Life Long Learning

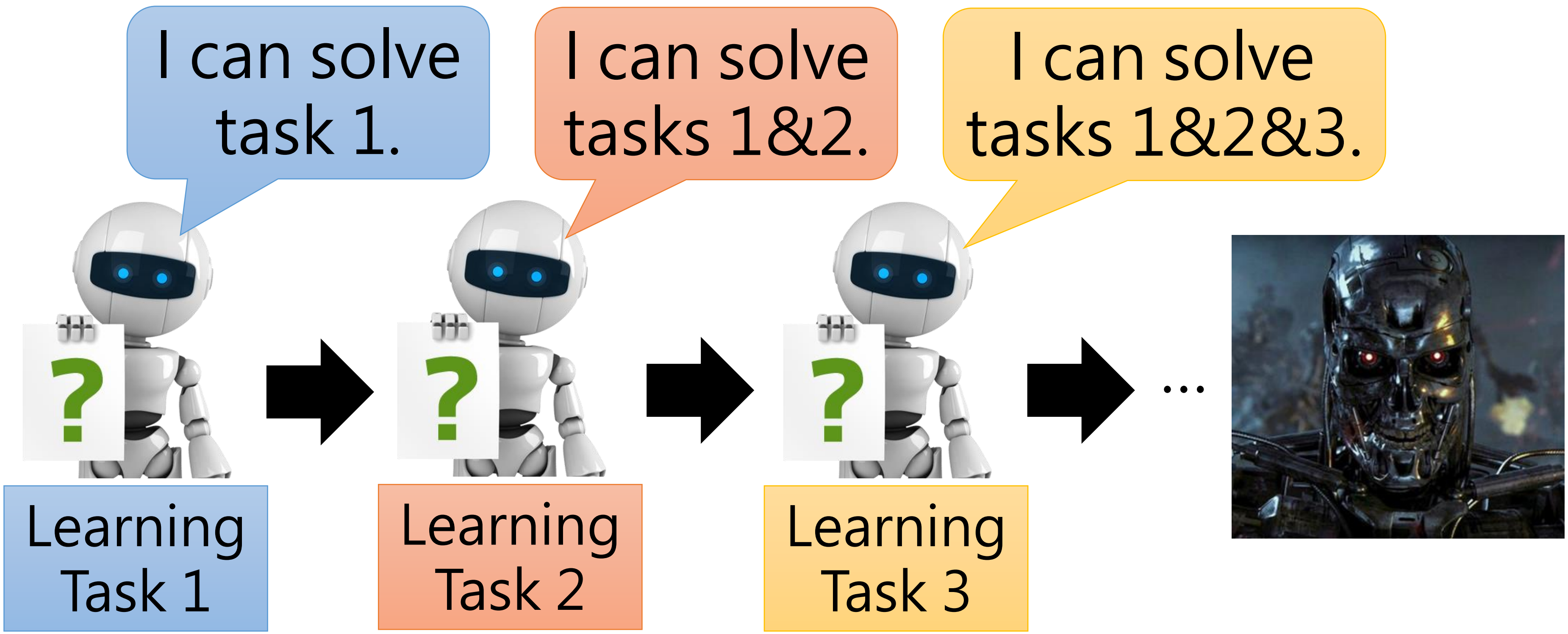
Hung-yi Lee
李宏毅

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

Life Long Learning (終身學習)

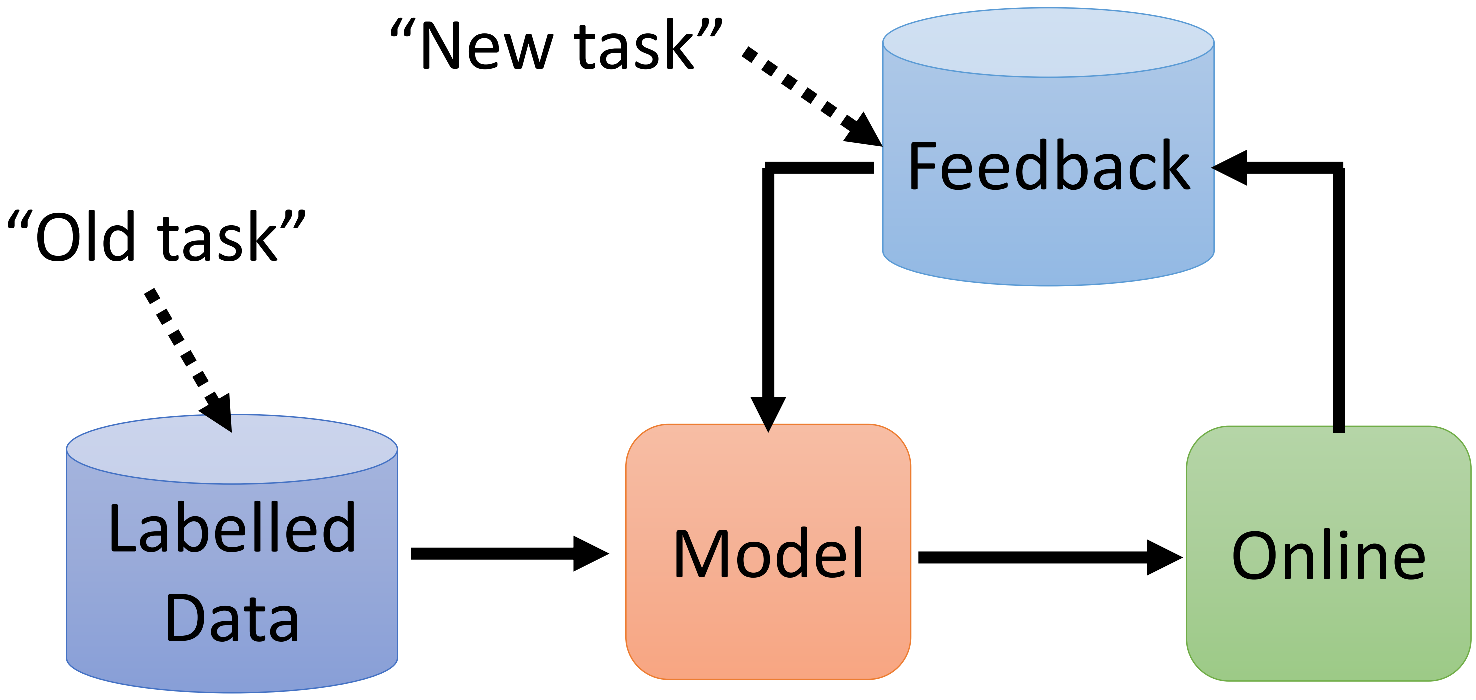
https://world.edu/lifelong-learning-part-time-undergraduate-provision-crisis/

What people think about AI …



Life Long Learning (LLL), Continuous Learning,
Never Ending Learning, Incremental Learning

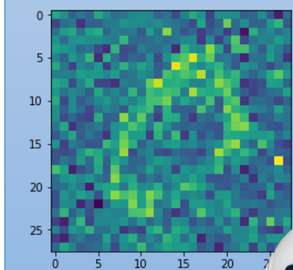
Life Long Learning
in real-world applications



Example

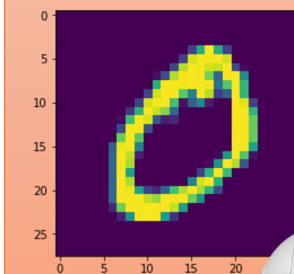
= 3 layers, 50 neurons each

Task 1

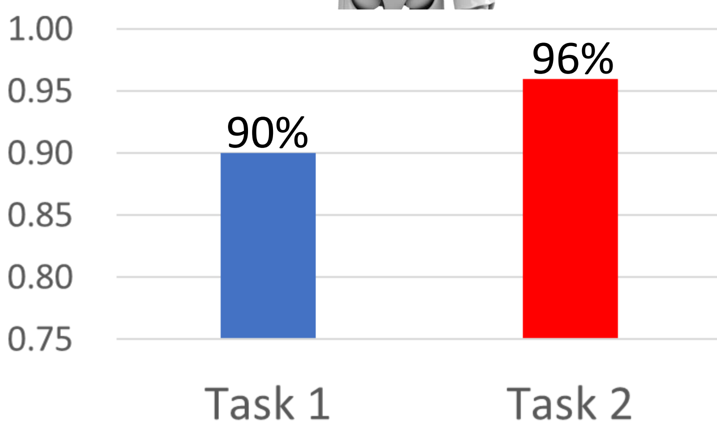


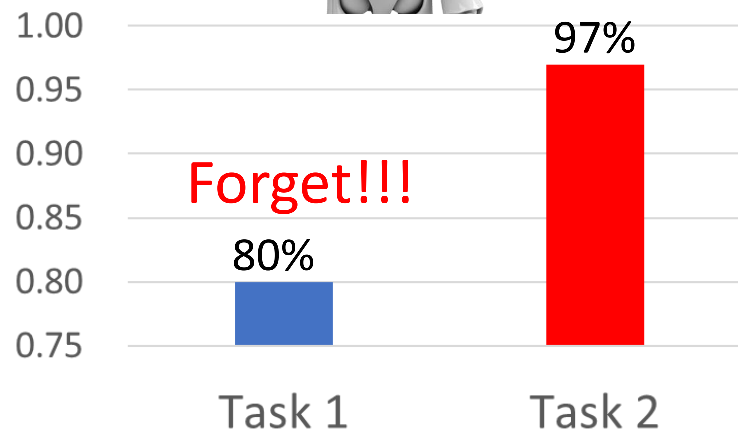
This is “0”.

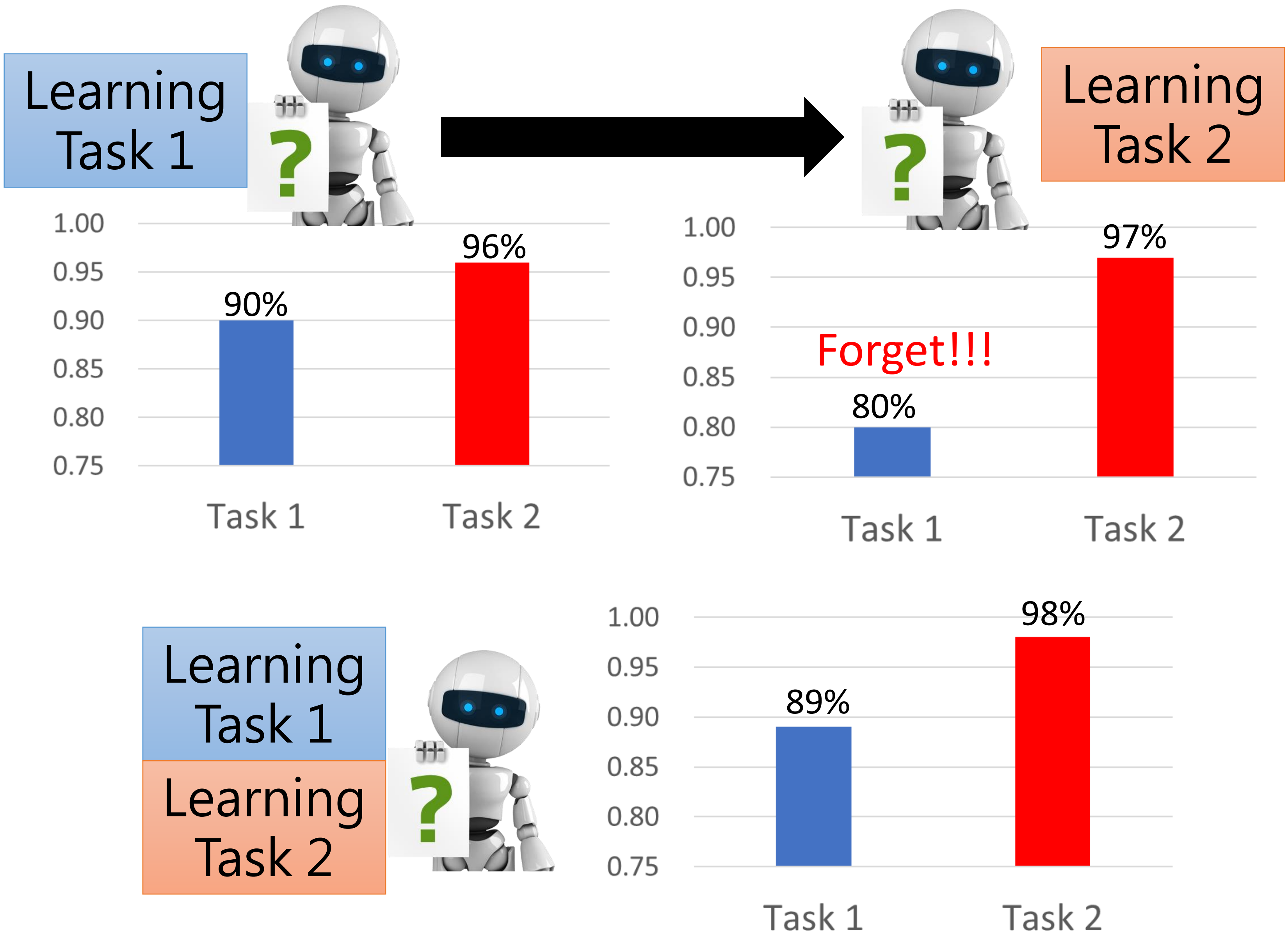
Task 2



This is “0”.







The network has enough capacity to learn both tasks.

Example

- QA: Given a document, answer the question based on the document.
- There are 20 QA tasks in bAbi corpus.

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 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

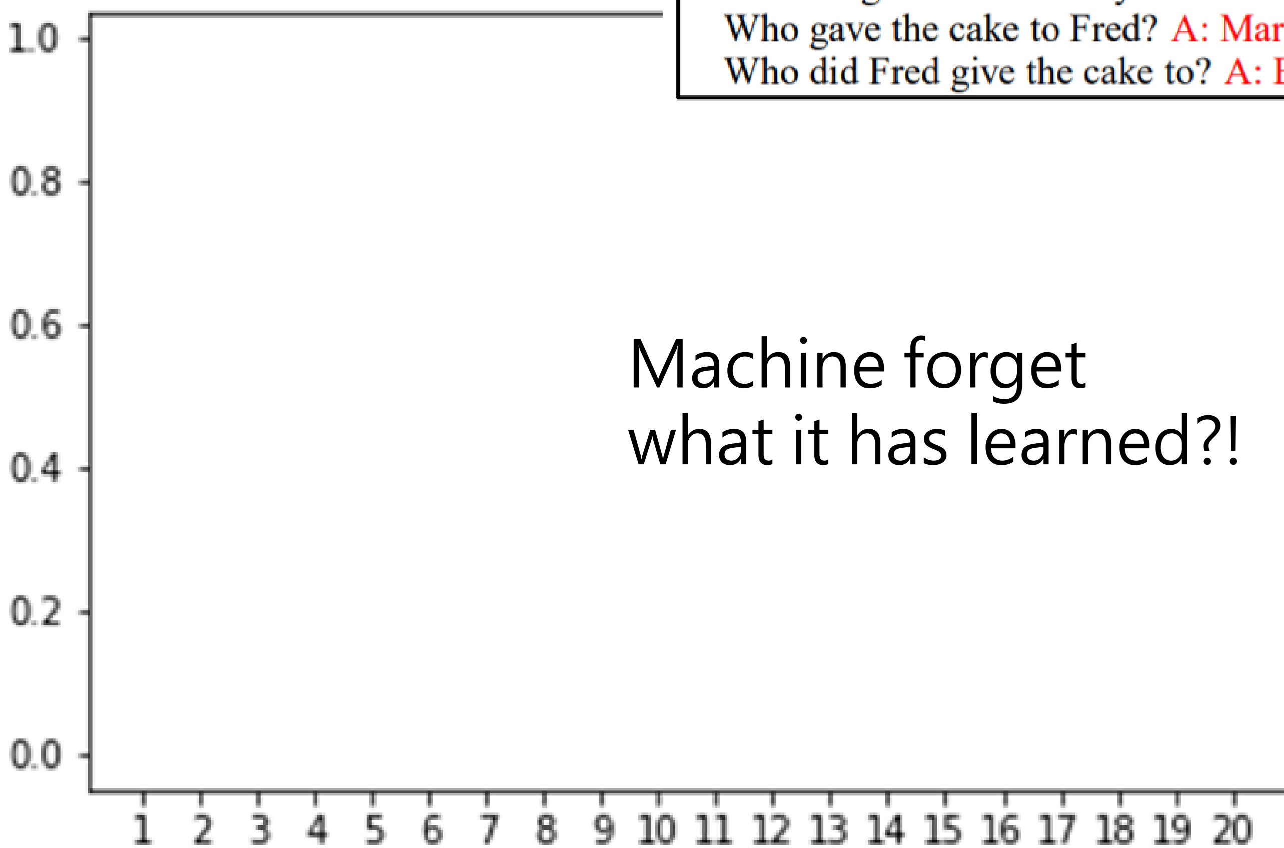
- Train a QA model through the 20 tasks

Example

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

Accuracy
Task 5



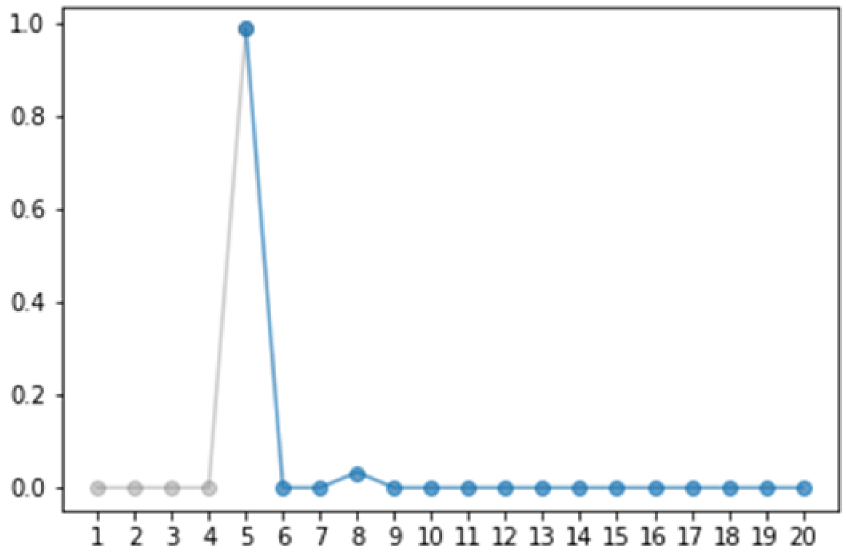
Machine forget
what it has learned?!

Machine learns 20 tasks sequentially

感謝何振豪同學提供實驗結果

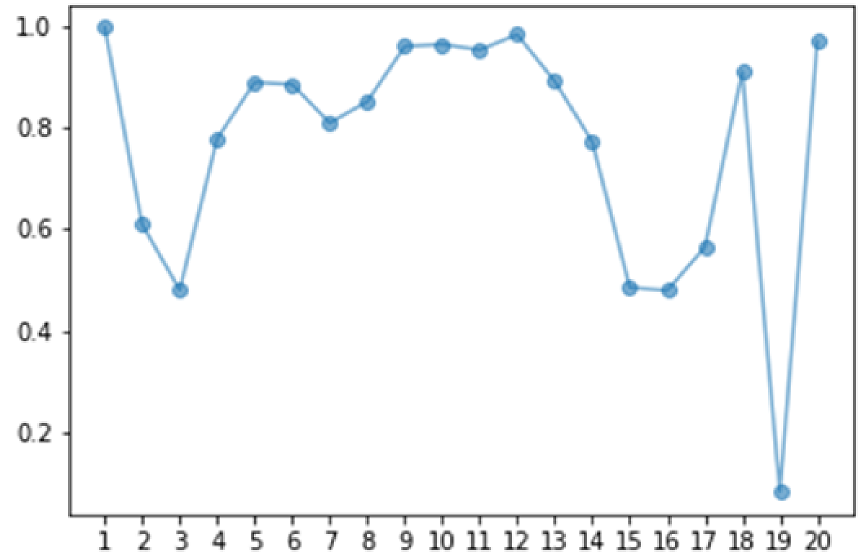
Example

Task 5 Accuracy



Learning 20 tasks sequentially

Accuracy of all 20 tasks



Learning 20 tasks simultaneously

Not because machine are not able to do it, but it just didn't do it.

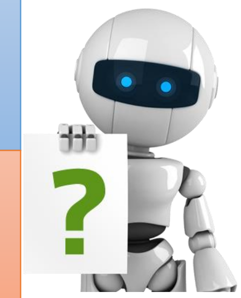
是不為也 非不能也

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**

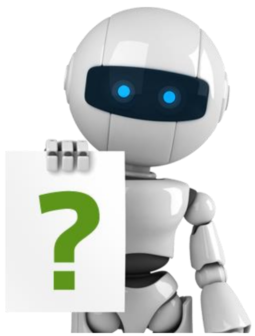
Training Data for Task 1 …… Training Data for Task 999 | Training Data for Task 1000

Always keep the data → **Storage issue**

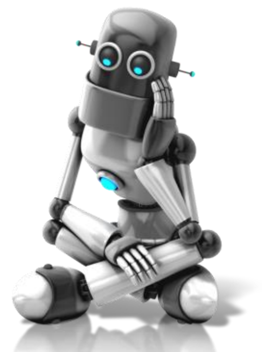
- Multi-task training can be considered as the upper bound of LLL.

Wait a minute ……

- Train a model for each task



Learning Task 1



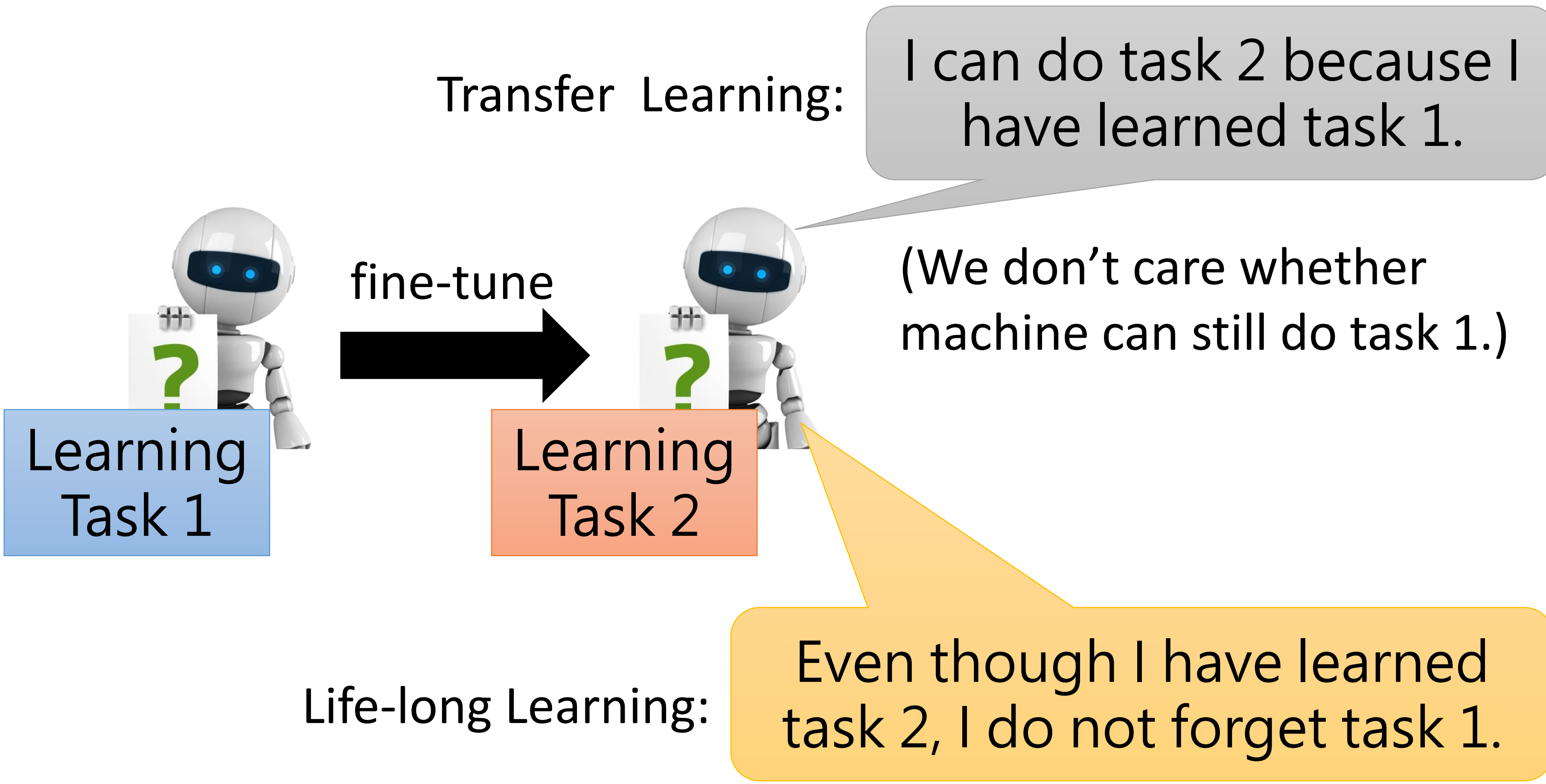
Learning Task 2

Learning Task 3

➢ Eventually we cannot store all the models …

➢ Knowledge cannot transfer across different tasks

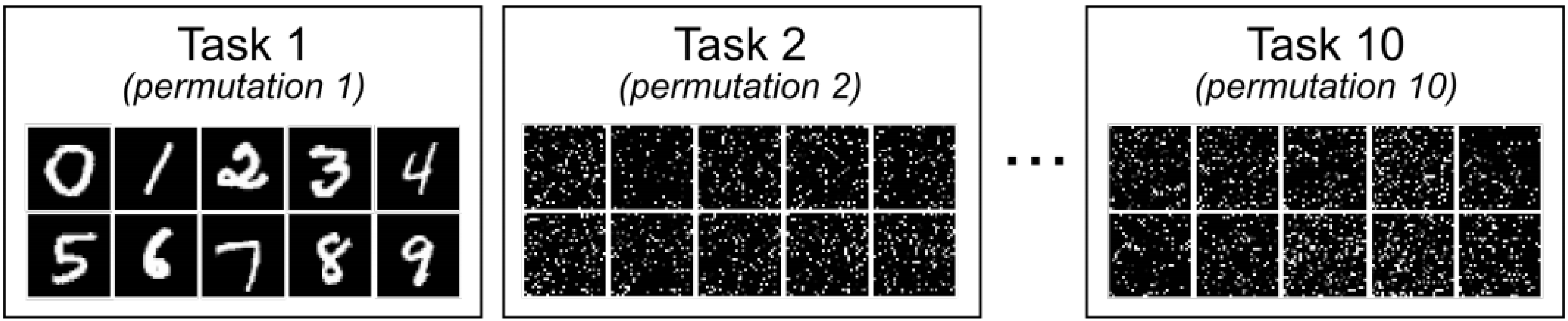
Life-Long v.s. Transfer

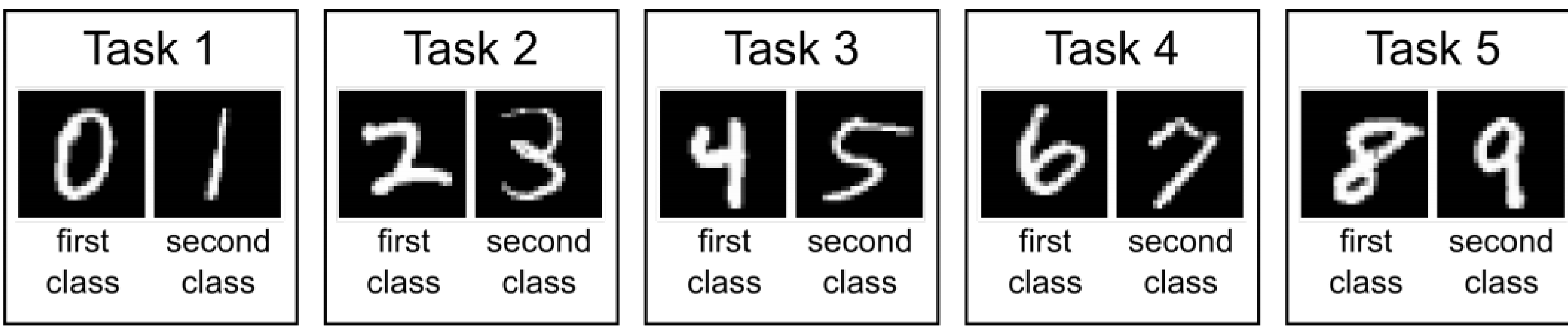


https://arxiv.org/pdf/1904.07734.pdf

Evaluation

First of all, we need a sequence of tasks.





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	$\boxed{R_{1,1}}$	$R_{1,2}$		$R_{1,T}$
	Task 2	$R_{2,1}$	$\boxed{R_{2,2}}$		$R_{2,T}$
	⋮				
	Task T-1	$R_{T-1,1}$	$R_{T-1,2}$		$R_{T-1,T}$
	Task T	$\boxed{R_{T,1}}$	$\boxed{R_{T,2}}$		$R_{T,T}$

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

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

(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	$R_{T,1}$	$R_{T,2}$		$R_{T,T}$

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

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

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

Research Directions

突觸的 可塑性

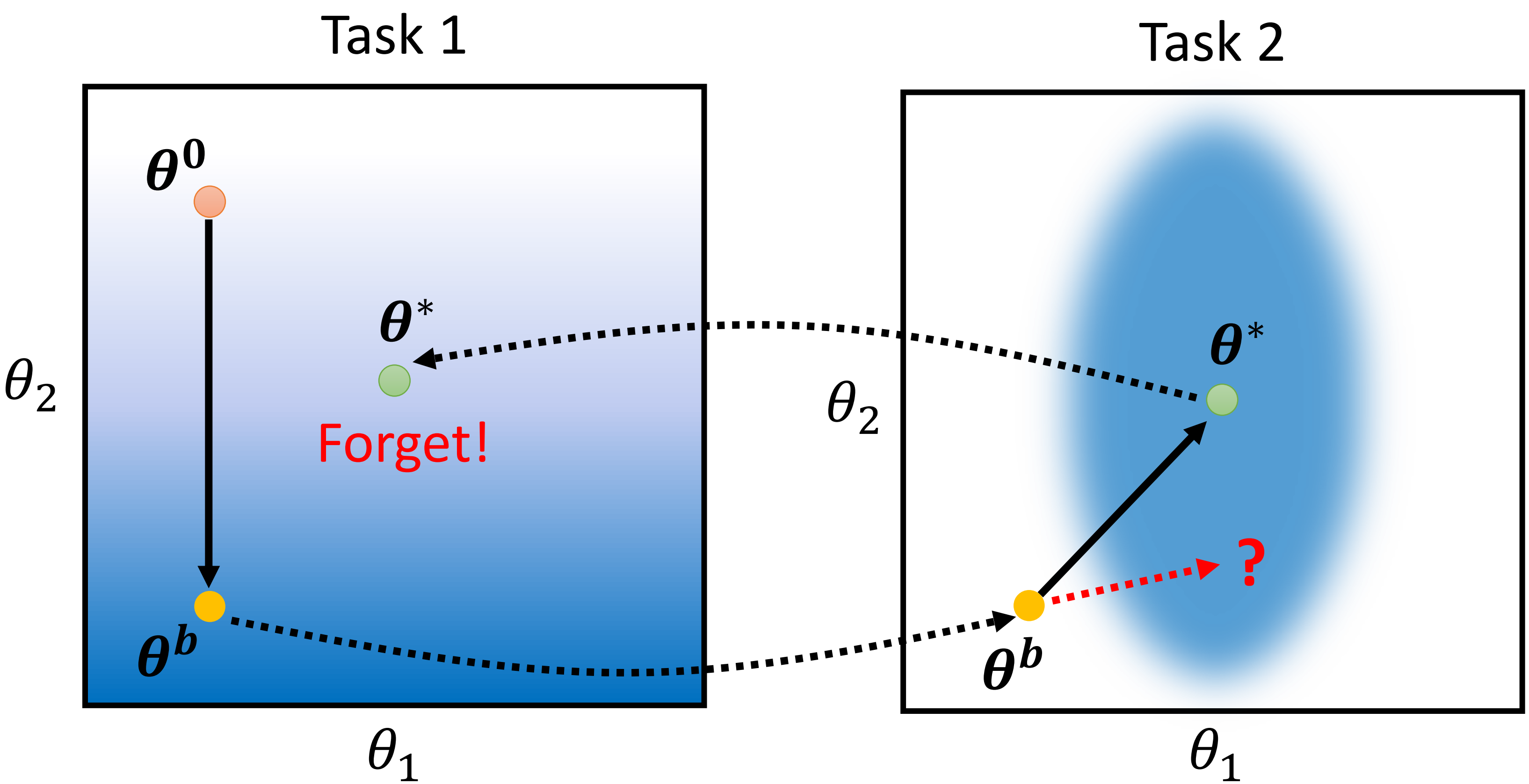
Regularization-based Approach

Selective Synaptic Plasticity

Additional Neural Resource Allocation

Memory Reply

Why Catastrophic Forgetting?



The error surfaces of tasks 1 & 2.

(darker = smaller loss)

Selective Synaptic Plasticity

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

$\boldsymbol{\theta}^{\boldsymbol{b}}$ is the model learned from the previous tasks.

Each parameter θ_i^b has a "guard" b_i

$$\underbrace{L'(\boldsymbol{\theta})}_{\text{Loss to be optimized}} = \overbrace{L(\boldsymbol{\theta})}^{\text{Loss for current task}} + \lambda \sum_i \overbrace{b_i}^{\text{How important this parameter is}} \big(\underbrace{\theta_i}_{\text{Parameters to be learning}} - \underbrace{\theta_i^b}_{\text{Parameters learned from previous task}}\big)^2$$

Selective Synaptic Plasticity

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

$\boldsymbol{\theta^b}$ is the model learned from the previous tasks.

Each parameter θ_i^b has a "guard" b_i

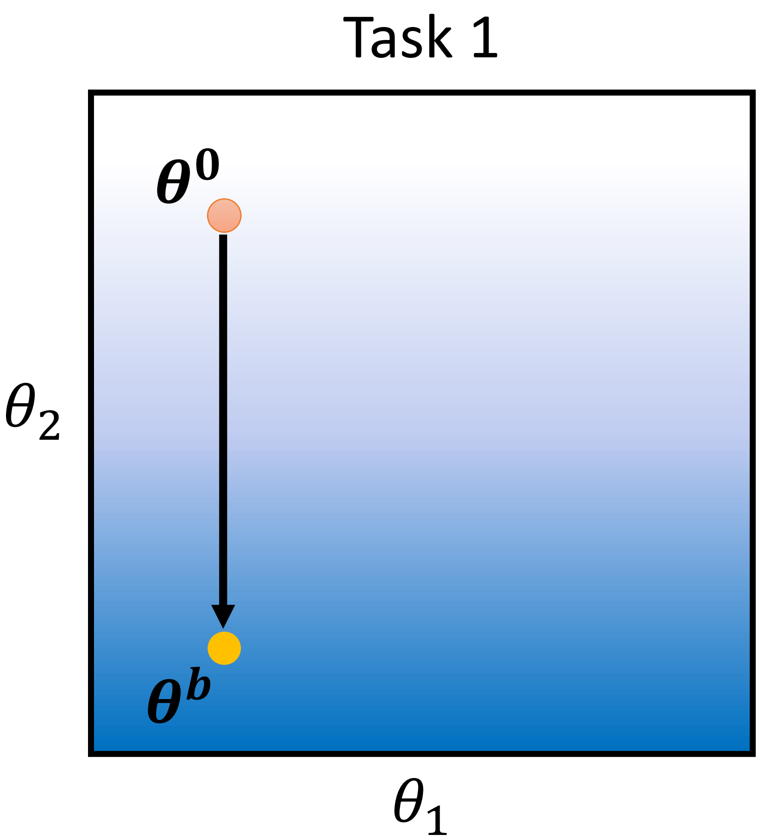
$\boldsymbol{\theta}$ should be close to $\boldsymbol{\theta^b}$ in certain directions.

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

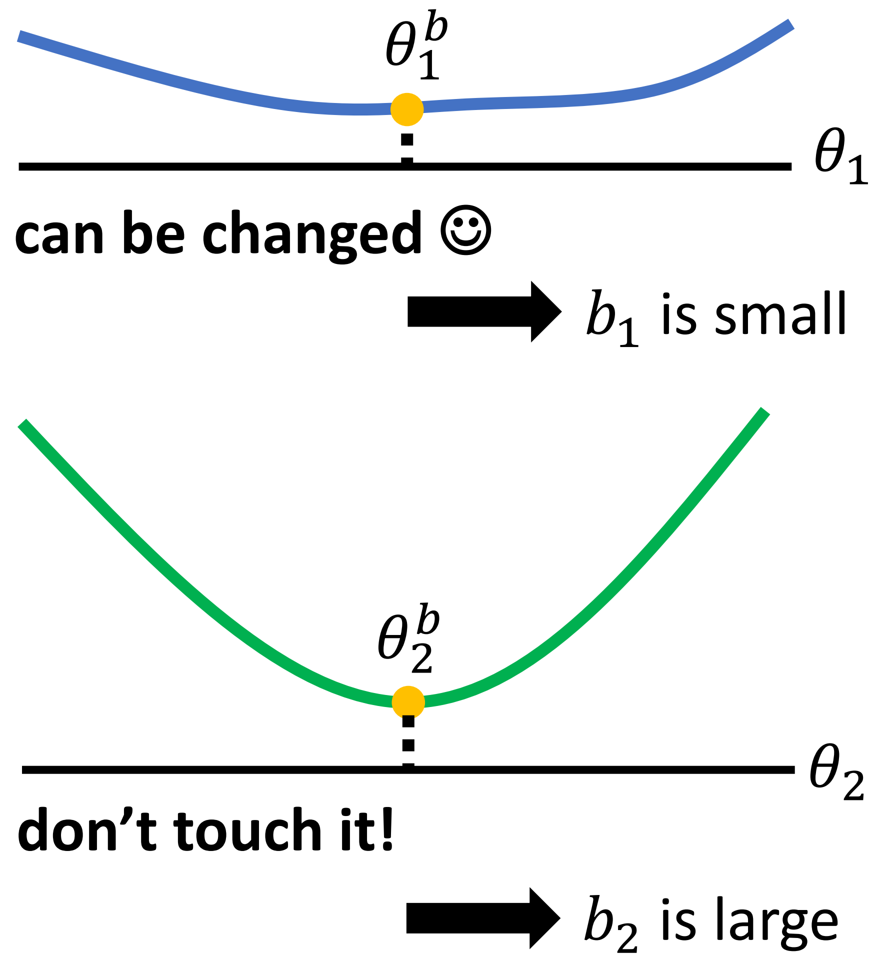
If $b_i = 0$, there is no constraint on θ_i ➡ Catastrophic Forgetting

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

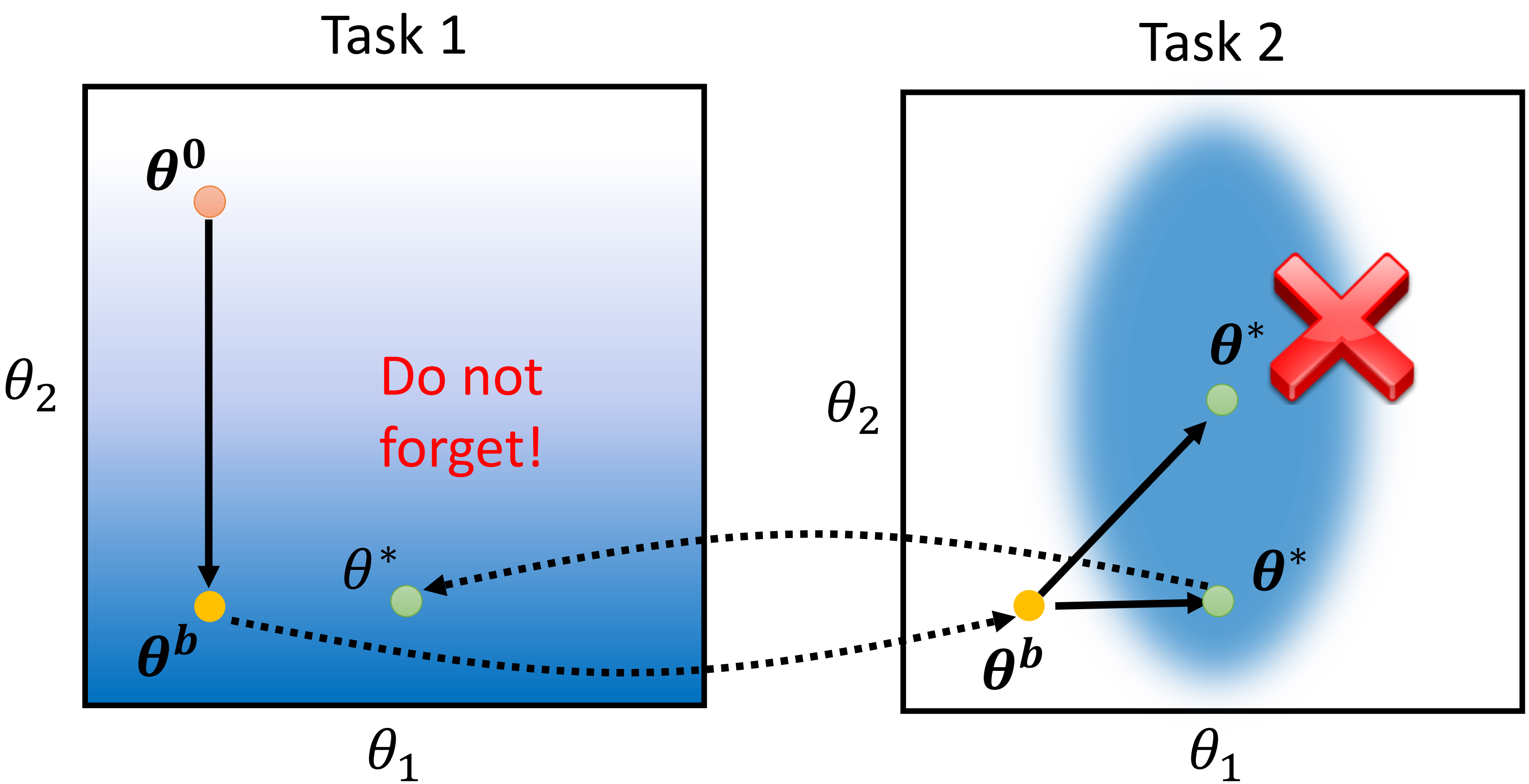
Selective Synaptic Plasticity



Each parameter has a "guard" b_i



Selective Synaptic Plasticity

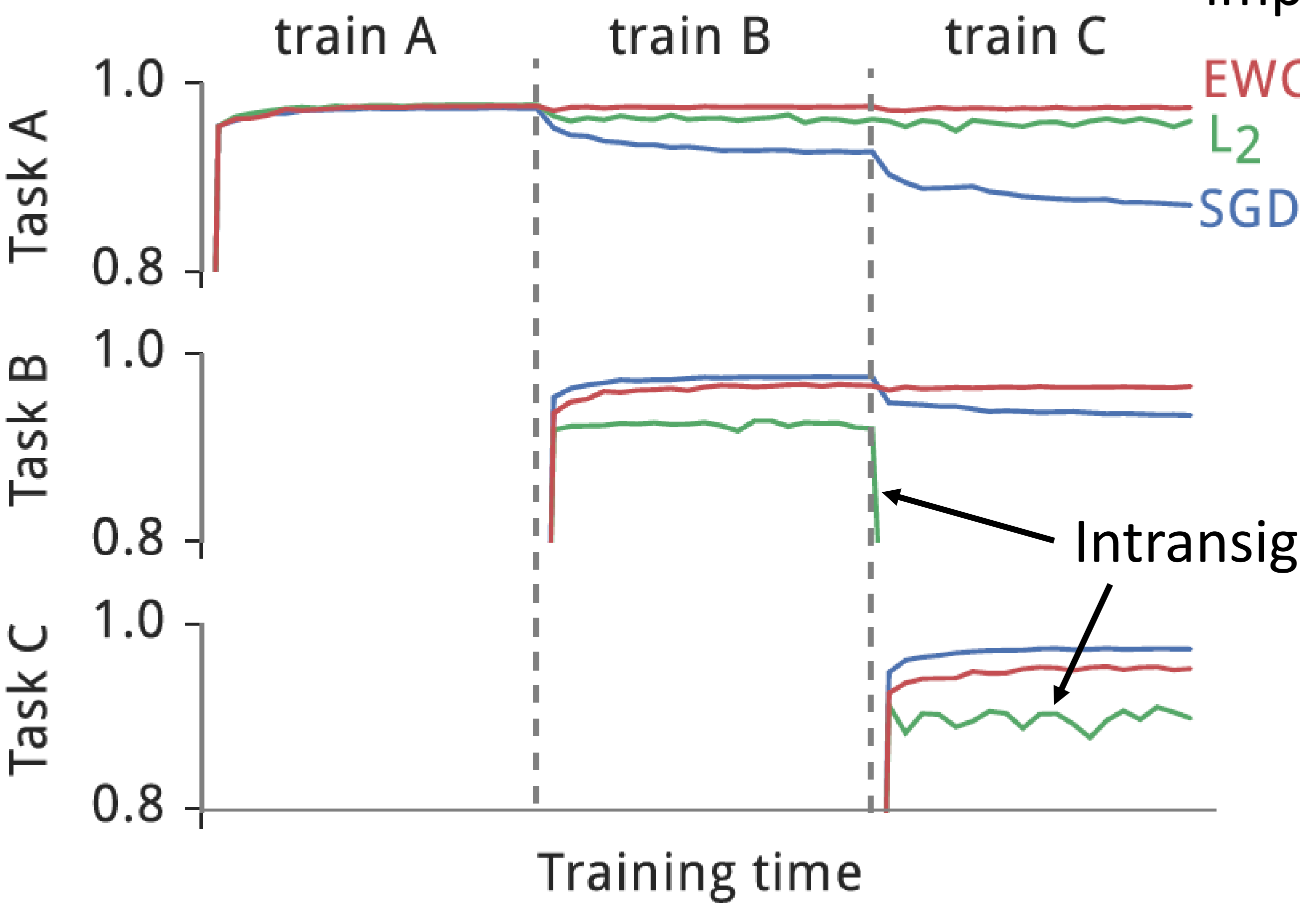


b_1 is small, while b_2 is large.

(We can modify θ_1, but do not change θ_2.)

Selective Synaptic Plasticity

b_i represents importance



$b_i = 1$

$b_i = 0$

Intransigence

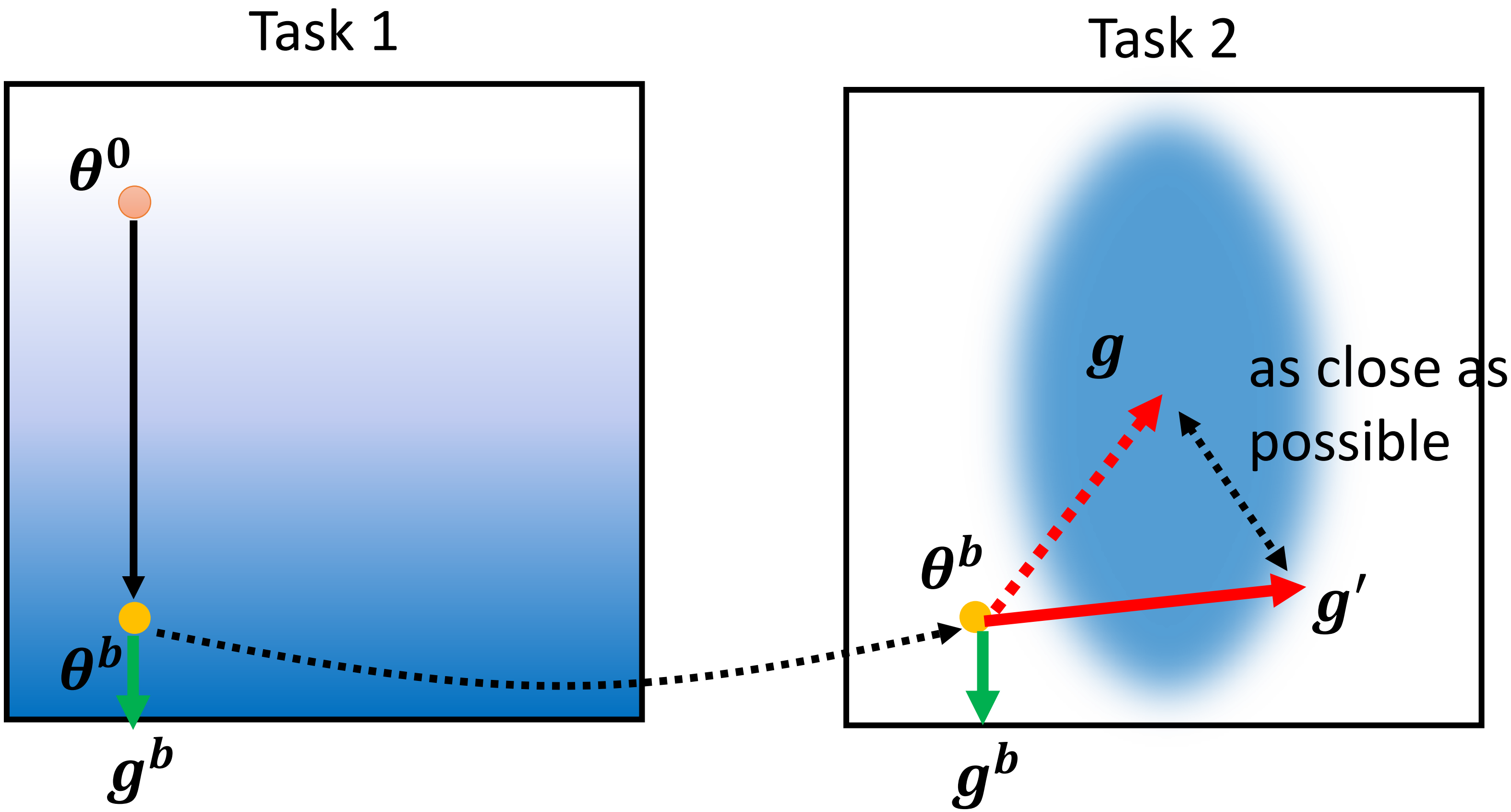
MNIST permutation, from the original EWC paper

Selective Synaptic Plasticity

- Elastic Weight Consolidation (EWC)
    - https://arxiv.org/abs/1612.00796
- Synaptic Intelligence (SI)
    - https://arxiv.org/abs/1703.04200
- Memory Aware Synapses (MAS)
    - https://arxiv.org/abs/1711.09601
- RWalk
    - https://arxiv.org/abs/1801.10112
- Sliced Cramer Preservation (SCP)
    - https://openreview.net/forum?id=BJge3TNKwH

Gradient Episodic Memory (GEM)

https://arxiv.org/abs/1706.08840



$$\boldsymbol{g}' \cdot \boldsymbol{g}^b \geq 0$$

: negative gradient of current task

: negative gradient of previous task

: update direction

Need the data from the previous tasks

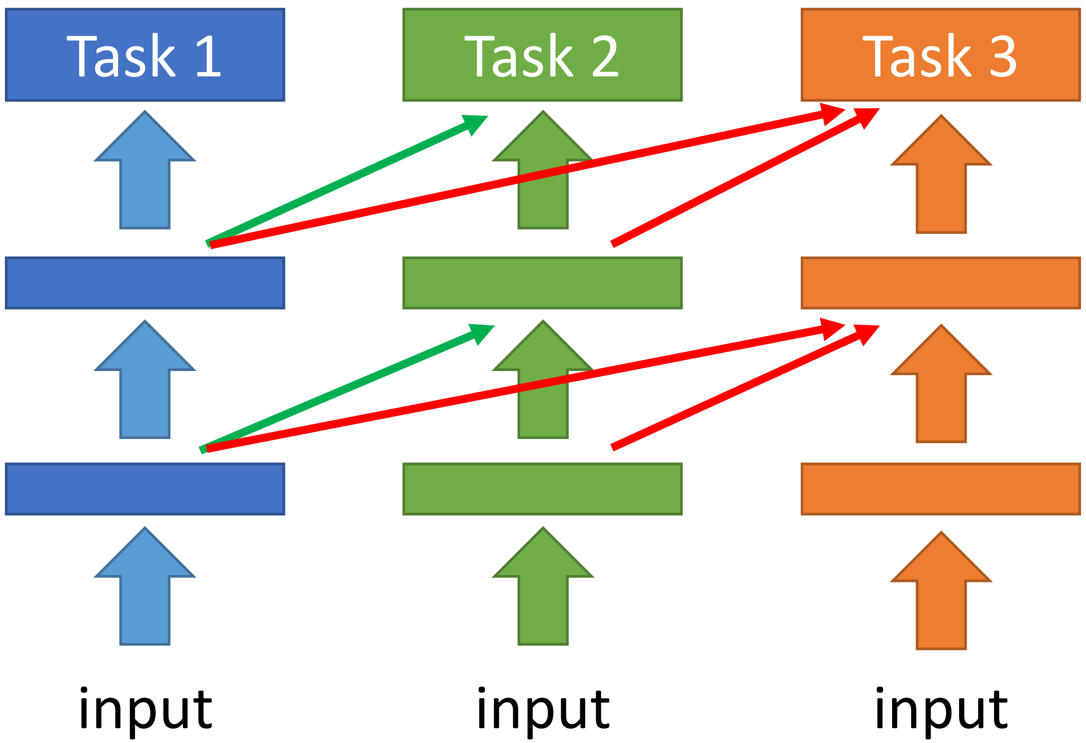
Research Directions

Selective Synaptic Plasticity

Additional Neural Resource Allocation

Memory Reply

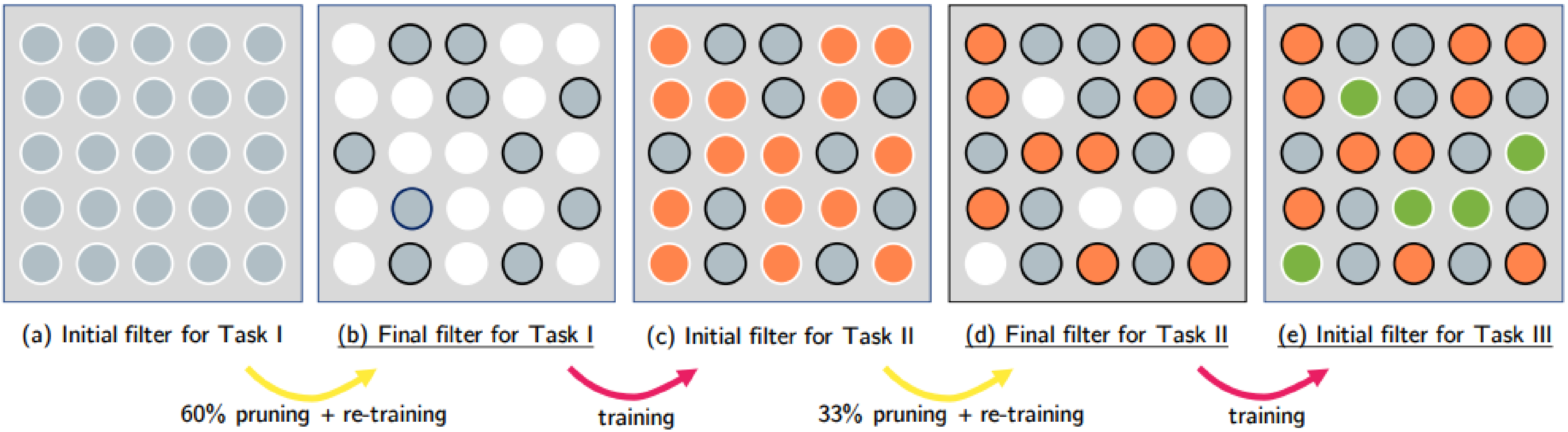
Progressive Neural Networks



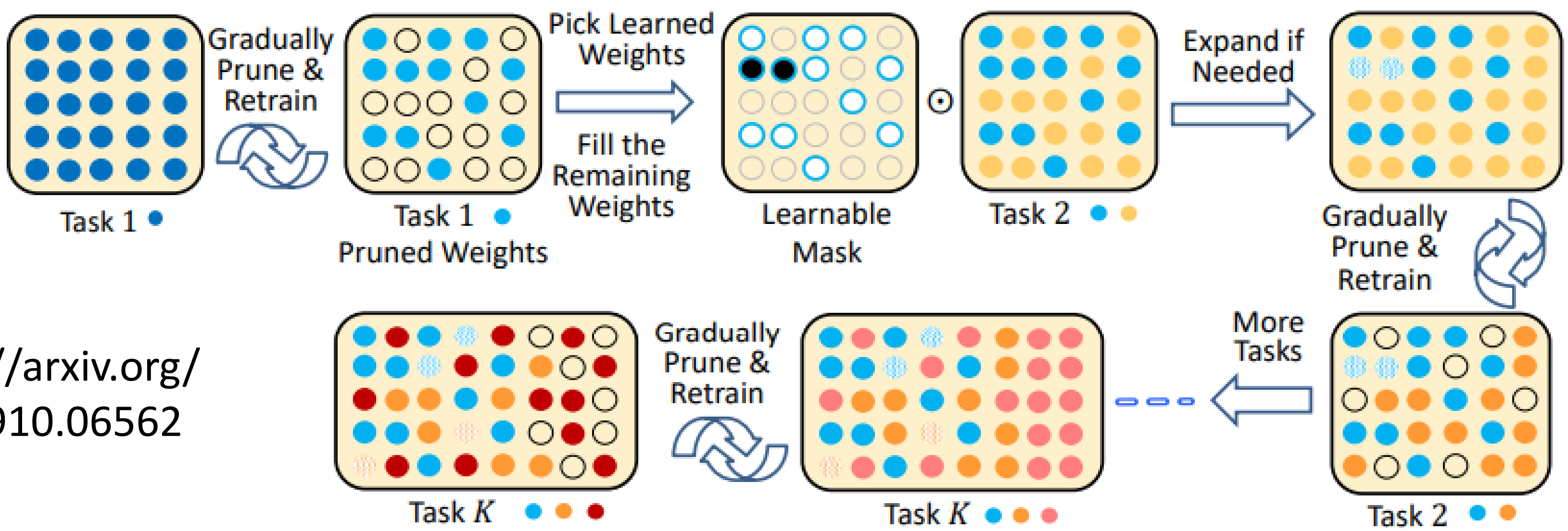
https://arxiv.org/abs/1606.04671

PackNet

https://arxiv.org/abs/1711.05769



Compacting, Picking, and Growing (CPG)



https://arxiv.org/abs/1910.06562

Research Directions

Selective Synaptic Plasticity

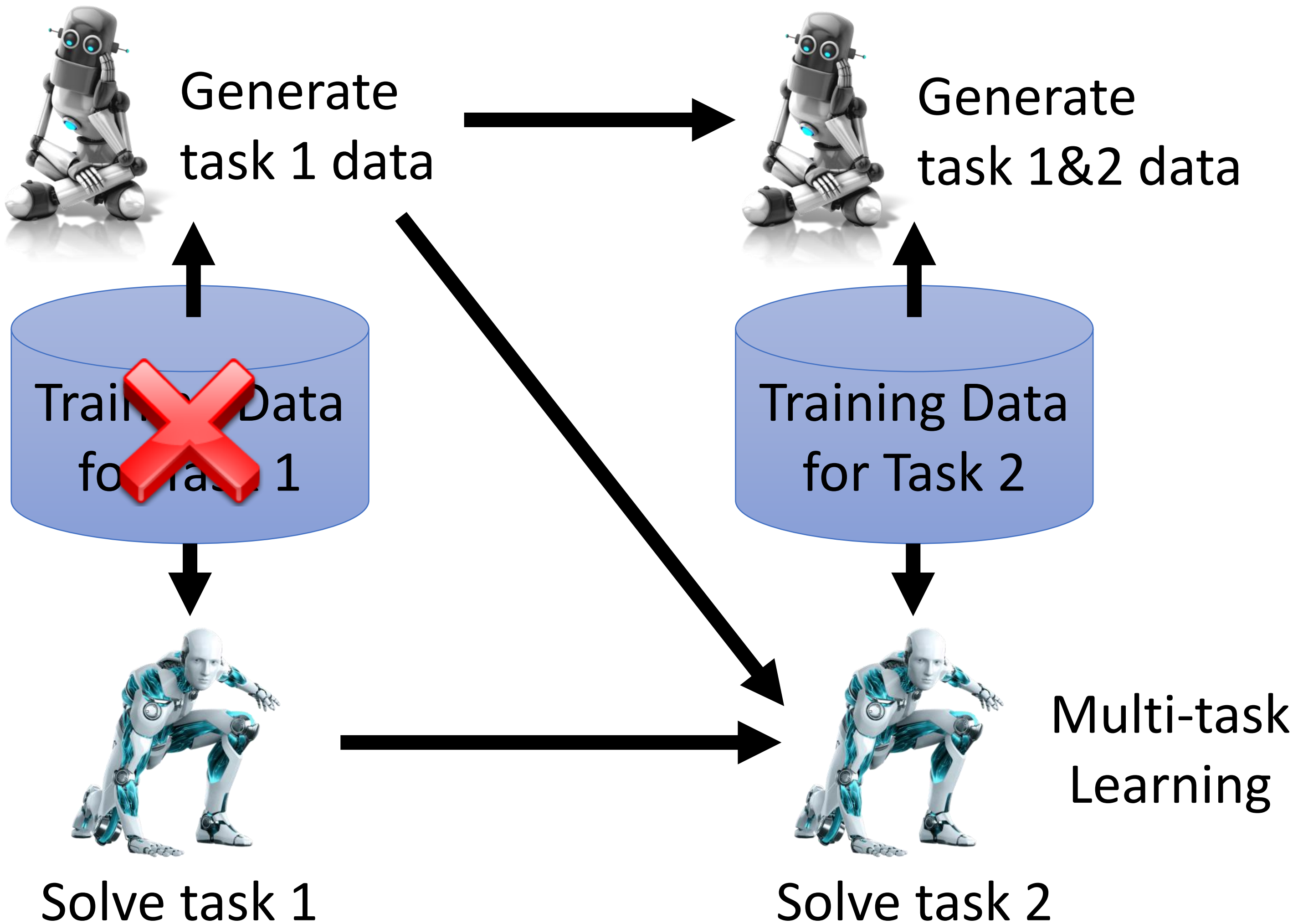
Additional Neural Resource Allocation

Memory Reply

Generating Data

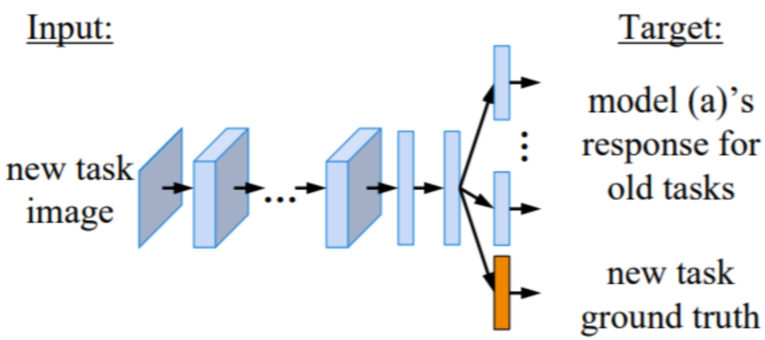
https://arxiv.org/abs/1705.08690
https://arxiv.org/abs/1711.10563
https://arxiv.org/abs/1909.03329

- Generating pseudo-data using generative model for previous tasks

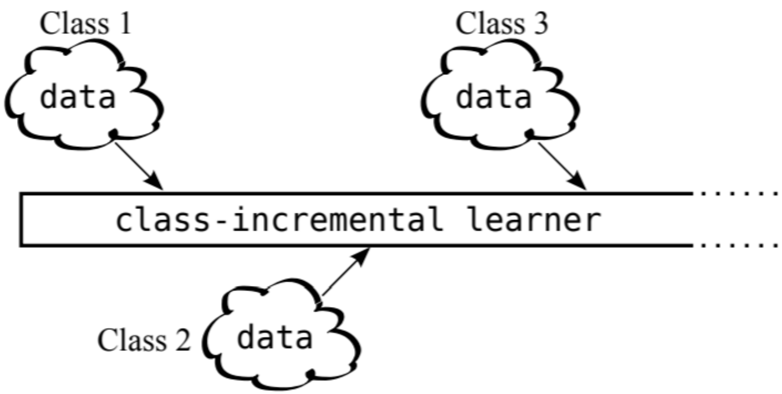


Adding new classes

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



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



Three scenarios for continual learning

https://arxiv.org/abs/1904.07734

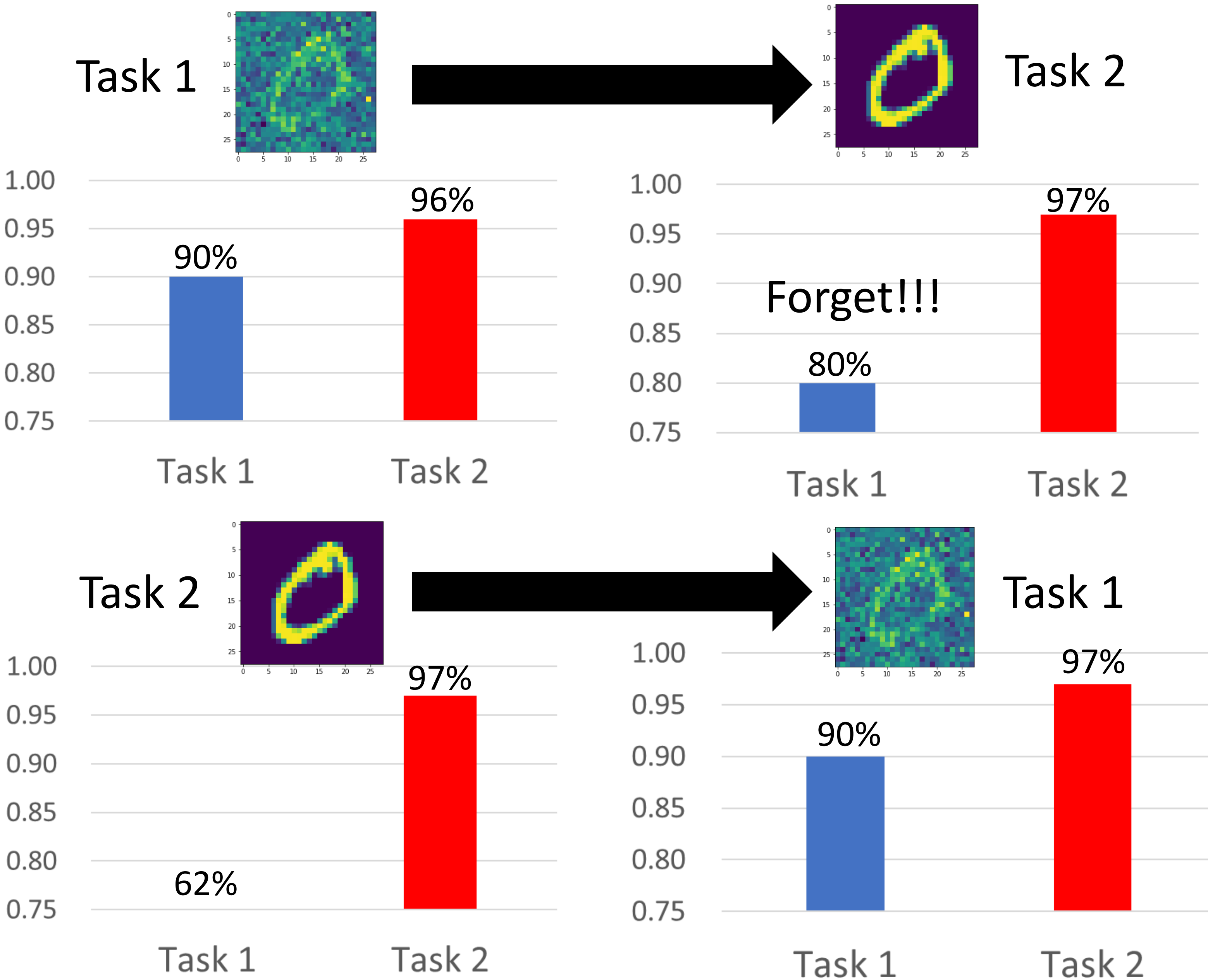
Concluding Remarks

Memory Reply

Additional Neural Resource Allocation

Selective Synaptic Plasticity

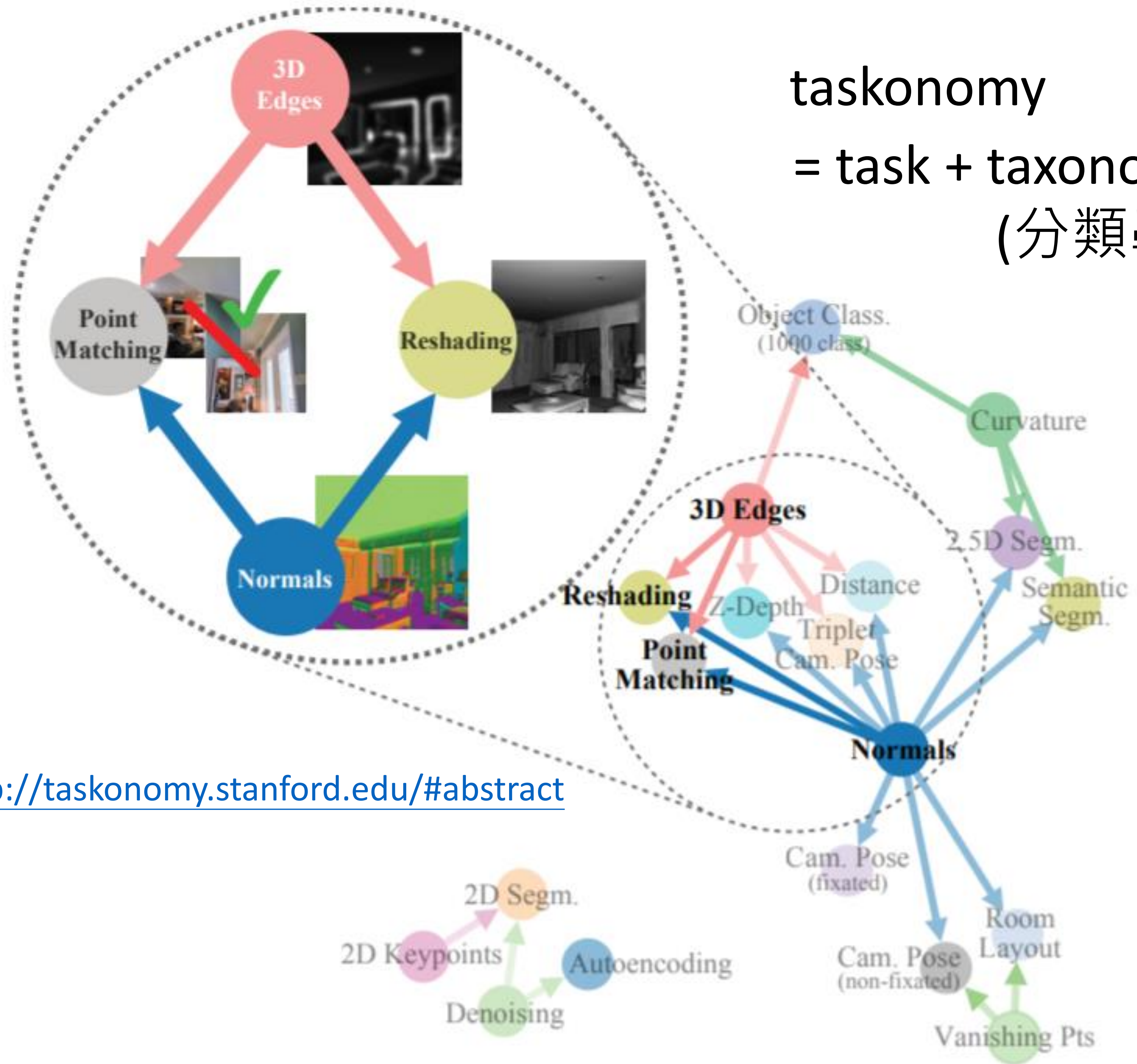
Curriculum Learning : what is the proper learning order?



taskonomy

= task + taxonomy

(分類學)



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