

终身机器学习

Lifelong Machine Learning

3 类增量学习

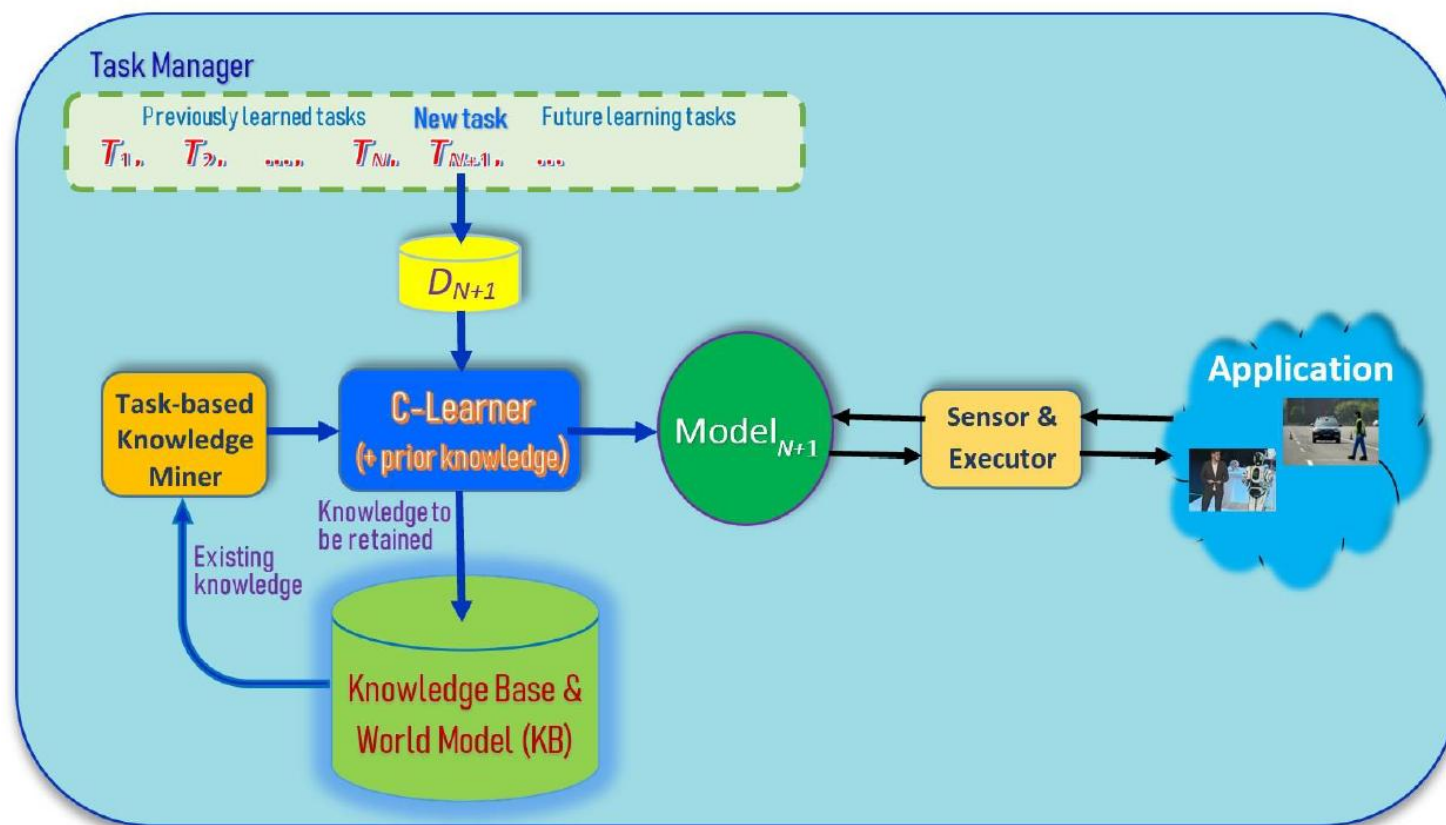
主讲：梁国强
gqliang@nwpu.edu.cn

3.1 持续学习

3.2 基于正则化的类增量学习

3.3 基于回放的类增量学习

3.4 基于模型结构的类增量学习



终身学习的关键特征

- 持续学习过程
- 明确的知识积累和保存
- 使用已学知识帮助学习
- 发现新任务能力
- 边工作边学习能力

以分类任务为例，介绍终身学习

又称持续学习

3.1 持续学习

数学描述

逐步学习一系列任务 $1, \dots, T$, 其中第 $k \in T$ 个任务的训练数据集为

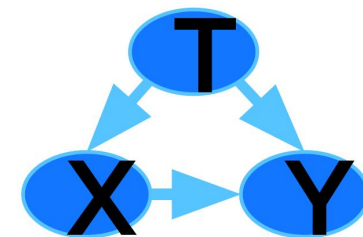
$$D_k = \left\{ \left((x_k^i, k), y_k^i \right)_{i=1}^{n_k} \right\}$$

n_k : 第 k 个任务的样本数量

$x_k^i \in X$: 输入样本; $y_k^i \in Y_k \in Y$ 对应标签, 且 $Y_k \cap Y_{k+1} = \emptyset$

目标: 学习映射 $f: X \times T \rightarrow Y$

测试: 在所有任务测试集上测试



Task ID; Task Boundary

3.1 持续学习

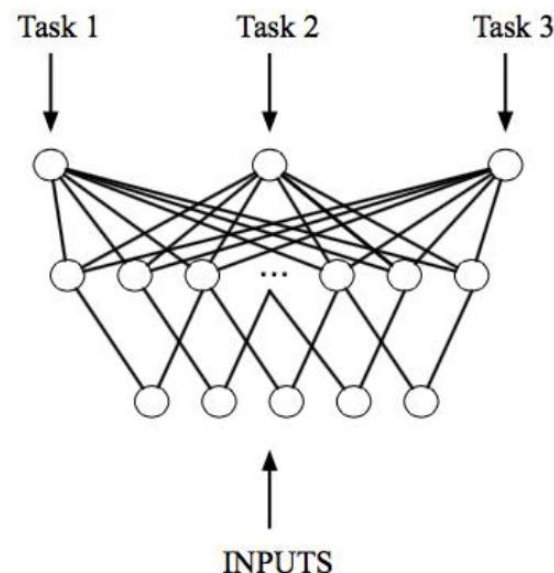
任务增量学习Task-Incremental CL

- 训练和测试阶段，模型已知Task ID
- 假设 $Y_k \cap Y_{k+1} = \emptyset$
- 可以为每个任务训练一个模块

最简单

典型的神经网络结构

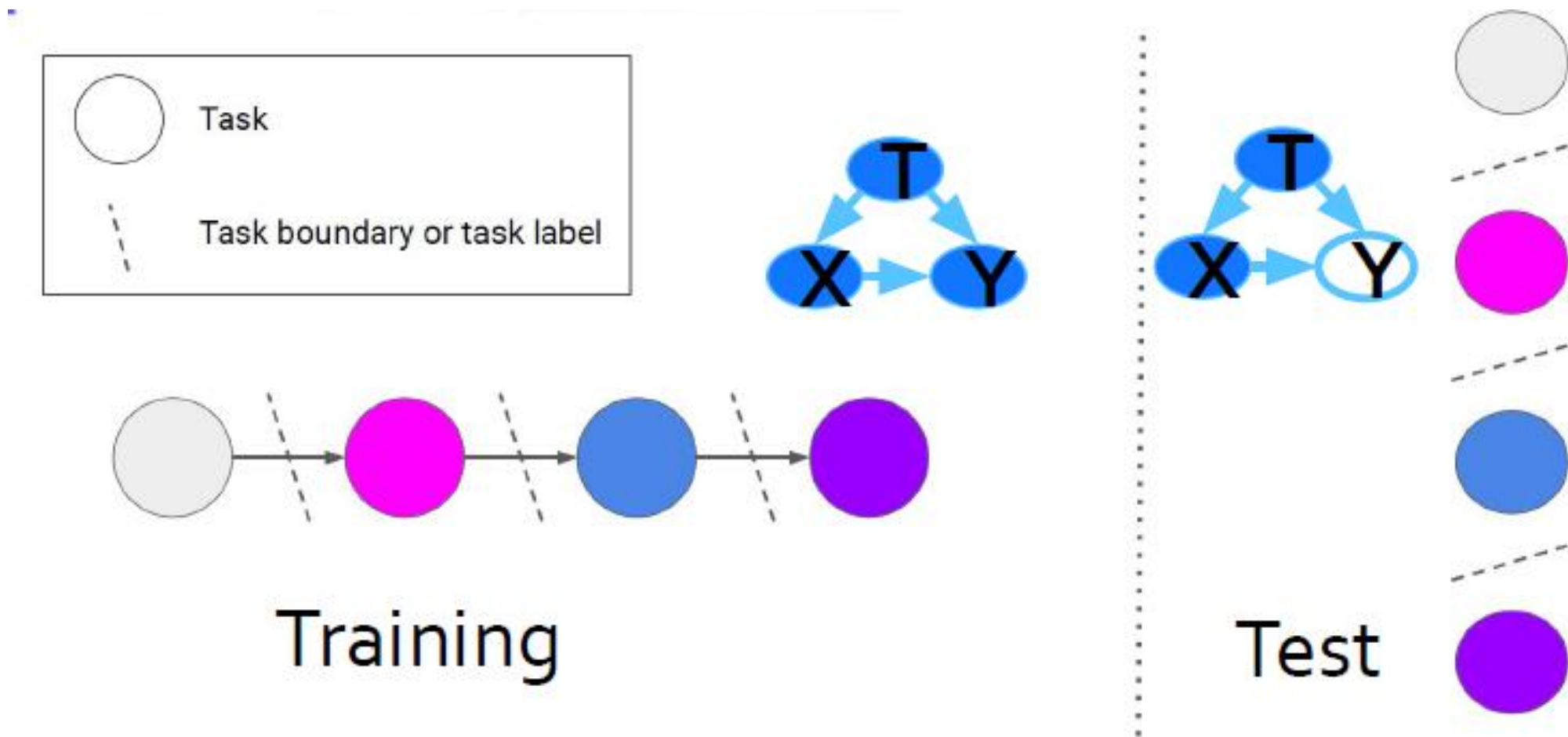
- 多头输出网络,
- 输出任务独立,
- 其它单元共享



3.1 持续学习



任务增量学习Task-Incremental CL



3.1 持续学习

类增量学习Class-Incremental CL

- 训练时，模型已知Task ID；测试阶段，Task ID未知
- 测试时，模型需要额外推断其Task ID，解决所有任务
- 典型场景：序列学习新类别
- 假设

$$Y_k \cap Y_{k+1} = \emptyset$$

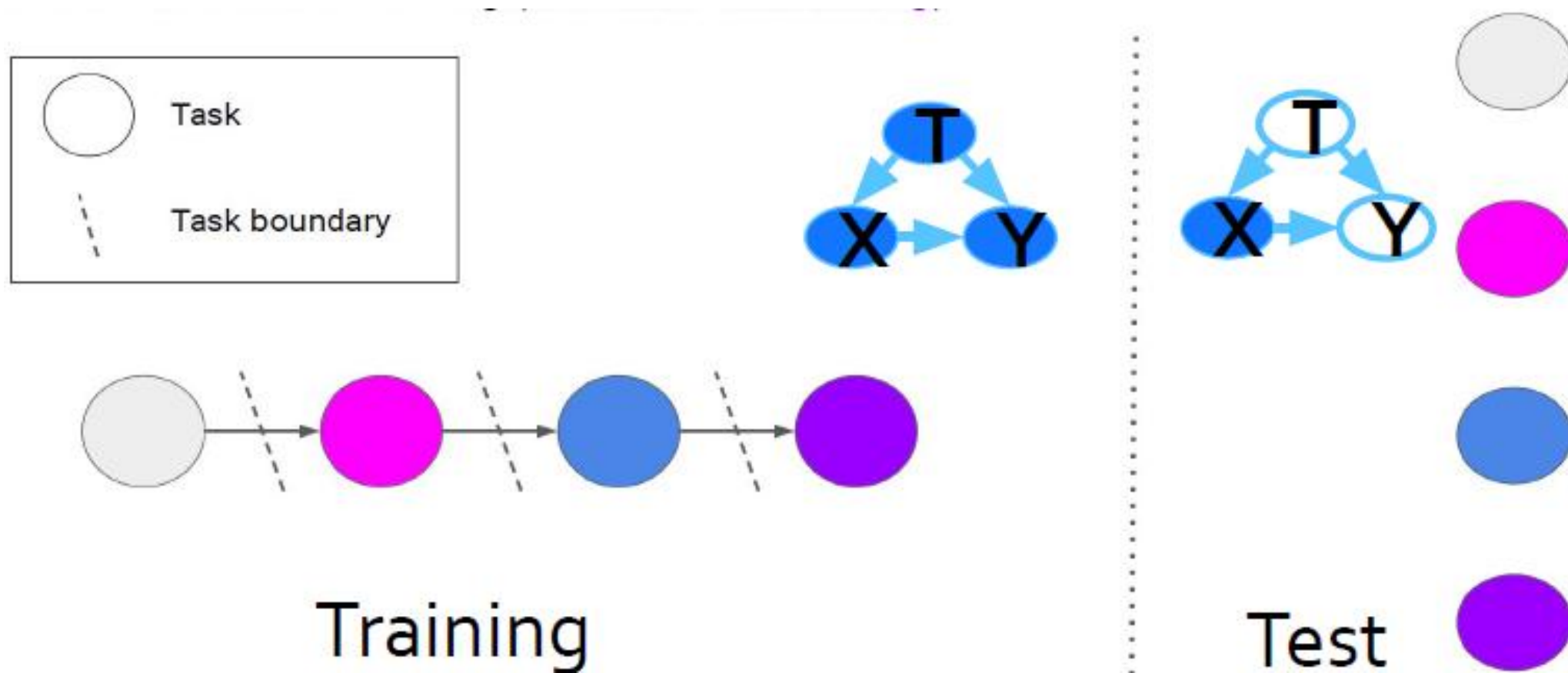
$$P(X_k) \neq P(X_{k+1})$$

$$P(Y_k) \neq P(Y_{k+1})$$

3.1 持续学习



类增量学习Class-Incremental CL



3.1 持续学习

类增量学习Class-Incremental CL: 两个典型子类

■ Batch CL

- 在一个任务学习阶段，其训练数据一直可见
- 可以训练任意轮次

■ Online CL

- 训练数据以流式方式到达
- 当汇聚到一小批数据时，迭代进行一次训练
- 只能执行一次训练epoch

存储资源有限
不能存储全部任务数据

3.1 持续学习

域增量学习Domain-Incremental CL

- 训练时，模型已知Task ID；测试阶段，Task ID未知
- 测试时，模型只需要解决任务，不需要推断Task ID
- 典型场景：任务结构一致，但是输入发生变化

$$Y_k = Y_{k+1}$$

$$P(X_k) \neq P(X_{k+1})$$

$$P(Y_k) = P(Y_{k+1})$$

域增量

$$Y_k \cap Y_{k+1} = \emptyset$$

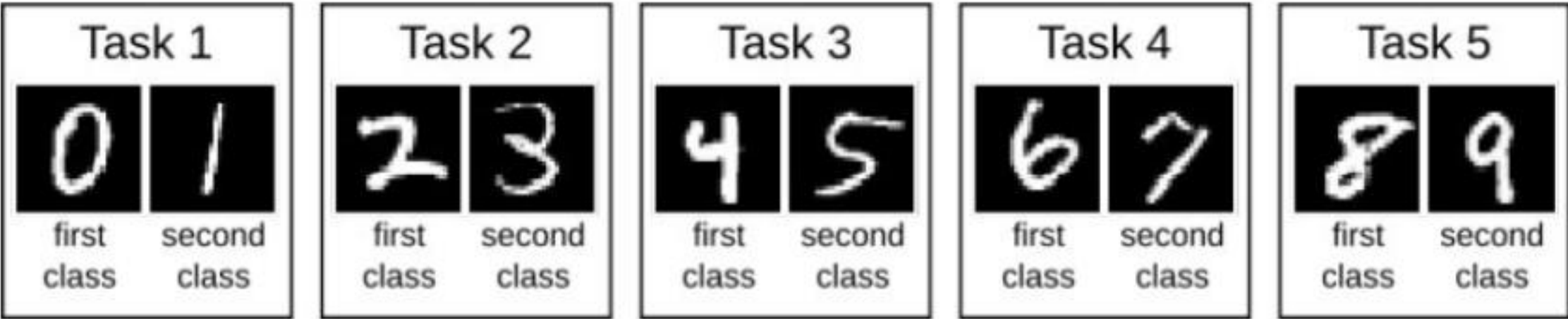
$$P(X_k) \neq P(X_{k+1})$$

$$P(Y_k) \neq P(Y_{k+1})$$

类增量

3.1 持续学习

例1：Split MNIST



不同持续学习下，测试阶段设定程序

类型	测试设定
任务增量	任务给定，判断其是第一类还是第二类（eg, 0 or 1）
域增量	任务未知，判断其是第一类还是第二类（eg, in [0,2,4,6,8] or [1,3,5,7,9]）
类增量	任务未知，判断其数字(eg, 属于0到9的哪一个)

任务不可知Task-Agnostic CL

- 训练和测试阶段，Task ID均未知



3.1 持续学习

对比

■ 任务增量持续学习

- 为每个任务单独训练模型
- 测试时，任务ID已知

■ 域增量持续学习

- 所有任务具有相同的类
- 测试时，任务ID未知

■ 类增量持续学习

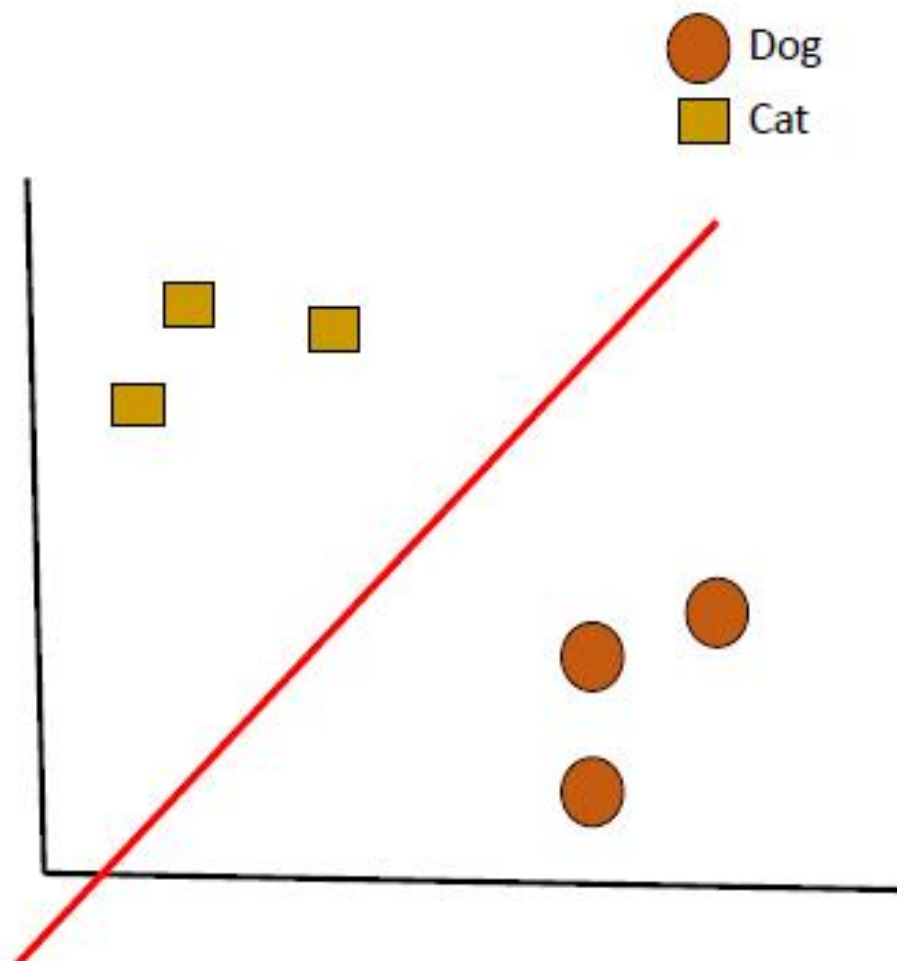
- 为所有模型训练一个模型
- 测试时，ID未知

3.1 持续学习

例子

- 2维特征
- 自由度为2

对于任务1，很容易获得
一个高性能的分类器



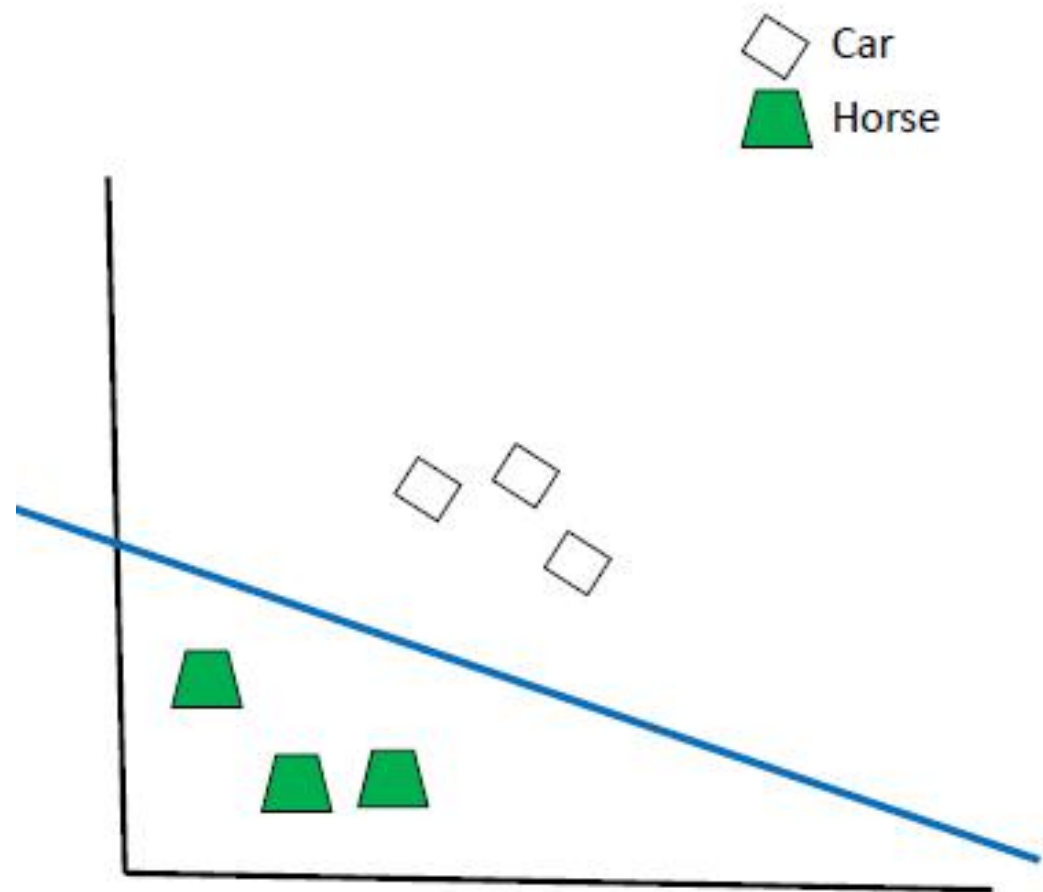
3.1 持续学习

例子

- 2维特征
- 自由度为2

新任务2到来

- 可以单独训练一个高性能的分类器
- 任务1和2的模型不同，即模型参数发生变化



3.1 持续学习

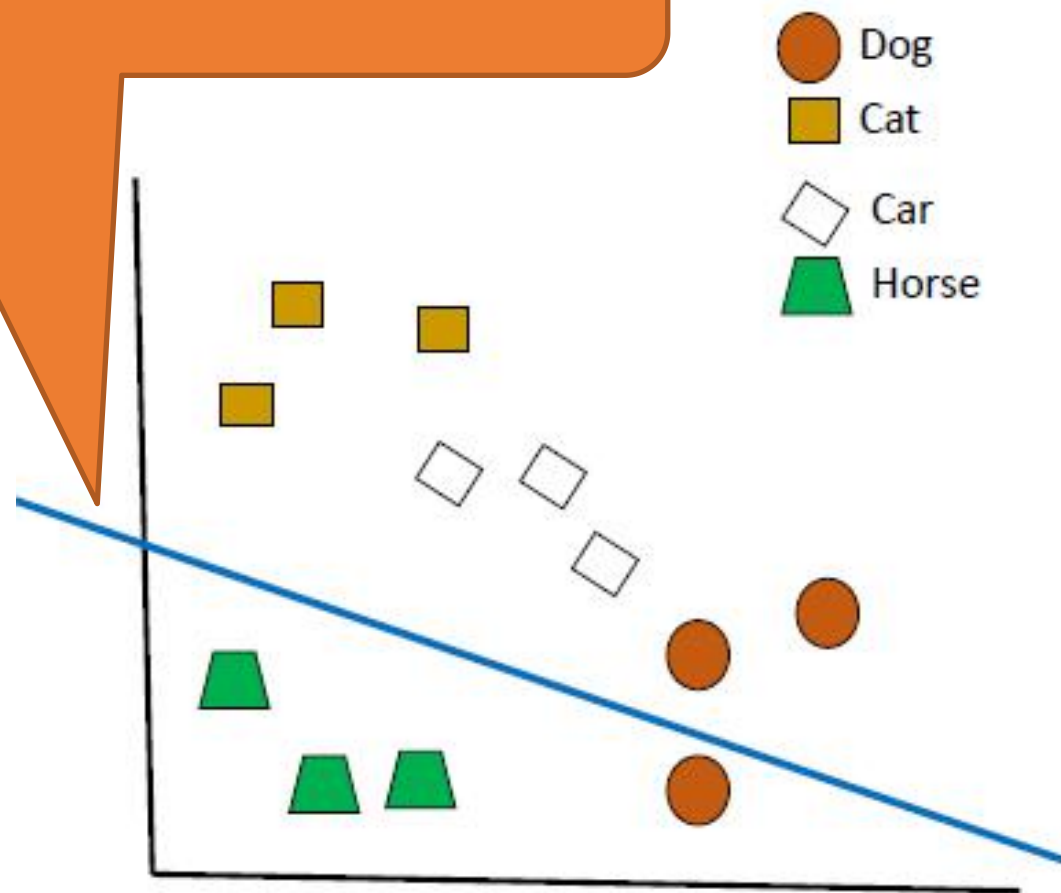
例子

- 2维特征
- 自由度为2

新任务2到来

- 可以单独训练一个高性能的分类器
- 任务1和2的模型不同，即模型参数发生变化

任务2的分类器不能完成任务1



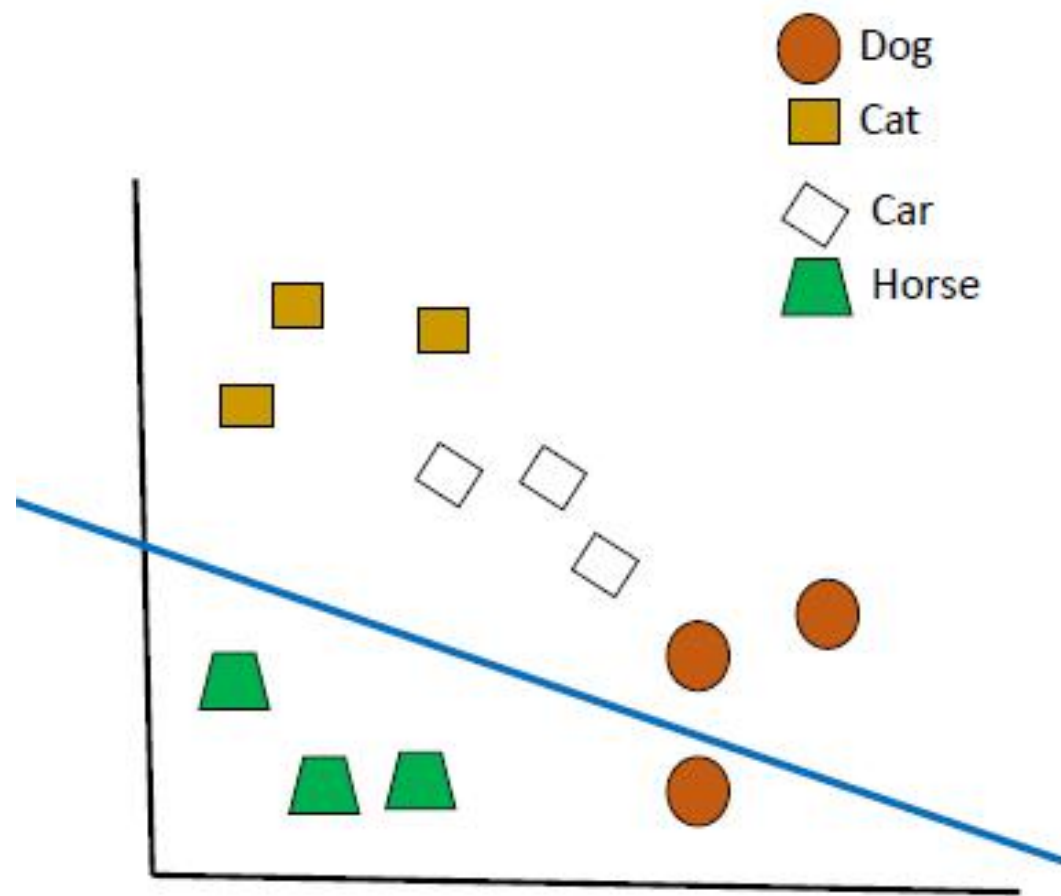
3.1 持续学习

例子

- 2维特征
- 自由度为2

CIL: 只需要一个最优的模型

- 如在所有任务上测试, 旧任务性能不可分

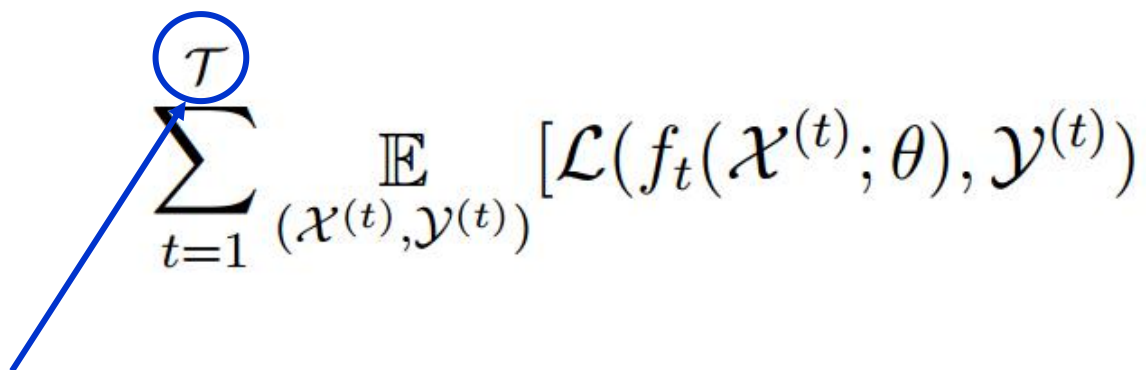


灾难性遗忘 (Catastrophic forgetting) : 在旧任务上的性能严重下降

3.1 持续学习

灾难性遗忘! Why

目标


$$\sum_{t=1}^{\tau} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})} [\mathcal{L}(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$$

所有已训练任务

3.1 持续学习



灾难性遗忘! Why

目标

The diagram shows the loss function $\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})} [\mathcal{L}(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$. A blue circle highlights the symbol \mathcal{T} above the summation, with a blue arrow pointing from the text '所有已训练任务' (All trained tasks) below. Another blue circle highlights the expectation term $\mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})}$, with a blue arrow pointing from the text '服从任务t分布的样本数据' (Sample data following the distribution of task t) below.

$$\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})} [\mathcal{L}(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$$

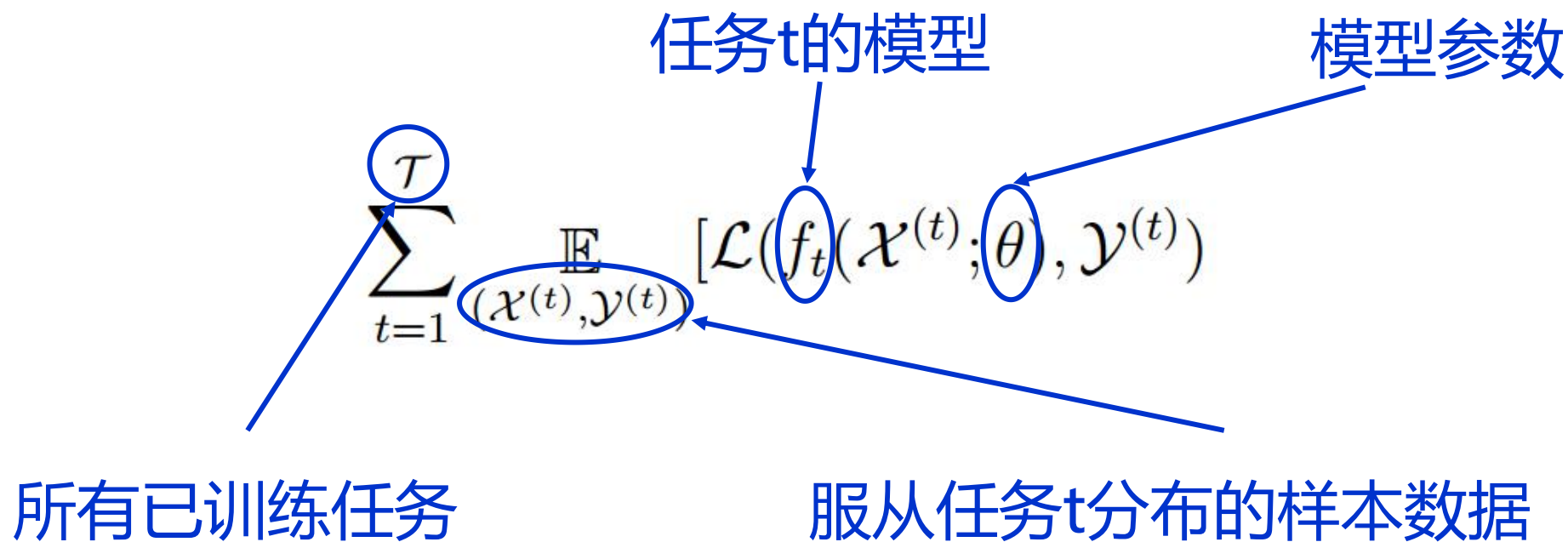
所有已训练任务

服从任务t分布的样本数据

3.1 持续学习

灾难性遗忘! Why

目标



The diagram illustrates the objective function for continuous learning, $\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})} [\mathcal{L}(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$. Annotations with blue arrows point to specific parts of the formula:

- 任务t的模型** (Task t's model) points to f_t .
- 模型参数** (Model parameters) points to θ .
- 所有已训练任务** (All previously trained tasks) points to the summation index \mathcal{T} .
- 服从任务t分布的样本数据** (Sample data following the distribution of task t) points to the expectation term $\mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})}$.

3.1 持续学习

灾难性遗忘！ Why

目标

$$\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})} [\mathcal{L}(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$$

旧任务数据不可见!!!

实际优化

$$\frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} \ell(f(x_i^{(\mathcal{T})}; \theta), y_i^{(\mathcal{T})}) .$$

当前任务的样本数量

3.1 持续学习

灾难性遗忘! Why

目标

$$\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})} [\mathcal{L}(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$$

实际优化

$$\frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} \ell(f(x_i^{(\mathcal{T})}; \theta), y_i^{(\mathcal{T})}) .$$

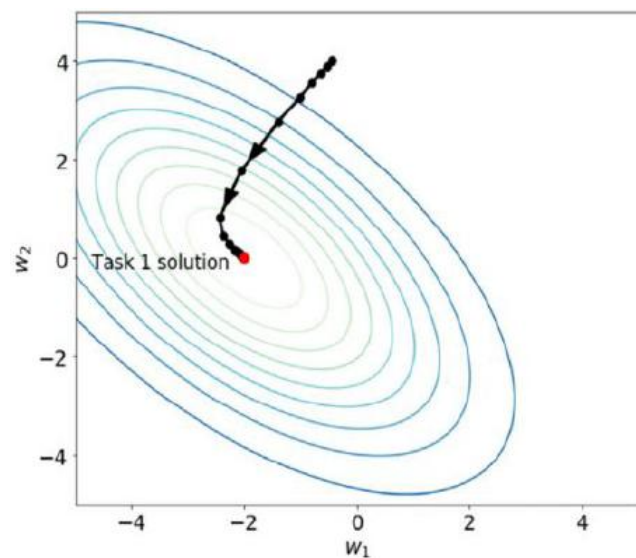
旧任务

旧任务数据不可见，不能计算风险损失，
发生灾难性遗忘

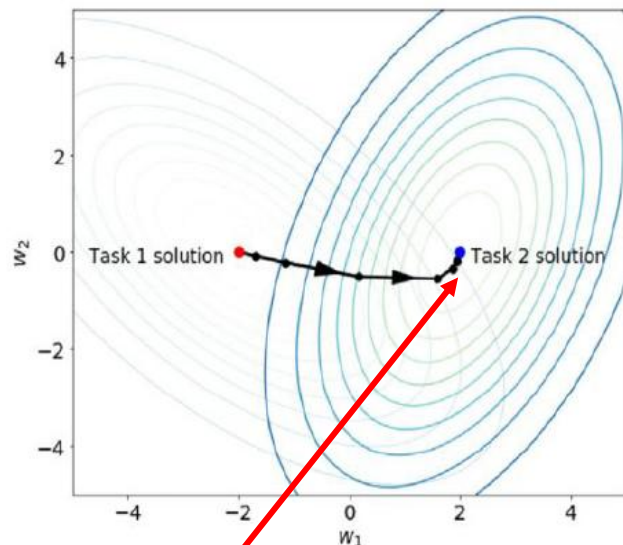
3.1 持续学习



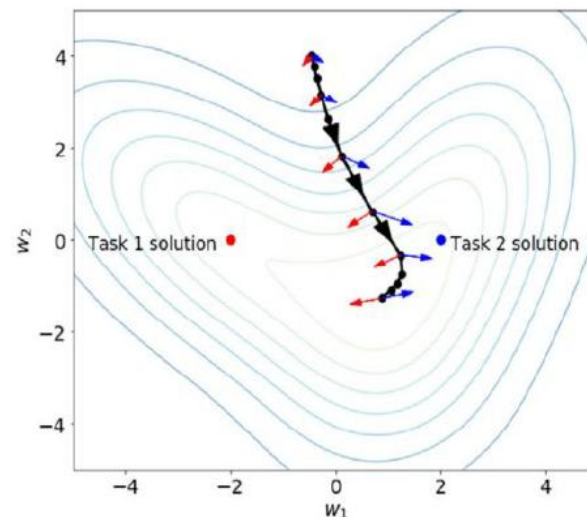
灾难性遗忘! Why



$Loss(Task1)$



$Loss(Task2)$



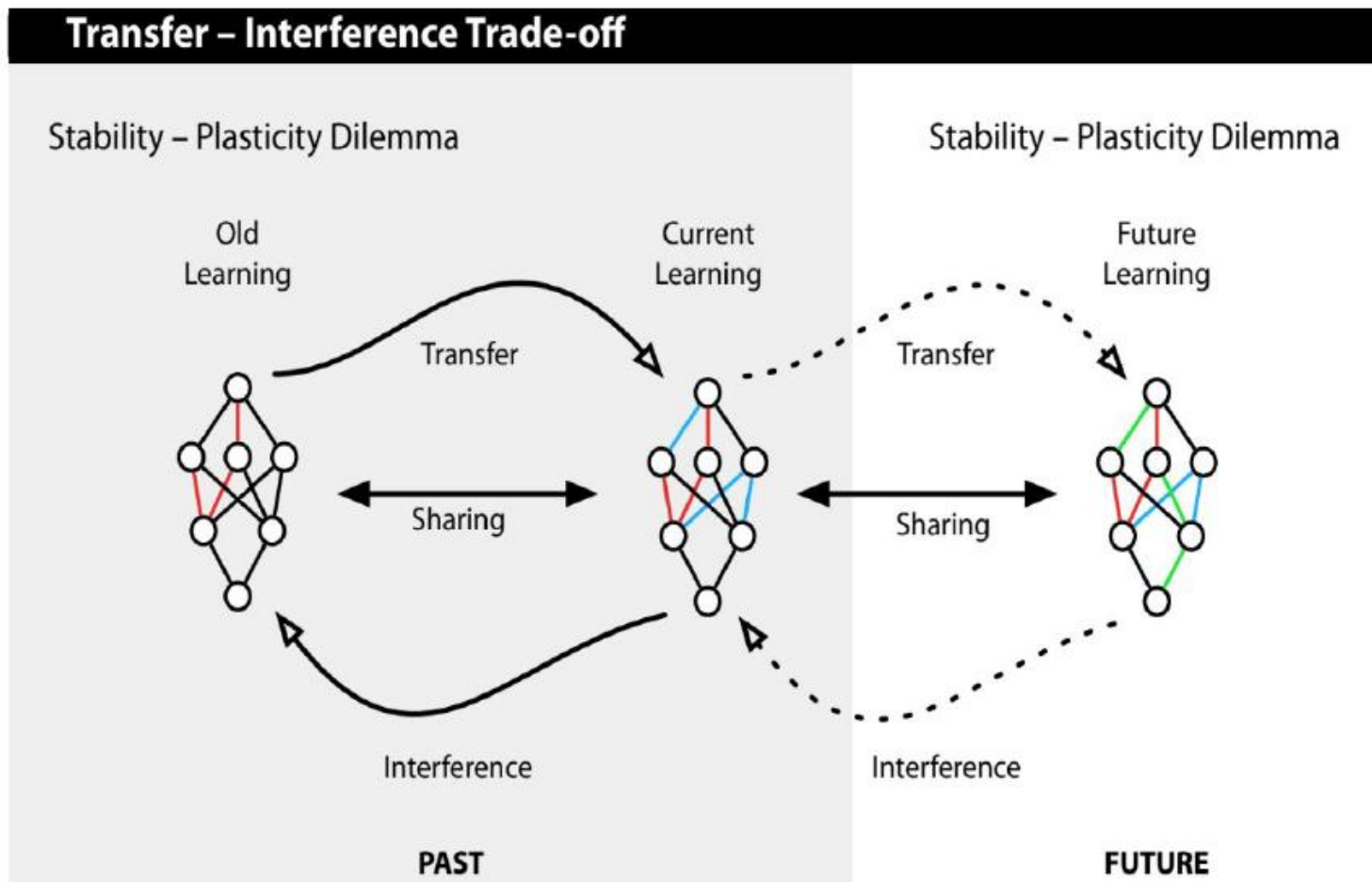
$Loss(Task1) + Loss(Task2)$

任务2性能变好
任务1性能变差

3.1 持续学习



灾难性遗忘！ Why



3.1 持续学习



已学习任务性能变化情况



灾难性遗忘

已学习任务性能急剧下降



3.1 持续学习

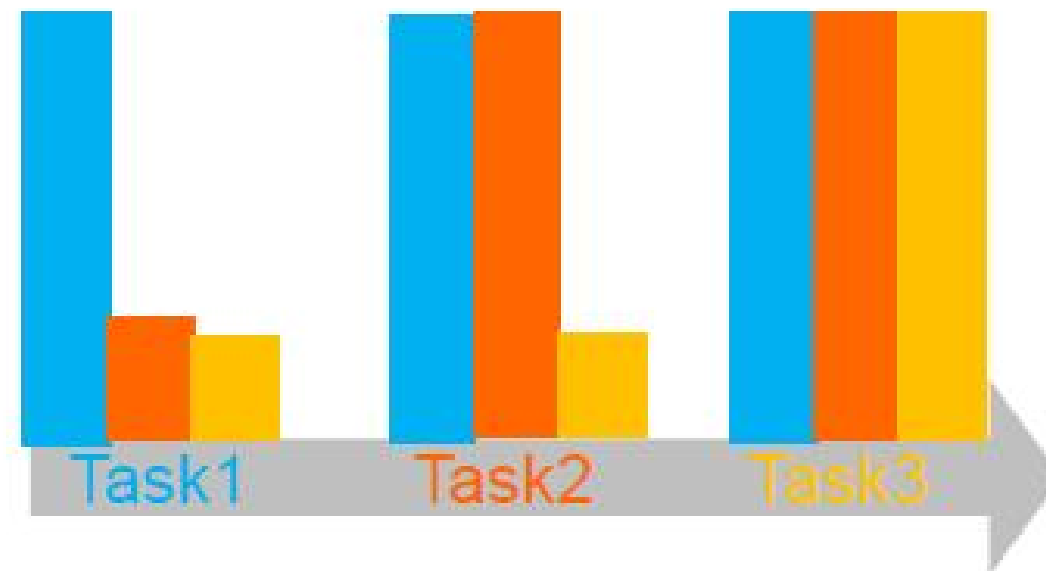


已学习任务性能变化情况



无遗忘

已学习任务性能保持不变



3.1 持续学习

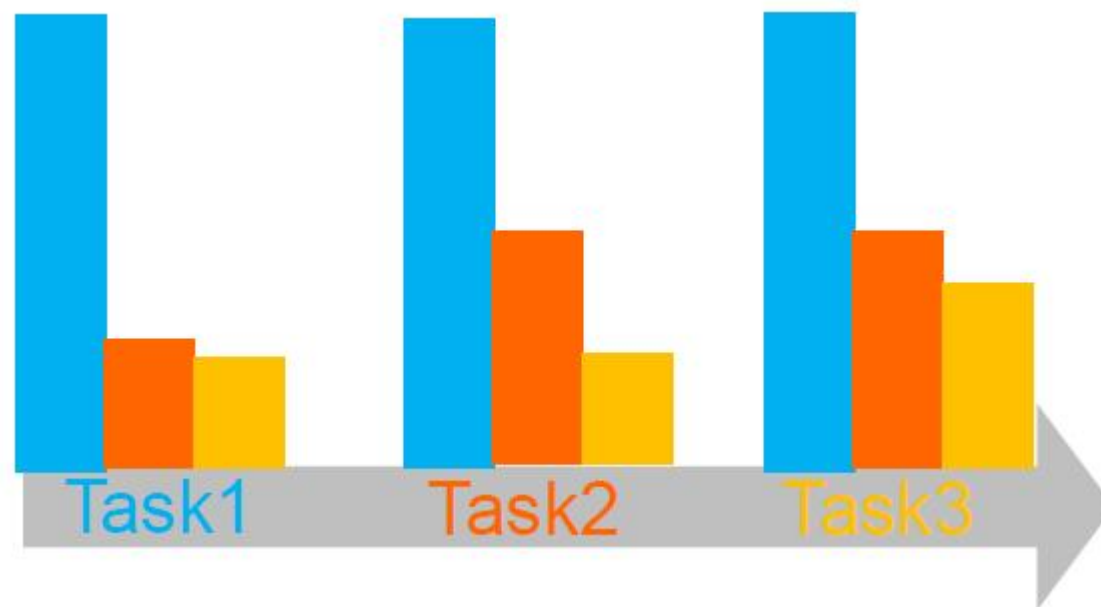
已学习任务性能变化情况



已学习任务性能保持不变,
新任务性能较差

Why

- 过度正则化
- 网络性能不够



3.1 持续学习

已学习任务性能变化情况

已学习任务性能保持不变,
新任务性能提高

前向传播:
学习的知识有助于新任务



3.1 持续学习

已学习任务性能变化情况

已学习任务和新任务性能都提高

前向和后向传播：

已学习的知识有助于新任务

新任务知识也有助于旧任务

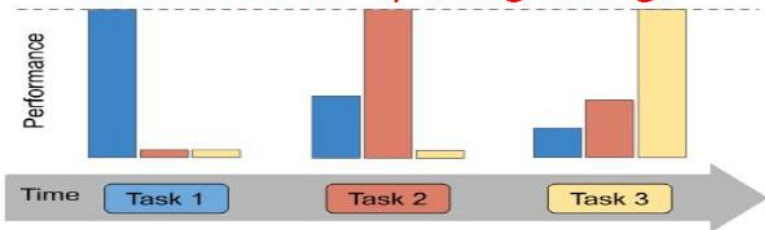


3.1 持续学习

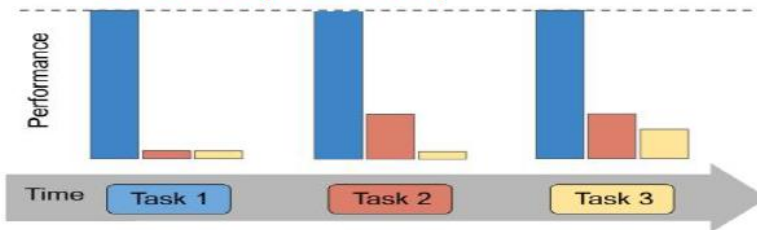
任务性能变化情况



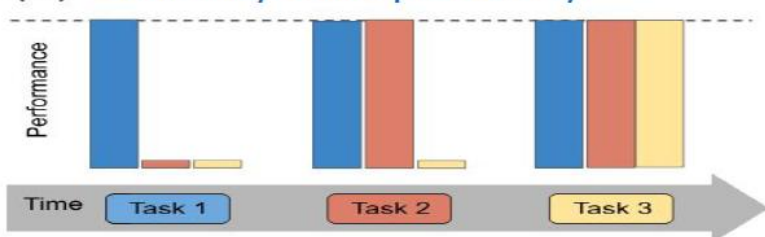
(A) Lack of stability (forgetting)



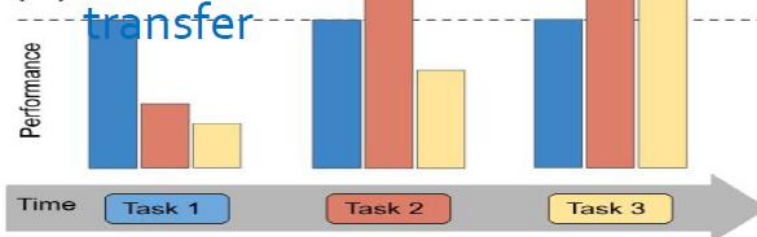
(B) Lack of plasticity



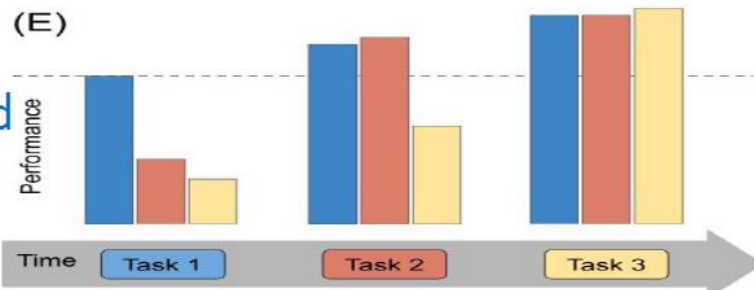
(C) Stability and plasticity



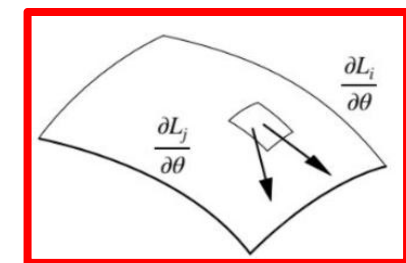
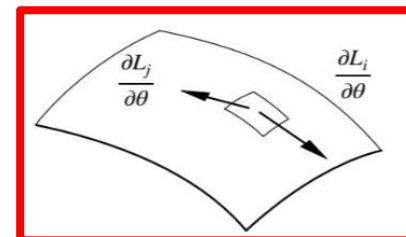
(D) Stability + (positive) forward transfer



(E)



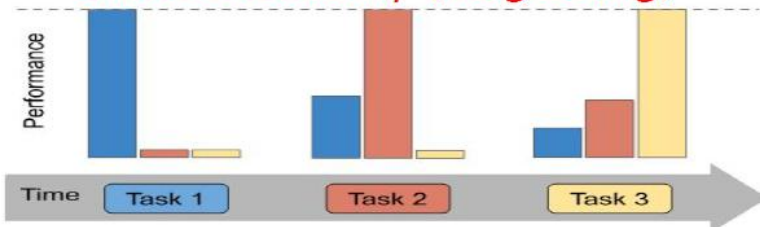
Both (positive) backward and forward transfer



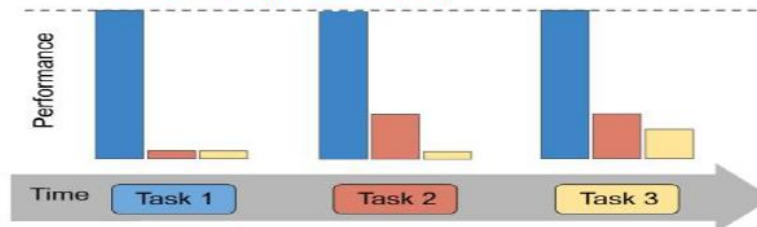
3.1 持续学习



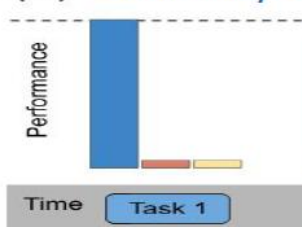
(A) Lack of stability (forgetting)



(B) Lack of plasticity



(C) Stability and plasticity

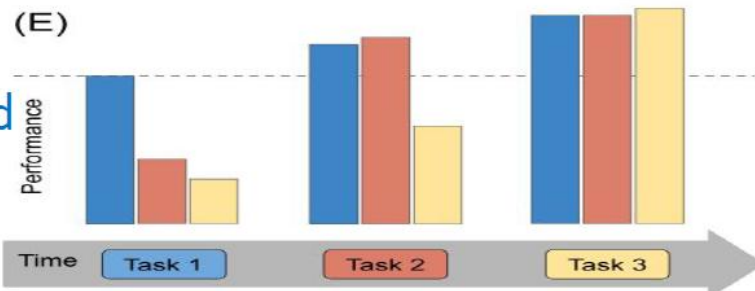


(D) Stability + (positive) forward transfer



如何量化不同场景?

Both (positive) backward and forward transfer



3.1 持续学习

评价准则

缺乏通用的评价准则，下面主要讨论Meta Table

		测试任务					
		T ₁	T ₂	T ₃	T ₄	T ₅
训练任务	T ₁	R _{1,1}					
	T ₂	R _{2,1}	R _{2,2}				
	T ₃	R _{3,1}	R _{3,2}	R _{3,3}			
	T ₄	R _{4,1}	R _{4,2}	R _{4,3}	R _{4,4}		
	T ₅	R _{5,1}	R _{5,2}	R _{5,3}	R _{5,4}	R _{5,5}	
	⋮	⋮					

$R_{m,n}$: 模型在训练完M任务后在任务n上的性能

3.1 持续学习

评价准则：Meta Table

$R_{m,n}$: 模型在训练完M任务后在任务n上的性能

		测试任务					
		T_1	T_2	T_3	T_4	T_5
训练任务	T_1	$R_{1,1}$					
	T_2	$R_{2,1}$	$R_{2,2}$				
	T_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$			
	T_4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$		
	T_5	$R_{5,1}$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$	
	⋮	⋮					
		$R_{t,1}$				

如果下降->遗忘

如果上升->Backward Transfer

3.1 持续学习



评价准则：Meta Table

$R_{m,n}$: 模型在训练完M任务后在任务n上的性能

训练任务	测试任务					
	T_1	T_2	T_3	T_4	T_5
	T_1	$R_{1,1}$				
	T_2	$R_{2,1}$	$R_{2,2}$			
	T_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$		
	T_4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$	
T_5	$R_{5,1}$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$	
⋮	⋮					
	$R_{t,1}$				

如果下降->遗忘
如果上升->Backward Transfer

■ Forgetting Rate(FR)

$$\frac{1}{T-1} \sum_{i=1}^{T-1} R_{i,i} - R_{t,i}$$

■ Backward Transfer(BWT)

$$\frac{1}{T-1} \sum_{i=1}^{T-1} R_{t,i} - R_{i,i}$$

Final results

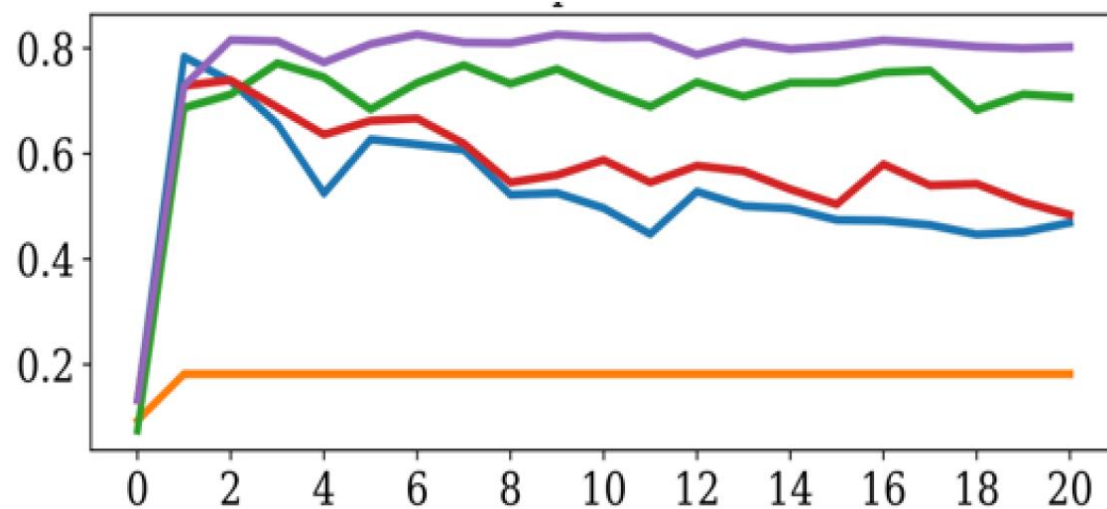
Forward results

3.1 持续学习

评价准则：Meta Table

$R_{m,n}$: 模型在训练完M任务后在任务n上的性能

		测试任务					
		T_1	T_2	T_3	T_4	T_5
训练任务	T_1	$R_{1,1}$					
	T_2	$R_{2,1}$	$R_{2,2}$				
	T_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$			
	T_4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$		
	T_5	$R_{5,1}$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$	
	⋮	⋮					
		$R_{t,1}$				



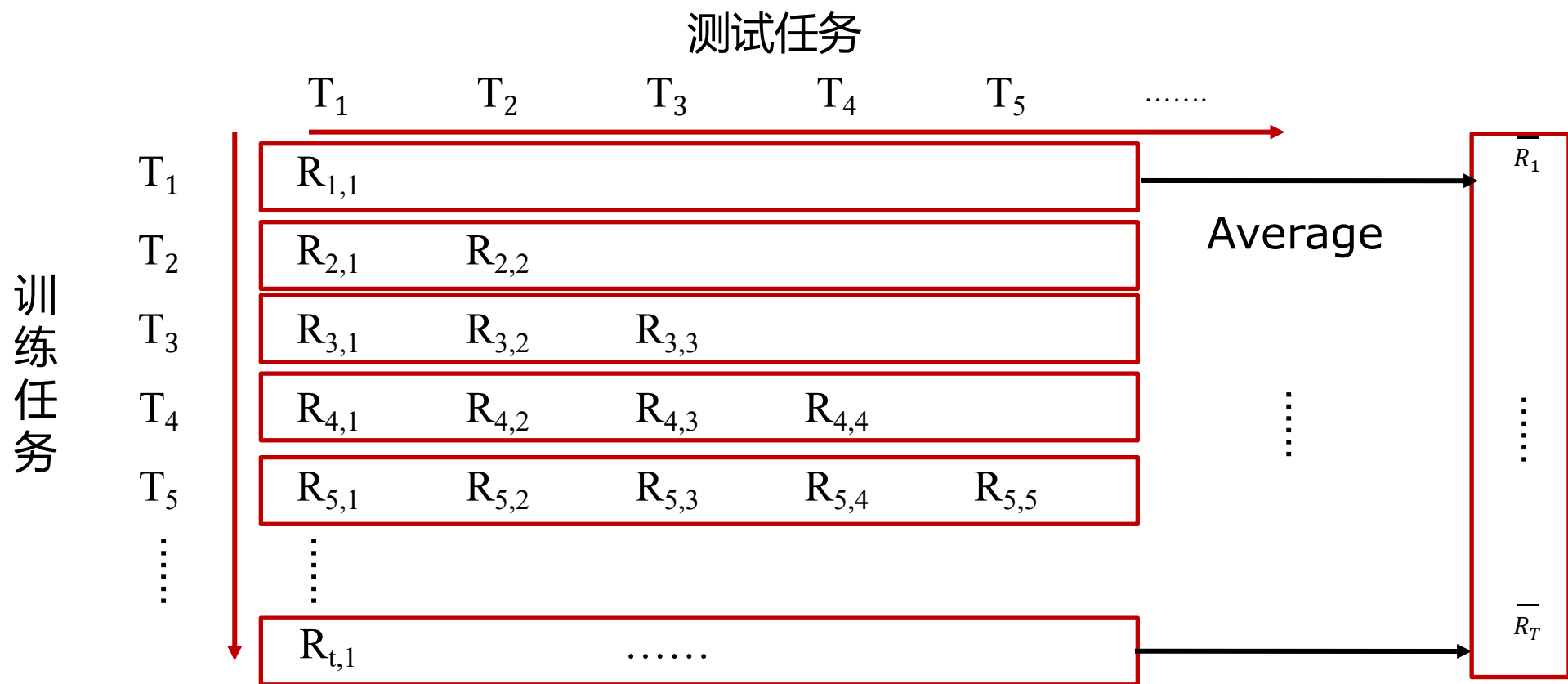
曲线图

如果下降->遗忘
如果上升->Backward Transfer

3.1 持续学习

评价准则：Meta Table

$R_{m,n}$: 模型在训练完M任务后在任务n上的性能



使用平均准确率

3.1 持续学习

评价准则：Meta Table

$R_{m,n}$: 模型在训练完M任务后在任务n上的性能

		测试任务					
		T_1	T_2	T_3	T_4	T_5
训练任务	T_1	R_1	R_2	R_3	R_4	R_5	
	T_2	$R_{2,1}$	$R_{2,2}$				
	T_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$			
	T_4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$		
	T_5	$R_{5,1}$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$	
	\vdots	\vdots					
		$R_{t,1}$				

为每一个任务训练一个模型，
作为非CL Baseline

■ Forward Transfer(FWT)

$$\frac{1}{T-1} \sum_{i=1}^{T-1} R_{i,i} - R_i$$

判断是否发生Forward
Transfer(FWT>0)

3.1 持续学习

评价准则：Meta Table

$R_{m,n}$: 模型在训练完M任务后在任务n上的性能

		测试任务					
		T_1	T_2	T_3	T_4	T_5
训练任务	T_1	$R_{1,1}$					
	T_2	$R_{2,1}$	$R_{2,2}$				
	T_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$			
	T_4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$		
	T_5	$R_{5,1}$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$	
	⋮	⋮					
		$R_{t,1}$				

最常用

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

平均准确率：在学习完成后，计算所有任务结果

3.1 持续学习

方法分类

$$\text{目标 } \sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})} [\mathcal{L}(f_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})]$$

$$\text{实际优化 } \frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} \ell(f(x_i^{(\mathcal{T})}; \theta), y_i^{(\mathcal{T})}).$$

旧任务数据不可见，不能计算风险损失，导致灾难性遗忘

■ 基于正则化的CL

无法计算旧任务风险 → 添加损失约束，保留旧任务知识

■ 基于回放的CL

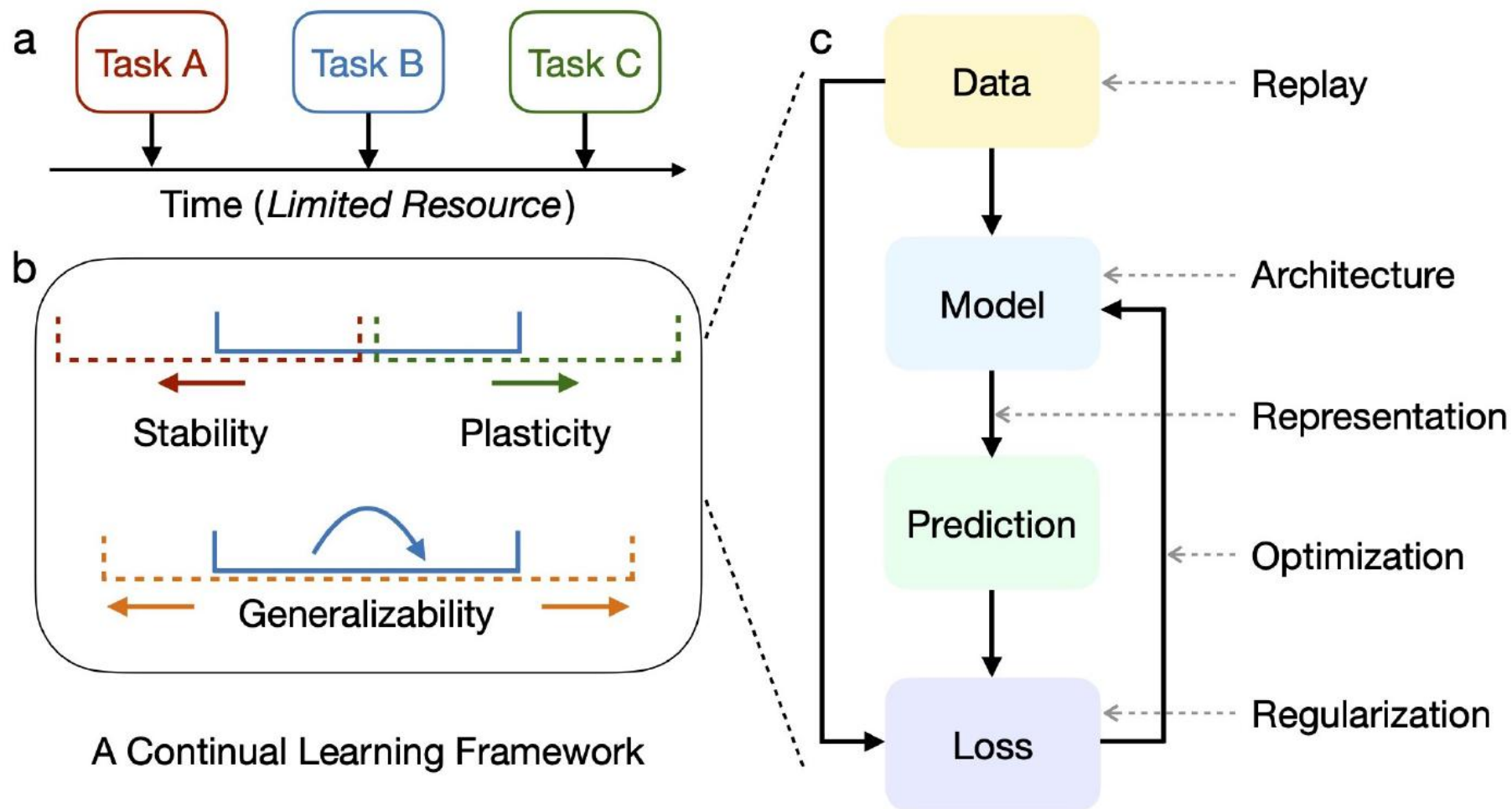
旧任务数据不可见 → 保存部分样本、生成旧任务数据

■ 基于网络结构的CL

模型能力弱 → 扩展网络结构，每一个任务是一个子网络

3.1 持续学习

其它分类方式



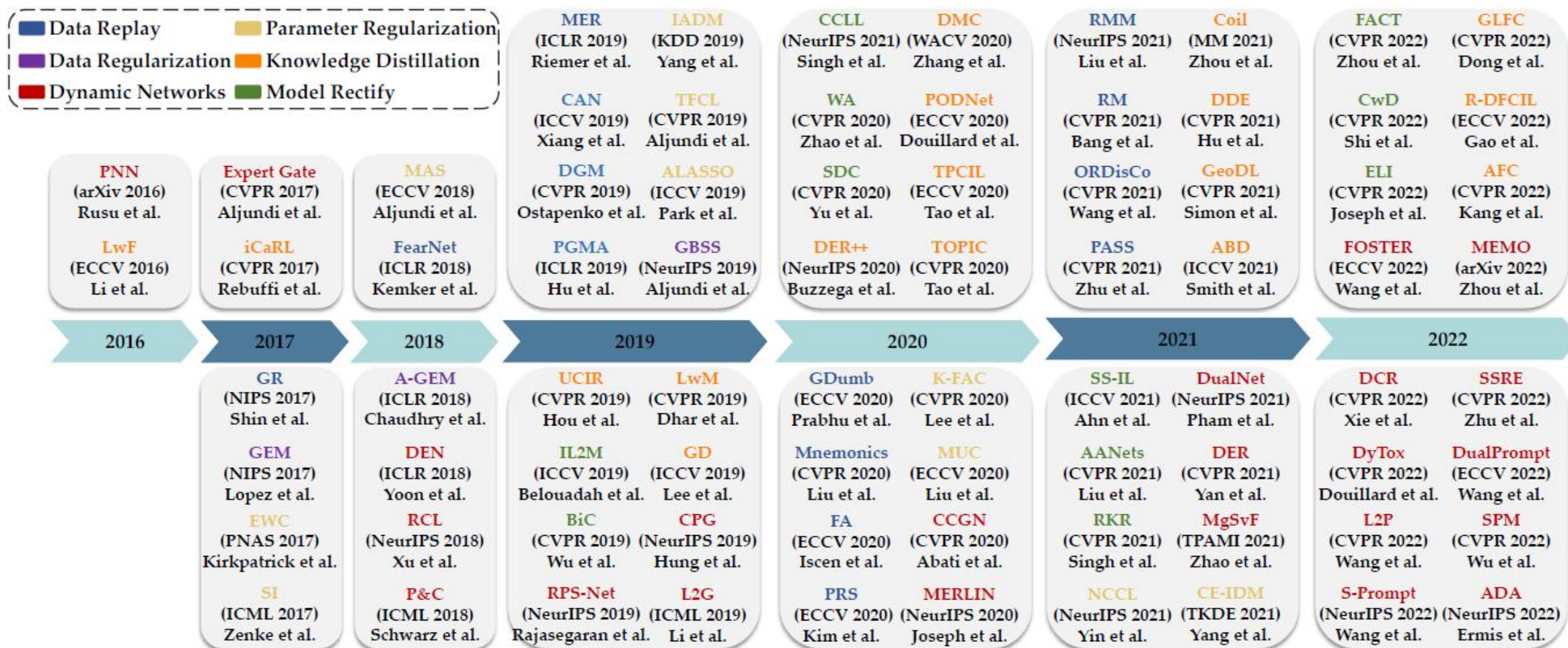
3.1 持续学习

其它分类方式

Algorithm Category	Subcategory		Reference
§ 3.1 Data-Centric Class-Incremental Learning	§ 3.1.1 Data Replay	Direct Replay	[35], [39], [40], [41], [42], [43], [44], [45], [46], [47]
		Generative Replay	[48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58]
	§ 3.1.2: Data Regularization		[40], [59], [60], [61], [62], [63]
§ 3.2 Model-Centric Class-Incremental Learning	§ 3.2.1 Dynamic Networks	Neuron Expansion	[64], [65], [66]
		Backbone Expansion	[20], [21], [22], [67], [68], [69], [70], [71]
		Prompt Expansion	[23], [72], [73], [74], [75], [76]
	§ 3.2.2: Parameter Regularization		[39], [77], [78], [79], [80], [81], [82], [83], [84]
§ 3.3 Algorithm-Centric Class-Incremental Learning	§ 3.3.1 Knowledge Distillation	Logit Distillation	[32], [85], [86], [87], [88], [89], [90], [91], [92]
		Feature Distillation	[93], [94], [95], [96], [97], [98], [99], [100]
		Relational Distillation	[101], [102], [103], [104], [105]
	§ 3.3.2 Model Rectify	Feature Rectify	[106], [107], [108], [109], [110], [111]
		Logit Rectify	[87], [93], [112], [113], [114]
		Weight Rectify	[115], [116], [117]

3.1 持续学习

Road Map



谢 谢！