



11-Domain Adaptation

1. 什麼是 Domain Adaptation ?

2. Domain Shift

2.1. 輸入分布不同

2.2 輸出分布不同

2.3 輸入輸出的關係發生變化

3. Domain Adaptation

3.1 少量有標註的資料 (Little but labeled)

3.2 大量無標註的資料 (Large amount of unlabeled data)

3.2.1 Feature Extractor

3.2.2 Domain Adversarial Training

3.2.3 Boundary

3.2.4 Universal domain adaptation

3.3 少量無標註的資料 (Little & unlabeled)

3.4 對 Target Domain 一無所知

1. 什麼是 Domain Adaptation ?

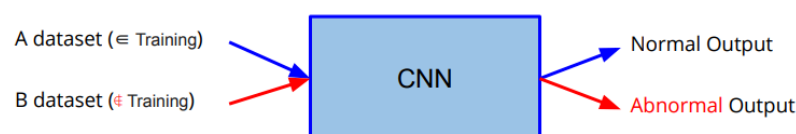
Domain Adaptation 可以看做是 Transfer Learning 的一種

- **Transfer Learning** : 模型在 A 任務上學到的技能，可以被用在 B 任務上
- **Domain Adaptation** : 在訓練資料的 domain 上學到的資訊，用到另外一個測試資料的 domain 上面

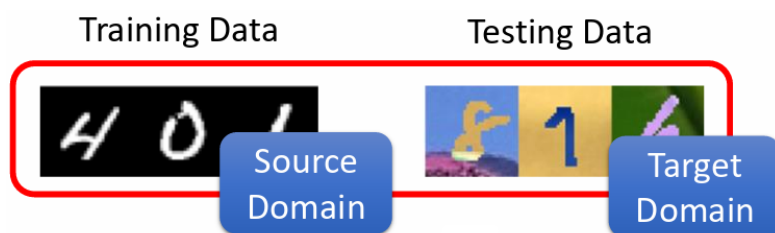
2. Domain Shift

domain shift 意思為訓練資料跟測試資料的分布不一樣，在訓練資料上訓練出來的模型，在測試資料上可能表現不好

- 如果一個模型在測試時吃到不是與訓練集同個 distribution 的輸入，那麼輸出往往會爆走，如下圖。
- 而為什麼不能讓圖中的 CNN 在輸入 B dataset 輸出正常的 output？因為你並沒有 B dataset 的 label 使模型學習。



- **Source Domain**：訓練資料
- **Target Domain**：測試資料



2.1. 輸入分布不同

訓練資料的數字是黑白的，但測試資料是彩色的，那麼正確率會下降



2.2 輸出分布不同

訓練資料每一個數字它出現的機率都一樣；但在測試資料中，每一個輸出的機率是不一樣的，有可能某一個數字的輸出機率特別大



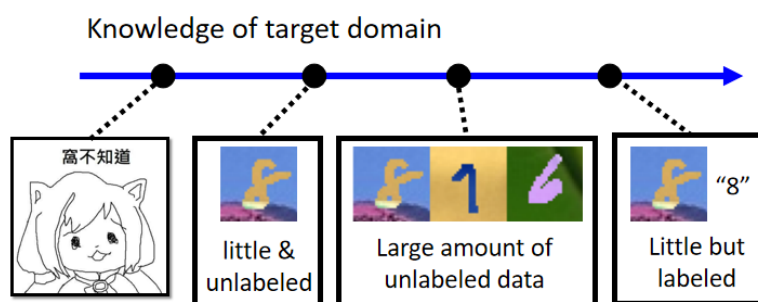
2.3 輸入輸出的關係發生變化

舉例而言，左圖的訓練資料代表 0，但有可能在測試資料中是代表 1。較罕見



3. Domain Adaptation

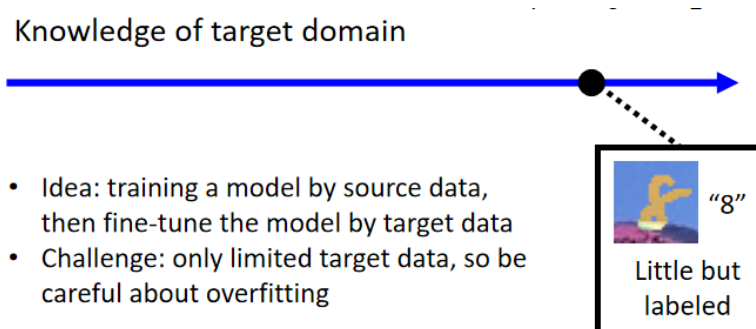
根據 target domain 中資料的情況，由右至左可分為：



- 少量有標註的資料 (Little but labeled)
- 大量無標註的資料 (Large amount of unlabeled data)
- 少量無標註的資料 (Little & unlabeled)
- 對 target domain 一無所知

3.1 少量有標註的資料 (Little but labeled)

target domain 為少量有標註的資料，可以用 target domain 的資料 **fine-tune** 用 source domain 訓練出來的模型



方法：

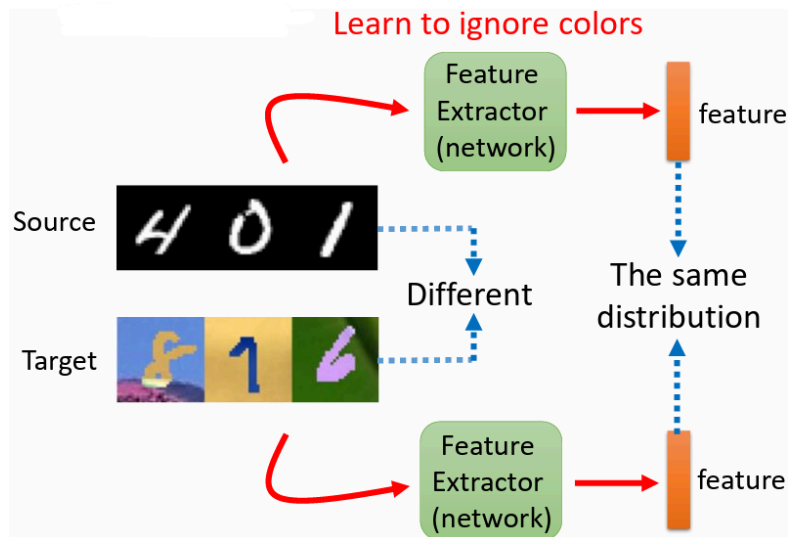
微調方式與 BERT 相似。已經有一個在 source domain 上訓練好的 model，拿 target domain 的資料丟進模型跑個二至三個 epoch 即可

注意：

由於 target domain 的資料量很少，**epoch** 的大小不宜過多，以免有 **overfitting** 的狀況

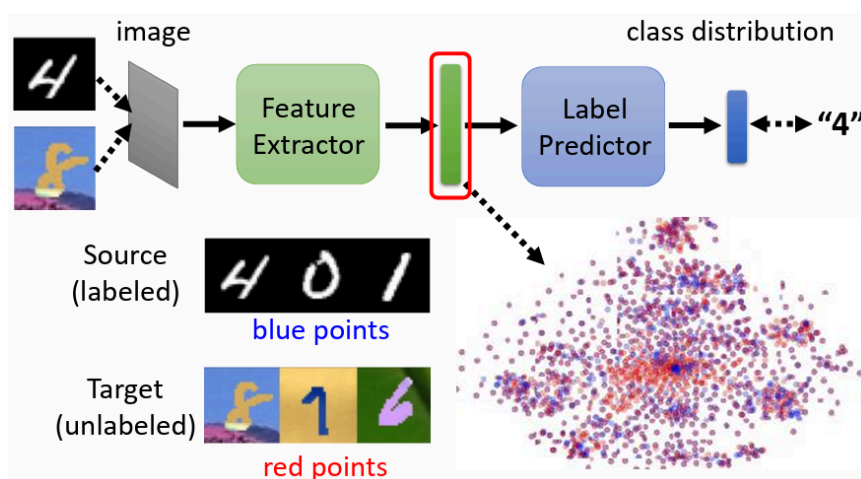
3.2 大量無標註的資料 (Large amount of unlabeled data)

訓練一個 network：**feature extractor**，source domain 跟 target domain 的圖片看起來不一樣，可以藉由 feature extractor 把不一樣的部分拿掉，只抽取出共同的部分，讓輸出具有相同的分布



3.2.1 Feature Extractor

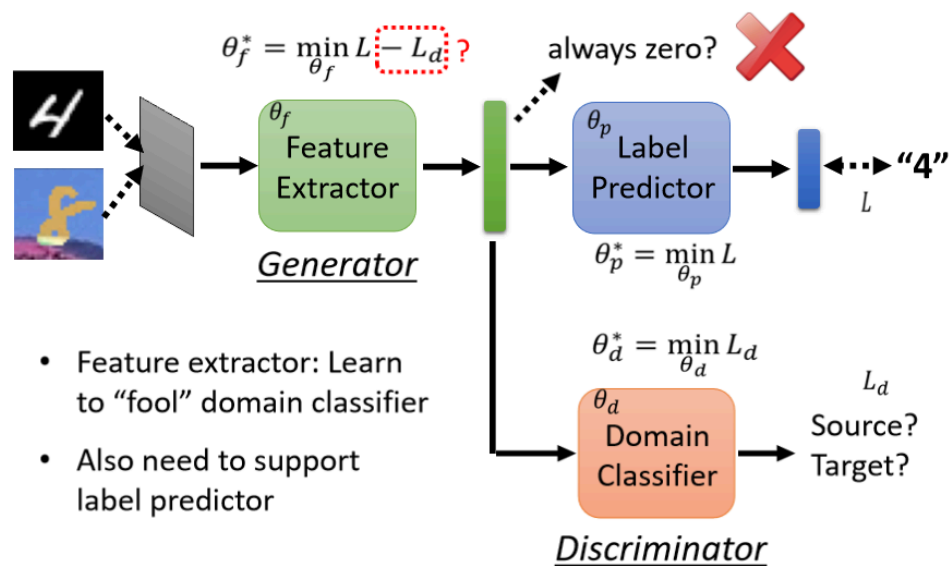
把一般的 classifier 分成 **feature extractor** 及 **label predictor** 兩個部分，例如將前 5 層算是 feature extractor，後 5 層算是 label predictor。層數為須自行調整的**超參數**



將 source domain 和 target domain 的圖片輸入進 image classifier，希望 feature extractor 針對兩個 domain 的資料的輸出要看起來沒有差異

3.2.2 Domain Adversarial Training

藉由 **Domain Adversarial Training** 來實現，類似於 GAN，新增一個 **domain classifier** 作為 **discriminator**，判斷輸入來自哪個 domain，**feature extractor** 如同 **generator**，要想辦法騙過 domain classifier



符號：

θ_p ：label predictor 的參數

θ_d ：domain classifier 的參數

θ_f ：feature extractor 的參數

L ：label predictor 對實際的分類任務進行分類，分類結果與實際分類之間的 cross entropy

L_d ：domain classifier 對 feature extractor 輸出的 feature 進行二元分類，判斷來自哪個 domain，分類結果與實際分類之間的 loss

目標：

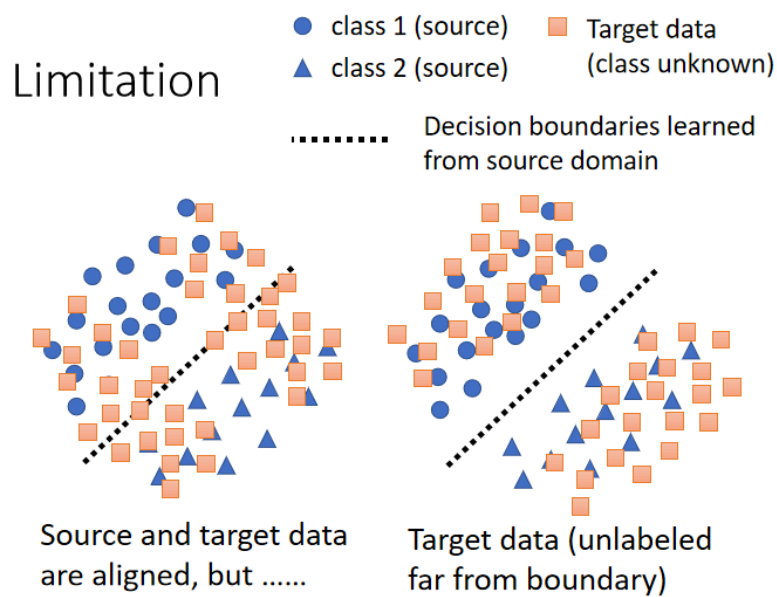
找到一組 θ_p 讓 L 越小越好

找到一組 θ_d 讓 L_d 越小越好

找到一組 θ_f 既能使 label predictor 分類準確（最小化 L ），又能使 domain classifier 難以分辨（最大化 L_d ） \Rightarrow 定義 feature extractor 的 loss 為 $L - L_d$ ，找到一組 θ_f 讓 $L - L_d$ 越小越好

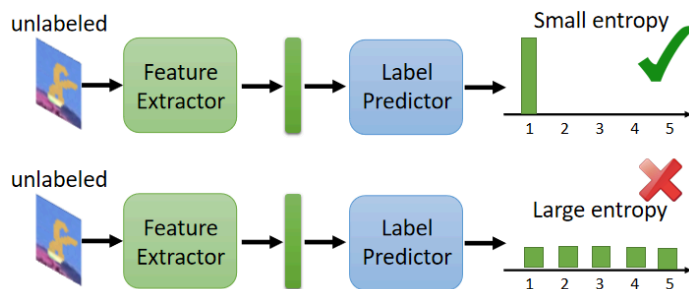
3.2.3 Boundary

希望兩個 domain 的分布更加接近，也就是讓 target domain 的資料盡量避開 source domain 中得到的 boundary，離 boundary 越遠越好



方法：

1. 讓不同類別之間的輸出差異盡可能地大，盡可能地集中在某一類別上



Used in Decision-boundary Iterative Refinement Training with a Teacher (DIRT-T)

<https://arxiv.org/abs/1802.08735>

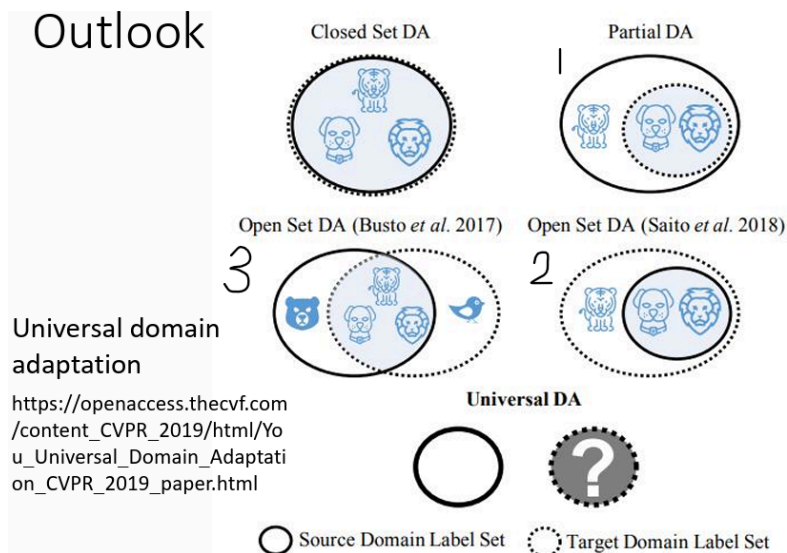
Maximum Classifier Discrepancy

<https://arxiv.org/abs/1712.02560>

2. A DIRT-T Approach to Unsupervised Domain Adaptation (DIRT-T)
3. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation

3.2.4 Universal domain adaptation

實線圈代表 source domain 中有的類別，虛線圈代表 target domain 中的類別



兩個 domain 的類別不一定都完全相同，有另外三種可能：

1. source domain 的類別比較多，而 target domain 的類別比較少
2. source domain 的類別比較少，而 target domain 的類別比較多
3. 兩者有交集，但也都各有獨有的類別

問題：

對 2. 這種可能來說，若要讓 source domain 的資料跟 target domain 的資料的 feature 完全 match 在一起，意味著硬是要讓老虎去變得跟狗或是獅子像，到時候就分不出老虎這個類別

解決：

Universal Domain Adaptation

3.3 少量無標註的資料 (Little & unlabeled)

問題：

target domain 沒有標註資料並且數量還很少，或甚至只有一張

解決：

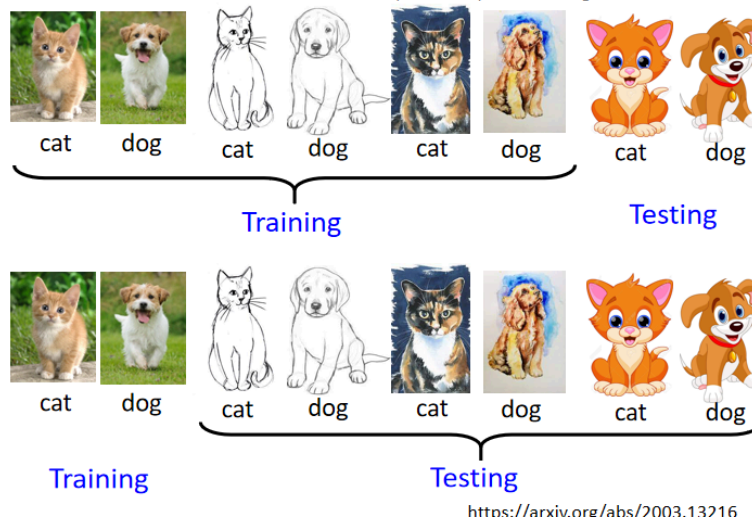
Testing Time Training (TTT)

3.4 對 Target Domain 一無所知

期望模型有 **Domain Generalization** 能力

Domain Generalization

<https://ieeexplore.ieee.org/document/8578664>



狀況：

1. 訓練資料非常豐富，具有各式各樣不同的 domain

期待因為訓練資料有**多個 domain**，模型可以學到如何**彌平 domain 間的差異**，測試資料有沒看過的 domain 也可以處理。可參考文獻：[Domain Generalization with Adversarial Feature Learning](#)

2. 訓練資料只有一個 domain，而測試資料有多種不同 domain

在概念上與 data augmentation 相似，從已知 domain 的資料產生多個 domain 的資料。可參考文獻：[Learning to Learn Single Domain Generalization](#)