神經網路裡的黑執事

Y.F. Liou

@DeepLearning101 20200821

All things are start from the "labeling"

Label smoothing

$$p_k = \frac{exp(x^T w_k)}{\sum_{l=1}^{L} exp(x^T w_l))}$$

Predictions as a function of activations in penultimate layer

p_k: Likelihood the model assigns to the *k-th* class

w_k: Weights and biases of the last layer

x: Vector containing the activations of the penultimate layer

$$y_k^{LS} = y_k(1-\alpha) + \frac{\alpha}{K}$$

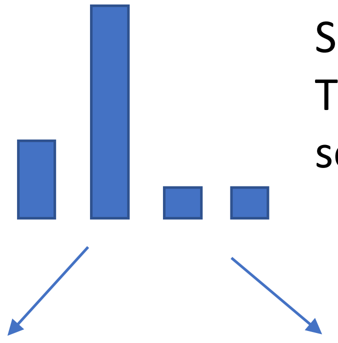
Applying label smoothing to hard targets

0	1	0	0	original

α= .8

.05	.85	.05	.05	smoothing

The purpose of the label smoothing is design to overcome the “overconfidence” which would also cause the overfitting.



Support :
The model give a softmax result.

0	1	0	0

The traditional method only considers the activated class

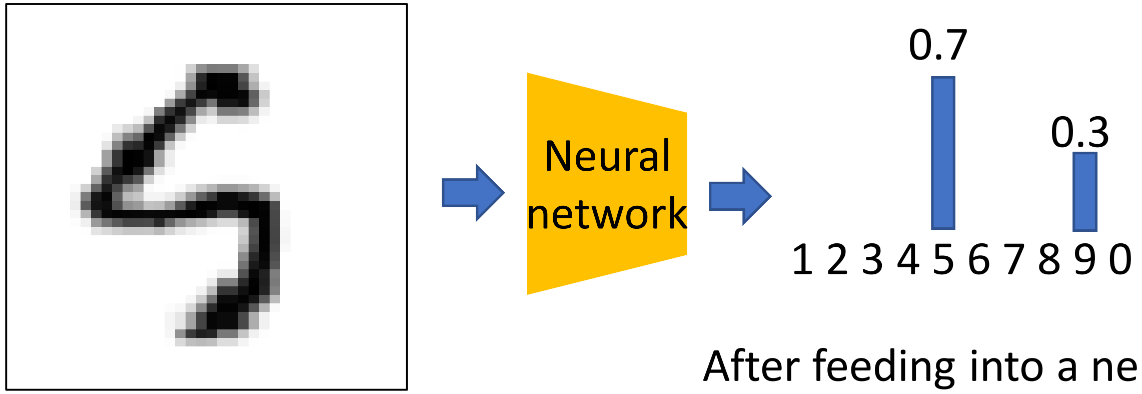
.05	.85	.05	.05

After label smoothing, all the logits are considered

https://medium.com/@nainaakash012/when-does-label-smoothing-help-89654ec75326

Considering the wrong or poor labeling

If there is a number like this



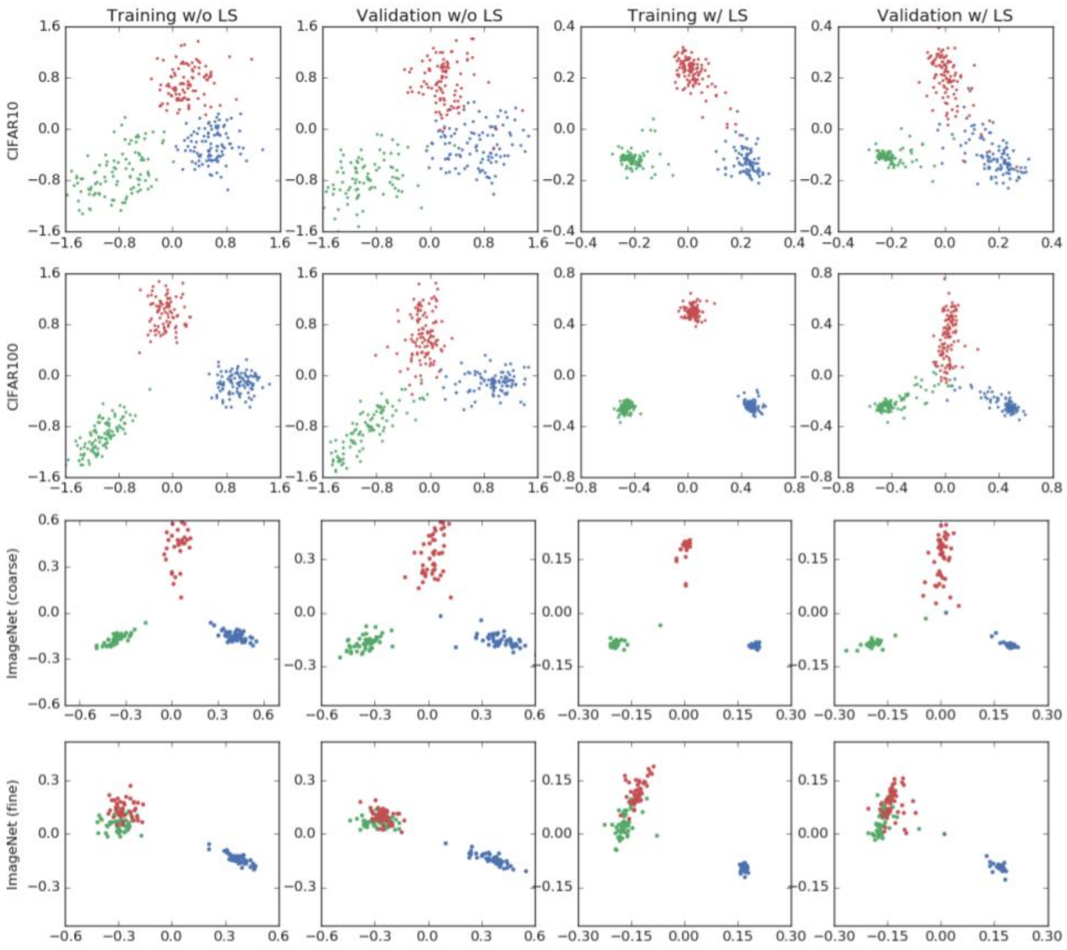
This number is 5, and it also look like 9.

After feeding into a neural network, The answer of 5 and 9 will both give a penalty.
但實際上選定這兩個答案也有道理。

Label smoothing helps to find the dense probability

If we use the label smoothing, the penultimate layer representations give less spread results.

smoothing



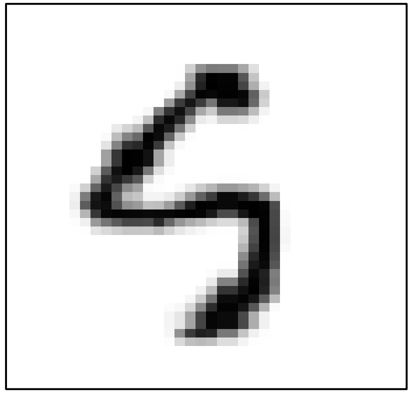
The “spreading” of the teacher is also containing some information. Using label smoothing (which improve the training in many tasks) in distillation work would hurt the training process.

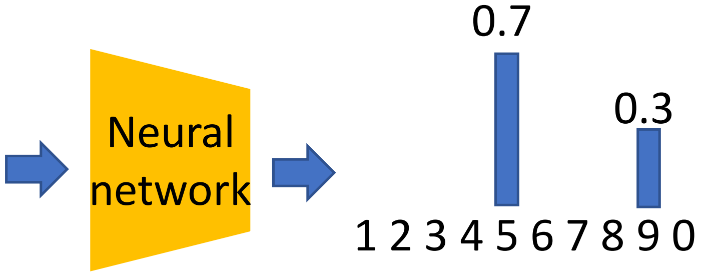
Rafael Müller*, Simon Kornblith, Geoffrey Hinton
Google Brain
Toronto
rafaelmuller@google.com

https://arxiv.org/pdf/1906.02629.pdf

Considering the minor output

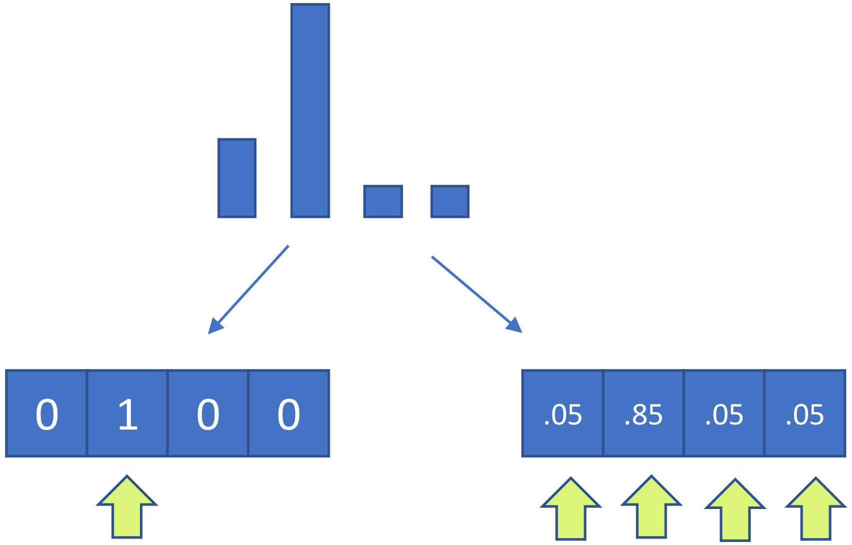
If there is a number like this





This number is 5, and it also look like 9.

After feeding into a neural network, The answer of 5 and 9 will both give a penalty. 但實際上選定這兩個答案也有道理。



Considering the minor outputs of logits is quite similar to consider the "dark knowledge" of model distillation.

從模型預測的其他label輸出似乎也代表特殊意義。

- 是否表示"在這些次要的輸出中，模型告訴我們看到了甚麼"。
- 也許某些資訊也可以透過控制這些次要的輸出，反饋給模型。

有時候黑知識就會影響模型行為

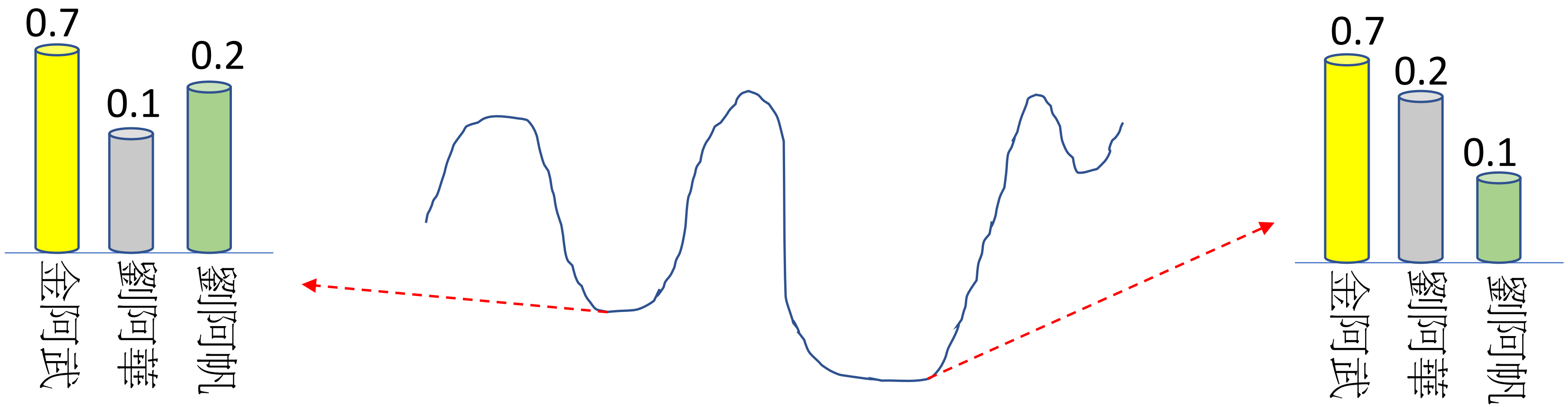
實驗結果由 雪豹科技 豹小秘提供

實驗方式

1. 輸入~~講者照片~~每次分享的摘要與時間(文字檔)
2. 進行一般性的問答。例如某天meetup的講者是誰及內容為何。
3. 確認豹小秘強大的理解力~~與不說謊的特性~~。

The Dark knowledge

- 這些非label的部分，在knowledge distillation的課題中稱呼”Dark knowledge”
- 影響Dark knowledge的表現，有可能也能影響整個模型
  - What if we control the dark knowledge directly?



這兩個結果對於最後使用SGD更新，回傳的loss是相同的。但是在dark knowledge部分表現非常不同。

Enhancing the dark knowledge

Usually,

The main logit will dominate most of output signals. If we want to enhancing the dark knowledge, softening the output would be a way …

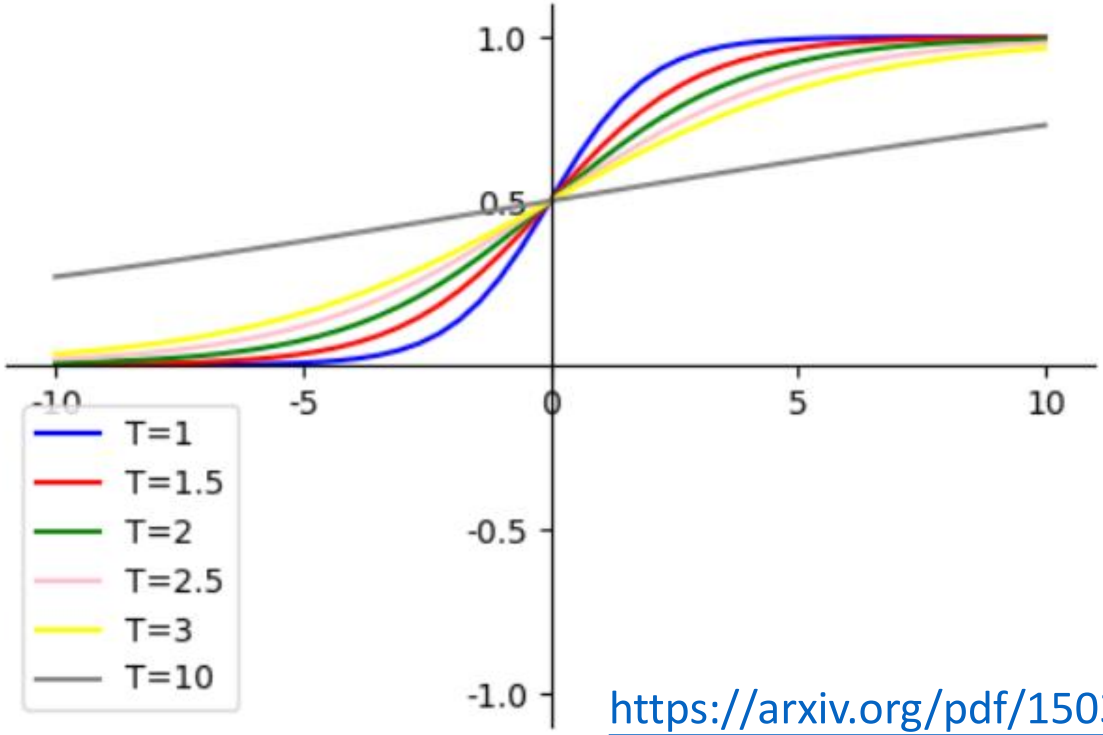
Temperature of activation function

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

z = logits
T = Temperature
j = index of class
q = new logits

目標是把teacher 的label做soft。
可能的原因是teacher所做出來的答案不一定正確，而且內部有許多Dark knowledge。因此做soft後，希望student可以學到teacher所給予的所有資訊。

Here, “Temperature” enlarge the signal of dark knowledge.



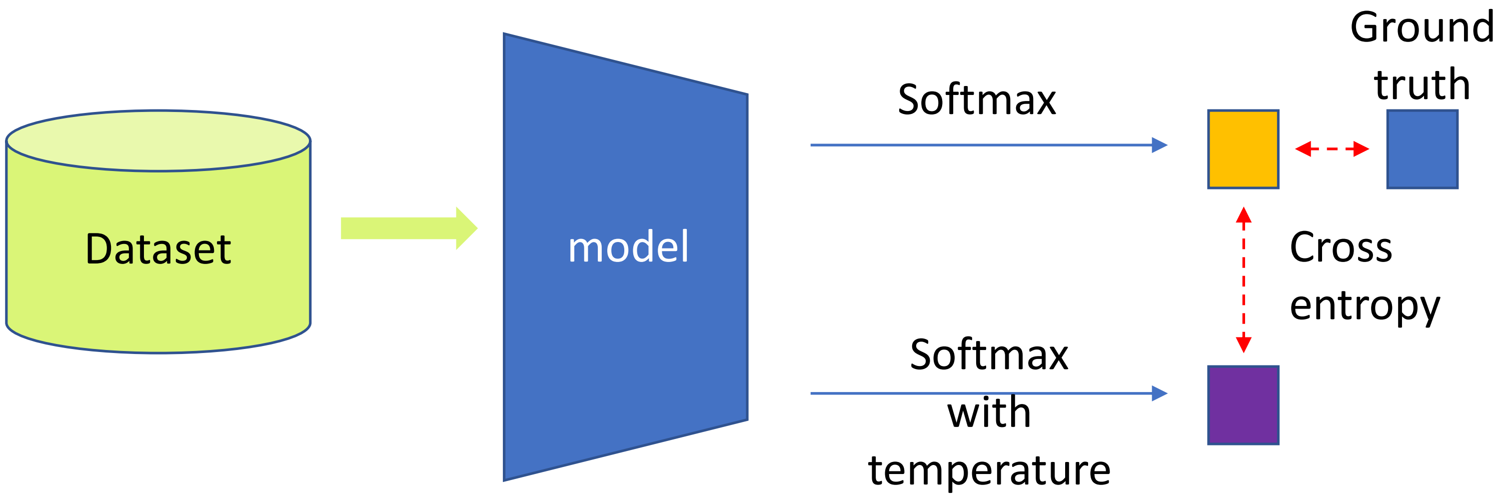

```
>>> np.array([1.4, 2.8,.9])
array([1.4, 2.8, 0.9])
```

```
>>> sess.run(tf.math.softmax(a))
array([0.17662444, 0.71624742, 0.10712814])
>>> sess.run(tf.math.softmax(a * .3))
array([0.2956245 , 0.44992913, 0.25444637])
>>> sess.run(tf.math.softmax(a * .5))
array([0.26367459, 0.53097543, 0.20534998])
```

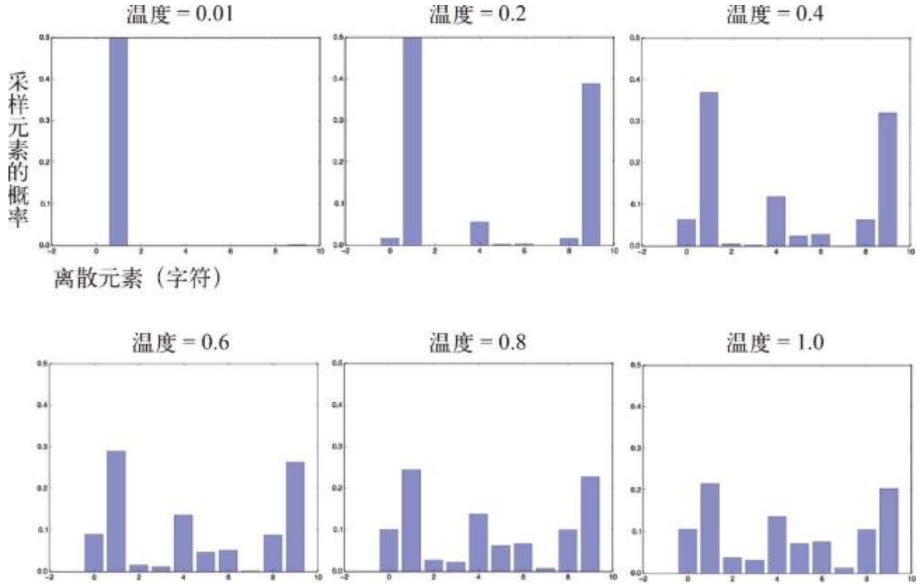
https://arxiv.org/pdf/1503.02531.pdf

Teacher free knowledge distillation

- Label smoothing 跟knowledge distillation 都專注在minor logits
  - Label smoothing – 不依賴任何prior，直接設定超參數來看結果
  - Knowledge distillation – 依賴teacher 給予比重
  - 兩種想法結合以後，也許可以達到self-regularize。因為如果模型將錯誤的minor logits提高，也許就能更進一步讓模型來“理解”問題。



使用不同溫度，可以凸顯出來的minor logits強度也不同。就看希望模型可以注意到多小的程度。



http://static.kancloud.cn/mikl_maple/python/1726331

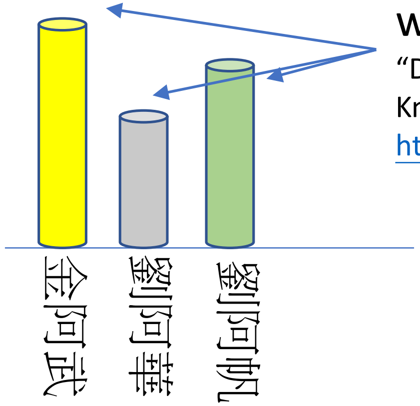
https://arxiv.org/pdf/1909.11723.pdf [CVPR2020]

What if controlling the dark knowledge directly

Noisy labeling

- 從Softmax with temperature來看，在整個label上是加上一些noise。因此就直接加上noise就好了。
- 介紹2種方法

Method 1.



Perturbation the logits would be better.

"Deep Model Compression: Distilling Knowledge from Noisy Teachers"

https://arxiv.org/pdf/1610.09650.pdf

$$z'^{(i)} = (\mathbf{1} + \xi).z^{(i)}$$

z = original logits

z' = noisy logits

ε = random from Gussian

$$L(x, z', \theta) = \frac{1}{2T}\sum_i \|g(x^{(i)};\theta) - z'^{(i)}\|_2^2$$

Method 2.

Output with Temperature

$$L_{\mathcal{D}}(y_i, f(x_i)) = \lambda l(y_i, f(x_i)) + (1-\lambda) l(s_i, f(x_i)), \quad (4)$$

一部分參考真實資料，一部分參考使用noise處理過的label。其比重為λ。

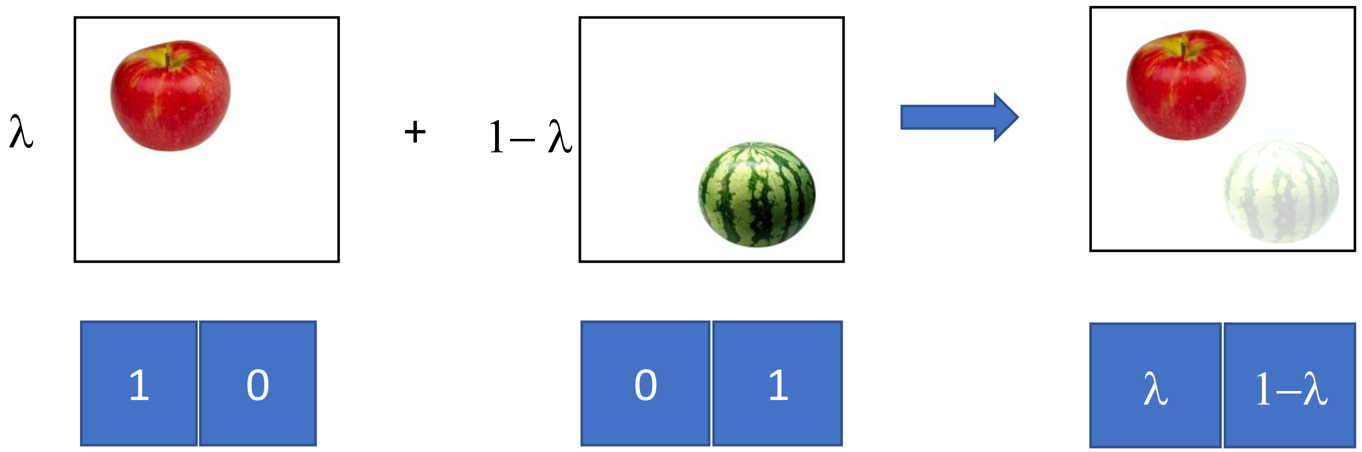
https://arxiv.org/pdf/1610.09650.pdf

證明noisy label有用:

https://papers.nips.cc/paper/5073-learning-with-noisy-labels.pdf

mixup

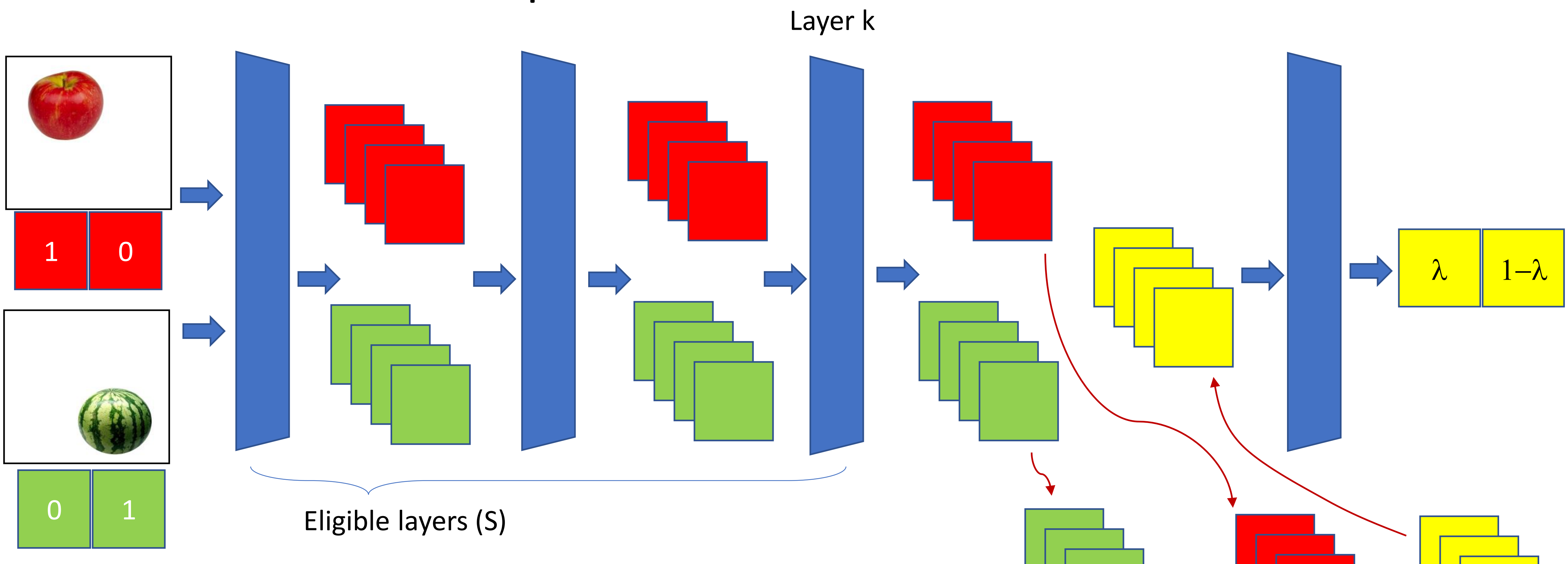
$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \qquad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \qquad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

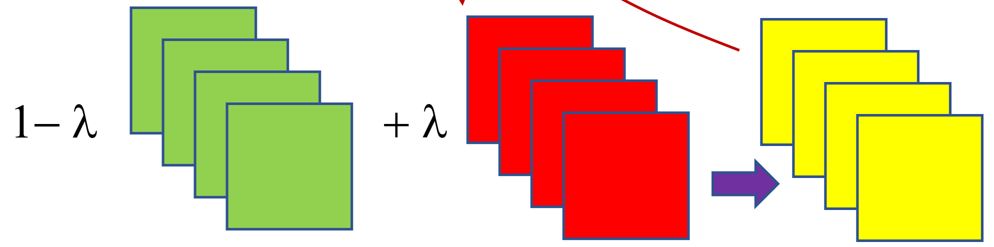


Main purpose:
1. Data augmentation
2. Maybe … Dark knowledge

ICLR2018
https://arxiv.org/pdf/1710.09412.pdf

Manifold mixup

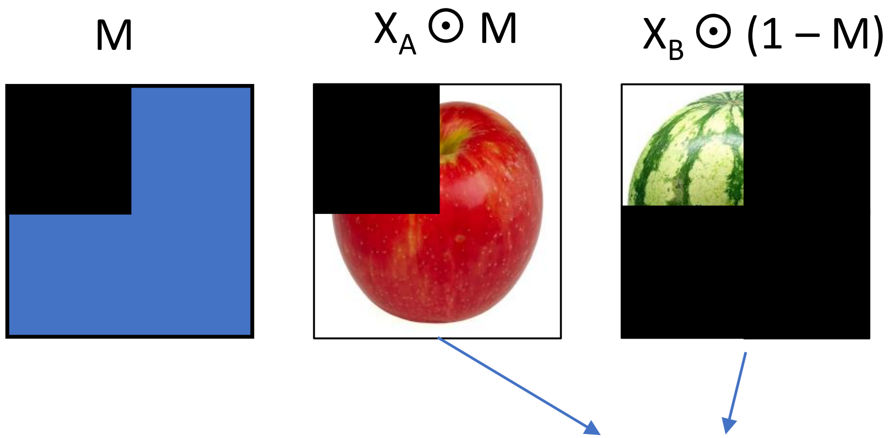


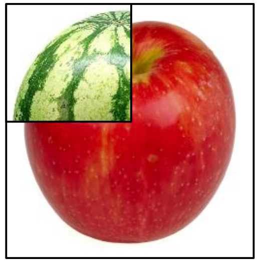


Manifold Mixup passes this sanity check (consult Appendix D for further details). While we found that using *Manifold Mixup* improves the robustness to single-step FGSM attack (especially over Input Mixup), we found that *Manifold Mixup* did not significantly improve robustness against stronger, multi-step attacks such as PGD (Madry et al., 2018).

https://arxiv.org/pdf/1806.05236.pdf

CutMix

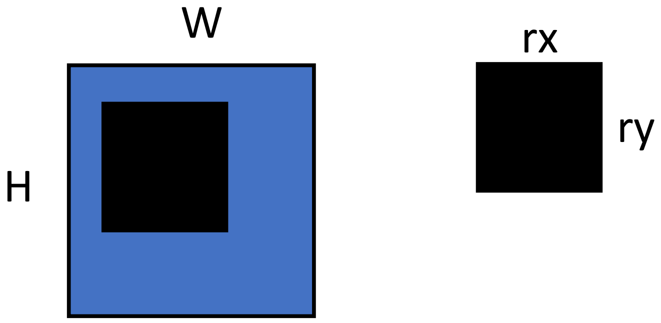




$$\tilde{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B$$

使用Mask遮蔽後，把兩張圖片再合成一張

label部分直接使用圖像所佔的比例，
舉例:



$HW - r_x r_y$ $\quad$ $r_x r_y$

$= 1 - (r_x r_y / HW)$ $\quad$ $= r_x r_y / HW$

如果Mask掉大小為$r_x r_y$，則圖A像原始比例剩下$HW - r_x r_y$，另外一張就是$r_x r_y$ 。
Label比例也調整為$(HW - r_x r_y)$跟$(r_x r_y)$

	Mixup	Cutout	CutMix
Usage of full image region	✔	✘	✔
Regional dropout	✘	✔	✔
Mixed image & label	✔	✘	✔

Table 2: Comparison among Mixup, Cutout, and CutMix.

內涵:
Dark knowledge部分，如果一開始無法從pretrain model中取得做蒸餾，也許我們可以自己製作這些dark knowledge 。

https://arxiv.org/pdf/1905.04899.pdf

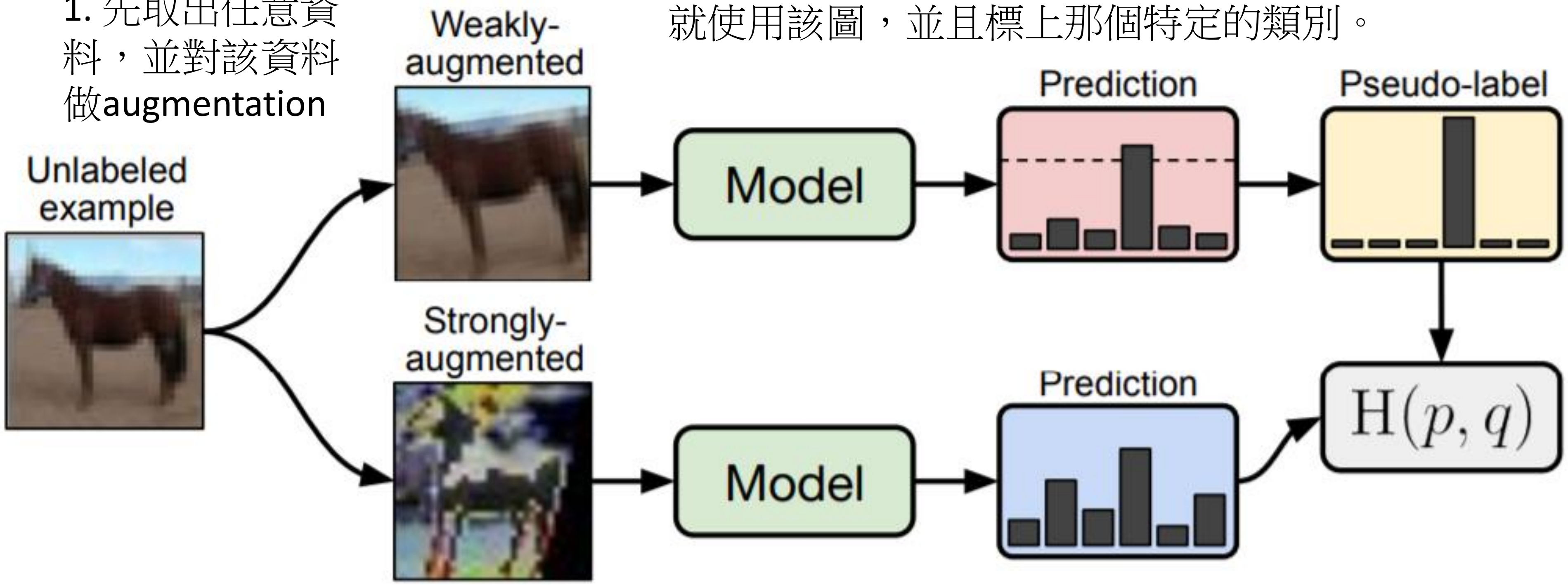
Extract the knowledge of unlabeled data

If you have lots of unlabeled data and you want to leverage the information inside them

FixMatch

1. 先取出任意資料，並對該資料做augmentation

2. 使用模型把augmented的資料放入模型中，如果模型有算出某個類別有高於特定的threshold(prediction的虛線)，就使用該圖，並且標上那個特定的類別。



依據接下來的任務來指定pseudo-label

3. 同一個圖加上更多的noise做augmenting，並放到同一個模型當中，最後把算出來的結果進行cross entropy的運算

https://arxiv.org/pdf/2001.07685.pdf

MixMatch

After various objections, the most important result will be get after average.

Consistency regularization

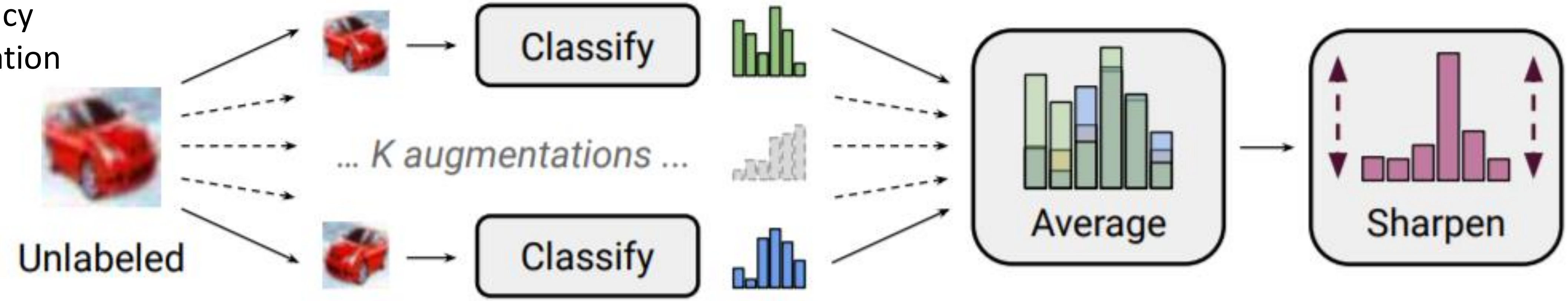
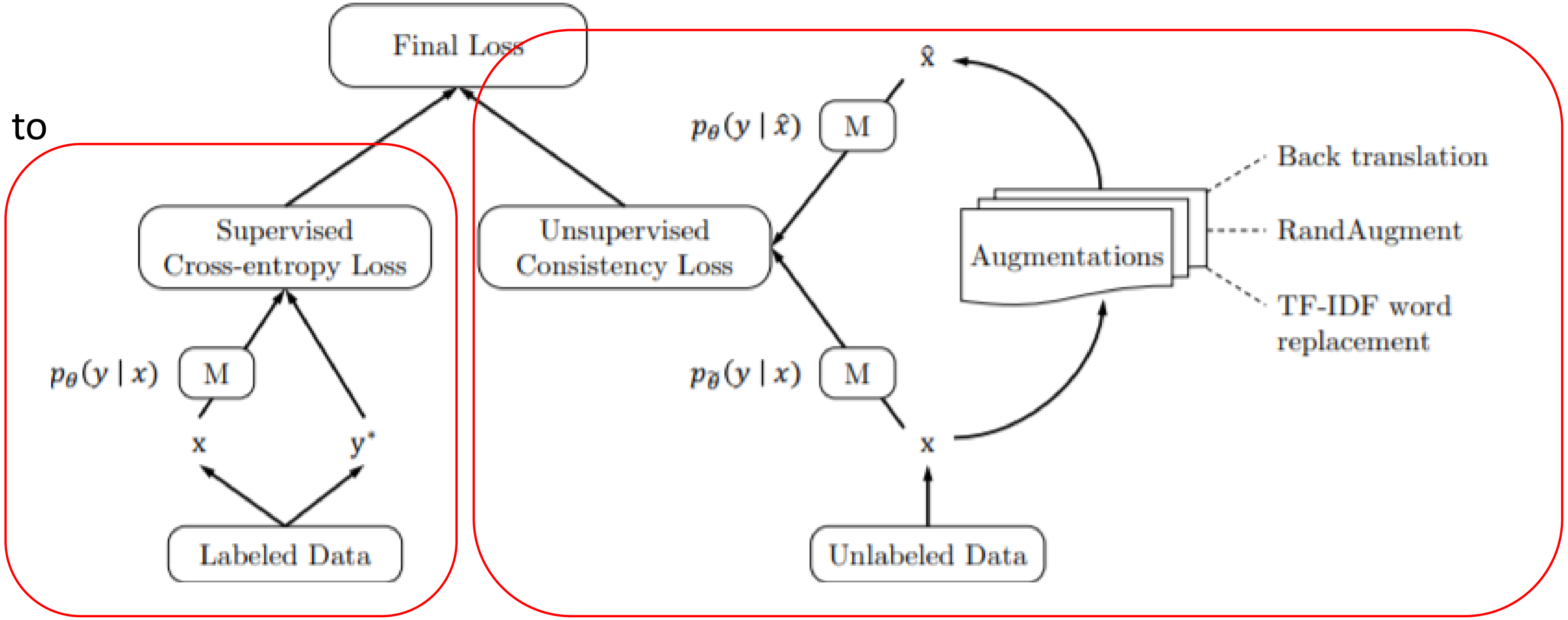


Figure 1: Diagram of the label guessing process used in MixMatch. Stochastic data augmentation is applied to an unlabeled image K times, and each augmented image is fed through the classifier. Then, the average of these K predictions is "sharpened" by adjusting the distribution's temperature. See algorithm 1 for a full description.

https://arxiv.org/pdf/1905.02249.pdf

Unsupervised data augmentation (UDA)

Normal process to train the neural network

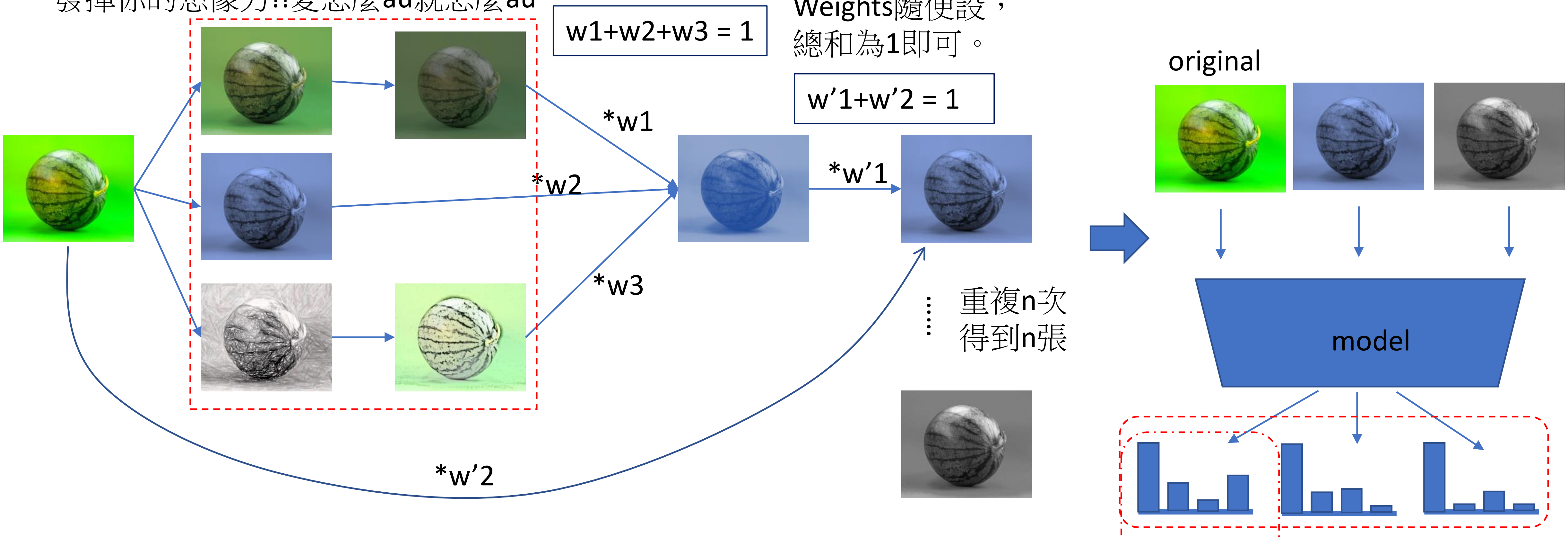


1. 把原始資料進行augmentation(加噪音等等的處理)
2. 以原始圖透過模型預測出來的數值作為答案，augmentation的圖最後答案應該要跟原始圖的答案一模一樣。

https://arxiv.org/pdf/1904.12848.pdf

AugMix

發揮你的想像力!!愛怎麼au就怎麼au



Loss 2 (*λ)

計算三者的JS divergence

Ground truth

$$M = (p_{\text{orig}} + p_{\text{augmix1}} + p_{\text{augmix2}})/3$$

$$\text{JS}(p_{\text{orig}}; p_{\text{augmix1}}; p_{\text{augmix2}}) = \frac{1}{3}\Big(\text{KL}[p_{\text{orig}} \| M] + \text{KL}[p_{\text{augmix1}} \| M] + \text{KL}[p_{\text{augmix2}} \| M]\Big).$$

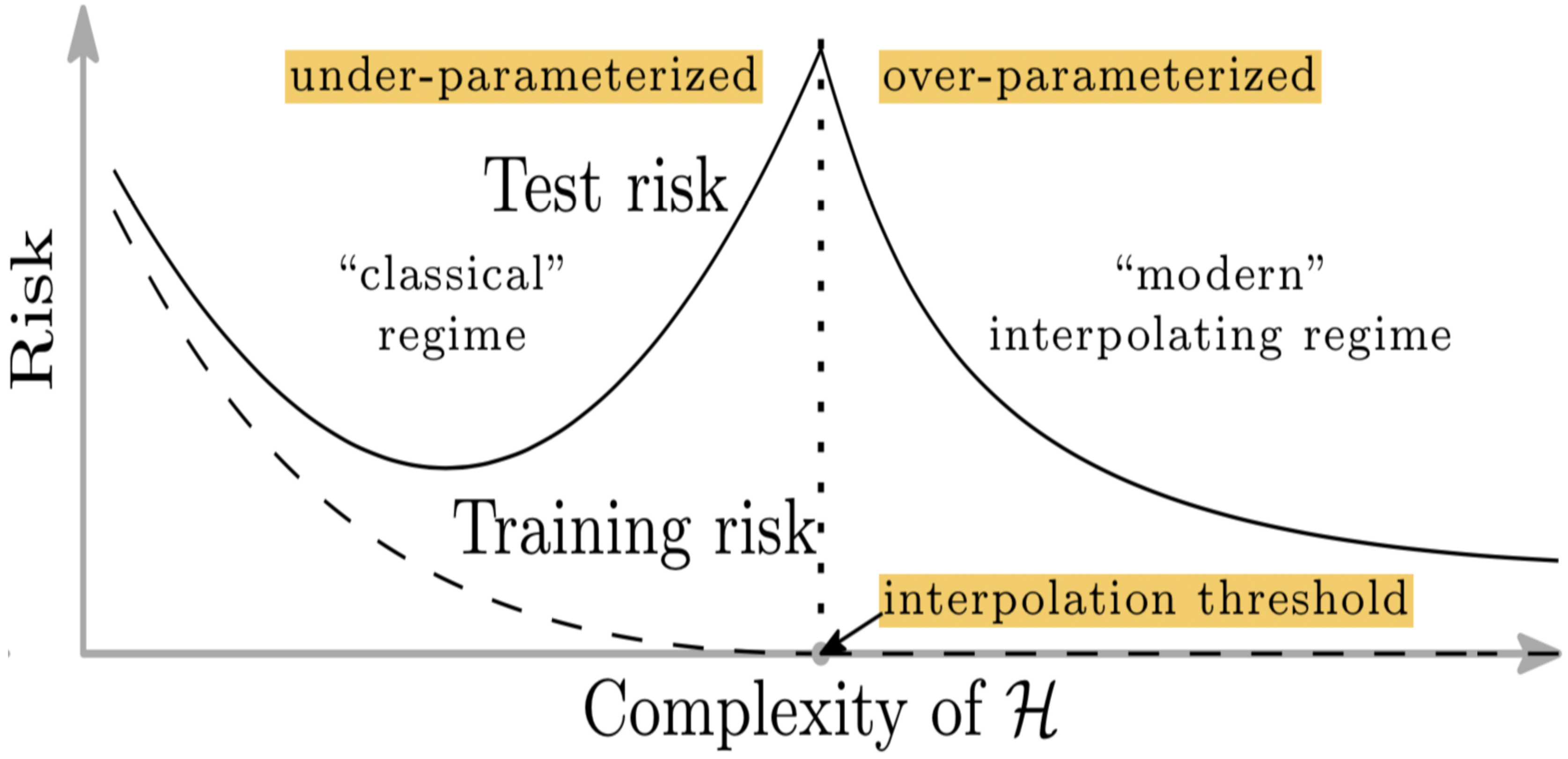
Loss 1 $\mathcal{L}(p_{\text{orig}}, y) + \lambda\, \text{JS}(p_{\text{orig}}; p_{\text{augmix1}}; p_{\text{augmix2}}).$

https://arxiv.org/pdf/1912.02781.pdf

Since the unlabeled data could give "information"

Using "data" for regularization would be a way to control the overparameterization models …

What is happening to overparameterization?

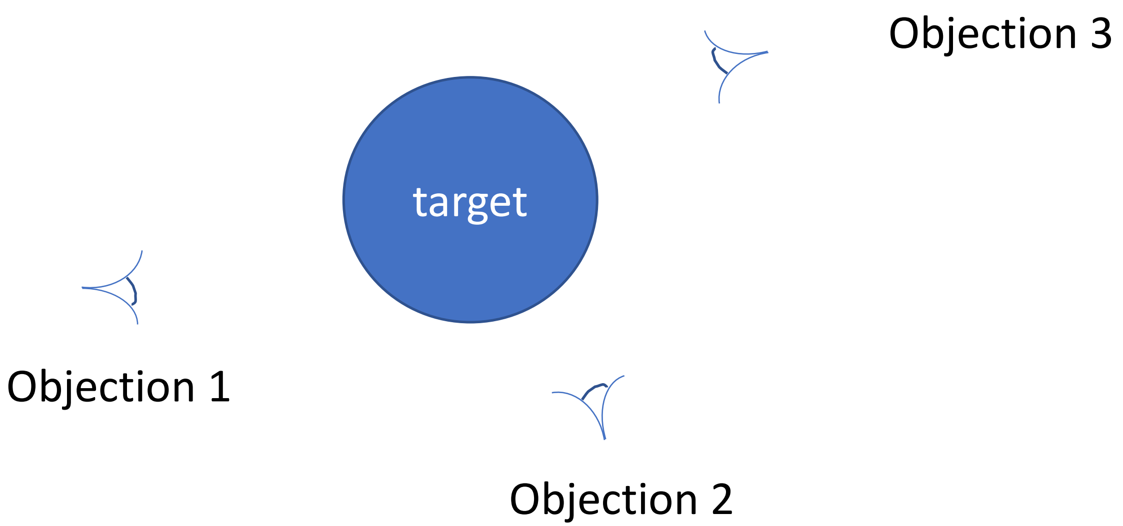


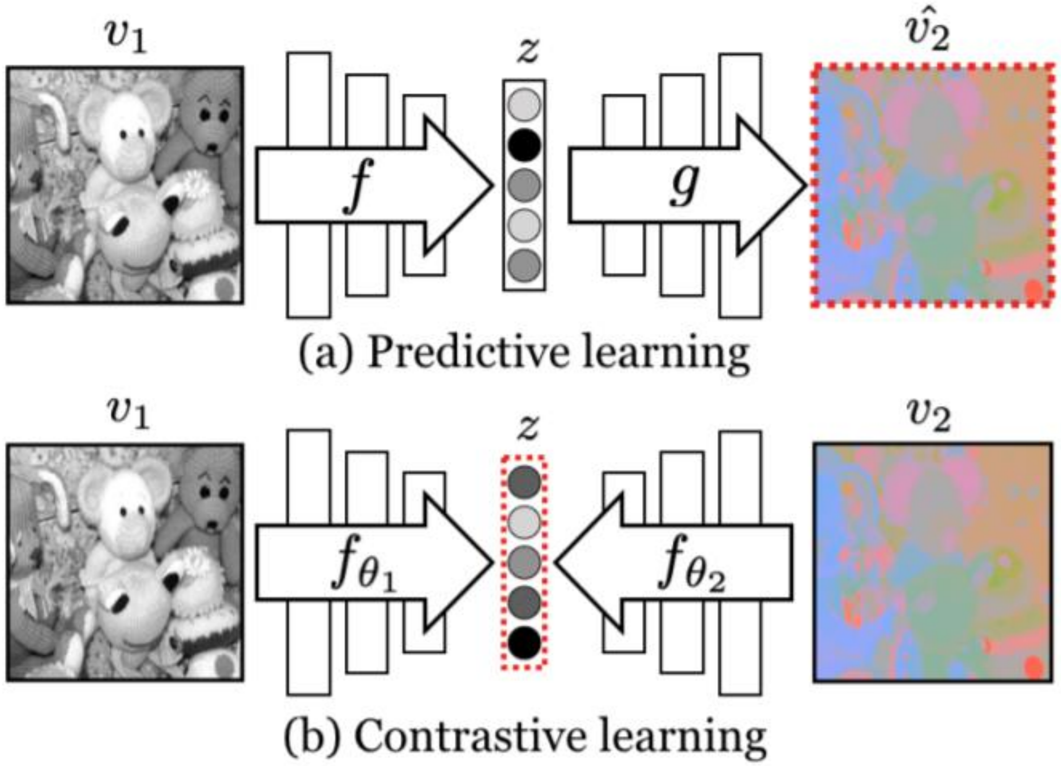
使用over-parameterization的model，基礎的machine learning concept多會不管用。
For example:
L1, L2 regularizations。在基礎的機器學習中是透過降低參數量達到generalized purpose。但是在over-parameter mode上，降低很多參數仍然會讓model處於over-parameterized的狀態。因此可能不會太管用。

解決方式:
目前最好的regularization method仍然是使用training data來達到目的。

https://lilianweng.github.io/lil-log/2019/03/14/are-deep-neural-networks-dramatically-overfitted.html

Contrastive learning





If we use different objection for the same target, we would get different presenting which are belong to the same objects

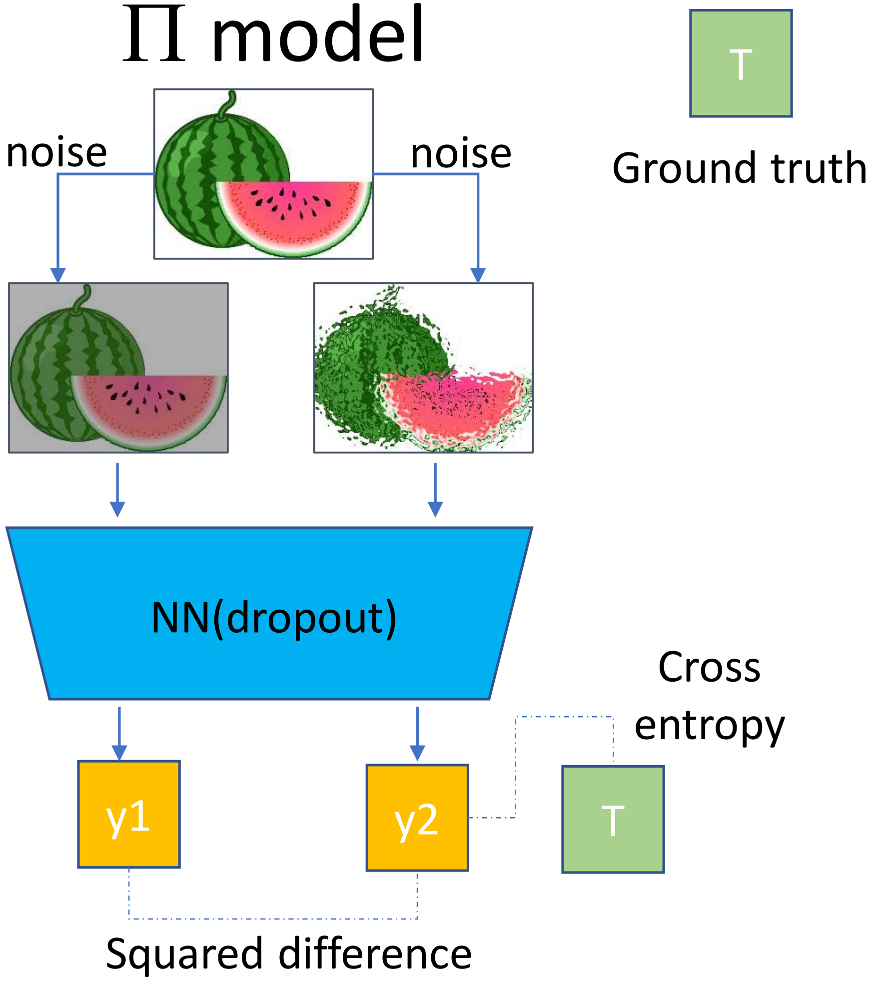
We need to tell the neural network that “these are the same thing”.

https://arxiv.org/pdf/1906.05849.pdf

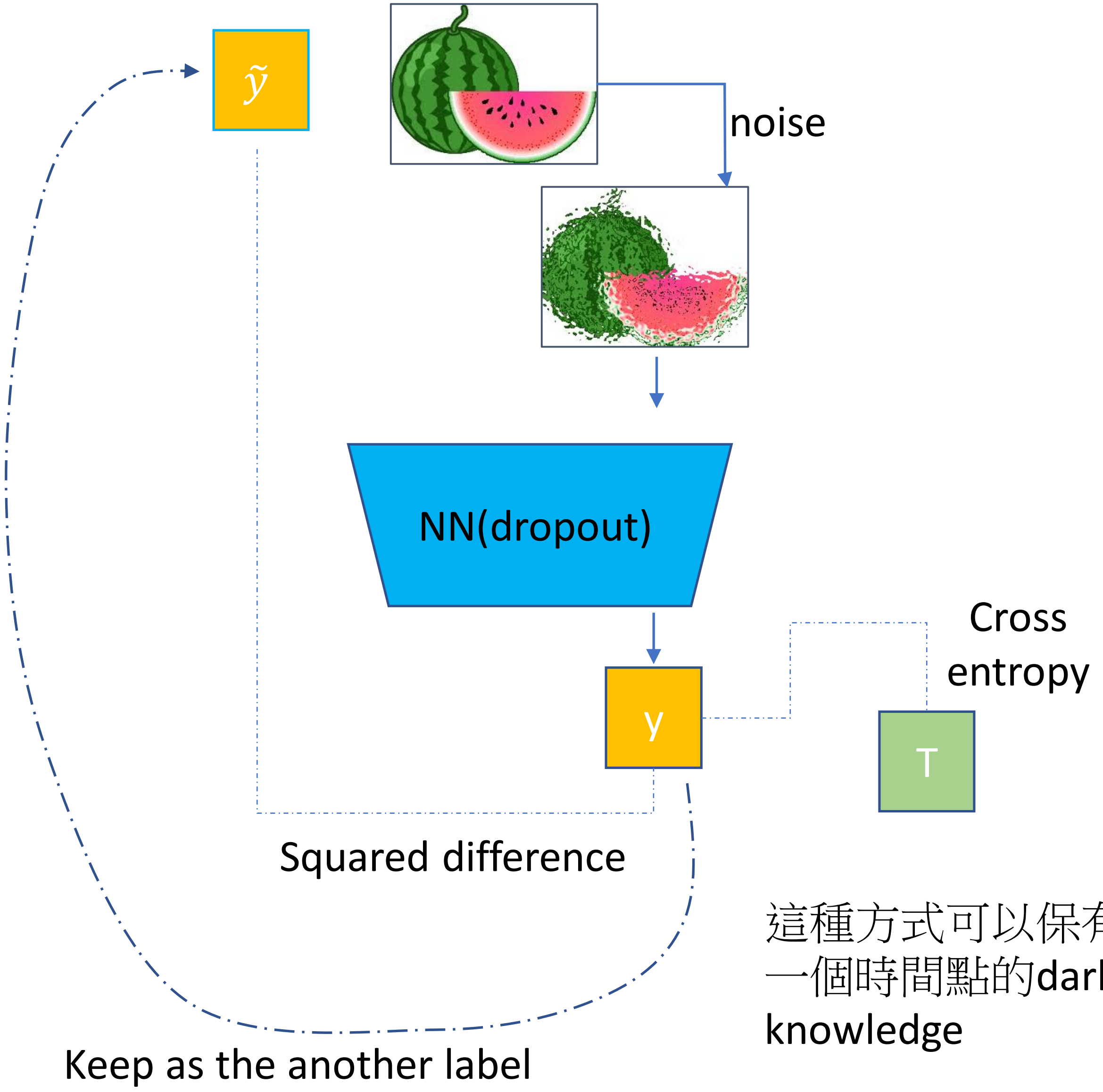
Self-supervised concept

一個標記不夠，可以創兩個

Π model



Temporal ensemble



這種方式可以保有上一個時間點的dark knowledge

https://arxiv.org/pdf/1610.02242.pdf

Mean teacher

Already proposing the concepts of self-supervise (consistency regularization)

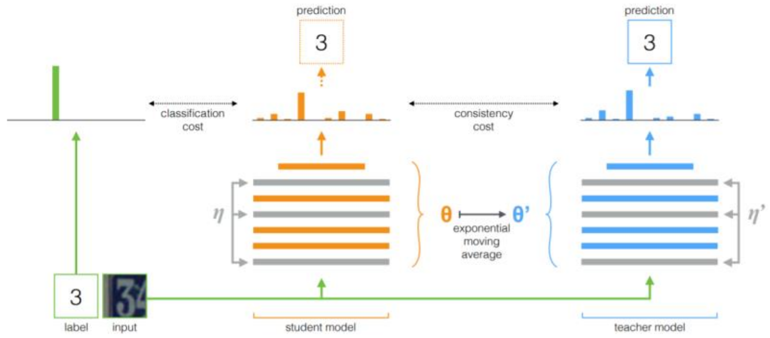
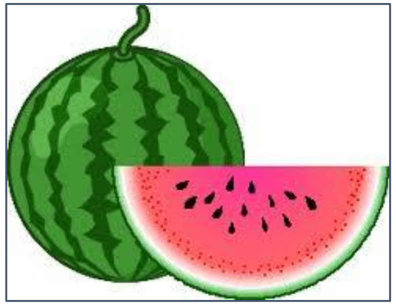
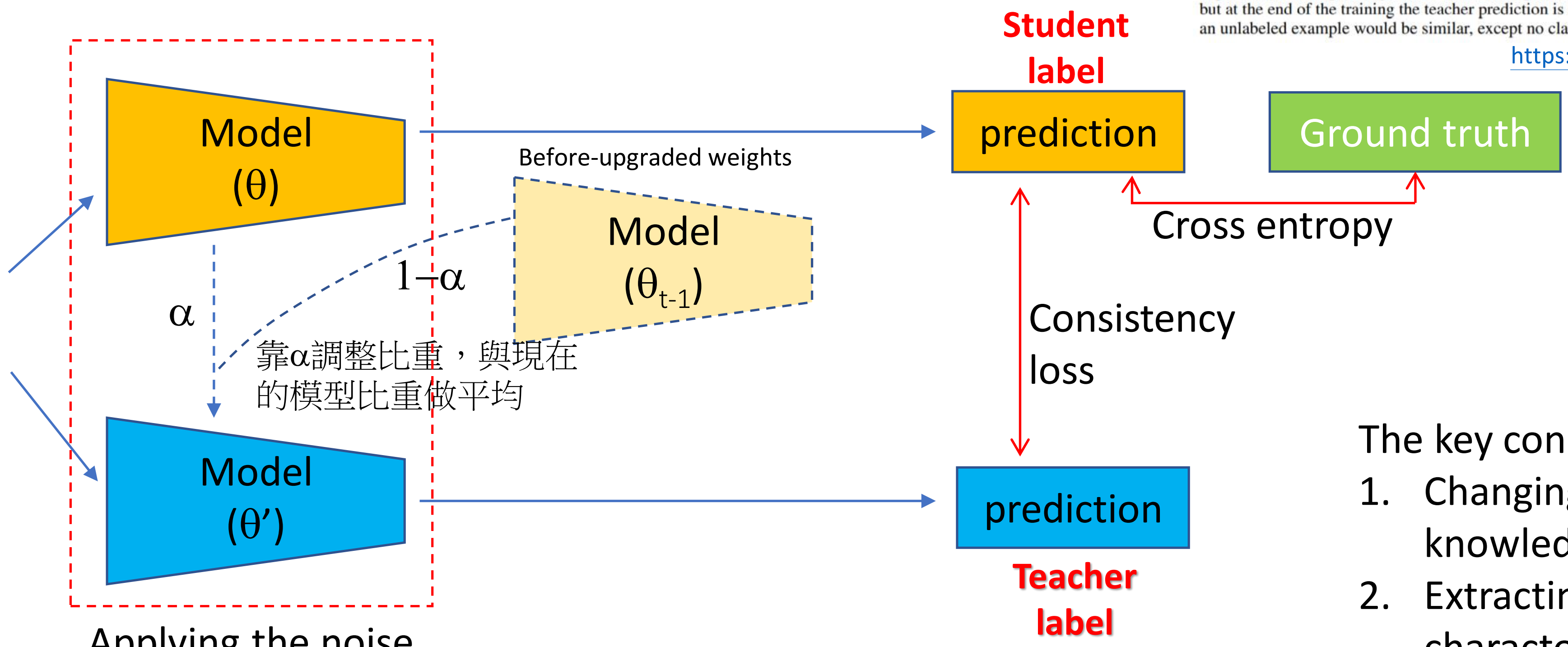


Figure 2: The Mean Teacher method. The figure depicts a training batch with a single labeled example. Both the student and the teacher model evaluate the input applying noise (η, η') within their computation. The softmax output of the student model is compared with the one-hot label using classification cost and with the teacher output using consistency cost. After the weights of the student model have been updated with gradient descent, the teacher model weights are updated as an exponential moving average of the student weights. Both model outputs can be used for prediction, but at the end of the training the teacher prediction is more likely to be correct. A training step with an unlabeled example would be similar, except no classification cost would be applied.

https://arxiv.org/pdf/1703.01780.pdf





Applying the noise to the weights (η and η')
(Dropout would be a good idea)

α would not to be large during initially training stage.

The key concept:

1. Changing the "dark knowledge"
2. Extracting the ensemble charactoristics.

自己跟自己比還不夠

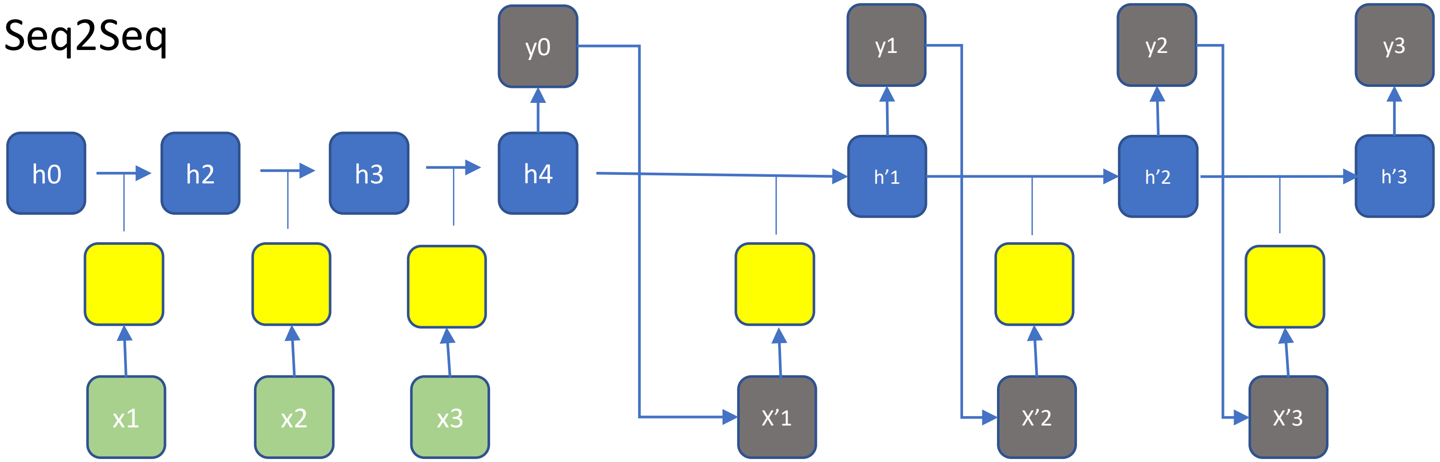
透過跟別人比較，更了解自己的定位

Metric learning (meta-learning)

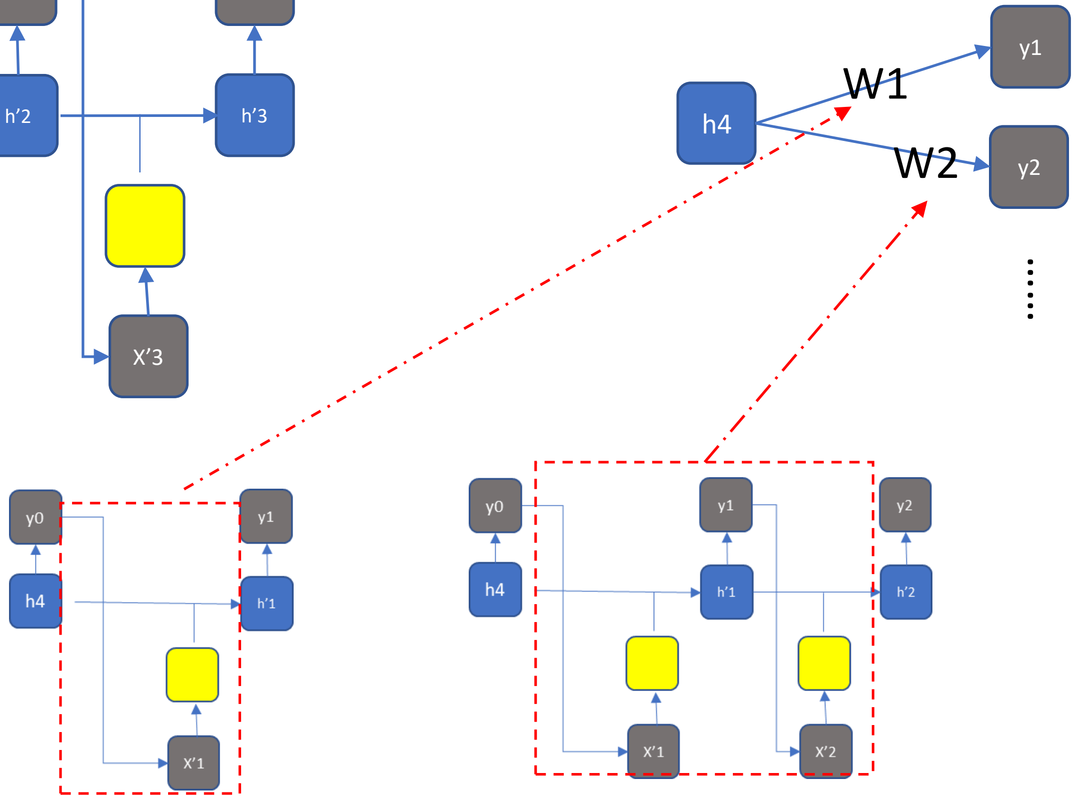
Rethinking of RNN

問題:
是否能學到一堆矩陣就當成不同位置的開關?

Seq2Seq



1. h'系列全部都是由h4所得到，換句話說，h4早就隱含了h'系列的所有資訊
2. h'系列都是h4透過某些方式"打開"開關。例如h'1接收了x'1；h'2等於接收了x'1及x'2，以此類推。因此可以把這些"接收訊息"(例如x'1, x'1+x'2, ...)當成"矩陣"，而這矩陣專門用來打開h4對於特定位置的開關。當h4接受到這些特定開關以後，就能把特定數值輸出即可。



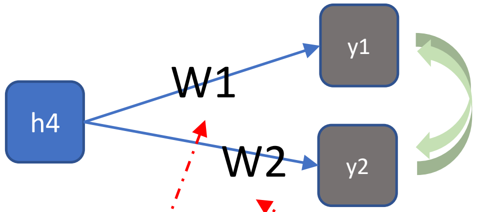
Y1根本是自己產生，只是透過x'1打開自己

Y2是透過x'1和x'2的訊息累加後做為開關。但x'1和x'2也是由h4自己產生

Contrastive prediction code (CPC)

What is the “Contrastive”?
如果有“同一個”物件，我們用“不同角度”來看它，就能找出不同處。

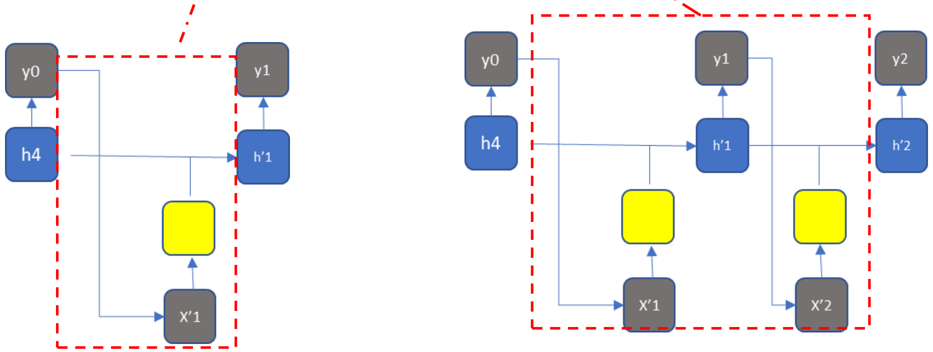
目前常用的方法就是加上noise或甚至直接乘以矩陣。



同一個h4，用不同的W處理，同等于用不同角度在看h4。

因此把h4投射到不同位置，等於在計算mutual information。

假設從RNN部分取得的資訊可以拆解(參閱上一頁的投影片)



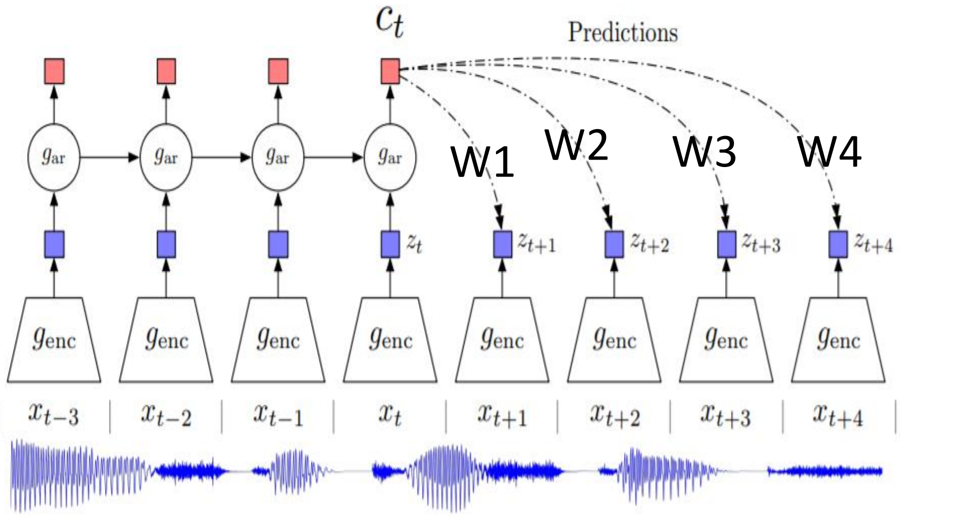
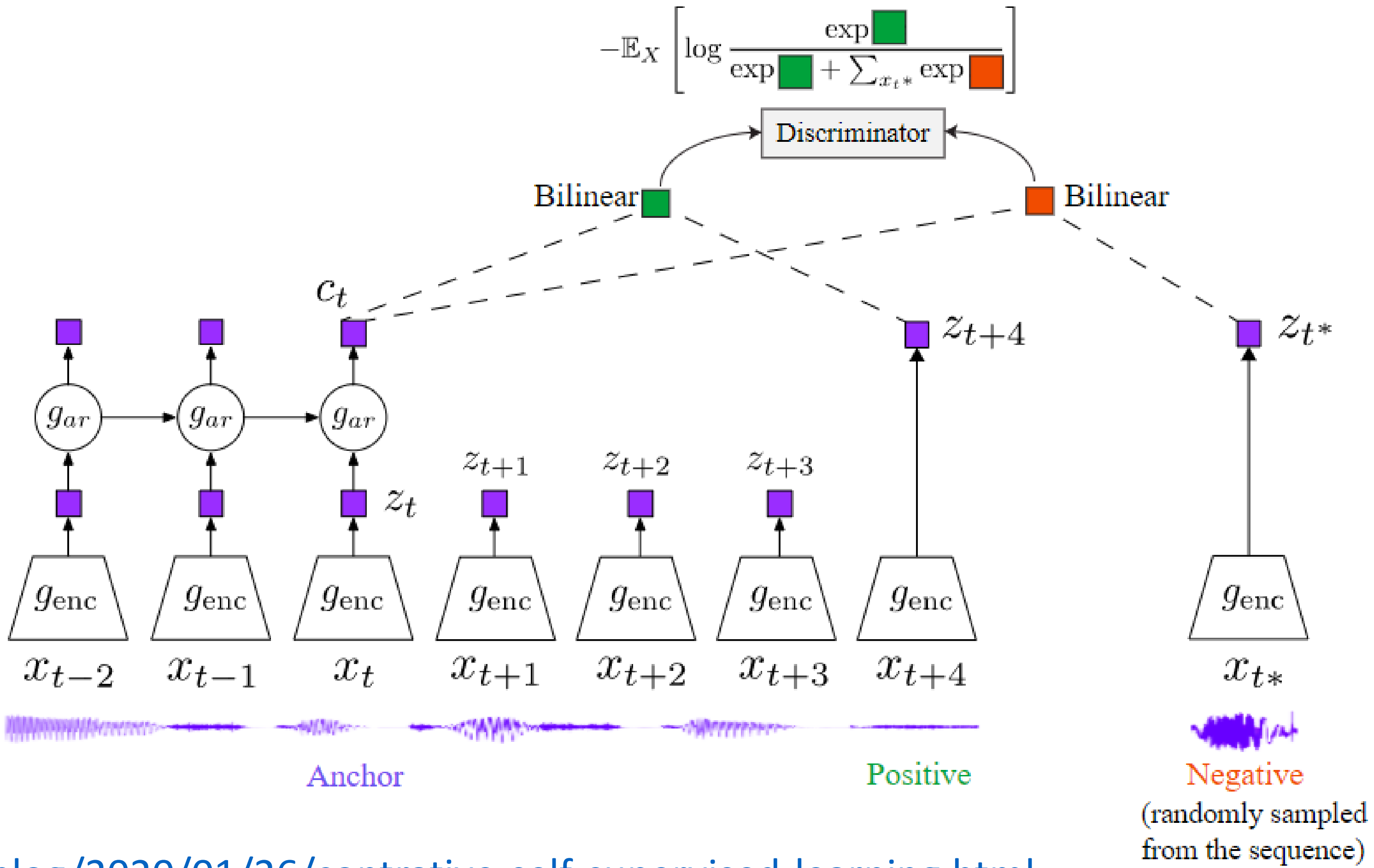


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

https://arxiv.org/pdf/1807.03748.pdf

CPC還是需要靠triplet loss

儘管理論上使用mutual information應該足夠把目標練起來。但是多加上Triplet loss可以讓最後網路更加穩定。



https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html

Triplet loss specification

If you want to comparing "the difference between object", just give them a ruler.

$$\mathcal{L}(\{x, x^+, \{x_i\}_{i=1}^{N-1}\}; f) = \log\left(1 + \sum_{i=1}^{N-1} \exp(f^\top f_i - f^\top f^+)\right)$$

This can be thought as triplet loss if we have only 3 samples :

$$\mathcal{L}_{(2+1)\text{-tuplet}}(\{x, x^+, x_i\}; f) = \log\left(1 + \exp(f^\top f_i - f^\top f^+)\right);$$
$$\mathcal{L}_{\text{triplet}}(\{x, x^+, x_i\}; f) = \max\left(0, f^\top f_i - f^\top f^+\right).$$

This can also be thought as "softmax"

$$\log\left(1 + \sum_{i=1}^{L-1} \exp(f^\top f_i - f^\top f^+)\right) = -\log \frac{\exp(f^\top f^+)}{\exp(f^\top f^+) + \sum_{i=1}^{L-1} \exp(f^\top f_i)}$$

Tuplet loss - https://papers.nips.cc/paper/6200-improved-deep-metric-learning-with-multi-class-n-pair-loss-objective.pdf

Negative sample

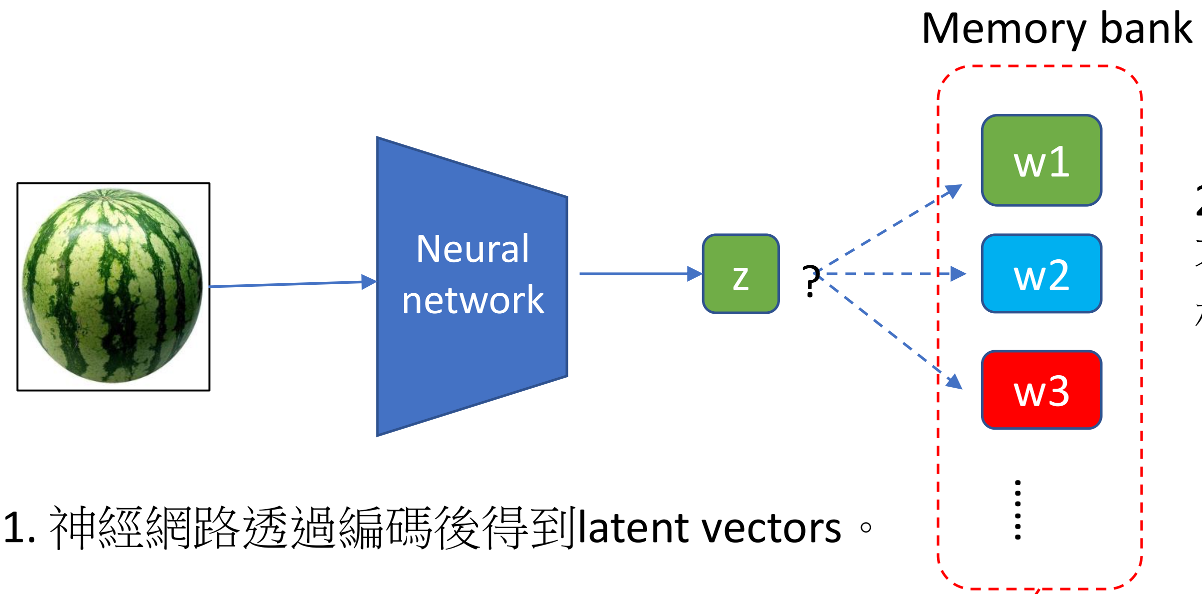
Z is a representation
$\hat{Z}$ is another view of Z

$$\mathcal{L}_{\text{CPC}} = -\sum_{i,j,k} \log p(\boldsymbol{z}_{i+k,j} | \hat{\boldsymbol{z}}_{i+k,j}, \{\boldsymbol{z}_l\})$$

$$= -\sum_{i,j,k} \log \frac{\exp(\hat{\boldsymbol{z}}_{i+k,j}^T \boldsymbol{z}_{i+k,j})}{\exp(\hat{\boldsymbol{z}}_{i+k,j}^T \boldsymbol{z}_{i+k,j}) + \sum_l \exp(\hat{\boldsymbol{z}}_{i+k,j}^T \boldsymbol{z}_l)}$$

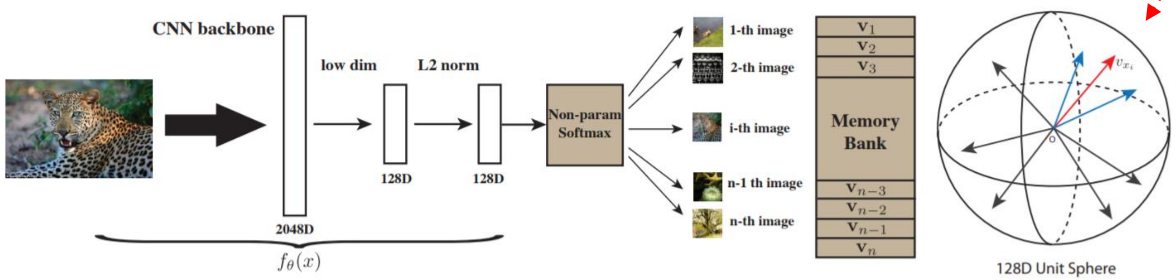
Info-NCE
https://arxiv.org/pdf/1505.00687.pdf
https://arxiv.org/pdf/1905.09272.pdf

Memory bank



2. latent vectors跟一群w相比。這些w分別代表。不同類別的representing。最後取得最大的那個相似度就認定z屬於那個類別。

1. 神經網路透過編碼後得到latent vectors。



memory

temperature

$$P(i|\mathbf{v}) = \frac{\exp\left(\mathbf{v}_i^T \mathbf{v}/\tau\right)}{\sum_{j=1}^{n} \exp\left(\mathbf{v}_j^T \mathbf{v}/\tau\right)}$$

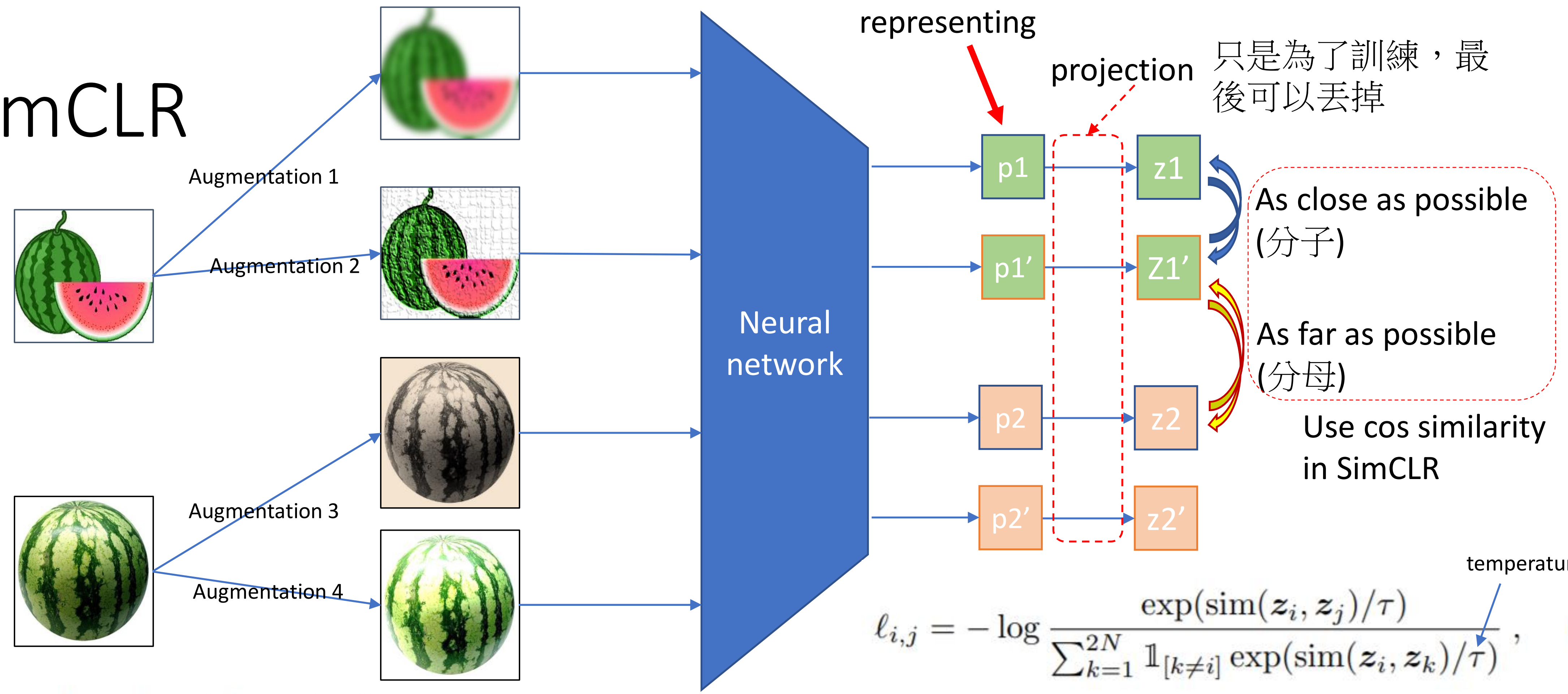
target

Minimize the log-liklihood

$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \log P(i|f_{\boldsymbol{\theta}}(x_i))$$

https://arxiv.org/pdf/1805.01978v1.pdf

SimCLR



$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \quad (1)$$

- Representation learning with contrastive cross entropy loss benefits from normalized embeddings and an appropriately adjusted temperature parameter.
- Contrastive learning benefits from larger batch sizes and longer training compared to its supervised counterpart. Like supervised learning, contrastive learning benefits from deeper and wider networks. https://arxiv.org/pdf/2002.05709.pdf

Tips

1. Because the sample size will be huge, LARS is recommended in such tasks.
2. Using layer normalization or global batch normalization.

More in SimCLR – crop and color distortions get better performance

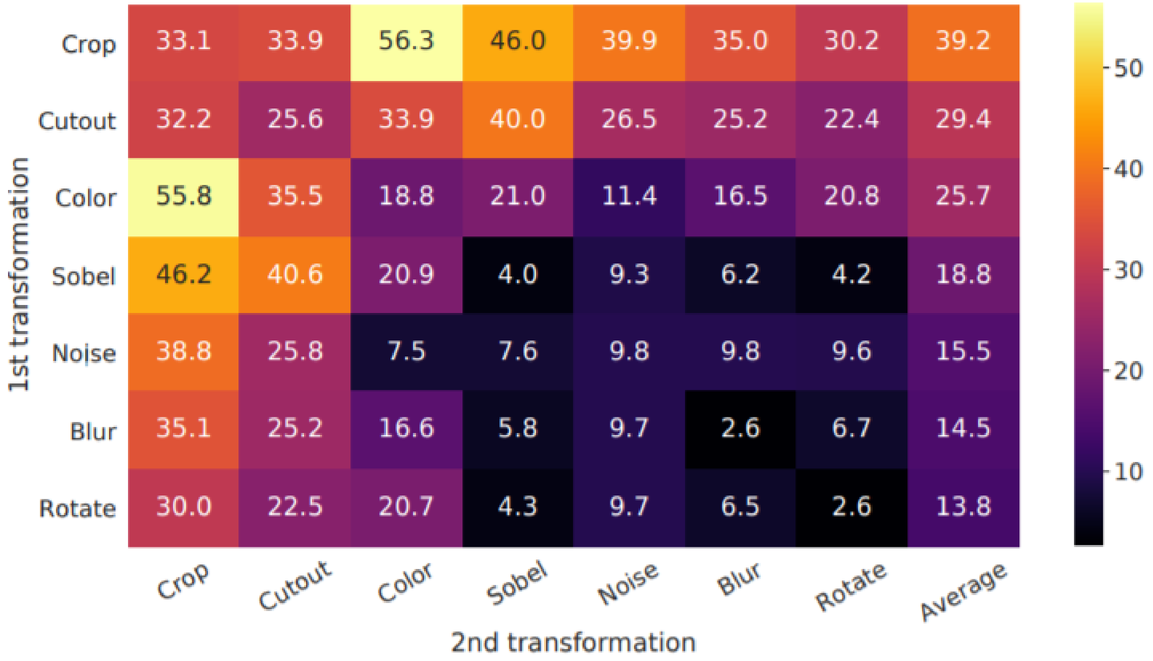
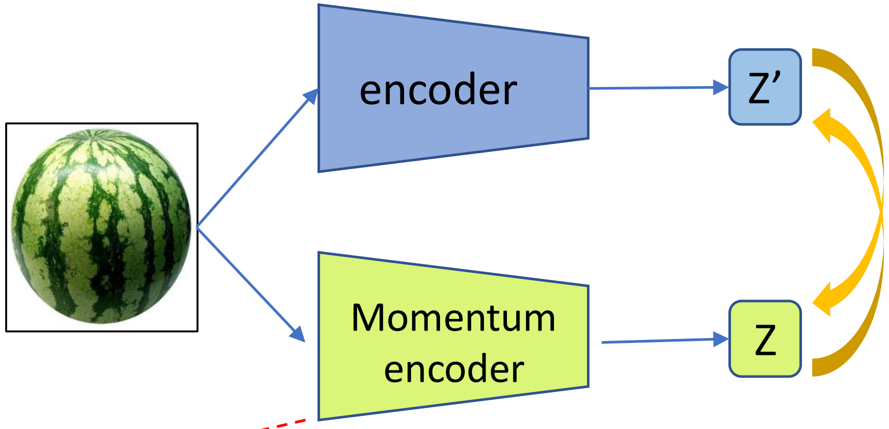


Figure 5. Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.

https://arxiv.org/pdf/2002.05709.pdf

MOCO

Knowledge can be transfer directly



Info-NCE

$$\mathcal{L}_q = -\log \frac{\exp(q{\cdot}k_+/\tau)}{\sum_{i=0}^{K}\exp(q{\cdot}k_i/\tau)}$$

$$\theta_Z \leftarrow \alpha\theta_Z + (1-\alpha)\theta_{Z'}$$

直接使用Z’比重更新Z。
其內涵為:
1. 兩個不同比重對同一件事情都能完整表達，表示這樣的表達是正確的。
2. 使用的info-NCE的並且搭配了temperature，表示在dark knowledge的部分模型也必須重視。

https://arxiv.org/pdf/1911.05722.pdf

Tips:

- Batch normalization(BN)必須要進行shuffling。

因為使用info-NCE的狀況下，BN會洩漏正樣本跟負樣本之間的訊息造成模型只看BN輸出。

- α越大越好，即momentum encoder更新幅度越小越好

MOCO v2?

1. 補上projection (from SimCLR)
2. 補上augmentation (from SimCLR)

https://arxiv.org/pdf/2003.04297.pdf

Contractive learning improve OOD detection

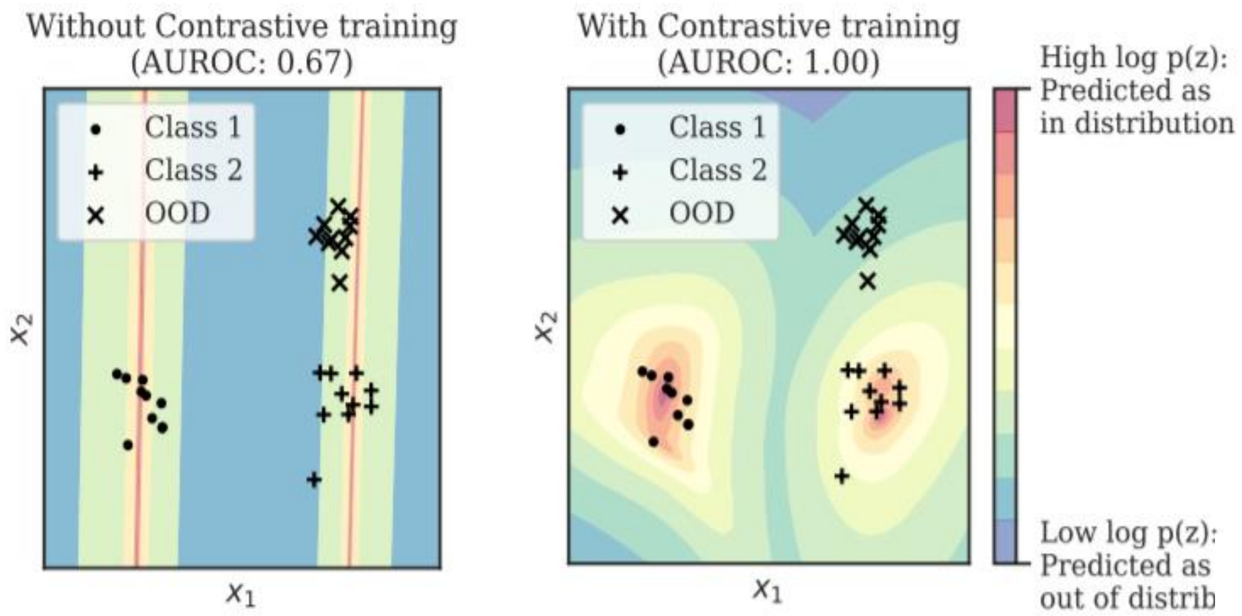


fig1

$$L_{\mathrm{con},i} = \sum_{a\in\{0,1\}} -\log \frac{\exp\left(\mathrm{sim}(\hat{\mathbf{z}}_i^a, \hat{\mathbf{z}}_i^{1-a})/\tau\right)}{\sum_{j\in\{1,\ldots,N\}} \exp\left(\mathrm{sim}(\hat{\mathbf{z}}_i^a, \hat{\mathbf{z}}_j^{1-a})/\tau\right) + \sum_{j\in\{1,\ldots,N\}\setminus i} \exp\left(\mathrm{sim}(\hat{\mathbf{z}}_i^a, \hat{\mathbf{z}}_j^{a})/\tau\right)}$$

https://arxiv.org/pdf/2007.05566.pdf

Method 1. Density estimation

$$s(\mathbf{x}) = \max_c \Big[\underbrace{-(f_\theta(\mathbf{x}) - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (f_\theta(\mathbf{x}) - \boldsymbol{\mu}_c)}_{\text{standardization}} \underbrace{- \log\left((2\pi)^n \det \boldsymbol{\Sigma}_c\right)}_{\text{ideal}} \Big]$$

Method 2. confusion log probability (CLP)

$$c_k(\mathbf{x}) = \frac{1}{N_e} \sum_{j=1}^{N_e} \hat{p}^j(\hat{y} = k|\mathbf{x}).$$

$$\mathrm{CLP}_{\mathcal{C}_{\mathrm{in}}}(\mathcal{D}_{\mathrm{test}}) = \log\left(\frac{1}{|\mathcal{D}_{\mathrm{test}}|} \sum_{\mathbf{x}\in\mathcal{D}_{\mathrm{test}}} \sum_{k\in\mathcal{C}_{\mathrm{in}}} c_k(\mathbf{x})\right)$$

Similar to SimCLR

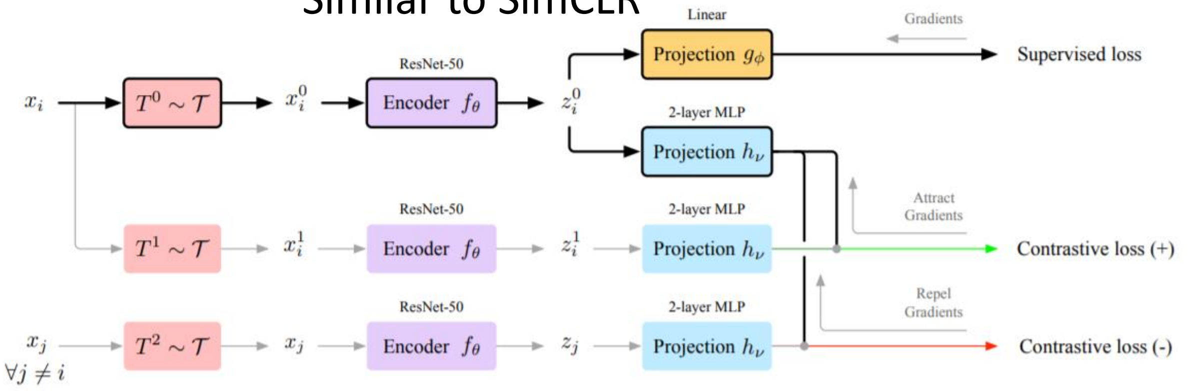
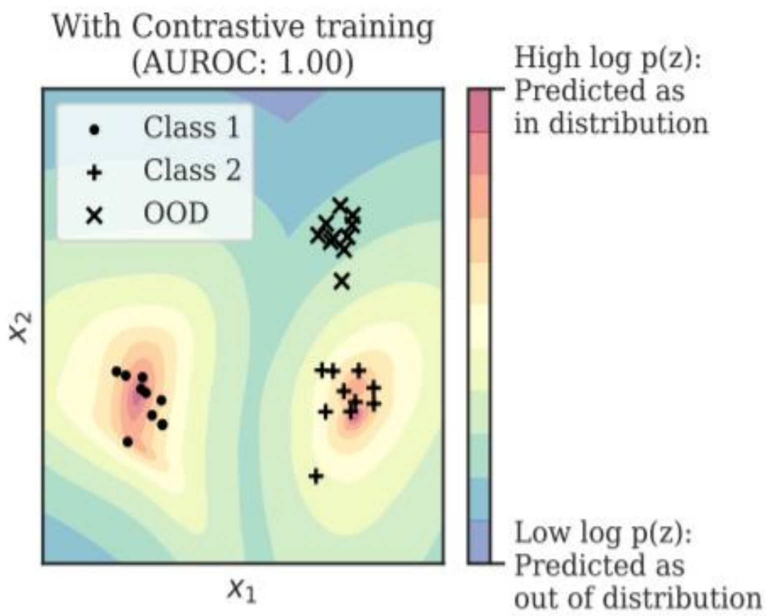
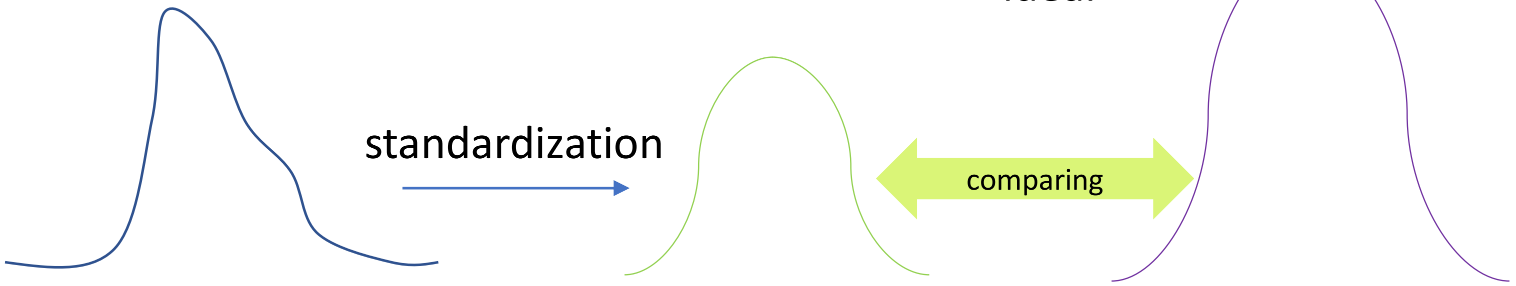


Figure 3: Schematic description of the multitask approach. $\mathbf{x}_i, \mathbf{x}_j$: training images. T: image transformation (cropping, brightness, etc.). f_θ: encoder network. $\mathbf{z}$: image represented in latent space. g_ϕ: projection to k classes. h_ν: projection to lower-dimensional embedding space.

Density estimation of contrastive learning

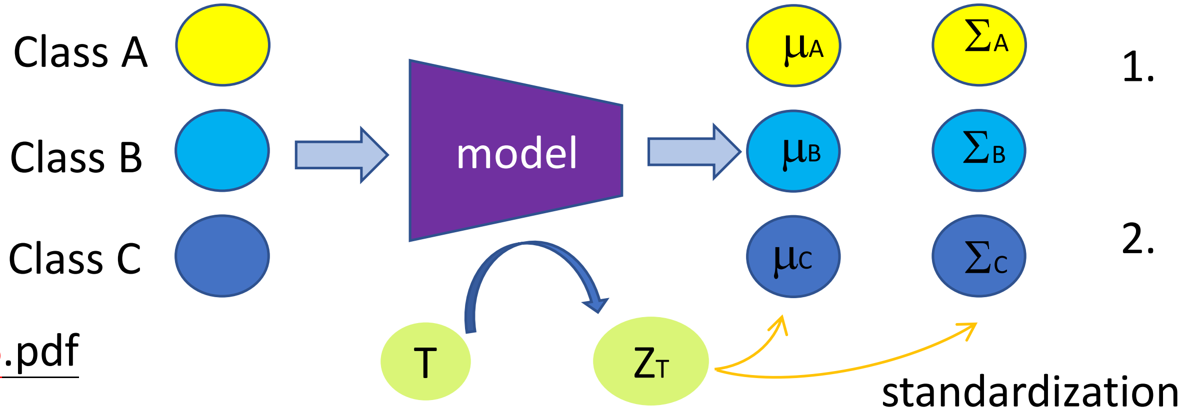


$$s(\mathbf{x}) = \max_c \Big[\underbrace{-(f_\theta(\mathbf{x}) - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (f_\theta(\mathbf{x}) - \boldsymbol{\mu}_c)}_{\text{standardization}} - \underbrace{\log\left((2\pi)^n \det \boldsymbol{\Sigma}_c\right)}_{\text{ideal}}\Big]$$



Suppose we can describe:
μ means the means
Σ means the covariance matrix

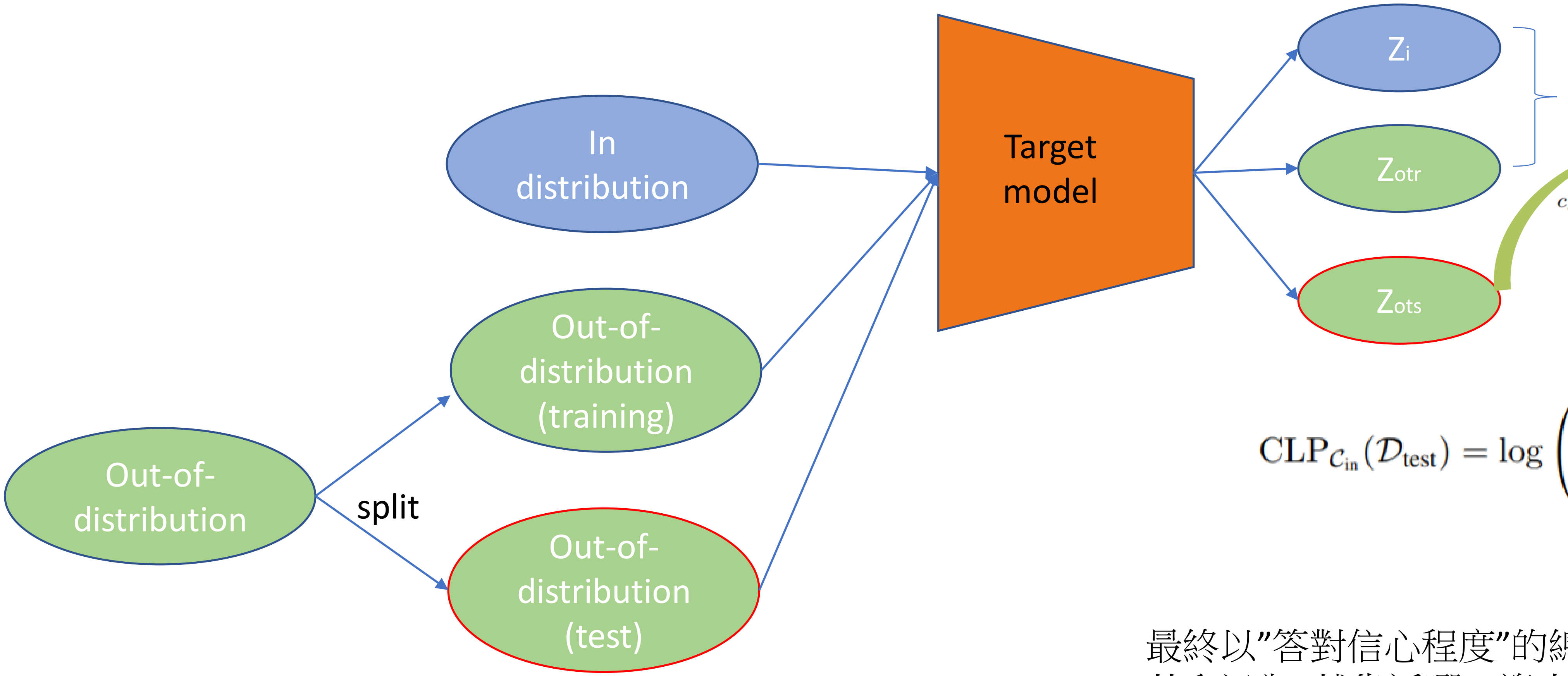
If we create the distribution only consider the ideal Σ (there are no interactions between the features)



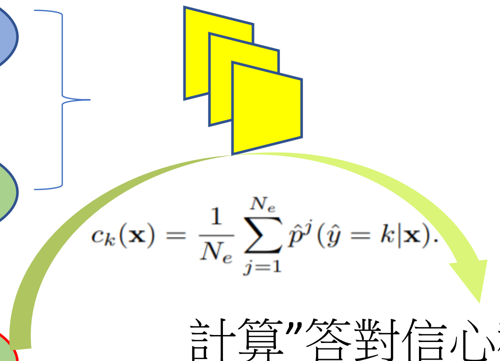
1. 把不同類別的所有訓練資料丟到模型中，算出Z的平均值及變異數
2. 未知的資料丟到模型中取的Z以後分別看會落在每個類別的哪個位置

https://arxiv.org/pdf/2007.05566.pdf

confusion log probability (CLP)



練N個小模型



$$c_k(\mathbf{x}) = \frac{1}{N_e} \sum_{j=1}^{N_e} \hat{p}^j(\hat{y} = k|\mathbf{x}).$$

計算"答對信心程度"的總和

$$\mathrm{CLP}_{\mathcal{C}_{\text{in}}}(\mathcal{D}_{\text{test}}) = \log\left(\frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathbf{x}\in\mathcal{D}_{\text{test}}} \sum_{k\in\mathcal{C}_{\text{in}}} c_k(\mathbf{x})\right)$$

最終以"答對信心程度"的總和作為結果。
其內涵為: 越靠近哪一邊表示越像該類別。

因此最後總合越小表示越屬於out-liner。

https://arxiv.org/pdf/2007.05566.pdf