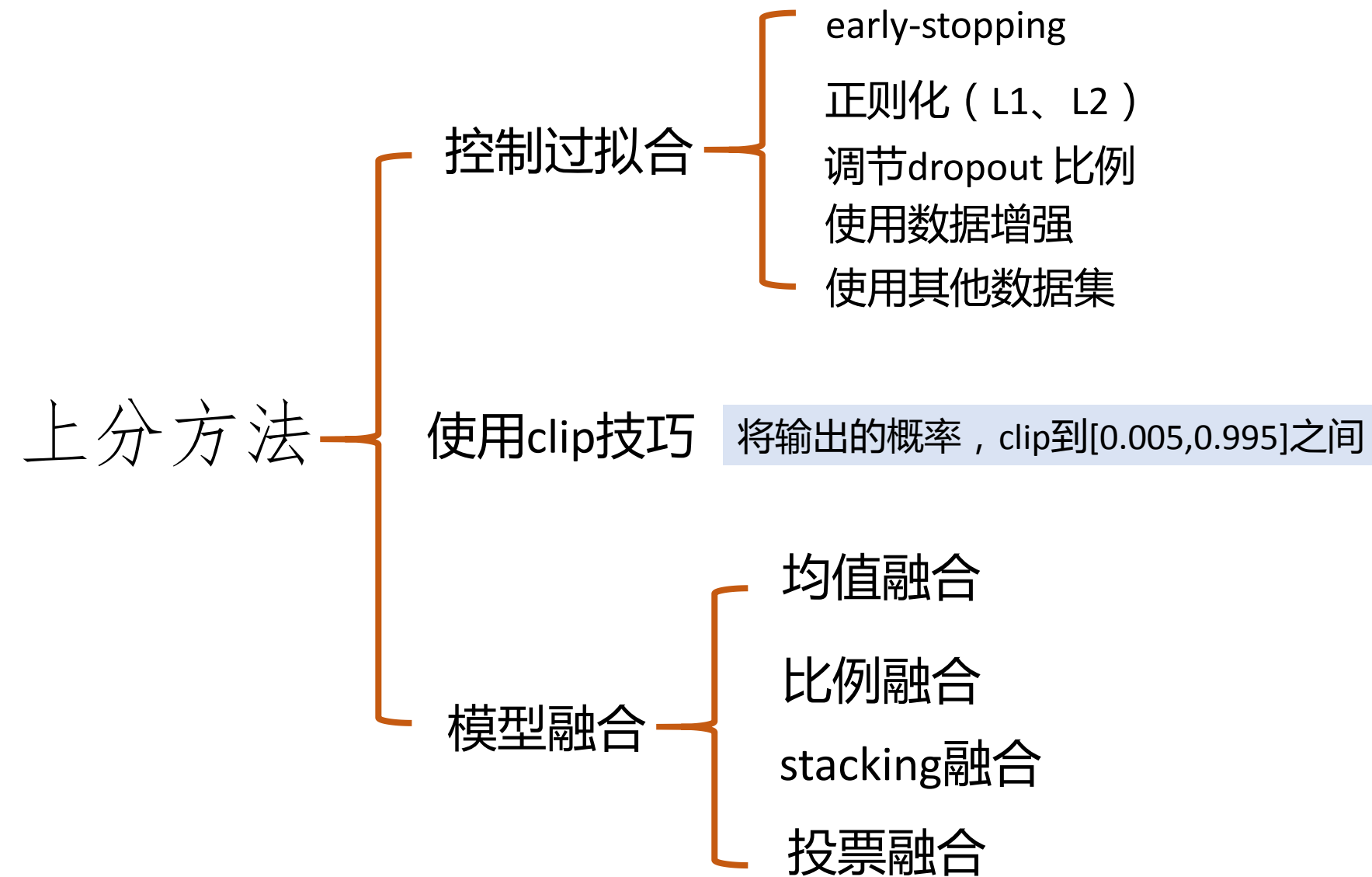
将预训练模型的最后一层全连接层删除，使用小学习率在自己的数据集上训练，各种层与参与训练。本质就是利用训练好的模型的权重作为自己模型的初始化参数

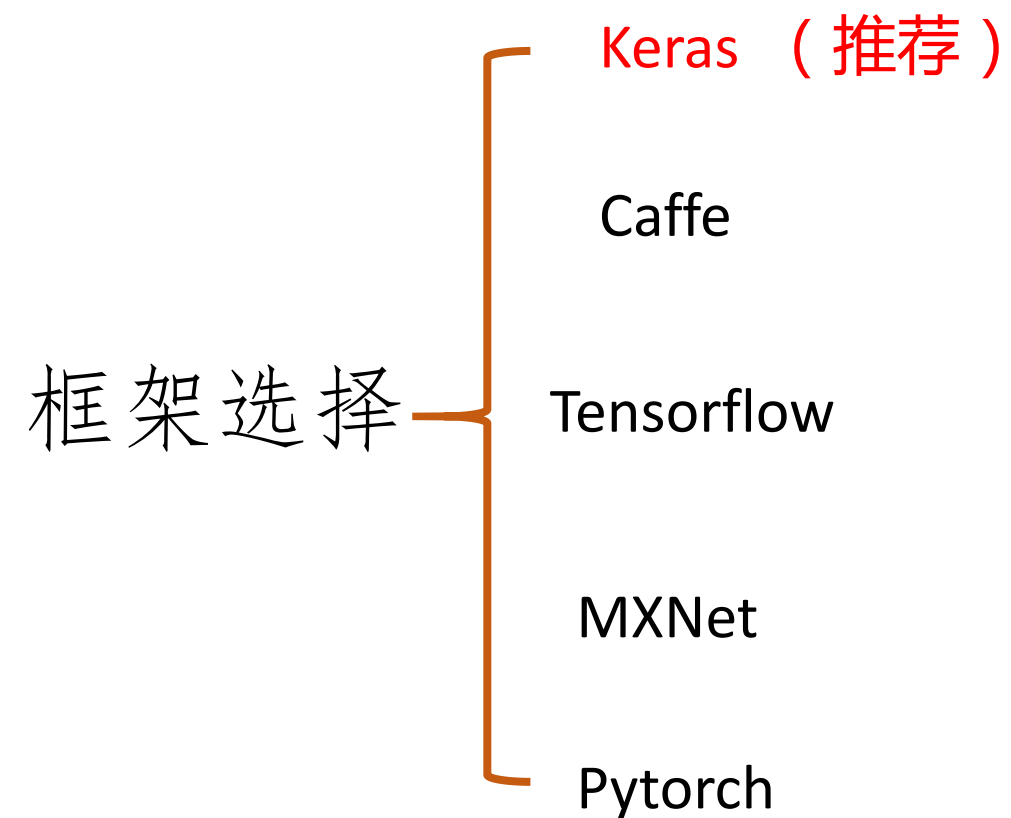


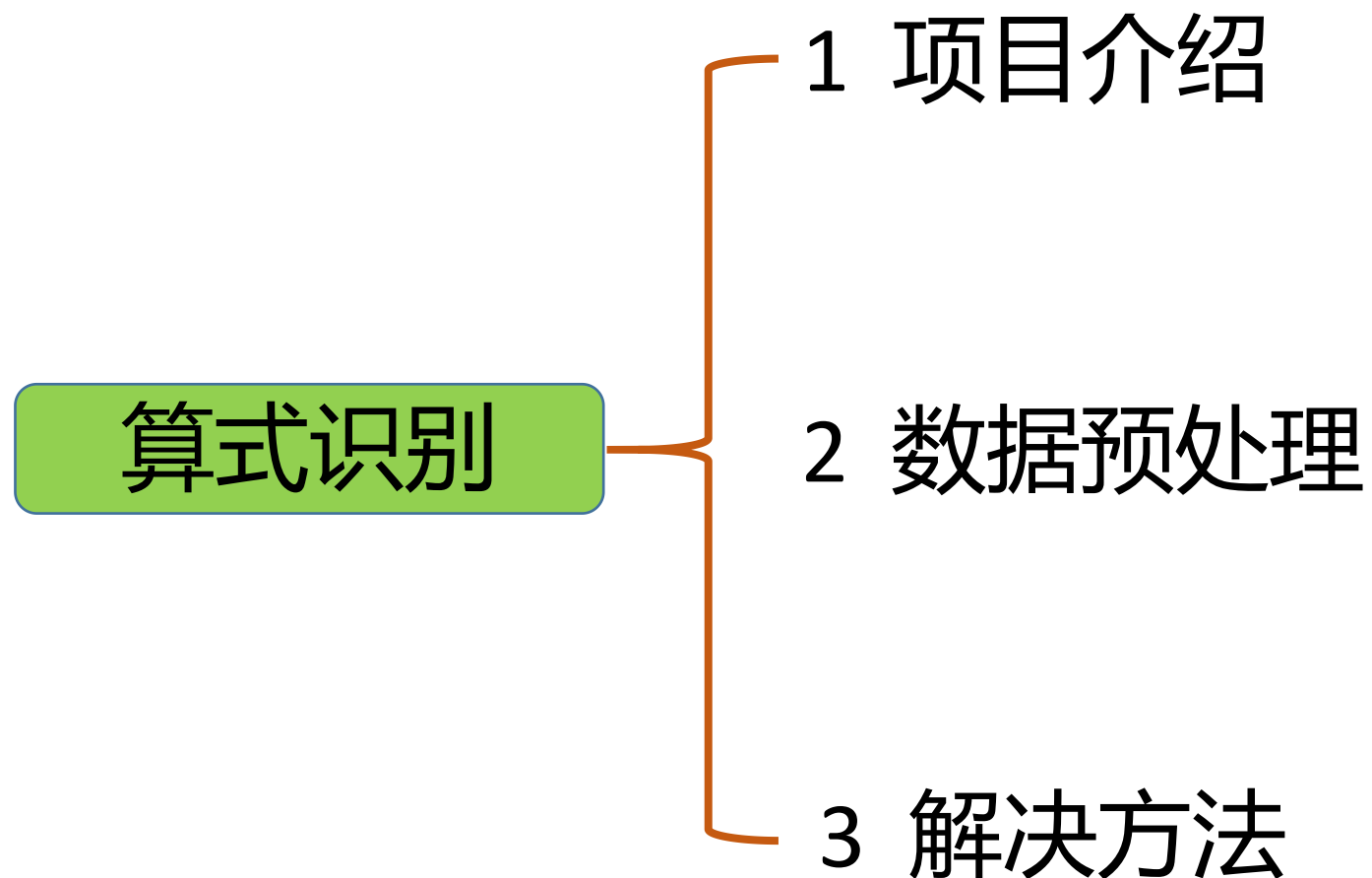
猫狗大战 -- 解决方法



模型表现







参考博客：https://github.com/ypwhs/baiduyun_deeplearning_competition

算式识别 -- 项目介绍



项目介绍

问题的本质

计算视觉任务

- 目标检测
- 语义分割
- 目标追踪
- 图像分类
- OCR文字识别

算式识别是序列识别问题

评价方式

准确率

算式识别 -- 数据预处理

数据预处理

数据认识

数据共100000张彩色图像，自己切分训练,验证，测试

图像中的字符包括0-9,+,-,*,(,),=

数据预处理

数据认识

数据共100000张彩色图像

图像中的字符包括0-9,+,-,*,(,),=

数据可视化探索

字符都是位于图像上下之间，所以不需要做文本检测工作

图像中有部分是空白的，字符不是沾满整个图像

图像中有字符东倒西歪，不是严格的摆放

图像中序列长短不一

算式识别 -- 数据预处理

$$(1-9)-4=-12$$

$$6-9*7=-57$$

$$1-(6-5)=6$$

$$6*(3+7)=76$$

$$6+5*9=51$$

$$(1*9)-9=54$$

$$(7-3)*3=-6$$

$$1-(2*5)=6$$

$$1-(2*5)=6$$

$$(8*9)+9=26$$

$$(6*8)+3=12$$

$$1*7*4=198$$

$$(1-8)*1=-6$$

$$4+9*4=17$$

$$1*(9*8)=13$$

$$(0*2)*8=0$$

$$(3*0)*8=8$$

$$8*2-4=12$$

$$8)=-4$$

$$(9*6)-1=14$$

$$(7+6)*8=21$$

$$5+9=14$$

$$5-(2+8)=5$$

$$(1-0)-8=-3$$

$$9-(3-7)=13$$

$$5-5*1=0$$

$$1*8=-8$$

$$(6*6)*4=144$$

$$(3*0)*8=8$$

$$9*6-8=46$$

$$(1-9)-4=-12$$

数据预处理

数据认识

数据共100000张彩色图像

图像中的字符包括0-9,+,-,*,(,),=

数据可视化探索

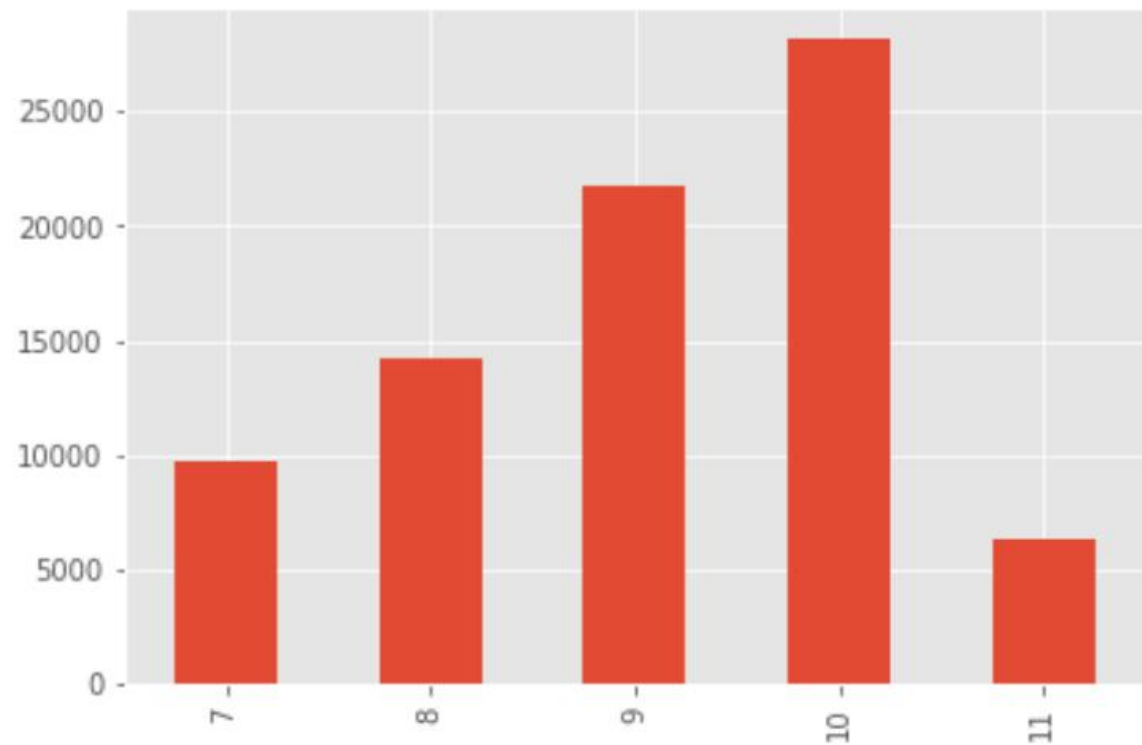
字符都是位于图像上下之间，所以不需要做文本检测工作

图像中有部分是空白的，字符不是沾满整个图像

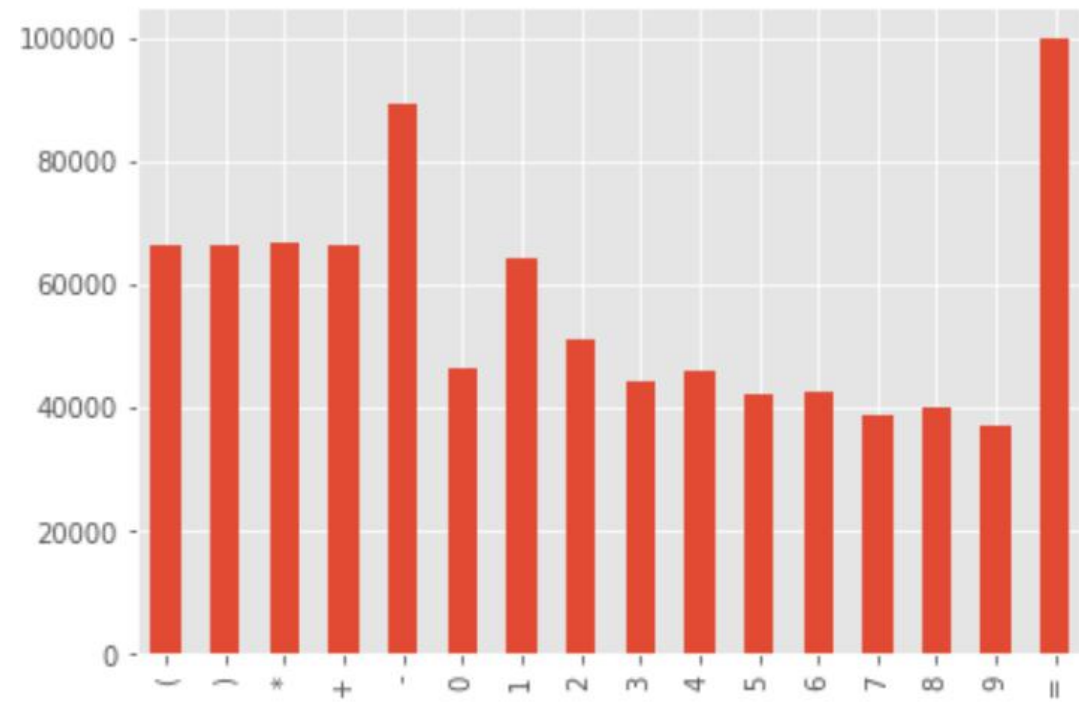
图像中有字符东倒西歪，不是严格的摆放

图像中序列长短不一

数据探索



数据序列字符长度分布



各个字符数量分布

数据预处理

数据认识

数据共100000张彩色图像

图像中的字符包括0-9,+,-,*,(,),=

数据可视化探索

字符都是位于图像上下之间，所以不需要做文本检测工作

图像中有部分是空白的，字符不是沾满整个图像

图像中有字符东倒西歪，不是严格的摆放

图像中序列长短不一

数据探索

数据增强

数据增强

自己生成数据

使用captcha 构造数据集

小角度旋转

不能大角度旋转，因为6、9以及会改变序列，
不能翻转，翻转改变了算式

对图像滤波处理

图像中有一些小点，可以用高斯滤波、中值
滤波去掉

二值化/灰度化

开闭运算

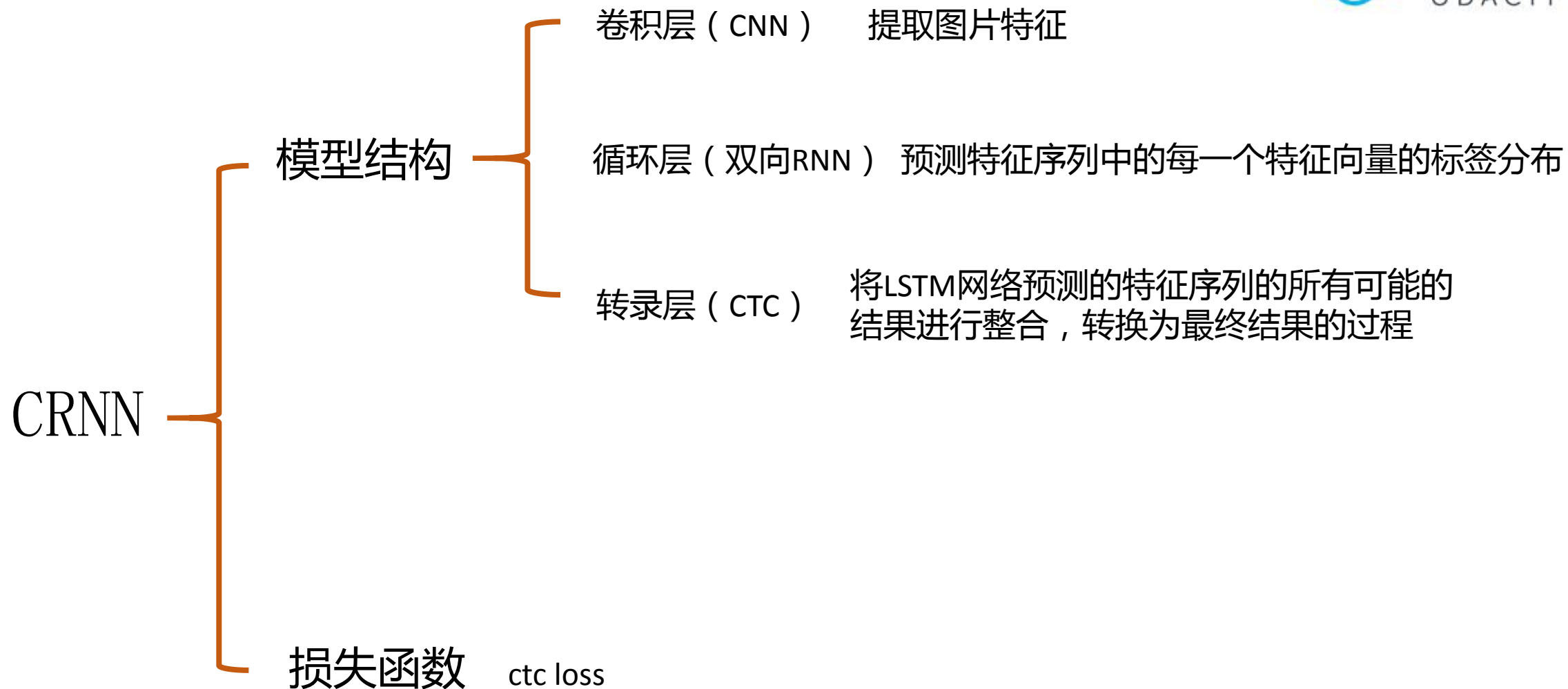
解决方法

CRNN

An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition

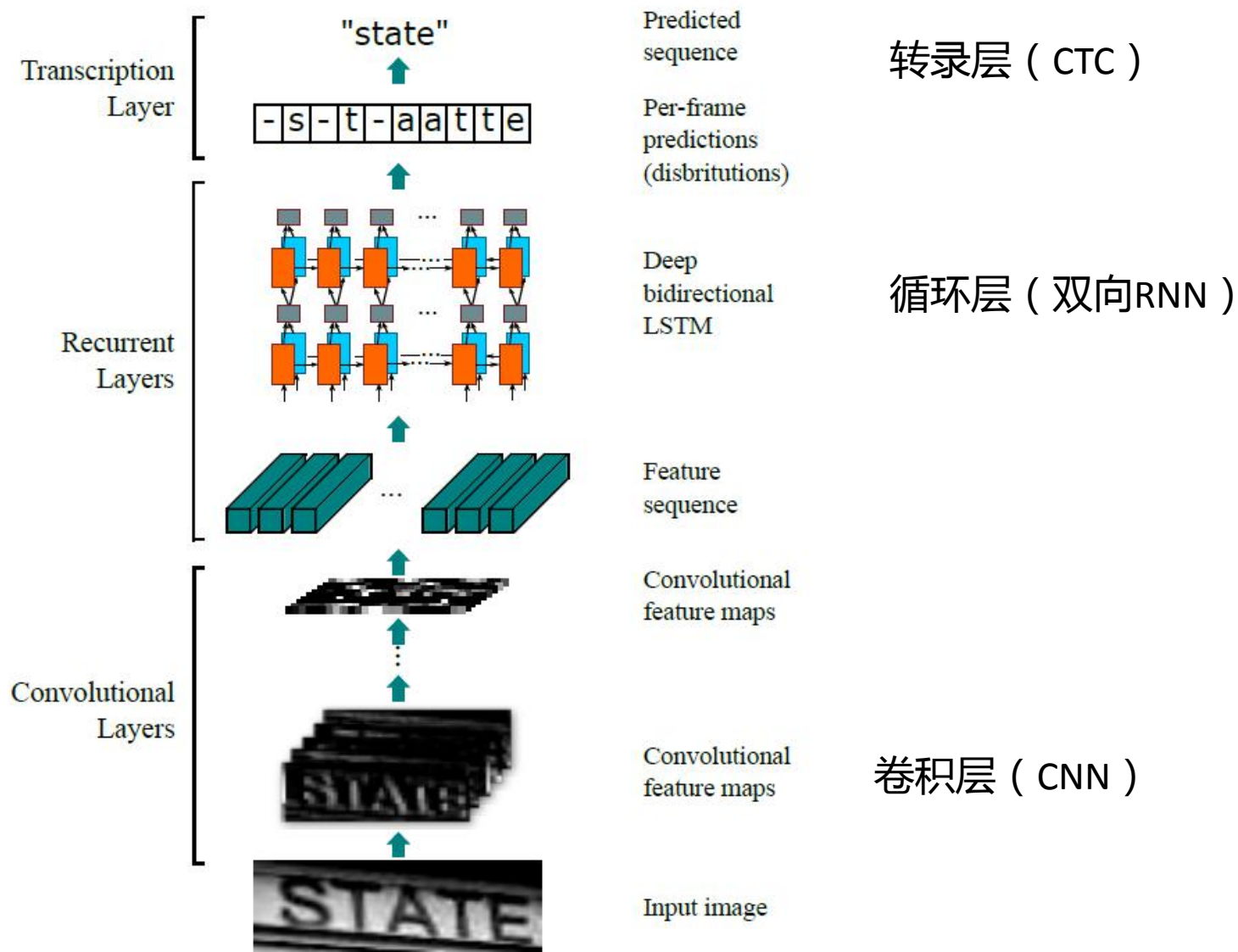
其他

算式识别 -- 解决方法

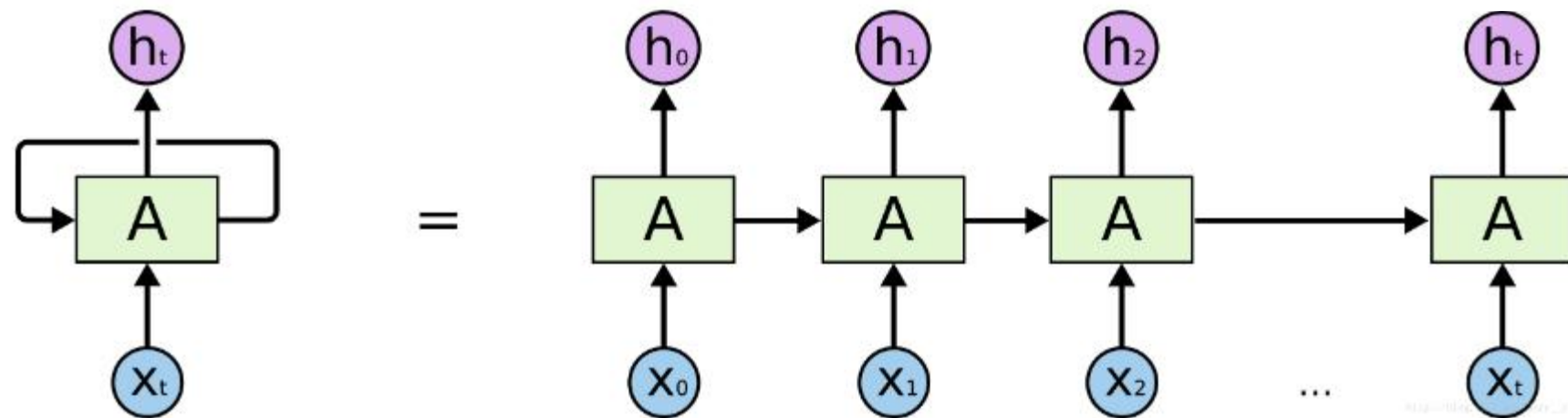


算式识别 -- 解决方法

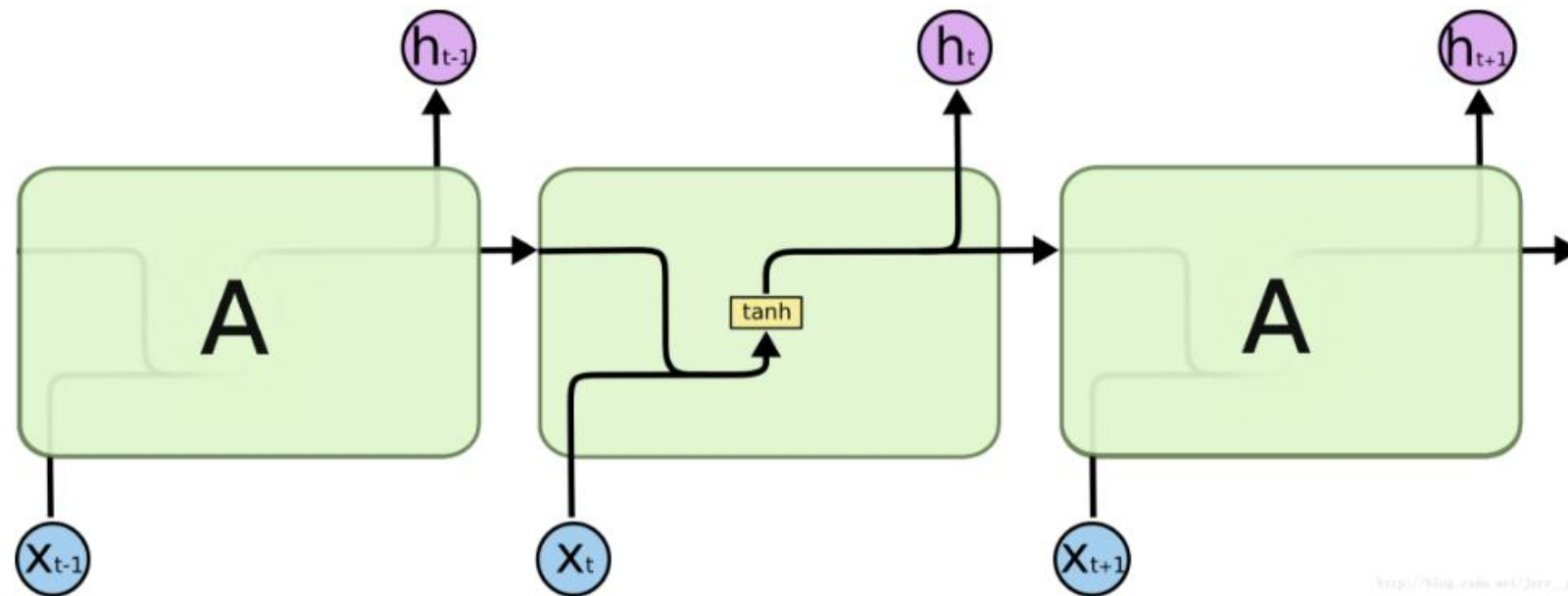
CRNN



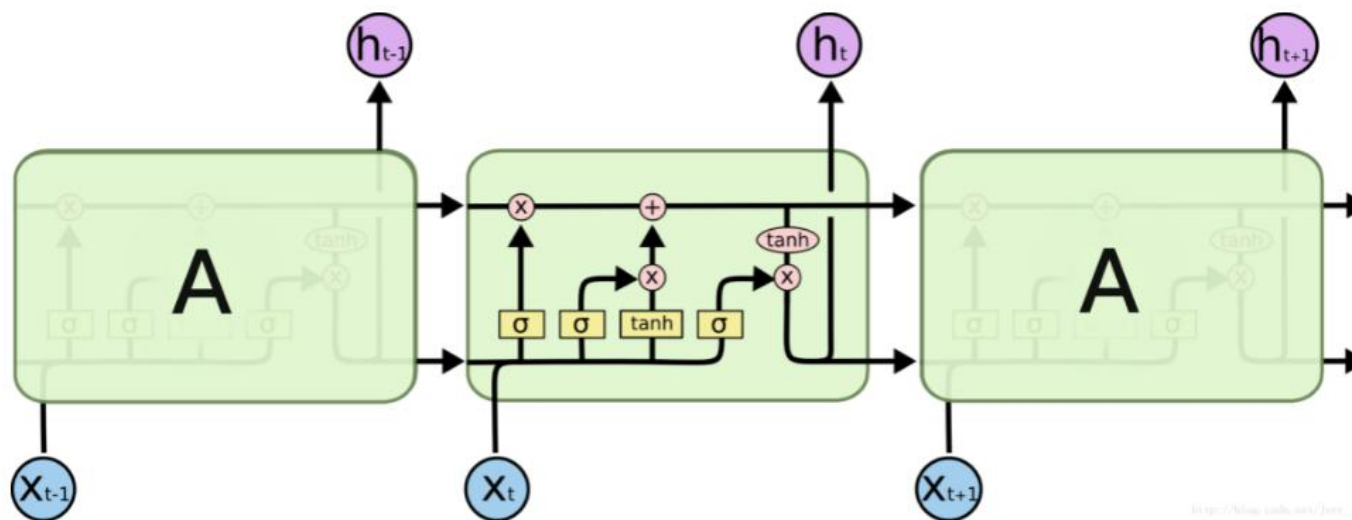
算式识别 -- 解决方法



RNN

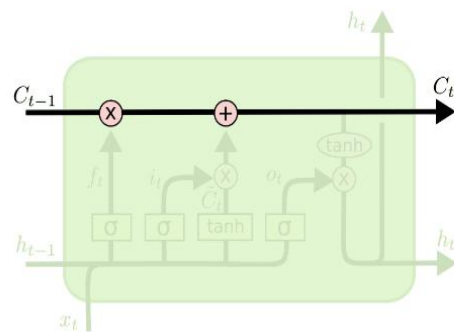


LSTM

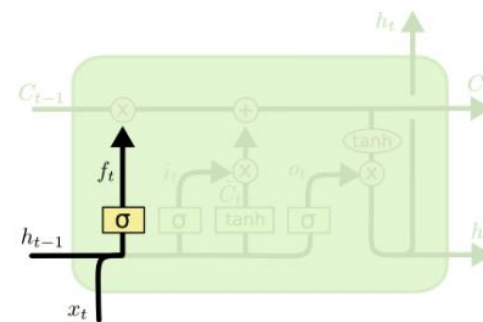


http://blog.csdn.net/jerry_xu

LSTM

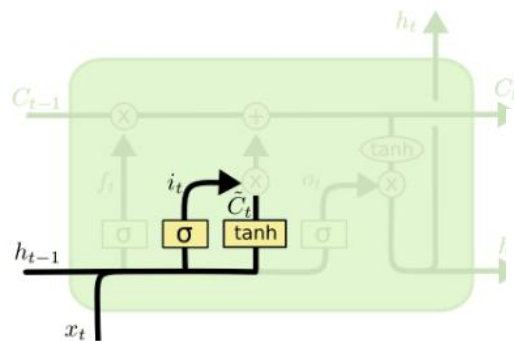


解决长期依赖，但是没法删除、增加信息



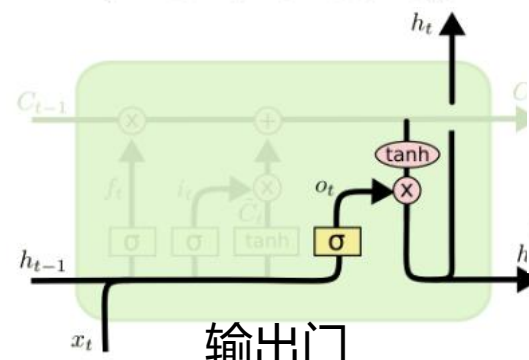
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

遗忘门



传入门

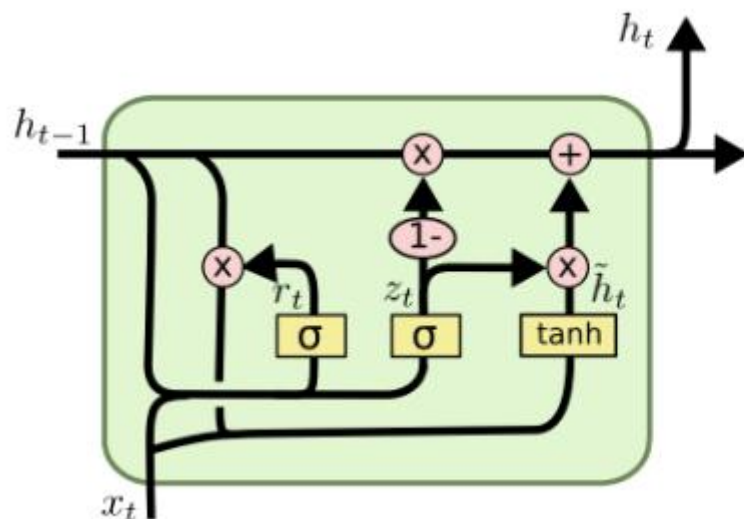
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$



输出门

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

GRU



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

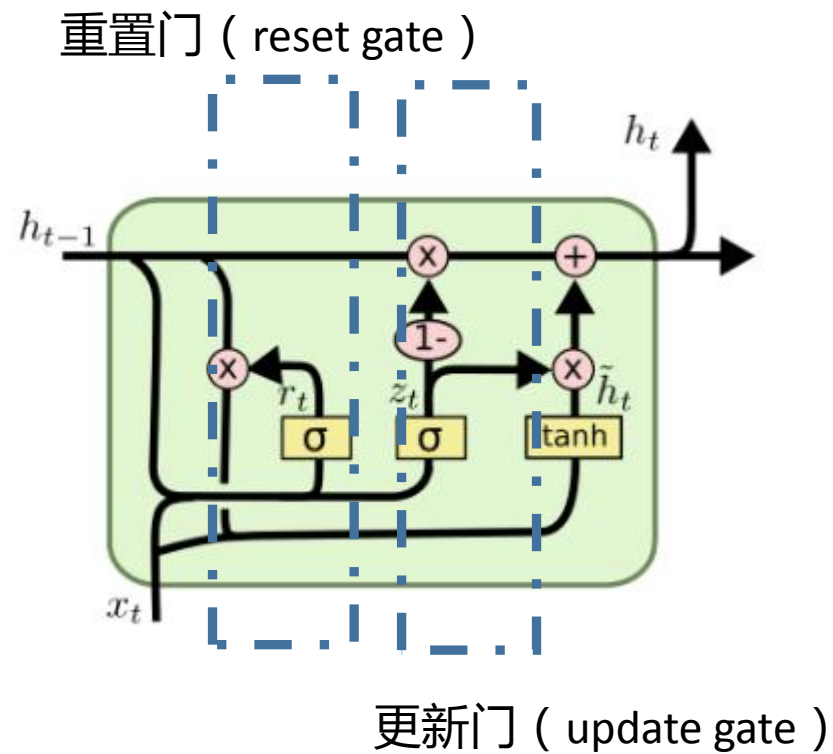
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

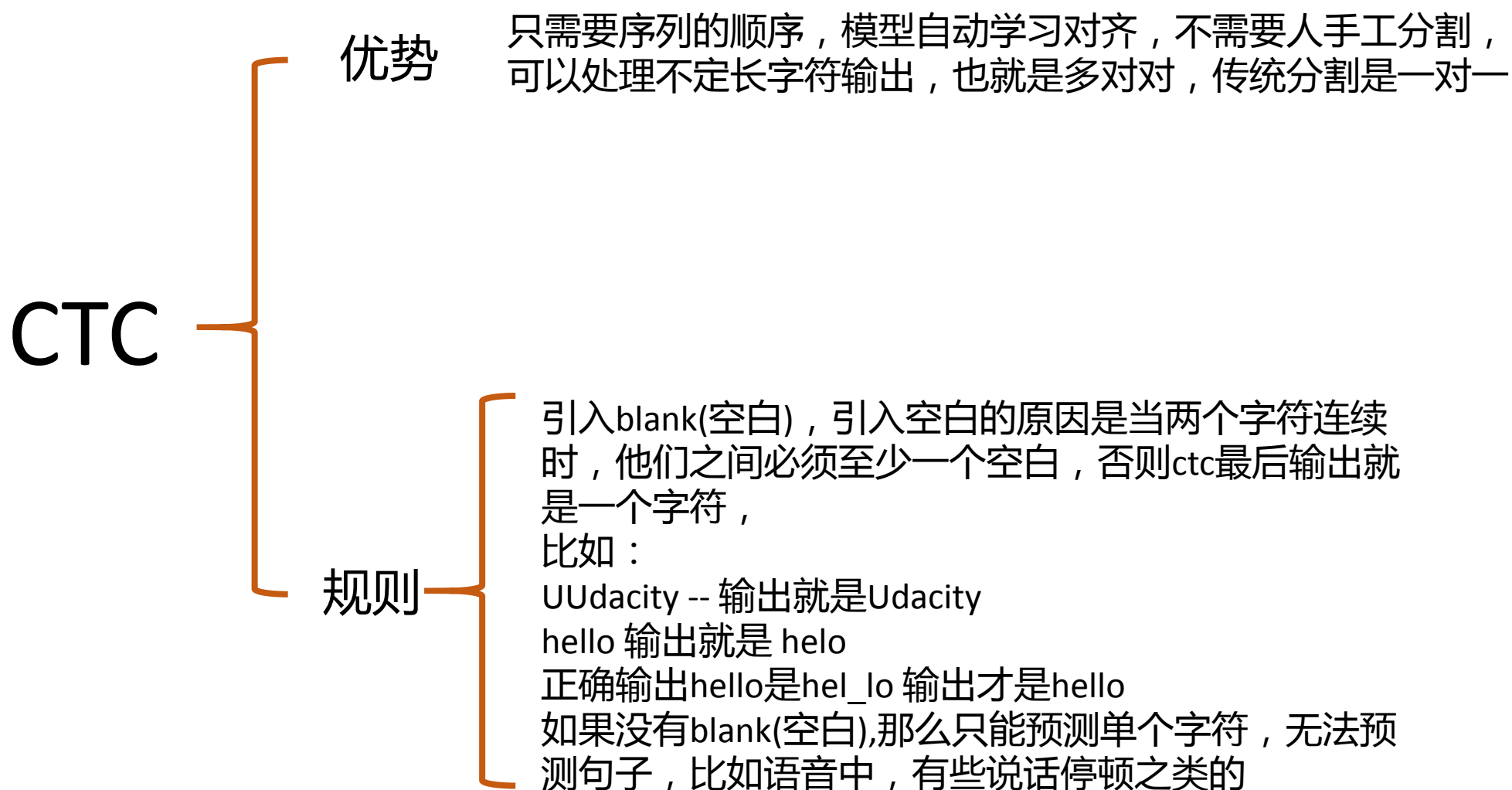
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

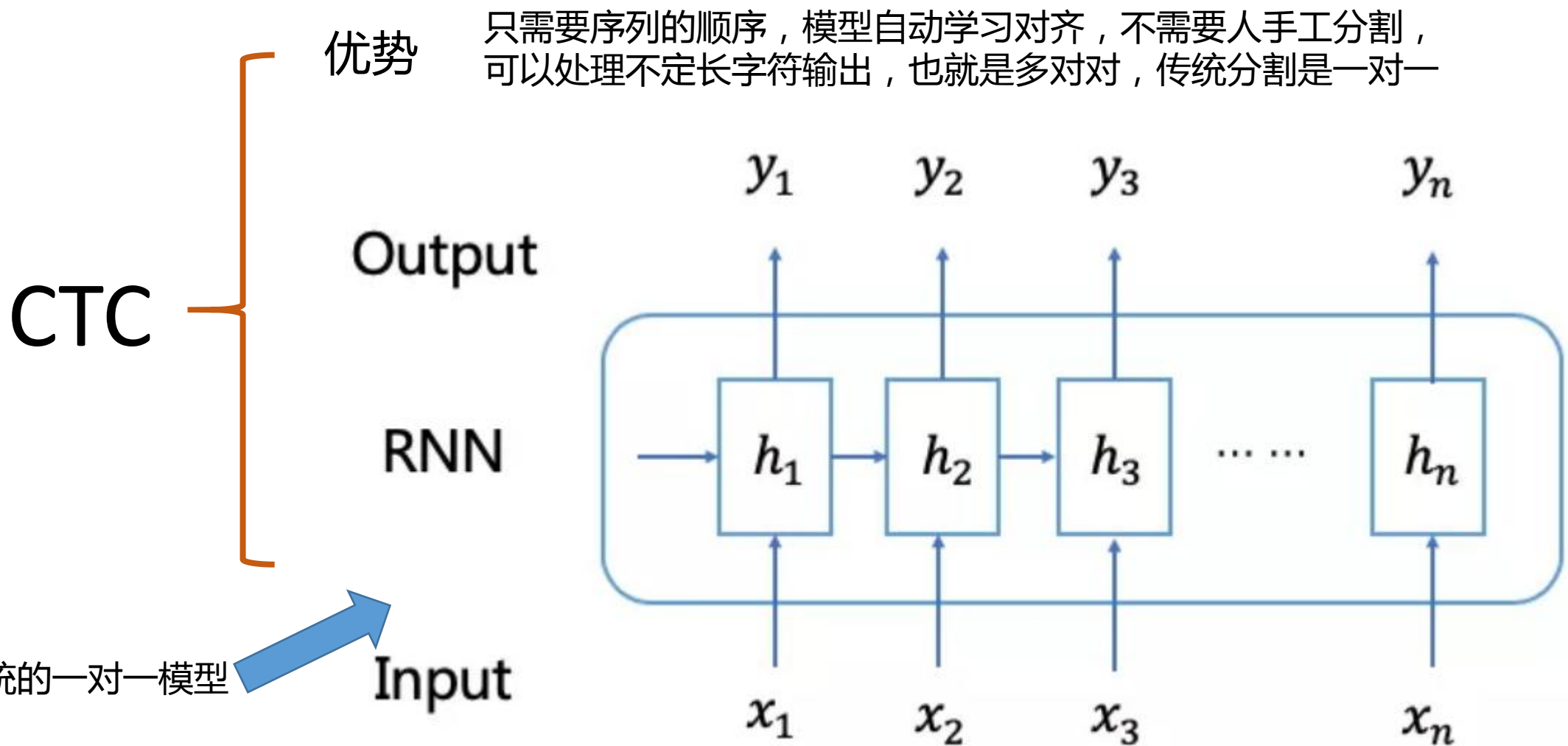
<http://blog.udacity.com/2016/03/gru.html>

LSTM变体重置门 (reset gate) 和更新门 (update gate)

GRU

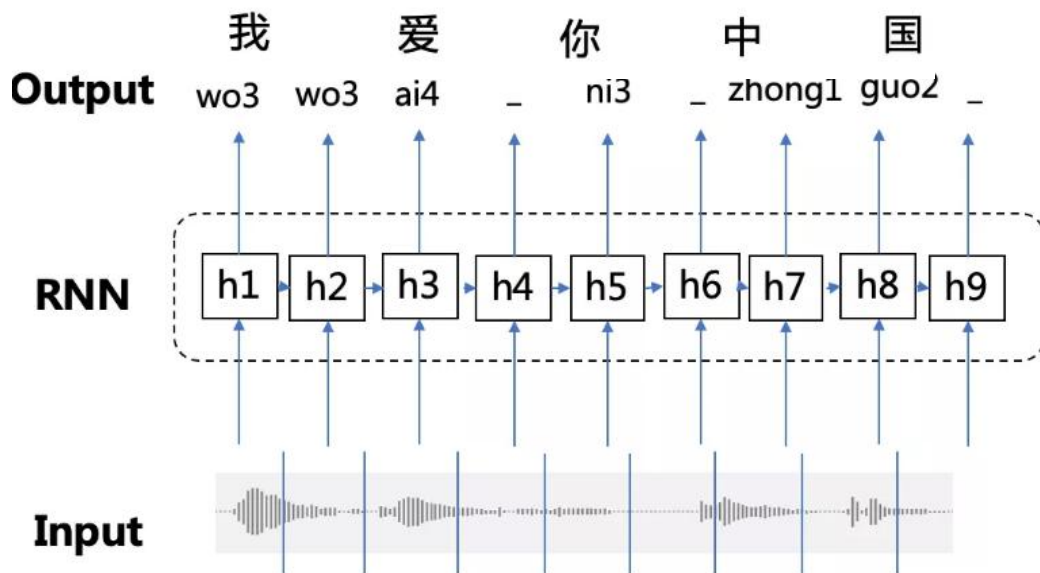




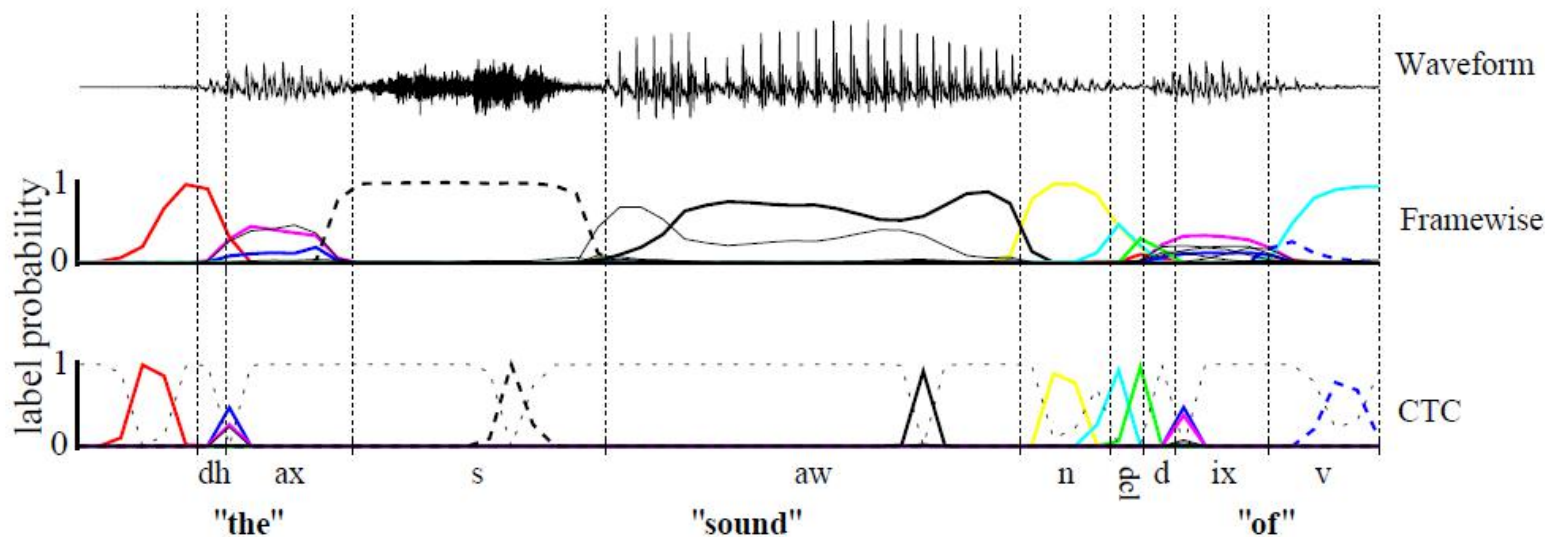


算式识别 -- 解决方法

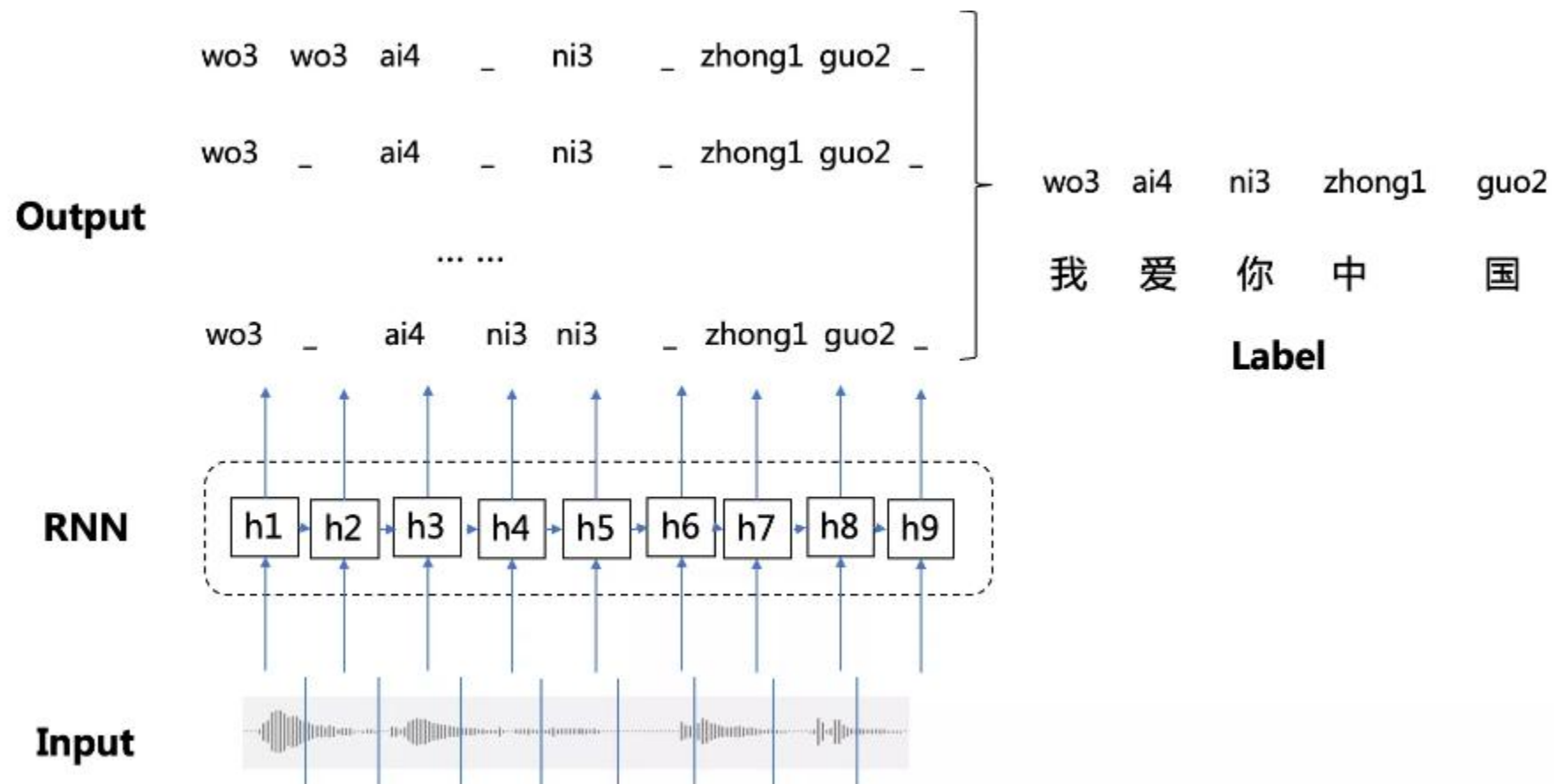
CTC



因为人说话之类的，中间是有停顿的，所以输出序列中存在重复序列，比如我我等，那么这个停顿我们用 blank(空白/间隔符 $-$)表示，那么我们最终输出的序列中需要把间隔符以及重复字符删除，所以输入到输出是1对多的关系，也就是输入一个序列，有多个输出结果得到同一个序列



CTC



CTC推导

定义

\mathbf{x} : 表示输入序列,
 $p(\pi|\mathbf{x})$: 输出路径 π 为的概率

$y_{\pi_t}^t$ 输出序列在第 t 步的输出为 π_t 的概率

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{x}).$$

代表给定输入输入 \mathbf{x} , 输出为序列 \mathbf{l} 的概率, 因此也就是所有输出序列为 \mathbf{l} 的路径输出概率的总和, \mathcal{B}^{-1} 表示输出序列为 \mathbf{l} 的所有路径。

理解这个公式: 输出序列 \mathbf{l} 的概率, 等于所有输出为该序列的路径的概率之和。

举个例子 如果输出序列长度为 Udacity 并且时间长度 T 为10, 那么合法的路径有:

__Udacity

__U_dacity

__U_d_acity

_U_d_a_city

_U_d_a_c_ity

----- 等等, 这些最终都是Udacity, 也就是有多条合法的路径

CTC推导

问题本质 $h(\mathbf{x}) = \arg \max_{\mathbf{l} \in L \leq T} p(\mathbf{l}|\mathbf{x})$.

目的就是使得输出为目标序列的概率最大，本质就是求最大似然的问题，输入 \mathbf{x} ，使得输出序列为 \mathbf{y} 的概率最大。因为输出序列很多种，我们要使得输出正确的序列的概率最大，而因为ctc是一对多，所以存在很多条有效的输出(路径)

求解问题方法

暴力搜索

暴力所以有效路径，时间复杂度为： T^N ，复杂度太大，行不通

动态规划

使用HMM中的Forward-Backward算法思路，利用动态规划算法求解

前向推导

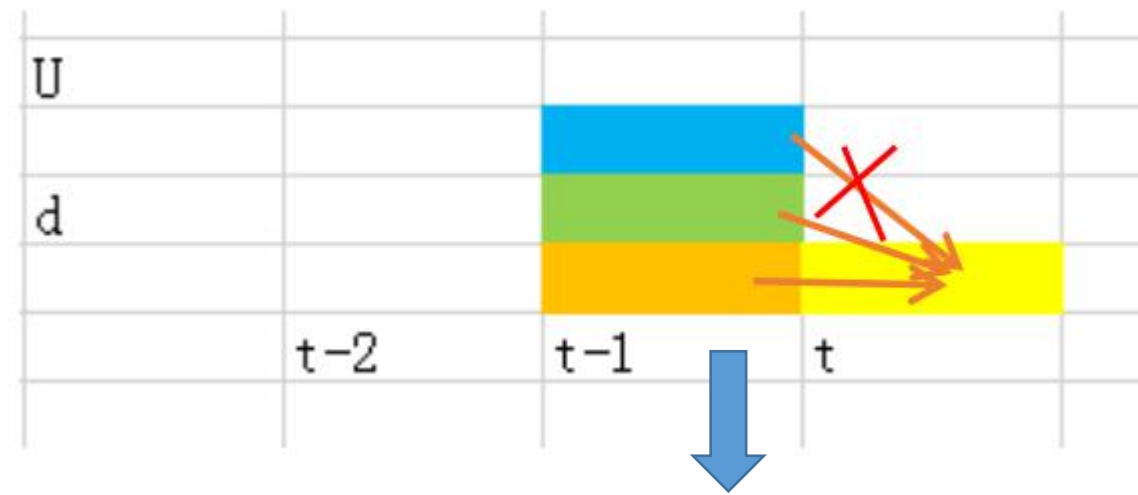
定义 $\alpha_t(s)$ 表示t时刻，经过节点s的全部的前缀子路径的概率总和，
比如我们第T时刻输出为a，那么他的前缀输出就是Ud，
而我們想在T时刻得到Uda

分情况

算式识别 -- 解决方法

分情况

当T时刻，s节点为空白



其中blank(_)到blank(_)条路径不行，因为直接由_(blank)到_(blank)，那么中间的d字符就没输出，最终输出__，就把d字符扔了，这条路径不合法。

而Ud_ 还是ud

Ud__ 还是Ud

而U__ 就是U 所以不行

得出

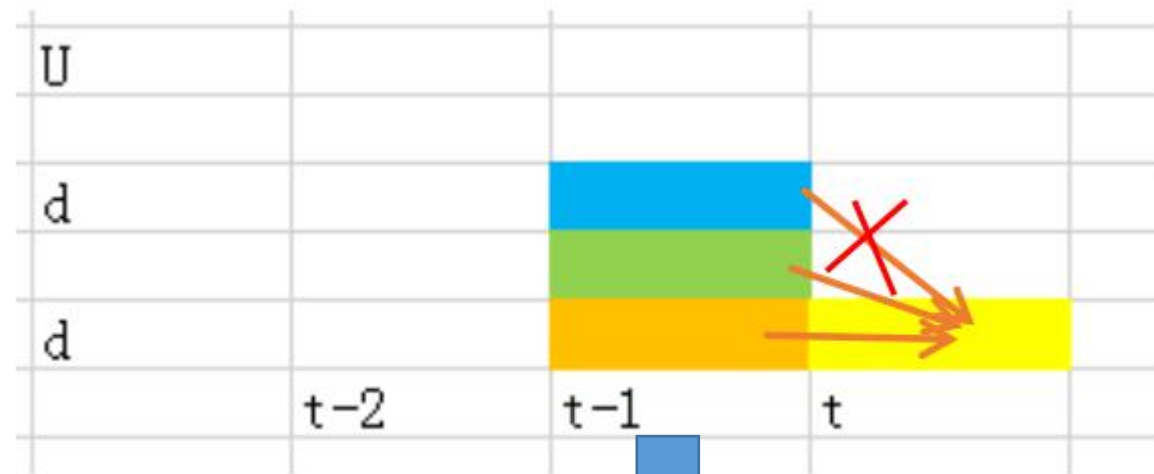
$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1))y_{l'_s}^t$$

算式识别 -- 解决方法

分情况

当T时刻，s节点为空白

当s-2字符等于s节点时



d到d这条路径不行，因为直接由d到d，那么中间没有blank，最终输出dd,合并后就只剩一个d了，因为我们说过相同字符之间必须要有一个blank，所以这条路径不合法

Udd -- 直接就是Ud，不合法

Ud_d 就是Udd 合法

Udd_ 就是 Udd 合法

备注，因为Udacity没有连续相同的字符，这个地方造的，想得到是Udd，而不是Ud，注意下，只是为了说明问题

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1))y_{l_s}^t$$

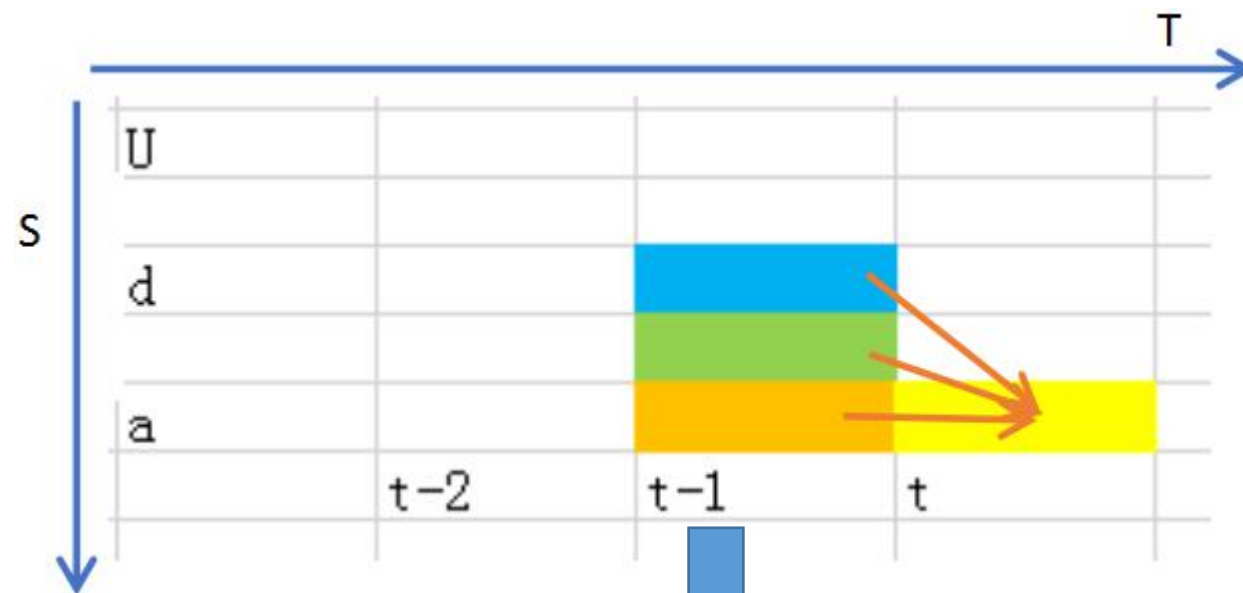
算式识别 -- 解决方法

分情况

当T时刻，s节点为空白

当s-2字符等于s节点时

其他



t时刻的s节点，不等于t-1时刻，不是blank的时候，共有三种路径到达

Uda 得到Uda合法

Ud_a 得到Uda合法

Uda_ 得到Uda合法

$$\alpha_t(s) = (\alpha_{t-1}(s) + \alpha_{t-1}(s-1) + \alpha_{t-1}(s-2))y_{l_s}^t$$

前向推导

定义 $\alpha_t(s)$ 表示t时刻，经过节点s的全部的前缀子路径的概率总和，
比如我们第T时刻输出为a，那么他的前缀输出就是Ud，
而我們想在T时刻得到Uda

分情况

This gives us the following rules for initialisation

$$\begin{aligned}\alpha_1(1) &= y_b^1 \\ \alpha_1(2) &= y_{l_1}^1 \\ \alpha_1(s) &= 0, \quad \forall s > 2\end{aligned}$$

and recursion

$$\alpha_t(s) = \begin{cases} \bar{\alpha}_t(s) y_{l'_s}^t & \text{if } l'_s = b \text{ or } l'_{s-2} = l'_s \\ (\bar{\alpha}_t(s) + \alpha_{t-1}(s-2)) y_{l'_s}^t & \text{otherwise} \end{cases} \quad (6)$$

where

$$\bar{\alpha}_t(s) \stackrel{\text{def}}{=} \alpha_{t-1}(s) + \alpha_{t-1}(s-1). \quad (7)$$

表示 t 时刻，经过节点 s 的全部的前缀子路径的概率总和，

定义 $\alpha_t(s)$ 比如我们第 T 时刻输出为 a ，那么他的前缀输出就是 Ud ，而我們想在 T 时刻得到 Uda

This gives us the following rules for initialisation

$$\alpha_1(1) = y_b^1$$

$$\alpha_1(2) = y_{l_1}^1$$

$$\alpha_1(s) = 0, \forall s > 2$$

and recursion

$$\alpha_t(s) = \begin{cases} \bar{\alpha}_t(s) y_{l'_s}^t & \text{if } l'_s = b \text{ or } l'_{s-2} = l'_s \\ (\bar{\alpha}_t(s) + \alpha_{t-1}(s-2)) y_{l'_s}^t & \text{otherwise} \end{cases} \quad (6)$$

where

$$\bar{\alpha}_t(s) \stackrel{\text{def}}{=} \alpha_{t-1}(s) + \alpha_{t-1}(s-1). \quad (7)$$

$$-\ln(p(l|x)) = -\ln(\alpha_T(|l'|) + \alpha_T(|l'|-1))$$

公式中右边有两项，因为最后输出两种情况，要么是最后一个blank，要么是倒数第二个非空白字符，这两种情况都可以

前向推导

分情况

前向ctc loss

后向同理推导

定义 $\beta_t(s)$

表示t时刻，经过节点s的全部的后缀子路径的概率总和，
后向就是从后到前，有y 推导 Udacity，过程是y-t-i-c-a-d-U的过程

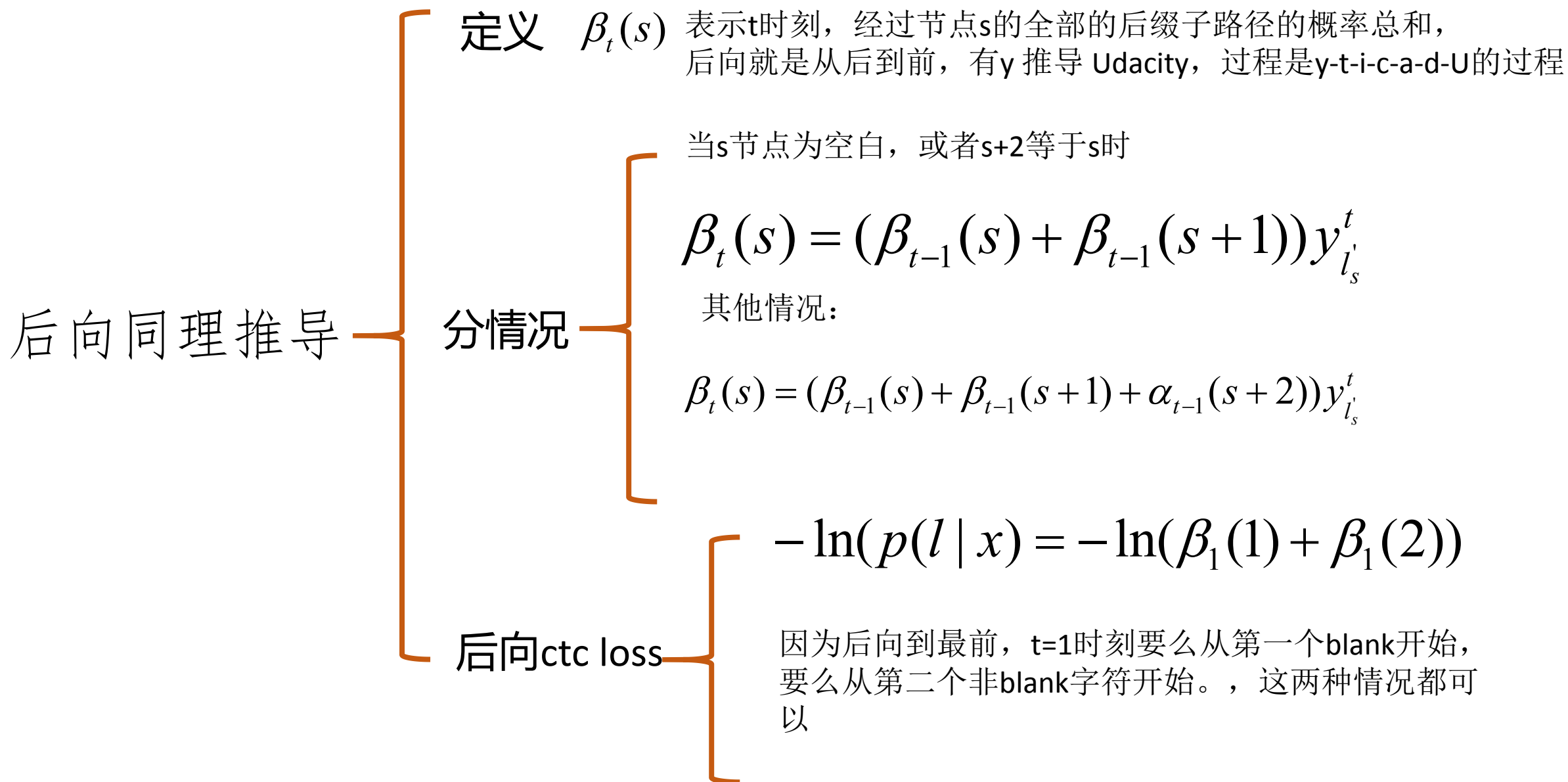
分情况

当s节点为空白，或者s+2等于s时

$$\beta_t(s) = (\beta_{t-1}(s) + \beta_{t-1}(s+1))y_{l_s}^t$$

其他情况：

$$\beta_t(s) = (\beta_{t-1}(s) + \beta_{t-1}(s+1) + \alpha_{t-1}(s+2))y_{l_s}^t$$



由前面推导，那么t时刻，经过 l'_s 节点的概率为：
$$\frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t}$$

解释：分子部分 $\alpha_t(s)\beta_t(s)$ ，不就是刚好前后向叠加起来就是一条完整的合法的路径嘛，为什么除以 $y_{l'_s}^t$ ，因为前后向都计算了这个概率，多了一个，所以要除以一个。

CTC推导

CTC推导

由前面推导，那么t时刻，经过 l'_s 节点的概率为：
$$\frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t}$$

得到：
$$\frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t} = \sum_{\substack{\pi \in B_{-1}(l) \\ \pi_t = l'_s}} p(\pi | x)$$
 这个是经过节点s的概率

得到：
$$p(l | x) = \sum_{s=1}^{|l'|} \frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t}$$
 就是从第一个节点遍历到最后
一个节点

两边取对数有：

$$-\ln(p(l | x)) = -\ln\left(\sum_{s=1}^{|l'|} \frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t}\right)$$

得到CTC LOSS的计算公式

CTC LOSS 推导

$$-\ln(p(l|x)) = -\ln\left(\sum_{s=1}^{|l|} \frac{\alpha_t(s)\beta_t(s)}{y_{l'_s}^t}\right)$$

得到CTC LOSS的计算公式

CTC推导

CTC LOSS 反向传播

$$-\frac{\partial \ln(p(l|x))}{\partial u_k^t} = y_k^t - \frac{1}{p(l|x)y_k^t} \sum_{s \in lab(l,k')} \alpha_t(s)\beta_t(s)$$

ctc loss的导数

CTC LOSS 预测方法

Greedy Search(贪心搜索)

Beam Search(束搜索)

CTC LOSS 预测方法

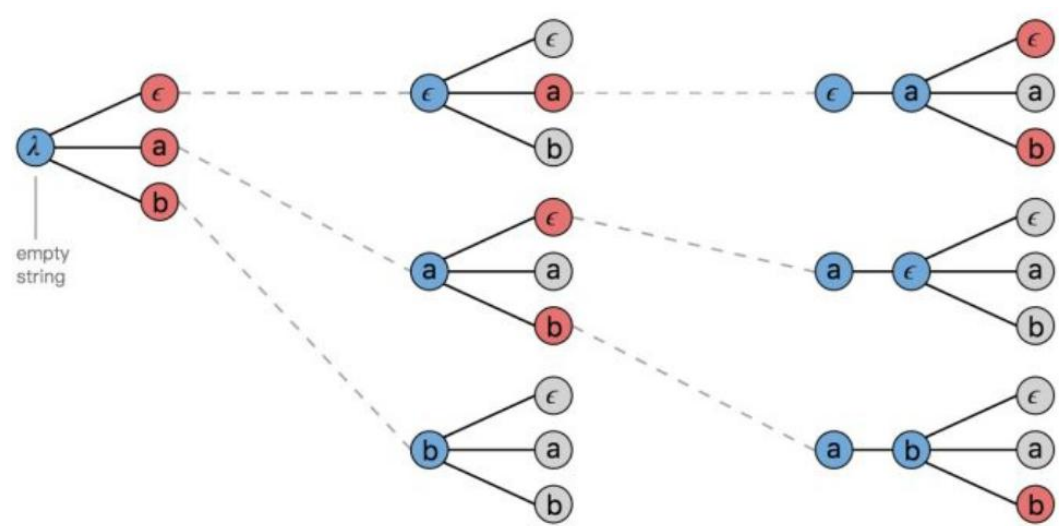
Greedy Search(贪心搜索)

对于每个时间片选择该时间片概率最高的节点作为输出，缺点是，忽略了一个输出可能对应多个对齐方式

Beam Search(束搜索)

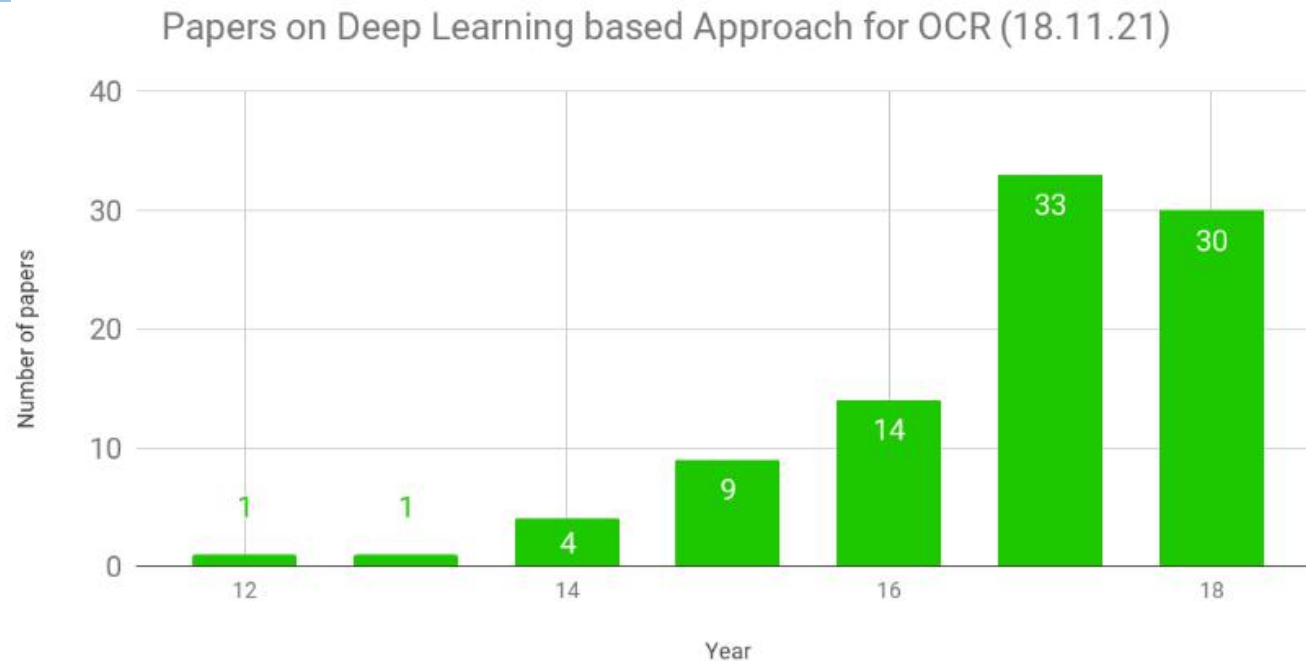
Beam Search是寻找全局最优值和Greedy Search在查找时间和模型精度的一个折中。选择当前时间片的概率最大的几个(人为设定，也就是beam size)组成一个组，或者一个序列，在将这个组与下一个时间片进行组合，从组合中继续选择概率最大组成一个新的组，直到最后一个时间片。

Beam Search(束搜索)

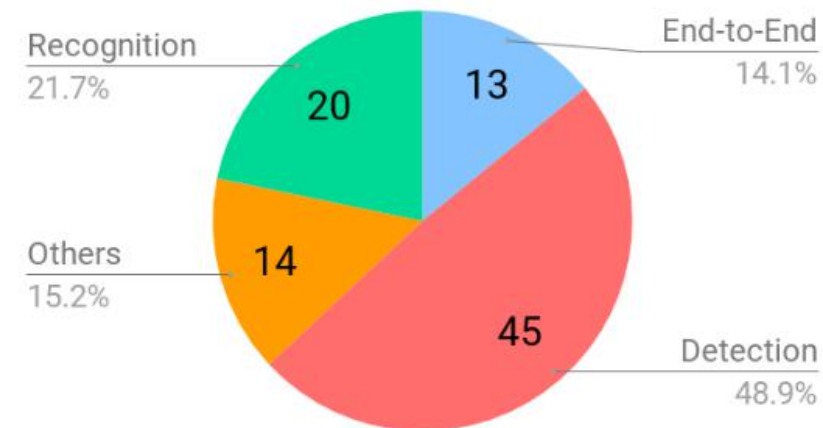


Beam size = 3





OCR技术文章的数量



OCR

文本检测

CTPN模型

Textboxes/Textboxes++模型

FTSN模型

WordSup模型

DMPNet模型

PixelLink模型

EAST模型

SegLink模型

字符识别

CRNN模型

RARE模型

ASTER

STR模型

端到端

FOTS

STN-OCR模型

Mask TextSpotter

文中部分内容与图参考链接：

<https://xiaodu.io/ctc-explained/>

<https://zhuanlan.zhihu.com/p/42719047>

https://github.com/ypwhs/baiduyun_deeplearning_competition

<https://zhuanlan.zhihu.com/p/34207613>

<https://blog.csdn.net/zhangjunbob/article/details/53258524>

参考文献：

Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 369-376.

Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(11): 2298-2304.

Canziani A, Paszke A, Culurciello E. An analysis of deep neural network models for practical applications[J]. arXiv preprint arXiv:1605.07678, 2016.

THANKS