人工智能系统测试日常笔记

学院：电子与信息工程学院

专业：计算机科学与技术

学号：2230771

姓名：包广垠

完成日期：2023 年 1 月 2 日

装

订

线

摘要

本报告为同济大学研究生课程《人工智能系统模型评估》的课程报告，该报告整理了我的每周笔记，将其整理归纳为四个板块，分别是：关于提升模型推理速度的研究、关于行人搜索的研究、关于人工智能模型的数据研究以及关于人工智能模型生命周期和评价标准的研究。

在关于提升模型推理速度的研究中，本文整理了视觉领域和自然语言处理领域的模型加速方法，主要为知识蒸馏和动态推理加速。

在关于行人搜索的研究中，本文首先从目标识别的经典方法 Faster RCNN 出发，过渡到行人搜索，调研了行人搜索领域的 SOTA 算法。

在关于人工智能模型的数据研究中，本文主要对数据的一致性和完整性展开研究，并且附上了一次相关讲座的笔记。

在关于人工智能模型生命周期和评价标准的研究中，本文首先阐述了 AI 系统的生命周期，然后以“基于深度学习的老照片修复系统”为例，探究其评价指标。

关键词：知识蒸馏，动态推理加速，行人搜索，AI 系统生命周期，评价标准，数据一致性

装

订

线

目 录

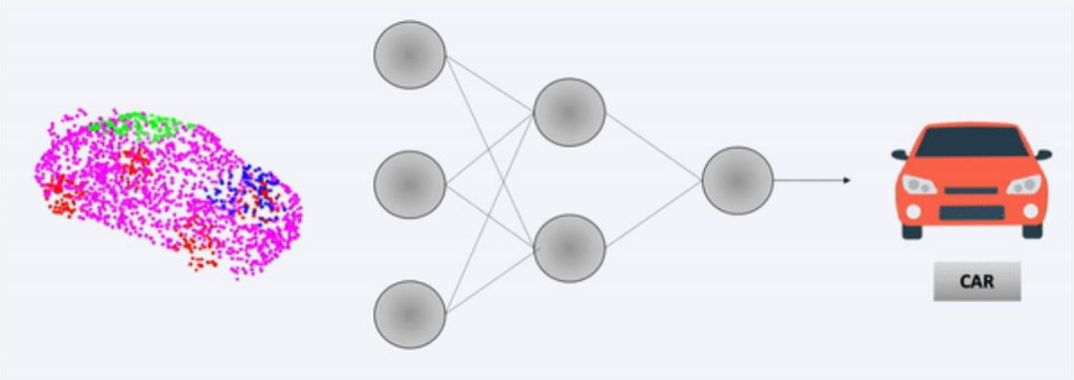
1 提升模型推理速度的研究

随着深度学习的发展，人工智能模型的体量越来越大，大规模预训练模型在提高模型精度方面有着很大的贡献。然而在实际的工业应用中，由于存储资源和计算资源的限制，大模型无法进行部署，即便进行部署也存在着模型推理速度缓慢的问题，因而需要进行模型推理速度提升的研究。

1.1 模型推理速度的度量标准

（1）模型推理的概念

在深度学习中，推理指的是神经网络的一次前向传播过程，也就是将输入数据送入神经网络，然后从中得到输出结果的过程。



例如，将三维点图输入神经网络，得到最终分类结果的过程视为一次推理。

（2）推理速度的度量标准

为了理解神经网络的推理速度，我们必须有一个指标，通常该指标为推理时间，推理时间指的是神经网络执行一次前向传播所需要的时间。通常我们用一秒钟内模型能够执行的推理次数来表示模型的推理速度，单位用 fps 表示。假如模型推理一次需要的时间为 0.1 秒，那么其推理速度可以表示为 1/0.1=10fps。

（3）用于衡量推理速度的指标

- FLOPs：FLOPs 的全称为 Floating Point Operations，即浮点运算操作，这些运算操作包括加、减、乘、除和其他任何跟浮点数相关的操作。一个模型的 FLOPs 为该模型全部的

FLOP 之和，这个数值可以告诉我们模型的复杂程度。

- FLOPS：FLOPS 全称是 Floating Point Operations per Second 的缩写，表示每秒钟内可以执行的浮点运算数。该指标与我们使用的计算平台的性能有关。但在相同的平台下，其值越大，表明在该计算平台上每秒钟内可执行的浮点操作数越多，那么模型的推理速度就越快。
- MACs：MACs 全称是 Multiply-Accumulate Computations，即加法-乘法计算。一次 MAC 操作表示执行了一次加法和一次乘法操作。
- Times：完成一次推理平均所用时间。

一般而言，使用 FLOPs、FLOPS 和单次推理时间作为模型推理速度的评价标准。

（4）模型推理加速的目标

- 降低模型推理所需要的浮点数计算量，即降低模型的 FLOPs；
- 提高模型在计算平台上的计算速度，即提高模型的 FLOPS。

1.2 加速模型推理速度的两个方向

（1）模型压缩

压缩模型的大小，减小模型的参数量，则在模型推理时，模型所需要的推理时间就会相应减少，从而达到推理加速的目的。

（2）推理加速

通常是改变模型的结构，使得模型的计算速度更快，或者是设计提前结束推理的算法。

1.3 自然语言处理领域的模型压缩

自然语言处理领域的模型压缩方法包括以下几种，我将从思想和策略两个方面来展开说明。

（1）量化

- **思想：**减小模型中数据所占的 bit 位数。
- **说明：**量化是模型压缩的一种通用方法。一个模型包含了模型构架和模型参数两个部

分，而量化是对模型的参数本身进行压缩。即选用精度不高的浮点数类型替代模型中精度较高的数据，从而达到减小模型所占存储空间的目的，实现模型的压缩。对模型参数进行量化，可以达到减低模型存储容量和降低模型推理时运行内存的开销的效果。同时，由于计算硬件可以向低精度数值计算进行定向优化，量化还可以在一定程度上提高模型的推理速度。

- **策略：**
  1) 直接量化：将所有参数的截断到目标比特位数；
  2) 针对非奇异值进行量化；
  3) 量化感知训练。

（2）剪枝

- **思想：**剪枝是模型压缩的另一种方法。剪枝即为剪掉模型中冗余的或者不重要的部分，剪枝的对象可以是模型中的参数，也可以是模型的部分结构。
- **非结构剪枝：**

1) 定义：非结构剪枝剪掉模型中不重要的参数，被剪掉的参数一般具有绝对值较小或者梯度较小的特点，也可人为设定评判标准挑选低重要性的参数。
2) 方法：基于幅度的剪枝、基于动量的剪枝。
3) 缺点：由于非结构剪枝独立地考虑并修剪模型中的参数，导致剪枝后的模型结构变得不确定和不规则，虽然减小了模型所占的存储空间，但存在着增加运行时模型所占内存空间和减慢模型推理速度的问题。

- **结构剪枝：**

1) 定义：结构剪枝将模型的结构部件作为剪枝的对象，它剪掉模型结构部件的一部分甚至整个结构部件。
2) 方法：注意力头的剪枝、编码单元数目的剪枝和 embedding 的剪枝。具体方法为：在模型训练阶段，随机失活注意力头、编码单元和 embedding 层神经元；在部署阶段从模型，从 BERT 模型中选择部分注意力头、编码单元和 embedding 层神经元进行部署，从而达到模型压缩的目的。

（3）知识蒸馏

知识蒸馏是指利用一个或多个大模型（教师模型）的输出去训练小模型（学生模型）。通过将教师模型的知识迁移到学生模型之上，从而达到在保证精度的同时减小模型所占空间和提高

模型的推理速度的效果。

（4）矩阵分解

将神经网络中线性层的大型矩阵运算分解两个低秩矩阵运算，可以压缩模型的大小和提高模型的计算推理速度。注意力层的矩阵分解：将 key-query 矩阵投影到较低维度，或计算 top-k 的 key-query 乘积值的 softmax 来减少注意力计算中所需的计算量。

（5）动态推理加速

- **定义：**动态推理加速根据输入样本的不同动态的调整推理的计算量，从而达到减少平均计算开销的效果。
- **方法：**提前退出通道、渐进式词向量消减。

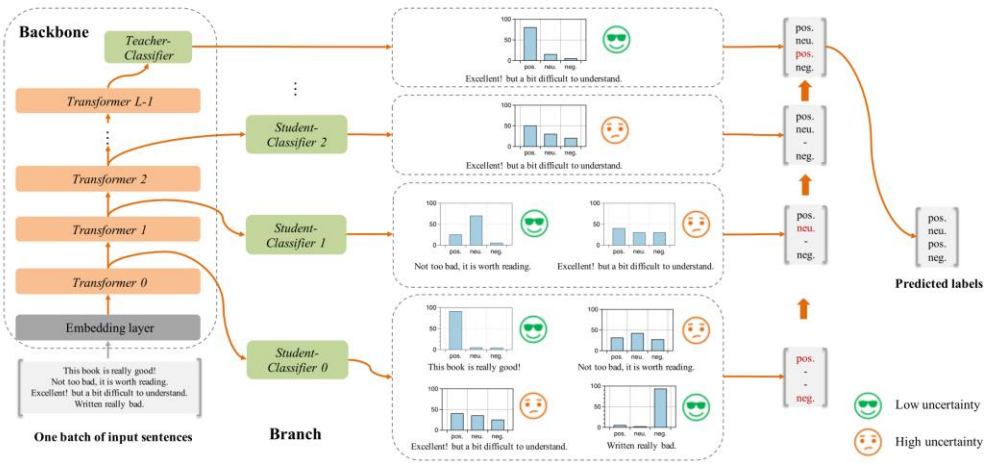
（6）其他方法

模型参数共享、embedding 层语料库压缩、权重挤缩。

1.4 自然语言处理领域的动态推理加速

（1）FastBERT

提出了自适应推理的方法，以及自蒸馏的训练方法：



自适应推理：Transformer（BERT）模型的每一层编码器都会输出目前提取出的语句特征，因此，在每层后都加上一个简单的分类器（两个线性层和一个 softmax 层），每个分类器都会输出

一个对于分类预测的概率分布，根据次概率分布的熵不确定性判断是否使用这个分类结果以及是否停止后续推理（存在一个阈值 speed，某种程度上反映了对推理速度的提升）。熵不确定性的计算如下：

$$Uncertainty = \frac{\sum_{i=1}^{N} p_s(i) \log p_s(i)}{\log \frac{1}{N}}$$

自蒸馏训练：此模型的训练分为三个阶段。

1) 预训练：backbone 的预训练，即 BERT 模型的训练；
2) 微调：在下游任务上微调整个 backbone 以及最高层的教师分类器；
3) 自蒸馏：固定 backbone 和教师分类器的参数，利用教师分类器输出的概率分布作为软目标进行知识蒸馏，采用每层的分类器输出的概率分布与教师分类输出的概率分布的 KL 散度作为损失函数。

$$Loss(p_{s_0}, ..., p_{s_{L-2}}, p_t) = \sum_{i=0}^{L-2} D_{KL}(p_{s_i}, p_t)$$

（2）DeeBERT

和（1）一样的方法，将该方法取名为提前退出通道。

（3）Early Exiting BERT

结构和（1）与（2）相似，但 backbone 是 MonoBERT，并且在训练时，将所有分类器一起微调：

$$\min_{\theta} \sum_{(x,y)\in\mathcal{D}} \sum_{i} L_i(x, y; \theta)$$

（4）BERT Loses Patience

基于耐心值的退出策略：在具有提前退出通道的 BERT 模型上提出了一种新的提前退出机制。前述三种的提前退出机制均是基于中间分类器的熵不确定性，本文提出的是基于耐心值的退出机制。当连续两个中间分类器等到相同的分类结果时，耐心值 cnt 就相应增加：

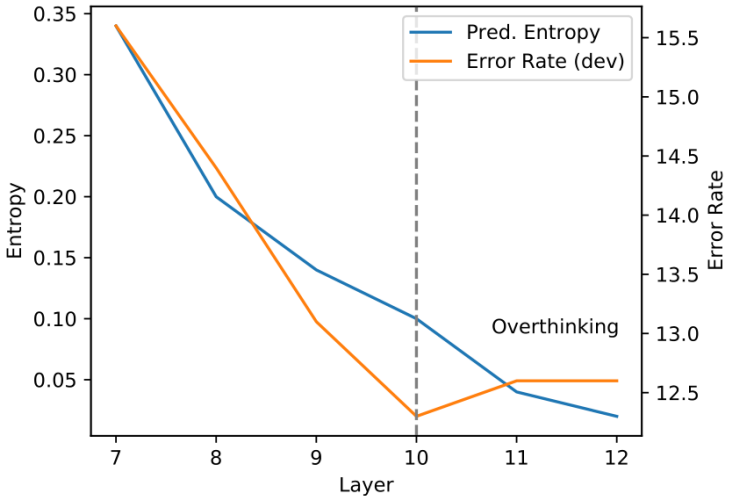
$$cnt_i = \begin{cases} cnt_{i-1} + 1 & \arg\max(\mathbf{y}_i) = \arg\max(\mathbf{y}_{i-1}), \\ 0 & \arg\max(\mathbf{y}_i) \neq \arg\max(\mathbf{y}_{i-1}) \vee i = 0. \end{cases}$$

对于回归任务，要求输出值的差值小于给定阈值：

$$cnt_i = \begin{cases} cnt_{i-1} + 1 & |y_i - y_{i-1}| < \tau, \\ 0 & |y_i - y_{i-1}| \geq \tau \vee i = 0. \end{cases}$$

在模型推理过程中，当耐心值大于预设值时，模型结束推理，采用当前中间分类器得到的结果；若未出现耐心值大于预设值的情况，则采用最终层的输出作为结果。

采用基于耐心值的提前退出机制，可以减少推理过程中的“过度思考”的情况出现：



此外，本文还通过公式计算耐心值的阈值，只需耐心值的阈值满足：

$$n - t < (\frac{1}{2q})^t (\frac{p}{q}) - p$$

t 为耐心值的阈值，n 是中间分类器的个数，p 是最终层错误率，q 是中间层平均错误率，则该方法就可以提高模型的推理速度。

逐层递进的损失权重：层数 l 作为第 l 层中间分类器损失的权重。

（5）LeeBERT

多级损失：使用多个提前退出通道的概率分布构造损失，有两种策略：

1) 从之后的通道学习（LLE）

2) 从全部的通道学习（LAE）

损失采用与硬目标的交叉熵和软目标的 KL 散度构成。

$$\mathcal{L}(x_n, y_n) = \sum_{m=1}^{M} w_i \mathcal{L}_{CE}(\mathbf{p}_m(x_n), y_n) + \sum_{m=1}^{M} \sum_{t \in \mathcal{T}(m)} w_{m,t} \frac{\mathcal{L}_{KD}(\mathbf{p}_t(x_n), \mathbf{p}_m(x_n))}{M * |\mathcal{T}(m)|}$$

二阶段优化：

- **思想：**在两个集合上训练模型参数和可学习的模型超参数

$$min_{\Omega}\mathcal{L}_{D_2}(\Theta^*(\Omega),\Omega),$$
$$s.t.,\Theta^*(\Omega)=\arg\min_{\Theta}\mathcal{L}_{D_1}(\Theta,\Omega)$$

- **两个数据集：**实际应用中是将数据集划分为两个部分，当作两个数据集
- **训练过程：**两阶段优化问题的数值解法

Algorithm 1: LeeBERT-CLO-v1

Parameters: Θ, Ω;
Return: the converged early exiting model; **while** *not converge* **do**
  for *t=1, ..., T* **do**
    sample batch B_1 and B_2 from D_1 and D_2, respectively
    update Θ with

$$\Theta = \Theta - \lambda_1 \nabla_{\Theta} L_{B_1},$$

    calculate L_{B_1} and L_{B_2} with the updated Θ, and update Ω with:

$$\Omega = \Omega - \lambda_1 \nabla_{\Omega} L_{B_1} - \lambda_2 \nabla_{\Omega} L_{B_2},$$

  end
end

或使用更为有效的两阶段优化方法：

Algorithm 2: LeeBERT-CLO-v2

Parameters: Θ, Ω, C;
Return: the converged early exiting model; **while** *not converge* **do**
  for *t=1, ..., T* **do**
    for *c = 1, 2, ..., C* **do**
      if *c != C* **then**
        sample batch B_1 from D_1, respectively update Θ and with

$$\Theta = \Theta - \lambda_1 \nabla_{\Theta} L_{B_1},$$
$$\Omega = \Omega - \lambda_1 \nabla_{\Omega} L_{B_1},$$

      end
      else
        sample batch B_1 and B_2 from D_1 and D_2, respectively
        update Θ with

$$\Theta = \Theta - \lambda_1 \nabla_{\Theta} L_{B_1},$$

        calculate L_{B_1} and L_{B_2} with the updated Θ, and update Ω with:

$$\Omega = \Omega - \lambda_1 \nabla_{\Omega} L_{B_1} - \lambda_2 \nabla_{\Omega} L_{B_2},$$

      end
    end
  end
end

（6）Early-Exit BERT

- 概述：采用“暂停-复制”的方法逐步减少词向量，从而减少推理的开销。
- 局部最大不确定度：用局部最大不确定度来判断一个词向量是否暂停向后传。

$$u_n'^{(l)} = \max\{u_{n-k}^{(l)}, \cdots, u_{n+k}^{(l)}\}$$

- 暂停-复制：不向后传的词向量，后续该位置都复制暂停时的词向量，用于其他词向量暂停与否的判断。

- 模型训练：
  1) 预训练；
  2) 所有分类器（最终层和中间层）及 backbone 一起微调；
  3) 对每个输入的样本 x，不使用暂停-复制，放入模型推理一遍得到每个词的退出层数，利用这个退出层数再对模型进行一次微调，从而缩小训练与推理的不同。

1.5 视觉领域的知识蒸馏

（1）基于单个教师的知识蒸馏

通过使用来自教师的逻辑或特征信息，可以实现从大型教师网络向小型学生网络的知识转移，具体方法包含以下几个方面：

- **从逻辑分布中获得知识**：该方法与传统的知识蒸馏相似，即通过蒸馏温度来软化输出的逻辑分布，让学生模型学习软化后的概率分布，从而获得知识。在该过程中，通常需要添加正则化的方法来减小模型的过拟合或者欠拟合。此外，面对噪声数据和分布不均匀的数据也需要进行特殊的处理，即有以下四类：

Method	Sub-category	Description
KD from logits	Softened labels and regularization	Distillation using soft labels and add regularizatio to avoid under-/over-fitting
	Learning from noisy labels	Adding noise or using noisy data
	Imposing strictness	Adding optimization methods to teacher or student
	Ensemble of distribution	Estimating model or data uncertainty

不同类别使用的优化目标不同。

对于软标签和正则化问题，使用的损失函数如下：

$$\mathcal{L}(x, W_{l-1}) = \alpha * H(y, \sigma(z_s^{l-1}; \rho = 1) + \beta * H(\sigma(z_t^l; \rho = \tau), \sigma(z_t^{l-1}, \rho = \tau))$$

对于噪声标签的学习，使用的损失函数如下：

$$\mathcal{L}(x + \delta, W) = \alpha * H(y, \sigma(z_s; \rho = 1) + \beta * H(\sigma(z_t; \rho = \tau), \sigma(z_s, \rho = \tau))$$

对于增强教师模型和学生模型的一致性问题，使用的损失函数如下：

$$\mathcal{L}(x, W^T) = \alpha * H(y, \sigma(z_t; \rho = 1)+$$
$$\beta * [f_{a_1}^T - \frac{1}{K-1}\sum_{k=2}^{K} f_{a_k}^T]$$

对于数据分布问题，使用的损失函数如下：

$$\mathcal{L}(x, W) = \frac{1}{K}\sum_{k=1}^{K}[\alpha * H(y, \sigma(p_s^k; \rho = 1)+$$
$$\beta * H(\sigma(p_t^k; \rho = \tau), \sigma(p_s^k, \rho = \tau))]$$

- **从中间层获得知识：**该方法是基于模型的中间层特征进行知识蒸馏，基于特征的蒸馏能够从教师那里学习更丰富的信息，并为提高绩效提供更大的灵活性。该方法是基于提示学习的思想，基于提示的中间层知识蒸馏通常可以写为如下形式：

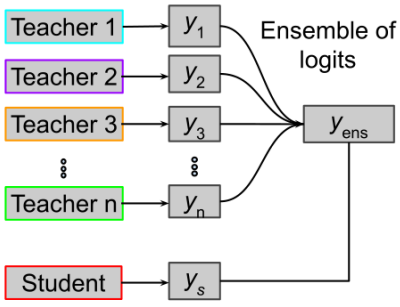
$$\mathcal{L}(F_T, F_S) = D(TF_t(F_T), TF_s(F_S))$$

即将学生模型和教师模型的特征图的某种表示对齐。

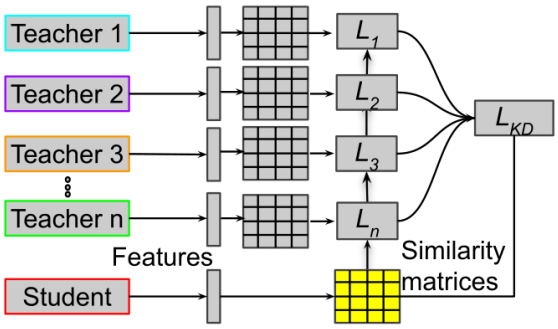
（2）基于多个教师的知识蒸馏

多个教师的知识蒸馏的出发点在于：学生可以从多个老师那里学到更好的知识，这些老师比一个老师更能提供信息和指导。关键问题包括两方面：一是多教师的选择，另一是多教师知识的集成策略。

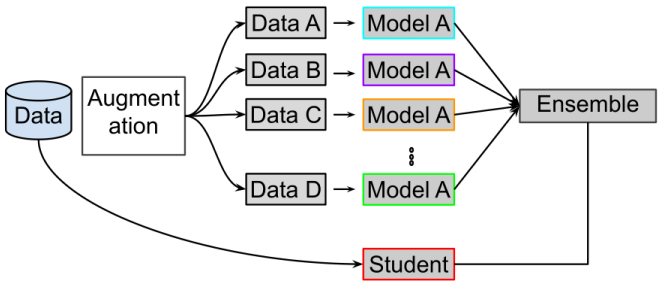
- **逻辑分布的集成：**逻辑分布的集成如下图所示。该方法将多个教师输出的概率分布软化后分别与学生模型计算损失，然后将学生模型与多个教师模型的损失相加并取平均值，从而得到最终的损失。



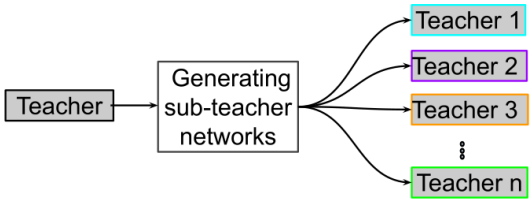
- **中间层损失的集成：**中间层损失的集成如下图所示。与逻辑分布的集成类似，该方法利用多个教师模型和学生模型的中间层分别构建损失，然后这些的损失相加并取平均值，从而得到最终的损失。



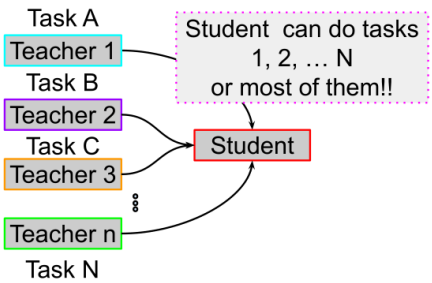
- **数据扩充蒸馏：**数据扩充蒸馏如下图所示。该方法提出了通过统一来自多个教师的数据源来进行数据蒸馏，目标是通过各种数据处理方法（例如，数据扩充）为未标记的数据生成标签，以训练学生模型。



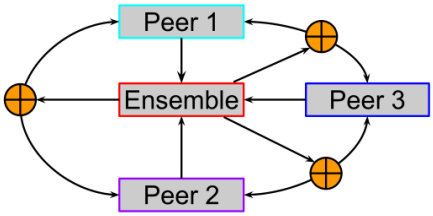
- **子教师生成：**子教师生成的蒸馏如下图所示。该方法通过教师模型来生成多个子教师模型，利用子教师模型展开多教师知识蒸馏方法。



- **跨领域蒸馏：**跨领域蒸馏如下图所示。该方法是将在不同领域（通常是相同任务的不同数据集或者不同任务）的教师模型的知识蒸馏到学生模型上，从而提高学生模型的泛化能力，学习到多个领域的知识。



- **组队蒸馏：**组队蒸馏如下图所示。在知识蒸馏的过程中，包含了多个教师模型和多个学生模型。每个学生模型不仅向教师模型学习，还会向同伴学习。



（3）基于数据格式的知识蒸馏

- **无数据的蒸馏：**大多数知识蒸馏方法假设原始网络（教师）的训练数据对目标网络（学生）的是可见的，然而，由于隐私和传输问题，训练数据集在现实应用中有时是不可见的。为了解决这个问题，需要进行无数据知识蒸馏。无数据的蒸馏有三种策略，分别是基于元数据的蒸馏、基于类别相似度的蒸馏和基于生成器的蒸馏。
- **基于少量数据的蒸馏：**大多数具有教师-学生结构的知识蒸馏方法都是基于匹配信息（例如，逻辑、提示），并使用拥有完整标注的大规模训练数据集来进行知识蒸馏的训练。因此，这样的知识蒸馏仍然是数据量大、处理效率低的。为了在使用少量训练数据的同时实现学生的有效学习，需要提出基于少量样本知识蒸馏策略。基于少量样本知识蒸馏策略一般是基于生成伪训练示例的，或者基于分层估计度量来调整教师模型和学生模型。
- **跨领域知识蒸馏：**跨模态学习的知识蒸馏通常使用包含模态特定表示或共享层的网络架构来进行，利用不同域对应的训练图像。而是否可以将一项任务的知识从预先训练好的教师网络转移到另一项任务中的学生模型上是重要的问题。由于用于跨领域学习的知识蒸馏与用于域自适应的知识蒸馏存在本质上的不同，即其中数据是从不同的域独立绘制的。与前面提到的侧重于在教师和学生之间的同一领域内转移知识的蒸馏方法相比，跨领域知识蒸馏使用教师的表示作为监督信号来训练学生去学习另一项任务，在这个问题设置中，学生需要依靠老师的视觉输入来完成任务。

（4）基于新型度量方式的蒸馏

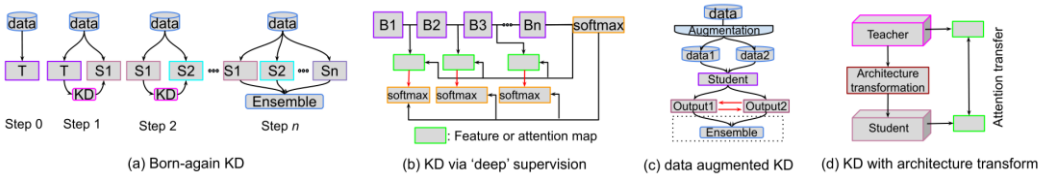
- **通过对抗学习的蒸馏：**主流的知识蒸馏方法存在一个问题，即学生很难从老师那里了解真实的数据分布，因为老师无法完美地模拟真实的数据分配。而生成对抗性网络（GAN）已被证明在学习图像翻译中的真实数据分布方面具有潜力。因此，最近的研究

试图探索对抗性学习以提高知识蒸馏的性能。

GAN 通过训练生成器 G 和鉴别器 D 来生成连续数据，这将惩罚生成器 G 产生不可信的结果。生成器 G 从使用特定分布采样的随机噪声 z 产生合成示例 G（z）。这些合成的示例与从真实数据分布 p（x）采样的真实示例一起被馈送到鉴别器 D。鉴别器 D 试图区分这两个输入，并且生成器 G 和鉴别器 B 都提高了它们在最小最大游戏中的各自能力，直到鉴别器 C 无法区分真假。

$$\min_G \max_D J(G,D) = \mathbb{E}_{x\sim p(x)}[log(D(x))] + \mathbb{E}_{z\sim p(z)}[log(1-D(G(z)))]$$

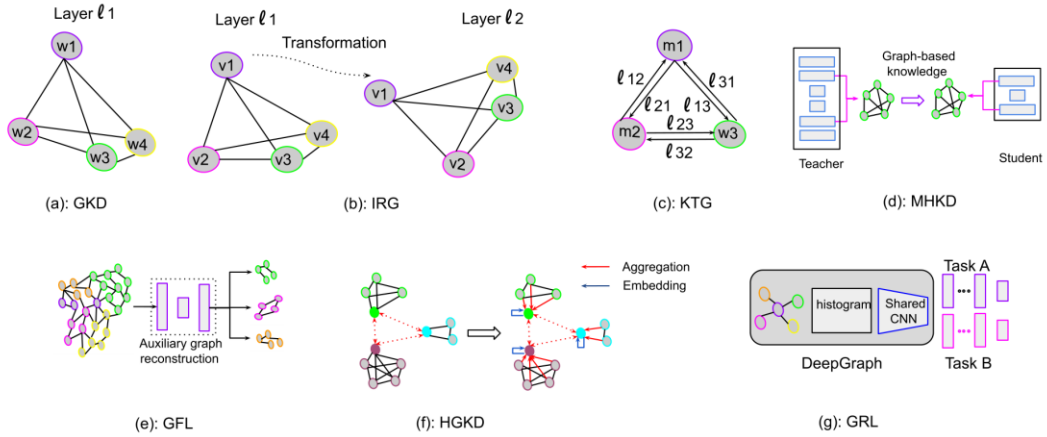
基于传统 GAN 的知识蒸馏如下图所示。



(a) Born-again KD　(b) KD via 'deep' supervision　(c) data augmented KD　(d) KD with architecture transform

基于鉴别器预测逻辑有几个好处：首先，学习损失可以在图像翻译任务中有效；其次，网络输出的多模态密切相关，不需要像通常那样精确模拟一个教师网络的输出以获得良好的学生表现。然而，由于鉴别器仅捕获教师和学生输出的高级统计信息，因此缺少低级特征对齐。

- **通过图表征的蒸馏：**常用的知识蒸馏方式是基于逻辑或特征信息对的知识蒸馏。然而，知识蒸馏的一个关键问题是数据。通常，训练神经网络需要嵌入高维数据集以便于数据分析。因此，训练教师模型的最佳目标不仅是将训练数据集转换为低维空间，而且是分析数据内关系。然而，大多数知识蒸馏方法不考虑这种关系。而基于图嵌入和知识图的知识蒸馏方法解决了这一问题。



(a): GKD　(b): IRG　(c): KTG　(d): MHKD

(e): GFL　(f): HGKD　(g): GRL

图形神经网络（GNN）是一种直接对图形结构进行操作的 DNN。一个典型的应用是关于节点分类。在节点分类问题中，第 i 个节点 vi 的特征是其特征 xvi 和真实标签 tvi。因此，给定标记图 G，目标是利用标记节点来预测未标记节点。它学习用包含其邻域信息的 d 维向量状态 hvi 表示每个节点。具体而言，hvi 在数学上可以描述为：

$$\mathbf{h}_{v_i} = f_t(\mathbf{x}_{v_i}, \mathbf{x}_{co[v_i]}, \mathbf{h}_{ne[v_i]}, \mathbf{x}_{ne[v_i]})$$

$$\mathbf{o}_{v_i} = f_o(\mathbf{h}_{v_i}, \mathbf{x}_{v_i})$$

基于以上关于图表示和 GNN 的基本原理的解释，现在深入研究现有的基于图的蒸馏技术。为了更好地捕捉两个词汇表之间的关系，可以构建了一个二分图 G=（V，E），将它们划分为视觉单词簇。通过这种方式，来自 BoVW 的知识可以被转移到视觉单词集群更具辨别力。

- **基于增量学习的蒸馏：**增量学习不断学习新知识以更新模型的知识，同时保持现有的旧知识。根据用于蒸馏的教师网络的数量，基于增量学习的知识蒸馏可以分为两种类型：从单个教师蒸馏和从多个教师蒸馏。

2 行人搜索

2.1 行人搜索的研究背景

行人搜索旨在从一系列未经裁剪的图像中对行人进行定位与识别，融合了行人检测和行人重识别两个子任务。该任务最早于 2013 年在 ACM 多媒体大会上提出，首次将行人检测与行人重识别任务整合为一个任务。

行人检测属于目标检测的子任务，旨在从大量的照片或者视频数据中找到行人的位置和大小，通常来说需要使用矩形的框将其框出来，而这一任务不需要识别框选出来的行人是谁；行人重识别则是行人匹配的任务，通常是从一系列已经裁剪好的行人图像中匹配一个和待匹配目标最像的人。将行人检测与行人重识别结合的行人搜索任务更具有实用价值，在可以应用在刑侦中，从而节省大量的人力资源开销。

由于行人搜索的任务结合了两个独立性较强的子任务，因此其解决方案通常有两种：两阶段模型和端到端模型。两阶段模型是分步骤解决行人搜索任务：首先使用深度学习方法将图片中的行人检测出来，对于检测出来的模型，使用行人重识别的模型进一步完成匹配任务。两阶段的模型需要分开进行两个子任务，使得原始任务的难度降低，但也会使得行人搜索的任务效率下降，因此端到端的行人搜索算法的研究受到了更多的重视和研究。端到端的行人搜索网络将行人检测和行人重识别集中到一个网络中处理，使得网络的训练和推理可以针对一个网络来完成，这样的模型会更具有实用性。

2.2 行人搜索的算法

（1）RCNN

RCNN 是目标检测领域的经典方法，出自于 2014 年的论文《Rich feature hierarchies for accurate object detection and semantic segmentation》。

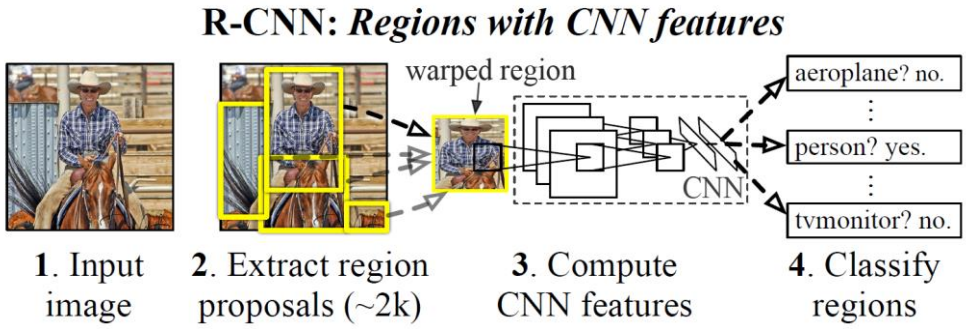
RCNN 将目标检测分为分类与回归两个并行进行的子任务，分类任务是指将 bounding box 中内容进行分类，回归任务是指通过回归的思想寻找 bounding box 的准确位置。其推理过程包含以下四个步骤：

1) 使用 Selective Search 算法生成区域提案（Proposals）；
2) 将面积大小不相同的 proposals 缩放到统一的大小；

3) 将得到的结果输入特征提取网络得到特征向量；

4) 使用分类器得到区域的类别和 bboxes 的回归值。

算法的示意图如下：



在实际使用中，对于一张图片，RCNN 的方法要独立的将提取的两千多个区域独立地输入特征提取网络，并且需要训练复杂的 SVM 分类器，因此该方法的效率很低。

（2）Fast RCNN

Fast RCNN 是对 RCNN 的改进，出自于 2015 年的论文《Fast R-CNN》。

Fast RCNN 提出了 ROI pooling 的方法，使得不需要单独地将 Selective Search 得到的 Proposals 输入特征提取网络，二是可以并行处理这些特征。该方法的推理过程包含以下几个步骤：

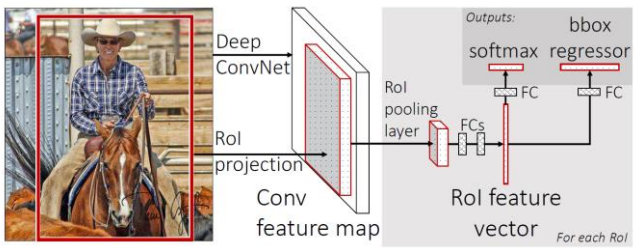
1) 使用 Selective Search 算法生成区域提案（Proposals）；

2) 将原始图像输入特征提取网络得到整个图像的特征图（feature map），将 proposals 与特征图上的区域对应起来，并将其 pooling 到指定大小；

3) 使用两层全连接网络将得到的结果提取为 ROI 特征向量；

4) 使用 softmax 和 bbox regressor 得到结果。

该方法由于使用了 ROI pooling 的方法和使用简单分类器，使得目标检测的效率大幅度提高，但仍然存在需要使用 Selective Search 算法生成大量冗余且不精确的区域提案（Proposals）的不足之处。算法的示意图如下：

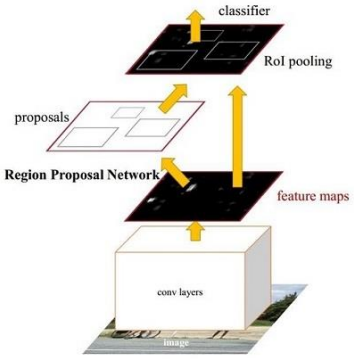


装 订 线

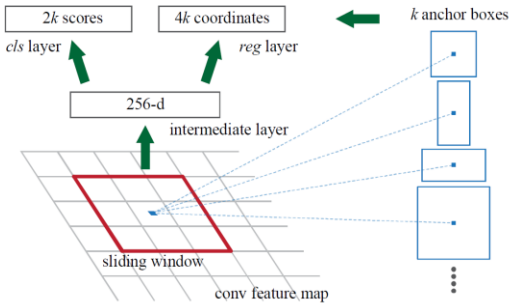
（3）Faster RCNN

Faster RCNN 由是对 Fast-RCNN 的进一步改进，出自于 2016 年的论文《Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks》。

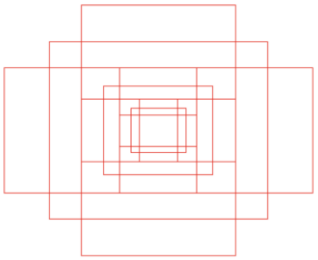
该方法的突出贡献在于提出了 RPN（region proposal network）网络，使得不再需要使用 Selective Search 算法生成区域提案。Faster RCNN 的网络结构如下，是由 Fast RCNN 和 RPN 组成：



RPN 网络的结构图如下：



其推理过程如下：首先在 feature map 上，使用 3×3 的滑动窗口提取 256 维特征向量，使用该特征向量得到一组 anchor boxes 的分类和回归结果。一组 anchor boxes 包含 9 个 anchor，对应原图上{1:1 1:2 2:1}×{128^2 256^2 512^2}的九个大小不同的区域，即中心点对应的如下区域：



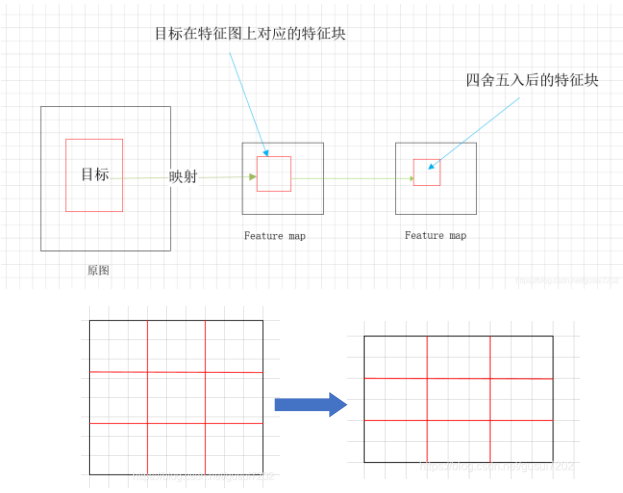
然后排除超过原始图像边界的区域，将这些区域计算其前景/背景的类别和回归参数，从而得到区域提案 proposals。然后使用非极大抑制算法（NMS）排除高度相似的 proposals，将最终剩下的 proposals 输入原来的 Fast RCNN 中。

Faster RCNN 的 RPN 结构使得模型可以自行生成较为精确的区域提案，同时 RPN 结构可以和 Fast RCNN 联合训练，这使得目标检测真正具有一个端到端的算法。同时，在 Faster RCNN 中，特征提取网络使用了性能更好的 VGG 网络，使得目标检测的精确度得到提升。

（4）ROI-Align

ROI-Align 是对 ROI pooling 的一个改进，该方法来源于 2017 年的论文《Mask R-CNN》中。

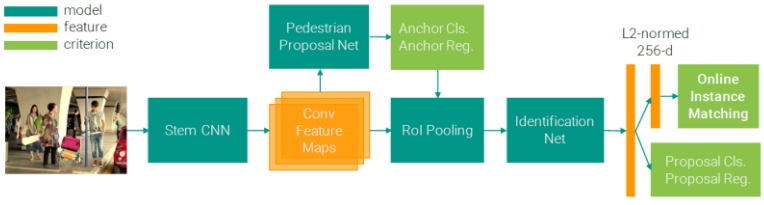
由于 ROI pooling 的输入是 proposals 的坐标，这些坐标由 RPN 计算得到，由于经过对 anchor boxes 的一次回归，所以 proposal 映射到 feature map 上是一个浮点数的坐标，往往是无法对应特定像素的，因此需要进行一次取整，让 proposals 与 feature map 对映。而计算 ROI pooling 时，还要经过一次取整，才能将其映射到指定大小。这两次取整操作其实让 ROI pooling 得到的特征对应的原始区域与 proposal 相差很大，导致了目标检测的性能不佳。



而 ROI-Align 使用双线性插值的方法计算每一个浮点坐标在 feature map 上对应的值，从而进行类似于 ROI pooling 时不会产生很大误差。

（5）OIM

OIM 是行人搜索网络的一个损失，其出自 19 年的论文《Joint Detection and Identification Feature Learning for Person Search》。其对应的网络结构如下图所示：



相较于 Faster RCNN，其不同之处在于：该网络原来提取的 ROI 特征向量从另一个分支使用全连接层降维到 256 维，并进行 L2 规范化，使用这个结果进行在线实例匹配（Online Instance Matching, OIM）。

在行人搜索中，需要将 query 中的人像与 gallery 中的检测到的人进行匹配，而 query 中的人像通常类别很多，因此使用 softmax 并不很合适。

在模型推理时，该模型首先将 query 中的人像通过网络的 ReID 阶段提取的特征向量保存到 LUT（lookup table）中，然后将 gallery 中的图像通过全部的网络，计算 proposal 区域得到的人的特征向量，与 LUT 中的一一比较余弦相似度，从而得到最匹配的人，同时还要在线更新 LUT 中最匹配记录的特征向量：

$$v_t \leftarrow \gamma v_t + (1-\gamma)x$$

并进行 L2 规范化。

在模型训练时，使用 OIM 的损失来自两部分，其思想是拉近同一行人的特征向量的距离、拉远不同行人间特征向量的距离，首先计算被框中的行人与被标注前景的余弦相似度，再计算被框中的行人与未被标注前景的余弦相似度，构建 softmax 概率：

$$p_i = \frac{\exp(v_i^T x/\tau)}{\sum_{j=1}^{L}\exp(v_j^T x/\tau)+\sum_{k=1}^{Q}\exp(u_k^T x/\tau)}$$

$$q_i = \frac{\exp(u_i^T x/\tau)}{\sum_{j=1}^{L}\exp(v_j^T x/\tau)+\sum_{k=1}^{Q}\exp(u_k^T x/\tau)}$$

其中，LUT 仍需在线更新，而未被标注前景是指最近的 batch 中的。

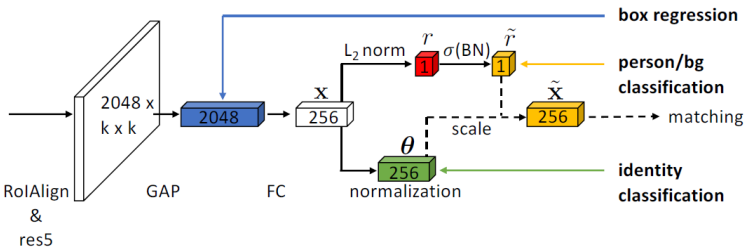
然后构建 OIM 部分的损失函数：

$$\mathcal{L} = \mathrm{E}_x\left[\log p_t\right]$$

整个网络在四个损失的监督下联合训练。

（6）NAE

NAE 也是对应一种特定行人搜索网络的损失，其出自 20 年的论文《Norm-Aware Embedding for Efficient Person Search》。该方法对应的网络结构图如下：



NAE 与 OIM 网络不同之处在于前景/背景分类所使用的特征，OIM 使用的是 ROI 特征向量，该特征向量无法很好地反映前景/背景信息，因此在 OIM 的基础上，NAE 网络将提取出的 L2 规范化后的特征向量的长度作为前景/背景分类依据，这个指标可以很好地反映前景/背景信息。

将特征向量 x 进行 L2 规范化的公式如下：

$$\mathbf{x} = r \cdot \boldsymbol{\theta}$$

θ 为 L2 规范化后的单位长度向量，与 OIM 一致，该向量用于训练过程匹配目标人像。而 r 作为 x 的长度，其经过 BN 操作后可以很好的代表前景/背景信息，由于匹配相似度计算为：

$$\text{sim}(\tilde{\mathbf{x}}_q, \tilde{\mathbf{x}}_g) = \tilde{\mathbf{x}}_q^T \tilde{\mathbf{x}}_g = \tilde{r}_g \cdot \boldsymbol{\theta}_q^T \boldsymbol{\theta}_g$$

r 接近 1，代表前景，此时匹配相似度越高；r 越接近 0，代表背景，此时匹配相似度越低。对于 r 的训练采用交叉熵作为监督：

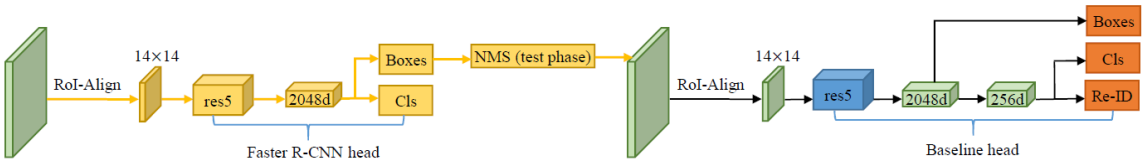
$$\mathcal{L}_{\text{det}} = -y \log(\tilde{r}) - (1-y) \log(1-\tilde{r})$$

NAE 网络在 OIM 基础上进一步精心设计前景/背景分类监督，使得特征向量 x 在在匹配过程中的余弦相似度更具有意义。从而进一步提高行人搜索的准确度。

（7）SeqNet

SeqNet 是序列化的行人搜索网络，出自 21 年的论文《Sequential End-to-end Network for Efficient Person Search》，在 CUHK-SYSU 数据集上以 94.8 的 mAP 成为当时的 SOTA。

SeqNet 在 NAE 的基础上增加了两个改进，一个是增长了网络结构，使得模型可以使用更精确的 proposals 来进行后续的 ReID，如下图所示：



可以看到，ReID 阶段使用的 proposals 是经过完整的 Faster RCNN 回归输出的更为准确的 proposals，这无疑为行人匹配提供了更高质量的区域提案，这使得端到端的模型准确度可以赶上两阶段模型的准确度。

另一个改进是 SeqNet 提出了一种上下文二分图匹配（CBGM）的方法，该方法充分利用了图像中的上下文信息，即图像中可能包含的其他行人的信息，使得匹配过程不是单一地用框出的人与所有待匹配人进行，而是考虑整个图像中所有人与待匹配人的最佳匹配。以论文中的例子进行说明，如下图：

若仅考虑(a)与(c)、(d)的匹配，那么(a)将与更高概率的(d)错误匹配；但如果考虑 Query 的整张图像的匹配，即考虑(a)、(b)与(c)、(d)的匹配，则(b)与(d)将先匹配，此时(a)就能正确地和(c)匹配。这个问题可以抽象为二分图的最佳匹配问题，使用经典的 Kuhn-Munkres 算法求解。

3 人工智能模型的数据研究

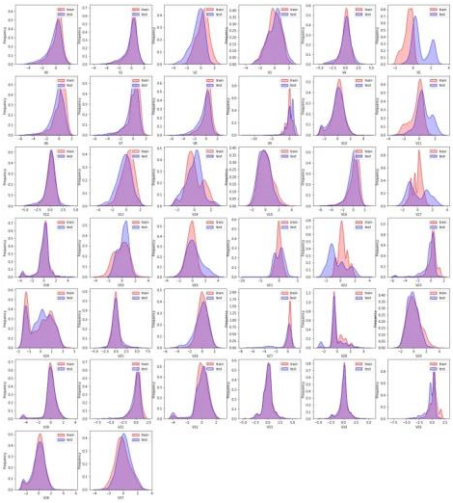
3.1 数据分布与数据一致性

（1）数据一致性的概念

数据一致性是指训练集的数据分布和测试集的数据分布是否一致（或近似）。

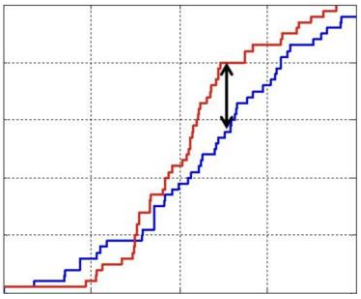
（2）数据一致性的判别

- **核概率密度估计（KDE）分布图：**KDE 分布图即 Kernel Density Estimation 核概率密度估计。可以理解为是对直方图的加窗平滑。通过 KDE 分布图，可以查看并对训练数据集和测试数据集中特征变量的分布情况。KDE 图的例子如下：



从图中可以通过观察判断训练集和测试集的数据一致性。

- **K-S 检验：**K-S 检验是一种非参数检验方法，用于检验两个数据分布是否一致。其利用两个数据分布的分布函数来进行计算。该方法首先构造训练集和测试集的经验分布函数：



然后计算下述检验统计量：

$$D_{n,m} = \sup_x |F_{1,n}(x), F_{2,m}(x)|$$

该检验统计量反映的是经验分布函数的最大差值。然后进行检验：

$$D_{n,m} > \sqrt{-\frac{1}{2}\ln\frac{\alpha}{2}} * \sqrt{\frac{m+n}{m*n}}$$

若满足上述条件，则来自同一分布，否则就来自不同分布。

- **对抗验证**
  - 构建一个样本的分类器，该二分类器的任务是区分样本来源于训练集还是测试集。
  - 将新的训练数据集进行划分，保留部分样本作为该样本分类任务的测试集，利用分类算法对数据集进行训练，AUC 作为模型指标；
  - 在测试集中进行验证，如果模型效果 AUC 在 0.5 左右，说明该样本分类模型无法区分样本来源训练集，还是测试集，说明原始数据中训练集，测试集分布是一致的；
  - 如果 AUC 较大，说明样本分类器很容易区分样本，间接说明训练集与测试集存在很大差异；根据第 3 步的结论，对于分布一致的，正常对目标任务训练即可。对于分布不一致的，可以继续进行样本挑选的尝试。

（3）数据分布不一致

数据分布一致性较差时，需要进行数据增强。

3.2 数据完整性

数据完整性是指数据生命周期内所有数据完整、一致、准确的程度。

3.3 深度学习里的数据科学——讲座笔记

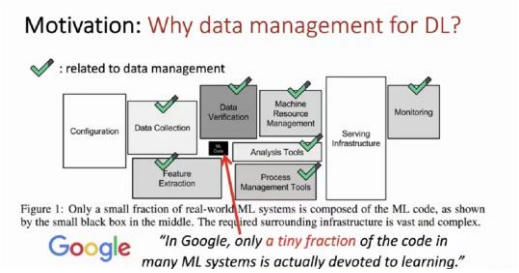
（1）背景

深度学习无处不在：

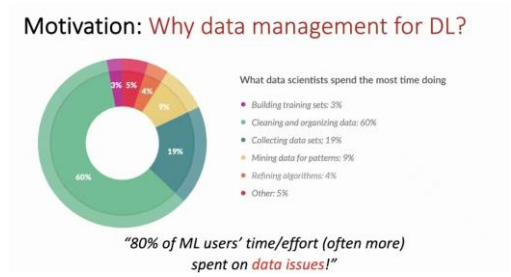
数据十分珍贵：

（2）动机

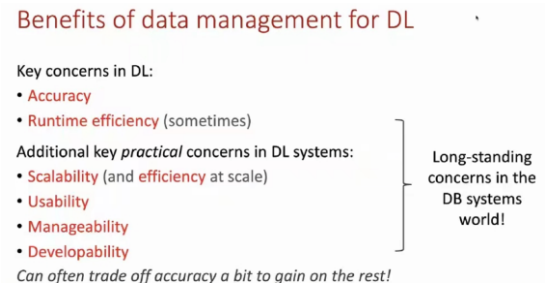
数据管理的意义重大：



大部分的时间都花在 data issues 上：



数据管理的优势：



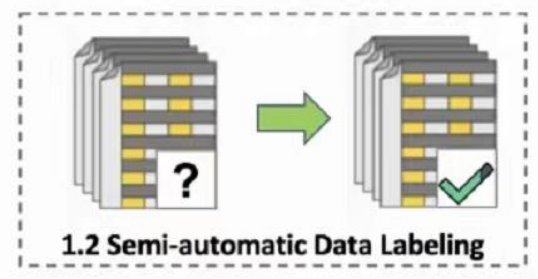
（3）数据管理面临的挑战

- data preparation for DL
- Optimized training in DL
- Result validation and explanation in DL

数据准备的挑战：

Data extraction and integration

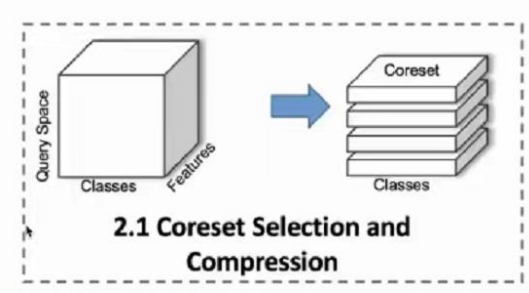
Semi-automatic data labeling



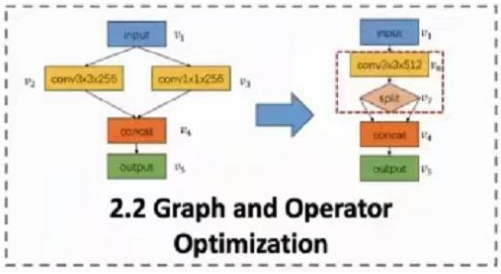
优化训练的挑战：

优化训练框架是效率和准确度的最好平衡。有两个优化训练的途径：

Coreset selection and data compression



Graph and operator optimization

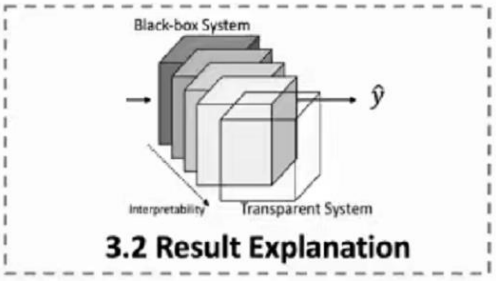


结果认证和结果解释的挑战：

结果认证确保了深度学习模型的效率，结果解释提高了深度学习模型的透明度。

Result validation

Explanation in DL

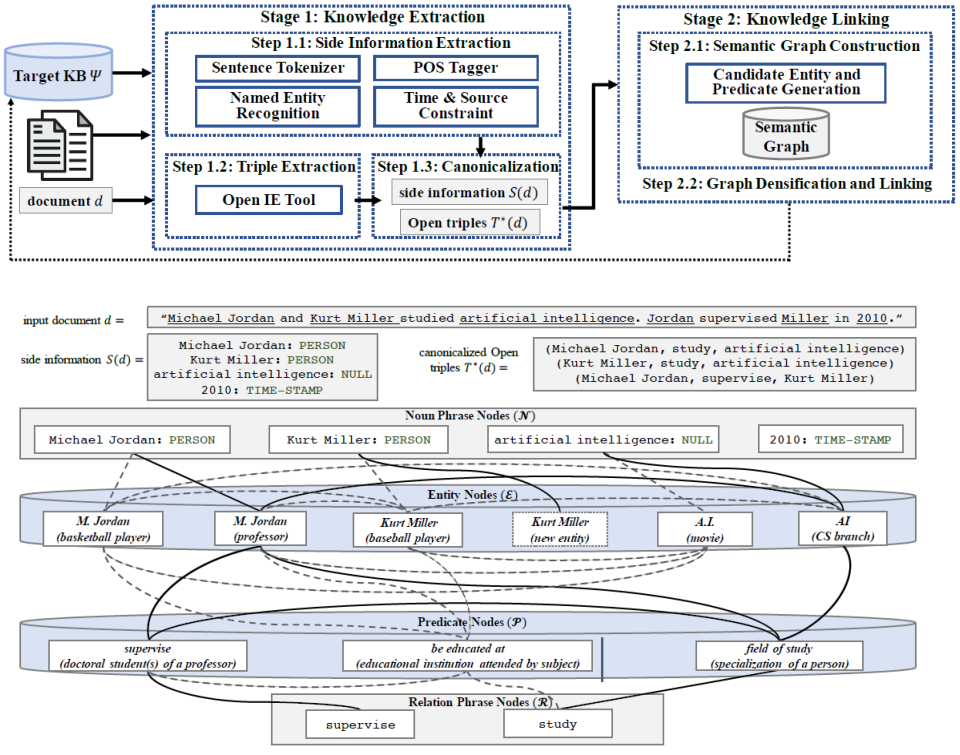


（4）现有工作

- 知识提取方面：

KBPearl: A Knowledge Base Population System Supported by Joint Entity and Relation Linking

KBPearl system 是一种可以在线扩充知识库的方法

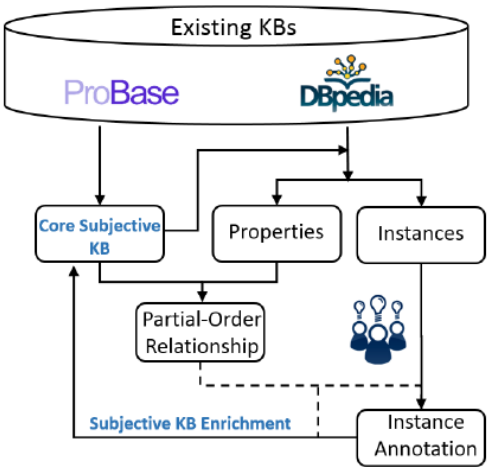


装 订 线

➢ 知识标注方面：

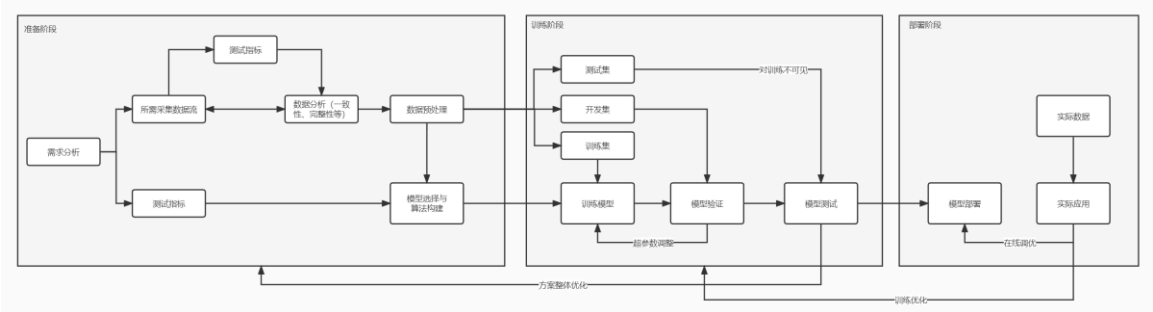
Subjective Knowledge Base Construction Powered By Crowdsourcing and Knowledge Base

一种自动化的标注方法：



4 人工智能模型的生命周期与评价标准

4.1 AI 模型生命周期



模型的生命周期总的来说分为准备、训练、部署三个阶段。

在准备阶段，首先需要对任务进行需求分析，了解 AI 系统需要完成的任务。随后，需要确定 AI 模型的评价指标，并且从数据流处采集数据。对采集到的数据需要进行一致性和完整性的分析，并且更据分析结果进行数据预处理。在此基础上，完成模型的选择与算法的构架。

在训练阶段，数据需要划分为训练集、开发集和测试集三个互相独立的部分，训练集用于训练模型，开发集用于进行模型超参数的调优，测试集用于模拟真实情况中的数据。通过反复的测试调优，得到最终的模型。

在部署阶段，模型首先被部署到平台上，用于处理真实情况中的数据，并且进行在线学习以逐步调优。与此同时，需要对整个 AI 系统进行维护。

4.2 一些评价标准

（1）F-score

F1-Score 是统计学中用来衡量二分类模型精确度的一种指标，它同时兼顾了分类模型的准确率和召回率。F1-Score 是模型准确率和召回率的一种加权平均，其最大值是 1，最小值是 0，并且值越大意味着模型越好。

	真实1	真实0
预测1	真阳性 TP	假阳性 FP
预测0	假阴性 FN	真阴性 TN

装 订 线

查准率定义如下：

$$precision = TP/(TP + FP)$$

召回率定义如下：

$$recall = TP/(TP + FN)$$

F1-score 的定义如下：

$$F1 = (2 \times precision \times recall)/(precision + recall)$$

F2-score 的定义如下：

$$F2 = (5 \times precision \times recall)/(4 \times precision + recall)$$

F_β-score 的定义如下：

$$F\beta = [(1 + \beta^2) \times precision \times recall]/(\beta^2 \times precision + recall)$$

（2）语义分割任务的评价标准

- Precision

准确度是检测出的像素中是边缘的比例：

$$precision = \frac{detectedtrueboundarypixels}{detectedboundarypixels}$$

- Recall Rate

召回率是检测出的正确边缘占所有边缘的比例：

$$recall = \frac{detectedtrueboundarypixels}{alltrueboundarypixels}$$

- F-score

F-score 是两者的调和平均数：

$$F - score = \frac{2 * precicion * recall}{presicion + recall}$$

- Accuracy

假定一定有 k+1 类（包括 k 个目标类和 1 个背景类），pij 表示本属于 i 类却预测为 j 类的像素点总数，pii 表示 true postives，pij 表示 false positives，pji 表示 false negatives

- Pixel Accuracy (PA)

分类正确的像素点数和所有的像素点数的比例：

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}}$$

优点：简单

缺点：如果图像中大面积是背景，而目标较小，即使将整个图片预测为背景，也会有很高的 PA 得分，因此该指标不适用于评价以小目标为主的图像分割效果

- Mean Pixel Accuracy (MPA)

计算每一类分类正确的像素点数和该类的所有像素点数的比例然后求平均：

$$MPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}}$$

- Mean Intersection over Union (MIoU)

计算每一类的 IoU 然后求平均：

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}}$$

- Frequency Weighted Intersection over Union (FWIoU)

根据每一类出现的频率对各个类的 IoU 进行加权求和：

$$FWIoU = \frac{1}{\sum_{i=0}^{k}\sum_{j=0}^{k}p_{ij}}\sum_{i=0}^{k}\frac{p_{ii}\sum_{j=0}^{k}p_{ij}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}}$$

（3）图像上色任务的评价标准

- 视觉真实性(AMT)：为了衡量图像真实程度，设计了如下评价方法——随机给出一组图片（里面包含真实图片和着色后的图片），每次呈现两张图片，让人类受试者从中选择他们认为包含不自然颜色的图片
- 原始准确率(AuC)：使用了真值图像和上色图像 ab 通道的 L2 距离
- 语义解释性(VGG 分类)：用 VGG 网络来分类上色图片，根据分类效果和原始图片分类效果的好坏来判断上色的逼真性

（4）图像修复任务的评价标准

- 峰值信噪比（PSNR）：

给定一个大小为 M×N 的干净图像 I 和噪声图像 K，均方误差（MSE）定义为：

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2$$

然后 PSNR（dB）定义为：

$$PSNR = 10\cdot log_{10}(\frac{MAX_I^2}{MSE})$$

其中 MAX_I^2 为图片可能的最大像素值。

- **结构相似性（SSIM）：**

SSIM 公式基于样本 x 和 y 之间的三个比较衡量：亮度、对比度和结构。

$$l(x,y)=\frac{2\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1}c(x,y)=\frac{2\sigma_x\sigma_y+c_2}{\sigma_x^2+\sigma_y^2+c_2}s(x,y)=\frac{\sigma_{xy}+c_3}{\sigma_x\sigma_y+c_3}$$

则 SSIM 的定义如下：

$$SSIM(x,y)=[l(x,y)^\alpha\cdot c(x,y)^\beta\cdot s(x,y)^\gamma]$$

若将指数位置的三个参数设置为 1，则得到

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$$

每次计算的时候，都从图片上去一个 N×N 的窗口，然后滑动窗口进行计算，最后取平均值作为全局的 SSIM

- Inception Score(IS)

计算 Inception Score 需要用到 Google 的预训练的 Inception-V3 网络，这个网络最初是在 ImageNet 上做图像分类的，输入一个图片，输出一个 1000 维的 tensor 代表输出类别。

Inception Score 从两个方面评价 GAN 生成的图片的质量：

清晰度：单看一张图片，把它输入到 Inception V3 中，得到输出 1000 维的张量 y。对于一个清晰的图片，它属于某一类的概率应该非常大，而属于其它类的概率应该很小。即 p(y|x) 的熵应该很小（因为熵代表了混乱度，分布的确定性越高，熵越低）。

多样性：从所有的图片的角度考虑，在生成的一堆图片中，如果这些图片是十分具有多样性的，那么应该是每个类别的数目是差不多一样的，也就是说 p(y) 的熵应该很大。

所以说最小化 p(y|x) ，最大化 p(y) ，也就是要让这两个概率分布的差距越大越好。衡量两个概率分布的相似度的方式，那就是 KL 散度：

$$D_{KL}(P,Q)=\mathbb{E}_{x\sim P}[log\frac{P(x)}{Q(x)}]=\mathbb{E}_{x\sim P}[log(P(x))-log(Q(x))]$$

所以 IS 的公式如下：

$$\mathbf{IS}(G)=\exp\left(\mathbb{E}_{\mathbf{x}\sim p_g}D_{KL}(p(y\mid\mathbf{x})\|p(y))\right)$$

- FID

令 p(·)和 pw(·)分别代表模型生成数据的概率分布和来自真实世界的概率分布，m 和 C 分别为 p(·)的均值和协方差，m_w 和 C_w 分别为 pw(·)的均值和协方差，FID 的计算如下：

$$d^2((m,C),(m_w,C_w)) = ||m-m_w||_2^2 + Trace(C+C_w-2(CC_w)^{1/2})$$

FID 正是衡量了生成样本与真实世界样本之间的距离