A Survey Of Incremental Learning

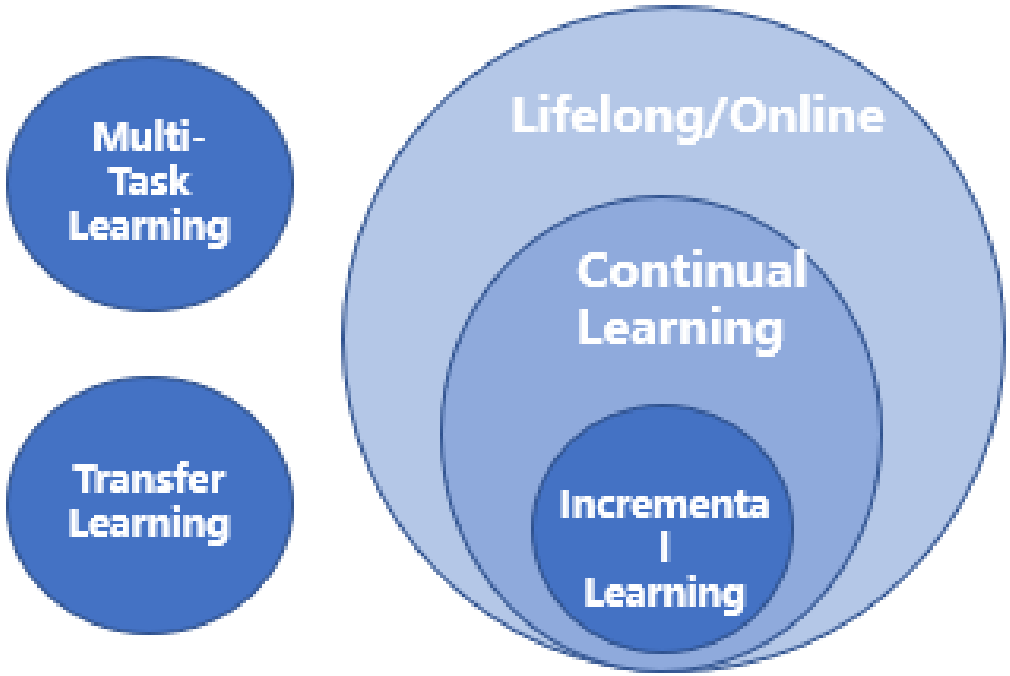
Tianchen Zhao

Index

- **概念辨析**
- 应用场景
- 发展路线
- 方法介绍与对比

Definition

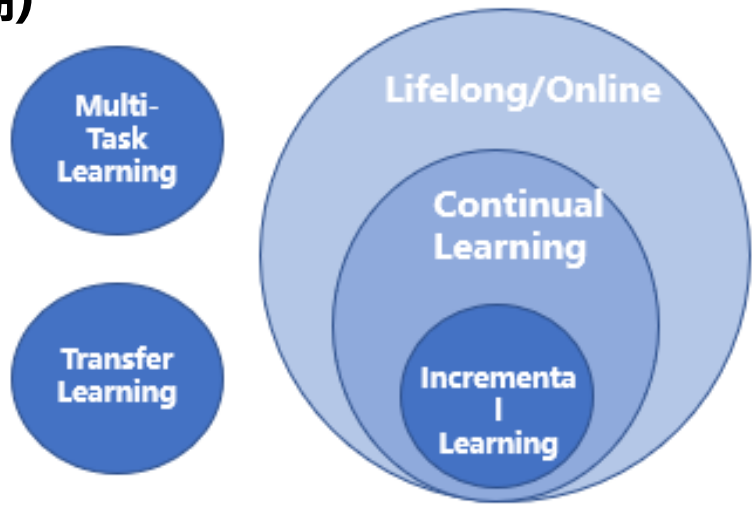
- 广义地来讲，是能够继续学习的模型，又可以分为以下几类，多有联系
  - LifeLong/Online Learning
  - Transfer Learning
  - Multitask-Learning
  - Continual Learning
  - **Incremental Learning**



Definition

- LifeLong/Online Learning　　可以继续学习的模型
- Transfer Learning　　由已知任务A的模型训练B的模型 (不Care任务A)
- Multitask-Learning　　能同时完成任务AB的模型 (AB数据同时Present)
- Continual Learning　　(在NN领域，与IL经常混用)

- **Incremental Learning (定义较为模糊)**
  - **学习任务B,并且保证任务A精度**
  - **在实际场景仅有任务B数据Available**
  - **有存储上限** //模型不会无限变大
  - **模型动态变化**
  - **(输入数据以Streaming形式)**



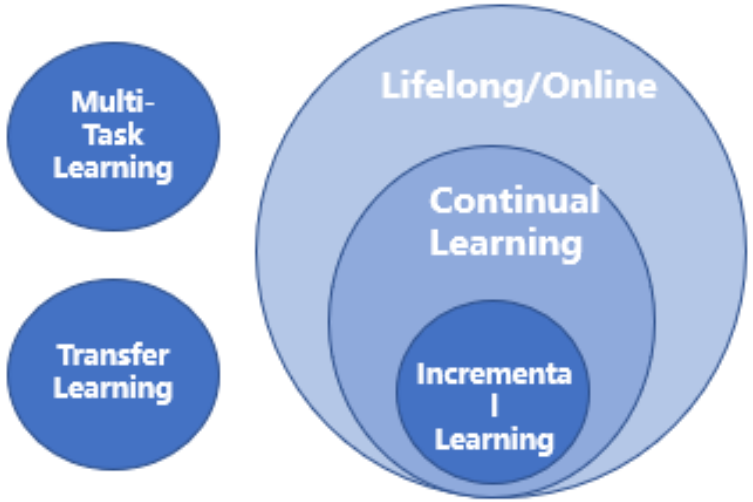
Definition

– Incremental Learning

- 是一个传统ML的问题，基于SVM，PCA，Clustering等算法在2015年之前有一些探索
- 在NN兴起之后，**研究方向**和**处理方案**

  与之前**完全不同**，甚至对IL的定义, 都有微妙的区别（比如 Sequential Data）
- 今天以**“后DL”时代** 的Incremental（Continual）

  Learning为主题

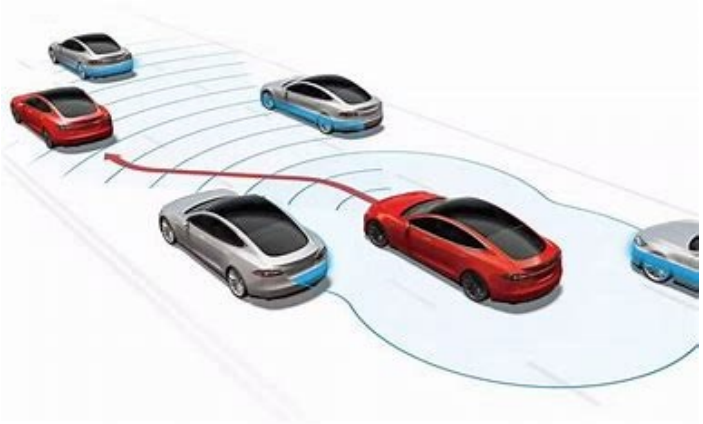


Index

- 概念辨析
- **应用场景**
- 发展路线
- 方法介绍与对比

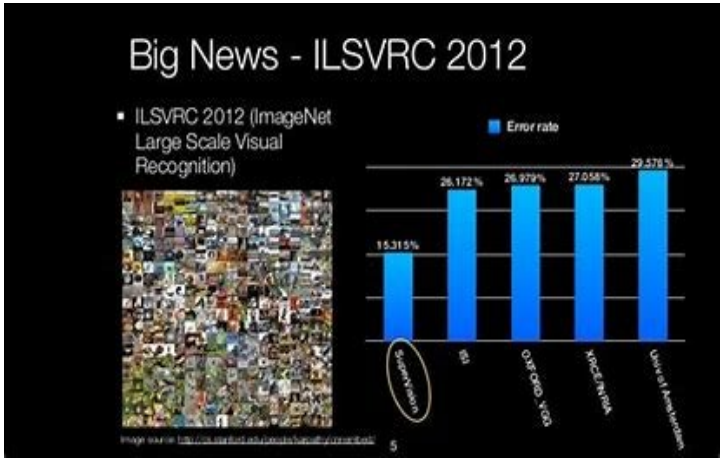
Application Field

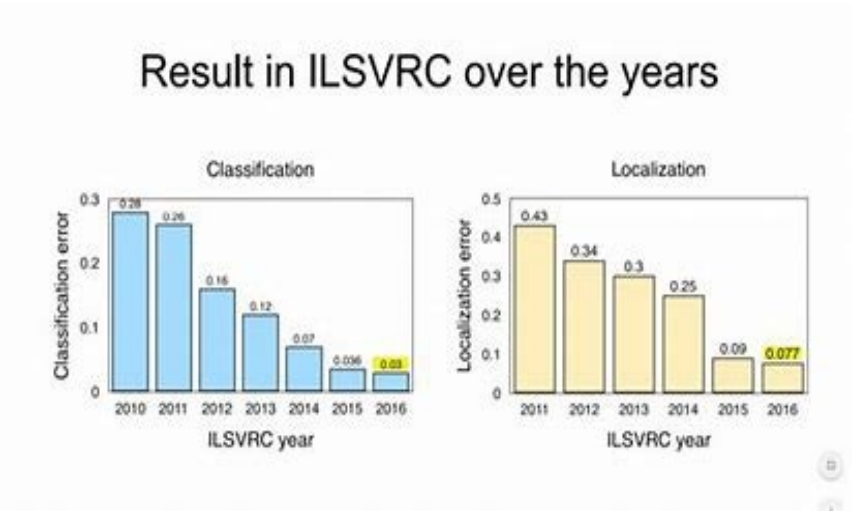
- 整个领域相对较为**理论**，实际**应用落地**较少
- 被提出可以用在**云端**（Cloud Service）
  - 与大数据联系，用于**推荐系统**等场景，针对用户修改模型，**更贴合用户使用习惯**
- 可以用于**终端**
  - 在Robotic，Auto-driving等领域，学习新的场景，减少**通信**需求，注重**数据隐私性**



Application Field

- 目前没有**实际的应用落地**的例子
- 主要还是在处理一些**简单**的问题（以 **Image Classification** 为常规做法）
  - 由于整个领域**缺少一个Baseline**标杆（类似于ILVSRC一样的标准化指标）
  - 不同的论文在实现时的**任务也有所不同** (即使都是**图像分类**问题)





Testing Scenario

- Incremental Learning在Image Classification问题上的常用数据集: Mnist的变体
  - **Split-Mnist**
  - **Permuted-Mnist**

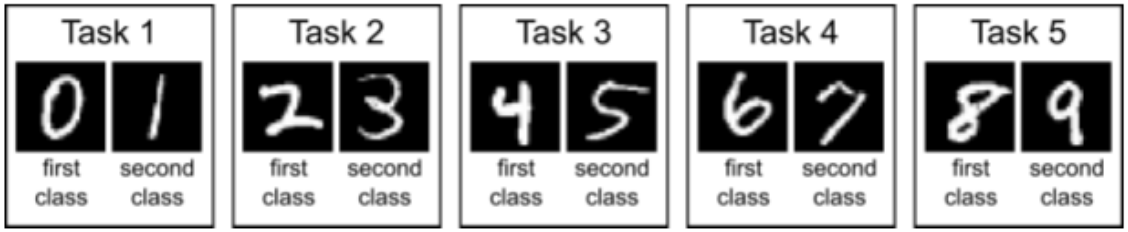


Figure 1: Schematic of split MNIST task protocol.

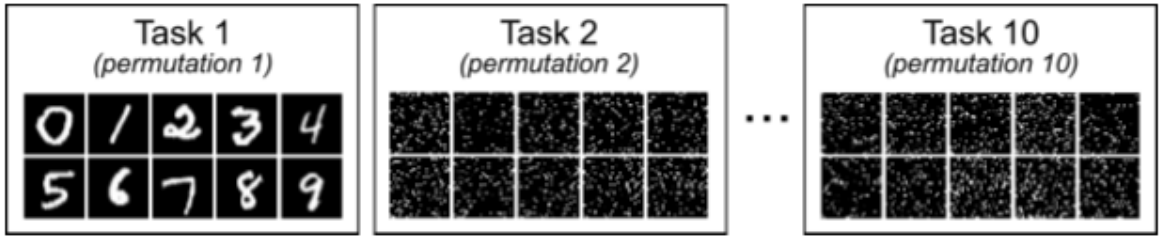
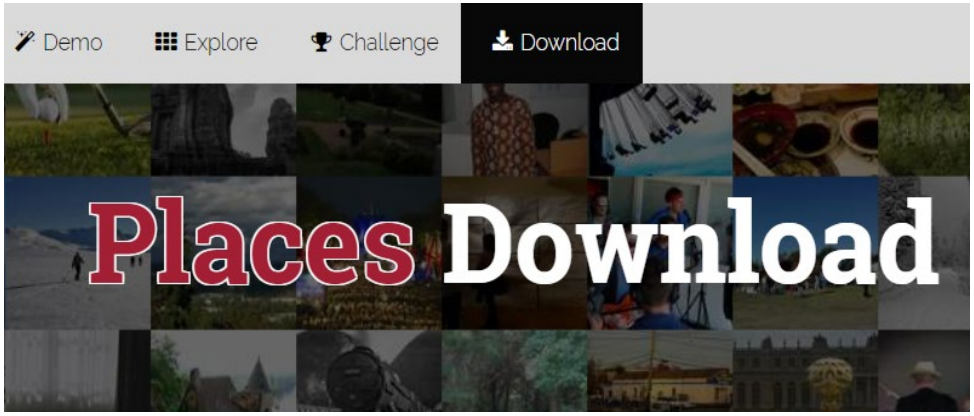


Figure 2: Schematic of permuted MNIST task protocol.

Testing Scenario

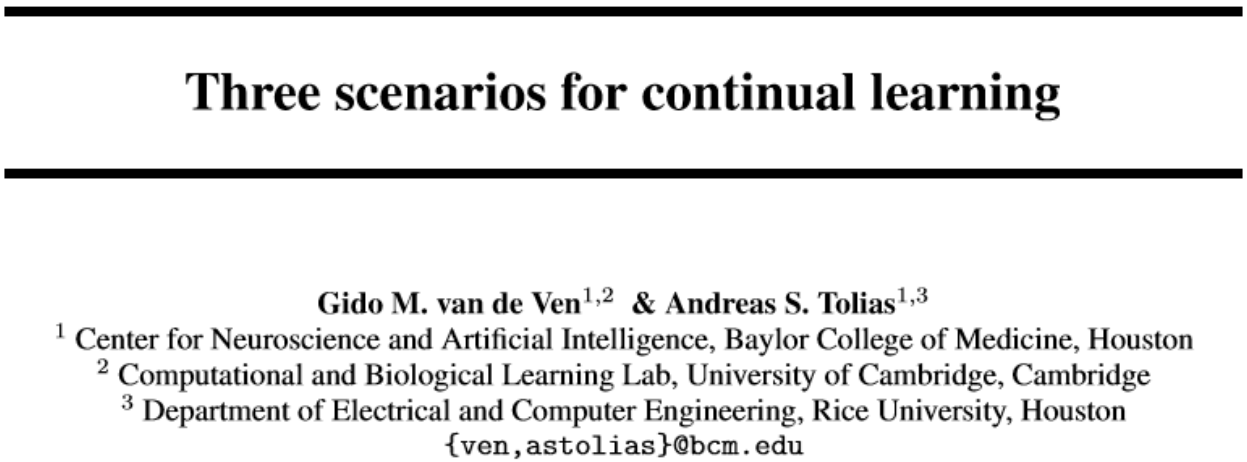
- 从一个数据集的学习,跳到另一个数据集的学习

  (ImageNet -> Place365)



Testing Scenario

- 在文章[1]中梳理出了Incremental Learning在Image Classification领域的三种场景(**Scenario**)
- 分别是
  - Task-IL
  - Domain-IL
  - Class-IL



Three scenarios for continual learning

Gido M. van de Ven[1,2] **& Andreas S. Tolias**[1,3]
[1] Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston
[2] Computational and Biological Learning Lab, University of Cambridge, Cambridge
[3] Department of Electrical and Computer Engineering, Rice University, Houston
{ven,astolias}@bcm.edu

Testing Scenario

- **Task-IL：** 给定任务标签,解决现有任务 (往往有Task-Specific的设计)
  - 例子: 先在一个数据集上训练,然后再更换数据集进行训练,测试时分开测试

- **Domain IL:** 不给定任务标签,解决现有任务 (但是多种任务类型相同,只是输入分布不同)
  - 实际场景: 一个智能体需要在多种环境下完成某一特定任务
  - 例子: Permuted Mnist (每个任务等价,只是输入数据Distribution不同)

- **Class IL:** 不给定任务标签,判断出属于哪种任务,并且完成任务
  - 实际场景: 可拓展的物体识别问题
  - 例子: Split Mnist (不断扩充直至为10分类)

Testing Scenario

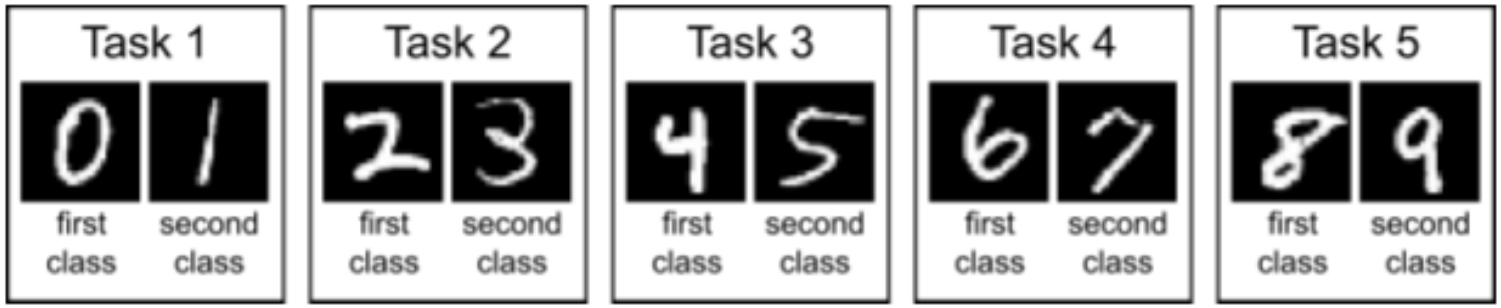


Figure 1: Schematic of split MNIST task protocol.

Table 2: Split MNIST according to each scenario.

Task-IL	With task given, is it the 1st or 2nd class? (e.g., 0 or 1)
Domain-IL	With task unknown, is it a 1st or 2nd class? (e.g., in $[0, 2, 4, 6, 8]$ or in $[1, 3, 5, 7, 9]$)
Class-IL	With task unknown, which digit is it? (i.e., choice from 0 to 9)

Testing Scenario

- Hard Baseline
  - 由于测试场景的不同,各种方法在不同的情景下效果不好(且因任务差异较大)
  - 但是有两种情况是通用的,可以作为上下界:
    - Finetune: 单纯地减小LR在新任务上训练,相当于Do Nothing **(下界)**
    - Joint Training: 将两个任务的数据同时给网络进行训练 **(上界)**

Index

- 概念辨析
- 应用场景
- **发展路线**
- 方法介绍与对比

TimeLine

2013 Goodfellow
《An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks》[2]
提出了问题的核心 Catastrophic Forgetting
为领域奠定了基础,提出Dropout可以缓解Catastrophic Forgetting

2018
基于以上三种主流方案思想
对模型进行**改进**(End2End IL,Online EWC)
对领域进行**拓展**(DGR,RL)

TimeLine

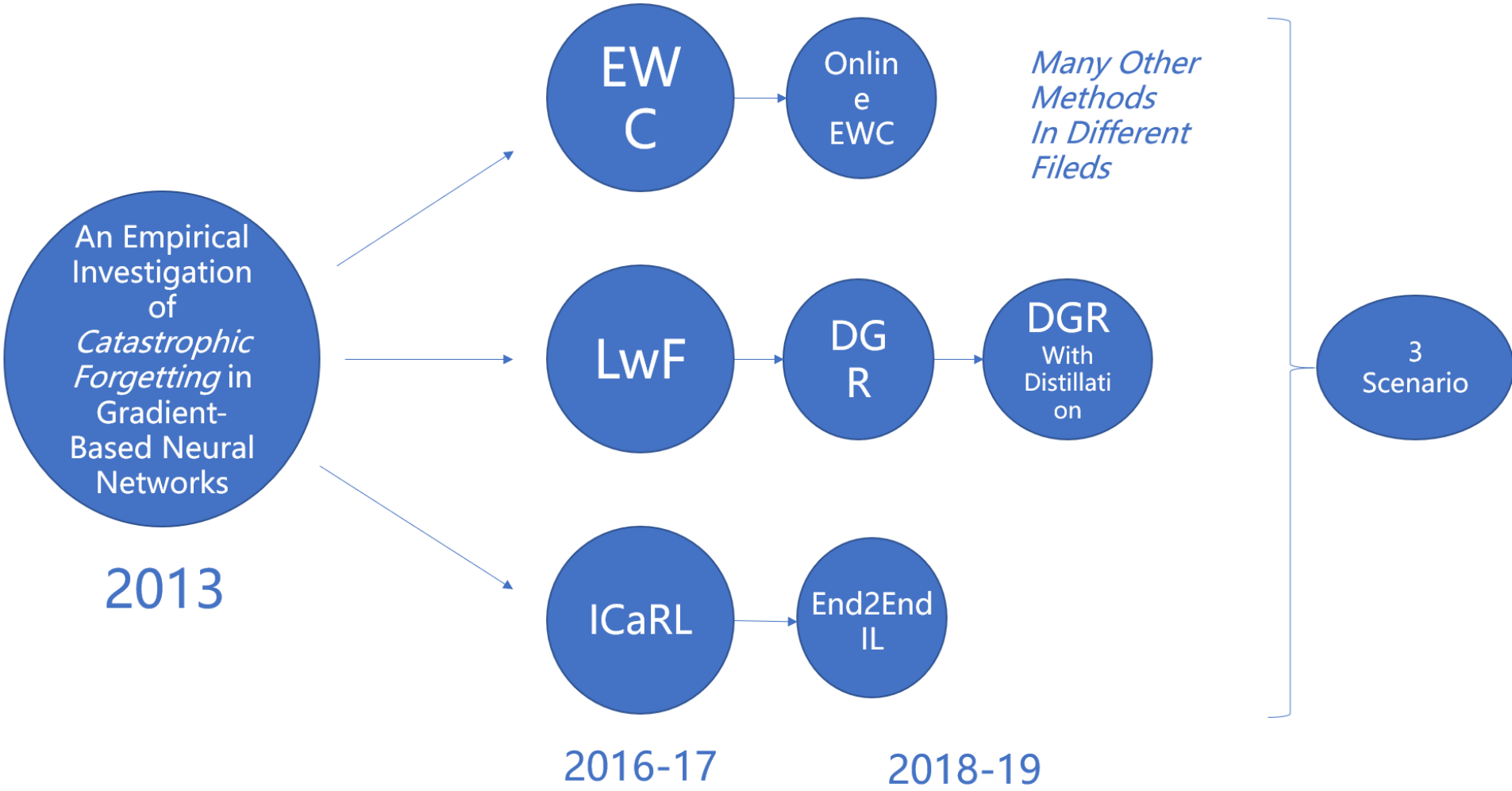
2016
《Learning Without Forgetting》[3]
提出了**LwF**的方法
最早的比较经典的IL解决方案之一

2017 DeepMind
《Overcoming catastrophic forgettinginneural networks》[4]
提出了**EWC**的方法
最早的比较经典的IL解决方案之二

2017 《**iCaRL**: Incremental Classifier and Representation Learning》[5] 经典方案之三

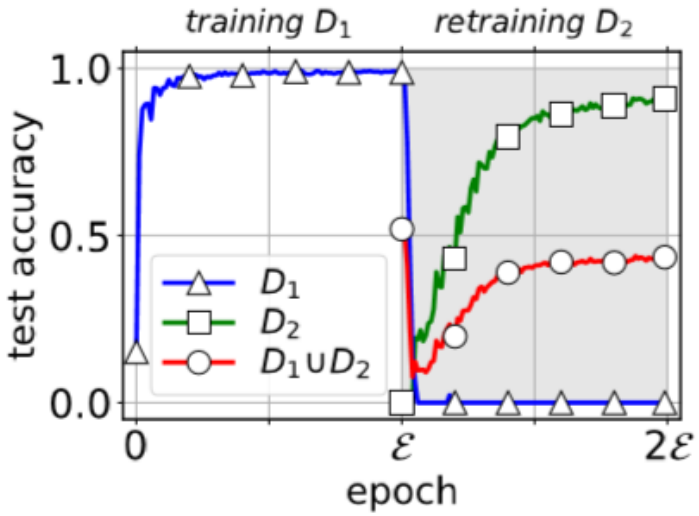
2019
《Three scenarios for continual learning》[1]
梳理了目前的几种主流IL方法并给出场景对比

TimeLine

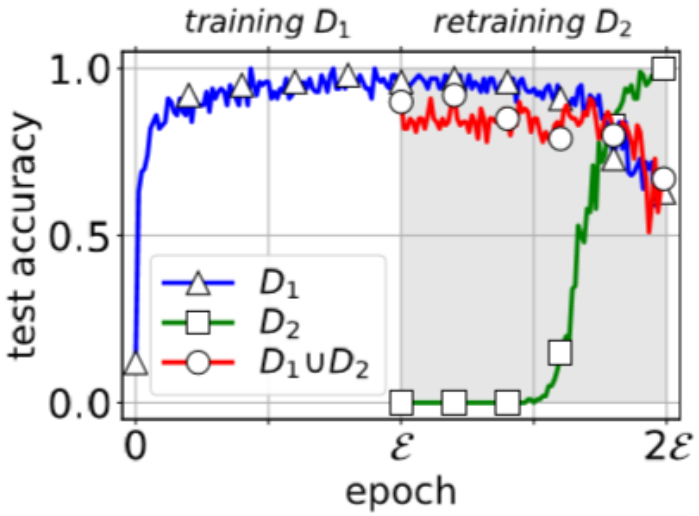

An Empirical Investigation of *Catastrophic Forgetting* in Gradient-Based Neural Networks
2013
EWC
Online EWC
LwF
DGR
DGR With Distillation
ICaRL
End2End IL
2016-17
2018-19
Many Other Methods In Different Fileds
3 Scenario

核心问题

- Catastrophic Forgetting (灾难性遗忘)
  当学习到新任务的时候,老任务的准确率迅速下降



(b) with CF



(c) without CF

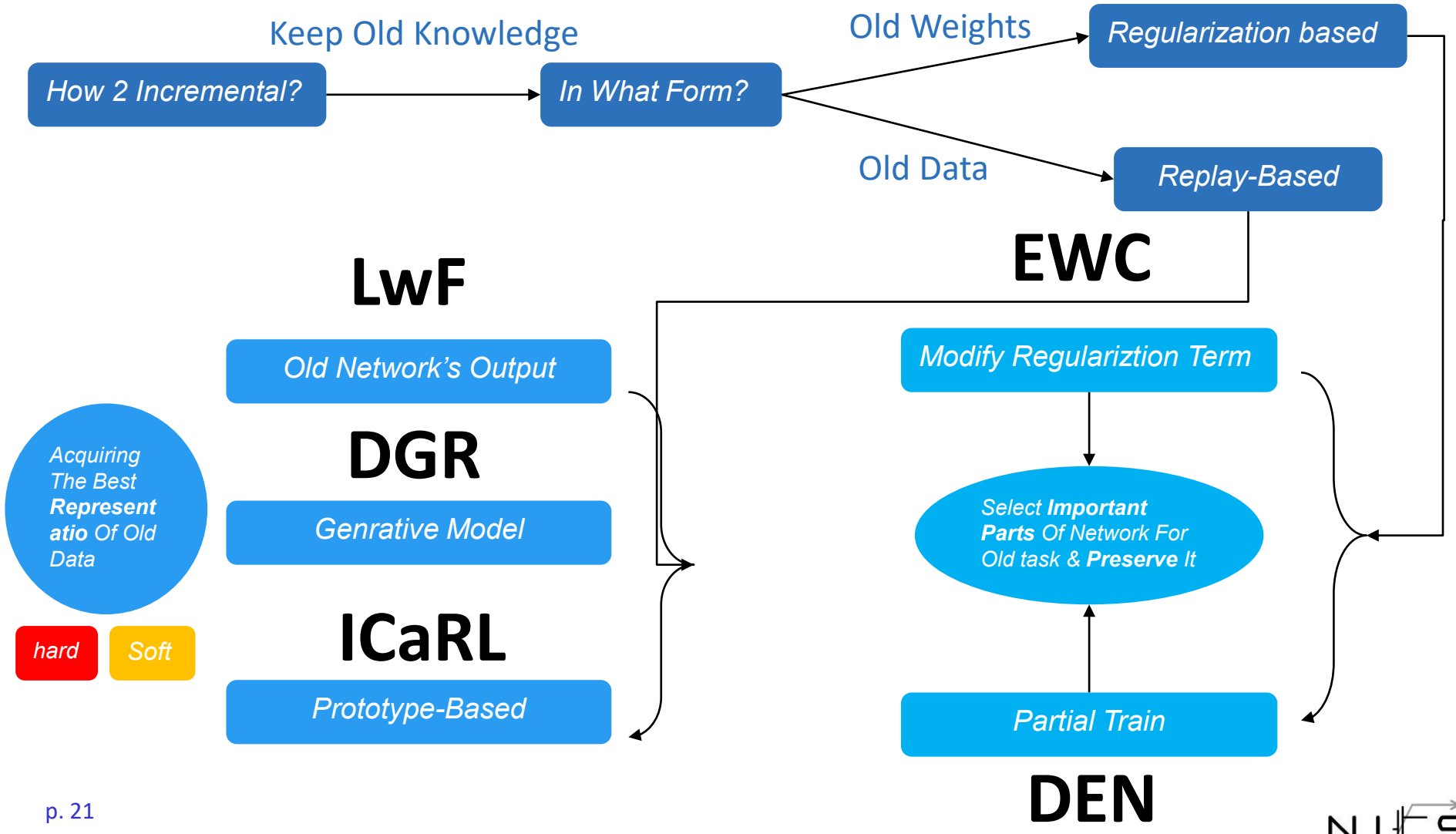
核心问题

- 需要解决 Catastrophic Forgetting (灾难性遗忘),目的是
  - **保留下Old Task的一部分信息** (与需要**学习新任务;Old Data Unavailable**矛盾)
  - 有**2**种流派
    - Regularization Based （一般是训练技巧)
    - Replay Based (修改结构)

核心问题

- Old task的信息(Old knowledge) 以什么样的形式被保存?
  - **Regularization Based** : 认为原先网络的参数分布保留了Old knowledge
    - 以一些训练上的Trick保证新的网络分布不会偏离原先分布太远
  - **Replay Based:** 认为原先任务的数据保留了Old Knowledge
    - 以不同的形式去保留原先任务的数据(虽然并不是直接availble,如果都available的那就是Joint Training了)

MindFlow


Keep Old Knowledge
How 2 Incremental?
In What Form?
Old Weights
Regularization based
Old Data
Replay-Based
EWC
LwF
Old Network's Output
DGR
Genrative Model
ICaRL
Prototype-Based
Acquiring The Best Represent atio Of Old Data
hard
Soft
Modify Regulariztion Term
Select Important Parts Of Network For Old task & Preserve It
Partial Train
DEN

Regularization Based

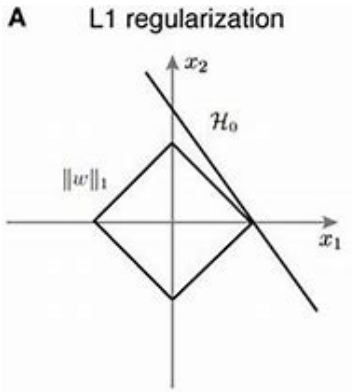
- 通过在损失函数上加正则化，来惩罚距离原分布过远的情况

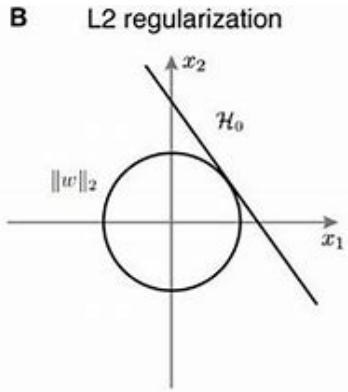
$$\mathrm{L}_{\mathrm{new}}(\theta) = L_{Original}(\theta) + \frac{\lambda}{2} * ||W^* - W_0||$$

(最朴素的L2正则)

- 通过一定程度上**保留原先网络参数的分布**以保留Old Task的知识

- **如何选择正则项?**
- ***EWC : 让对原先Task重要的参数改变更慢***
- ***Partial Train: 找出对原来Task重要的只改变其他***





Replay Based

- Replay是什么?
  - 在其他系统研究中,人们发现,当需要训练New Task而Old Task的数据不可获得,又需要保持Old Task Performance的时候
  - 输入Training Sample中加上与Old Task有关的**Pseudo-Data**,可以减缓CF
  - 该方法被叫做**Replay(重现)/Rehearsal(排练)**
  - **如何选择Pseudo-Data?**

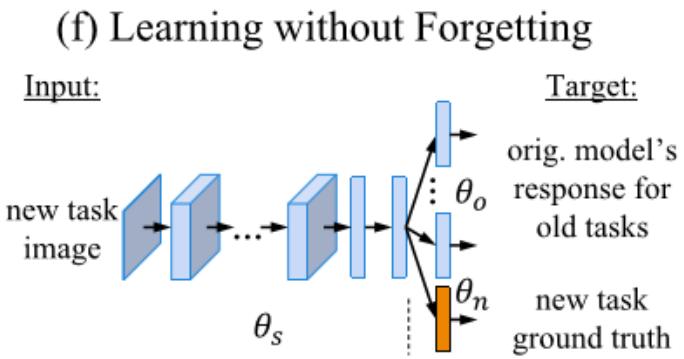
Replay **Hard/Soft**

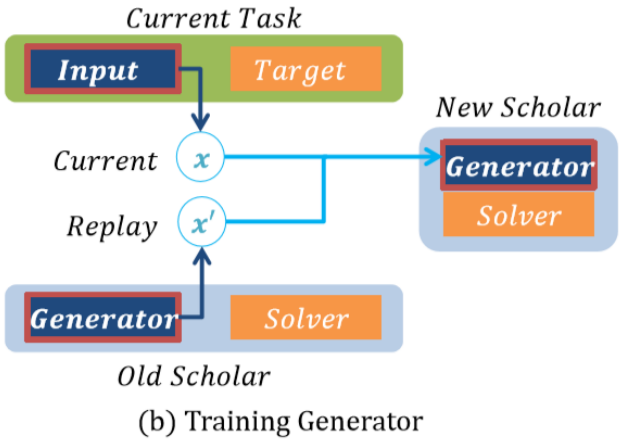
- 在选择Pseudo Data的形式时,有两种方式
  - 一种是**Hard**表达,即One-Hot形式,只有最大概率的类别信息
    - 相对更容易获得,但是损失了信息
  - 一种是**Soft**表达(类似Softmax形式),给出每个类置信度
    - 保留了更多信息,但是对Pseudo Data的生成机制提出了高要求
- 目前研究结果证明Soft形式效果更好(显然),也有一些Work是基于将原先Work的Pseudo数据表示形式从Hard变为Soft
  - DGR – DGR with Distillation

Replay Based

- LwF (Learning Without Forgetting)
  - 将新的Training Sample在训练前输入原模型获得输出

- DGR(Deep Generation Replay)
  - 专门以一个生成网络(GAN/VAC)去生成能反映旧网络的Pseudo-Data

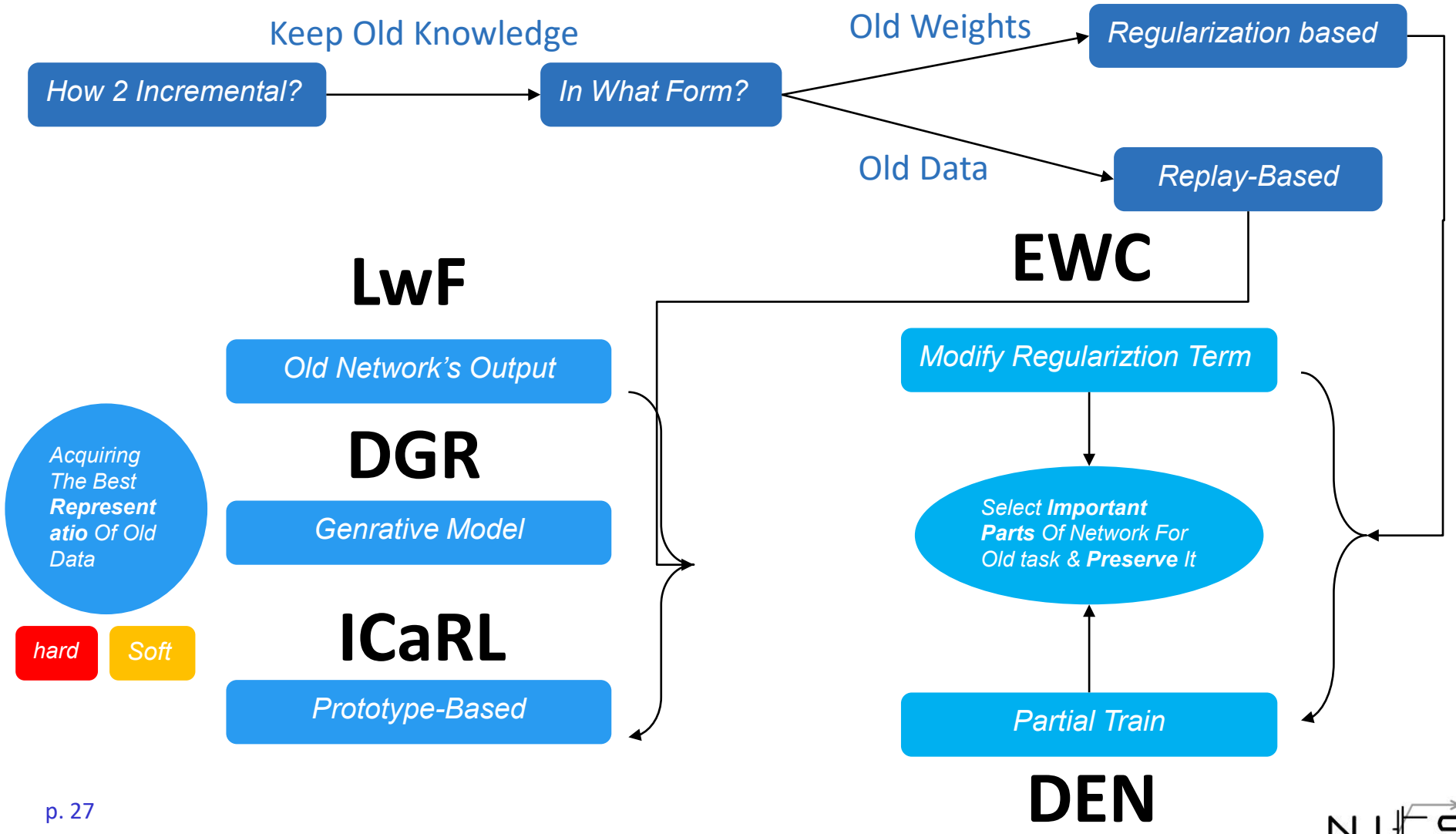




Replay Based

- ICaRL: 将目前已有的每一种类别表征成一个Weight Vector并基于其完成判别,也可以认为是一种Replay
- (与以上的方法有微妙的不同，该方法仍然需要保留旧数据,只是借助Weight Vector选择出一部分有代表性的旧数据保留)

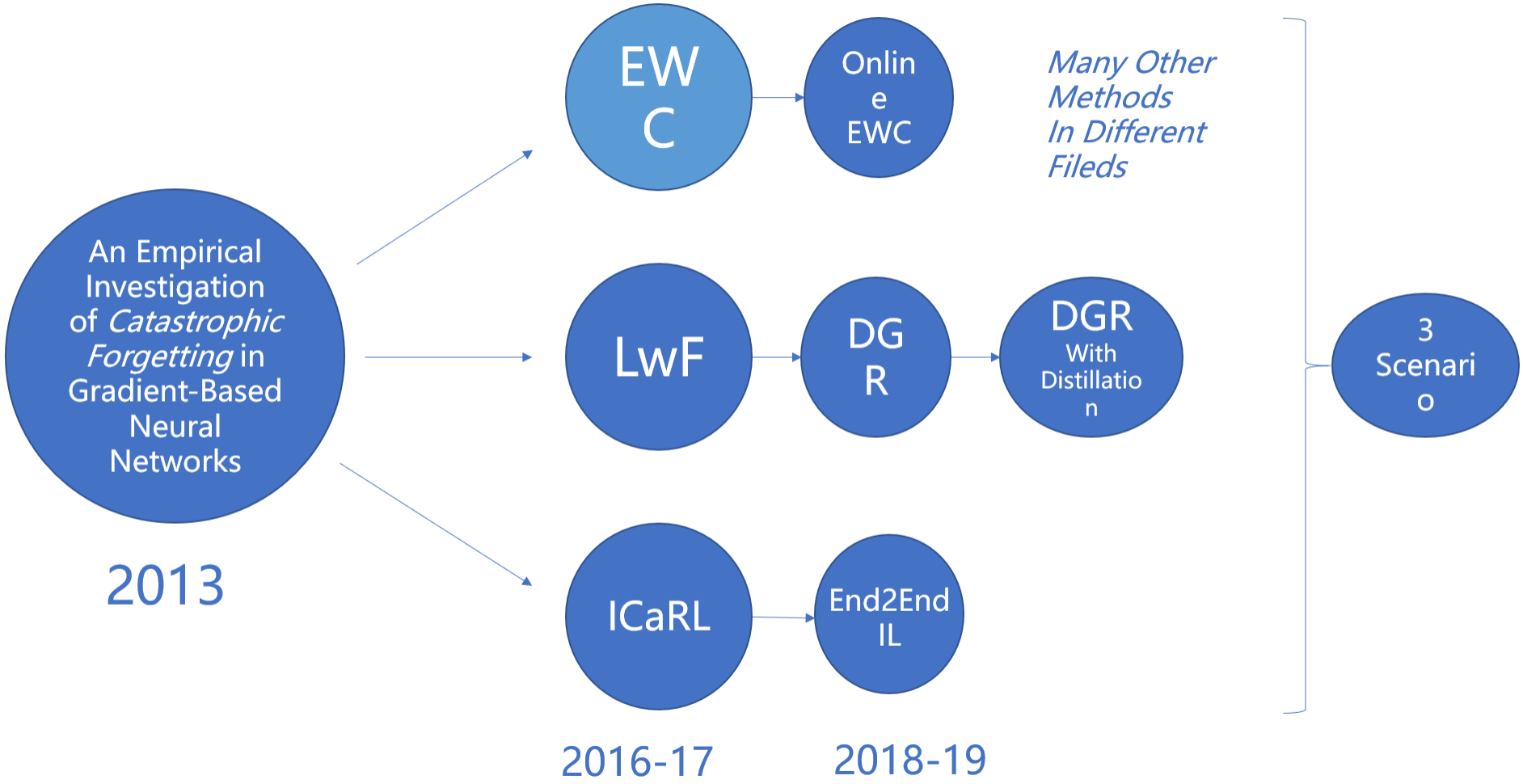
MindFlow


Keep Old Knowledge
How 2 Incremental?
In What Form?
Old Weights
Regularization based
Old Data
Replay-Based
EWC
LwF
Old Network's Output
DGR
Genrative Model
ICaRL
Prototype-Based
Acquiring The Best Represent atio Of Old Data
hard
Soft
Modify Regulariztion Term
Select Important Parts Of Network For Old task & Preserve It
Partial Train
DEN

Index

- 概念辨析
- 应用场景
- 发展路线
- **方法介绍与对比**

Methods


An Empirical Investigation of *Catastrophic Forgetting* in Gradient-Based Neural Networks
2013
EWC
Online EWC
LwF
DGR
DGR With Distillation
ICaRL
End2End IL
2016-17
2018-19
Many Other Methods In Different Fileds
3 Scenario

《Overcoming catastrophic forgetting in neural networks》

- Cite : 541
- DeepMind
- NIPS 2017

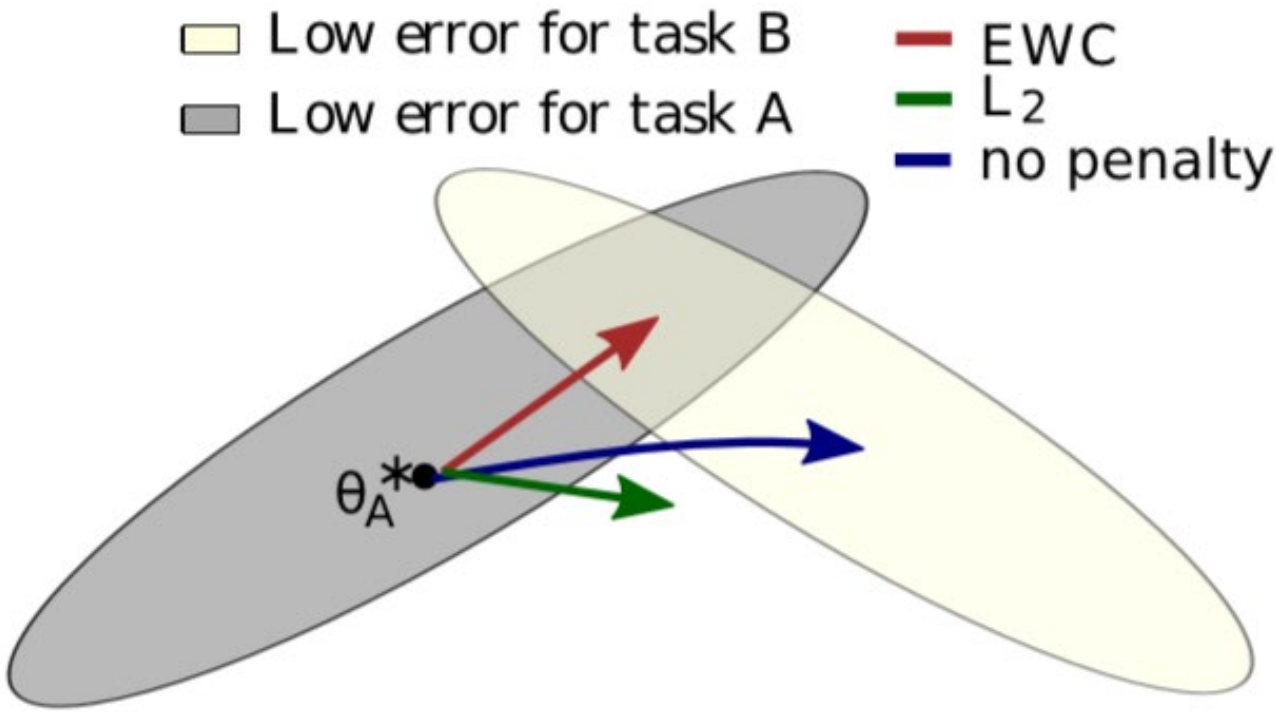
Overcoming catastrophic forgetting in neural networks

James Kirkpatrick[a], Razvan Pascanu[a], Neil Rabinowitz[a], Joel Veness[a], Guillaume Desjardins[a], Andrei A. Rusu[a], Kieran Milan[a], John Quan[a], Tiago Ramalho[a], Agnieszka Grabska-Barwinska [a], Demis Hassabis[a], Claudia Clopath[b], Dharshan Kumaran[a], and Raia Hadsell[a]

[a]DeepMind, London, N1C 4AG, United Kingdom
[b]Bioengineering department, Imperial College London, SW7 2AZ, London, United Kingdom

EWC[3]（Elastic Weight Consolidation）

- 受神经系统 Synaptic Consolidation的启发,在简单L2正则的基础上,寻找对原先任务更重要的Weight,并使其更难被改变



EWC[3]（Elastic Weight Consolidation）

- 实际在网络上的改动只在**损失函数**, 更多的是一种Training Skill

$$L(\theta) = \mathrm{L}_B(\theta) + \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2$$

- $L_B(\theta)$表示Task B 的Loss
- λ 表示New Task 与Old Task的重要关系
- 对于Task A,由于我们不能获得真实的分布,采取*Laplace Approximation*的思想,将Task A的后验近似为一个高斯分布,均值由$\theta^*_{A,i}$(Task A 训练完成所得的参数)确定.利用Fisher Information Matrix的对角元衡量Precision
  - Fisher Informat Matrix 衡量样本信息量
  - 该Term的意义是目前的参数分布与原先网络的分布之间的样本互信息量,以L2范数限制,不让其变得过大

EWC – Testing Results

- 测试数据集 Permuted-Mnist
  - 10 task之后,EWC基本不下降
  - 单纯SGD掉到80%

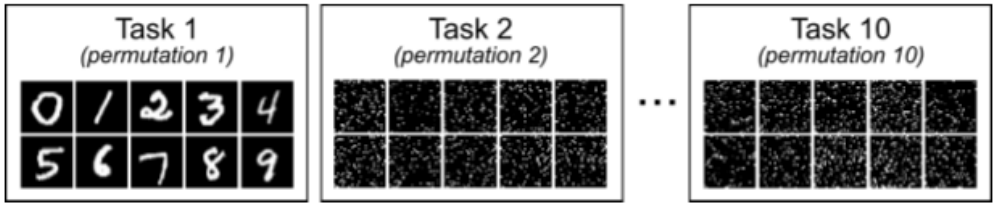
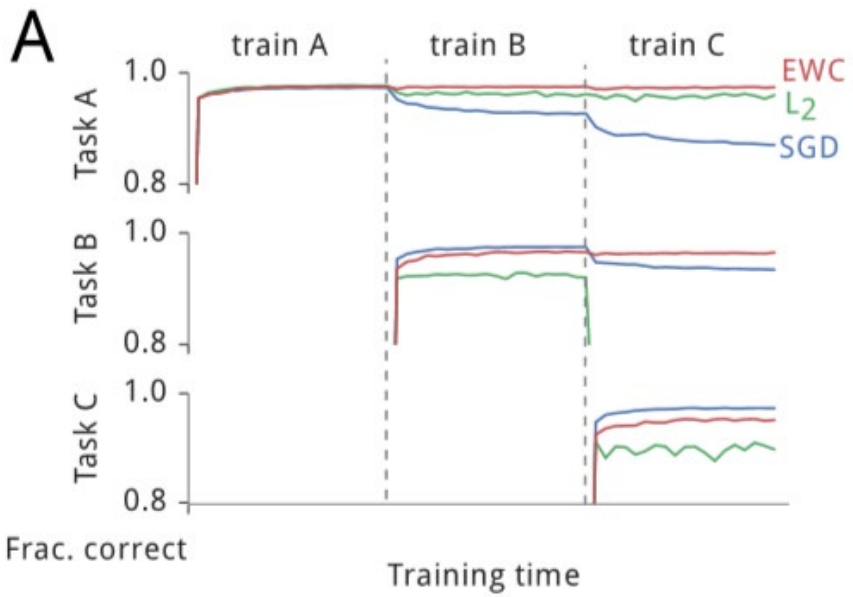
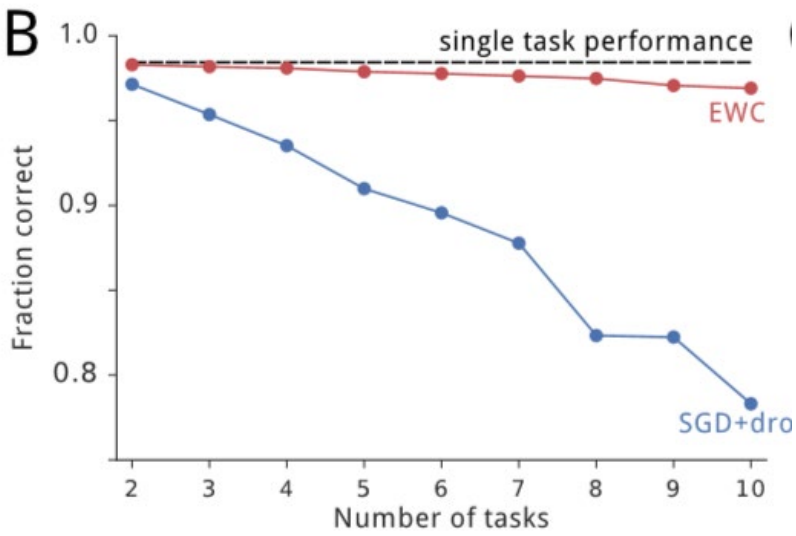


Figure 2: Schematic of permuted MNIST task protocol.

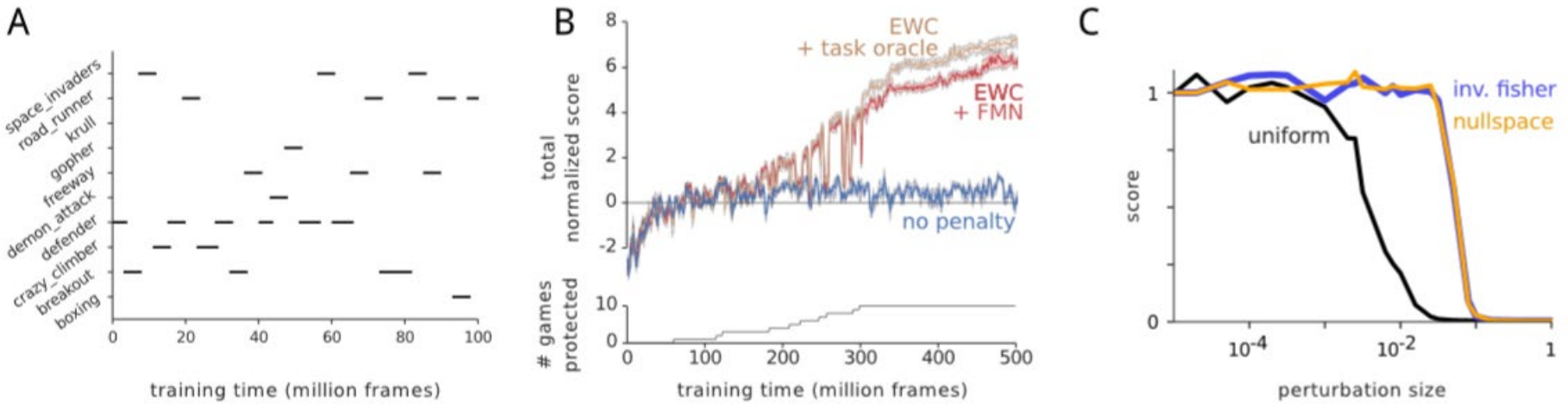
创造出N个与原本Mnist相同难度且互相独立的Task





EWC – Testing Results

- RL
  - 数据集: Atari 2600 task set
  - DQN
  - (与后续研究没有太多的参考,未展开,实现细节见论文)



EWC - Conclusion

- 是一种 Soft,Quadratic Regularization,是一种Training Skill
- 核心是利用NN的Over-Param,去寻找能够处理两个Task的交集
- 说明能够适用于Image Classification与Reinforcement Learning
- 有一定的数理以及生物学基础(模仿了Synaptic Consolidation)

- 另外一篇文章与本文类似(SI – Synaptic Intelligence[6])
- 后续其他研究认为此算法效果不是很好
- 且所有Regularization Based的方法,都要求训练一个比完成单独任务所需要的大得多的网络.

Partial Train

- 与Regularization流派类似的流派 – Partial Train
  - Regularization的思想是以一定的规范化让新网络参数分布不偏离原先网络太远
  - Partial Train通过只训练与新Task更有关部分(或者是剔除对老Task重要的部分)来保持网络的大致参数分布
- 针对EWC中需要处理的两大问题
  - 需要训练一个比完成一个任务大得多的模型(模型冗余)
    - DEN尝试用动态拓展来减少这一部分的冗余度
  - 需要寻找出对任务重要的部分网络
    - 采用Heuristic甚至是一个Sub-Network来学习出网络重要的部分

《Lifelong Learning With Dynamically Expandable Networks》

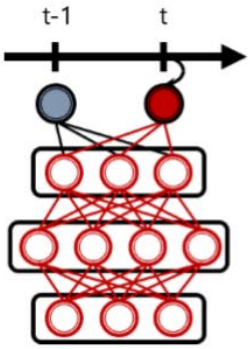
- Cite 63　(ICLR2018)
- South Korea
- 思想与NAS的Structure Estimation相关
- 主要场景在Lifelong Learning:
  - 虽然是作为Incremental Learning的问题处理
  - 但是主要的目的其实在从Old Task中取得有用信息,使新任务
    - 收敛更快
    - 效果更好

LIFELONG LEARNING WITH
DYNAMICALLY EXPANDABLE NETWORKS

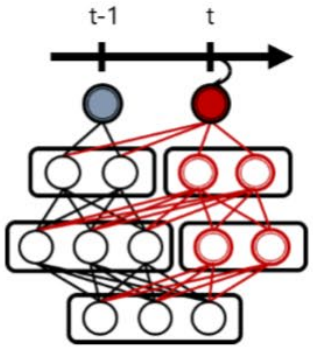
Jaehong Yoon[1,3]*, **Eunho Yang**[1,3], **Jeongtae Lee**[2], **Sung Ju Hwang**[1,3]
KAIST[1], Daejeon, South Korea, UNIST[2], Ulsan, South Korea, AITrics[3], Seoul, South Korea
{jaehong.yoon, eunhoy, sjhwang82}@kaist.ac.kr, jtlee@unist.ac.kr

DEN(Dynamical Expandable Network)

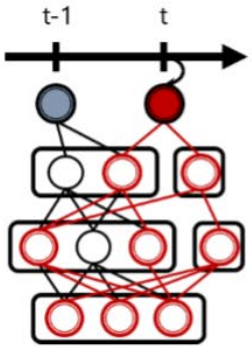
- Related Work
  - EWC (Full Train) 不增加新部分,采用Regularization的方法防止过度偏移原先
  - Progressive (No Train) 不修改原来部分,只增加新部分
  - DNE (Partial Train) 选择性地训练Old Part,并在有需要的时候Expand
  - (理论上还是对每个Task用新的Sub-Net,还是会Share很大一部分)



(a) Retraining w/o expansion



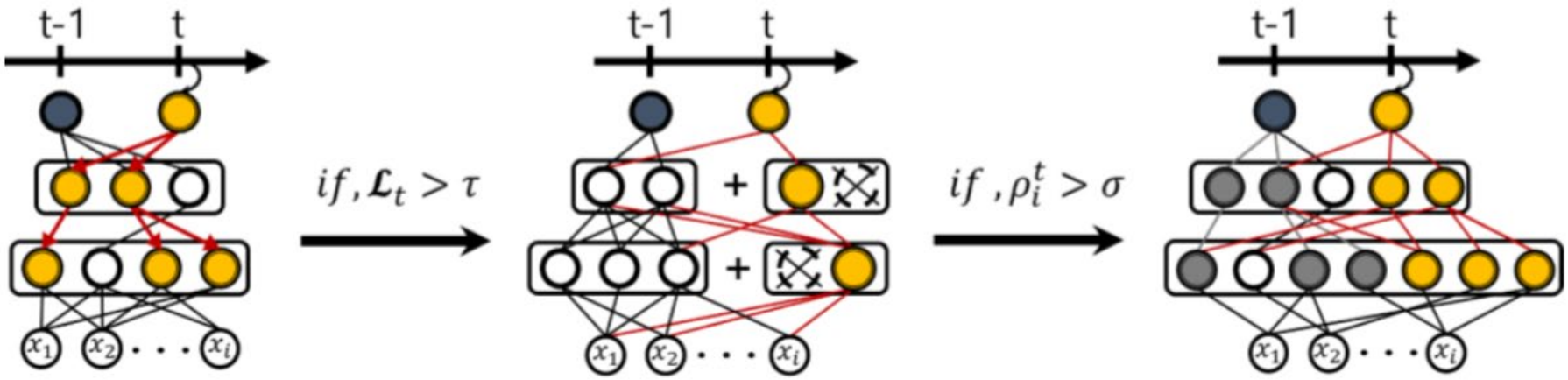
(b) No-retraining w/ expansion



(c) Partial retraining w/ expansion

DEN - Training

- Trainig Method (3 Stage)
  - Background : New Task 的分布和数量都未知,对Old task已知的只有现在模型参数
  - Selective Training
  - Dynamic Expansion
  - Network Split/ Duplication



DEN – Selective Training

- *需要选取New Task相关的Node, 并只训练他们*
- 初始训练时,通过加一个L1 Regularization让初始Weight具有一定稀疏性
- 当新Task来到时,用最后层(TopMost Hidden Units),先Fit一个Sparse Linear Model(用于解决New Task),然后逐下而上找出与输出相连的所有Node, 并训练他们

$$\underset{\boldsymbol{W}^t_{L,t}}{\text{minimize}}\, \mathcal{L}(\boldsymbol{W}^t_{L,t}; \boldsymbol{W}^{t-1}_{1:L-1}, \mathcal{D}_t) + \mu \|\boldsymbol{W}^t_{L,t}\|_1$$

Algorithm 2 Selective Retraining

Input: Datatset $\mathcal{D}_t$, Previous parameter $\boldsymbol{W}^{t-1}$
Output: network parameter $\boldsymbol{W}^t$
Initialize $l \leftarrow L-1, S = \{o_t\}$
Solve Eq. 3 to obtain $\boldsymbol{W}^t_{L,t}$
Add neuron i to S if the weight between i and o_t in $\boldsymbol{W}^t_{L,t}$ is not zero.
for $l = L-1, \ldots, 1$ **do**
  Add neuron i to S if there exists some neuron $j \in S$ such that $\boldsymbol{W}^{t-1}_{l,ij} \neq 0$.
Solve Eq. 4 to obtain $\boldsymbol{W}^t_S$

DEN – Dynamic Expansion

- 如果通过以上的Selctive Training的Loss达不到某一个阈值
- 以Top-Down的顺序逐层添加Node(每层加k个Neuron)同时利用Group Sparsity Regularization去除不需要的Node

$$\underset{\boldsymbol{W}_l^{\mathcal{N}}}{\text{minimize}}\, \mathcal{L}(\boldsymbol{W}_l^{\mathcal{N}}; \boldsymbol{W}_l^{t-1}, \mathcal{D}_t) + \mu\|\boldsymbol{W}_l^{\mathcal{N}}\|_1 + \gamma \sum_g \|\boldsymbol{W}_{l,g}^{\mathcal{N}}\|_2$$

Algorithm 3 Dynamic Network Expansion

Input: Datatset $\mathcal{D}_t$, Threshold τ
Perform Algorithm 2 and compute $\mathcal{L}$
if $\mathcal{L} > \tau$ **then**
  Add k units $\boldsymbol{h}^{\mathcal{N}}$ at all layers
  Solve for Eq. 5 at all layers
for $l = L-1, \ldots, 1$ **do**
  Remove useless units in $\boldsymbol{h}_l^{\mathcal{N}}$

DEN – Network Split/Duplication

- 目的是防止Catastrophic Forgetting或者是Semantic Drift(可以理解为缓慢地CF)
- 计算每个Node相对原先值的Drift, 找到Drift大于一定阈值的,将其Duplicate

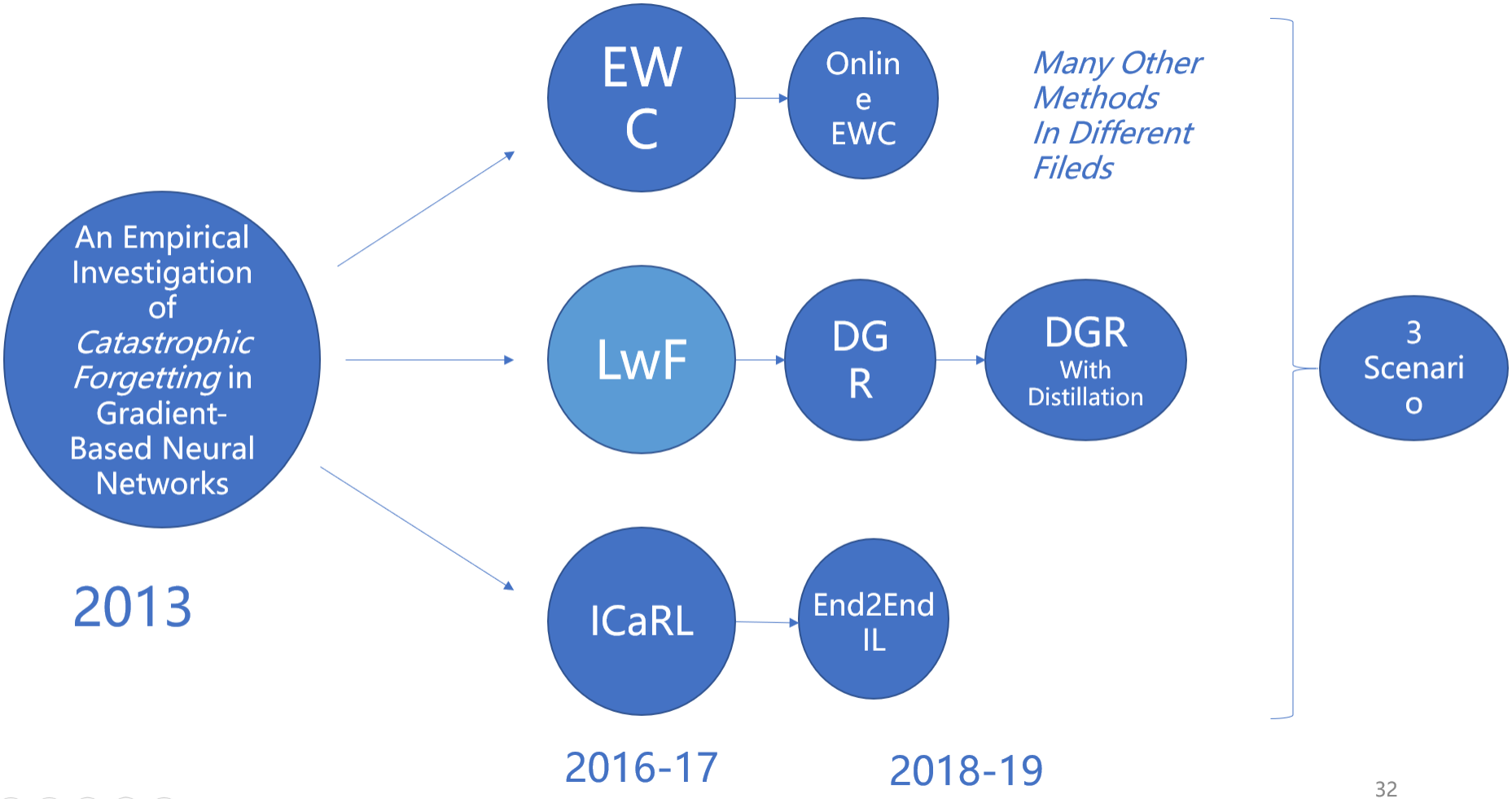
Algorithm 4 Network Split/Duplication

Input: Weight $\boldsymbol{W}^{t-1}$, Threshold σ
Perform Eq. 6 to obtain $\widetilde{\boldsymbol{W}}^t$
for all hidden unit i **do**
  $\rho_i^t = \|\boldsymbol{w}_i^t - \boldsymbol{w}_i^{t-1}\|_2$
  if $\rho_i^t > \sigma$ **then**
    Copy i into i' ($\boldsymbol{w}'$ introduction of edges for i')
Perform Eq. 6 with the initialization of $\widetilde{\boldsymbol{W}}^t$ to obtain $\boldsymbol{W}^t$

Rethinking Regularization

- Partial Train的方法,除了DEN,还有XdG(context dependent gating)以及其他一些采用启发式方法或是Sub-Network[9]进行挑选
- 与Regularization方法思想有共通之处
- 仅局限于Task-IL (需要Task-Specific,与实际应用相对脱节)

Methods


An Empirical Investigation of *Catastrophic Forgetting* in Gradient-Based Neural Networks
2013
EWC
Online EWC
LwF
DGR
DGR With Distillation
ICaRL
End2End IL
2016-17
2018-19
Many Other Methods In Different Fileds
3 Scenario

《Overcoming catastrophic forgetting in neural networks》

- Cite : 294
- CMU
- IEEE Trans. PR / ECCV 2016

Learning without Forgetting

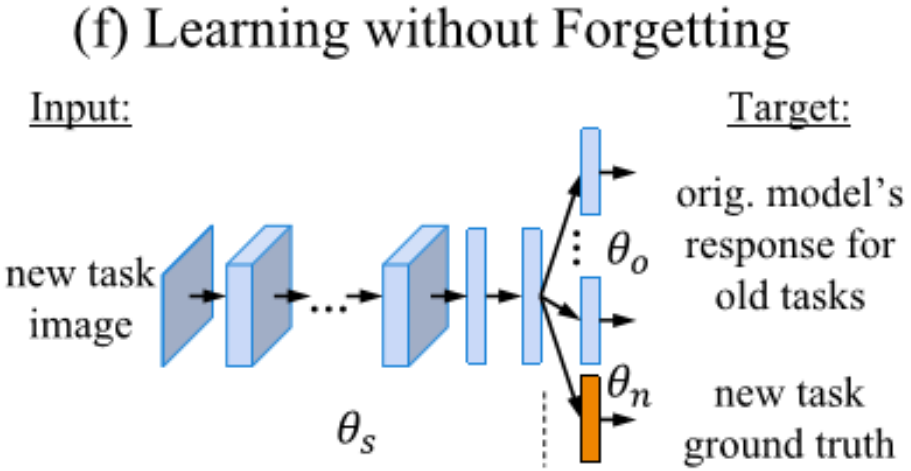
Zhizhong Li and Derek Hoiem

Abstract—When building a unified vision system or gradually adding new apabilities to a system, the usual assumption is that training data for all tasks is always available. However, as the number of tasks grows, storing and retraining on such data becomes infeasible. A new problem arises where we add new capabilities to a Convolutional Neural Network (CNN), but the training data for its existing capabilities are unavailable. We propose our Learning without Forgetting method, which uses only new task data to train the network while preserving the original capabilities. Our method performs favorably compared to commonly used feature extraction and fine-tuning adaption techniques and performs similarly to multitask learning that uses original task data we assume unavailable. A more surprising observation is that Learning without Forgetting may be able to replace fine-tuning with similar old and new task datasets for improved new task performance.

Index Terms—Convolutional neural networks, transfer learning, multi-task learning, deep learning, visual recognition

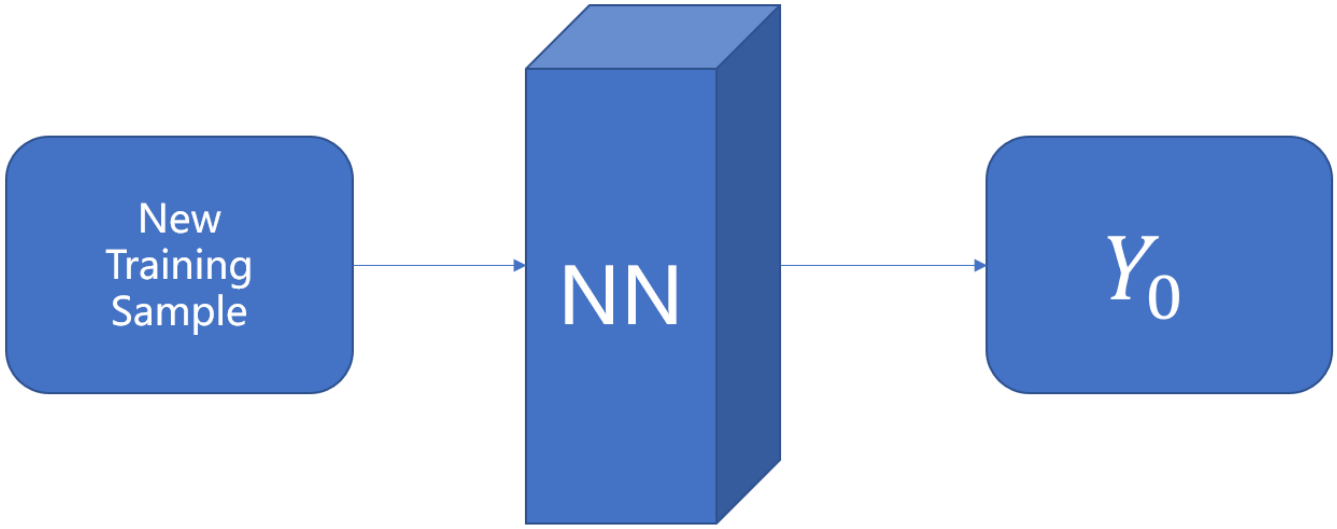
LwF (Learning Without Forgetting)

- 2016年发表,2018年他们又更新了他们的论文,和后来出现的几种方法对比了一下.
- 创新点在于无需Accessible to老数据,也可以完成Incremental Learning(当时来看) – 核心在于**Replay**的思想
- 文章认为: Preserving Old Knowledge By Replay 是一种更好的Regularization

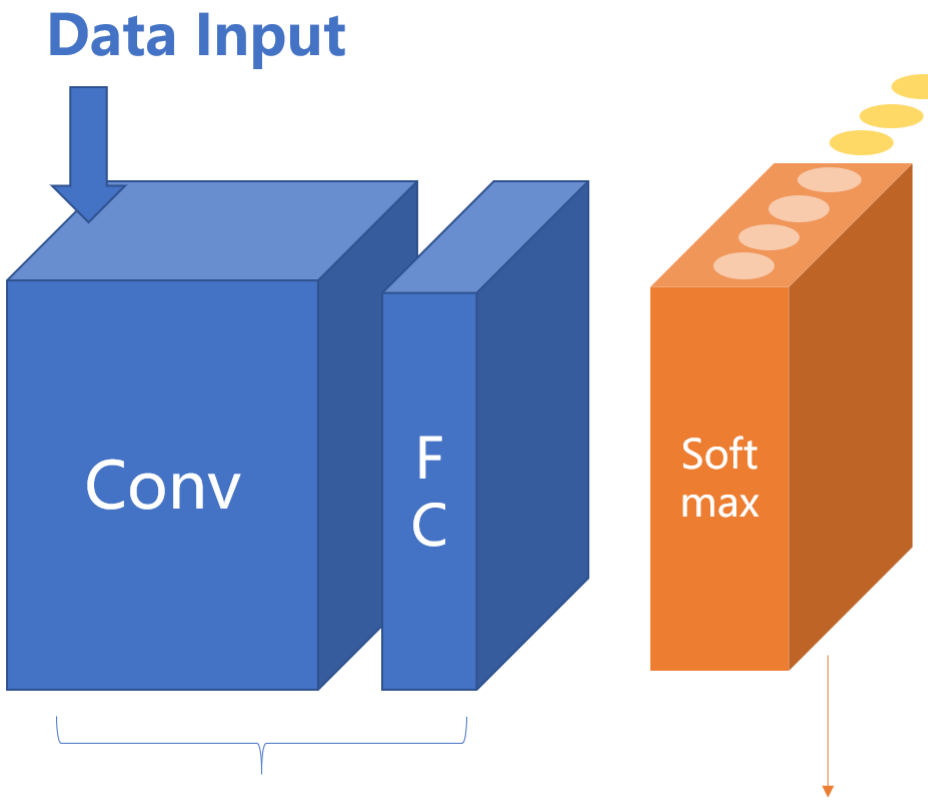


LwF – How To Replay?

- Replay的核心问题: 如何表征Old Knowledge?
  - Pseudo-Data如何产生?



LwF – How To Replay?



LEARNINGWITHOUTFORGETTING:

Start with:

θ_s: shared parameters

θ_o: task specific parameters for each old task

X_n, Y_n: training data and ground truth on the new task

Initialize:

$Y_o \leftarrow \text{CNN}(X_n, \theta_s, \theta_o)$ // *compute output of old tasks for new data*

$\theta_n \leftarrow \text{RANDINIT}(|\theta_n|)$ // *randomly initialize new parameters*

Train:

Define $\hat{Y}_o \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_o)$ // *old task output*

Define $\hat{Y}_n \equiv \text{CNN}(X_n, \hat{\theta}_s, \hat{\theta}_n)$ // *new task output*

$$\theta_s^*,\ \theta_o^*,\ \theta_n^* \leftarrow \underset{\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n}{\text{argmin}} \left(\lambda_o \mathcal{L}_{old}(Y_o, \hat{Y}_o) + \mathcal{L}_{new}(Y_n, \hat{Y}_n) + \mathcal{R}(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n) \right)$$

LwF – How To Replay?

$\check{Y}_N \longrightarrow Y_N$ (新Training Sample对应的标签) **6**

$\check{Y}_O \longrightarrow Y_O$ (刚才在训练开始之前生成的) **4**

分别优化

LwF – How To Replay?

$$\theta_s^*,\ \theta_o^*,\ \theta_n^* \leftarrow \underset{\hat{\theta}_s,\hat{\theta}_o,\hat{\theta}_n}{\operatorname{argmin}} \left(\lambda_o \mathcal{L}_{old}(Y_o, \hat{Y}_o) + \mathcal{L}_{new}(Y_n, \hat{Y}_n) + \mathcal{R}(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n)\right)$$

$$\mathcal{L}_{old}(\mathbf{y}_o, \hat{\mathbf{y}}_o) = -H(\mathbf{y}'_o, \hat{\mathbf{y}}'_o)$$

$$= -\sum_{i=1}^{l} y_o'^{(i)} \log \hat{y}_o'^{(i)},$$

$$y_o'^{(i)} = \frac{(y_o^{(i)})^{1/T}}{\sum_j (y_o^{(j)})^{1/T}}, \quad \hat{y}_o'^{(i)} = \frac{(\hat{y}_o^{(i)})^{1/T}}{\sum_j (\hat{y}_o^{(j)})^{1/T}}.$$

Distillation Loss

$$\mathcal{L}_{new}(\mathbf{y}_n, \hat{\mathbf{y}}_n) = -\mathbf{y}_n \cdot \log \hat{\mathbf{y}}_n,$$

Normal Loss

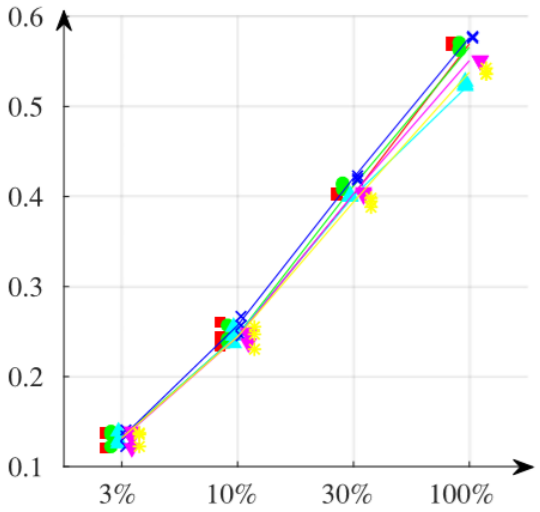
Proposed By Hinton,
Modified CrossEntropy
增大小出现概率的Weight

LwF – Trainig Tricks

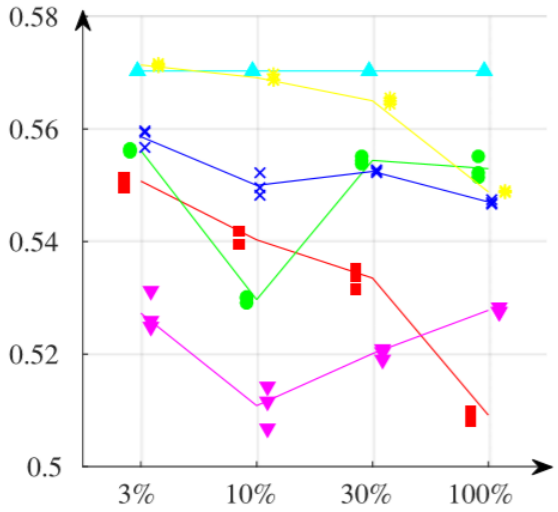
- 2 Stage
  - 首先固定θ_s与θ_o, 只训练θ_n (Warm-Up)
  - 然后一起训练θ_s, θ_o, θ_n

LwF – Experiments

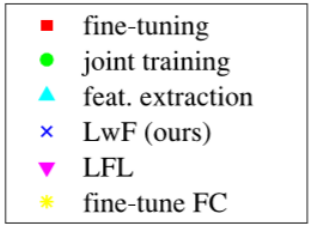
- AlexNet/VGGNet Structure
- Place365, Scenes,VOC,ImageNet 等图像识别数据集(从A到B)
- 比Plain finetune与LFL等其他方法下降慢



(a) CUB accuracy (new)



(b) ImageNet accuracy (old)



LwF – Experiments

- 这个任务本质上属于 Task-IL (给定Task,完成任务)
- 对比的方式主要有几种
  - Finetune : 单纯地减小LR对新Task进行训练,一般是**下限**
    - **Finetune理论上并不能减小CF**
  - Joint Training: 新老数据一起训练,一般是**上限**
    - **多任务时,计算量和存储都会迅速增加,不可行**

Task-IL	With task given, is it the 1^{st} or 2^{nd} class? (e.g., 0 or 1)

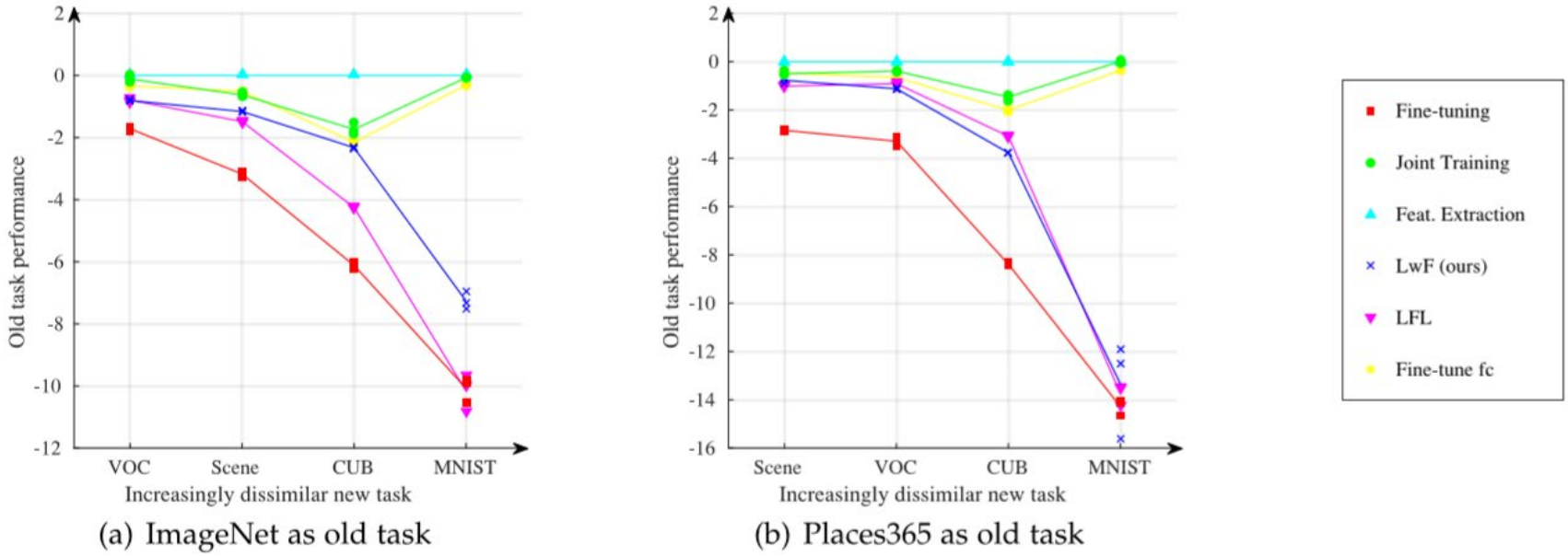
LwF – Conclusion

- 处于上限下限之间,比其他方法 (Less Forgetting Learning,L2 Norm等) 较好

	Fine Tuning	Duplicating and Fine Tuning	Feature Extraction	Joint Training	Learning without Forgetting
new task performance	good	good	X medium	best	best
original task performance	X bad	good	good	good	good
training efficiency	fast	fast	fast	X slow	fast
testing efficiency	fast	X slow	fast	fast	fast
storage requirement	medium	X large	medium	X large	medium
requires previous task data	no	no	no	X yes	no

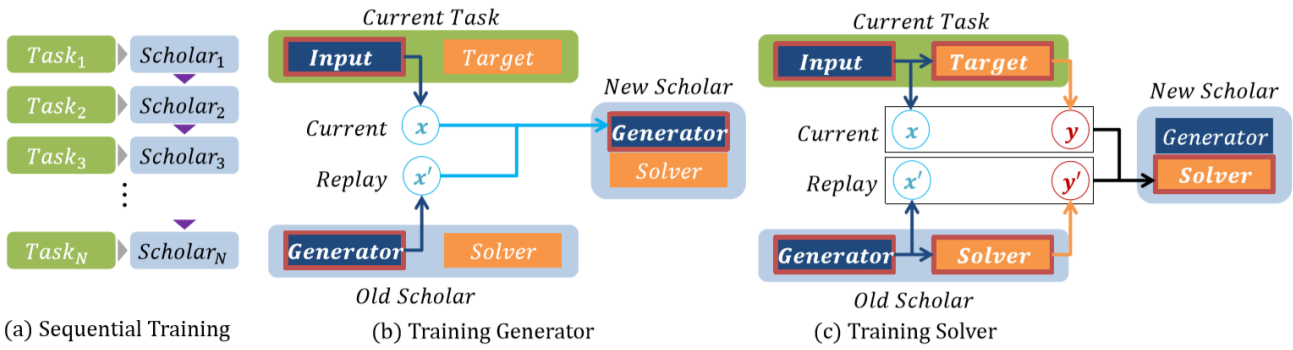
LwF – Conclusion

- 对于差距较大的问题(ImageNet -> Mnits) 除了Joint Training之外效果都比较差



LwF – Conclusion

- 作者认为保存 Old Task Knowledge是一种更为Advanced的Regularization方式(不仅针对Weight规范化,而是规范化了Output)
- 开创了利用Pseudo-Data (Replay机制)来改善Catastrophic Forgetting的方法,缺点在于Data Representation还不是很Elegant
  - 后续对其改进: Deep Generative Replay
    - (用GAN/VAC来生成Pseudo Data)



LwF – Conclusion

- 该方法不能处理连续变化的Domain(比如从Top-View变为Front-View)
- 该方法需要所有的新训练数据Present,不能处理A Stream Of Data

  (严格来说大部分NN的IL方法都不能处理Streaming Data)
- 方法相对Native,后续研究表示这种方式在其他情景下效果也不是很好

《Continual Learning with Deep Generative Replay》

- Cite： 70+
- MIT Paper
- 利用一个GAN/VAC来Preserve Old Knowledge

Continual Learning with Deep Generative Replay

Hanul Shin
Massachusetts Institute of Technology
SK T-Brain
skyshin@mit.edu

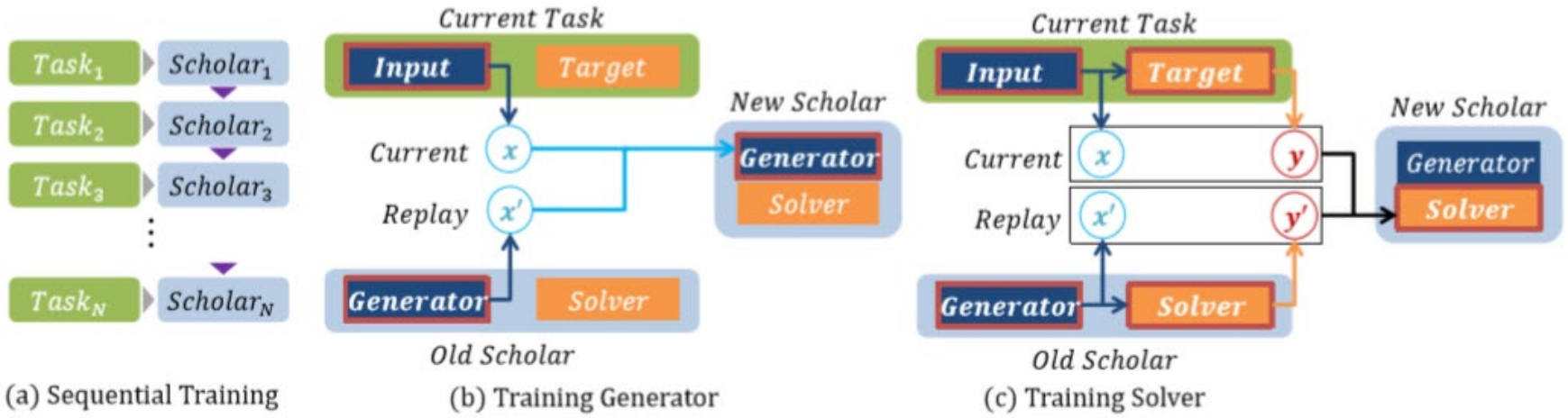
Jung Kwon Lee,* Jaehong Kim,* Jiwon Kim
SK T-Brain
{jklee,xhark,jk}@sktbrain.com

DGR intro

- 模型分为两大部分: Generator & Solver
- 灵感同样源于生物学, Episodic Memory System (Short-Term-Memory)
- Retain Old Knowledge By Generative Model
- 利用GAN/VAC 来学习复杂输入数据(Image)的表征

DGR(Deep Generative Replay)

- 处理 A Stream Of Task,每个T_i目的是去针对某个数据分布D_i优化模型,将训练好的模型叫做一个Scholar[*] (每个Task对应一个)



[*]: 这里的Scholar和Teacher-Student Network结构中的Scholar不一样,后者不能同时Teach & Learn

DGR(Deep Generative Replay)

- 一个Scholar H 是一个Tuple [G,S]
  - Generator : 产生类似真实的Sample
  - Solver: 解决对应问题
    - 需要解决Task Sequence里的所有Task,Loss是所有Task的Loss之和
    - 当训练Task T_i时, 采用Di中抽取的Sample
- 训练分两个Stage: (Training A New Scholar from An Old Scholar)
  - New G 接受New Task的输入x,同时从Old G,获得Fake Old Data x`
    - 两种数据的比例可以表征新老任务的重要性
  - New S: 从Real New Data的Fake Old Data的混合中学习映射
    - Fake Old Data的Label是经过Old Scholar的S的结果
- Generator 可以是任意的Deep Genertaive Model
- Solver依据需要解决的问题确定

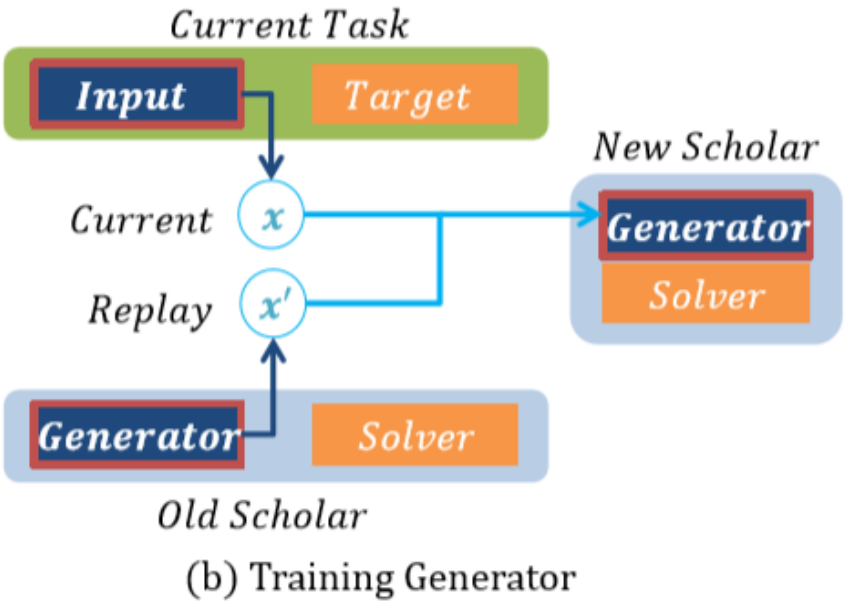
DGR(Deep Generative Replay)

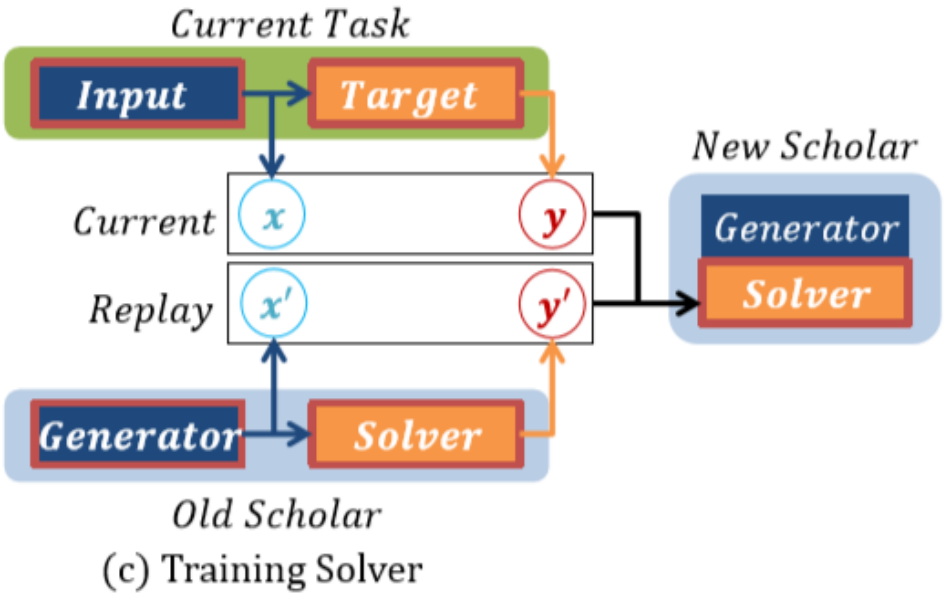
- Training Loss

$$L_{train}(\theta_i) = r\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_i}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})] + (1-r)\,\mathbb{E}_{\boldsymbol{x}'\sim G_{i-1}}[L(S(\boldsymbol{x}';\theta_i),S(\boldsymbol{x}';\theta_{i-1}))]$$

- Test Loss

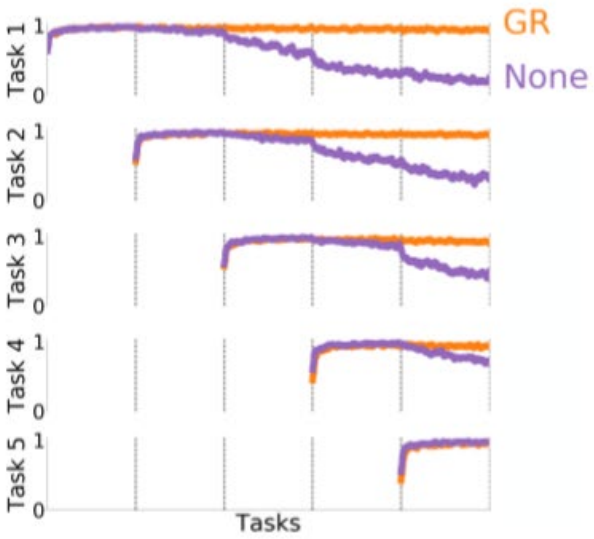
$$L_{test}(\theta_i) = r\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_i}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})] + (1-r)\,\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_{past}}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})]$$



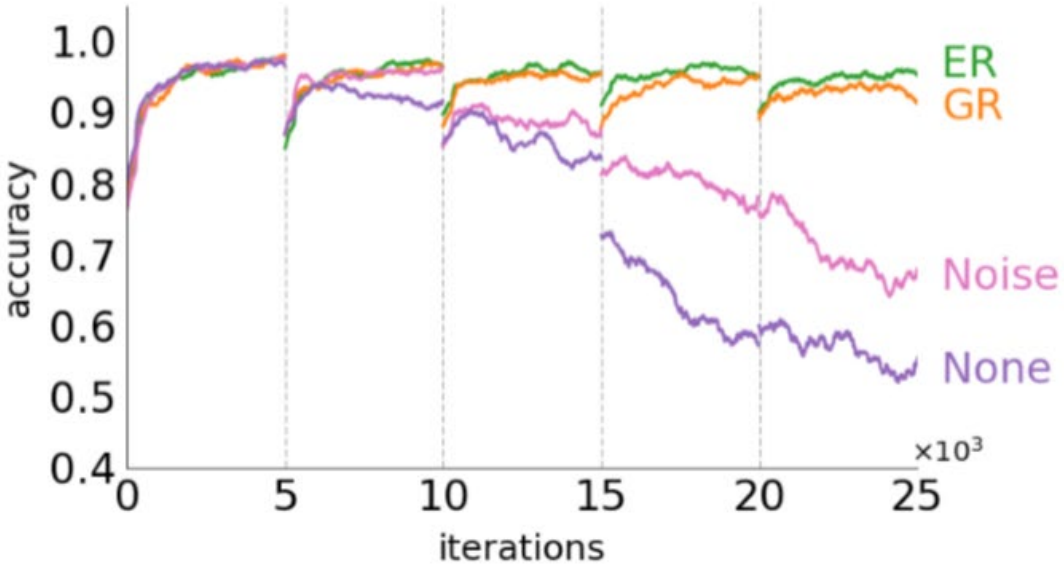


DGR - Experiment

- Permuted Mnist



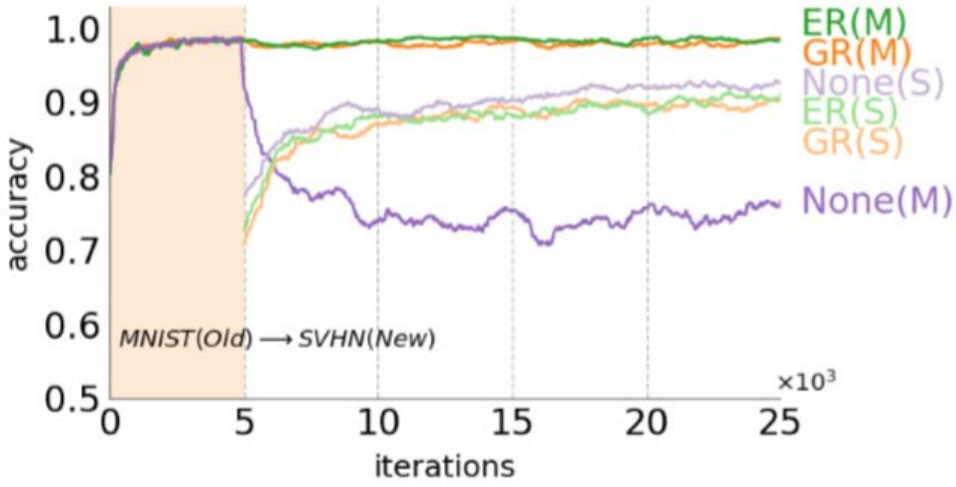
(a)



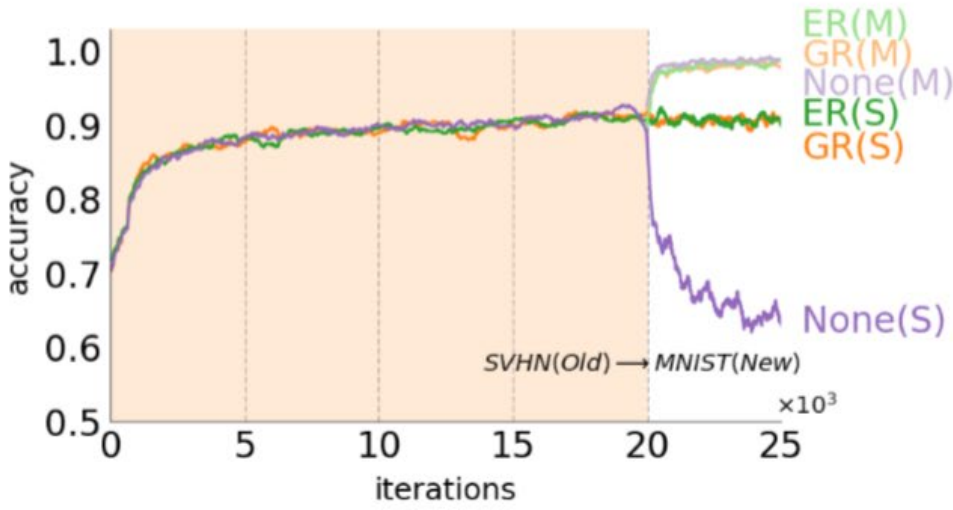
(b)

DGR - Experiment

- Crossin’ Domain : Mnist – SVHN
- **Class IL**: 无需Task-Specific (LwF Can’t)



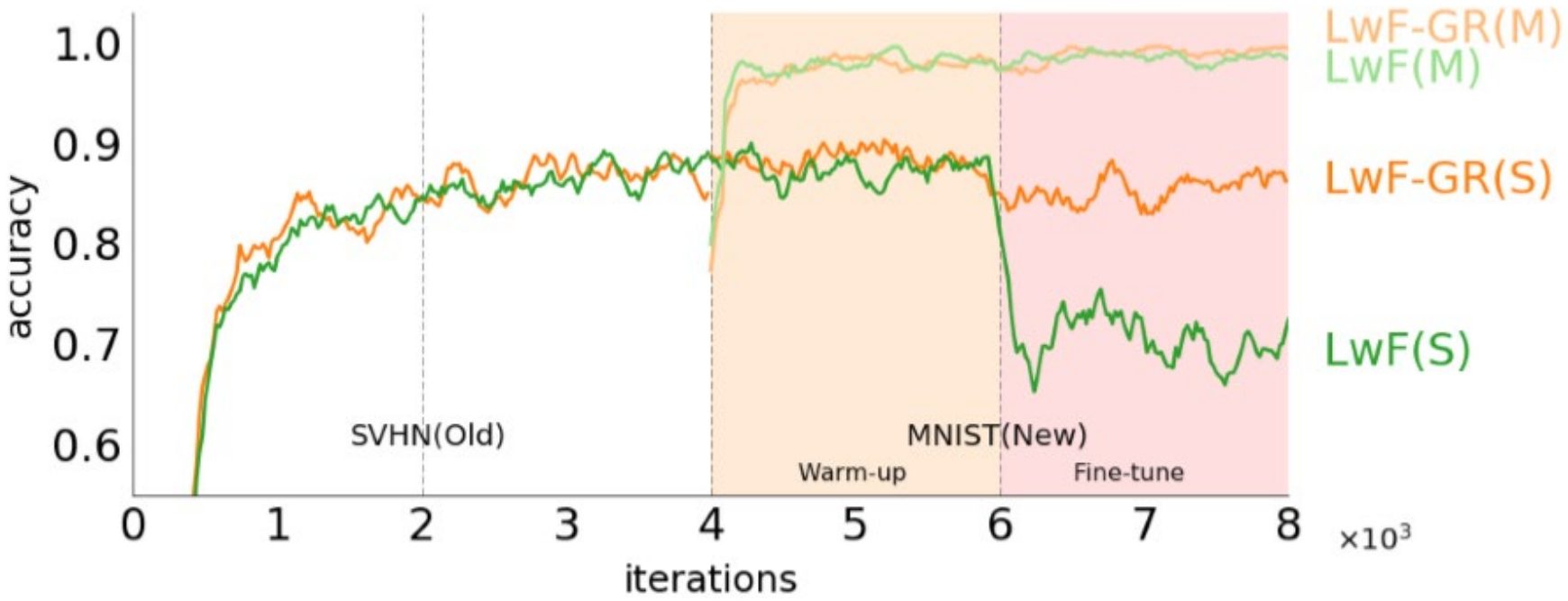
(a) MNIST to SVHN



(b) SVHN to MNIST

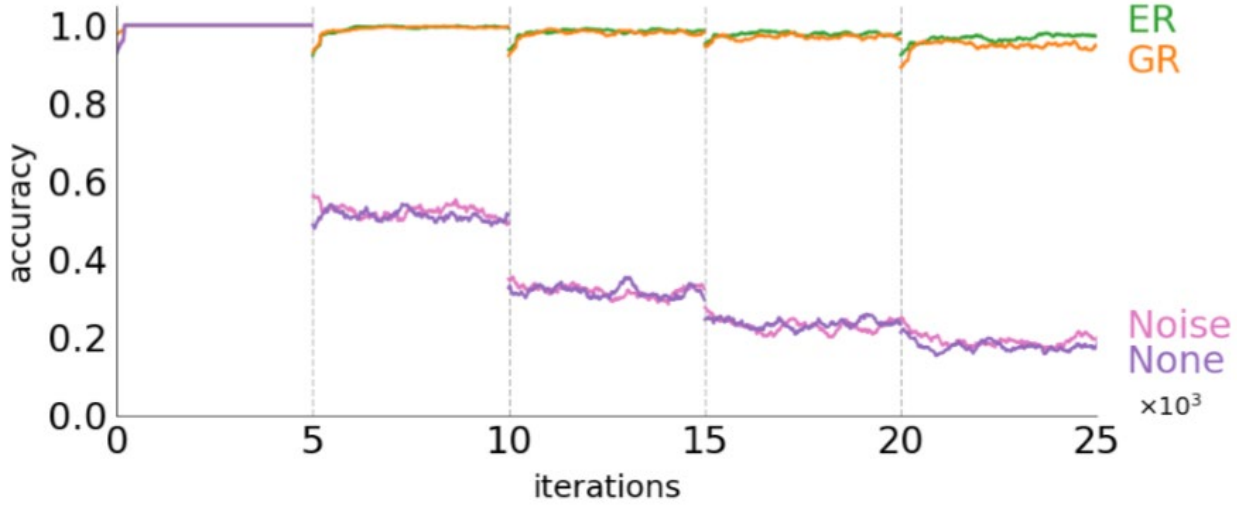
DGR - Experiment

- 作者表示Generative Replay的方式也可用于Enhance其他算法,比如LwF
- 由于LwF输出Task-specific,所以做了分别测试



DGR - Experiment

- **Class IL:** Split Mnist



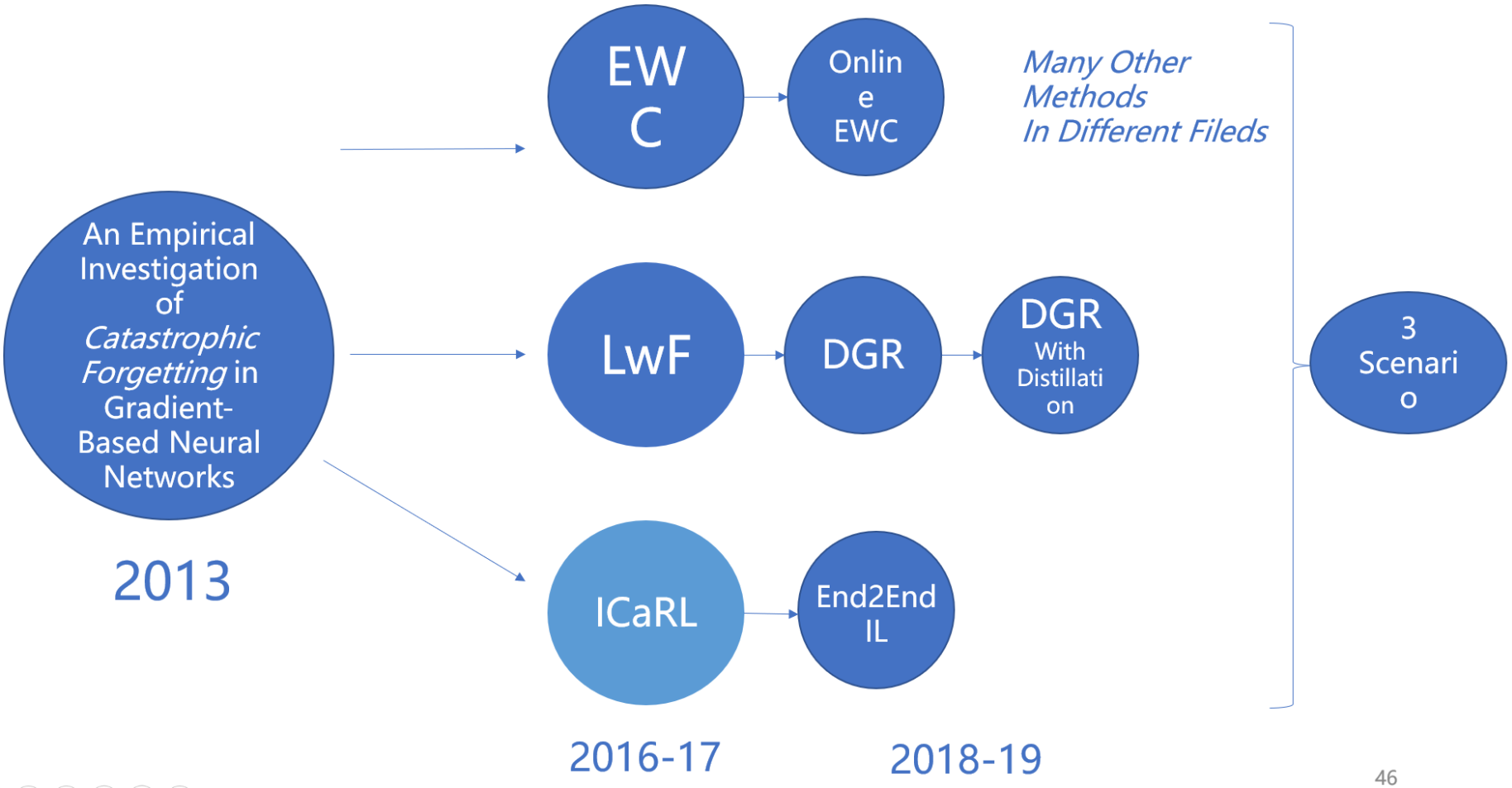
DGR(Deep Generative Replay)

- 作者认为,DGR的性能主要受Generative Model的限制,只要选择够好,理论上等价于Joint Training.
- (?) 实验的数据集较小.对于复杂数据(ImageNet...)要取得一定的表征能力,GAN是否会变得非常大?
- 计算量消耗巨大[8]

Conclusion Replay-Based

- 目前主流思想里面比较Work的一种(比Regularization方法结果更好)
- 也是唯一能适用于各种Scenario的方法[1]
- 缺点在于Pseudo-Data的生成需要花费不少额外的计算(GAN)

Methods


An Empirical Investigation of *Catastrophic Forgetting* in Gradient-Based Neural Networks
2013
EWC
Online EWC
LwF
DGR
DGR With Distillation
ICaRL
End2End IL
2016-17
2018-19
Many Other Methods In Different Fileds
3 Scenario

- Cite : 172
- Oxford Univ.
- CVPR 2017

iCaRL: Incremental Classifier and Representation Learning

Sylvestre-Alvise Rebuffi
University of Oxford/IST Austria

Alexander Kolesnikov, Georg Sperl, Christoph H. Lampert
IST Austria

ICaRL (Incremental Classifier & Representive Learning)

- 2017年提出,所采取的方法类似于前NN时代的算法设计(类似于ISVM)
- 文章里还梳理了IL的**三大准则**(较为严格的IL,ICaRL都能满足)
  - 可接受Streaming Data
  - 当前时刻Performance Acceptable
  - 内存有限(Memory Bounded)
- 采用NMC(Nearest Mean Classifier)做分类准则,利用Example-based方法描述数据,利用CNN完成Representive Learning

Inference

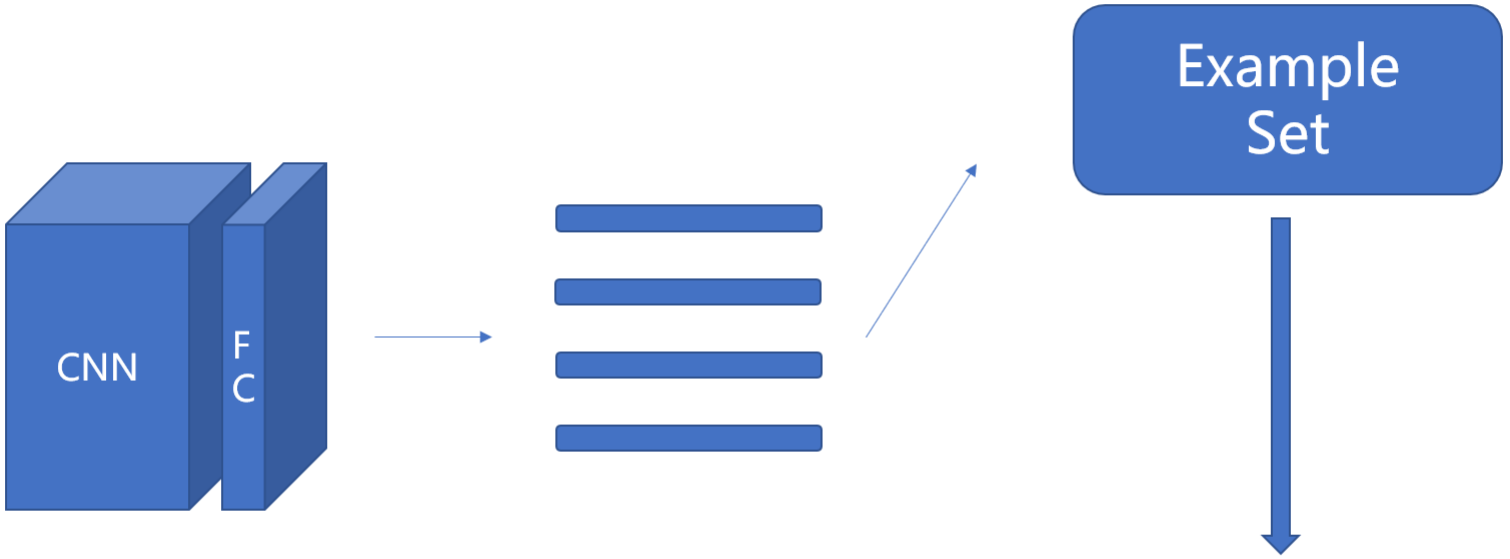
Algorithm 1 iCaRL CLASSIFY

input x // image to be classified
require $\mathcal{P} = (P_1, \ldots, P_t)$ // class exemplar sets
require $\varphi : \mathcal{X} \to \mathbb{R}^d$ // feature map
  for $y = 1, \ldots, t$ **do**
    $\mu_y \leftarrow \frac{1}{|P_y|} \sum_{p \in P_y} \varphi(p)$ // mean-of-exemplars
  end for
  $y^* \leftarrow \underset{y=1,\ldots,t}{\operatorname{argmin}} \|\varphi(x) - \mu_y\|$ // nearest prototype
output class label y^*

NMC (Nearest Mean Classifier)　邻近平均分类器

Inference

一个不断更新的Example Set,存储Weight Vector,按类存储

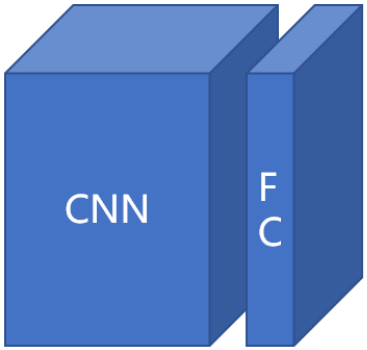


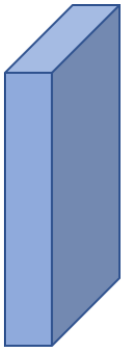
CNN
仅用作完成数据表征,不作分类

将每一Sample表征为Weight Vector(基于Example/Prototype的表示方式)

将新Sample与每个Class对应的Example的平均对比,归类为最接近的那一类
(Nearest Mean Classification)

Training





CNN
仅用作完成数据表征,不作分类

Softmax
(只在训练时候有!)

1st Stage Update Representation

Example Set

Example Set
训练时更新

2nd Stage Update Example Set

2.1 Add

2.2 Reduce

51

Training - 1st Update Representation

Algorithm 3 iCaRL UPDATEREPRESENTATION

input $X^s, \ldots, X^t$ // training images of classes $s, \ldots, t$
require $\mathcal{P} = (P_1, \ldots, P_{s-1})$ // exemplar sets
require Θ // current model parameters
// form combined training set:

$$\mathcal{D} \leftarrow \bigcup_{y=s,\ldots,t} \{(x,y) : x \in X^y\} \cup \bigcup_{y=1,\ldots,s-1} \{(x,y) : x \in P^y\}$$

// store network outputs with pre-update parameters:
for $y = 1, \ldots, s-1$ **do**
$q_i^y \leftarrow g_y(x_i)$ for all $(x_i, \cdot) \in \mathcal{D}$
end for
run network training (*e.g.* BackProp) with loss function

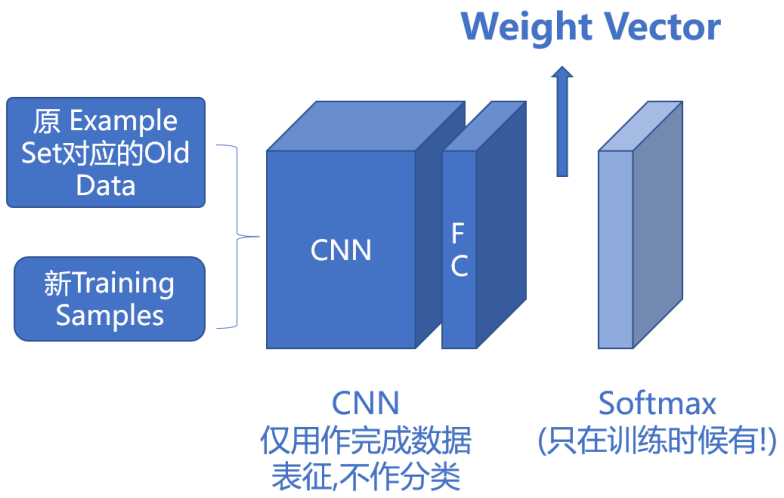
$$\ell(\Theta) = -\sum_{(x_i,y_i)\in\mathcal{D}} \Big[\sum_{y=s}^{t} \delta_{y=y_i} \log g_y(x_i) + \delta_{y\neq y_i} \log(1-g_y(x_i)) + \sum_{y=1}^{s-1} q_i^y \log g_y(x_i) + (1-q_i^y) \log(1-g_y(x_i)) \Big]$$

that consists of *classification* and *distillation* terms.

- 正常的NN训练流程(加入Softmax以满足)

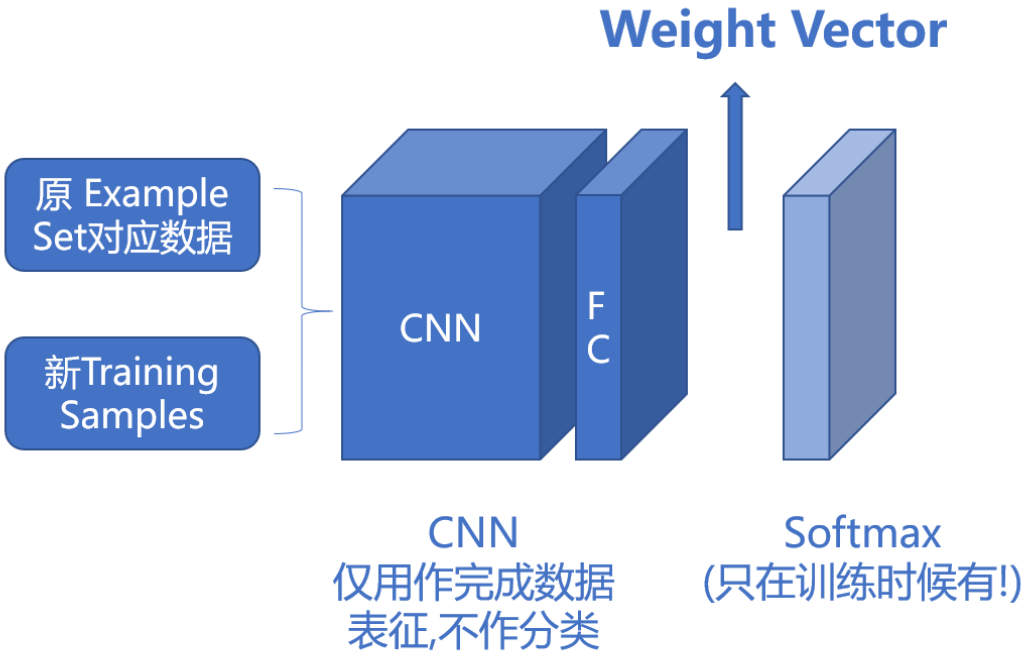
但是在实际推断的时候是没有Softmax的,而是采用NMC的方式完成判别

- Update Representation时不用Example Set,通过Softmax和label计算Loss



Training - 1st Update Representation

$$\ell(\Theta) = -\sum_{(x_i,y_i)\in\mathcal{D}} \Big[\sum_{y=s}^{t} \delta_{y=y_i} \log g_y(x_i) + \delta_{y\neq y_i} \log(1-g_y(x_i)) + \sum_{y=1}^{s-1} q_i^y \log g_y(x_i) + (1-q_i^y) \log(1-g_y(x_i)) \Big]$$



Loss Function同样由
Classification与**Distillation**组成,
要求完成新任务的同时最大程度上
Reproduce Old Task

Training 2nd Update Example Set

Algorithm 4 iCaRL ConstructExemplarSet

input image set $X = \{x_1, \ldots, x_n\}$ of class y
input m target number of exemplars
require current feature function $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$
  $\mu \leftarrow \frac{1}{n} \sum_{x \in X} \varphi(x)$ // current class mean
  for $k = 1, \ldots, m$ **do**
    $p_k \leftarrow \underset{x \in X}{\operatorname{argmin}} \left\| \mu - \frac{1}{k} [\varphi(x) + \sum_{j=1}^{k-1} \varphi(p_j)] \right\|$
  end for
  $P \leftarrow (p_1, \ldots, p_m)$
output exemplar set P

Algorithm 5 iCaRL ReduceExemplarSet

input m // target number of exemplars
input $P = (p_1, \ldots, p_{|P|})$ // current exemplar set
  $P \leftarrow (p_1, \ldots, p_m)$ // *i.e.* keep only first m
output exemplar set P

Training 2nd Update Example Set

如何扩充Example Set?

The one that causes the **average feature vector** over all **exemplars** to best approximate the average feature vector over **all training examples**

逐类别对Example Set为空开始,Iteratively,每次加一个最符合以上标准的入集直到Example Set中有m个为止.

使当前的Example Set最贴合当前所有数据的分布

Training 2nd Update Example Set

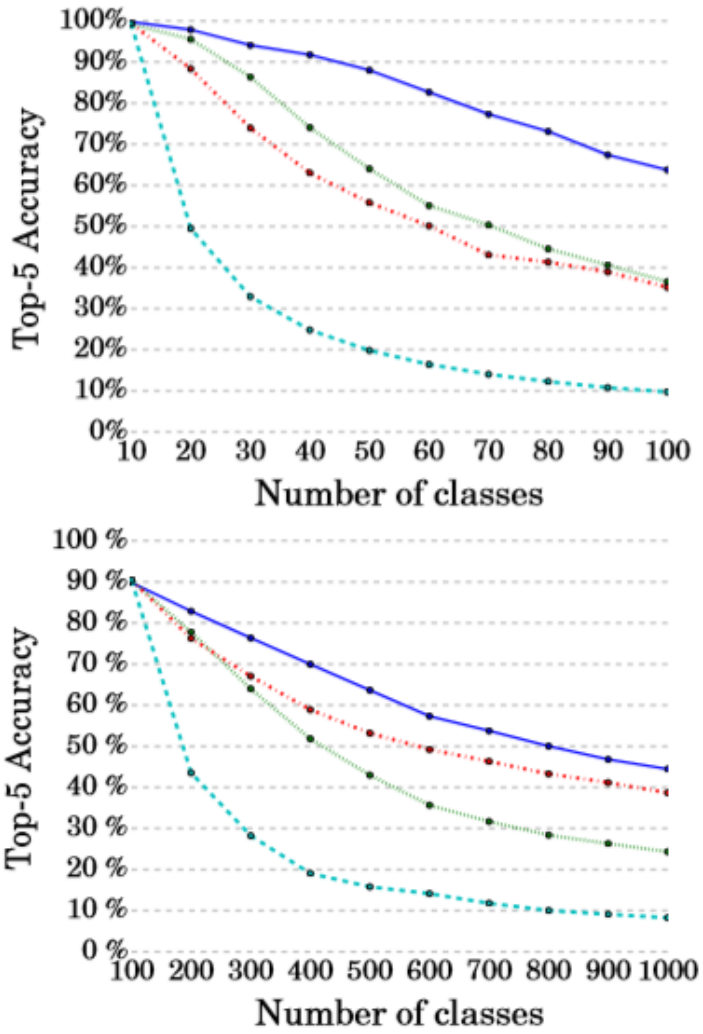
Example Set

如何减小Example Set?

之前Iterative的方式能证明ExampleSet中的Example是以重要性排序的,直接舍去末尾的Example即可

ICaRL-Experiment

- 32Layer ResNet
- Class-IL: 不断添加新类别
- Cifar 100
  - 累加到100类仍然有70%+ 准确率
- ImageNet
  - 累加到1000类有50 % 的准确率



ICaRL-Conclusion

- 借鉴了原先一些ML的方法,与NN相关的只是拿CNN做了Data Representation,解决了传统的NMC难以处理非线性的问题
- 做到了真正意义上的Incremental Learning(定义比较严谨)
  - 特别是能处理Streaming的数据
- 文章中作者也提及了: Example 的方式也是一种Replay

Other Problems

- 在后DL时代,人们主要研究方向在于缓解NN中的CF的问题,而对其他一些问题没有成熟的专门化研究,在ML时代的研究中涉及到:
  - Concept Drift (输入数据的分布和实际需要学习的映射会改变) – 依靠NN的强Robust无视
  - Hyper-Param Setting (不知道完整数据分布时难以确定训练超参数) – 用Grid Search确定
  - Memory Bottleneck – 针对NN还没有具体落地到硬件…所以只要不是无限作者都说这是自己优点…

Index

- 概念辨析
- 应用场景
- 发展路线
- **方法介绍与对比**

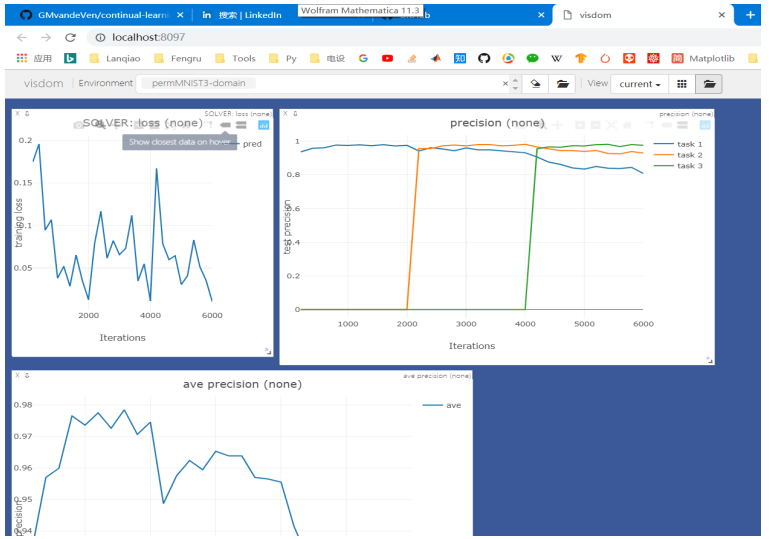
Baseline

- 没有一个明确的评判标准,大家的网络架构不同,数据集不同,Scenario也不同,一把论文中都是提取Finetune/Joint Training为上下界,选取几个类似的方法做一下对比
- 一般不对比**绝对值**,对比**相对趋势**
- 会更改模型大小的方法,较为主流的都与原始**增量不大**

Comparison

- 对现有的多方法进行**对比汇总**比较基本可以发文章了
- 代表文章: 3 Scenario
- 将多方法针对多情景进行了公平比较,并且开源了实现代码(内附Visdom可视化)

Method	*+ task-ID in output layer*	*+ task-ID in hidden layers*
None	81.79 (± 0.48)	90.41 (± 0.32)
EWC	94.74 (± 0.05)	96.94 (± 0.02)
Online EWC	95.96 (± 0.06)	96.89 (± 0.03)
SI	94.75 (± 0.14)	96.53 (± 0.04)
LwF	69.84 (± 0.46)	84.21 (± 0.48)
DGR	92.52 (± 0.08)	95.31 (± 0.04)
DGR+distill	97.51 (± 0.01)	97.67 (± 0.01)



Comparison

Table 4: Average test accuracy (over all tasks) on the split MNIST task protocol. Each experiment was performed 20 times with different random seeds, reported is the mean (± SEM) over these runs.

Approach	Method	Task-IL	Domain-IL	Class-IL
Baselines	*None – lower bound*	*87.19 (± 0.94)*	*59.21 (± 2.04)*	*19.90 (± 0.02)*
	Offline – upper bound	*99.66 (± 0.02)*	*98.42 (± 0.06)*	*97.94 (± 0.03)*
Task-specific	XdG	99.10 (± 0.08)	-	-
Regularization	EWC	98.64 (± 0.22)	63.95 (± 1.90)	20.01 (± 0.06)
	Online EWC	99.12 (± 0.11)	64.32 (± 1.90)	19.96 (± 0.07)
	SI	99.09 (± 0.15)	65.36 (± 1.57)	19.99 (± 0.06)
Replay	LwF	99.57 (± 0.02)	71.50 (± 1.63)	23.85 (± 0.44)
	DGR	99.50 (± 0.03)	95.72 (± 0.25)	90.79 (± 0.41)
	DGR+distill	99.61 (± 0.02)	96.83 (± 0.20)	91.79 (± 0.32)
Replay + Exemplars	iCaRL (budget = 2000)	-	-	94.57 (± 0.11)

Comparison

Table 5: Idem as Table 4, except on the permuted MNIST task protocol.

Approach	Method	Task-IL	Domain-IL	Class-IL
Baselines	*None – lower bound*	*81.79 (± 0.48)*	*78.51 (± 0.24)*	*17.26 (± 0.19)*
	Offline – upper bound	97.68 (± 0.01)	97.59 (± 0.01)	97.59 (± 0.02)
Task-specific	XdG	91.40 (± 0.23)	-	-
Regularization	EWC	94.74 (± 0.05)	94.31 (± 0.11)	25.04 (± 0.50)
	Online EWC	95.96 (± 0.06)	94.42 (± 0.13)	33.88 (± 0.49)
	SI	94.75 (± 0.14)	95.33 (± 0.11)	29.31 (± 0.62)
Replay	LwF	69.84 (± 0.46)	72.64 (± 0.52)	22.64 (± 0.23)
	DGR	92.52 (± 0.08)	95.09 (± 0.04)	92.19 (± 0.09)
	DGR+distill	97.51 (± 0.01)	97.35 (± 0.02)	96.38 (± 0.03)
Replay + Exemplars	iCaRL (budget = 2000)	-	-	94.85 (± 0.03)

Comparison

- 上述比较在Split/Permuted Mnist上,Task相对较简单,对原先在ImageNet这样大数据上的模型不太公平。
- 由于任务简单，上述绝对精度差距较小时的参考价值较小
- 实际上证明的是： 目前除了DGR，其他模型基本很难应用于多个Scneario —— 基本还是针对Scenario选择方法
- 目前几个主流方法，对于最难的Class IL，仅有ICaRL和DGR能取得较好效果

Experiment Environment

DataSet	Paper	Network Structure
Split & Permu MNIST[*]	3 Scenario	MLP(2 Layer fc)
MNIST	EWC	MLP(6 Layer)
SVNH	DGR	MLP + VAC
IMAGENET & CIfar100	ICaRL	34 Layer ResNet
IMAGENET & PASCAL VOC & Place365	LwF	AlexNet
Cifar 100	DEN	Modified AlexNet

[*] 当数据集为Permuted MNIST时,不用CNN(数据集破坏了空间特征)

Comparison — DataSet

- 上述方法中，有一些方法仅在小数据集上进行了试验,而在大数据集上的准确率不能保证
  - EWC : 后续研究证明该方式不能特别好的泛化
  - DGR : 对于大数据集,虽然能够Handle,但是其Generative Model的计算存储开销都会大幅提升
- 目前已经在ImageNet等数据集上证明准确率的有
  - DEN, LwF, ICaRL

Comparison — Model Size

- 上述方法中,大部分都不会对原本的模型大小有太大改变,基本大小相当于解决问题所需要的模型(比如ImageNet分类任务用AlexNet或者是ResNet)
  - ICaRL(除了Example Set会引入额外的存储开销), LwF
- 有几个Exception
  - DEN: 会对网络进行拓展 (论文中未提及如果训练出现失误而导致网络大小一直增长到很大才收敛的情况)
  - DGR: 引入了一个额外生成模型,在大数据集时会增大很多开销
  - EWC: 初始需要训练一个比原先网络大得多的模型

Comparison — Computation

- 相比于原本的模型
  - EWC在计算Fisher Information Matrix时需要引入许多额外计算
  - ICaRL最终判别是NMC,计算量相对较小
  - DEN引入的额外计算量较小
  - DGR需要额外训练一个GAN
  - LwF没有太多的额外计算量

Comparison — Accuracy

Table 4: Average test accuracy (over all tasks) on the split MNIST task protocol. Each experiment was performed 20 times with different random seeds, reported is the mean (± SEM) over these runs.

Approach	Method	Task-IL	Domain-IL	Class-IL
Baselines	*None – lower bound*	*87.19 (± 0.94)*	*59.21 (± 2.04)*	*19.90 (± 0.02)*
	Offline – upper bound	*99.66 (± 0.02)*	*98.42 (± 0.06)*	*97.94 (± 0.03)*
Task-specific	XdG	99.10 (± 0.08)	-	-
Regularization	EWC	98.64 (± 0.22)	63.95 (± 1.90)	20.01 (± 0.06)
	Online EWC	99.12 (± 0.11)	64.32 (± 1.90)	19.96 (± 0.07)
	SI	99.09 (± 0.15)	65.36 (± 1.57)	19.99 (± 0.06)
Replay	LwF	99.57 (± 0.02)	71.50 (± 1.63)	23.85 (± 0.44)
	DGR	99.50 (± 0.03)	95.72 (± 0.25)	90.79 (± 0.41)
	DGR+distill	99.61 (± 0.02)	96.83 (± 0.20)	91.79 (± 0.32)
Replay + Exemplars	iCaRL (budget = 2000)	-	-	94.57 (± 0.11)

Comparison — Others

- Convergent Time
  - 没有太多的模型在论文中提及自己训练的收敛步数,目前不太能够量化
- Generalization
  - 能表现泛化能力的目前仅有Task IL(算法设计上往往有Task Specific的部分)
  - 相关文章也表示当新的DataSet与原本差距大的时候效果差
  - 目前仅有用Gan的方法能够做到较好的泛化性能
- Hyper-Param Setting
  - 目前的文章基本没有在这方面做工作

Reference

[1] Three scenarios for continual learning(Not Published, Only On Arxiv)

[2] Goodfellow, Ian J., et al. "An empirical investigation of catastrophic forgetting in gradient-based neural networks." *arXiv preprint arXiv:1312.6211* (2013).

[3] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." *IEEE transactions on pattern analysis and machine intelligence* 40.12 (2017): 2935-2947

[4] Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *Proceedings of the national academy of sciences*114.13 (2017): 3521-3526.

[5] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

Reference

[6] Zenke, Friedemann, Ben Poole, and Surya Ganguli. "Continual learning through synaptic intelligence." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

[7] Yoon, Jaehong, et al. "Lifelong learning with dynamically expandable networks." arXiv preprint arXiv:1708.01547 (2017).

[8] van de Ven, Gido M., and Andreas S. Tolias. "Generative replay with feedback connections as a general strategy for continual learning." *arXiv preprint arXiv:1809.10635* (2018).

[9] Oswald, J.V., Henning, C., Sacramento, J., & Grewe, B.F. (2019). Continual learning with hypernetworks. ArXiv, abs/1906.00695.

Reference

[10] Zhang, Mengmi, et al. "Prototype Reminding for Continual Learning." arXiv preprint arXiv:1905.09447 (2019).

[11] Hsu, Yen-Chang et al. “Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines.” ArXiv abs/1810.12488 (2018): n. pag.