

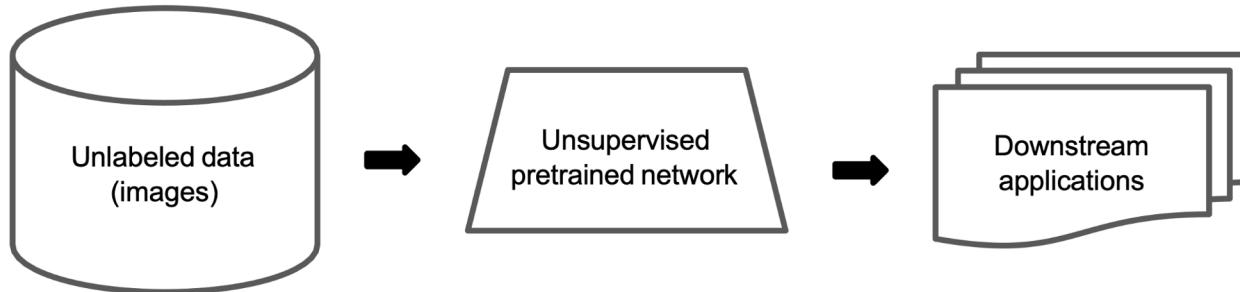
Self-Supervised Learning論文介紹： SimCLR

SimCLR

SimCLR論文全名“*A Simple Framework for Contrastive Learning of Visual Representations*”是Google研究團隊發表在ICML2020的論文。

SimCLR目標是利用無標註的圖片訓練模型成為好的特徵抽取器，而這個特徵抽取器能運用在其他的下游任務，像是圖像分類或是其他電腦視覺任務。

自監督特徵學習的想法：



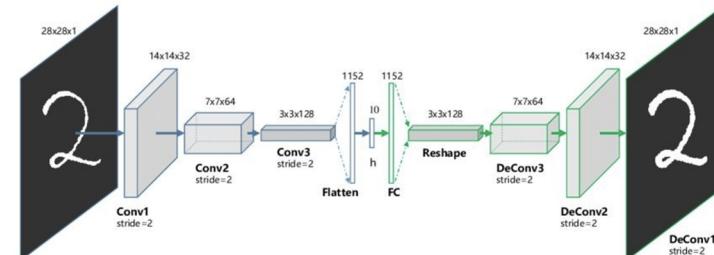
SimCLR並非傳統的無監督式學習方式

SimCLR既不是生成類別（ generative modeling ）也不是利用間接任務（ pretext task ）的訓練方式。

生成類別的缺點：生成高度還原的圖片對於學習特徵並不必要。

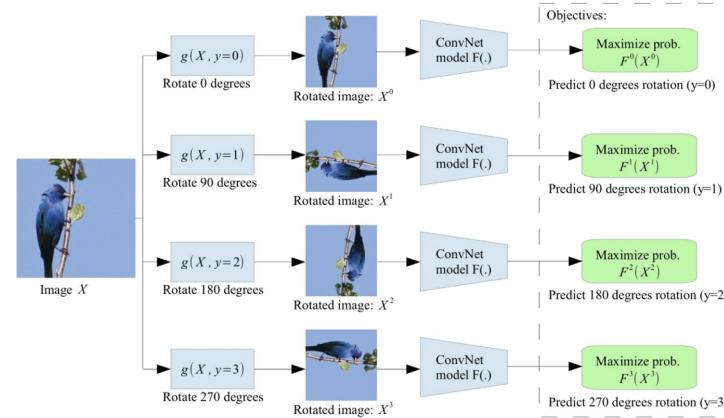
間接任務的缺點：相對來說需要更多人為的知識與前處理。

generative modeling



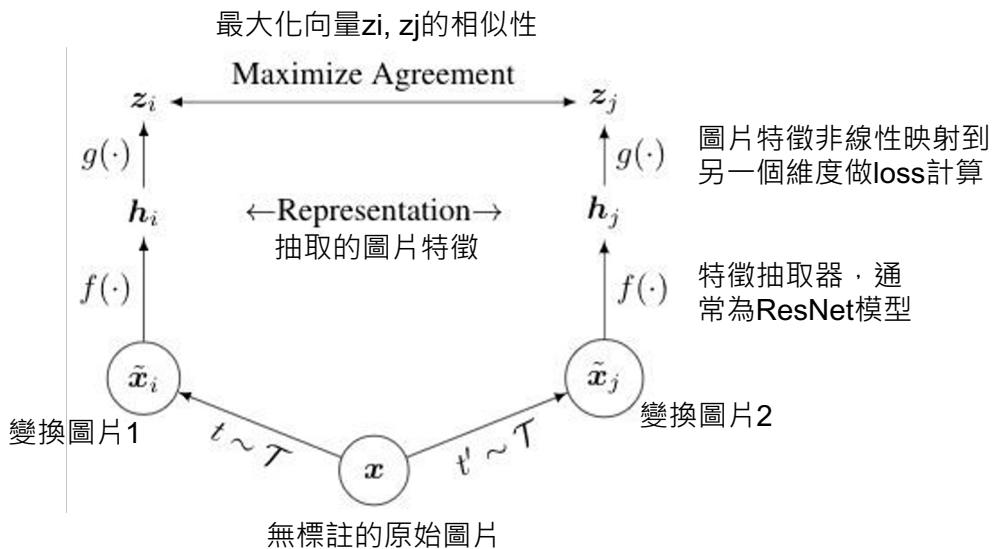
Autoencoder

pretext task



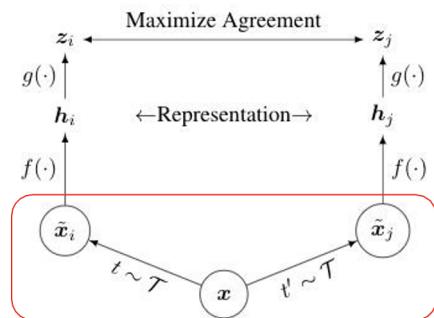
SimCLR的模型框架

SimCLR提出一個簡單的訓練框架，其中包含隨機性的圖片變換 T 、一個特徵抽取模型 f 和一個映射層 g 。

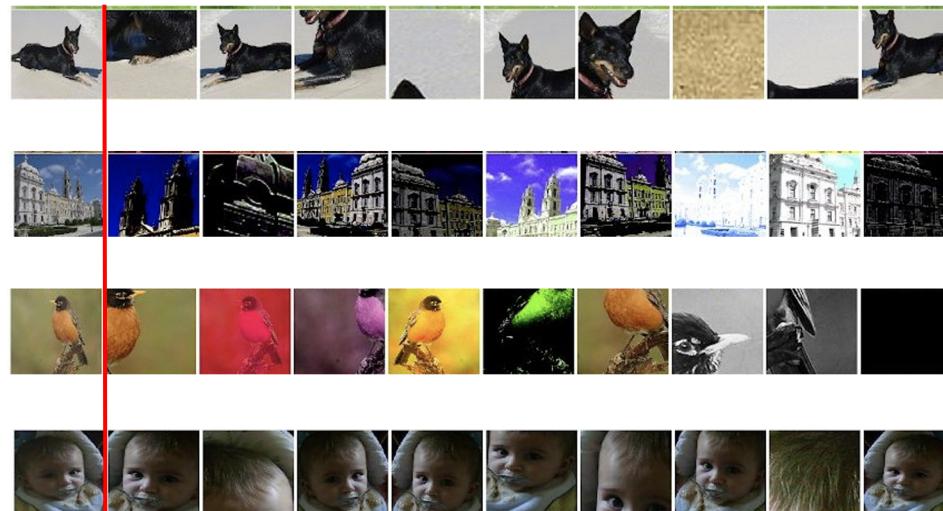


SimCLR第一階段的圖片變換

圖片變換目的是改動圖片並保有特徵，讓模型學習圖片不變性。

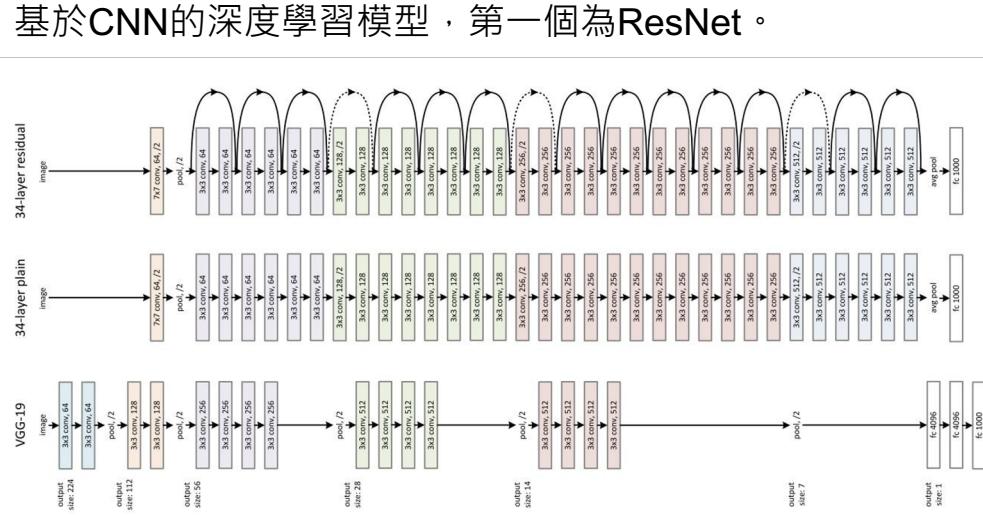
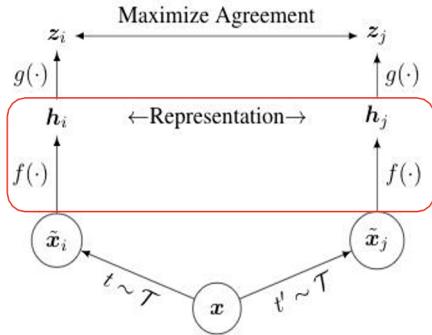


原圖 經過變換後的圖片（因為變換有隨機性，所以都會不一樣）



SimCLR第二階段的特徵抽取

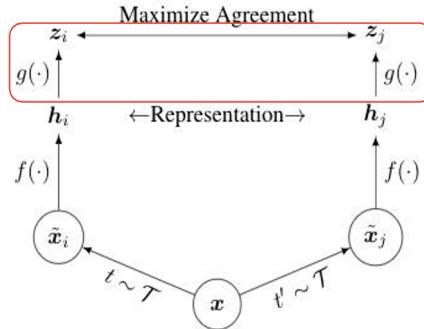
f 是用來抽取圖片特徵的模型，在SimCLR中是使用ResNet，但其他的模型也適用於這個學習框架。



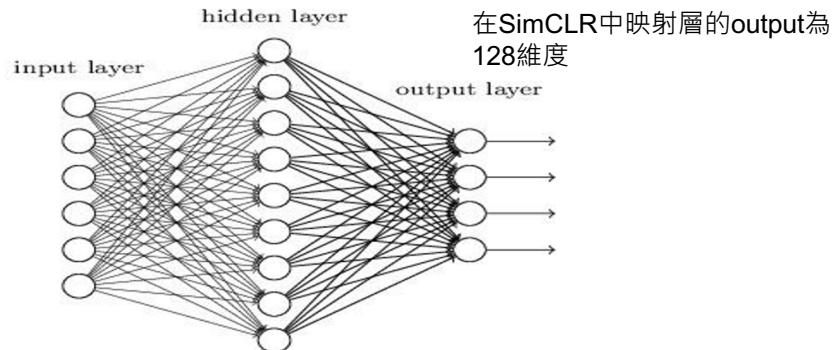
SimCLR第三階段的特徵映射

g 是一個映射層，把圖片特徵映射到一個較低的維度，並作loss的計算。

SimCLR設計的映射層為兩層的非線性全連接層，功能上可以消除不必要的圖片特徵。

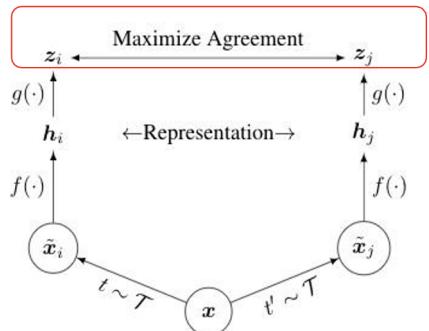


以ResNet18來說input維度是512，
而ResNet50是2048



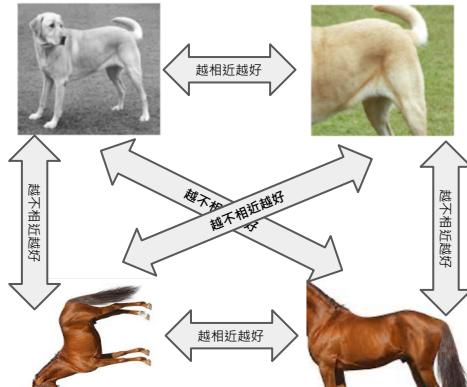
SimCLR第四階段的Loss計算

在這一階段 z_i, z_j 在向量空間上理當要是相似的，因為都是由相同原始圖片 x 所衍生，我們稱為正樣本對 (positive pair)，為了使loss降低，模型需要讓正樣本向量相似度增加、負樣本相似度減低。這裡的負樣本就是指其他所有與正樣本對不同原始圖片的樣本。



SimCLR第四階段的損失計算

舉例來說batch N=2，裡面有一張狗跟馬的圖片，這兩張圖片經由變換後都會各自生成一組正樣本對，正樣本對之間要越相近越好，並且跟其他的樣本越遠越好。



SimCLR第四階段的損失計算

S : cosine similarity , 越接近1表示越相近

τ : 常數 , 通常為0.1

N : batch大小

$$s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \|\mathbf{z}_j\|)$$

define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$

分子越大越好

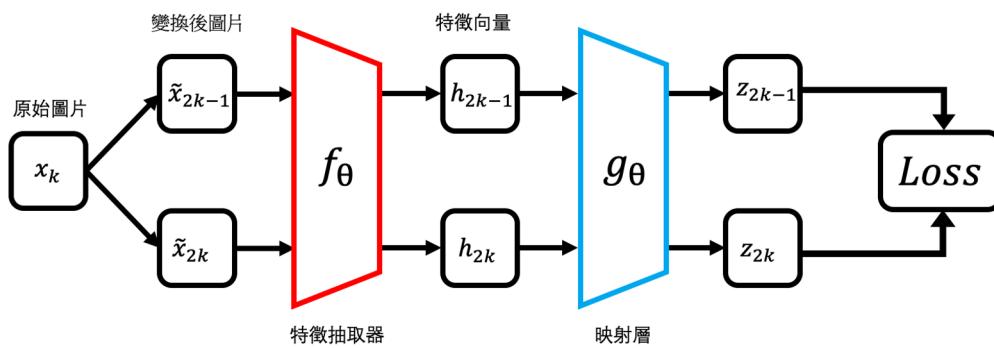
$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$$

2k-1, 2k互為正樣本

SimCLR演算法

訓練完成後只會保留抽取特徵的模型 f 。

SimCLR Framework



Algorithm 1 SimCLR's main learning algorithm.

```

input: batch size  $N$ , constant  $\tau$ , structure of  $f, g, \mathcal{T}$ .
for sampled minibatch  $\{\mathbf{x}_k\}_{k=1}^N$  do
    for all  $k \in \{1, \dots, N\}$  do
        draw two augmentation functions  $t \sim \mathcal{T}, t' \sim \mathcal{T}$ 
        # the first augmentation
         $\tilde{\mathbf{x}}_{2k-1} = t(\mathbf{x}_k)$ 
         $\mathbf{h}_{2k-1} = f(\tilde{\mathbf{x}}_{2k-1})$                                 # representation
         $\mathbf{z}_{2k-1} = g(\mathbf{h}_{2k-1})$                             # projection
        # the second augmentation
         $\tilde{\mathbf{x}}_{2k} = t'(\mathbf{x}_k)$ 
         $\mathbf{h}_{2k} = f(\tilde{\mathbf{x}}_{2k})$                                 # representation
         $\mathbf{z}_{2k} = g(\mathbf{h}_{2k})$                             # projection
    end for
    for all  $i \in \{1, \dots, 2N\}$  and  $j \in \{1, \dots, 2N\}$  do
         $s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \|\mathbf{z}_j\|)$       # pairwise similarity
    end for
    define  $\ell(i, j)$  as  $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$ 
     $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$ 
    update networks  $f$  and  $g$  to minimize  $\mathcal{L}$ 
end for
return encoder network  $f(\cdot)$ , and throw away  $g(\cdot)$ 

```

評估模型效果的方式

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**模型f的參數凍結**，並在模型f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。

SimCLR各階段細節

到這邊大家應該對整體SimCLR演算法有初步理解了，後面會繼續針對各個訓練階段補充一些細節。

1. 圖片變換的使用效果
2. 特徵抽取模型大小差異
3. 映射層的必要性
4. Loss跟batch size之間的關係

常見的圖片變換

在原論文中展示了一些常見的圖片變換，如下圖。

大家可以試著猜猜哪些是對SimCLR好的圖片變換方式。



(a) Original



(b) Crop and resize



(c) Crop, resize (and flip)



(d) Color distort. (drop)



(e) Color distort. (jitter)



(f) Rotate { $90^\circ, 180^\circ, 270^\circ$ }



(g) Cutout



(h) Gaussian noise



(i) Gaussian blur

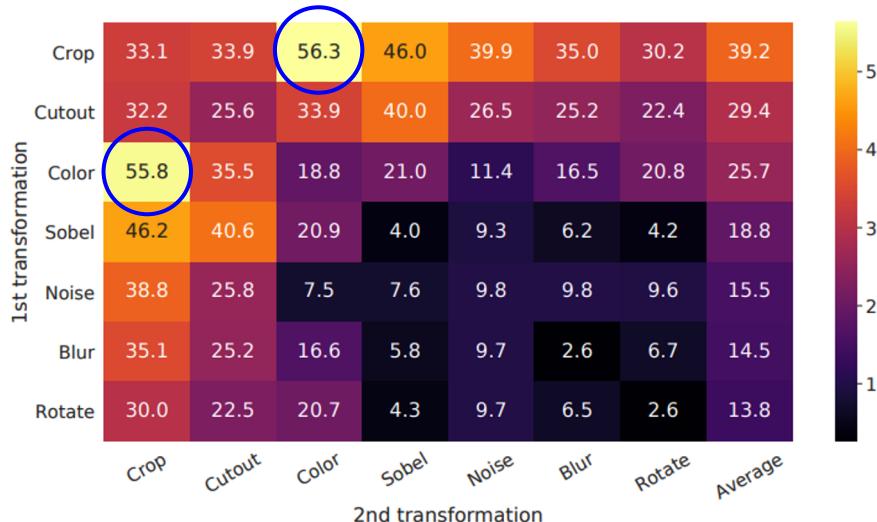


(j) Sobel filtering

最適合SimCLR的圖片變換

SimCLR實驗了七種圖片變換，並且也實驗兩種變換的組合，結果顯示裁切跟顏色抖動是最好的，而加在一起使用又遠遠勝過單一變換的效果。在實作上也是以裁切與顏色抖動這兩個變換為主。

另外可以看到Rotate變換基本上無法突顯出圖片的重要特徵，效果也是最差的。



最適合SimCLR圖片的強度

論文中進一步比較顏色抖動的強度對後續分類準確率的影響。結果發現，強度越高效果越好，如果使用太低的強度會嚴重影響分類的準確率。

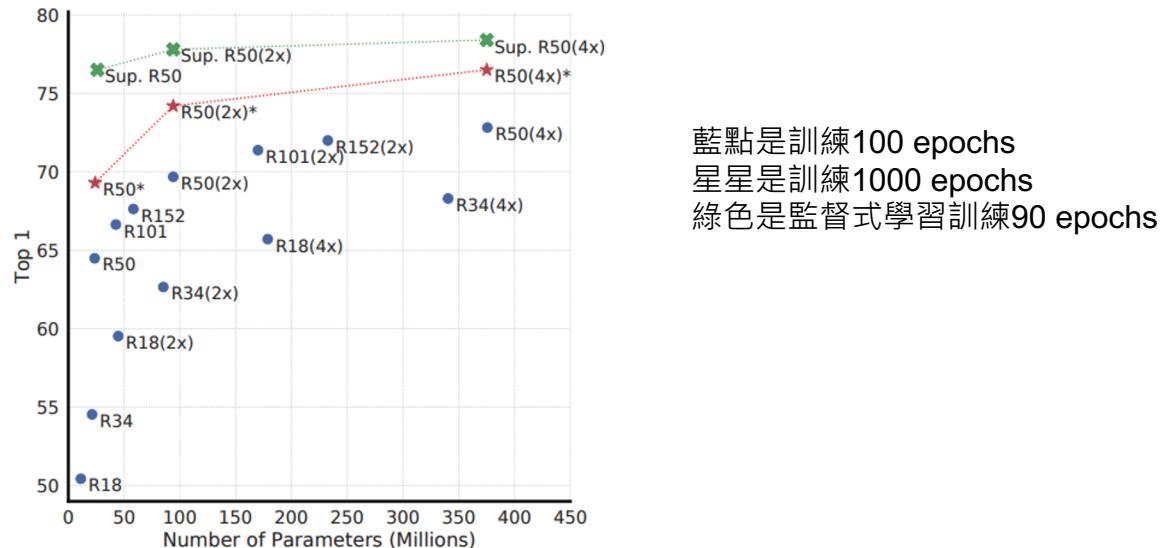
相反的，顏色抖動對於監督式學習的影響就比較小，並且是傾向低強度的變換。

Methods	Color distortion strength				
	1/8	1/4	1/2	1	1 (+Blur)
SimCLR	59.6	61.0	62.6	63.2	64.5
Supervised	77.0	76.7	76.5	75.7	75.4

特徵抽取模型大小的效果

如圖顯示不同的模型大小對SimCLR的影響，越大的模型準確率越高，進步幅度也比監督式學習明顯。

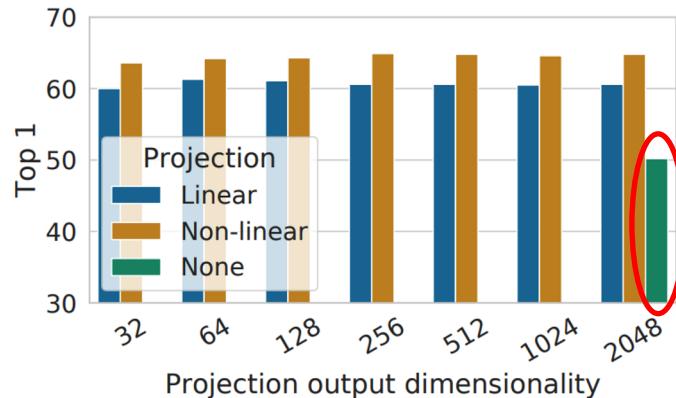
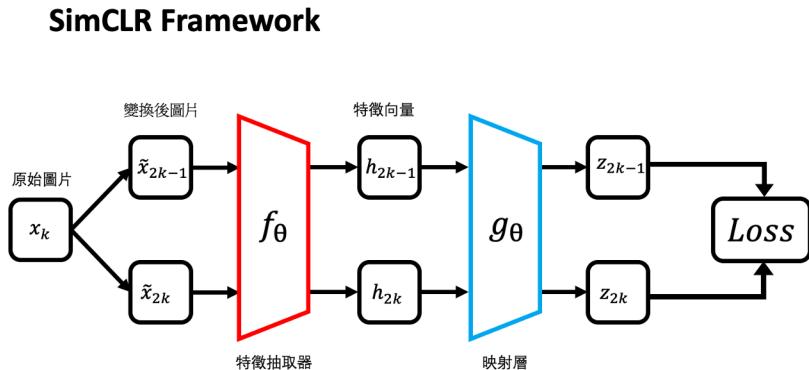
在給予大模型與長時間訓練的基礎下，自監督學習是可以堪比監督式學習的。



SimCLR中映射層的作用

大家可能會好奇，SimCLR架構上為什麼需要映射層？難道不能直接把抽取完的特徵向量做loss的計算嗎？我們來看一下有沒有映射層的差異。

結果顯示，映射層可以讓特徵抽取模型的能力更好，模型架構不用映射層（綠色）會比使用映射層還差10%以上，而使用非線性的映射層又比線性的好大約4%。

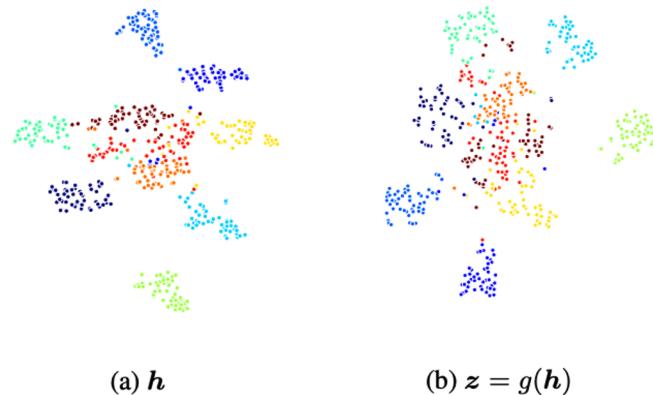
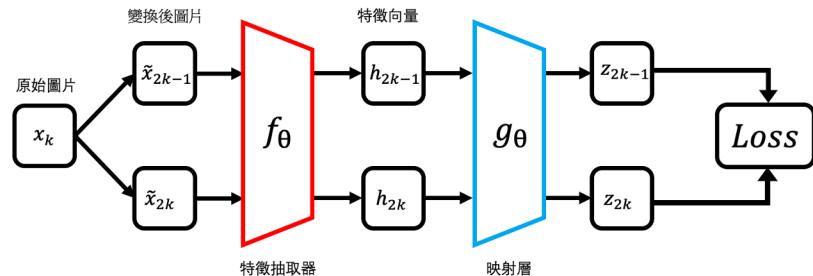


SimCLR中映射層的作用

我們知道非線性映射層是有效的了，那為什麼最後保留的特徵抽取器只有 f 而不是 $f+g$ 的整個模型呢？換句話說，我們要的圖片特徵為什麼是 h 而不 z 呢？

實驗結果顯示，使用 h 當作圖片特徵向量的準確率比使用 z 當圖片特徵向量還高 10%以上，右下圖顯示 h 能更好的做分類任務。

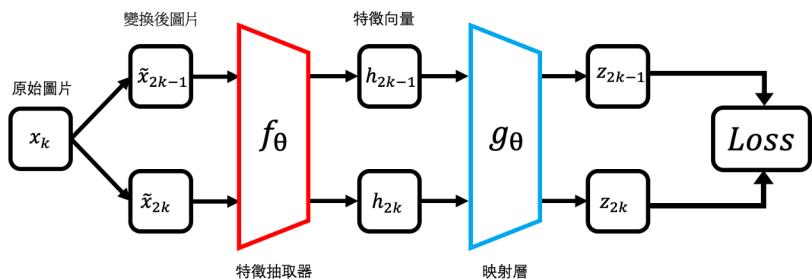
SimCLR Framework



SimCLR中映射層的作用

原論文提出一個假說，認為映射層會把圖片變換的資訊過濾掉，所以在做後續的分類任務時需要用 h 而非 z 。

SimCLR Framework



SimCLR中映射層的作用

對此SimCLR做實驗來驗證他們的假說，他們設計四組實驗，每組實驗做的圖片變換為：

1. (0.8機率color/0.2機率grayscale)
2. (旋轉0, 90, 180, 270度，機率各0.25)
3. (0.5機率做corrupted)
4. (0.5機率做soble filtered)

然後拿 h 和 z 當做“圖片變換分類任務”的input，結果顯示使用 h 可以分類的很準，而使用 z 的準確率跟亂猜差不多（除了1以外）。

SimCLR中映射層的作用

實驗結果如下。

結論就是模型架構上要加入非線性映射層，並且使用 \mathbf{h} 來代表圖片特徵。

What to predict?	Random guess	Representation	
		\mathbf{h}	$g(\mathbf{h})$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

Table 3. Accuracy of training additional MLPs on different representations to predict the transformation applied. Other than crop and color augmentation, we additionally and independently add rotation (one of $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$), Gaussian noise, and Sobel filtering transformation during the pretraining for the last three rows. Both \mathbf{h} and $g(\mathbf{h})$ are of the same dimensionality, i.e. 2048.

SimCLR中loss function與batch size

原論文中比較了其他對比學習的loss function，實驗結果顯示SimCLR用的loss function(NT-Xent)所得到的準確率是最好的。可以看到其他兩個loss都只有一個負樣本 \mathbf{v}^- 。

Name	Negative loss function			
NT-Xent	$\mathbf{u}^T \mathbf{v}^+ / \tau - \log \sum_{\mathbf{v} \in \{\mathbf{v}^+, \mathbf{v}^-\}} \exp(\mathbf{u}^T \mathbf{v} / \tau)$			
NT-Logistic	$\log \sigma(\mathbf{u}^T \mathbf{v}^+ / \tau) + \log \sigma(-\mathbf{u}^T \mathbf{v}^- / \tau)$			
Margin Triplet	$-\max(\mathbf{u}^T \mathbf{v}^- - \mathbf{u}^T \mathbf{v}^+ + m, 0)$			
Margin	NT-Logi.	Margin (sh)	NT-Logi.(sh)	NT-Xent
50.9	51.6	57.5	57.9	63.9

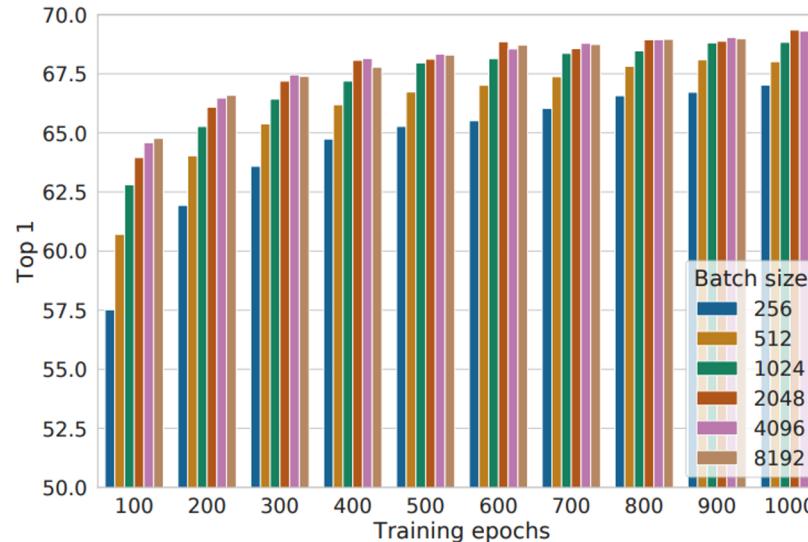
Table 4. Linear evaluation (top-1) for models trained with different loss functions. “sh” means using semi-hard negative mining.

sh: 挑選過的負樣本

SimCLR中loss function與batch size

對SimCLR的loss function來說，負樣本的數量決定了對比的效果。愈大的batch size可以帶來更多的對比樣本，也就有更好的對比效果。

相較於監督式學習，對比學習受惠於更多的trainging epochs，而不太會發生過擬合(overfitting)。



Batch size越大效果越好，如果只能用比較小的batch size，那使用更多training epochs可以稍微彌補不足。

SimCLR中loss function與batch size

SimCLR的loss function中的相似性函數是cosine similarity，如果只用內積（沒有normalize）效果會下降。而超參數 τ 也會影響準確率，實驗發現0.1是相對最好的值。

define $\ell(i, j)$ as $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
 $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$

ℓ_2 norm?	τ	Entropy	Contrastive acc.	Top 1
Yes	0.05	1.0	90.5	59.7
	0.1	4.5	87.8	64.4
	0.5	8.2	68.2	60.7
	1	8.3	59.1	58.0
No	10	0.5	91.7	57.2
	100	0.5	92.1	57.0

SimCLR實驗結果比較

SimCLR與其他“self-supervised”並且是“對比學習”的演算法做比較，其中包括 Exemplar, InstDist, CPC, DIM, AMDIM, CMC, MoCo, PIRL, ...



Figure 1: Exemplary patches sampled from the STF unlabeled dataset which are later

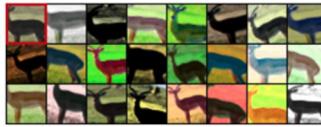
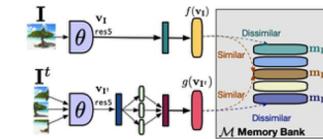
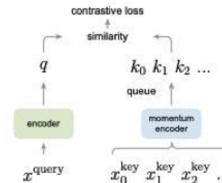
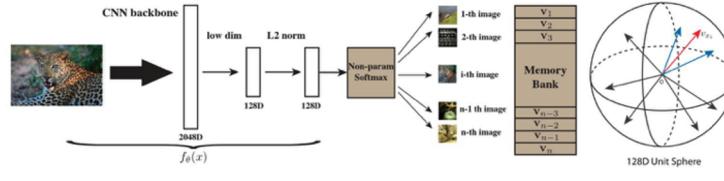
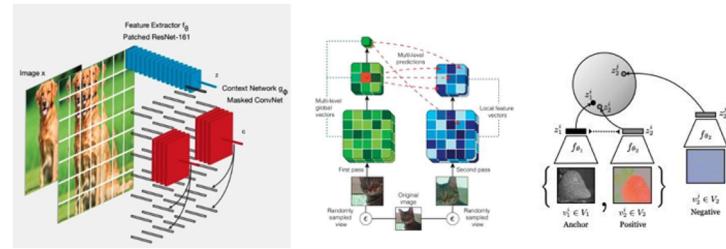


Figure 2: Several random transformations applied to one of the patches extracted from

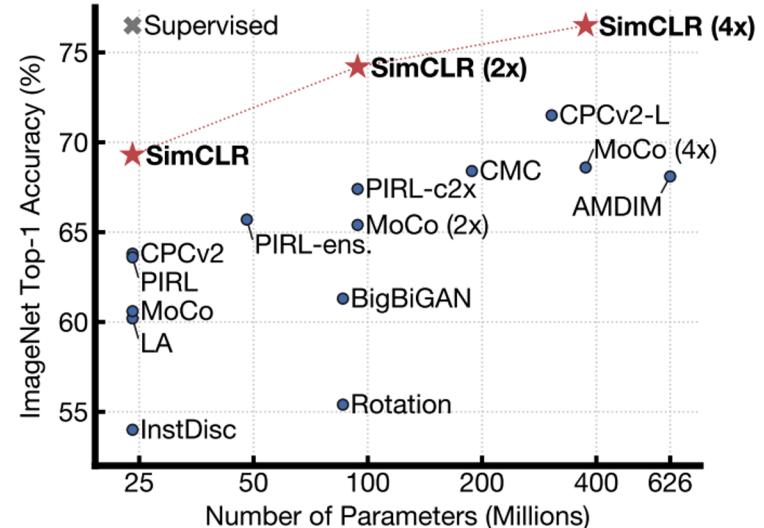


SimCLR實驗結果比較 -Linear Evaluation

SimCLR在Linear evaluation (模型f 參數不更新) 下，分類的準確率比其他的方法還要好，並且在使用大模型下，可以相當於監督式學習的準確率。

Method	Architecture	Param (M)	Top 1	Top 5
<i>Methods using ResNet-50:</i>				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	69.3	89.0
<i>Methods using other architectures:</i>				
Rotation	RevNet-50 (4x)	86	55.4	-
BigBiGAN	RevNet-50 (4x)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2x)	188	68.4	88.2
MoCo	ResNet-50 (4x)	375	68.6	-
CPC v2	ResNet-161 (*)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2x)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4x)	375	76.5	93.2

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.



SimCLR實驗結果比較 -Semi-Supervised Learning

如果只用1%和10%的有標註圖片來更新模型f 跟分類層，SimCLR的表現也比其他的semi-supervised方法還要好。

Method	Architecture	Label fraction		
		1%	10%	Top 5
Supervised baseline	ResNet-50	48.4	80.4	
<i>Methods using other label-propagation:</i>				
Pseudo-label	ResNet-50	51.6	82.4	
VAT+Entropy Min.	ResNet-50	47.0	83.4	
UDA (w. RandAug)	ResNet-50	-	88.5	
FixMatch (w. RandAug)	ResNet-50	-	89.1	
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2	
<i>Methods using representation learning only:</i>				
InstDisc	ResNet-50	39.2	77.4	
BigBiGAN	RevNet-50 (4×)	55.2	78.8	
PIRL	ResNet-50	57.2	83.8	
CPC v2	ResNet-161(*)	77.9	91.2	
SimCLR (ours)	ResNet-50	75.5	87.8	
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2	
SimCLR (ours)	ResNet-50 (4×)	85.8	92.6	

Table 7. ImageNet accuracy of models trained with few labels.

SimCLR實驗結果比較 -Semi-Supervised Learning

在100% fine-tuning的情況下，有SimCLR的pretrained model當作初始參數的話，可以比普通的supervised learning還要好2%左右。

Architecture	Label fraction					
	1%		10%		100%	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
ResNet-50	49.4	76.6	66.1	88.1	76.0	93.1
ResNet-50 (2×)	59.4	83.7	71.8	91.2	79.1	94.8
ResNet-50 (4×)	64.1	86.6	74.8	92.8	80.4	95.4
Supervised ResNet-50 (4x)					78.4	94.2

SimCLR實驗結果比較 -Transfer learning

把在ImageNet訓練完的pretrained model遷移到其他的資料集上面，SimCLR的pretrained model表現比Supervised learning的pretrained model還要好一些。

模型為ResNet-50 (4x)

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
<i>Linear evaluation:</i>												
SimCLR (ours)	76.9	95.3	80.2	48.4	65.9	60.0	61.2	84.2	78.9	89.2	93.9	95.0
Supervised	75.2	95.7	81.2	56.4	64.9	68.8	63.8	83.8	78.7	92.3	94.1	94.2
<i>Fine-tuned:</i>												
SimCLR (ours)	89.4	98.6	89.0	78.2	68.1	92.1	87.0	86.6	77.8	92.1	94.1	97.6
Supervised	88.7	98.3	88.7	77.8	67.0	91.4	88.0	86.5	78.8	93.2	94.2	98.0
Random init	88.3	96.0	81.9	77.0	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

SimCLR實驗結果比較 -Transfer learning

但如果是使用小模型比較的話SimCLR會比較差！(SSL需要更深度的模型)

模型為ResNet-50

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
<i>Linear evaluation:</i>												
SimCLR (ours)	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
Supervised	72.3	93.6	78.3	53.7	61.9	66.7	61.0	82.8	74.9	91.5	94.5	94.7
<i>Fine-tuned:</i>												
SimCLR (ours)	88.2	97.7	85.9	75.9	63.5	91.3	88.1	84.1	73.2	89.2	92.1	97.0
Supervised	88.3	97.5	86.4	75.8	64.3	92.1	86.0	85.0	74.6	92.1	93.3	97.6
Random init	86.9	95.9	80.2	76.1	53.6	91.4	85.9	67.3	64.8	81.5	72.6	92.0

結論

SimCLR雖然使用簡單的模型架構，卻勝過其他self-supervised learning演算法！

SimCLR透過找出適合的圖片變化 + 引入**映射層**和**大batch size**，來達到媲美監督式學習的結果！

SimCLR透過一系列研究顯示出各個訓練階段對自監督學習的重要性！

補充：SimCLR v2

SimCLR v2(NIPS2020)目的是提升**semi-supervised learning**的準確率，也就是當訓練資料只有少量的標註（1%, 10%）時，如何用SimCLR架構去達成更好的效果。

我們可以看到在SimCLR的實驗結果，1%有標註訓練資料其實並沒有很好，ResNet50大約50% Top1準確率。

Architecture	Label fraction					
	1%		10%		100%	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
ResNet-50	49.4	76.6	66.1	88.1	76.0	93.1
ResNet-50 (2×)	59.4	83.7	71.8	91.2	79.1	94.8
ResNet-50 (4×)	64.1	86.6	74.8	92.8	80.4	95.4

補充：SimCLR v2

下面是SimCLR v2的實驗結果。

左圖可以看到當有標註的訓練資料越少，使用更複雜的模型有更大的進步幅度。

右圖可以看到SimCLR v2確實在低標註的情況下大幅領先SOTA，甚至比100%的監督式學習還好。

在只有1%
labeled data的情況下，使用
更複雜的模型
進步幅度更大。

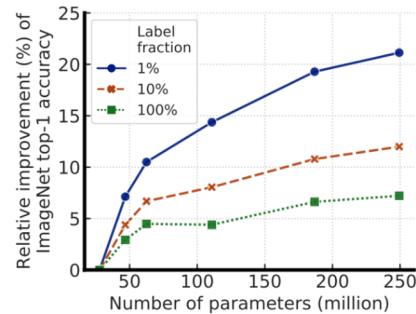


Figure 1: Bigger models yield larger gains when fine-tuning with fewer labeled examples.

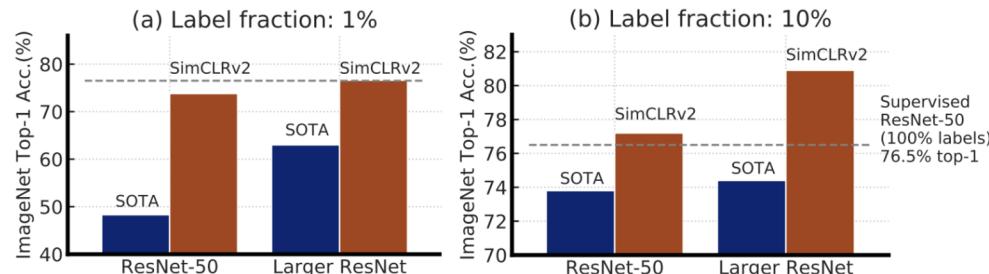
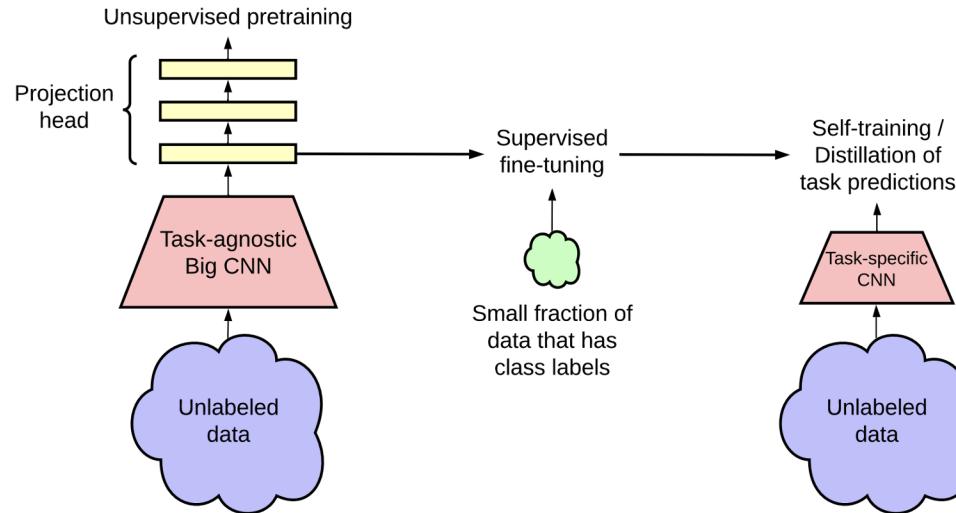


Figure 2: Top-1 accuracy of previous state-of-the-art (SOTA) methods [1, 2] and our method (SimCLRv2) on ImageNet using only 1% or 10% of the labels. Dashed line denotes fully supervised ResNet-50 trained with 100% of labels. Full comparisons in Table 3.

補充：SimCLR v2方法

1. 使用更大的模型 (ResNet50->ResNet152) ，並且多加一層的映射層。
2. Fine-tune時有包含第一層映射層 (SimCLR是整個映射層都捨棄) 。
3. Fine-tune完成的大模型再蒸餾成小模型。



補充：SimCLR v2一些實驗結果

下圖實驗顯示，可以看到監督式學習（a）幾乎沒有受惠更大的模型。
而使用SimCLR v2的方法（c），更大的模型可以達到更高的準確率，且在低標註情況下進步幅度更多。

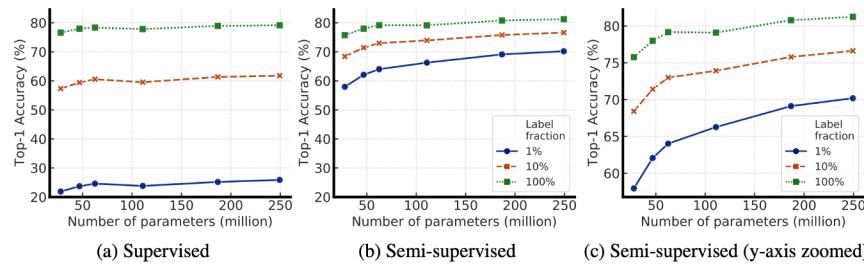


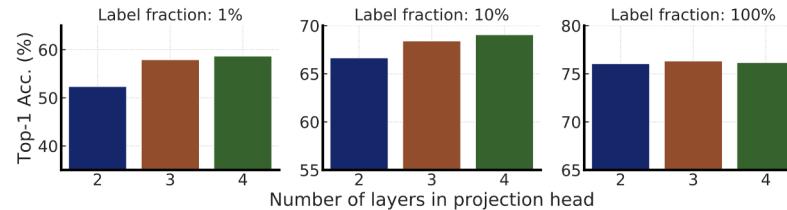
Table 1: Top-1 accuracy of fine-tuning SimCLRV2 models (on varied label fractions) or training a linear classifier on the representations. The supervised baselines are trained from scratch using all labels in 90 epochs. The parameter count only include ResNet up to final average pooling layer. For fine-tuning results with 1% and 10% labeled examples, the models include additional non-linear projection layers, which incurs additional parameter count (4M for 1× models, and 17M for 2× models). See Table H.1 for Top-5 accuracy.

Depth	Width	Use SK [28]	Param (M)	Fine-tuned on			Linear eval	Supervised
				1%	10%	100%		
50	1×	False	24	57.9	68.4	76.3	71.7	76.6
		True	35	64.5	72.1	78.7	74.6	78.5
	2×	False	94	66.3	73.9	79.1	75.6	77.8
		True	140	70.6	77.0	81.3	77.7	79.3
	101	False	43	62.1	71.4	78.2	73.6	78.0
		True	65	68.3	75.1	80.6	76.3	79.6
152	1×	False	170	69.1	75.8	80.7	77.0	78.9
		True	257	73.2	78.8	82.4	79.0	80.1
	2×	False	58	64.0	73.0	79.3	74.5	78.3
		True	89	70.0	76.5	81.3	77.2	79.9
	2×	False	233	70.2	76.6	81.1	77.4	79.1
		True	354	74.2	79.4	82.9	79.4	80.4
	3×	True	795	74.9	80.1	83.1	79.8	80.5

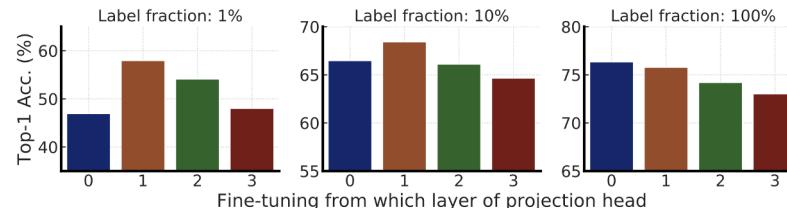
補充：SimCLR v2一些實驗結果

多增加一層映射層，並且fine tune時從第一層開始會有比較好的結果（原論文沒有提出原因，可能是由實驗嘗試得出的結果）。

圖（a）可以看到，三層映射比兩層好。圖（b）可以看到，從第一層映射的輸出 fine-tune最理想。



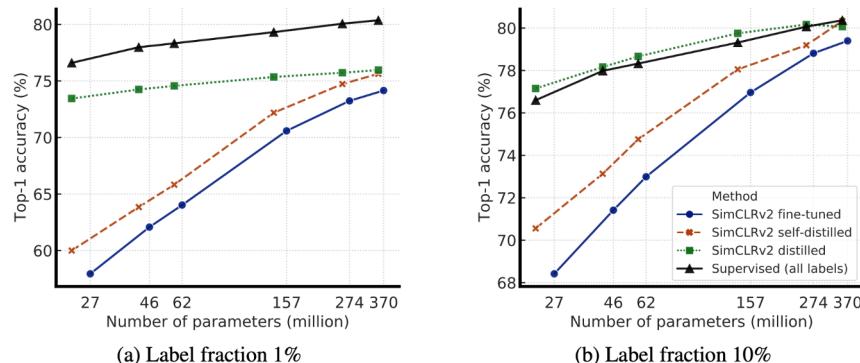
(a) Effect of projection head's depth when fine-tuning from optimal middle layer.



(b) Effect of fine-tuning from middle of a 3-layer projection head (0 is SimCLR).

補充：SimCLR v2一些實驗結果

模型蒸餾 (distilled)：用訓練完成的超大模型 (teacher model)，蒸餾出一個較小的子模型 (student model)，較小的子模型要盡可能保留重要的特徵抽取能力。下圖實驗結果顯示，蒸餾機制在子模型偏小 (圖中左邊區域) 的時候效果很好，隨著模型越大 (圖中右邊區域) 與單純fine-tune的差異逐漸減少。



黑色：100%監督式學習
綠色：大的teacher model->小的student model
橘色：自蒸餾 (student model大小不變)
藍色：單純的fine-tune

Figure 6: Top-1 accuracy of distilled SimCLRV2 models compared to the fine-tuned models as well as supervised learning with all labels. The self-distilled student has the same ResNet as the teacher (without MLP projection head). The distilled student is trained using the self-distilled ResNet-152 ($2\times+SK$) model, which is the largest model included in this figure.

補充：SimCLR v2一些實驗結果

整體來說，SimCLR v2在semi-supervised learning有很突出的表現，不僅比原本的SimCLR還要好，也比很多其他的演算法還好。

Table 3: ImageNet accuracy of models trained under semi-supervised settings. For our methods, we report results with distillation after fine-tuning. For our smaller models, we use self-distilled ResNet-152 ($3 \times +SK$) as the teacher.

Method	Architecture	Top-1		Top-5	
		Label fraction 1%	10%	Label fraction 1%	10%
Supervised baseline [30]	ResNet-50	25.4	56.4	48.4	80.4
<i>Methods using unlabeled data in a task-specific way:</i>					
Pseudo-label [11, 30]	ResNet-50	-	-	51.6	82.4
VAT+Entropy Min. [37, 38, 30]	ResNet-50	-	-	47.0	83.4
Mean teacher [39]	ResNeXt-152	-	-	-	90.9
UDA (w. RandAug) [14]	ResNet-50	-	68.8	-	88.5
FixMatch (w. RandAug) [15]	ResNet-50	-	71.5	-	89.1
S4L (Rot+VAT+Entropy Min.) [30]	ResNet-50 (4 \times)	-	73.2	-	91.2
MPL (w. RandAug) [2]	ResNet-50	-	73.8	-	-
CowMix [40]	ResNet-152	-	73.9	-	91.2
<i>Methods using unlabeled data in a task-agnostic way:</i>					
InstDisc [17]	ResNet-50	-	-	39.2	77.4
BigBiGAN [41]	RevNet-50 (4 \times)	-	-	55.2	78.8
PIRL [42]	ResNet-50	-	-	57.2	83.8
CPC v2 [19]	ResNet-161(*)	52.7	73.1	77.9	91.2
SimCLR [1]	ResNet-50	48.3	65.6	75.5	87.8
SimCLR [1]	ResNet-50 (2 \times)	58.5	71.7	83.0	91.2
SimCLR [1]	ResNet-50 (4 \times)	63.0	74.4	85.8	92.6
BYOL [43] (concurrent work)	ResNet-50	53.2	68.8	78.4	89.0
BYOL [43] (concurrent work)	ResNet-200 (2 \times)	71.2	77.7	89.5	93.7
<i>Methods using unlabeled data in both ways:</i>					
SimCLRV2 distilled (ours)	ResNet-50	73.9	77.5	91.5	93.4
SimCLRV2 distilled (ours)	ResNet-50 (2 \times +SK)	75.9	80.2	93.0	95.0
SimCLRV2 self-distilled (ours)	ResNet-152 (3 \times +SK)	76.6	80.9	93.4	95.5

補充：SimCLR v2結論

SimCLR v2修改SimCLR映射層的設置、使用更大的ResNet模型，並加上模型蒸餾後，在低標註的semi-supervised learning情況下可以大幅進步。

使用更大模型的缺點：訓練時間倍數成長 + 記憶體被模型佔據後，batch size可用會減少。