**Transfer Learning**

**引言**

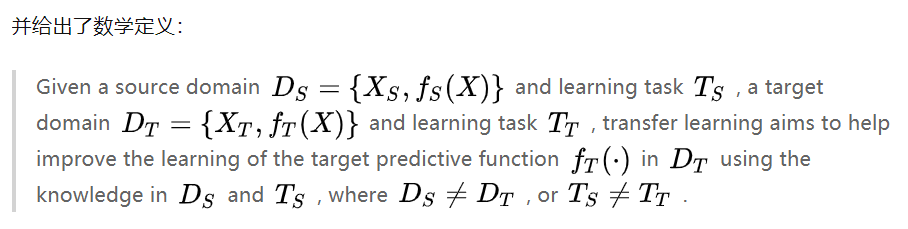
**数据依赖**是深度学习中最严重的问题之一。与传统的机器学习方法相比，深度学习对海量训练数据的依赖性非常强，因为它需要大量的数据来理解数据的潜在模式。一个有趣的现象是，模型的规模和所需数据量的大小几乎是线性关系。一个较为合理的解释是，对于一个特定的问题，模型的表达空间必须足够大，以发现数据下的模式。模型中的预定层可以识别训练数据的高级特性，后续层可以识别帮助做出最终决策所需的信息。

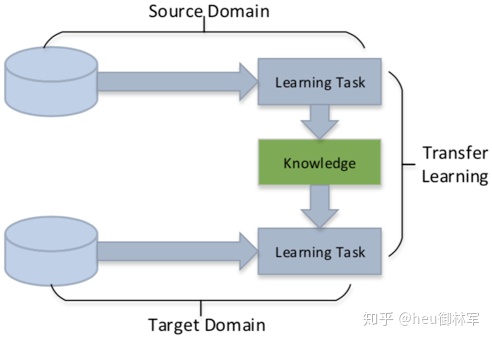
**训练数据不足**是一些特殊领域中不可避免的问题。数据的收集是复杂和昂贵的，这使得构建大规模、高质量的带注释的数据集非常困难。例如，生物信息学数据集中的每个样本经常显示一个临床试验或一个痛苦的病人。此外，即使我们付出了昂贵的代价来获取训练数据集，也很容易过时，不能有效地应用于新的任务中。

迁移学习放松了训练数据必须与测试**数据独立且同分布(i.i.d)**的假设，激励我们利用迁移学习来解决训练数据不足的问题。在迁移学习中，训练数据和测试数据不需要是i.i.d。不需要对目标域内的模型进行从零开始的训练，可以显著降低对目标域内训练数据和训练时间的需求。

**简介**

Transfer Learning是机器学习的分支，很多方法其实是不需要用NN的。Transfer Learning和神经网络联系如此紧密，主要因为NN的发展太快，太强大，太火爆，导致Transfer Learning的研究都往神经网络靠了。





**通俗理解**

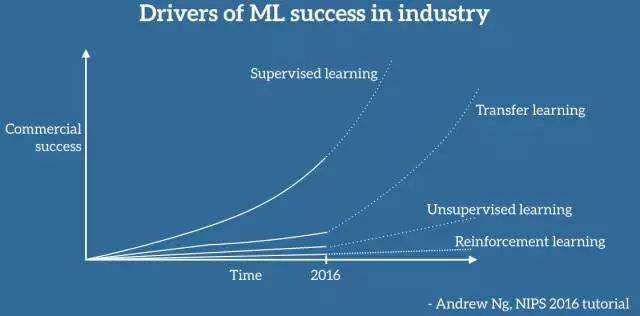
迁移学习(transfer learning)通俗来讲，就是运用已有的知识来学习新的知识，核心是找到已有知识和新知识之间的相似性，用成语来说就是举一反三。比如已经会编写Java程序，就可以类比着来学习C#。如何合理地找寻任务之间的相似性，进而利用这个桥梁来帮助学习新知识，是迁移学习的核心问题。

**负迁移**

理论上，任何领域之间都可以做迁移学习。但是，如果源域和目标域之间相似度不够，迁移结果并不会理想，出现所谓的负迁移情况。比如，一个人会骑自行车，就可以类比学电动车；但是如果类比着学开汽车，那就有点天方夜谭了。如何找到相似度尽可能高的源域和目标域，是整个迁移过程最重要的前提。

**前景**

吴恩达在 2016 年 NIPS 上谈论了迁移学习，据吴恩达说，迁移学习将成为机器学习在产业界取得成功的一个关键驱动力。



**迁移学习的分类——按学习方式分类**

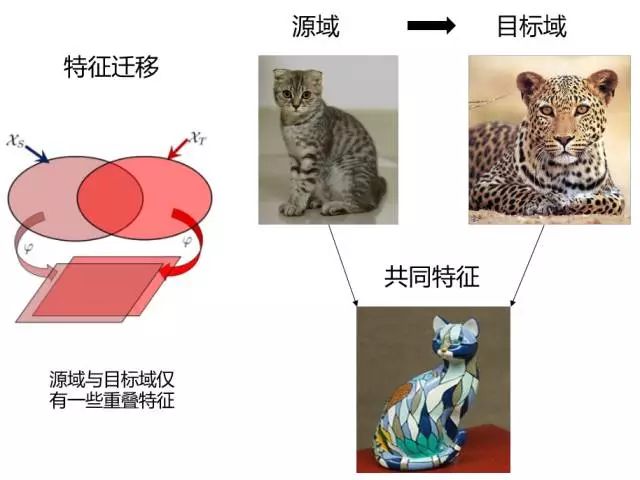
1. **基于样本的迁移：**通过对源域中有标定样本的加权利用完成知识迁移

在源域中找到与目标域相似的数据，把这个数据的权值进行调整，使得新的数据与目标域的数据进行匹配。下图的例子就是找到源域的例子3，然后加重该样本的权值，使得在预测目标域时的比重加大。优点是方法简单，实现容易。缺点在于权重的选择与相似度的度量依赖经验，且源域与目标域的数据分布往往不同。



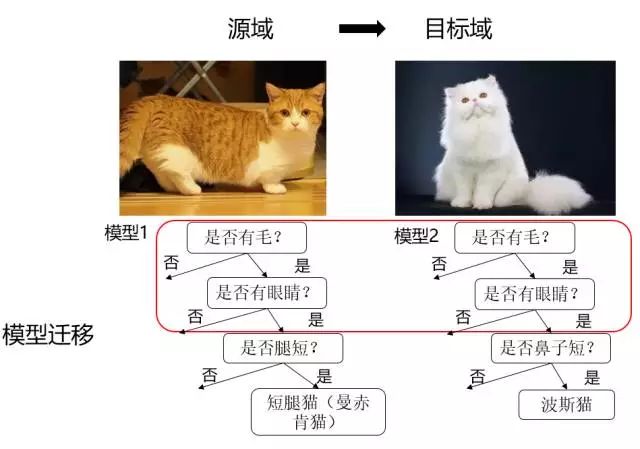
1. **基于特征的迁移：**通过将源域和目标域映射到相同的空间（或者将其中之一映射到另一个的空间中）并最小化源域和目标域的距离来完成知识迁移

假设源域和目标域含有一些共同的交叉特征，通过特征变换，将源域和目标域的特征变换到相同空间，使得该空间中源域数据与目标域数据具有相同分布的数据分布，然后进行传统的机器学习。优点是对大多数方法适用，效果较好。缺点在于难于求解，容易发生过适配。



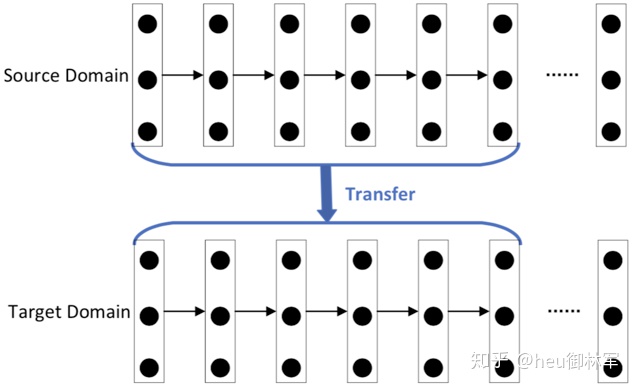
1. 基于模型的迁移：将源域和目标域的模型与样本结合起来调整模型的参数

假设源域和目标域共享模型参数，是指将之前在源域中通过大量数据训练好的模型应用到目标域上进行预测，比如利用上千万的图象来训练好一个图象识别的系统，当我们遇到一个新的图象领域问题的时候，就不用再去找几千万个图象来训练了，只需把原来训练好的模型迁移到新的领域，在新的领域往往只需几万张图片就够，同样可以得到很高的精度。优点是可以充分利用模型之间存在的相似性。缺点在于模型参数不易收敛。



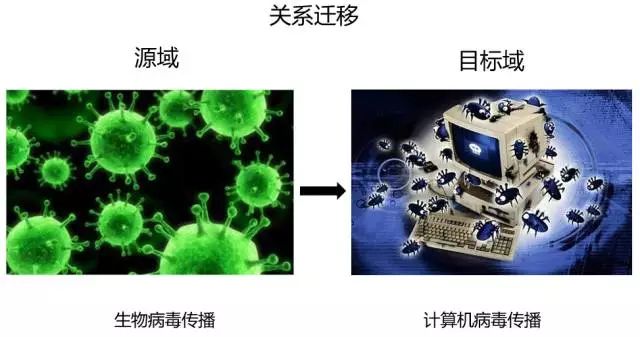
**应用例子**

为什么深度神经网络模型具有强大的可迁移能力？回答这一点需要了解DNN的原理，DNN是一个通过pre-train获得数据的分层特征表达，然后用高层语义分类的模型，模型的底层是低级语义特征（比如说，边缘信息，颜色信息等），这样的特征实际上在不同的分类任务中都是不变的，而真正区别的是高层特征，这也解释了通常使用新的数据集去更新AlexNet，GoogleNet的最后几层网络权值，来实现简单的“迁移”这个小trick。



1. 以及基于关系的迁移：通过在源域中学习概念之间的关系，然后将其类比到目标域中，完成知识的迁移

假设两个域是相似的，那么它们之间会共享某种相似关系，将源域中逻辑网络关系应用到目标域上来进行迁移，比方说生物病毒传播到计算机病毒传播的迁移。

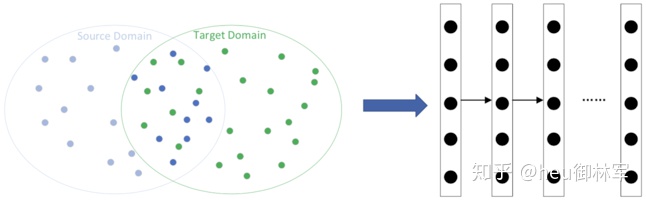


**迁移学习的分类——深度迁移学习的分类**

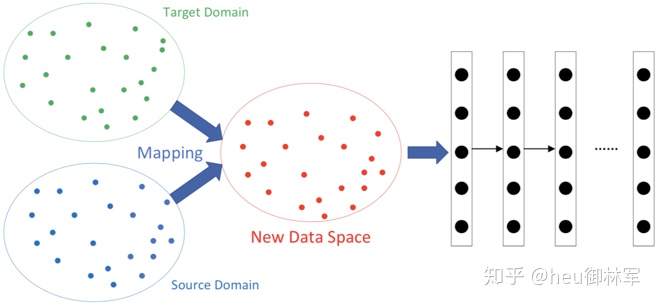


深度迁移学习分为四类：

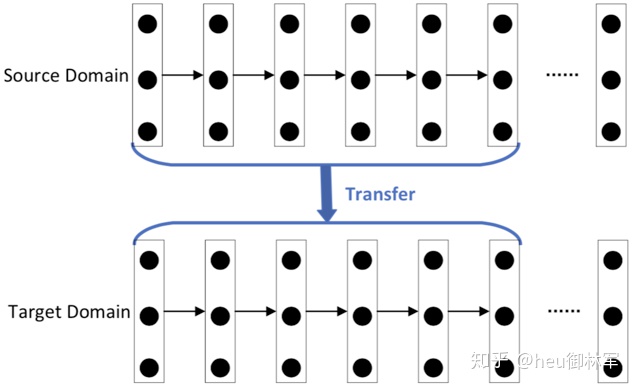
* 基于实例的深度迁移学习



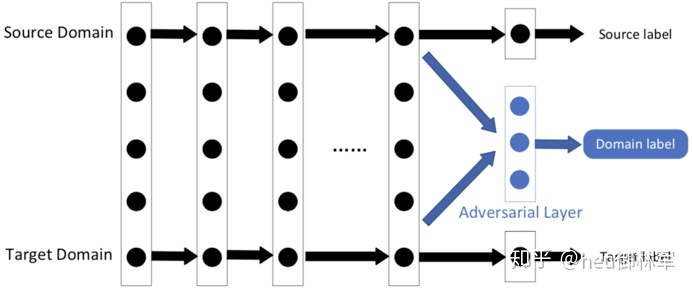
* 基于映射的深度迁移学习



* 基于网络的深度迁移学习



* 基于对抗的深度迁移学习

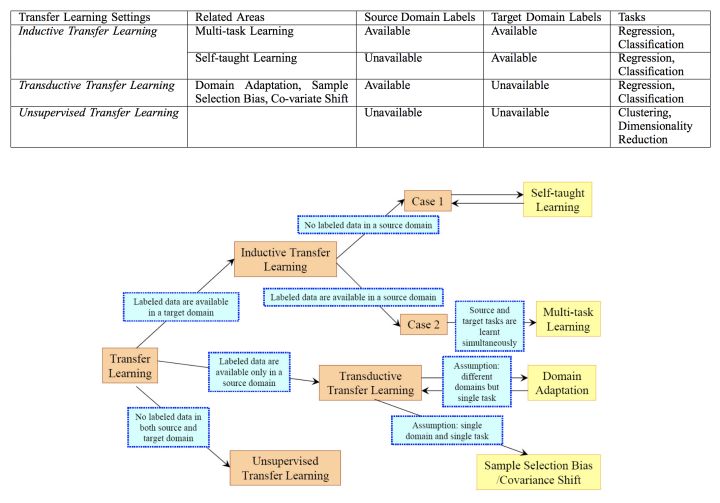


基于对抗性的深度迁移学习是指在生成对抗性网络(GAN)的启发下，引入对抗性技术，寻找既适用于源域又适用于目标域的可迁移表达。它基于这样的假设:“为了有效的迁移，良好的表征应该是对主要学习任务的区别性，以及对源域和目标域的不加区分。”

**迁移学习的分类——按特征空间分类**

1. 同构迁移学习（Homogeneous TL）: 源域和目标域的特征空间相同，XS=XT
2. 异构迁移学习（Heterogeneous TL）：源域和目标域的特征空间不同，XS≠XT

**迁移学习的分类——按迁移情景分类**



**Inductive Transfer Learning**

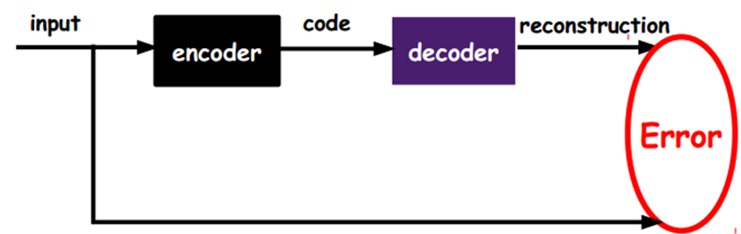
Inductive 的语义解释为归纳，直接理解为在Source Domain中学习适用于Target Domain的网络特征。在inductive的迁移学习里，我们有 Target Domain 的 Ground Truth Label，这就意味着Source Domain将直接对Target Domain的学习提供指导作用

**Inductive Transfer Learning——Self-Taught Learning**

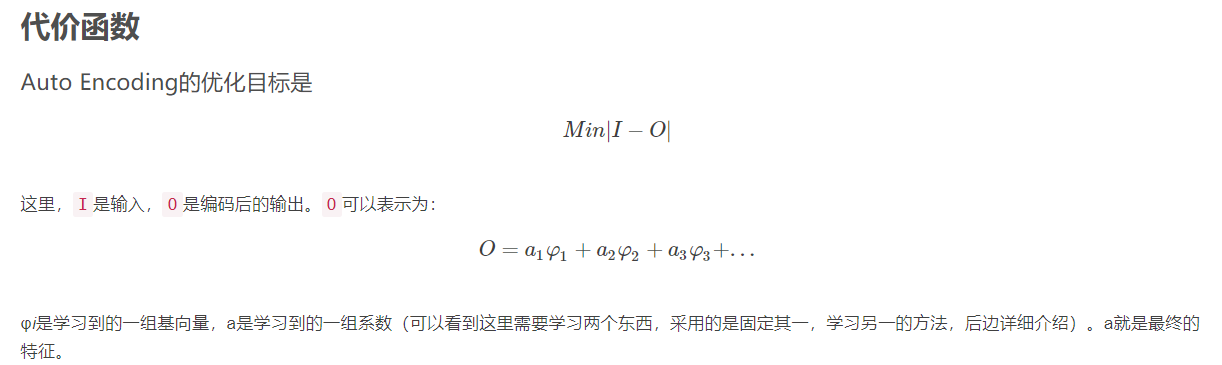
原作者引用的文献来看，他们先从网络上下载大量的图片，通过Sparse Coding的方式学习通用普世的图片特征，再直接从中找给定的任务比如图片分类所用的图片特征来做迁移学习。

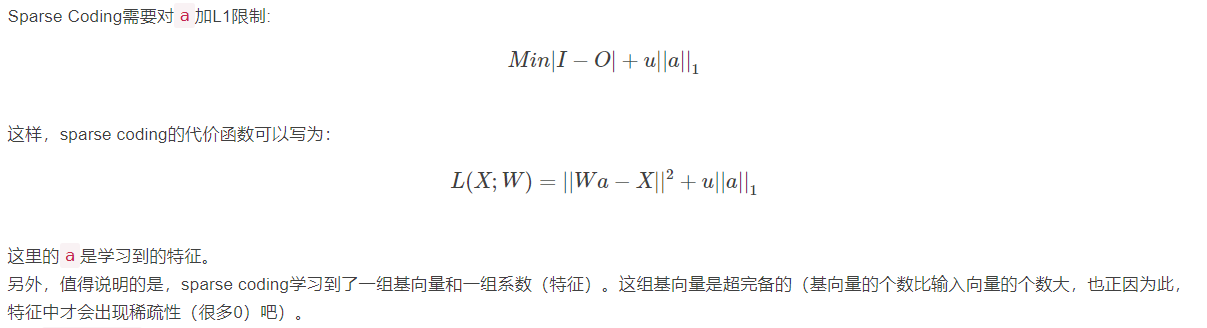
**sparse coding**

Sparse Coding**（稀疏编码）**是一种无监督学习方法，本质是找到原始输入的一个稀疏表达，也就是用一个稀疏的特征向量来表达原始输入数据。稀疏编码与自编码的结构很相似，自编码网络：

****

通过enconde和decode两个过程，多次优化，找到原输入的一个编码表示（图中的code）。稀疏编码也是如此，不过就是在自编码的基础上加了L1正则限制（约束向量的元素大部分为0）





1.Training

给定一系列的输入x1, x2, x3…, xn需要学习到一组基向量φ1，φ2，φ­…

训练过程是一个重复迭代的过程，交替更新a和φ，使代价函数最小：

step1：固定φ，更新a，这是一个LASSO问题

step2：固定a，更新φ，这是一个QP问题

2.测试

训练已经完成，给定一个新的数据X，通过解一个LASSO问题即可得到稀疏向量a。

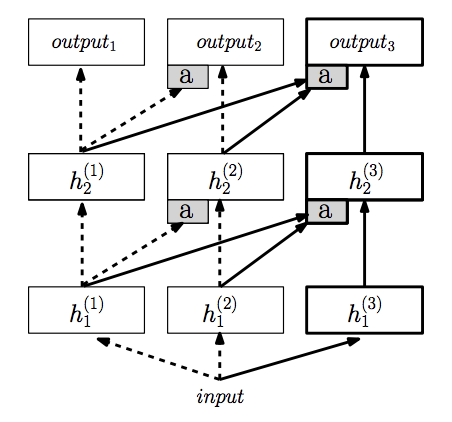
**Catastrophic Forgetting**

神经网络训练局限于一个性质叫Catastrophic Forgetting，意思是如果将pre-trained 的模型作为初始值训练一个全新的任务，模型之前学习到的特征将灾难性遗忘，也意味着之前网络训练的任务结果将非常糟糕。由于这一的性质，所有单一基于最原始的深度学习的任务看上去都像暴力求解，只适用于特定任务，无法得到像人类一样通向普世的学习特征。

**Progressive Neural Networks**

Progressive Neural Networks通过 Lateral Connection 的方式将一学习好的模型参数通过另一层网络教给一个新的任务。由于在网络设计中，已训练完成的模型将会被Freeze，因此在 back-propagation 优化参数的方式并不影响已学习完成的网络。这种网络设计很自然的避免了 Catastrophic Forgetting 的出现，但也引出了一些新的问题：

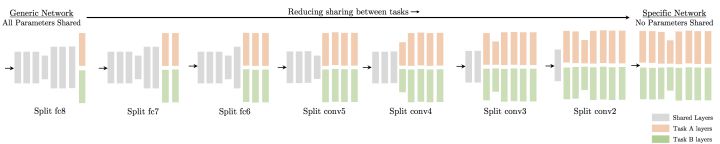
* 不断的加入新的任务会引出新的网络分支，以及 lateral connection 分支，这种学习效率显然是很低的，因为我们一次只能学一个任务。
* 我们无法确定先学哪个任务将会给后续的任务有一个最好的结果。i.e. 我们不清楚任何关于任务的相关和层级信息，因此找到最优方法需要 n 个任务的排列也就是 n! 种方式，再一次突出了学习效率之低。



**Inductive Transfer Learning——Multi-Task Learning**

多任务学习与上述Self-Taught Learning最大的不同是我们需要把 Source 和 Target 两个或者更多任务同时学好。在多任务学习中，任一任务的训练目标都是同等重要的。因此，这里 Source/Target Domain 的分类姑且没有太大意义。

最简单直接的多任务学习网络就是直接应用一个普通的网络结构在最后输出层前分叉到不同的任务预测层。



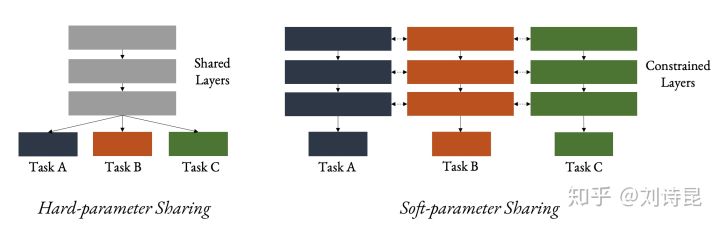
Cross-stitch Networks for Multi-task Learning 这篇文章就直接暴力了测试了所有分叉的可能性，并指出不同的任务在不同的分叉上效果。后续他们提出了一个叫 Cross-Stitch Network 的网络结构，通过矩阵中 linear combination 的方式来融合不同任务里学到的特征。

**多任务学习的问题：**

除了网络设计，还有如何寻找训练权重？

由于任务难度和数据集不同的缘故，在进行多任务学习中，有一种情况是网络中学到的特征被某一任务 dominate。一个简单的例子：我们同时训练 MNIST 和 ImageNet。由于我们知道MNIST是一个极其简单的数据集，而ImageNet是一个数据量大且复杂的多的数据集。那么模型训练将飞快的收敛 MNIST 的识别任务，且慢慢变成是基于 MNIST 特征网络的初始值的 ImageNet 训练。

**网络设计**

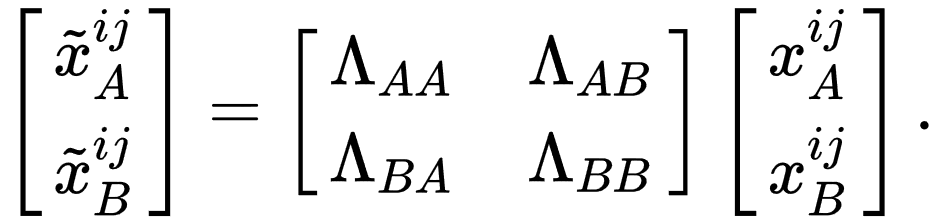
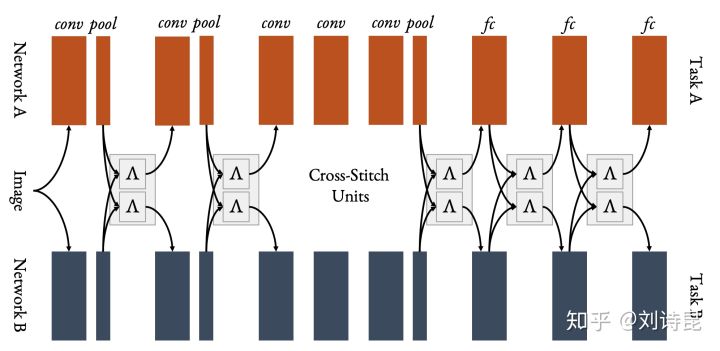


任一 MTL 网络设计可以看做是找 hard 和 soft parameter sharing 的平衡点：

1. 如何网络设计可以小巧轻便。

2. 如何网络设计可以最大幅度的让不同任务去共享信息。

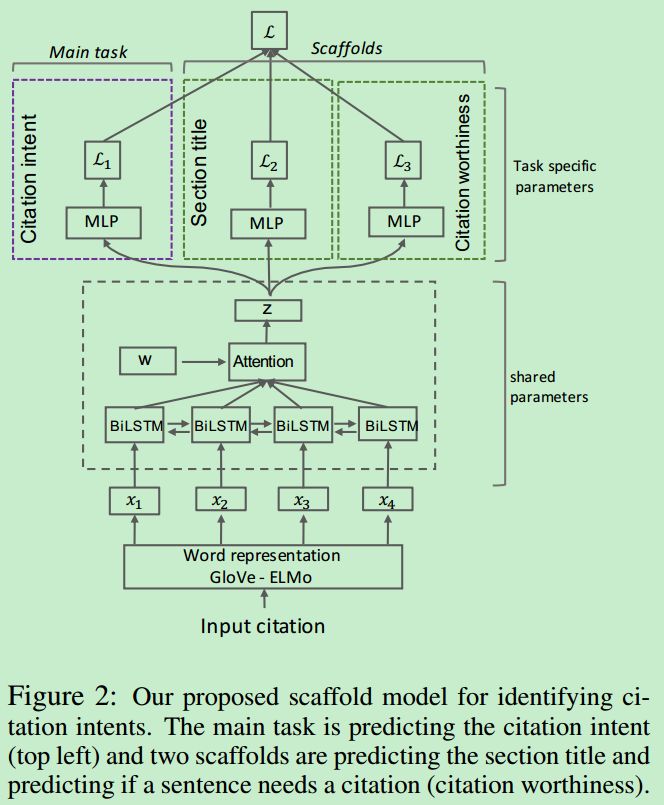
**Cross-Stitch Network**



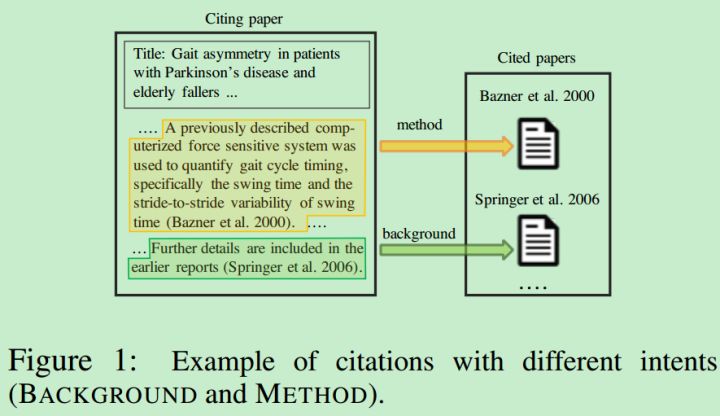
目前的讨论：

* Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics - 这篇文章就探讨了这个问题，提出了同样基于 Bayesian 的 maximise log-likelihood 来优化 loss function。
* GRADNORM: GRADIENT NORMALIZATION FOR ADAPTIVE LOSS BALANCING IN DEEP MULTITASK NETWORKS - 在最新的 ICLR 2017 会议中也出现了一篇文章讨论这个问题。他们的方法更偏向于解决：如何数值定义什么是一个 easy task。文章的观点是在每个training step，如果你的 training loss 相比较 step 0 的 training loss 越小，那么就越可能是 easy step。 以此通过计算 relative training rate 来更新每个任务的权重。

**辅助学习**



**Citation Intent**



**Citation worthiness**



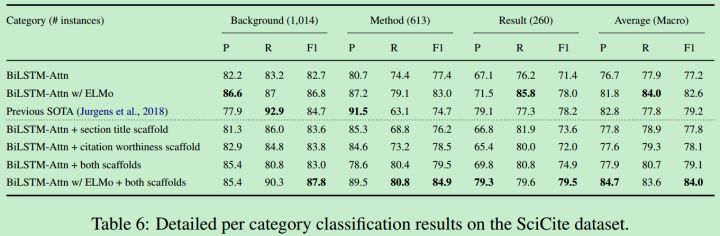
判断输入句子是否需要引用，是一个二分类任务

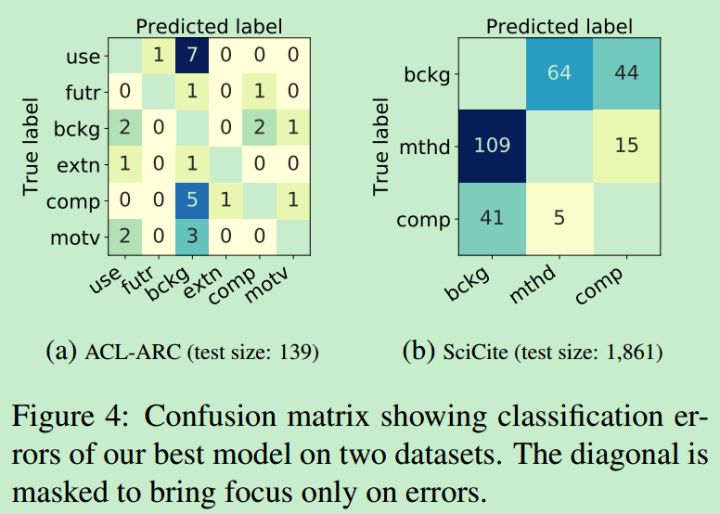
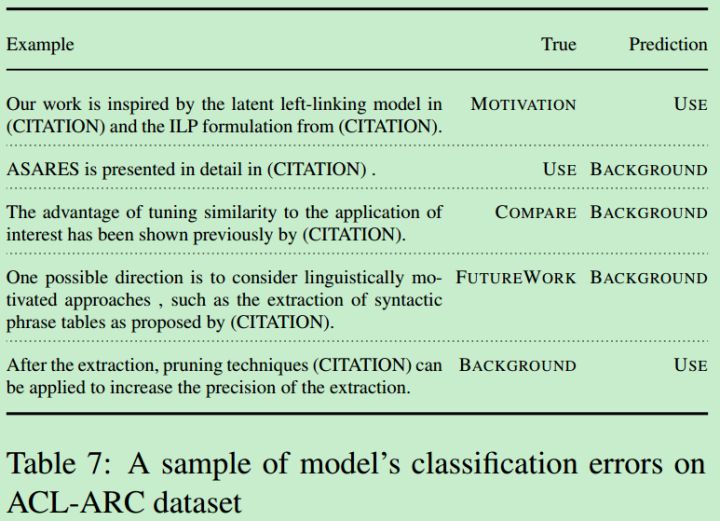
**Section title**

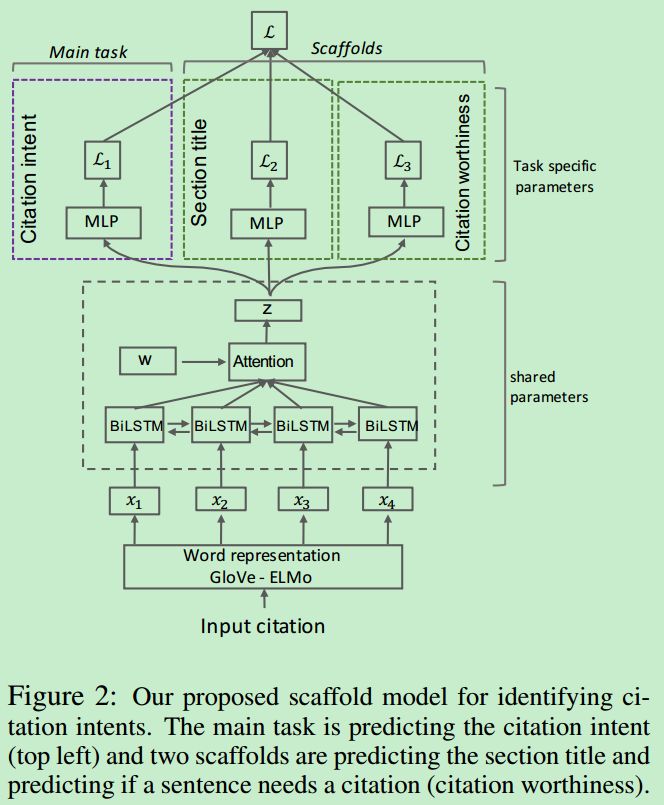
预测引文的章节标题。

作者使用正则表达式对section titles 进行标准化为“introduction”, “related work”, “method”, “experiments”, “conclusion”这五个部分

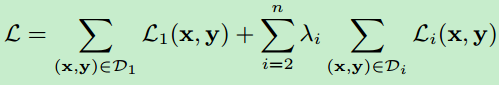
**分析**

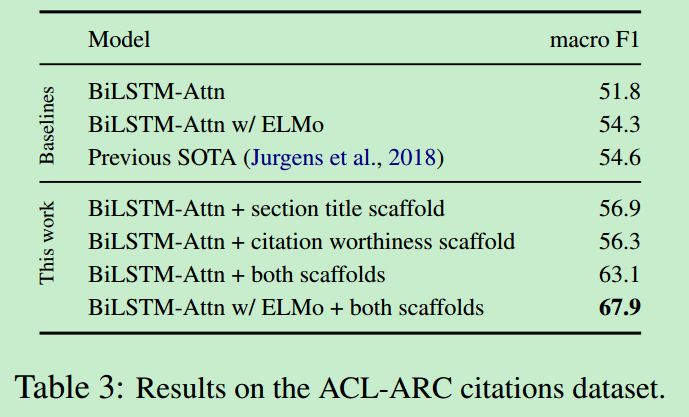












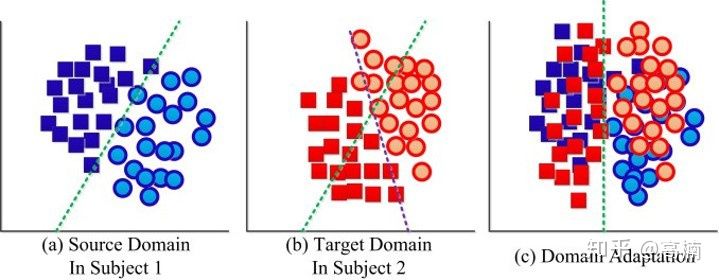
**Transductive Transfer Learning**

Transductive的字面意思是转导，Transductive Transfer Learning翻译是直推式迁移学习，通常来说要比 inductive transfer learning 要难的多。由于直接缺乏 Target Domain Label，在这个情况，我们只能依赖于 Source Domain 信息来尽可能的训练出泛化能力强的特征。

**Transductive Transfer Learning——**

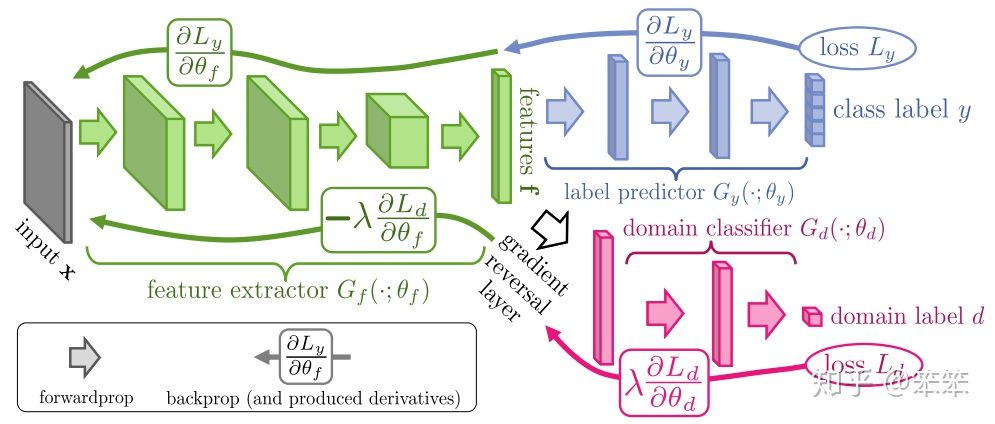
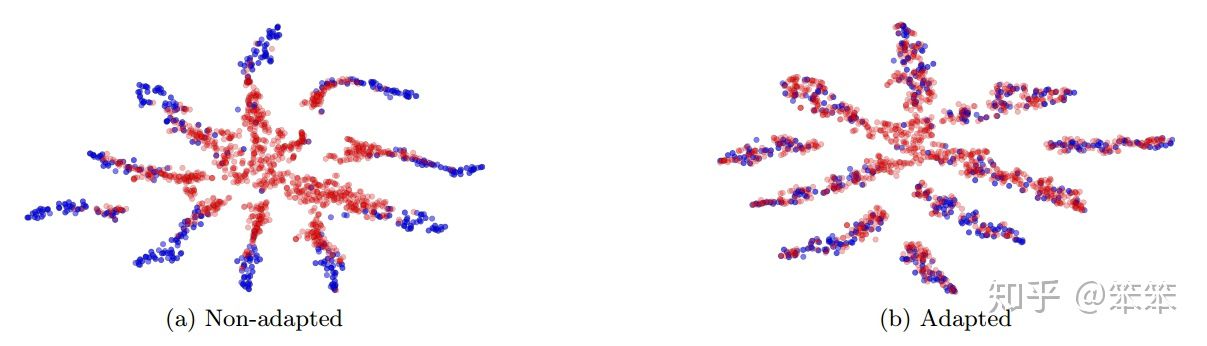
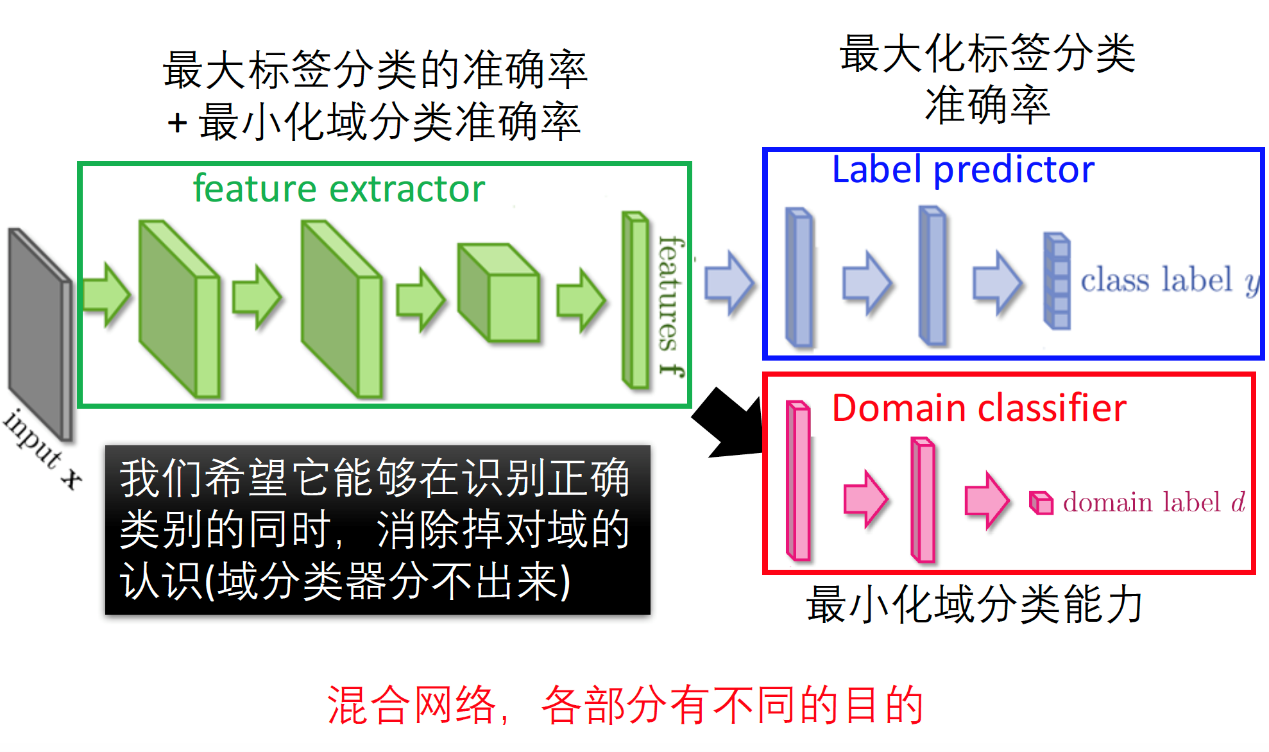
**Domain Adaptation / Generalisation**

Transfer Learning的初衷是节省人工标注样本的时间，让模型可以通过已有的标记数据（source domain data）向未标记数据（target domain data）迁移。从而训练出适用于target domain的模型。Domain Adaptation（域适配）目标是学习 domain-invariant feature 使得学习到的特征不受限于 Source Domain 而导致 over-fitting。当然，评估模型的时候目标域的数据是要有标签的。



***回顾上述深度迁移学习的分类：基于对抗的深度迁移学习***

**Domain-Adversarial Training of Neural Networks(DANN)**

目标：通过调整feature把Domain标签逐步消除，得到一个通用的特征，该特征既在原始黑白的MNIST数据集上有良好效果，又在含有干扰的MNIST-M数据集上表现良好。

该网络架构分为三个部分

**特征提取器**（feature extractor）：特征提取器的功能包括两部分：（1）提取后续网络完成任务所需要的特征。（2）将源域样本和目标域样本进行映射和混合

**类别预测器**(Label predictor)：用于预测数字图片所属类别（0-9）

**域分类器**(Domain classifier)：用于预测数字图片来自哪个域（MNIST或者MNIST-M）

**梯度反转层（Gradient reversal layer）**

在反向传播更新参数的过程中，域分类器和特征提取器中间有一个梯度反转层（Gradient reversal layer），也就是说域分类器的训练目标是尽量将输入的信息分到正确的域类别（源域还是目标域），而特征提取器的训练目标却恰恰相反（由于梯度反转层的存在），特征提取器所提取的特征（或者说映射的结果）目的是是让域判别器不能正确的判断出信息来自哪一个域，因此形成一种对抗关系。可见，当域分类器不能将接收的信息正确分为源域样本还是目标域样本时，特征提取器的任务就圆满完成了，因为此时源域样本和目标域样本在某个空间中已经被混合在一起不能分开了。

然而上述过程存在一个问题，我们最终的目的是对目标域样本进行分类，那么我们如何保证特征提取器提取的信息是能够用来分类的呢？假如无论输入什么样本给特征提取器，它都输出一个单位向量，这样依旧可以“骗过”域分类器，但是却无法完成后续的分类工作。

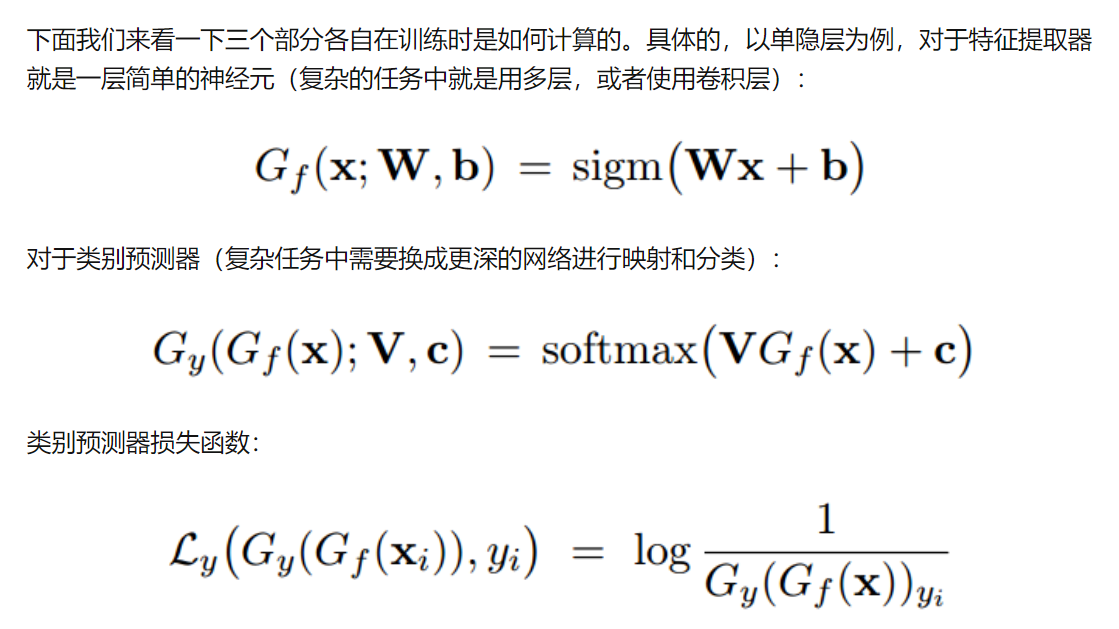
这时就要靠Label predictor (类别预测器)了，因为源域样本是有标记的，所以在提取特征时不仅仅要考虑后面的域判别器的情况，还要利用源域的带标记样本进行有监督训练从而兼顾分类的准确性。

**总体来看**

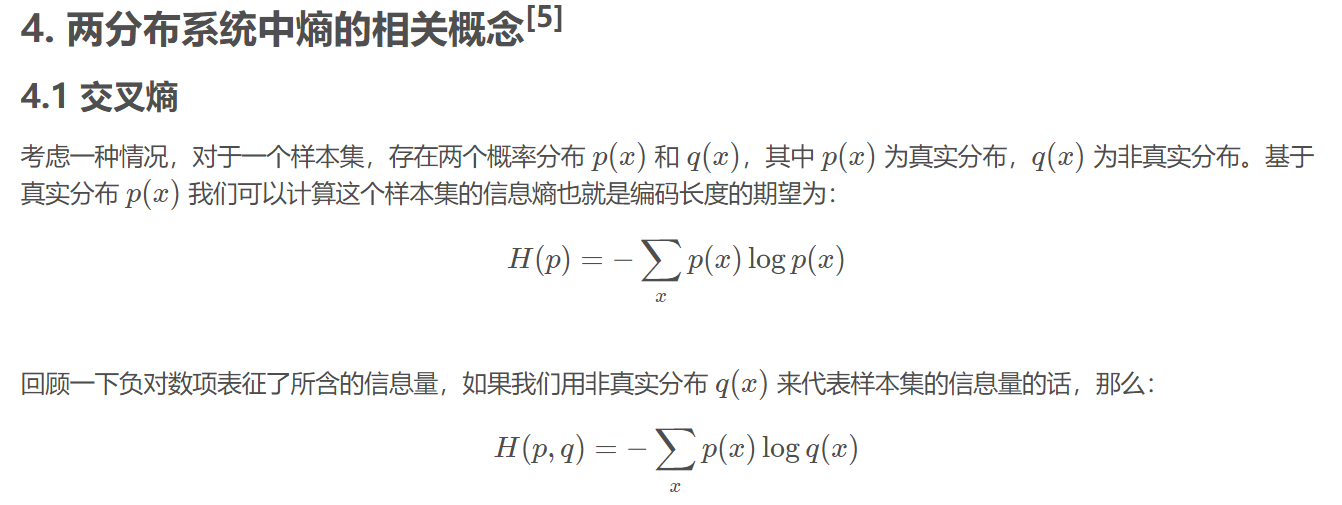
我们希望它能够在识别正确类别的同时，消除掉对域的认识（域分类器分不出来），既要最优化Label predictor，又要在Domain classifier很“努力”的前提下，最差化Domain classifier，或者说是**反向调优Domain classifier**。

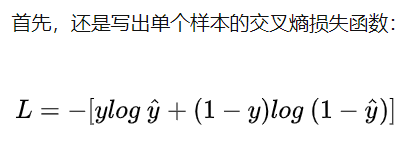
其中在通过Label predictor时，由于需要做label预测，仅需要MNIST数据集。而在通过Domain classifier时，是需要消除域标签的，所以需要MNIST和MNIST-M数据集的共同参与。

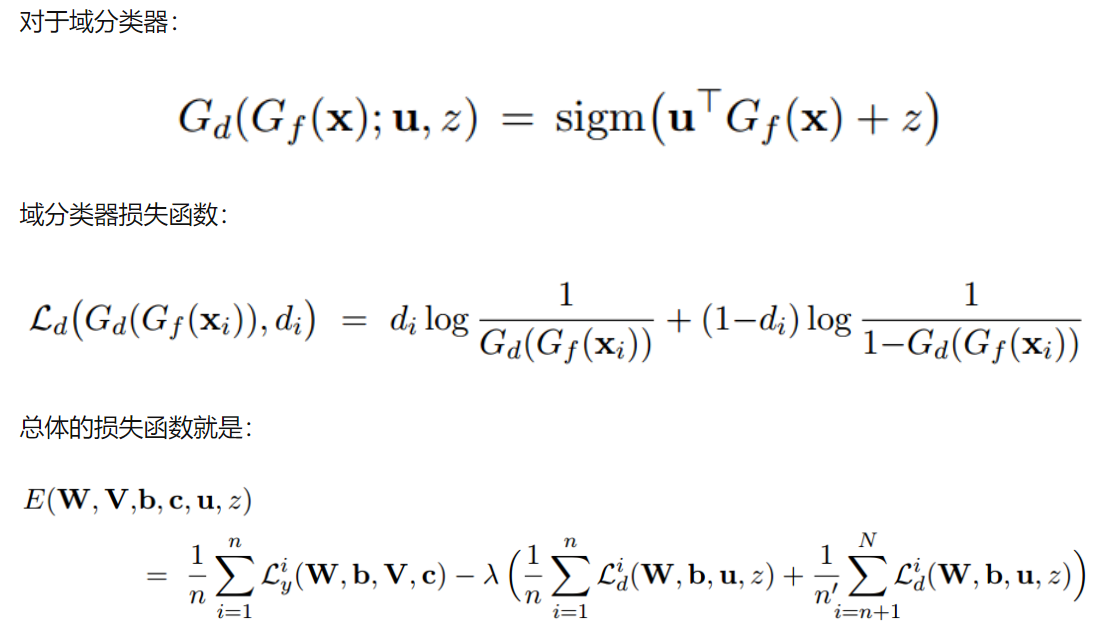
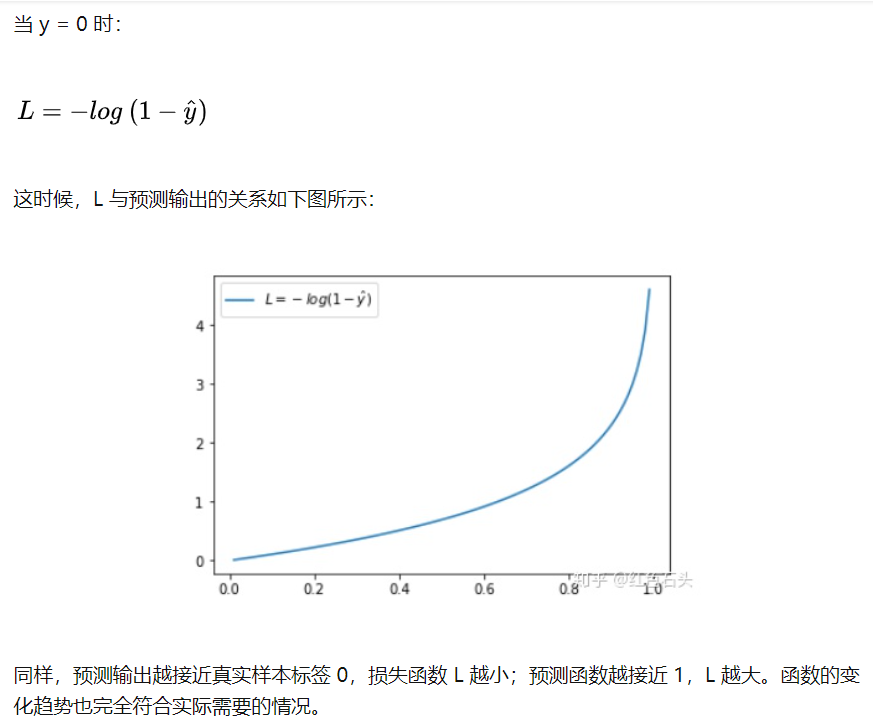
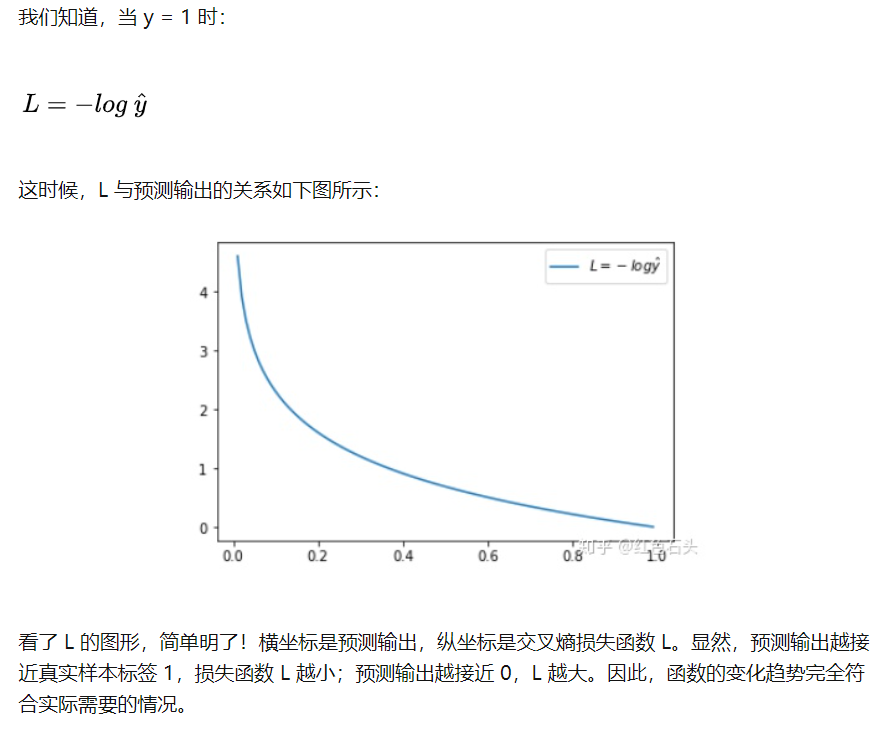
**损失函数**

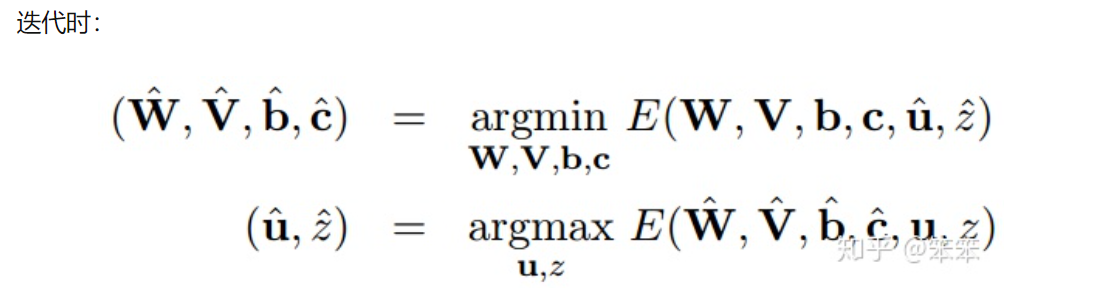


**交叉熵损失函数**









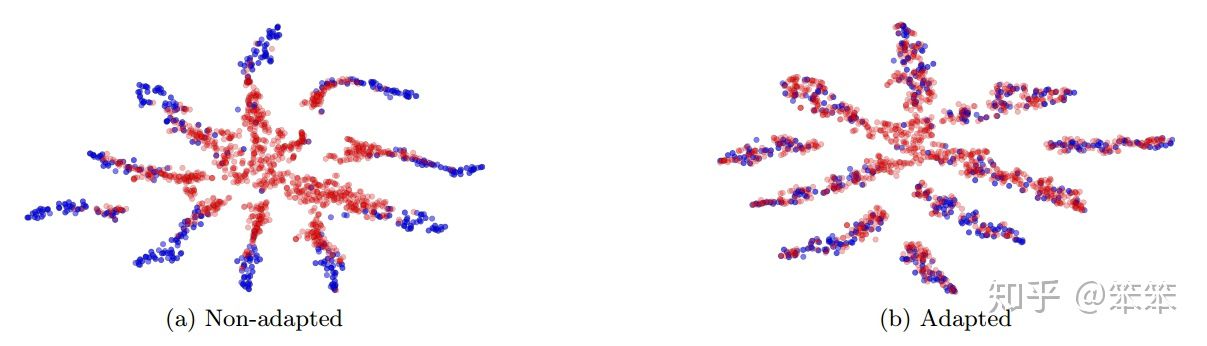
Backpropagation中需要注意的地方

针对Label Predictor：使用SGD优化feature extractor，即上述迭代时式子的第一行

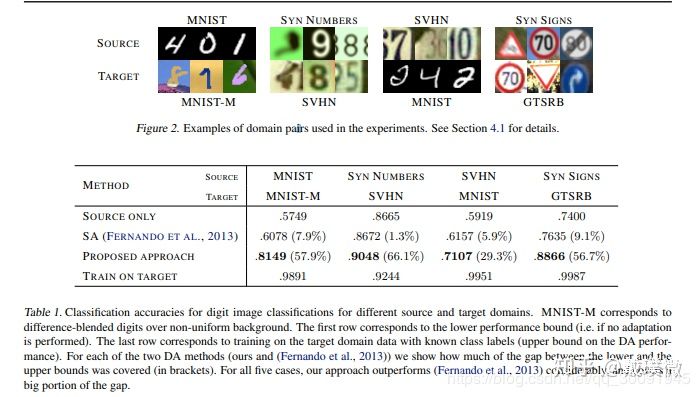
针对Domain Classifier：先SGD优化Domain局部网络，再反向SGD优化feature extractor，即上述迭代时式子的第二行

总损失函数中的 **λ** 为超参数

**结果**

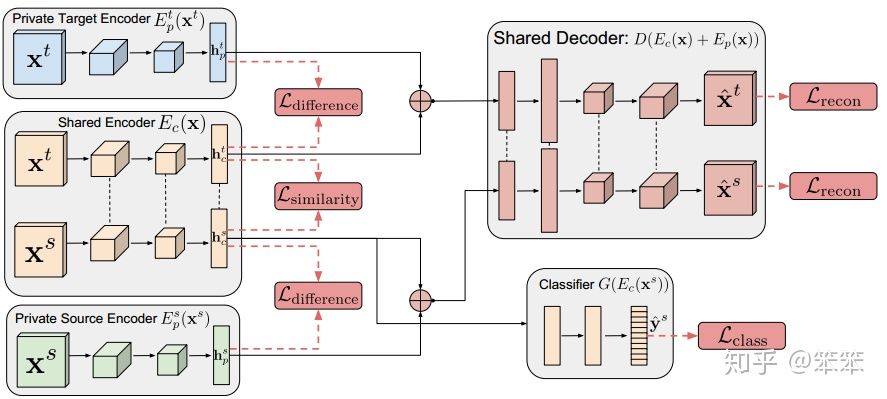


MNIST和MNIST-M都是十个类别，在图4 的（a）中，两个数据集还没有进行域适配，可以看出虽然各个类别之间是独立的，但是源域样本（蓝色点）和目标域样本（红色点）却是分开的，没有办法对没有标记的目标域进行分类。在图4的（b）中，经过域适配之后，各个类别不仅是独立的，而且源域样本和目标域样本在该空间中还是混合在一起的，这样就可以实现同时分类了。

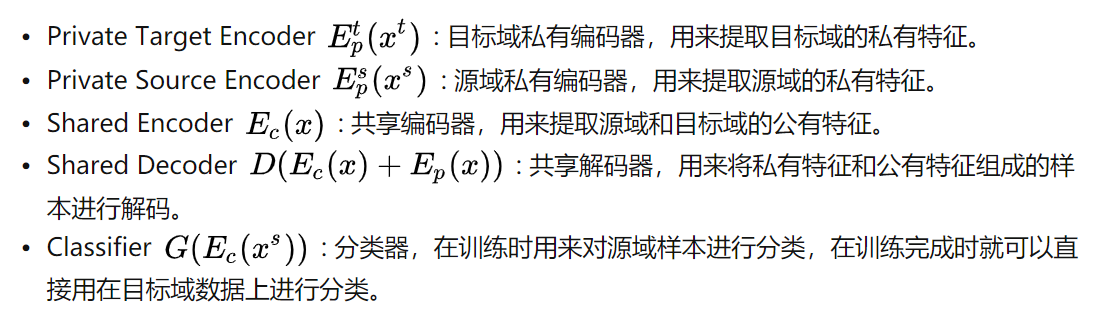


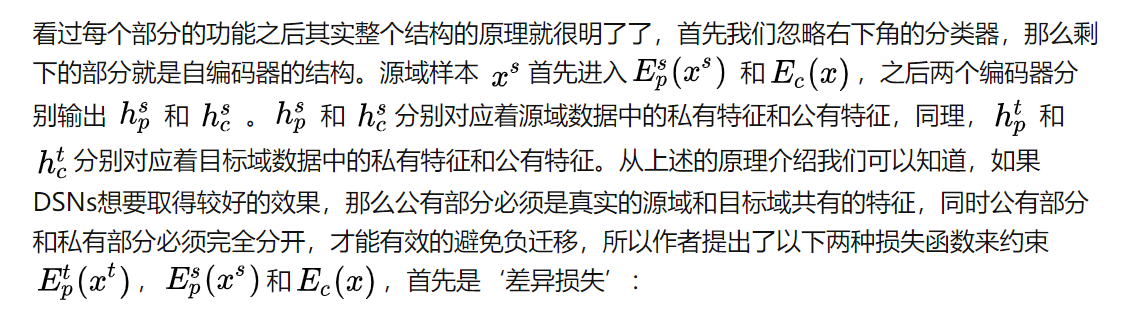
**Domain Separation Networks(DSNs)**

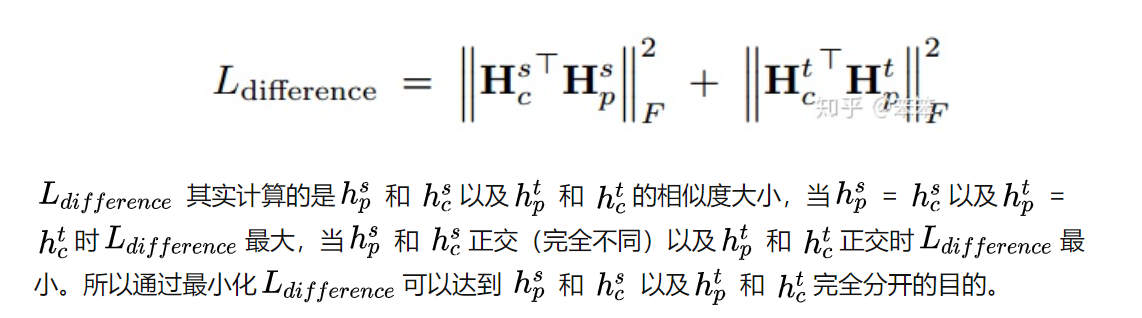
作者等人从另一个角度来考虑了迁移学习问题，他们认为所有的域之间有着公有的特征(Shared)和私有的特征(Private)，如果将各个域的私有特征也进行迁移的话就会造成负迁移(negative transfer)。基于这一理念，他们提出了Domain Separation Networks(DSNs)。同时，作者为问题假设的前提是无监督迁移学习，也就是说目标域数据是没有标记的，源域数据有标记（Domain Adaptation）。

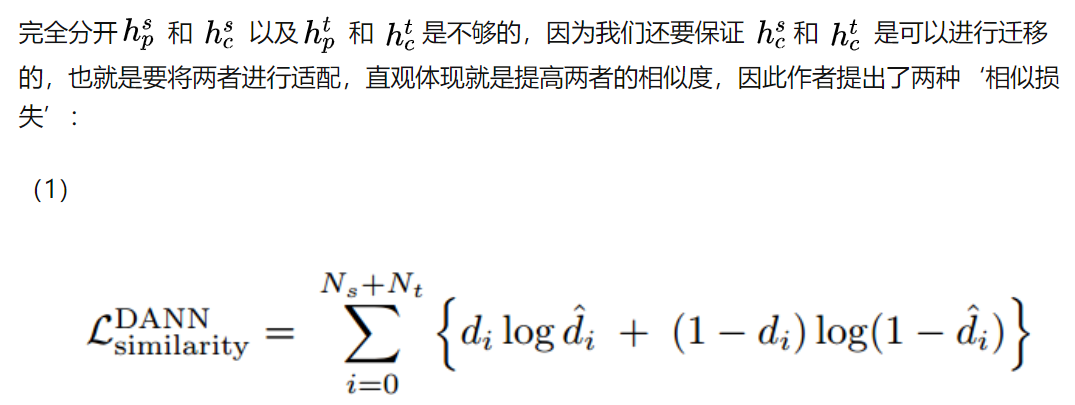


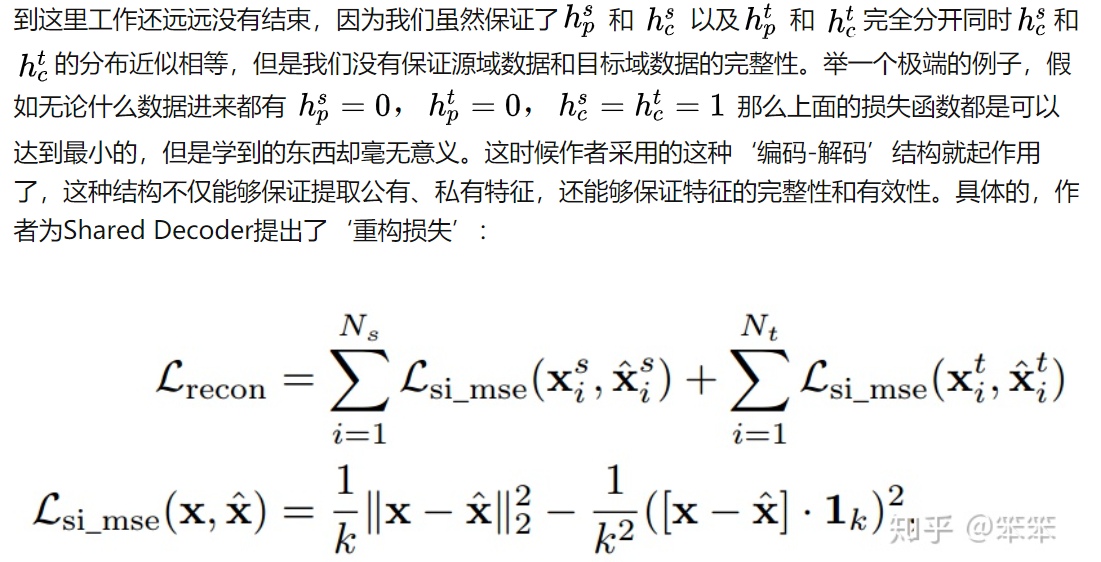
DSNs的主体结构是一个类似于自编码器的结构。整个结构可以分为如下五部分：

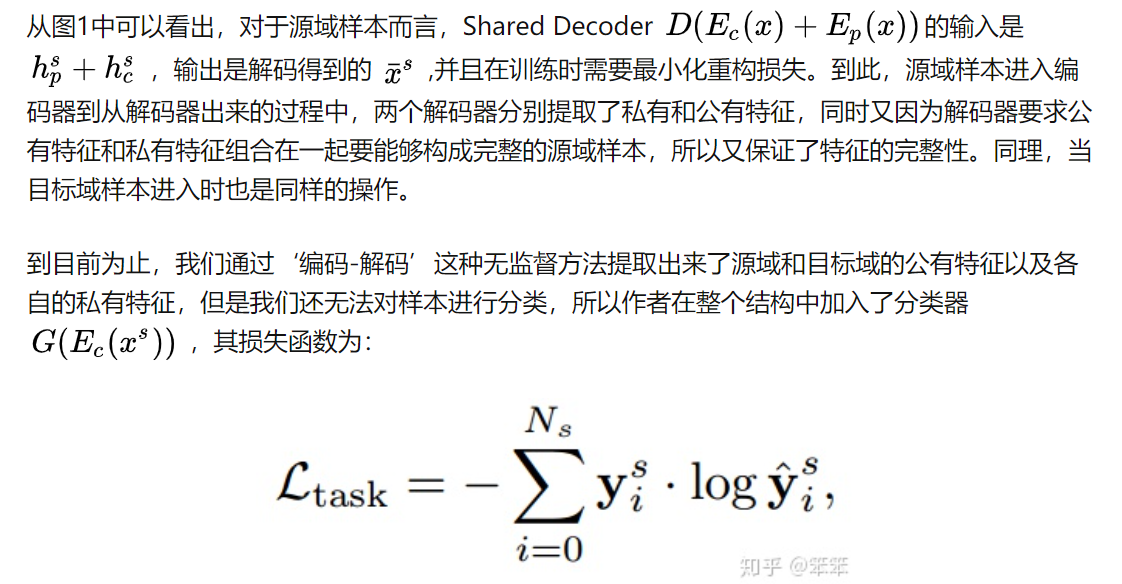


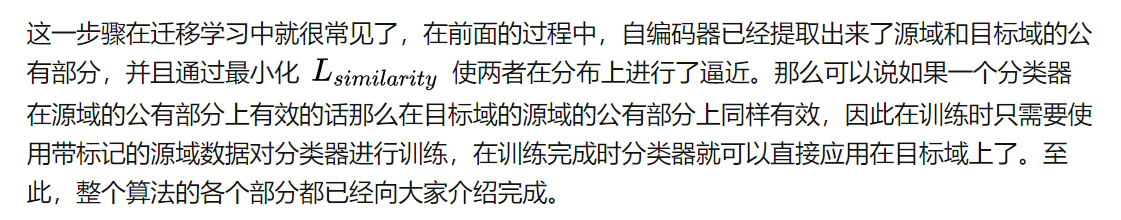


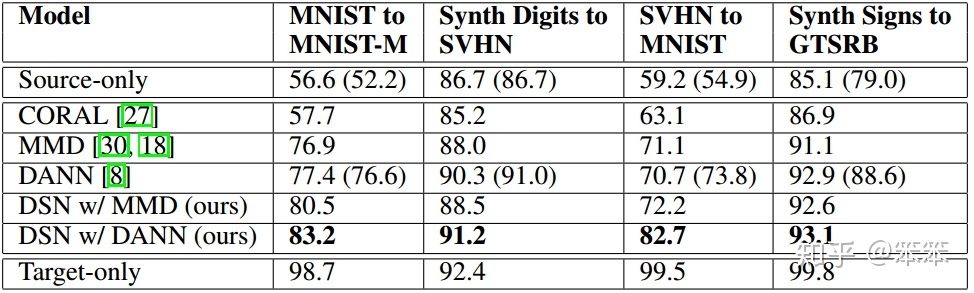












**Transductive Transfer Learning——**

**Unsupervised Transfer Learning**

在这个方向可适用于的任务非常有限，比如 Dimension Reduction。由于缺乏 Source Label 因此也无法采用绝大部分的 Deep Learning 方法，（但比如Auto-encoder 是其中一个基于 Deep Learning 的降维方式，但很难归类于 Transfer Learning）。

**Multi-Task Learning**