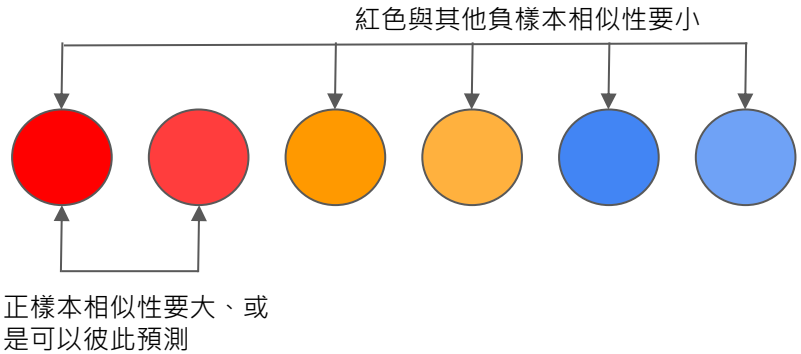
Self-Supervised Learning論文介紹：BarlowTwins

回憶之前介紹的自監督學習方法的理念

SimCLR、MoCo：藉由圖片變換使樣本更多元後，讓模型學習去增加正樣本的特徵相似度、減少負樣本的特徵相似度，以此方式學習圖片中保存的重要特徵。

BYOL：一樣採用圖片變換後，讓正樣本之間做特徵向量預測，若是能預測準確則代表前期的抽取特徵有能力抽取重點特徵。

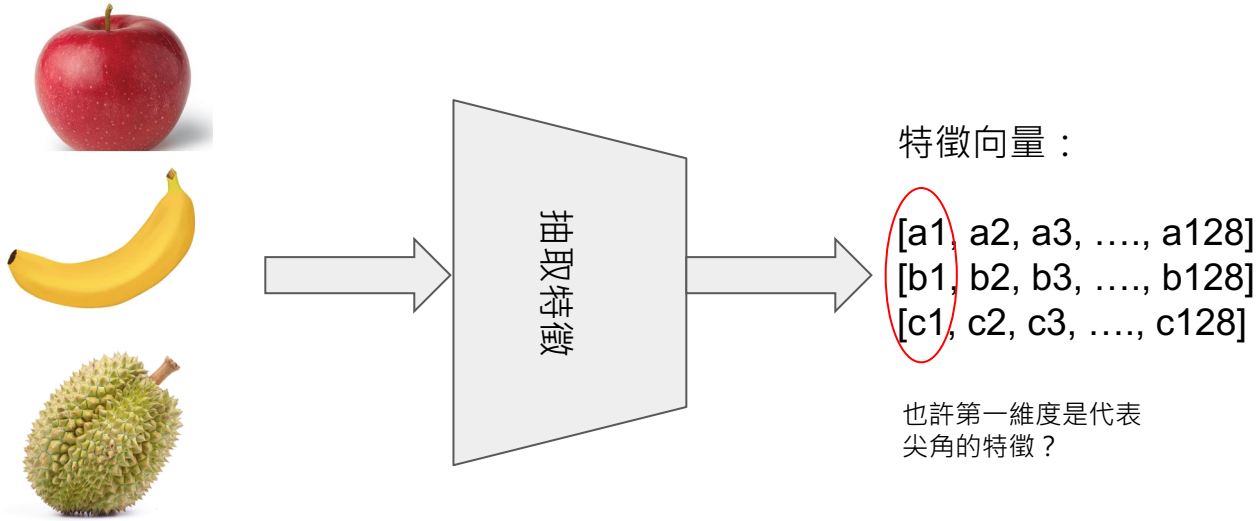
=>以上方法著眼點都是**“樣本”之間的比較**！



樣本對比之外

除了樣本之間的對比之外，還有其他可以做對比的東西嗎？

可以思考一下，圖片經過變換後，再經過模型所抽取的特徵向量，每個維度的意義是什麼？有沒有可能每個維度表示著某種特徵？



向量維度的對比

BarlowTwins這篇論文所提出的想法不再是樣本之間的比較（正樣本相似、負樣本不相似），而是對向量維度之間的對比，也就是假設每個向量維度所代表的意義是獨立且需要一致性的。

舉例來說，一張圖片經由圖片變換後生成兩張變換後圖片，其每一個維度的相似性應該不至於太大。

左右哪種輸出結果比較合理？

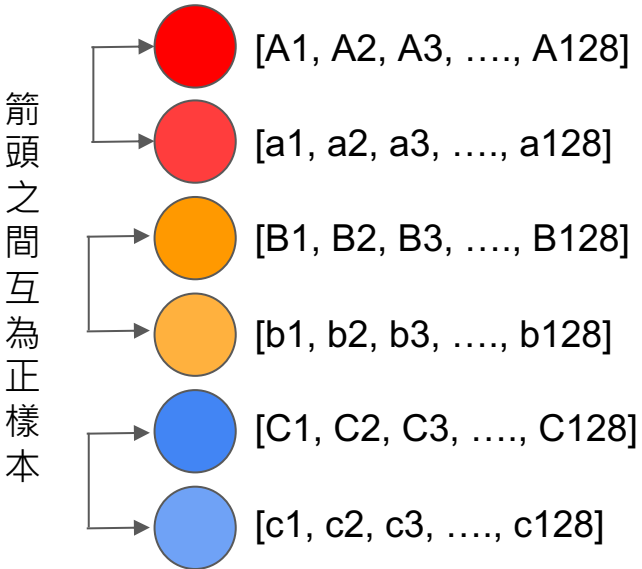
=> [0.2, 0.8, -0.9, 0]　　=> [0.2, 0.8, -0.9, 0]

=> [0.1, 0.9, -0.7, 0.1]　　=> [-0.6, 0.1, 0, 0.8]

多個向量做維度間的對比

上一張投影片是以一張圖片舉例，但是做訓練時batch size不會只有1，那多個圖片的特徵向量怎麼做呢？

對不同樣本的同一維度的值做合併後，再去比較向量的相似性。



[A1, B1, C1]、[a1, b1, c1]彼此之間向量要盡可能相似
[A2, B2, C2]、[a2, b2, c2]彼此之間向量要盡可能相似
（因為假設同一維度都代表某一特徵）
其他3~128維度以此類推

而維度（特徵）間最好是彼此獨立，所以
[A1, B1, C1]與[A2, B2, C2]、[A3, B3, C3].....
[A1, B1, C1]與[a2, a2, a2]、[a3, a3, a3].....
最好都是盡可能不相似

BarlowTwins理念

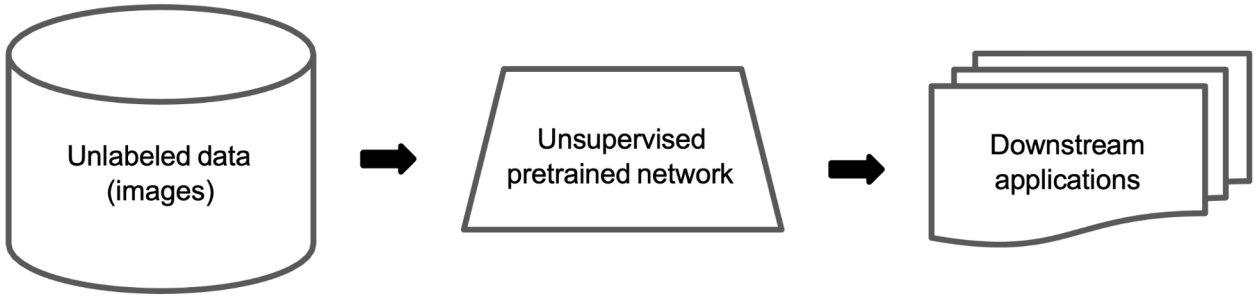
以上就是本篇論文（BarlowTwins）的理念，不像之前的方法是對於正負樣本之間做對比，而是對於特徵維度之間做對比。

後面投影片會對此論文再做講解。

BarlowTwins目標

BarlowTwins全名是“Barlow Twins: Self-Supervised Learning via Redundancy Reduction”，是由Facebook AI研究團隊發表在ICML2021的論文。

BarlowTwins目標與其他自監督學習一樣，都要利用**無標註的資料**訓練模型成為好的特徵抽取器。而這特徵抽取器可以遷移到其他的電腦視覺任務。



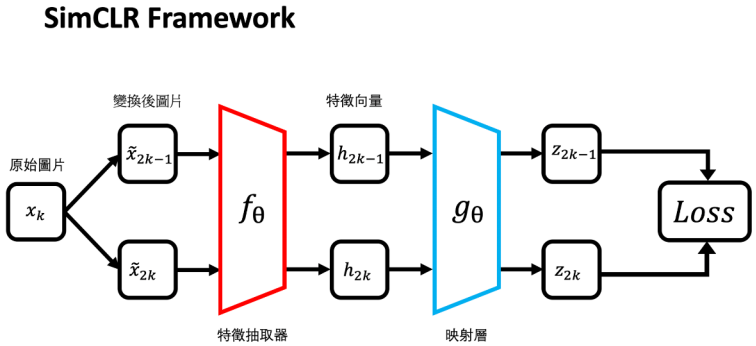
BarlowTwins貢獻

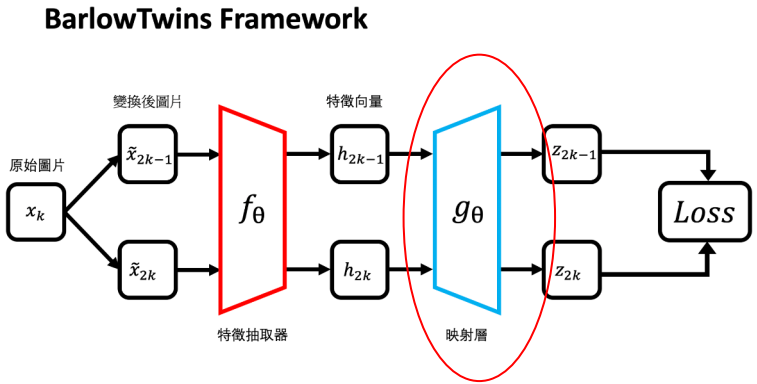
此篇論文提出了以**向量維度作為對比的方式**，不同於使用樣本的對比學習演算法，也因此BarlowTwins不太需要在乎樣本數目的多寡，讓小batch size的學習也可以有不錯的表現。

另外此篇論文提出一個不同的映射層架構，不同於以往的降維映射層，這篇論文使用**增維的映射層**，並且做實驗證明這種設計有利於後續的圖像分類任務。

BarlowTwins模型架構

架構上與SimCLR相似，皆使用對稱性的模型架構，分別為使用gradient更新的特徵抽取模型與映射層（f_θ、g_θ），差異在映射層的輸出維度是放大的。

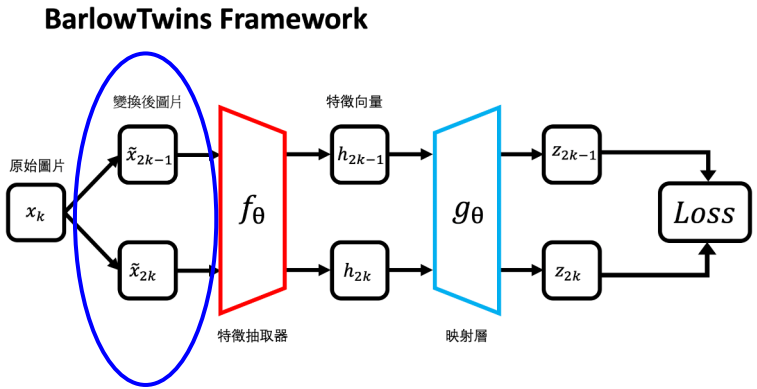




BarlowTwins第一階段的圖片變換

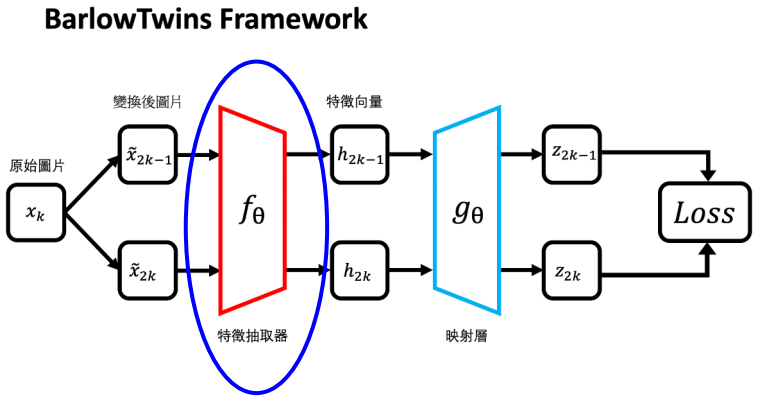
與其他對比學習演算法（SimCLR, MoCo, BYOL）一樣，這一階段做圖片變換，目的是讓圖片多樣化並保有重要特徵讓特徵抽取模型學習。

圖片變換也是以裁切（RandomResizedCrop）與顏色更動（ColorJitter）為主。



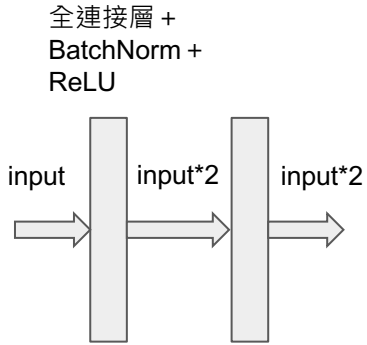
BarlowTwins第二階段的特徵抽取

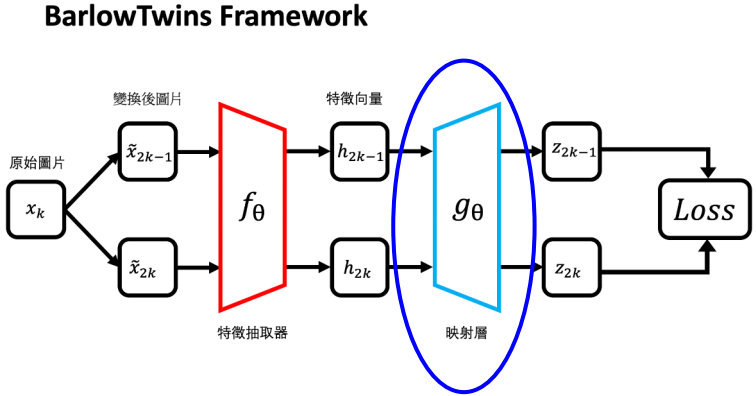
BarlowTwins使用的特徵抽取器是採取CNN理念的ResNet架構，只有單一模型，更新方式是gradient更新。



BarlowTwins第三階段的映射層

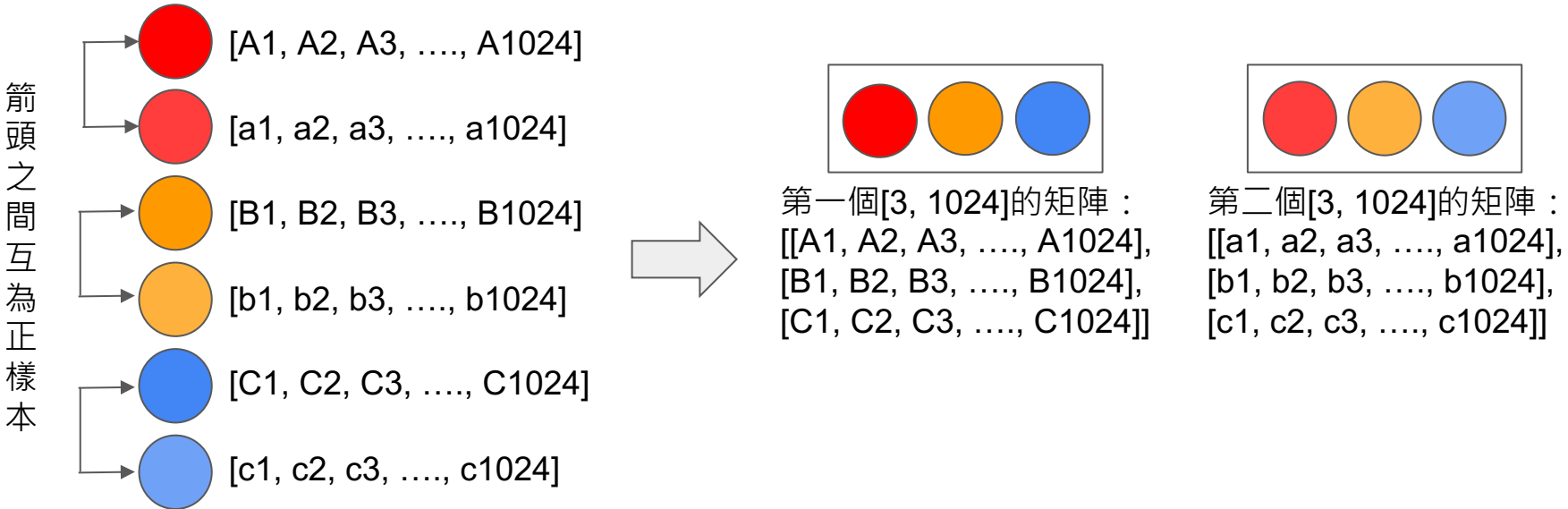
在這一階段BarlowTwins採取跟其他演算法不同的映射層邏輯，在SimCLR, MoCo, BYOL中皆是降維到128維度，而BarlowTwins是增加維度（通常會是輸入維度的兩倍以上，ResNet18就會是512*2以上的維度，ResNet50就是2048*2以上的維度）。





BarlowTwins第四階段的Loss計算

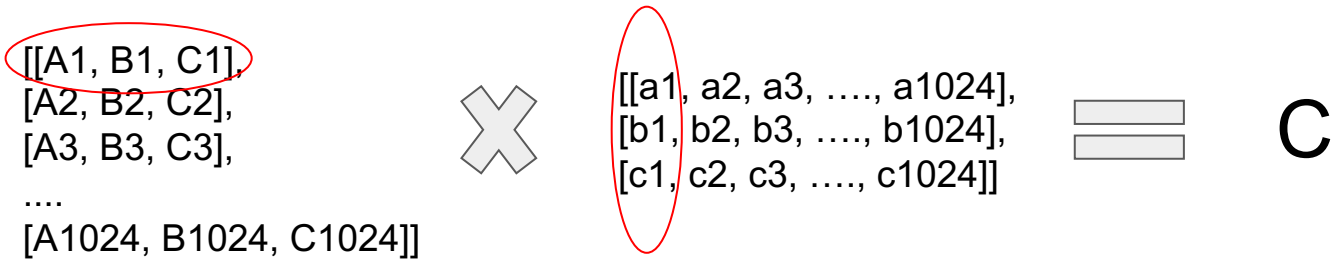
如同P5的圖示，BarlowTwins要計算的是維度間的cosine similarity，以batch=3，映射層輸出維度1024為例，第一、二個圖片變換後的batch經過特徵抽取模型和映射層後會分別得到[3, 1024], [3, 1024]的矩陣。



BarlowTwins第四階段的Loss計算

兩個矩陣做normalize後，其中一個矩陣做Transpose（變成[1024, 3]）再做矩陣乘法就等於向量維度間的cosine similarity（變成[1024, 1024]的矩陣）。

這一個[1024, 1024]的矩陣我們稱為C。



BarlowTwins第四階段的Loss計算

而很明顯的，C矩陣的對角線元素是越接近1越好，其餘越接近0越好。

對角線接近1的原因：維度所代表的特徵要一致。
非對角線接近0的原因：不同維度代表的特徵盡可能獨立。

λ是一個超參數，設置上通常會再0.005~0.01之間（但也可能需要去試）。

$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}}$$

BarlowTwins整體演算法流程

Algorithm 1 PyTorch-style pseudocode for Barlow Twins.

```
# f: encoder network
# lambda: weight on the off-diagonal terms
# N: batch size
# D: dimensionality of the embeddings
#
# mm: matrix-matrix multiplication
# off_diagonal: off-diagonal elements of a matrix
# eye: identity matrix

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    y_a, y_b = augment(x)

    # compute embeddings
    z_a = f(y_a) # NxD
    z_b = f(y_b) # NxD

    # normalize repr. along the batch dimension
    z_a_norm = (z_a - z_a.mean(0)) / z_a.std(0) # NxD
    z_b_norm = (z_b - z_b.mean(0)) / z_b.std(0) # NxD

    # cross-correlation matrix
    c = mm(z_a_norm.T, z_b_norm) / N # DxD

    # loss
    c_diff = (c - eye(D)).pow(2) # DxD
    # multiply off-diagonal elems of c_diff by lambda
    off_diagonal(c_diff).mul_(lambda)
    loss = c_diff.sum()

    # optimization step
    loss.backward()
    optimizer.step()
```

評估模型效果的方式(same as SimCLR)

在Self-Supervised Learning訓練中，因為不使用標註資料，所以訓練過程無法知道準確率如何，只能觀察loss是否下降，那要怎樣評估模型的好壞呢？

Linear Evaluation Protocol：在SSL訓練結束後，我們會把**特徵抽取模型f的參數凍結**，並在後面接上全連接層做分類任務，這個階段我們會用**有標註**的資料訓練全連接層，這樣的方式可以評估模型f的特徵抽取能力好不好。

Fine-tune：**模型f的參數可以一起更新**，但通常這樣的評估方式會只使用小部分的標註資料，可能10%或1%訓練資料。

Transfer Pretrained Model：把在A資料集訓練好的模型f當作起始參數用在其他的資料集B上。

BarlowTwins實驗：Linear Evaluation

只使用標註資料訓練分類層（全連接層），而特徵抽取模型f_θ不會更新參數。

由下圖的結果可以看到，BarlowTwins比SimCLR, MoCo好但比BYOL跟SwAV演算法差。

Method	Top-1	Top-5
Supervised	76.5	
MoCo	60.6	
PIRL	63.6	-
SimCLR	69.3	89.0
MoCo v2	71.1	90.1
SimSiam	71.3	-
SwAV (w/o multi-crop)	71.8	-
BYOL	74.3	91.6
SwAV	75.3	-
Barlow Twins (ours)	73.2	91.0

BarlowTwins實驗：Fine Tune

這一個評估方法是使用部分的有標註資料來訓練特徵抽取模型和分類層，模擬少量標註的情形。

由下圖實驗結果可以看到，只使用1%, 10%的資料時，BarlowTwins的表現算是最好的。

Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised	25.4	56.4	48.4	80.4
PIRL	-	-	57.2	83.8
SIMCLR	48.3	65.6	75.5	87.8
BYOL	53.2	68.8	78.4	89.0
SWAV	53.9	**70.2**	78.5	**89.9**
BARLOW TWINS (ours)	**55.0**	69.7	**79.2**	89.3

BarlowTwins實驗：Transfer Learning

把使用BarlowTwins演算法所訓練的特徵抽取模型用在其他的圖像分類資料集上面，來測試特徵抽取的遷移能力。
在這篇論文中的Transfer Learning實驗並沒有更新特徵抽取模型，所以是像Linear Evaluation的方式但做在其他資料集上面。

由下圖實驗結果來看，BarlowTwins並不是最好的。

Method	Places-205	VOC07	iNat18
Supervised	53.2	87.5	46.7
SimCLR	52.5	85.5	37.2
MoCo-v2	51.8	86.4	38.6
SwAV (w/o multi-crop)	52.8	86.4	39.5
SwAV	56.7	88.9	48.6
BYOL	54.0	86.6	47.6
BARLOW TWINS (ours)	54.1	86.2	46.5

BarlowTwins實驗：Loss Function探討

BarlowTwins演算法的loss有兩項，前面一項是在計算同一維度的一致性，而後面一項是在算不同維度的獨立性。

由下圖可以看到如果loss function只使用前面一項，會讓準確率大幅下降，而只用後面一項訓練是無效的。

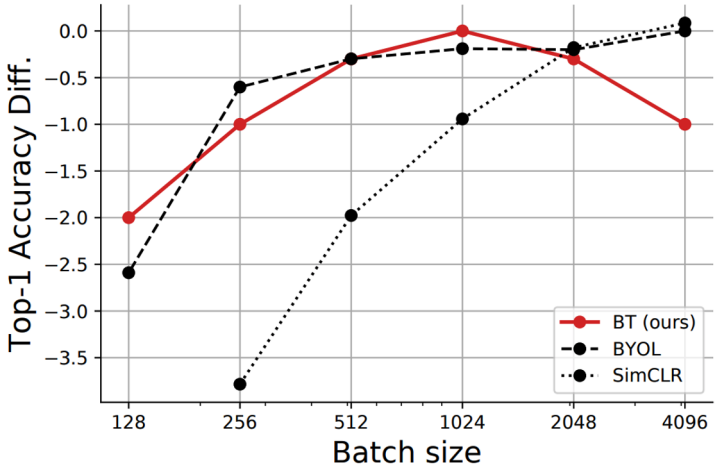
$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}{}^2}_{\text{redundancy reduction term}}$$

Loss function	Top-1	Top-5
Baseline	71.4	90.2
Only invariance term (on-diag term)	57.3	80.5
Only red. red. term (off-diag term)	0.1	0.5

BarlowTwins實驗：Batch Size的影響

我們之前介紹的SimCLR, MoCo, BYOL演算法都是batch size越大效果越好，但在此篇論文中卻不是這樣。

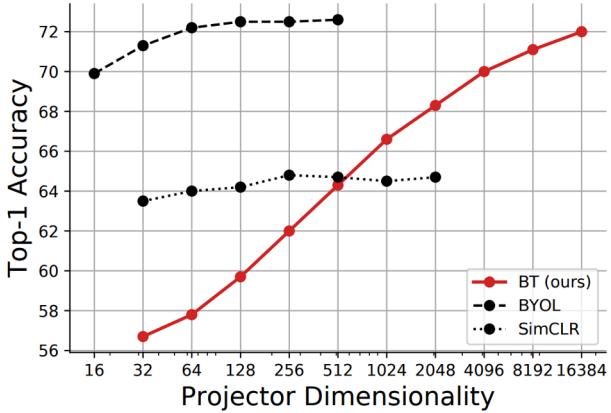
因此在使用此種方法的時候，batch size可能會是一個需要調整的超參數。



BarlowTwins實驗：高維度映射

BarlowTwins使用了跟其他方法不一樣的映射層，可以看到SimCLR, BYOL演算法使用降維度的映射，在不同的維度間差異不是很大，而BarlowTwins使用增加維度的映射層，進步的幅度是可以隨著映射層維度增大而提高。

但要注意的是，隨著映射層的輸出維度增加，整體參數量也會大幅增加，所以訓練時間也會變得更長。



BarlowTwins：結論

此篇論文提出一個不同的對比方式，不同於之前主流的正負樣本間的對比，這裡使用向量維度間的對比，想法是讓各個維度代表的特徵一致且彼此獨立。

而BarlowTwins對比方式可以使用更大的映射維度來達成更好的準確率，但受限於硬體資源所以16000以上的映射維度目前很難達成。

從實驗結果來看，BarlowTwins演算法並不特別突出，但此種對比方法相對來說很新，可能需要未來更多的研究。