Life Long Learning

Hung-yi Lee
李宏毅

https://www.pearsonlearned.com/lifelong-learning-will-help-workers-navigate-future-work/

你用同一個腦學習
機器學習的每一堂課

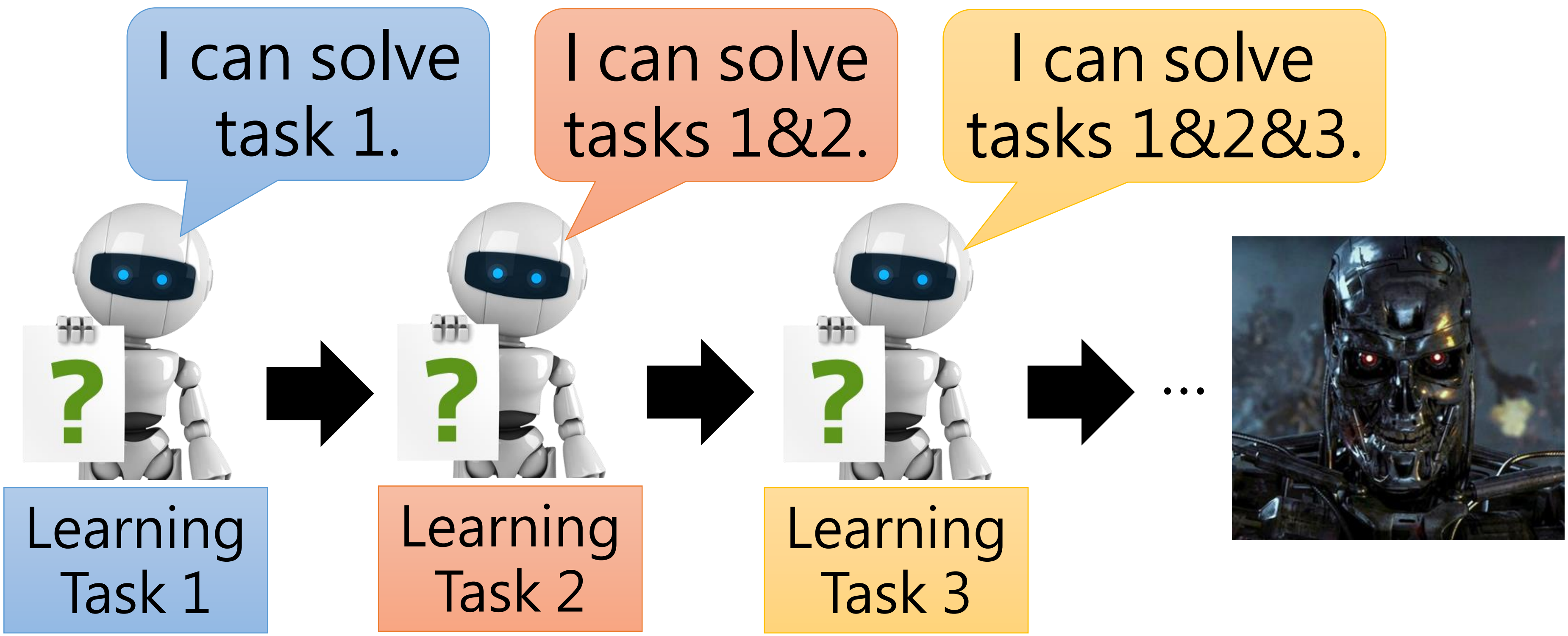
但是每一個作業你都
訓練不同的類神經網路

能不能每次作業都訓練同一個類神經網路呢？

https://www.forbes.com/sites/kpmg/2018/04/23/the-changing-nature-of-work-why-lifelong-learning-matters-more-than-ever/#4e04e90e1e95

Life Long Learning (LLL)

Continuous Learning, Never Ending Learning, Incremental Learning



Life-long Learning

Knowledge Retention

- **but NOT Intransigence**

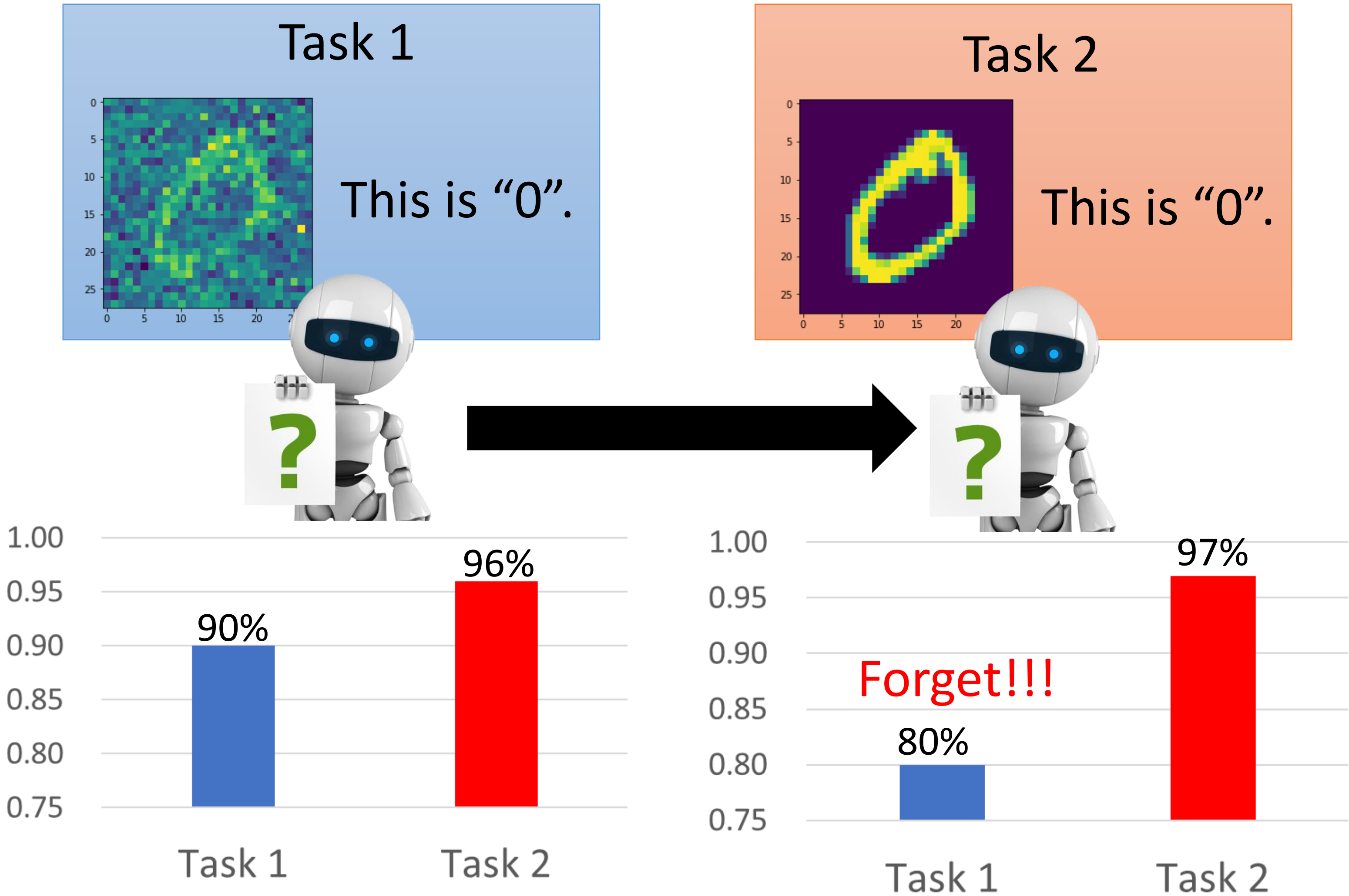
Knowledge Transfer

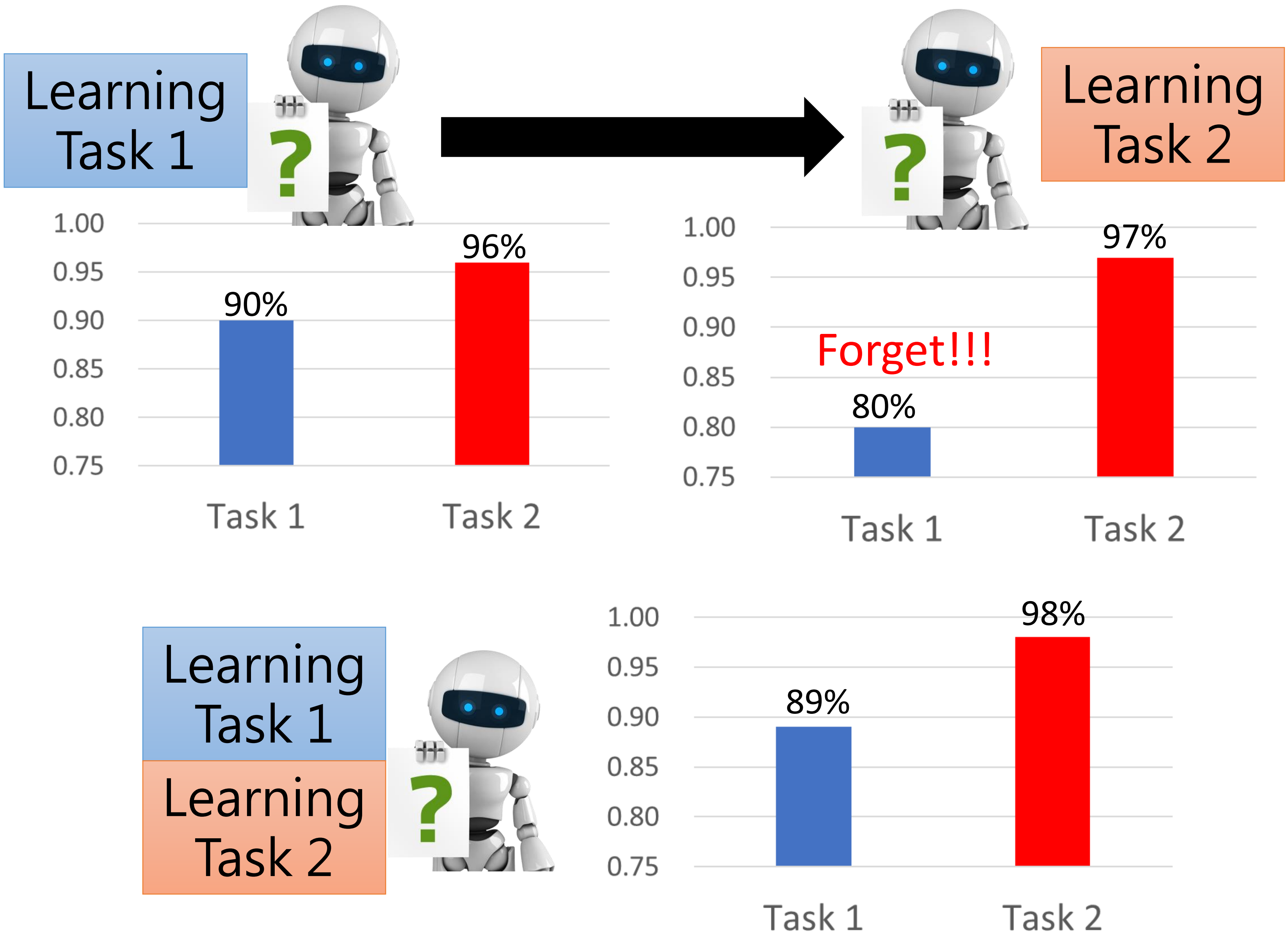
Model Expansion

- **but Parameter Efficiency**

Example – Image

= 3 layers, 50 neurons each



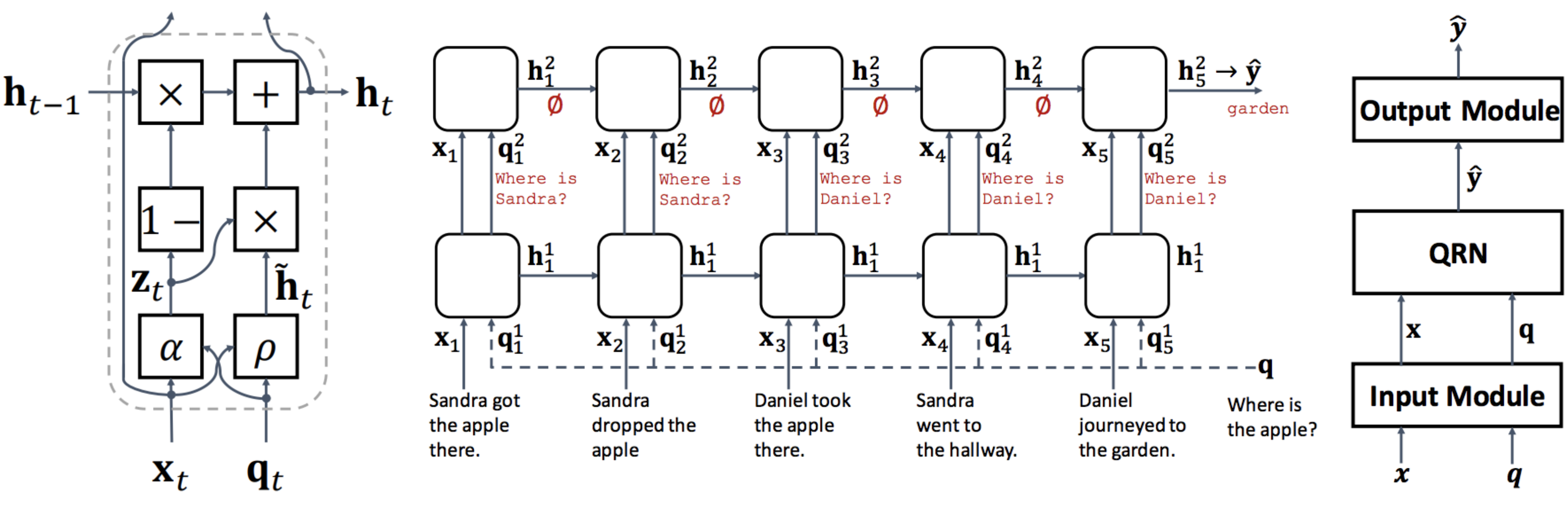


明明可以把 Task 1 和 2 都學好，為什麼會變成這樣子呢!?

https://arxiv.org/pdf/1502.05698.pdf

Example – Question Answering

- Given a document, answer the question based on the document.

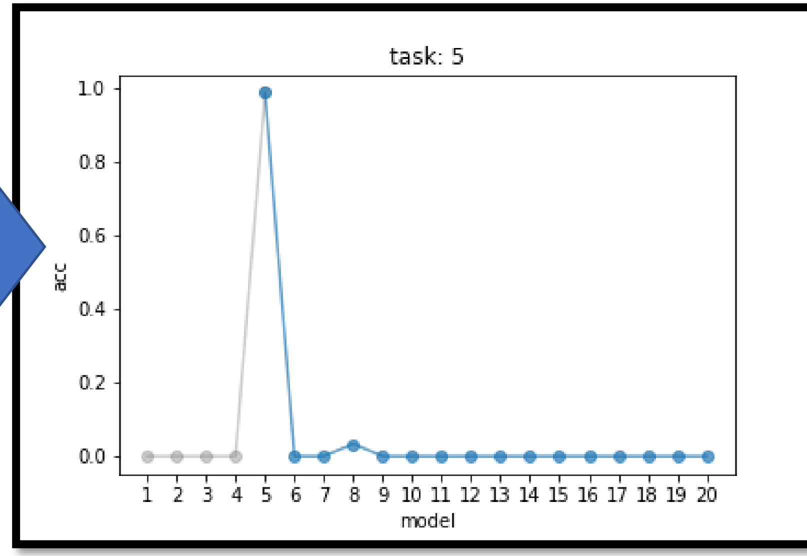


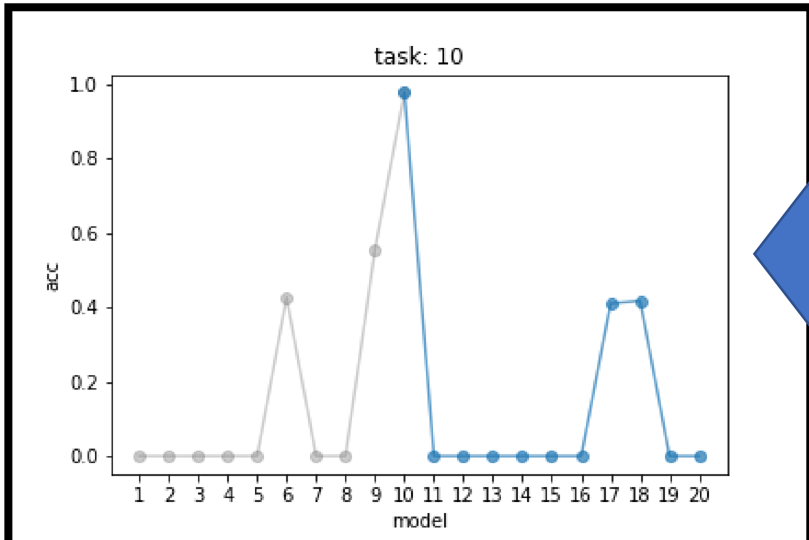
(a) QRN unit (b) 2-layer QRN (c) Overview

- There are 20 QA tasks in bAbi corpus.
- Train a QA model through the 20 tasks

Task 5: Three Argument Relations

Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill



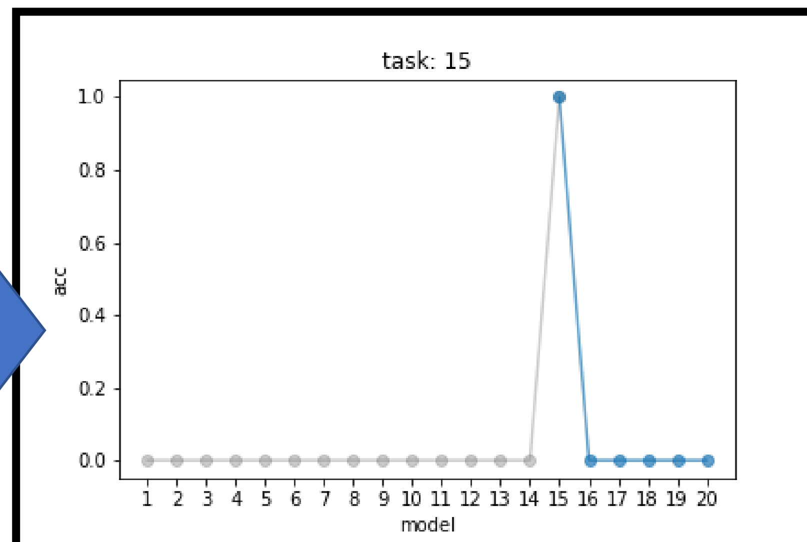


Task 10: Indefinite Knowledge

John is either in the classroom or the playground.
Sandra is in the garden.
Is John in the classroom? A:maybe
Is John in the office? A:no

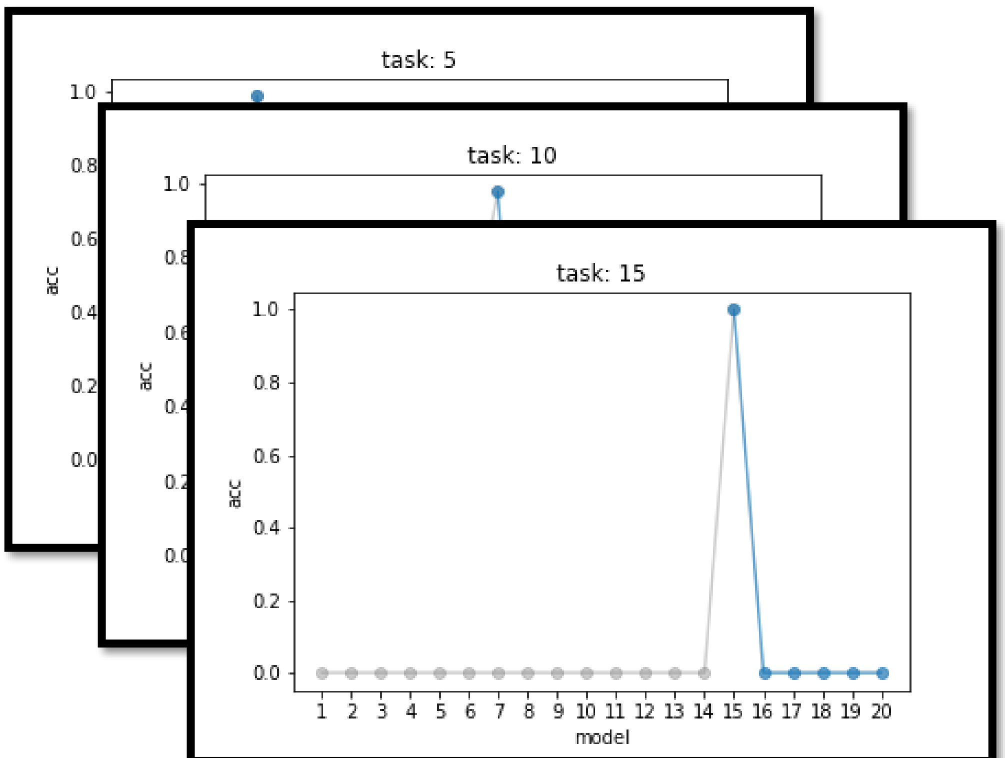
Task 15: Basic Deduction

Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves

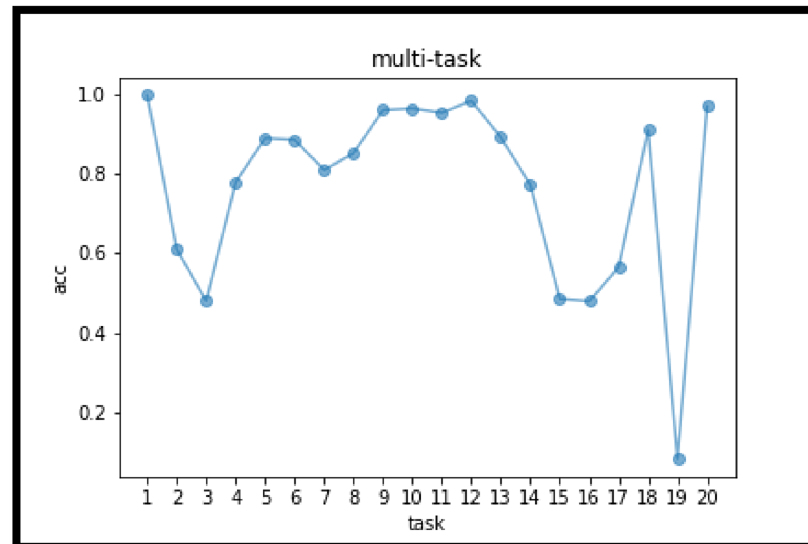


Example – Question Answering

Sequentially train the 20 tasks



Jointly training the 20 tasks



是不為也

非不能也

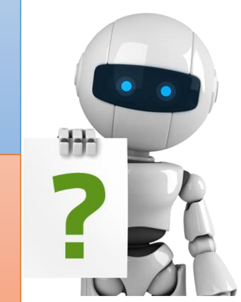
感謝何振豪同學提供實驗結果

Catastrophic Forgetting

Wait a minute ……

Learning Task 1

Learning Task 2



- Multi-task training can solve the problem!

Using all the data for training → **Computation issue**

Training Data for Task 1 …… Training Data for Task 999 | Training Data for Task 1000

Always keep the data → **Storage issue**

- Multi-task training can be considered as the upper bound of LLL.

Elastic Weight Consolidation (EWC)

Basic Idea: Some parameters in the model are important to the previous tasks. Only change the unimportant parameters.

θ^b is the model learned from the previous tasks.

Each parameter θ_i^b has a “guard” b_i

How important this parameter is

Loss for current task

$$\underline{L'(\theta)} = \underline{L(\theta)} + \lambda \sum_i \underline{b_i}\left(\underline{\theta_i} - \underline{\theta_i^b}\right)^2$$

Loss to be optimized

Parameters to be learning

Parameters learned from previous task

Elastic Weight Consolidation (EWC)

Basic Idea: Some parameters in the model are important to the previous tasks. Only change the unimportant parameters.

θ^b is the model learned from the previous tasks.

Each parameter θ_i^b has a "guard" b_i

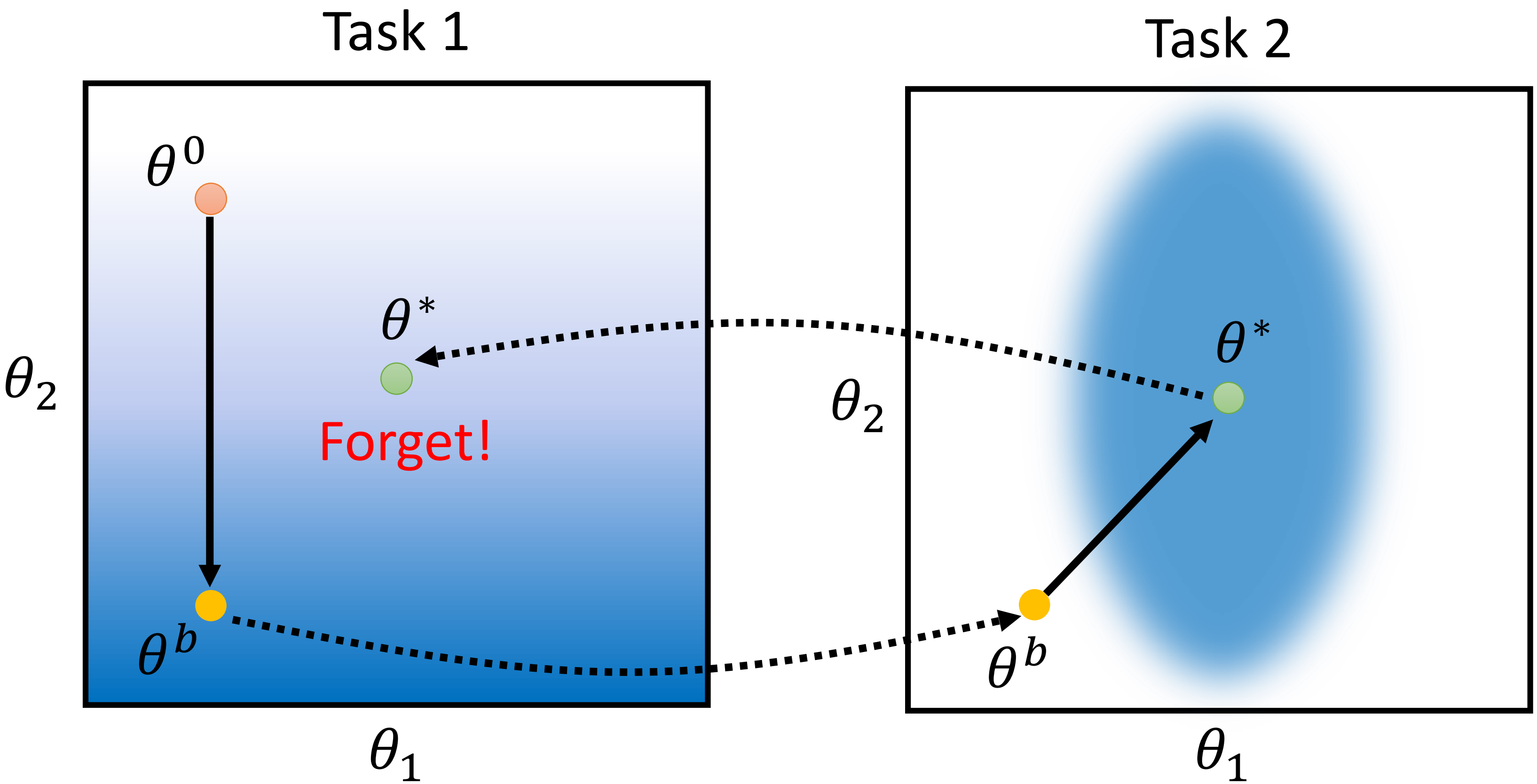
One kind of regularization. θ_i should be close to θ^b in certain directions.

$$L'(\theta) = L(\theta) + \lambda \boxed{\sum_i b_i \left(\theta_i - \theta_i^b\right)^2}$$

If $b_i = 0$, there is no constraint on θ_i

If $b_i = \infty$, θ_i would always be equal to θ_i^b

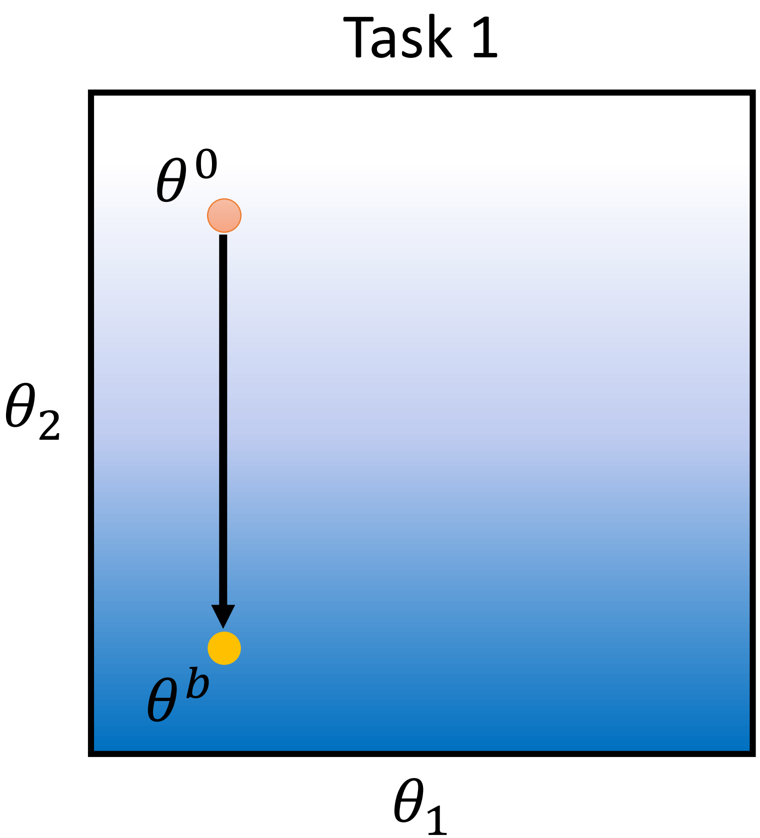
Elastic Weight Consolidation (EWC)



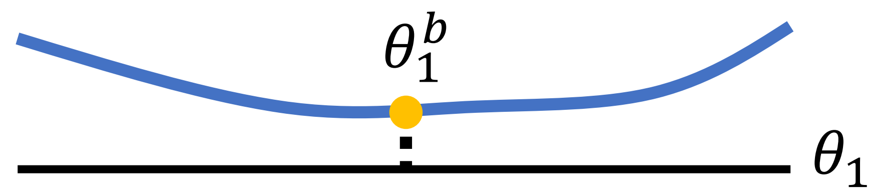
The error surfaces of tasks 1 & 2.

(darker = smaller loss)

Elastic Weight Consolidation (EWC)

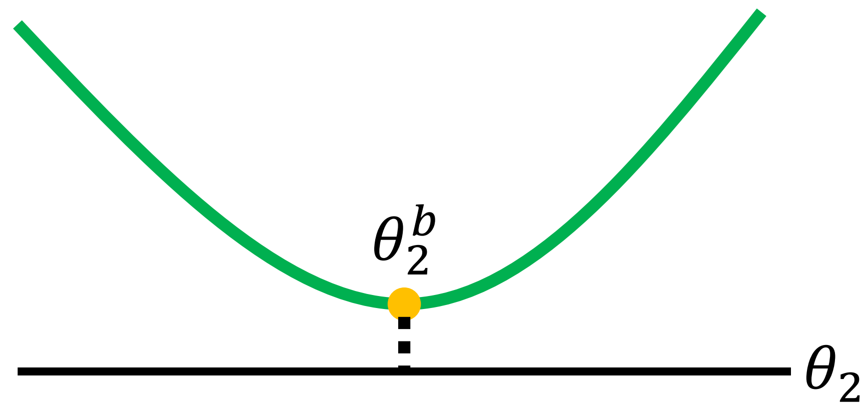


Each parameter has a "guard" b_i



Small 2nd derivative

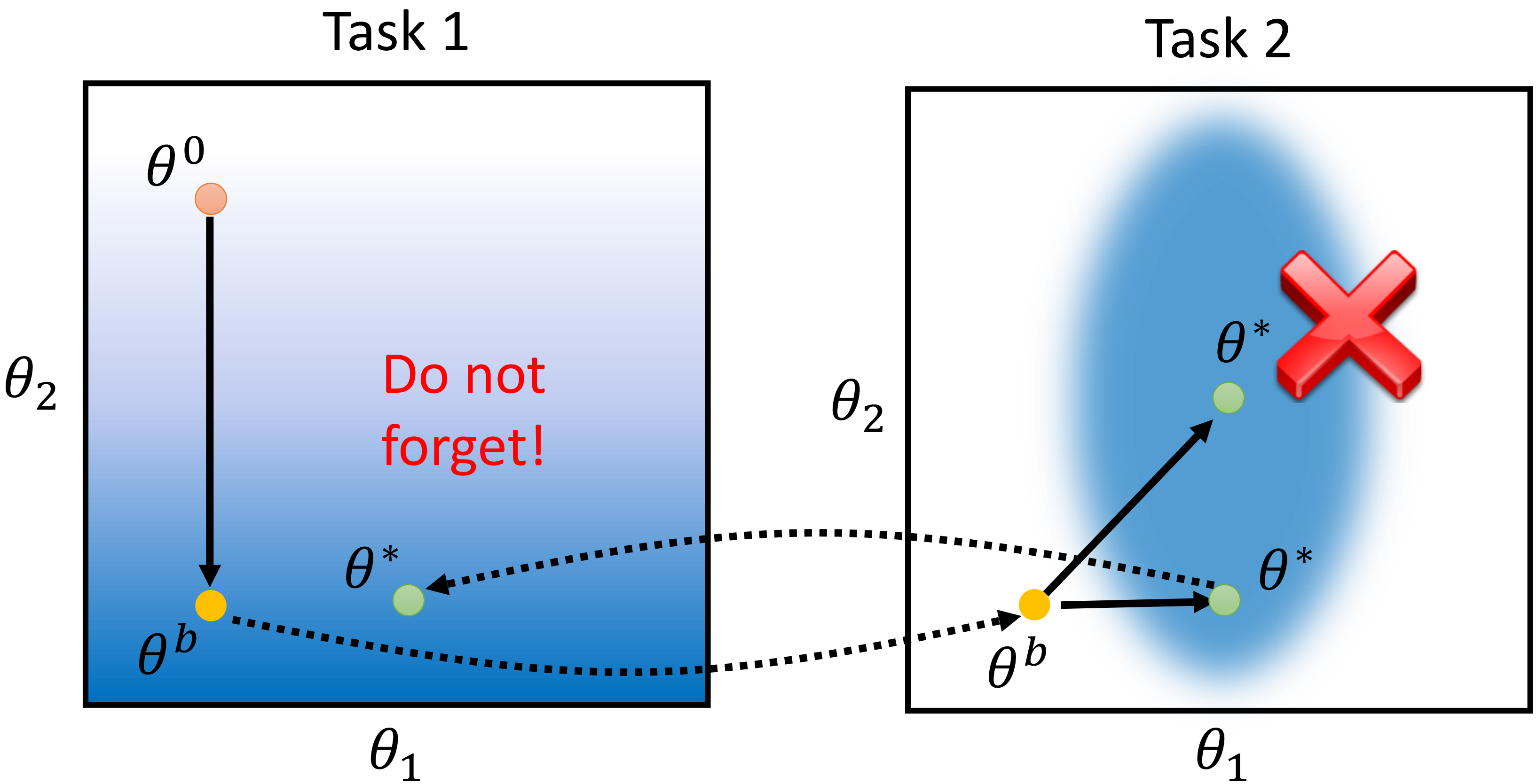
➡ b_1 is small 動到沒關係



Large 2nd derivative

➡ b_2 is large 動到會出事

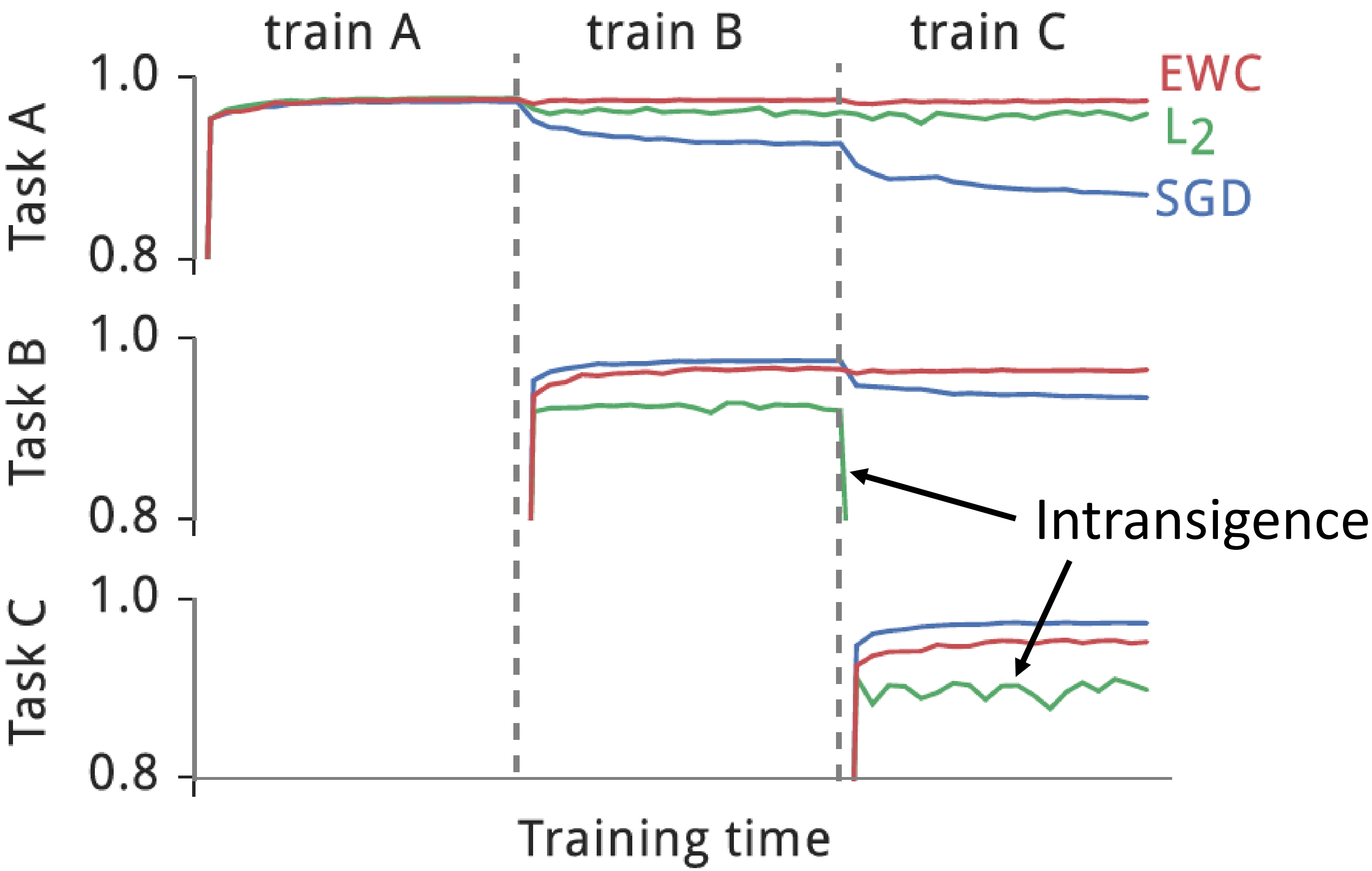
Elastic Weight Consolidation (EWC)



b_1 is small, while b_2 is large.

(可以動 θ_1，但儘量不要動到 θ_2)

Elastic Weight Consolidation (EWC)



MNIST permutation, from the original EWC paper

Elastic Weight Consolidation (EWC)

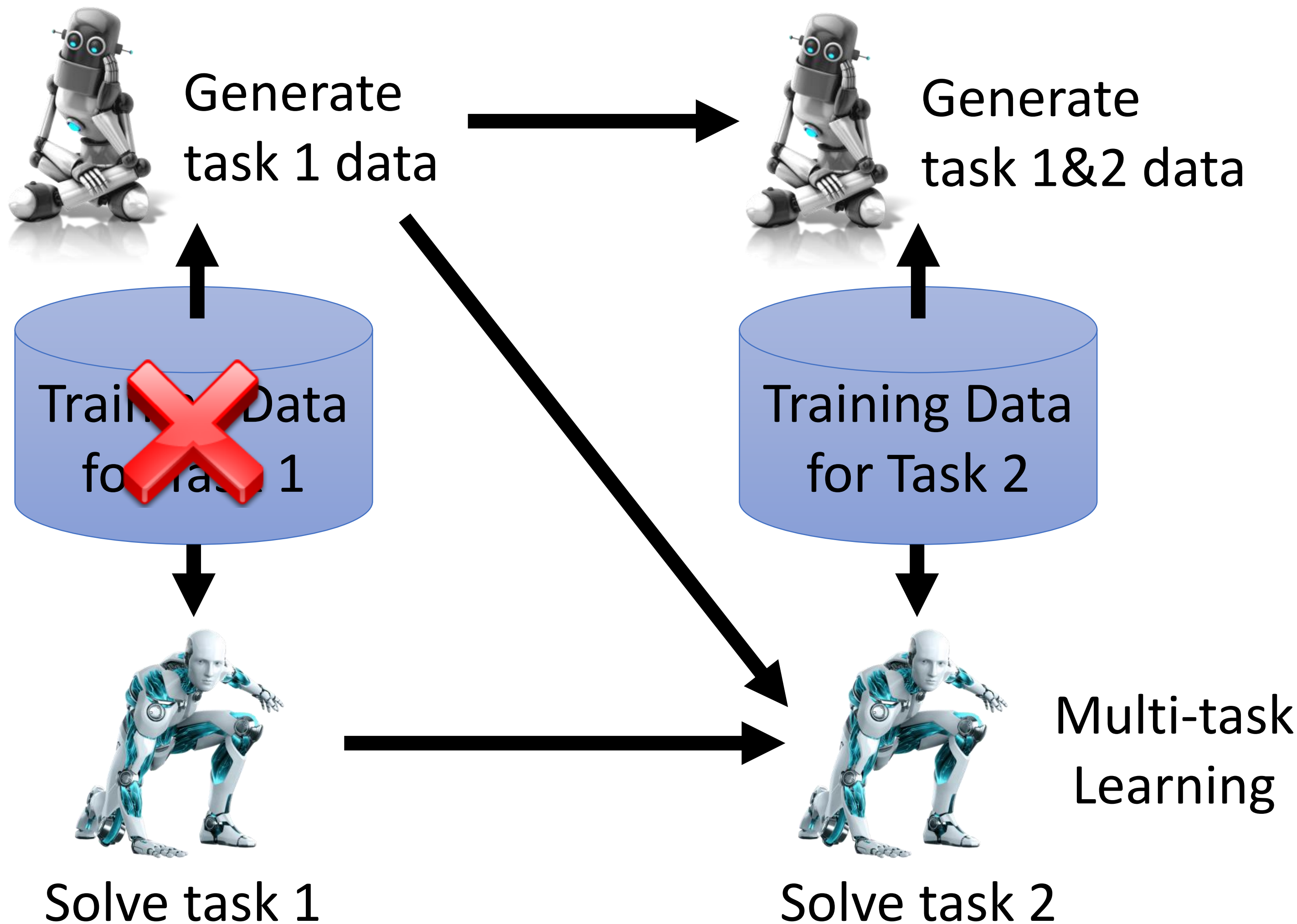
- Elastic Weight Consolidation (EWC)
  - http://www.citeulike.org/group/15400/article/14311063
- Synaptic Intelligence (SI)
  - https://arxiv.org/abs/1703.04200
- Memory Aware Synapses (MAS)
  - Special part: Do not need labelled data
  - https://arxiv.org/abs/1711.09601

Synaptic: 突觸的
Synapsis: 突觸

Generating Data

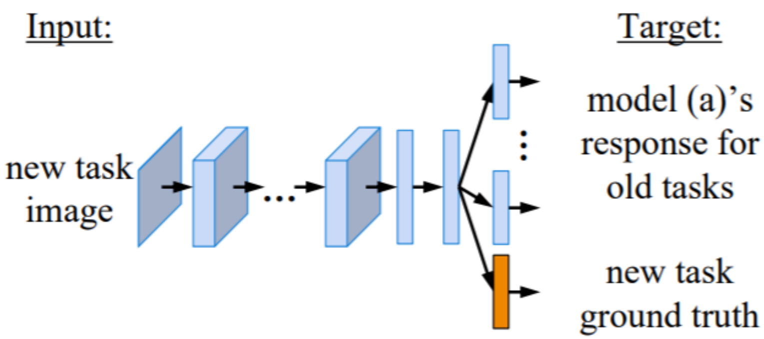
https://arxiv.org/abs/1705.08690
https://arxiv.org/abs/1711.10563

- Conducting multi-task learning by generating pseudo-data using generative model

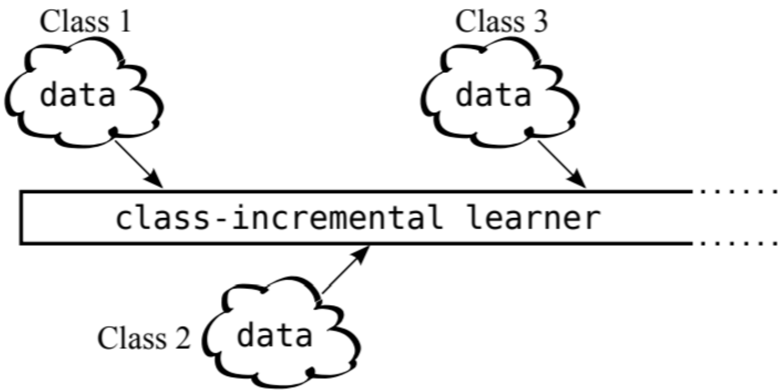


Adding New Classes

- Learning without forgetting (LwF)
  - https://arxiv.org/abs/1606.09282



- iCaRL: Incremental Classifier and Representation Learning
  - https://arxiv.org/abs/1611.07725



Life-long Learning

Knowledge Retention

- **but NOT Intransigence**

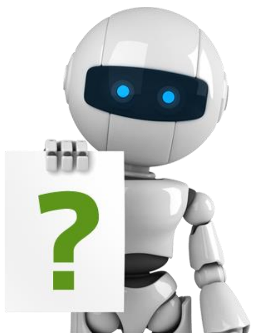
Knowledge Transfer

Model Expansion

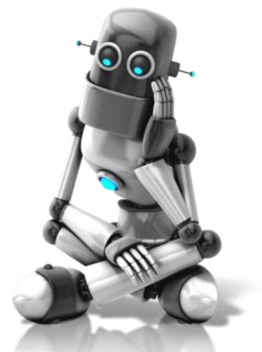
- **but Parameter Efficiency**

Wait a minute ……

- Train a model for each task



Learning Task 1



Learning Task 2

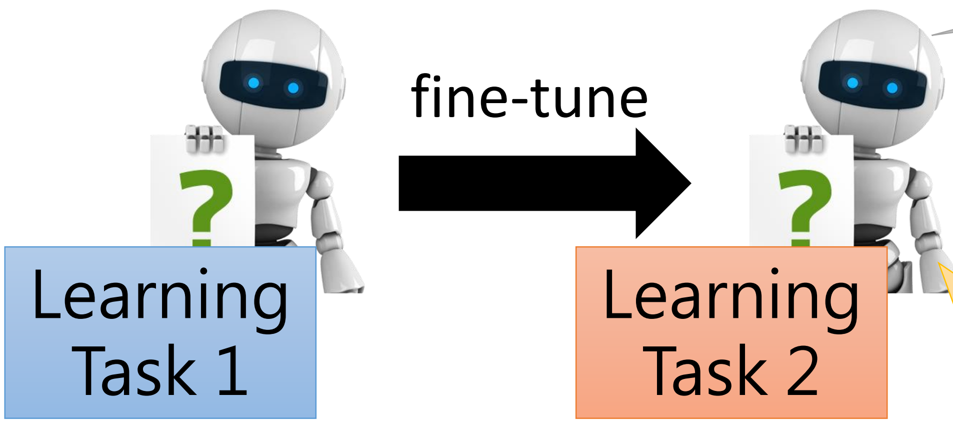
Learning Task 3

➢ Knowledge cannot transfer across different tasks

➢ Eventually we cannot store all the models …

Life-Long v.s. Transfer

Transfer Learning: I can do task 2 because I have learned task 1.



(We don't care whether machine can still do task 1.)

Life-long Learning: Even though I have learned task 2, I do not forget task 1.

Evaluation

$R_{i,j}$: after training task i, performance on task j

If $i > j$,

After training task i, does task j be forgot

If $i < j$,

Can we transfer the skill of task i to task j

		Test on			
		Task 1	Task 2	……	Task T
Rand Init.		$R_{0,1}$	$R_{0,2}$		$R_{0,T}$
After Training	Task 1	$R_{1,1}$	$R_{1,2}$		$R_{1,T}$
	Task 2	$R_{2,1}$	$R_{2,2}$		$R_{2,T}$
	⋮				
	Task T-1	$R_{T-1,1}$	$R_{T-1,2}$		$R_{T-1,T}$
	Task T	$R_{T,1}$	$R_{T,2}$		$R_{T,T}$

$$\text{Accuracy} = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}$$

$$\text{Backward Transfer} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i}$$

(It is usually negative.)

Evaluation

$R_{i,j}$: after training task i, performance on task j

If $i > j$,

After training task i, does task j be forgot

If $i < j$,

Can we transfer the skill of task i to task j

		Test on			
		Task 1	Task 2	……	Task T
Rand Init.		$R_{0,1}$	$R_{0,2}$		$\boxed{R_{0,T}}$
After Training	Task 1	$R_{1,1}$	$R_{1,2}$		$R_{1,T}$
	Task 2	$R_{2,1}$	$R_{2,2}$		$R_{2,T}$
	⋮				
	Task T-1	$R_{T-1,1}$	$R_{T-1,2}$		$\boxed{R_{T-1,T}}$
	Task T	$R_{T,1}$	$R_{T,2}$		$R_{T,T}$

$$\text{Accuracy} = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}$$

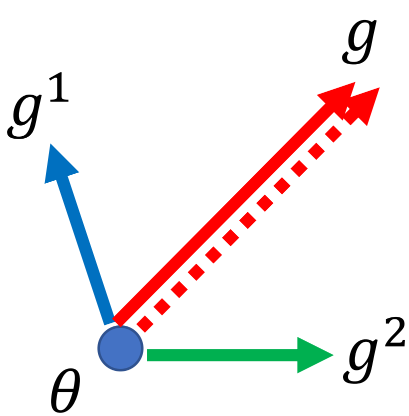
$$\text{Backward Transfer} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i}$$

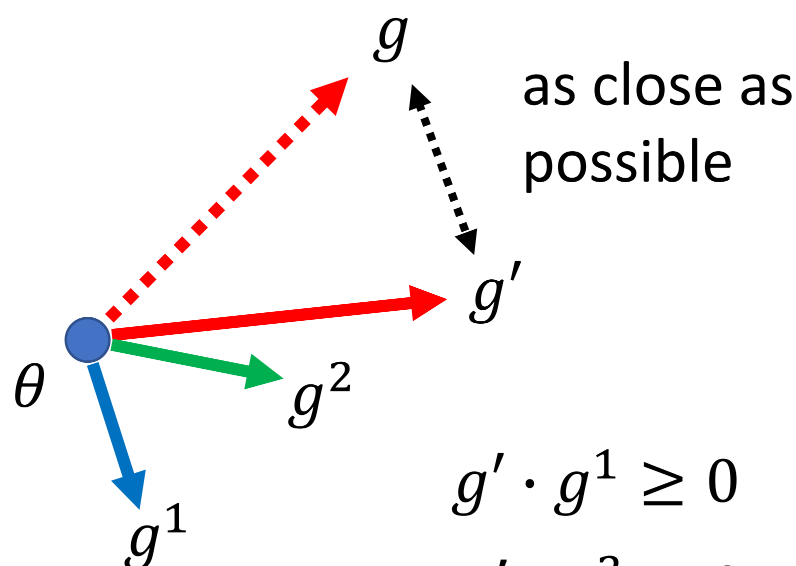
$$\text{Forward Transfer} = \frac{1}{T-1}\sum_{i=2}^{T} R_{i-1,i} - R_{0,i}$$

GEM: https://arxiv.org/abs/1706.08840
A-GEM: https://arxiv.org/abs/1812.00420

Gradient Episodic Memory (GEM)

- Constraint the gradient to improve the previous tasks





$$g' \cdot g^1 \geq 0$$

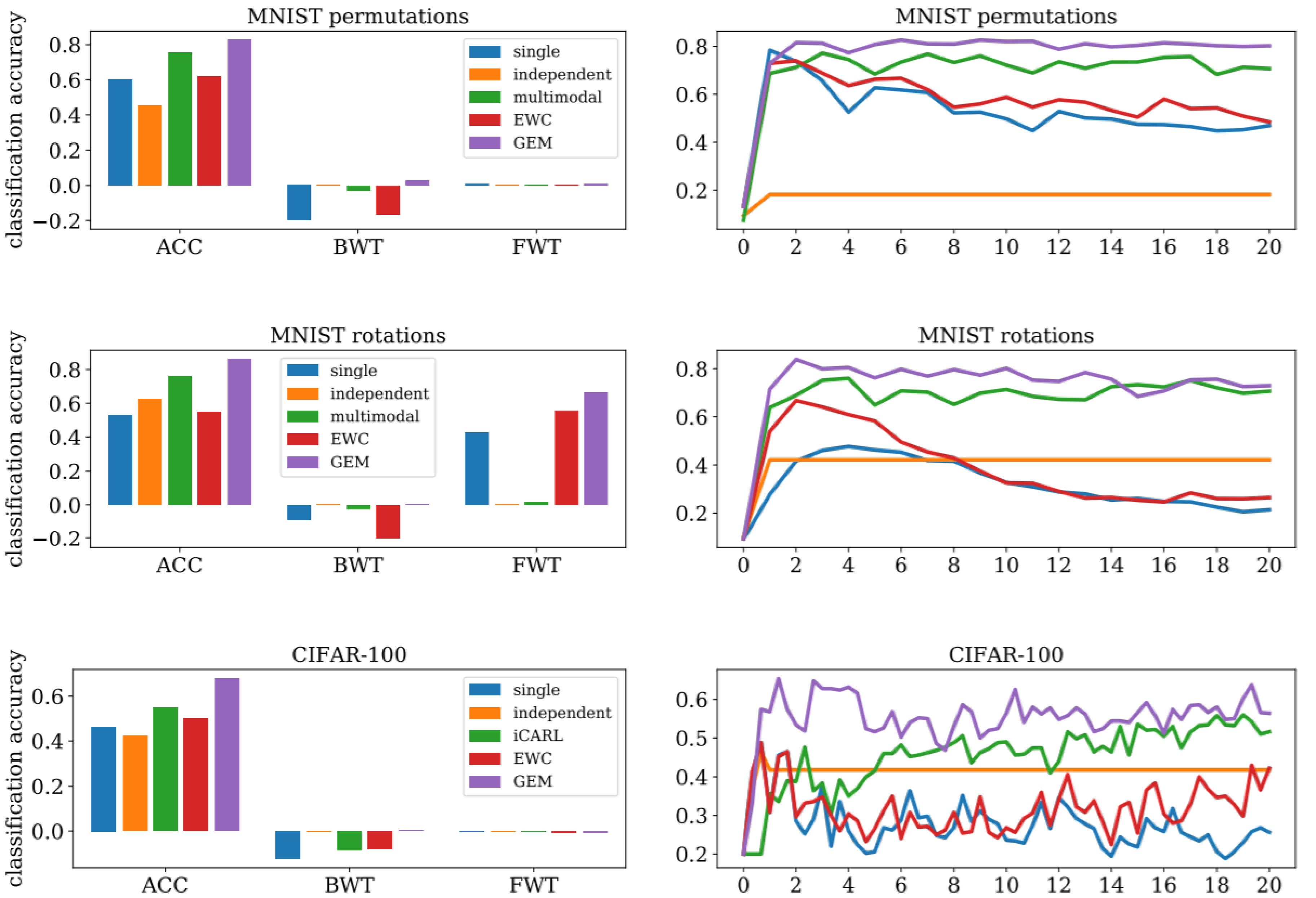
$$g' \cdot g^2 \geq 0$$

: negative gradient of current task

negative gradient of previous task

: update direction

Need the data from the previous tasks


MNIST permutations
classification accuracy
ACC
BWT
FWT
single
independent
multimodal
EWC
GEM
MNIST permutations
MNIST rotations
classification accuracy
ACC
BWT
FWT
single
independent
multimodal
EWC
GEM
MNIST rotations
CIFAR-100
classification accuracy
ACC
BWT
FWT
single
independent
iCARL
EWC
GEM
CIFAR-100

Life-long Learning

Knowledge Retention

- **but NOT Intransigence**

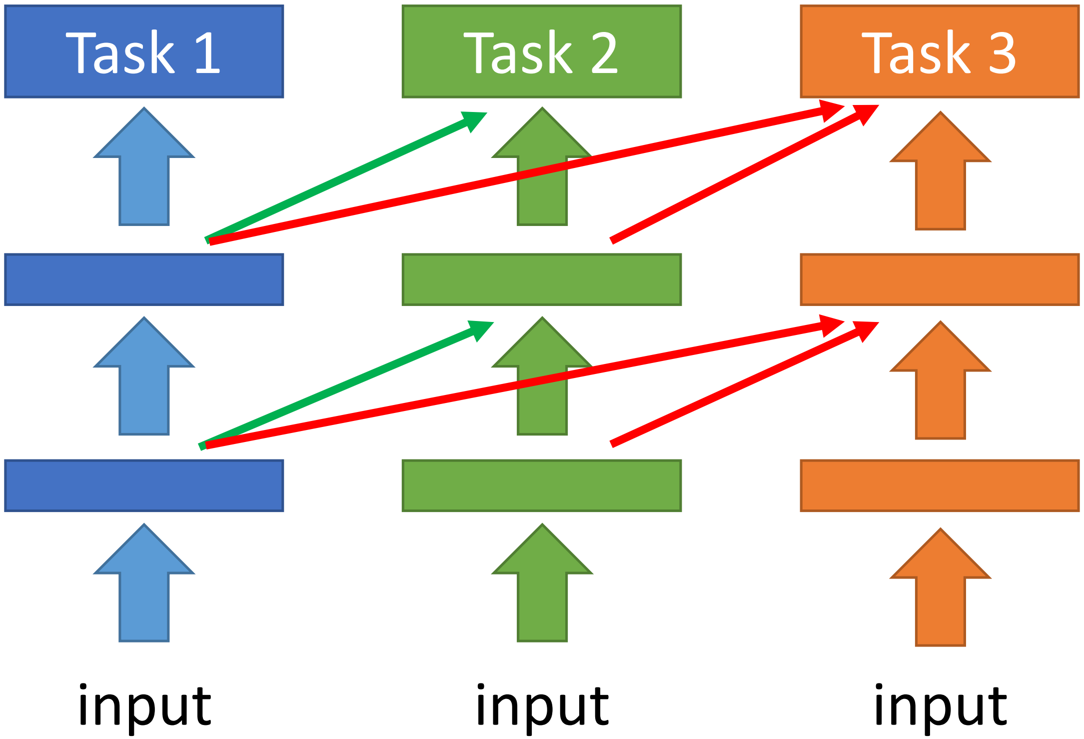
Knowledge Transfer

Model Expansion

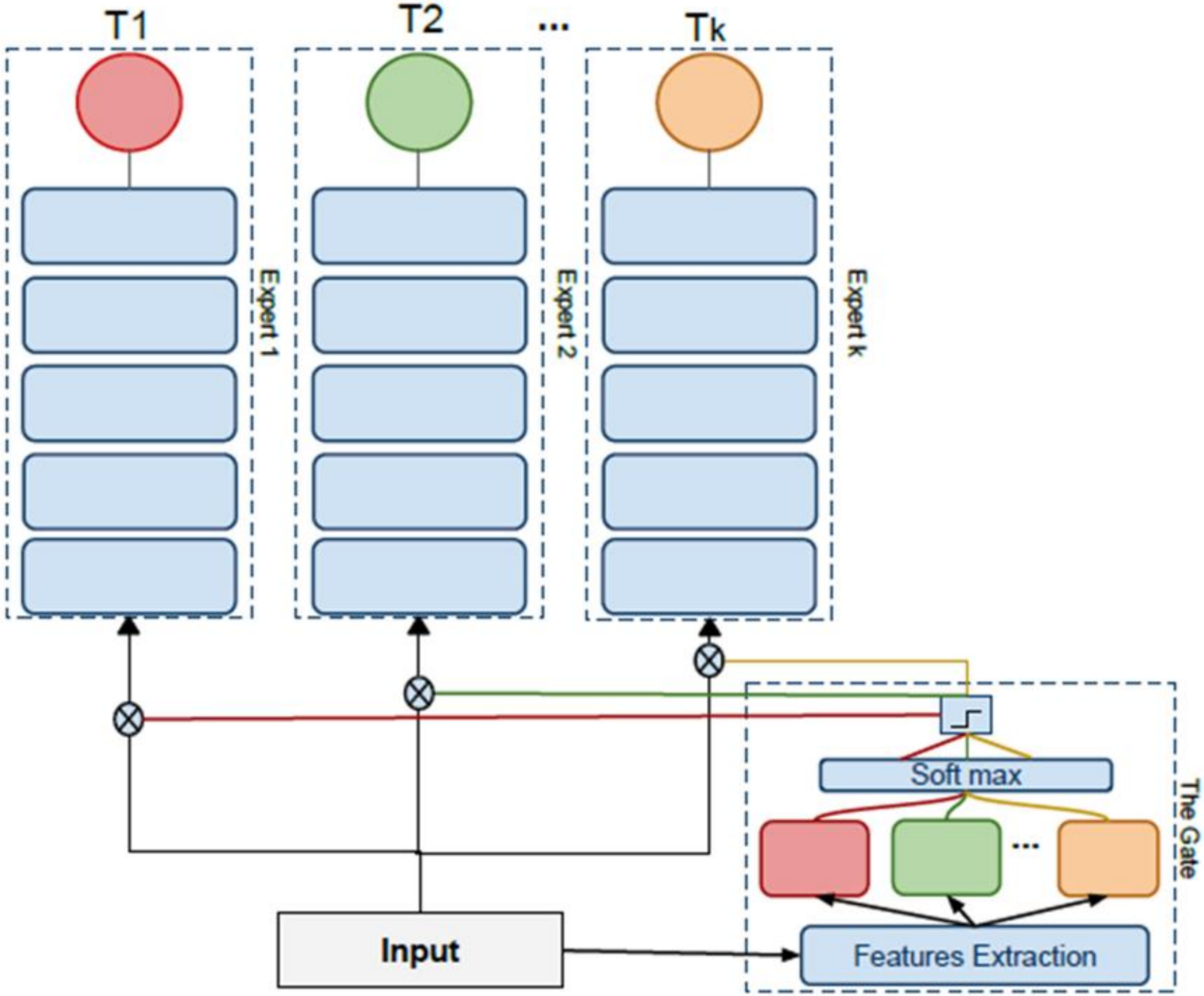
- **but Parameter Efficiency**

https://arxiv.org/abs/1606.04671

Progressive Neural Networks

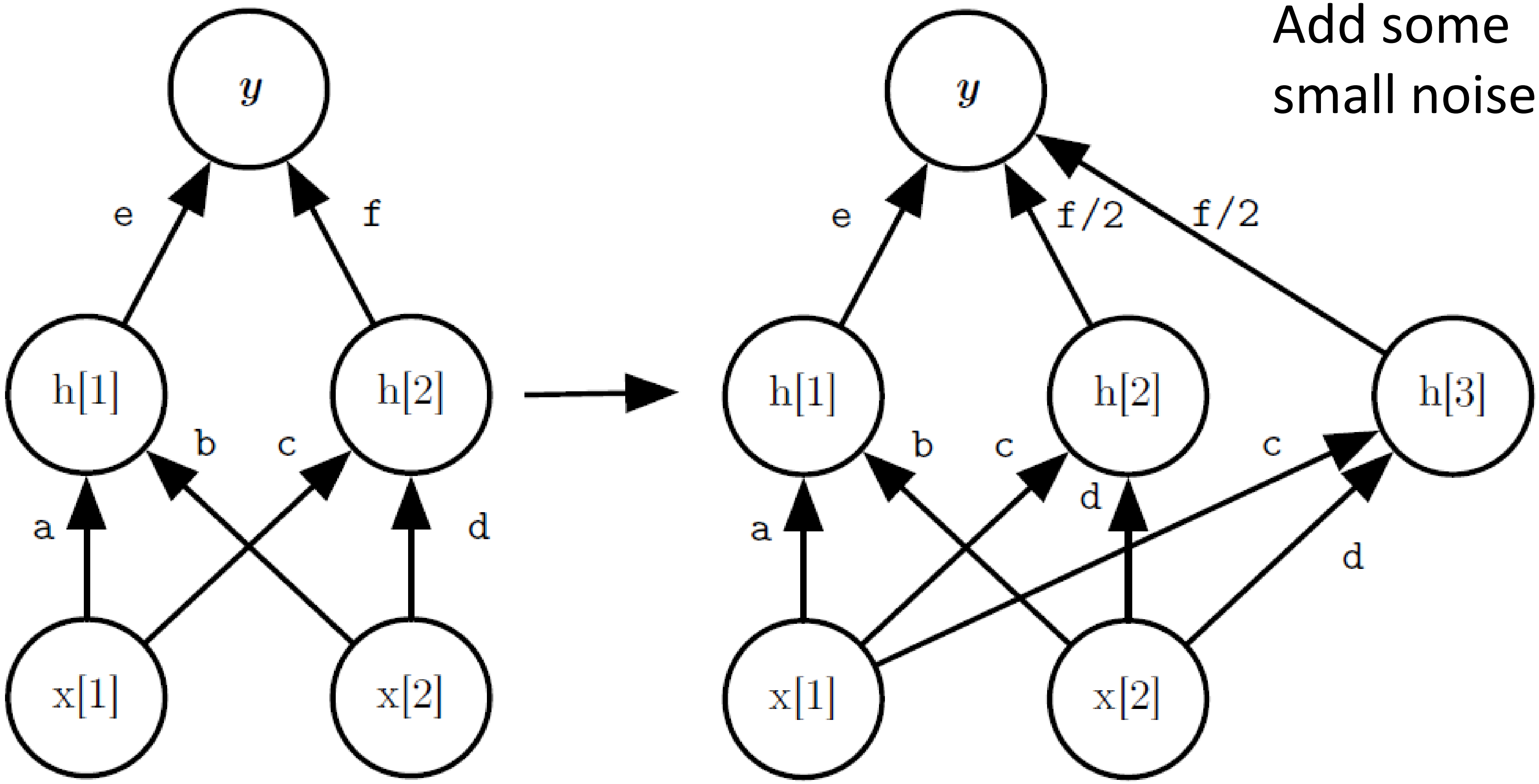


Expert Gate
https://arxiv.org/abs/1611.06194



https://arxiv.org/abs/1511.05641

Net2Net



Add some small noises

Expand the network only when the training accuracy of the current task is not good enough.

https://arxiv.org/abs/1811.07017

Concluding Remarks

Knowledge Retention

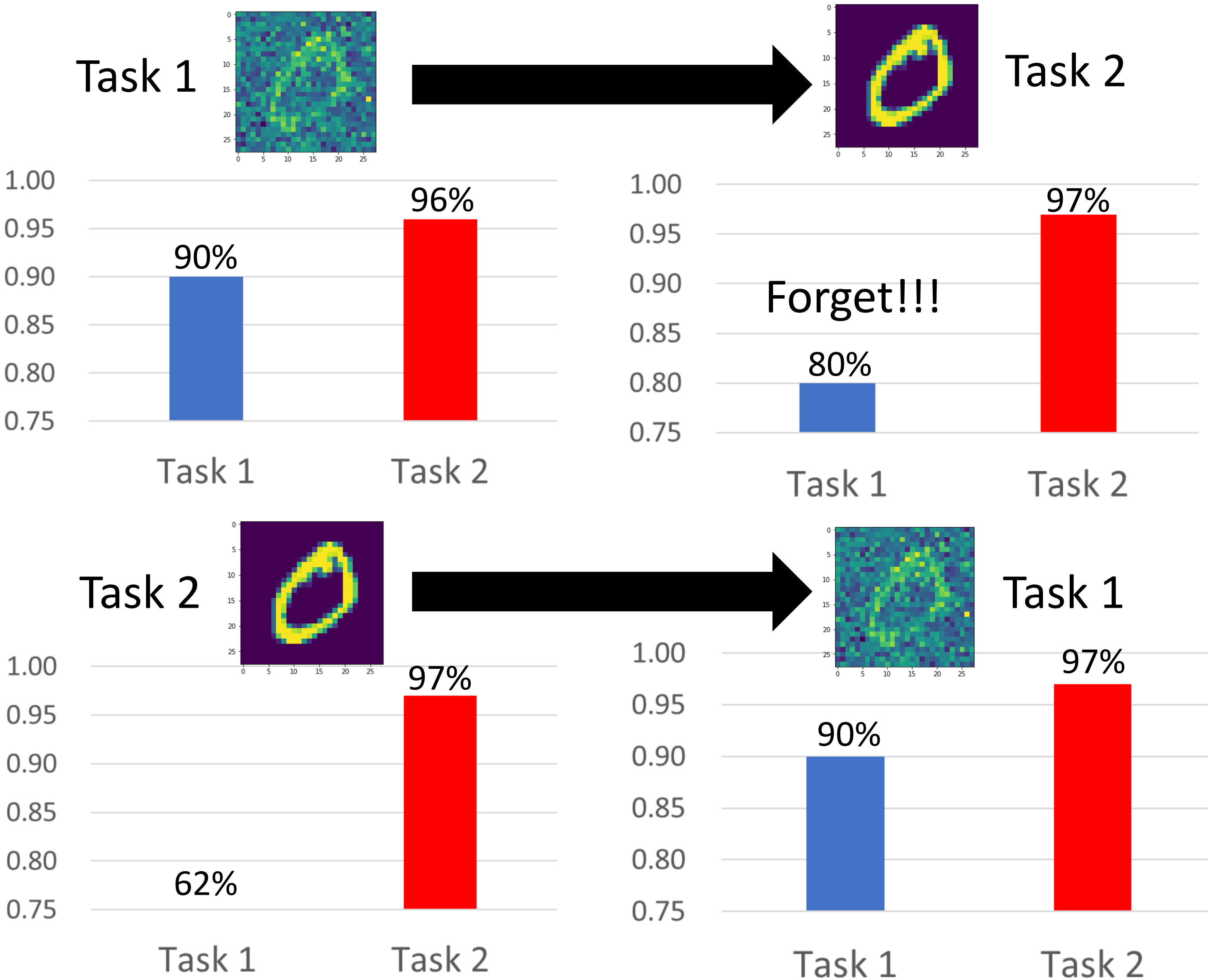
- **but NOT Intransigence**

Knowledge Transfer

Model Expansion

- **but Parameter Efficiency**

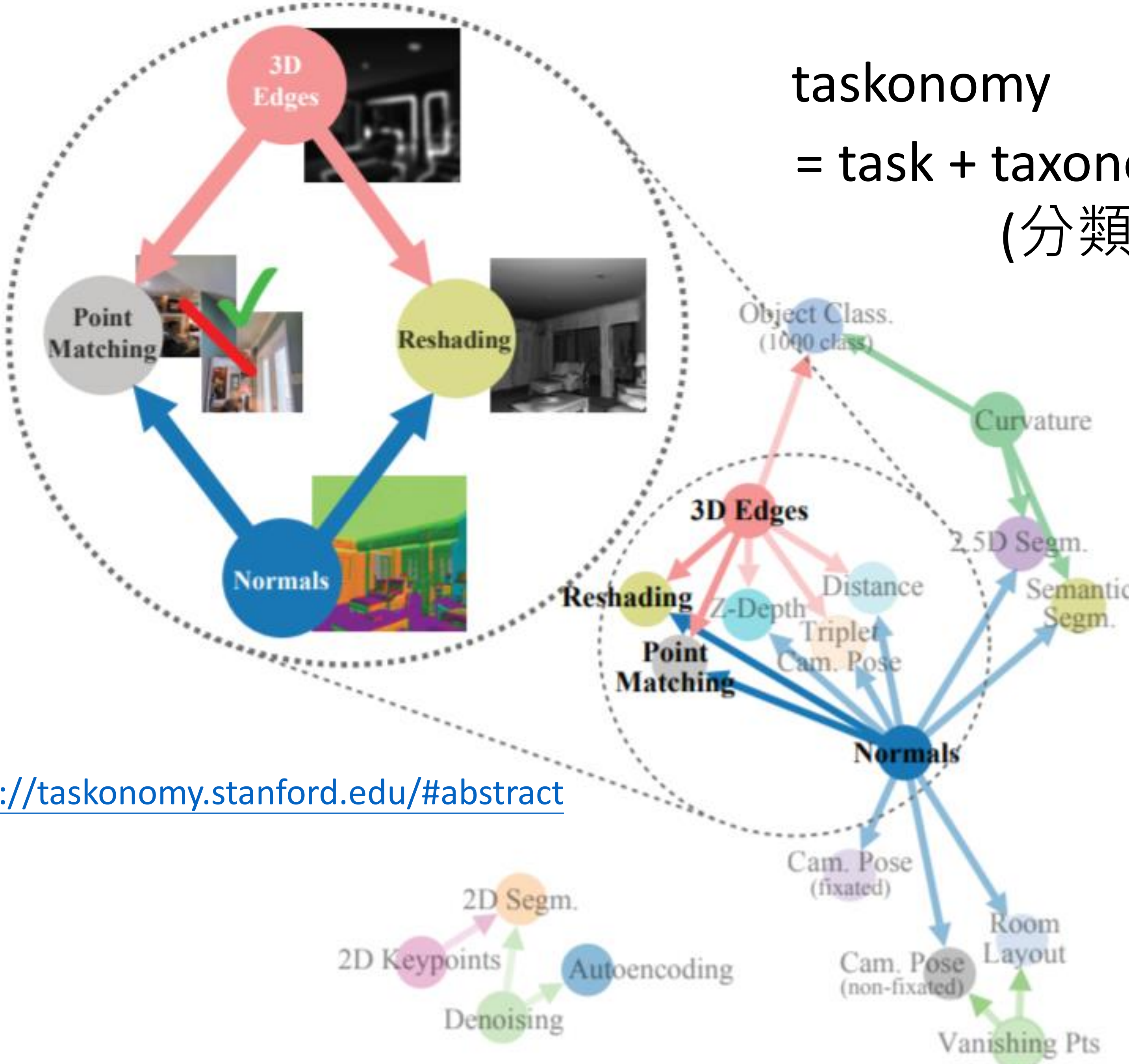
Curriculum Learning : what is the proper learning order?



taskonomy

= task + taxonomy

(分類學)



http://taskonomy.stanford.edu/#abstract