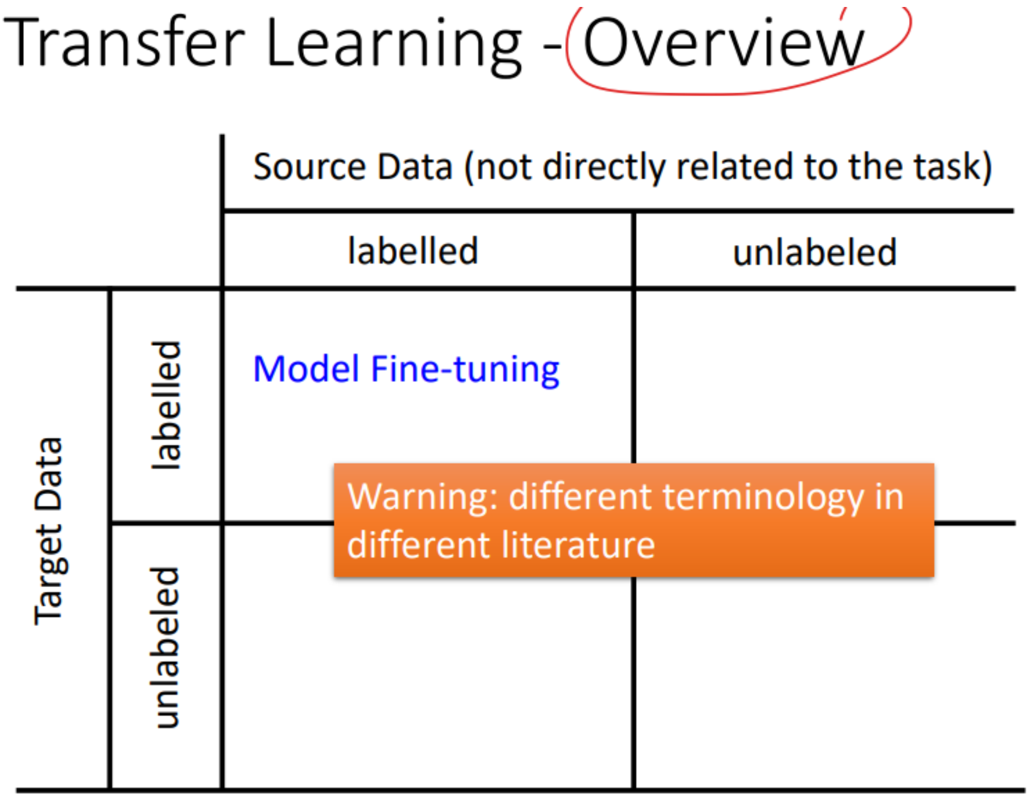
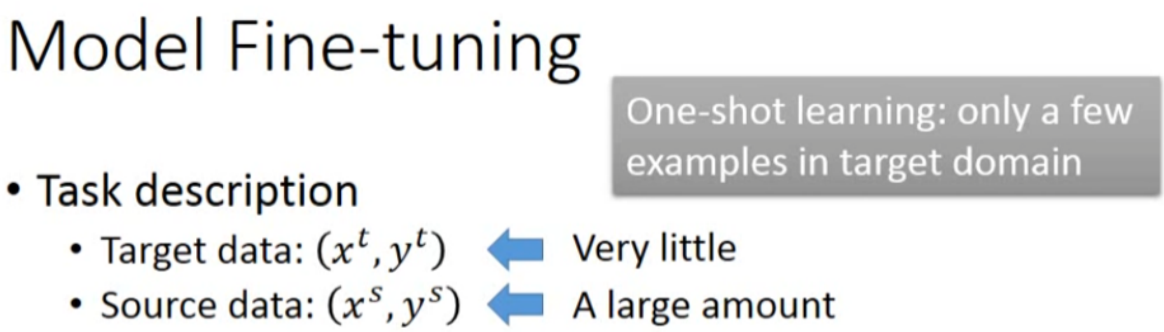
target data的量很少，因此无法直接训练模型，需要 transfer learning

我们现在有一些target data，是和我们的目标相关的，有一些source data，是和我们的目标无没有直接的关系。分成四个不同的象限来讨论（target data，source data都可能是有 label 或者没有 label的）



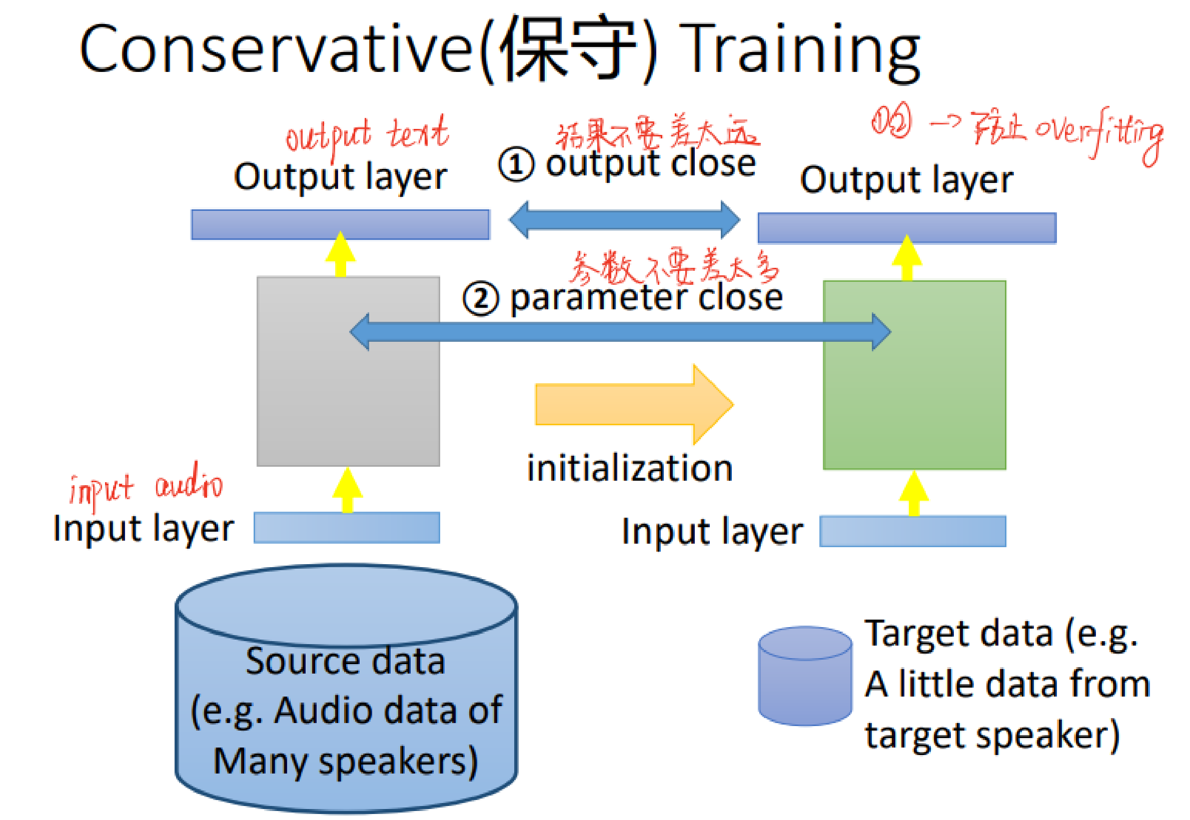
Fine-tuning



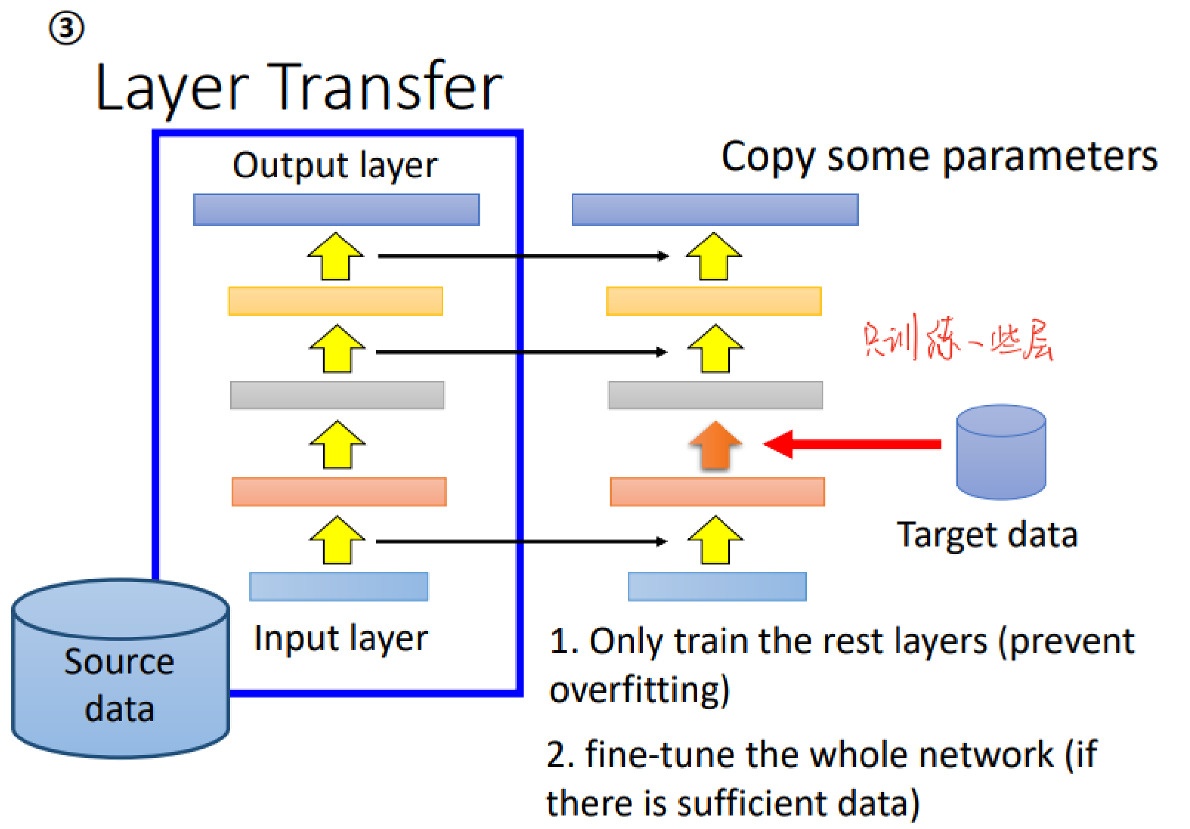
先用 source data预训练一个模型，作为target data 模型的初始值，然后在 target data 上进行训练（微调）

做一些限制，使得在 target data上训练的时候不要过拟合

保守训练



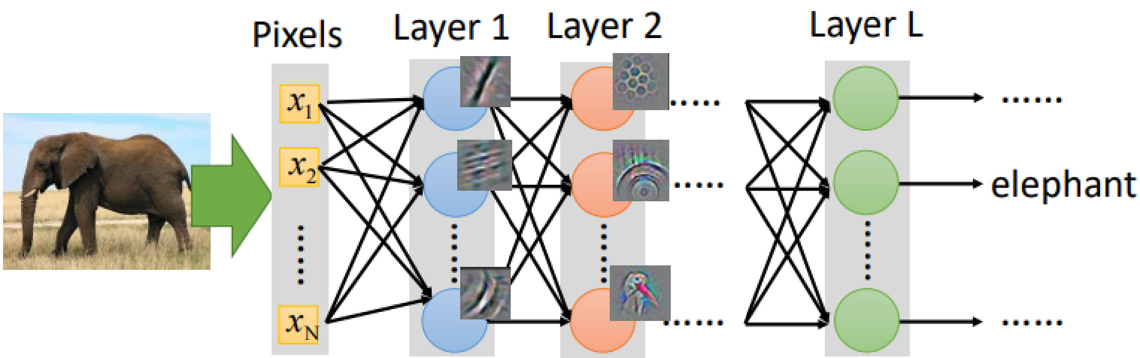
只训练一些层，其他层直接copy



不同的任务copy的层是不一样的

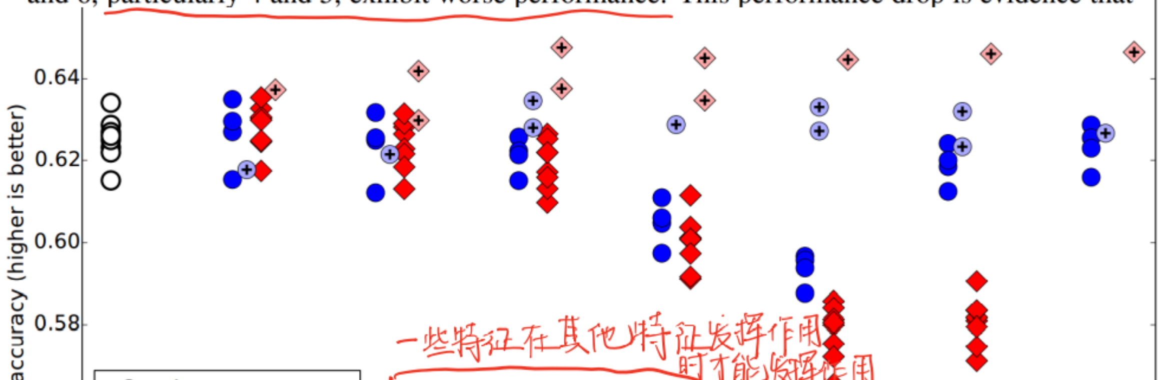
- Which layer can be transferred (copied)?
  - Speech: usually copy the last few layers
  - Image: usually copy the first few layers → 后面识别任务越来越具体。

识别纹理，线条等简单特征



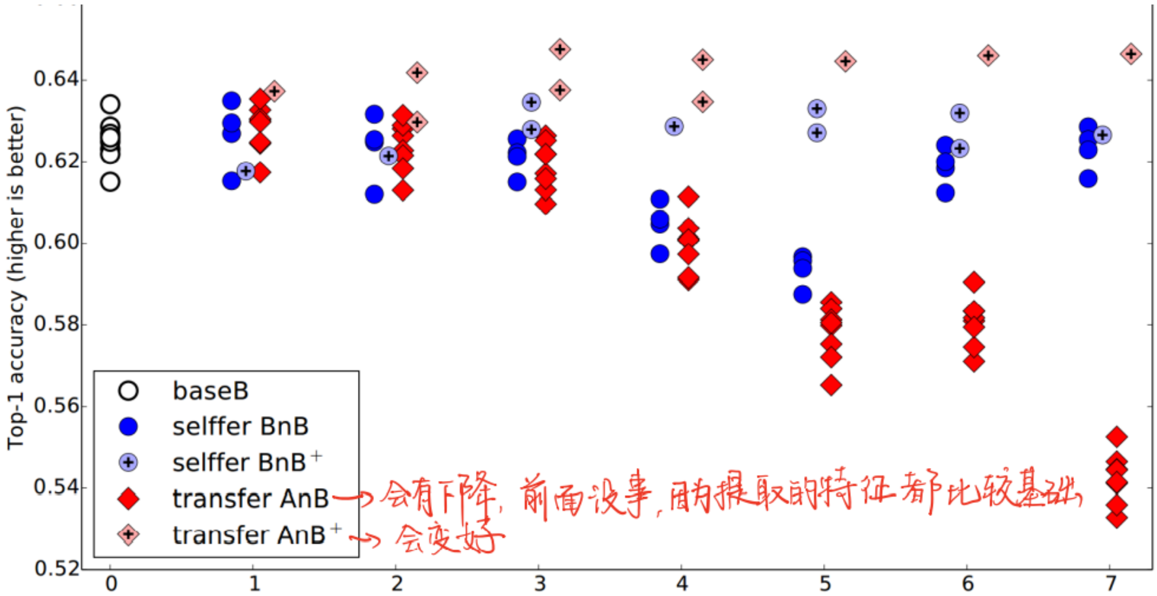
ImageNet上的实验（相似数据）

The dark blue BnB points show a curious behavior. As expected, performance at layer one is the same as the baseB points. That is, if we learn eight layers of features, save the first layer of learned Gabor features and color blobs, reinitialize the whole network, and retrain it toward the same task, it does just as well. This result also holds true for layer 2. However, layers 3, 4, 5, and 6, particularly 4 and 5, exhibit worse performance. This performance drop is evidence that

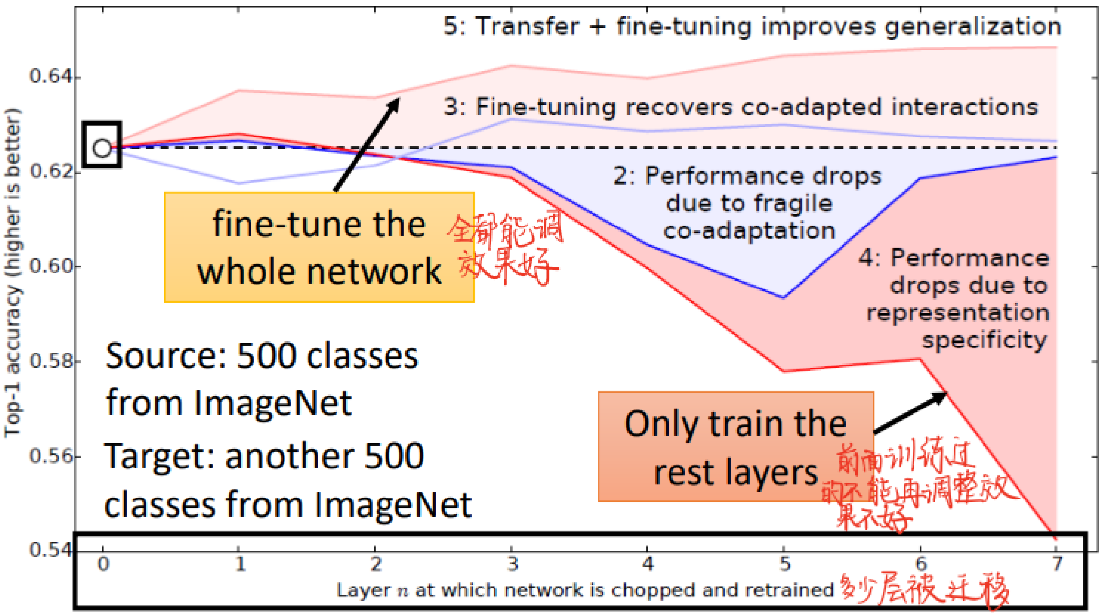


一些特征在其他特征发挥作用时才能发挥作用

the original network contained *fragile co-adapted features* on successive layers, that is, features that interact with each other in a complex or fragile way such that this co-adaptation *could not be relearned* by the upper layers alone. Gradient descent was able to find a good solution the first time, but this was only possible because the layers were jointly trained. By layer 6 performance is nearly back to the base level, as is layer 7. As we get closer and closer to the final, 500-way softmax output layer 8, there is less to relearn, and apparently relearning these one or two layers is simple enough for gradient descent to find a good solution. Alternately, we may say that there is less co-adaptation of features between layers 6 & 7 and between 7 & 8 than between previous layers. To our knowledge it has not been previously observed in the literature that such optimization difficulties may be worse in the middle of a network than near the bottom or top.



Layer Transfer - Image

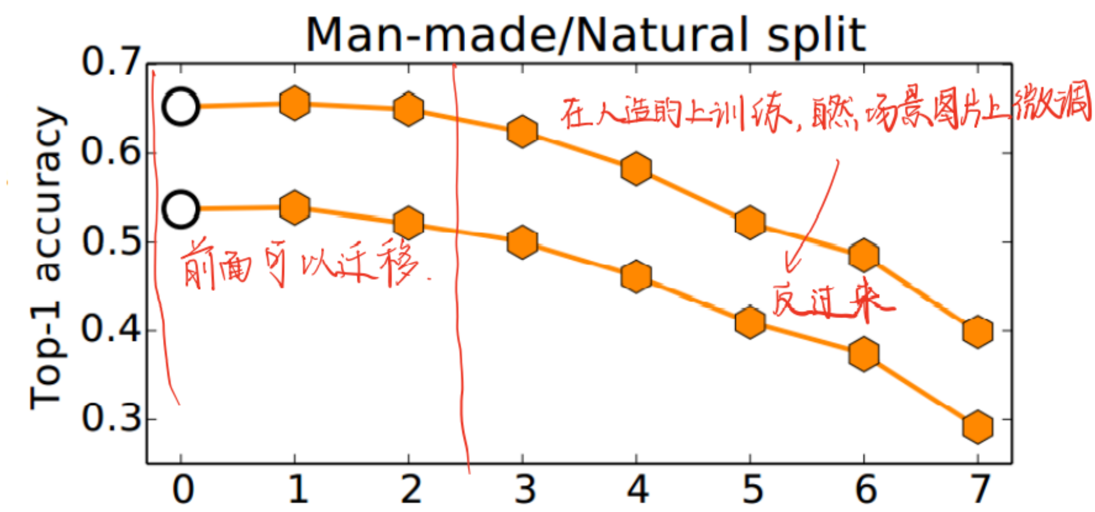


Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, "How transferable are features in deep neural networks?", NIPS, 2014

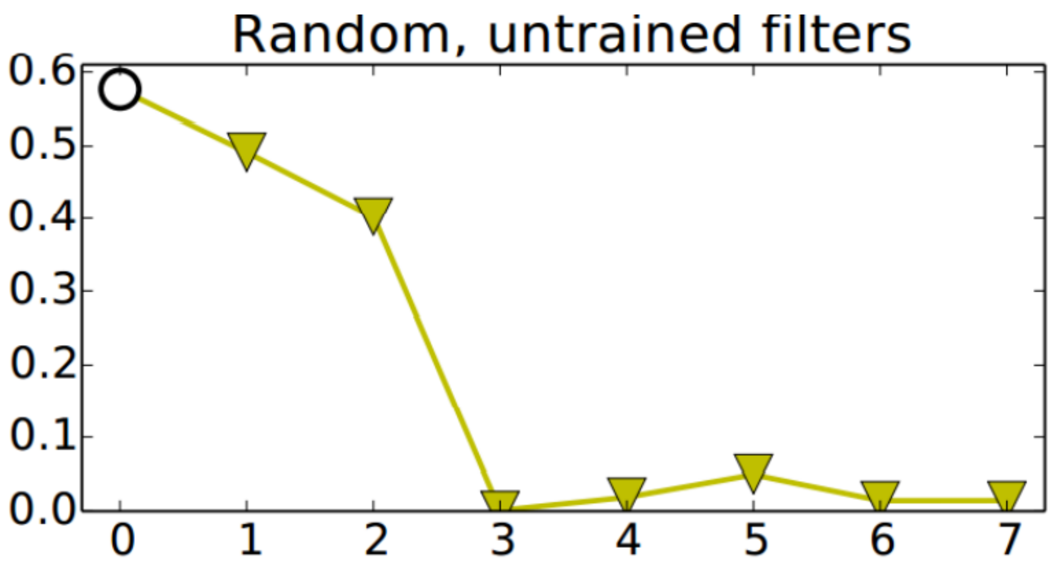
不相似的数据集上的实验

Contrasting split of dataset into manmade and natural images

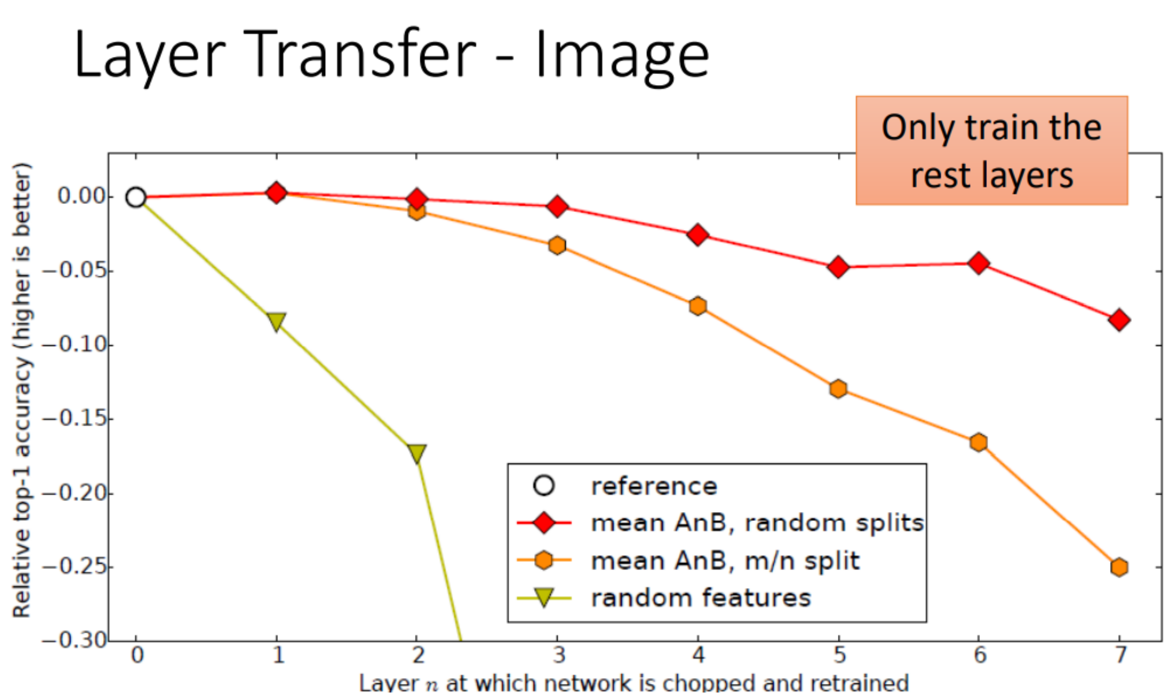
- Set A: manmade images. 551 classes. 人造场景
- Set B: natural images. 449 classes. 自然场景



小数据集上使用随机权重初始化

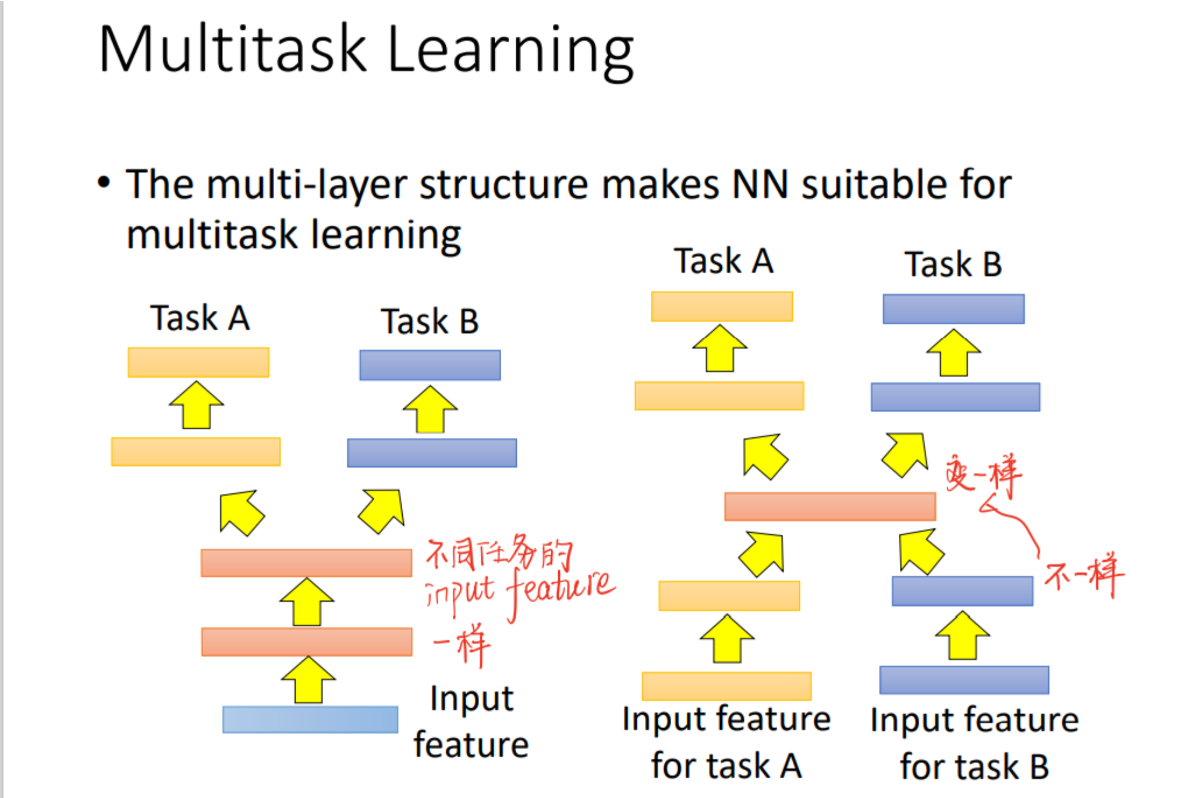


前面copy 过来的都不再进行调整

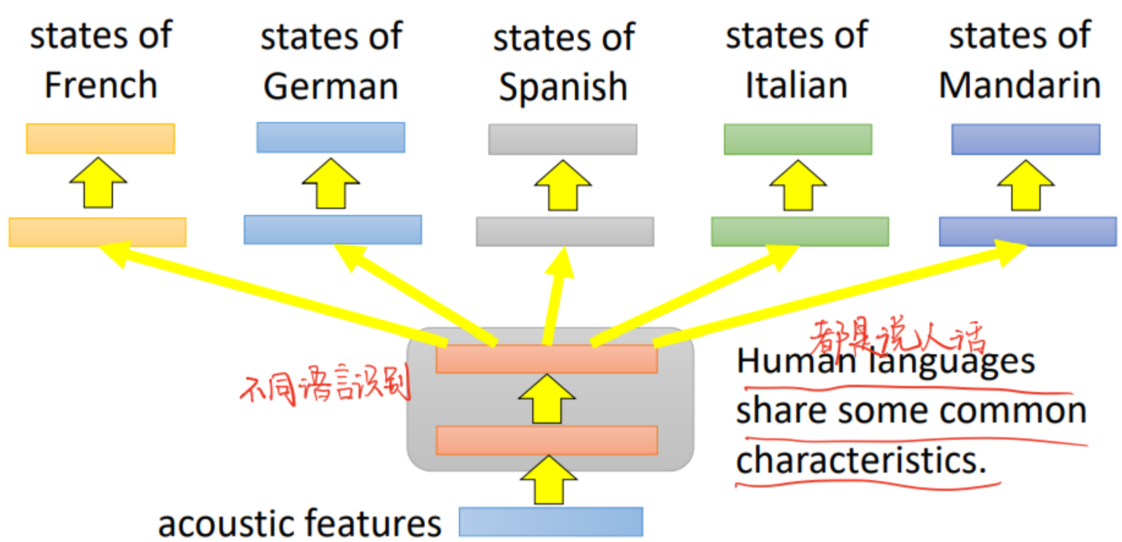


Multitask learning

- 不同任务的 input feature 一样，我们可以共用前面几层
- 不同任务的 input feature 不一样，我们可以先将两个 domain 转成一样的，然后再做后面的，说不定可以有共用的layer

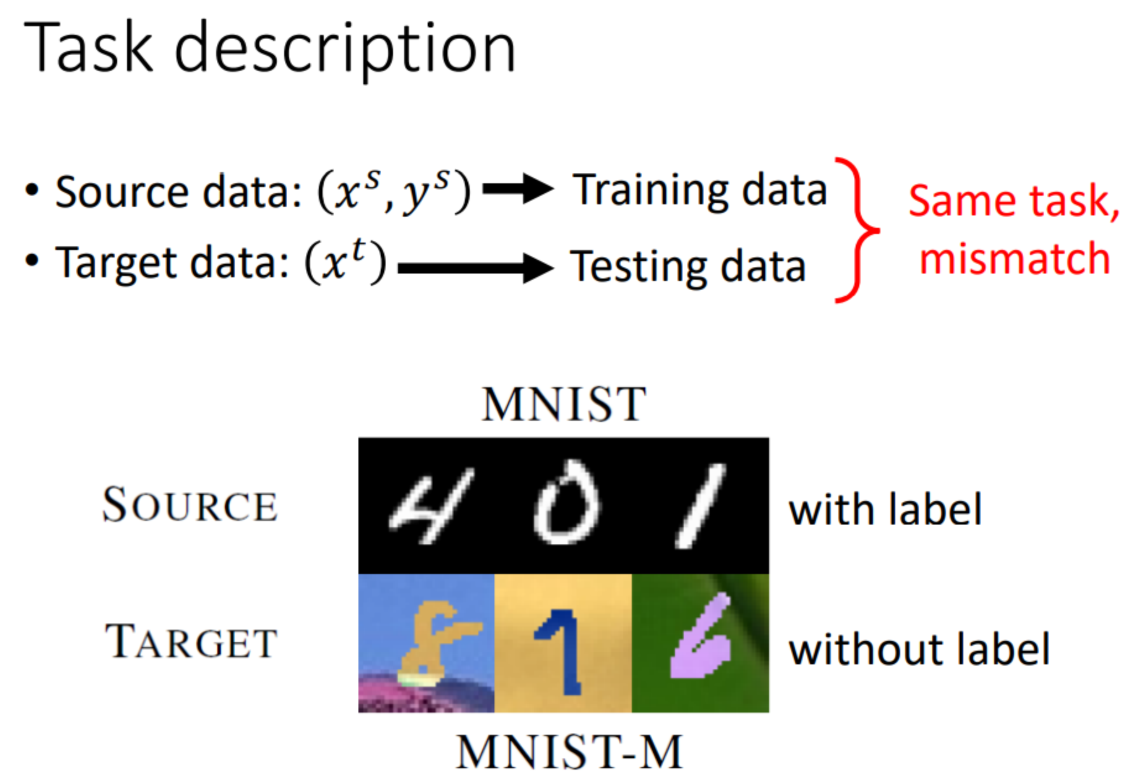


样例

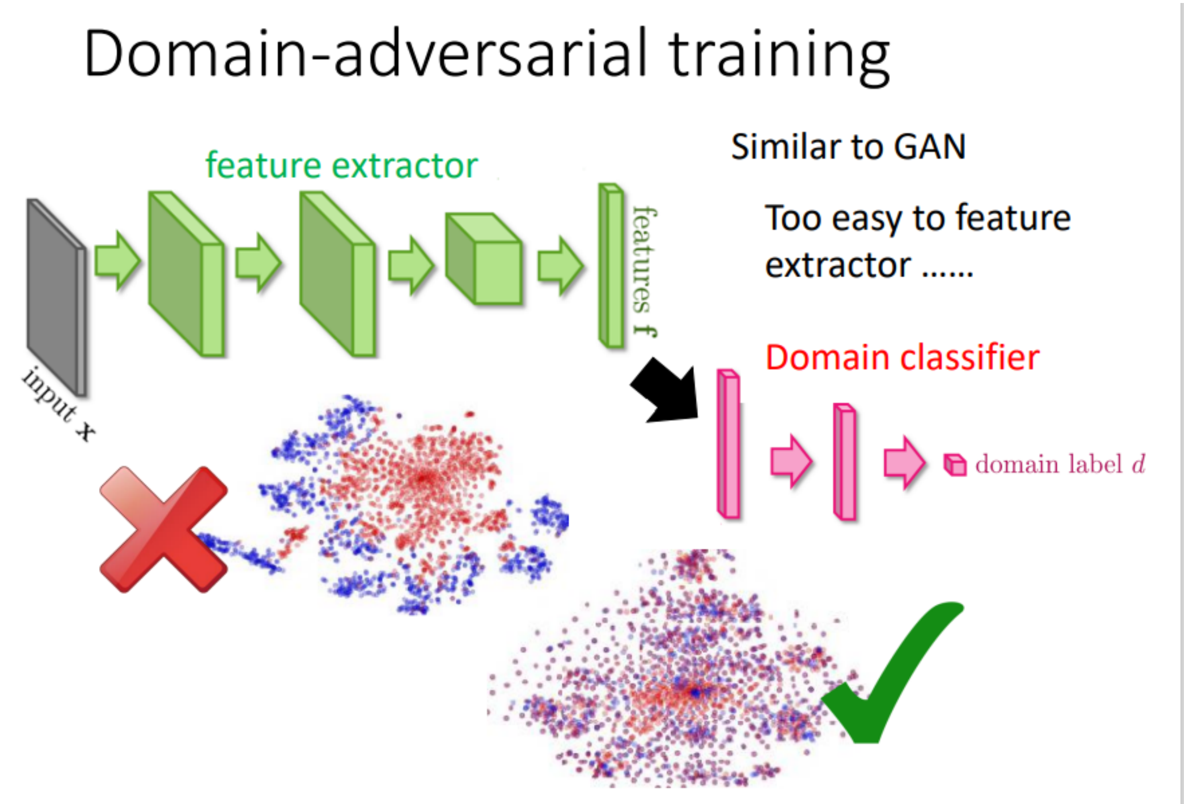


Domain-adversarial training

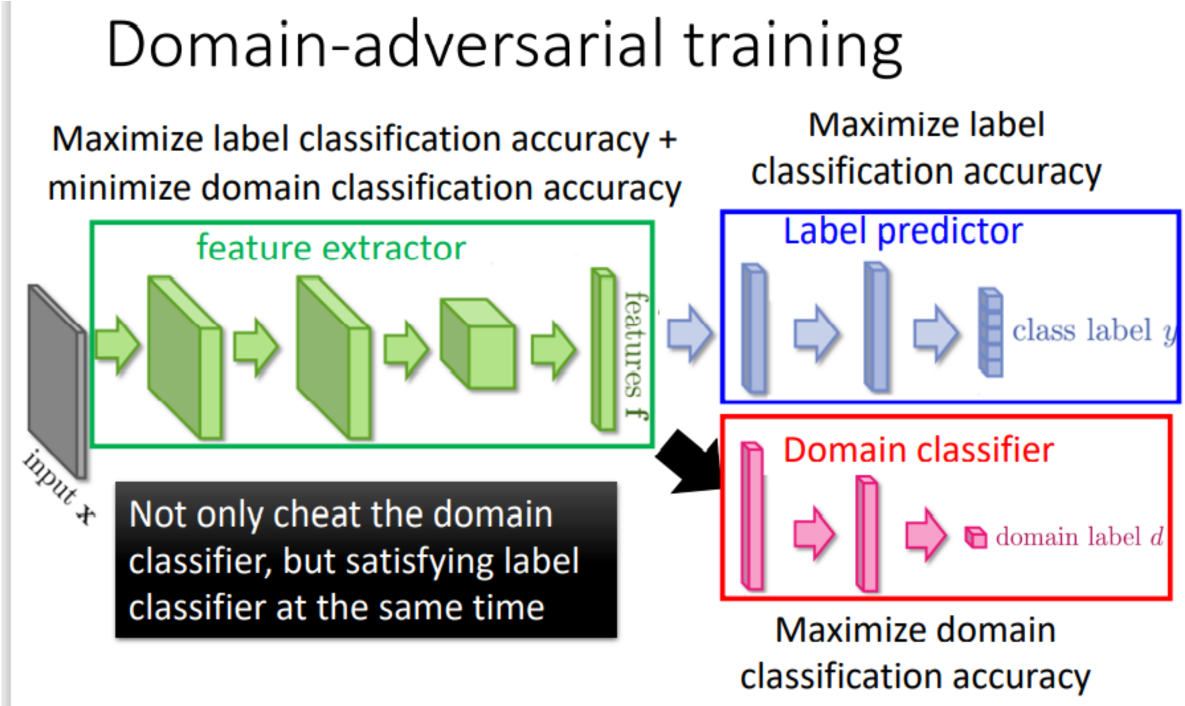
两个数据集很不一样，但是任务相同



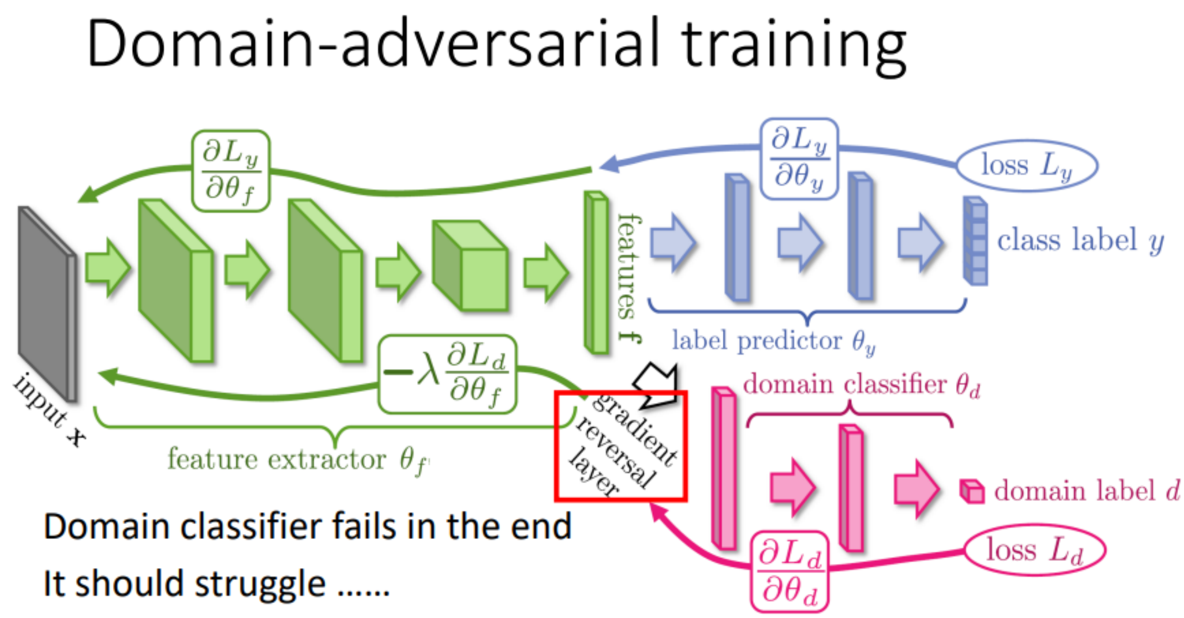
希望 feature extractor 可以消除不同 domain 的特性。加一个 Domain classifier 用来判断数据是哪个 domain的，我们需要 feature extractor输出的数据能够骗过 Domain classifier



这样还不够。消除Domain特性，并且保留原有特性（让label predictor也能做好）



feature extractor做根domain classifier相反的事情，把传过来的梯度都乘以一个负号



zero-shot learning

Transfer Learning - Overview

Target Data		Source Data (not directly related to the task): labelled	Source Data (not directly related to the task): unlabeled
Target Data	labelled	Fine-tuning Multitask Learning	
Target Data	unlabeled	Domain-adversarial training Zero-shot learning	

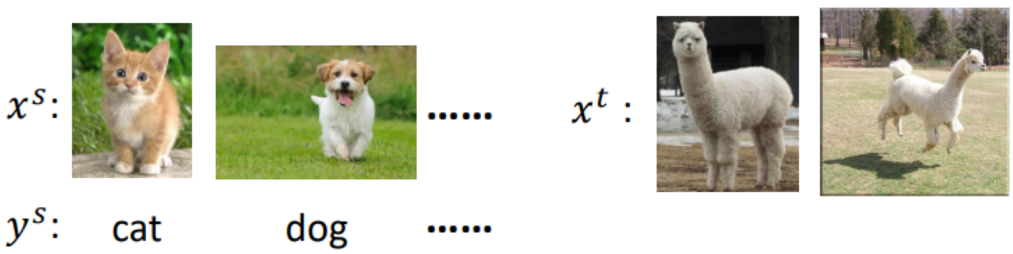
任务也是不一样的。target data 中的东西在 source data 中可能从来没有出现过

Zero-shot Learning

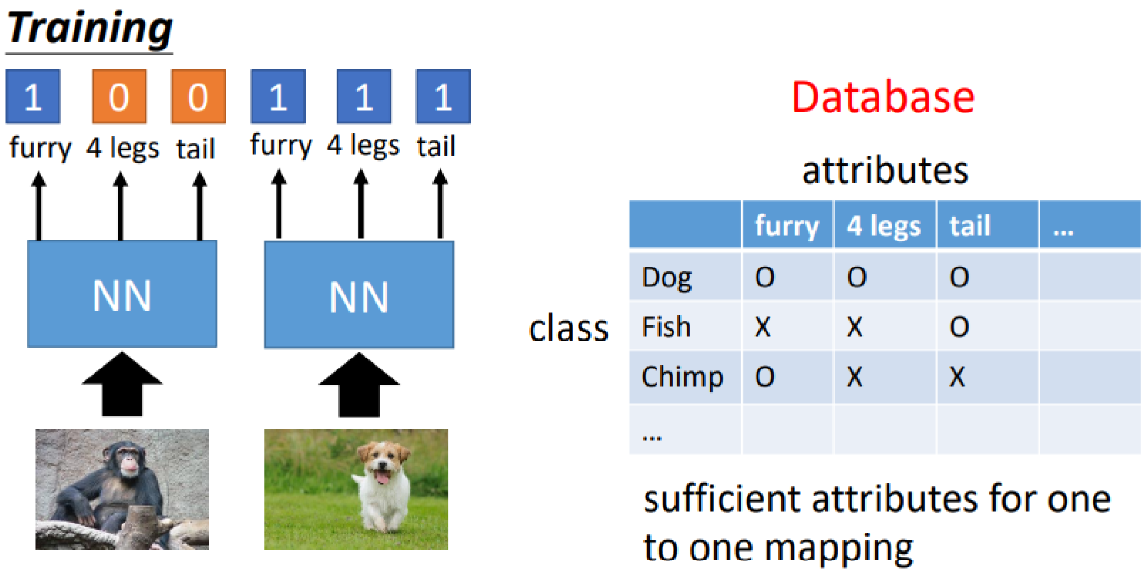
http://evchk.wikia.com/wiki/%E8%8D%89%E6%B3%A5%E9%A6%AC

- Source data: (x^s, y^s) → Training data
- Target data: (x^t) → Testing data

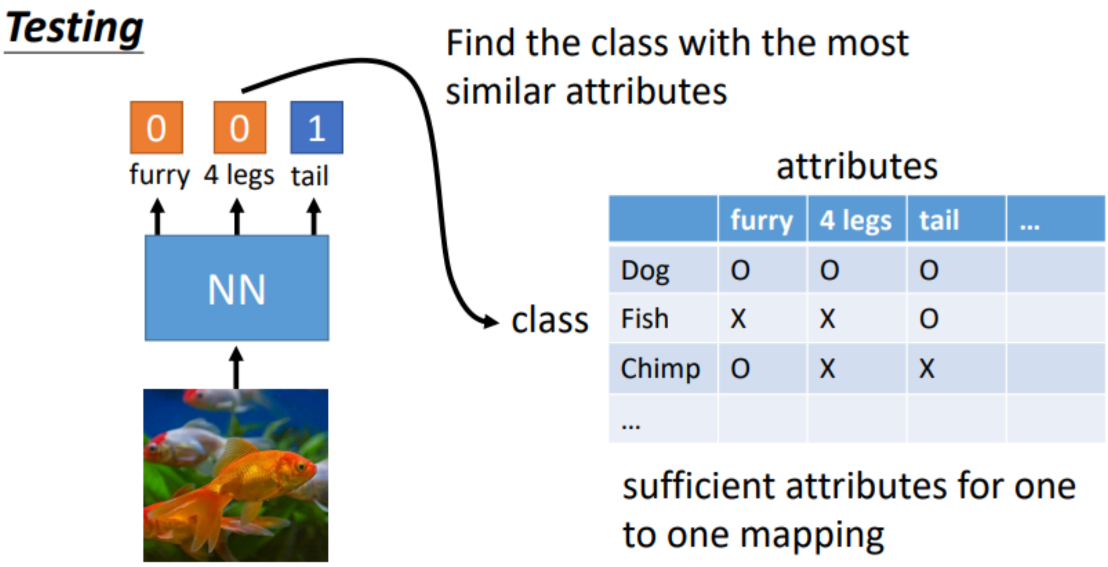
Different tasks



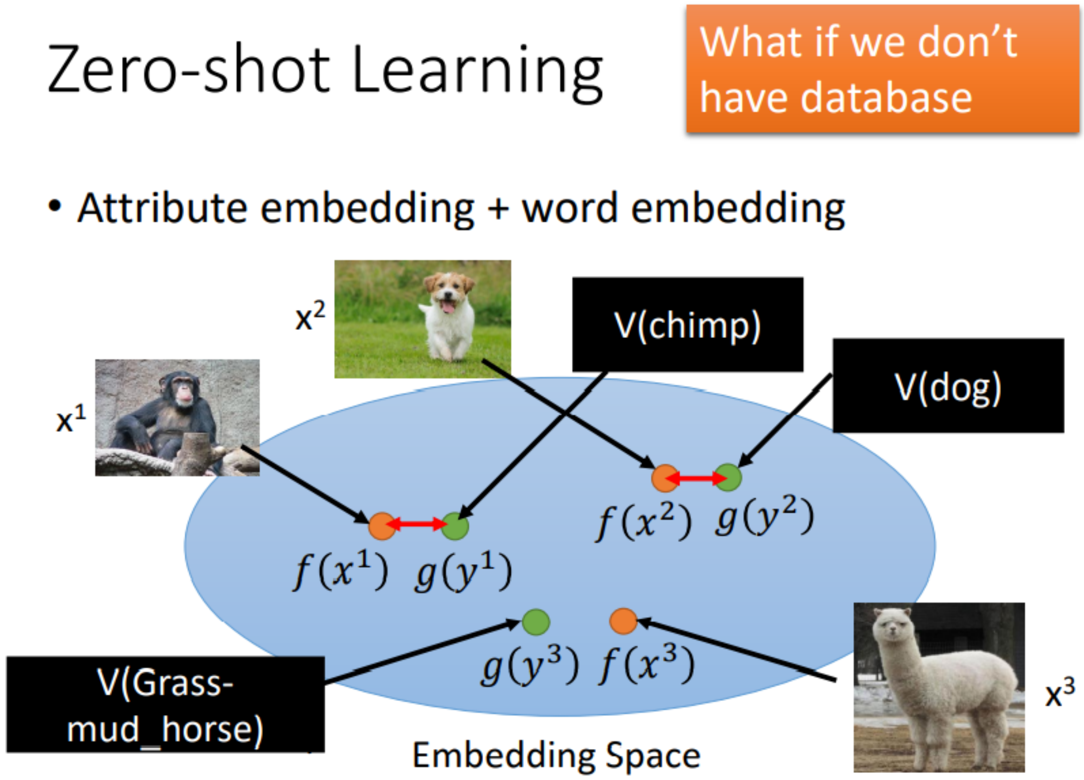
用属性来表示每一个 class。辨识的时候，我们去辨认里面包含哪些属性，而不是直接分类



找到一张图像中的属性，查表，看看和哪个最接近



如果没有database来表示各个类都有哪些属性，可以用将每个图用 word vector表示



能pair起来的一对，比不能pair起来的任何一对的得分都大 k， loss才是 0。否则产生loss

Zero-shot Learning

$$f^*, g^* = arg \min_{f,g} \sum_n \|f(x^n) - g(y^n)\|_2$$ Problem?

$$f^*, g^* = arg \min_{f,g} \sum_n max\left(0, k - f(x^n) \cdot g(y^n) + \max_{m \neq n} f(x^n) \cdot g(y^m)\right)$$

Margin you defined

Zero loss: $k - f(x^n) \cdot g(y^n) + \max_{m \neq n} f(x^n) \cdot g(y^m) < 0$

$$f(x^n) \cdot g(y^n) - \max_{m \neq n} f(x^n) \cdot g(y^m) > k$$

$f(x^n)$ and $g(y^n)$ as close $f(x^n)$ and $g(y^m)$ not as close

其他

Transfer Learning - Overview

		Source Data (not directly related to the task): labelled	Source Data (not directly related to the task): unlabeled
Target Data	labelled	Fine-tuning Multitask Learning	Self-taught learning Rajat Raina , Alexis Battle , Honglak Lee , Benjamin Packer , Andrew Y. Ng, Self-taught learning: transfer learning from unlabeled data, ICML, 2007
Target Data	unlabeled	Domain-adversarial training Zero-shot learning	Self-taught Clustering Wenyuan Dai, Qiang Yang, Gui-Rong Xue, Yong Yu, "Self-taught clustering", ICML 2008