

ML Lecture 19: Transfer Learning

臺灣大學人工智慧中心 科技部人工智慧技術暨全幅健康照護聯合研究中心 <http://ai.ntu.edu.tw/>

Overview

- **使用時機：**利用一些與 task 不直接相關的 data 來幫助現在要進行的 task
舉例：input domain 相似，但 task 不一樣；task 相似，但 input domain 不一樣

<http://weebly110810.weebly.com/396403913129399.html>

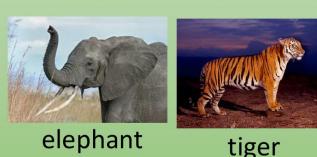
<http://www.sucaitianxia.com/png/cartoont/200811/4261.html>

Transfer Learning

Dog/Cat
Classifier



Data **not directly related** to the task considered

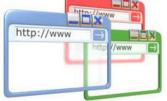


Similar domain, different tasks

Different domains, same task

- **使用原因：**有些 task 的 data 是較少的，我們可以試著用其他語言的 data 來 improve 原本的 task
舉例：要做台語的語音辨識，但台語的資料量少，我們可以嘗試用中文、英文等其他語言的 data 來 improve 台語的 task

<http://www.bigr.nl/website/structure/main.php?page=researchlines&subpage=project&id=64>
<http://www.spear.com.hk/Translation-company-Directory.html>

Task Considered	Data not directly related
Speech Recognition Taiwanese	 English Chinese
Image Recognition Medical Images	
Text Analysis Specific domain	 Webpages

舉例：現實生活中的 transfer learning，研究生的生活可以參考漫畫「爆漫王」

- 名詞介紹

- **target data**: 跟要做的 task 有直接相關的 data, 可能是 labelled 或 unlabelled
- **source data**: 跟 task 沒有直接相關的 data, 也可能是 labelled 或 unlabelled

故可將 transfer learning 分成四個象限討論,

先介紹 source data 及 target data 都 laballed 的情況下, 最常見也最簡的就是對 model 做 **Fine-tuning**

Transfer Learning - Overview

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

Model Fine-tuning

- 概覽

- **使用時機**: 有 labelled 的 大量 source data(x^s, y^s) 及 少量 target data (x^t, y^t)
- **想法**: 我們想知道在 target data 很少的情況下, 一大堆不相干的 source data 有沒有可能對 task 有幫助
- **作法**: 使用 source data 去 train 一個 model, 再用 target data 去 fine-tuning 這個 model;
亦即將 source data training 出的 model 當作初始值, 再用 target data 做 training.
- **One-shot learning**: 如果 target data 的量非常少, 少到只有幾個 example, 就叫做 One-shot learning
- **舉例**: 語音上, 最典型的例子是 speaker adaption

target data 是某一個人的聲音, source data 是一大堆來自不同人的 audio data

Model Fine-tuning

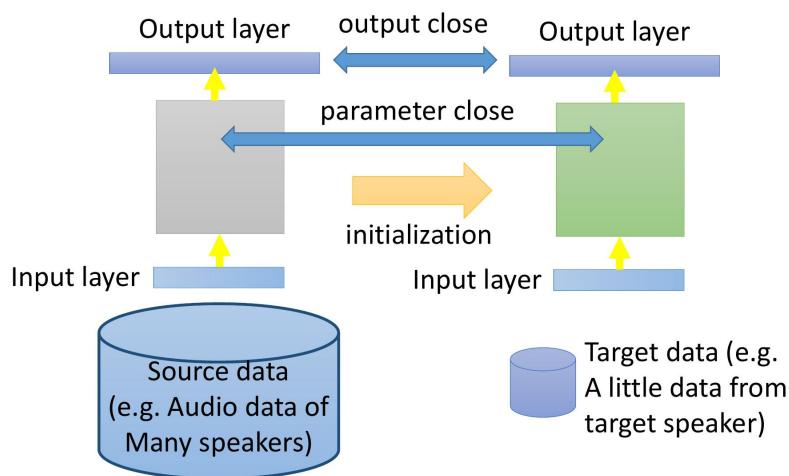
One-shot learning: only a few examples in target domain

- Task description
 - Source data: (x^s, y^s) ← A large amount
 - Target data: (x^t, y^t) ← Very little
- Example: (supervised) speaker adaption
 - Source data: audio data and transcriptions from many speakers
 - Target data: audio data and its transcriptions of specific user
- Idea: training a model by source data, then fine-tune the model by target data
 - Challenge: only limited target data, so be careful about overfitting

- **Conservative Training:** Fine-tuning 時的技巧之一

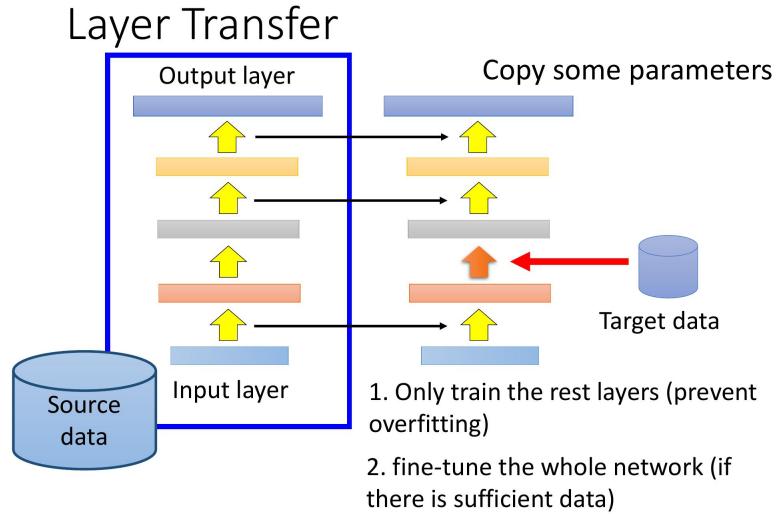
- 作法: 在 target data training 時加一些 constrain (即 regularization), 讓 train 完後的新 model 跟原本的舊 model 不要差太多
這樣可以防止 overfitting, 防止如果 target data 非常少, 一 train 就壞掉的情況。
 - 舉例: 可以加 constrain 讓新 model 跟舊 model 看到同一筆 data 的時候, output 越接近越好; 或是 L2-norm 的差距越小越好

Conservative Training



- **Layer Transfer:** Fine-tuning 的技巧之二

- 作法: 將 source data train 好的 model 中某幾個 layer 拿出來, 直接 copy 到新的 model 中, 再用 target data train 剩下 (沒有 copy) 的 layer; 如果 target data 夠多, 要 fine-tuning 整個 model 也是可以的



- 技巧：如何選擇哪些 layer 應該被 transfer，哪些不應該被 transfer？

- 語音辨識：通常是 copy 最後幾層，重新 train input 那幾層

前幾層：從聲音訊號得知說話者的發音方式，而每個人的口腔結構不同，同樣的發音方式，得到的聲音是不一樣的

後幾層：根據發音方式判斷現在說的是哪一個詞彙，即可得辨識的結果，這個過程跟說話者較無關

- 圖片辨識：通常是 copy 前面幾層，重新 train 最後幾層

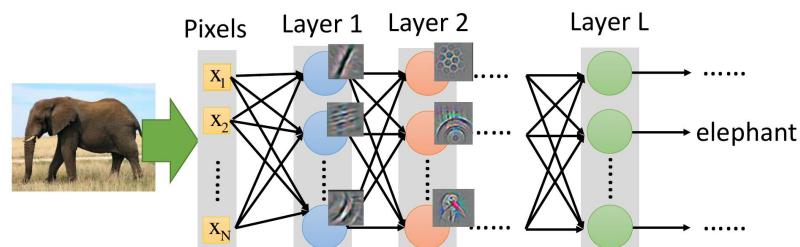
前幾層：往往是 detect pattern，如直線、橫線、幾何圖形等，可被 transfer

後幾層：往往是比較抽象的概念，沒有辦法 transfer

故不同的 task，需要被 transfer 的 layer 往往是不一樣的，需要一些 domain know-how 來幫助判斷

Layer Transfer

- Which layer can be transferred (copied)?
- Speech: usually copy the last few layers
- Image: usually copy the first few layers



- 舉例：image 在 layer transfer 上的實驗，出自 Bengio 在 NIPS, 2014 的 paper

- 定義

橫軸：代表做 transfer learning 時 copy 的 layer 數目

縱軸：為 Top-1 accuracy，越高代表表現越好

Data：將 ImageNet 中 120 萬張 image，其中 500 個 class 歸為 source data，另外 500 個 class 歸為 target data

Baseline：圖中空白圓點，完全沒做 transfer learning

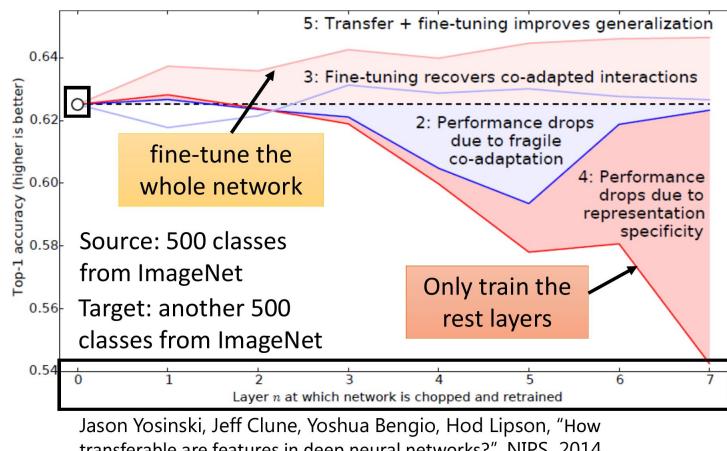
■ 結果

- 只 copy 第一個 layer 的時候，performance 稍有進步；但 copy 越多層 layer，performance 就壞掉了

所以，實驗顯示出在不同的 data 上面，前面幾層 layer 是可以共用的，後面幾層 layer 是無法共用的

- 最上面那條橙色的線是「Transfer learning + Fine-tuning」，可以發現在做有的 case 上 performance 都有進步

Layer Transfer - Image

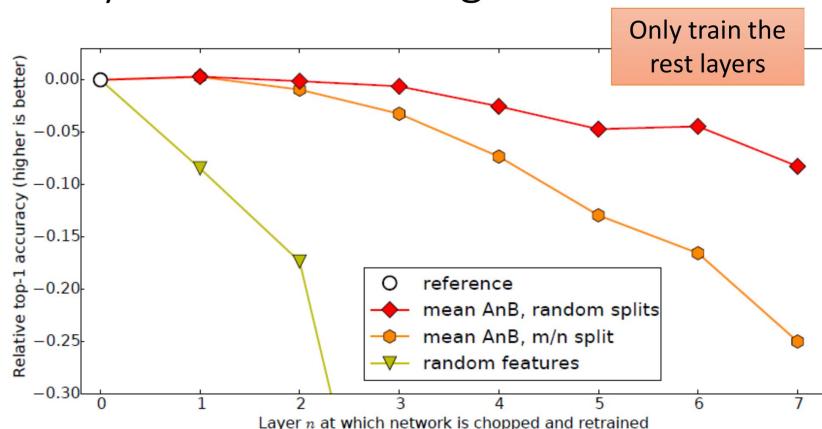


■ 發現

紅色的線：跟上圖中的紅線是同一條

綠色的線：假設參數是 random 時，結果壞掉

Layer Transfer - Image



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

Multitask Learning

- 概覽

- **使用時機**: 同時關心 target domain 及 source domain 的表現，希望兩者表現都好
- **訓練模型**: Deep learning based 的方法，特別適合用來做 multitask learning
- **舉例**:

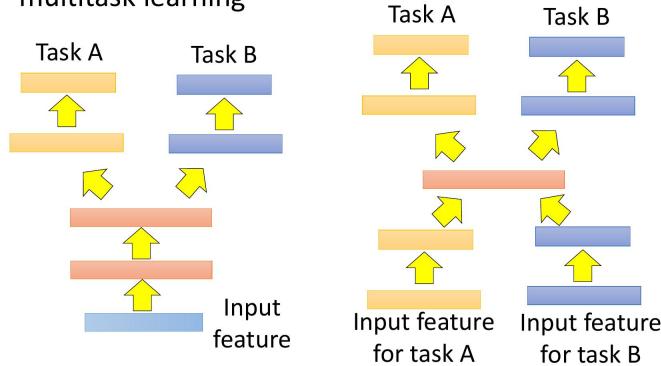
(左圖): 兩個不同的 task，相同的 input feature，只是影像辨識的 class 不同
我們就可以 learn 一個 NN，input 相同的 feature，但是中間岔開分別 output task A 及 B 的結果。

這麼做的前提是這兩個 task 有共通性；優點是前面幾個 layer 使用較多的 data train，表現可能較好

(右圖): 兩種不同的 task，不同的 input feature，使用不同的 NN
我們可以把它 transform 到同一個 domain 上，這樣中間幾個 layer 也可以做 multitask learning，最後，再 apply 到不同的 NN，分別 output 各自的結果

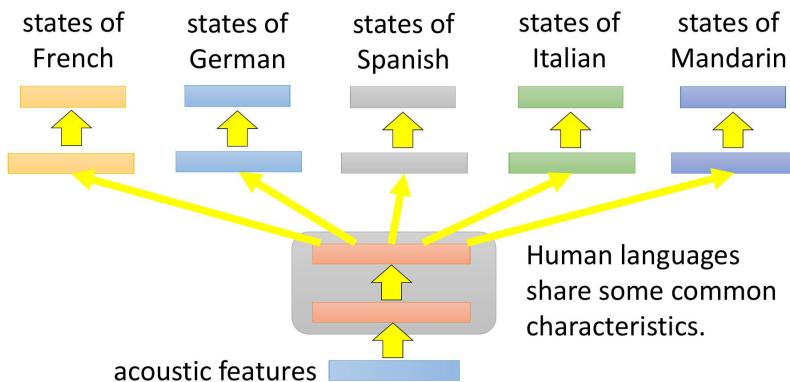
Multitask Learning

- The multi-layer structure makes NN suitable for multitask learning



- **應用:** Multitask learning 一個很成功的例子是「多語言的語音辨識」
 - Input 是各種不同語言的 data，在 train model 時，前面幾層的 layer 會共用參數，後面分岔；
因為不同的語言都是人類說的，所以前面幾層可以 share 同樣的資訊。

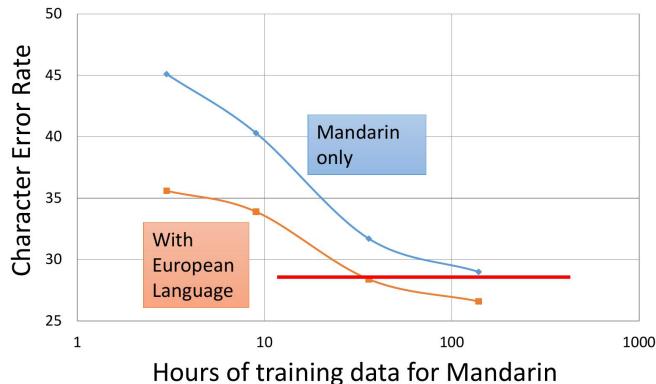
Multitask Learning - Multilingual Speech Recognition



Similar idea in translation: Daxiang Dong, Hua Wu, Wei He, Dianhai Yu and Haifeng Wang, "Multi-task learning for multiple language translation.", ACL 2015

- **推廣:** 「翻譯」也可以做同樣的事
 - 在中翻英、中翻日時，因為都要先 process 中文 data 的部分，因此一部分的 network 就可以共用
 - 這種語言 transfer 的範圍有多大呢？目前的發現是幾乎所有的語言都可以 transfer

Multitask Learning - Multilingual



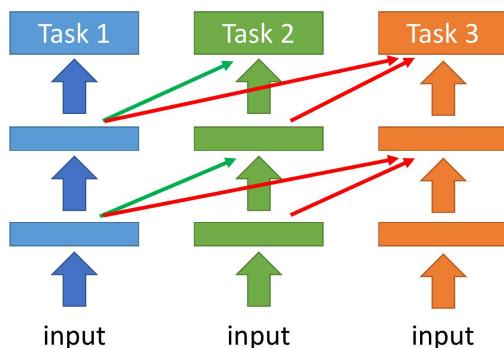
Huang, Jui-Ting, et al. "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers." ICASSP, 2013

- 藍線是只有中文 data training 的結果，橘線是從歐洲語言 transfer 到中文上面，可以發現 單獨 train 中文 100 小時的 performance 跟 train with 歐洲語言 50 小時的效果是一樣的，亦即在這個例子中，我們只需要 1/2 以下的 data，就可以跟原來有兩倍的 data 的效果一樣好。

- **Progressive Neural Network:**

- 想法：先 train 一個 task 1 的 NN，train 好後，task 2 的每一層 hidden layer 就去接 task 1 某一層 hidden layer 的參數，而 task 2 也可以把這些參數直接設成 0，亦即最糟的情況跟 task 2 自己 train 的 performance 是一樣的。而 task 3 再從 task 1 及 task 2 的 hidden layer 得到 information

Progressive Neural Networks



Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, Raia Hadsell, "Progressive Neural Networks", arXiv preprint 2016

Domain-adversarial Training

- 概覽

- 使用時機：資料是 **labeled source data** 及 **unlabeled target data** 時

source data: (x^s, y^s) , target data: (x^t)

- 舉例：以下圖為例

source data 是 MNIST 上有 labeled 的 image, target data 是 MNIST-M 上沒有 labeled 的 image

這種情況下，我們通常將 source data 視為 train data, target data 視為 testing data
但遇到一個問題是：training data 跟 testing data 非常的 **mismatch**

Task description

- Source data: $(x^s, y^s) \rightarrow$ Training data
 - Target data: $(x^t) \rightarrow$ Testing data
- } Same task, mismatch

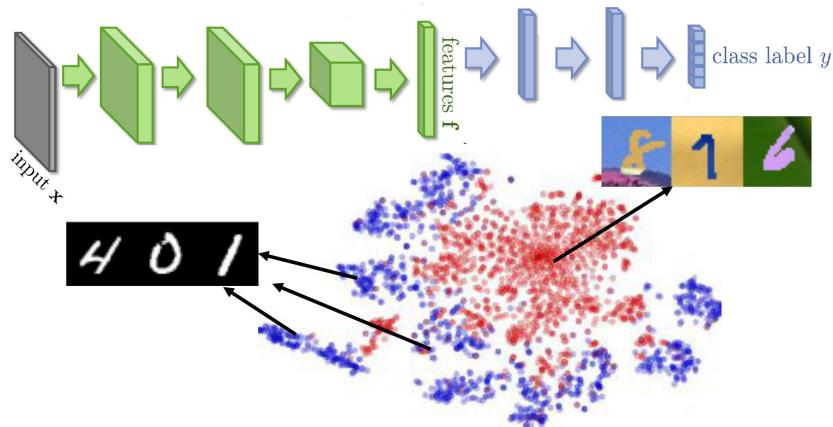


- 實作

- 想法：

- 如果直接 learn 一個 model, 如下圖, 會發現前面幾層抽 feature 的結果是爛的。
藍色很明顯地分成十群, 紅色這群卻是一坨。
- 在 feature extraction 時, source domain 跟 target domain 不在同一位置上, 所以,
我們希望
能把 **domain** 的特性去除掉。

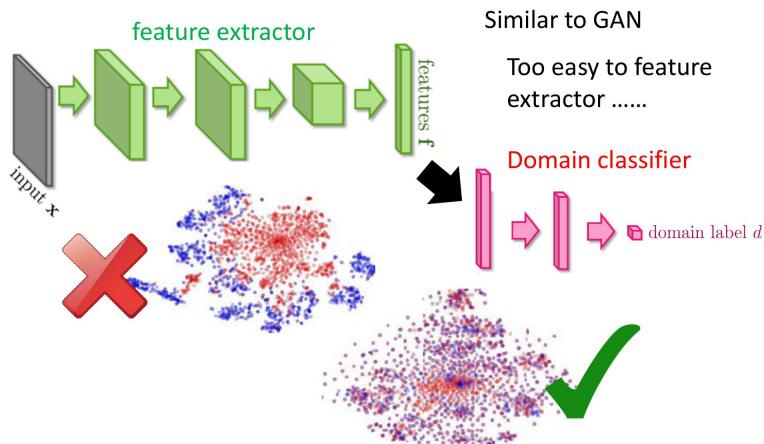
Domain-adversarial training



- 實現：

- 在 feature extractor 後面接一個 **domain classifier**, 將 output 丟到 domain classifier, 如此一來, 就能將 domain 的特性消掉, 將不同 domain 的 image 混在一起
- 而有一個 generator 的 output 跟有一個 discriminator 這樣的架構, 非常像 GAN
- 但在 domain-adversarial training 中, 要產生一張 image 騙過 classifier 太簡單了, 所以 feature extractor 的 output 不只要騙過 **domain classifier** 還要同時讓 **label predictor** 做好

Domain-adversarial training



- domain-adversarial training 三個 part 的目標是各自不同的

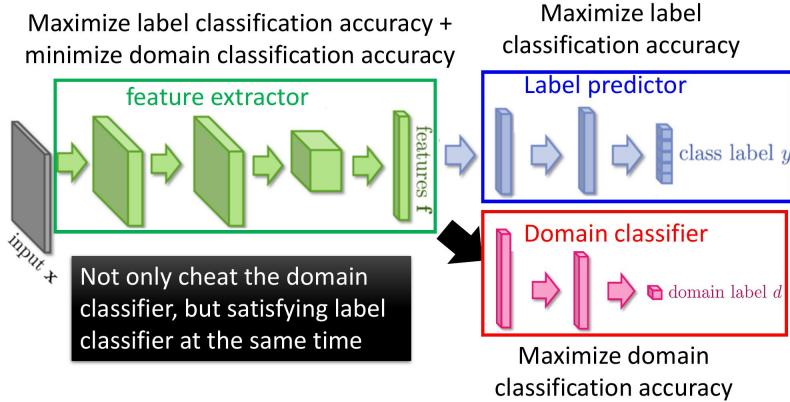
Feature extractor : 增進 label predictor 正確率的同時，最小化 domain classifier 的正確率

Label predictor : 最大化 classification 的正確率

Domain classifier : 正確的預測一個 image 屬於哪一個 domain

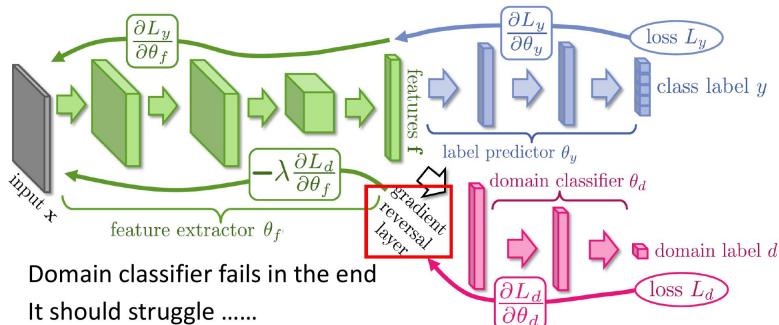
所以, feature extractor 跟 domain classifier 想做的事情是相反的

Domain-adversarial training



- Feature extractor 只要在最後加上一個 gradient reversal 的 layer；這樣 domain classifier 做 back propagation，計算 backward path 時，feature extractor 會將 domain classifier 傳進來的 output 故意乘上負號，做跟 domain classifier 要求相反的事
- 這樣，domain classifier 會因看不到真正的 image，而 fail 掉
但是，domain classifier 一定會 **struggle** 完才 fail，這樣才能把 domain 的特性去掉

Domain-adversarial training



Yaroslav Ganin, Victor Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML, 2015

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Domain-Adversarial Training of Neural Networks, JMLR, 2016

- 故 domain-adversarial training 實際上也沒有很好 train
- 例子

引用 ICML,2015 跟 JMLR,2016 的 paper 中一些實驗的結果

- 概覽

- 資料：包括 MNIST 到 MNIST-M、一個數字的 corpus 到另一個數字的 corpus、數字的 corpus 到 MNIST 及兩種不同道路號誌 data 互相 transfer 等
- 縱軸：每筆資料用四種不同方法得到的實驗結果

- 比較

1. **Source only** 是直接在 source domain 上 train 一個 model，再到 testing domain 上 test

2. **Proposed** 的即 domain-adversarial training
 3. **Train on target** 則是直接拿 target domain 的 data 做 training, 得到的 performance 即 upper bound
- 發現
 - source only 跟 train on targert 間有很大的 gap, 而用 domain-adversarial training 在不同的 case 上, 都可以得到很好的 improvement

Domain-adversarial training

	MNIST	SYN NUMBERS	SVHN	SYN SIGNS	
SOURCE					
TARGET					
	MNIST-M	SVHN	MNIST	GTSRB	
METHOD	SOURCE TARGET	MNIST MNIST-M	SYN NUMBERS SVHN	SVHN MNIST	SYN SIGNS GTSRB

Yaroslav Ganin, Victor Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML, 2015

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Domain-Adversarial Training of Neural Networks, JMLR, 2016

Zero-shot Learning

- 概覽
 - 使用時機: 資料是 **labeled source data** 及 **unlabeled target data**, 且 source data 跟 target data 要做的 task 是不一樣的
 - 舉例:

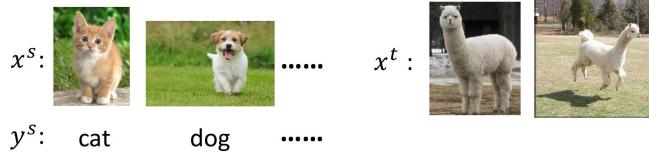
如下圖, source data 要分辨貓跟狗, target data 的 image 則是草泥馬
最常見的則是語音上的應用

Zero-shot Learning

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

- Source data: $(x^s, y^s) \rightarrow$ Training data
- Target data: $(x^t) \rightarrow$ Testing data

Different tasks



In speech recognition, we can not have all possible words in the source (training) data.

How we solve this problem in speech recognition?

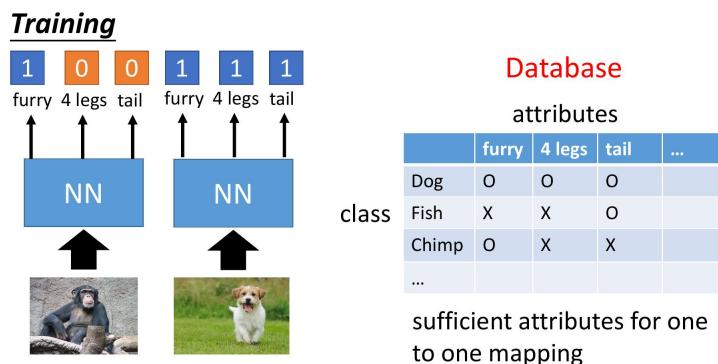
• 實作

◦ Representing:

- 將每一個 class 用他的 **attribute** 表示；
亦即，有一個 database 存每一個 object 跟所有可能的特性（如下圖右下表）
- ex: 狗：毛茸茸、四隻腳、有尾巴；魚：不毛茸茸、沒有四隻腳、有尾巴.....等
- 每個 class 都要有獨一無二的 **attribute**，亦即 attribute 要定的夠豐富才行
一旦有兩個 class 有一模一樣的 attribute，這個方法就會 fail 掉

Zero-shot Learning

- Representing each class by its attributes



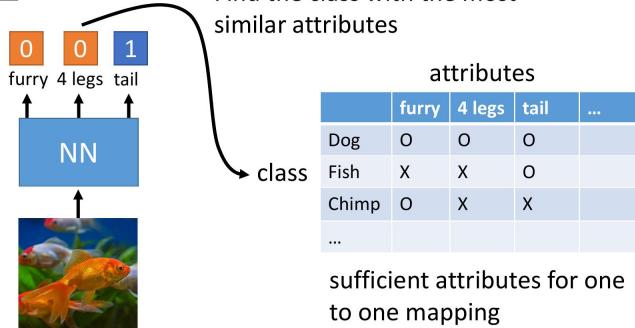
◦ Training & Testing

- 在 training 時，要做的是辨識每一張 image 具備什麼樣的 attribute，而不是直接辨識這張 image 屬於哪一個 class；所以即使 testing 時出現一張不存在的動物，我們只要辨識出它具有哪些 attribute，查 database 找最接近的動物，就可以了

Zero-shot Learning

- Representing each class by its attributes

Testing



- Attribute embedding

- 當 attribute 的 **dimension** 很龐大時，我們可以做 attribute 的 embedding
- 將 training data 上的每一張 image 跟每一個 attribute 都 transform 成 embedding space 上的一個點 $f(x^1), g(y^1)$
- f, g 都可以是 neural network, training 時希望 $f(x^n)$ 跟 $g(x^n)$ 越接近越好
- 出現一張沒看過的草泥馬 x^3 時，就投影到 $f(x^3)$ ，再找最近的 $g(y^3)$ ， y^3 就是他的 attribute，再看對應到哪一個動物，就結束了

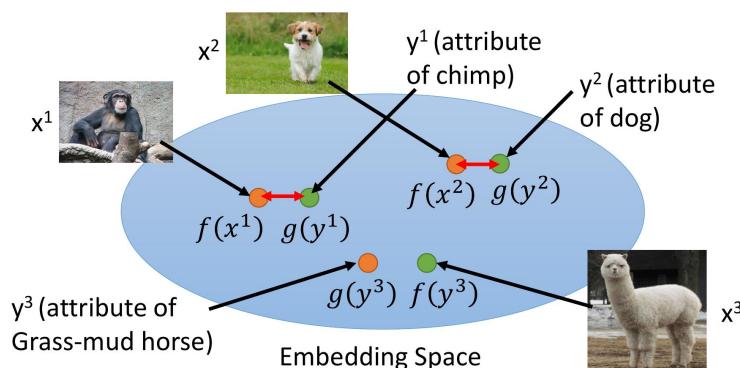
Zero-shot Learning

$f(*)$ and $g(*)$ can be NN.

Training target:

$f(x^n)$ and $g(y^n)$ as close as possible

- Attribute embedding



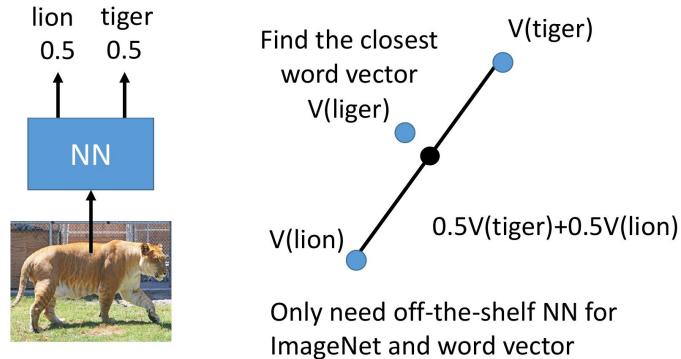
- Convex combination of semantic embedding

- 條件：有一組 **word vector** + 一個語音辨識系統
- 作法：

1. 將圖片丟到 NN 中，得到有0.5的機率是獅子、0.5的機率是老虎
2. 再找獅子跟老虎的 word vector 用上面的比例(1:1)混合，得到新的 vector
3. 再找哪一個 word 跟混合出的新 vector 最接近，最後得到獅虎，這張圖片就是獅虎

Zero-shot Learning

- Convex Combination of Semantic Embedding



Self-taught Learning

- 概覽
 - 使用時機：資料是 **unlabeled source data** 及 **labeled target data**
 - 作法：
 - 用夠多的 unlabeled source data 去 learn 一個 feature extractor
 - 再用這個 feature extractor 在 target data 上抽 feature

Self-taught learning

- Learning to extract better representation from the source data (unsupervised approach)
- Extracting better representation for target data

Domain	Unlabeled data	Labeled data	Classes	Raw features
Image classification	10 images of outdoor scenes	Caltech101 image classification dataset	101	Intensities in 14x14 pixel patch
Handwritten character recognition	Handwritten digits ("0"-“9”)	Handwritten English characters ("a"-“z”)	26	Intensities in 28x28 pixel character/digit image
Font character recognition	Handwritten English characters ("a"-“z”)	Font characters ("a"/“A” - “g”/“Z”)	26	Intensities in 28x28 pixel character image
Song genre classification	Song snippets from 10 genres	Song snippets from 7 different genres	7	Log-frequency spectrogram over 50ms time windows
Webpage classification	100,000 news articles (Reuters newswire)	Categorized webpages (from DMOZ hierarchy)	2	Bag-of-words with 500 word vocabulary
UseNet article classification	100,000 news articles (Reuters newswire)	Categorized UseNet posts (from “SRAA” dataset)	2	Bag-of-words with 377 word vocabulary