Few-Shot Lifelong Learning: A Tiny Survey

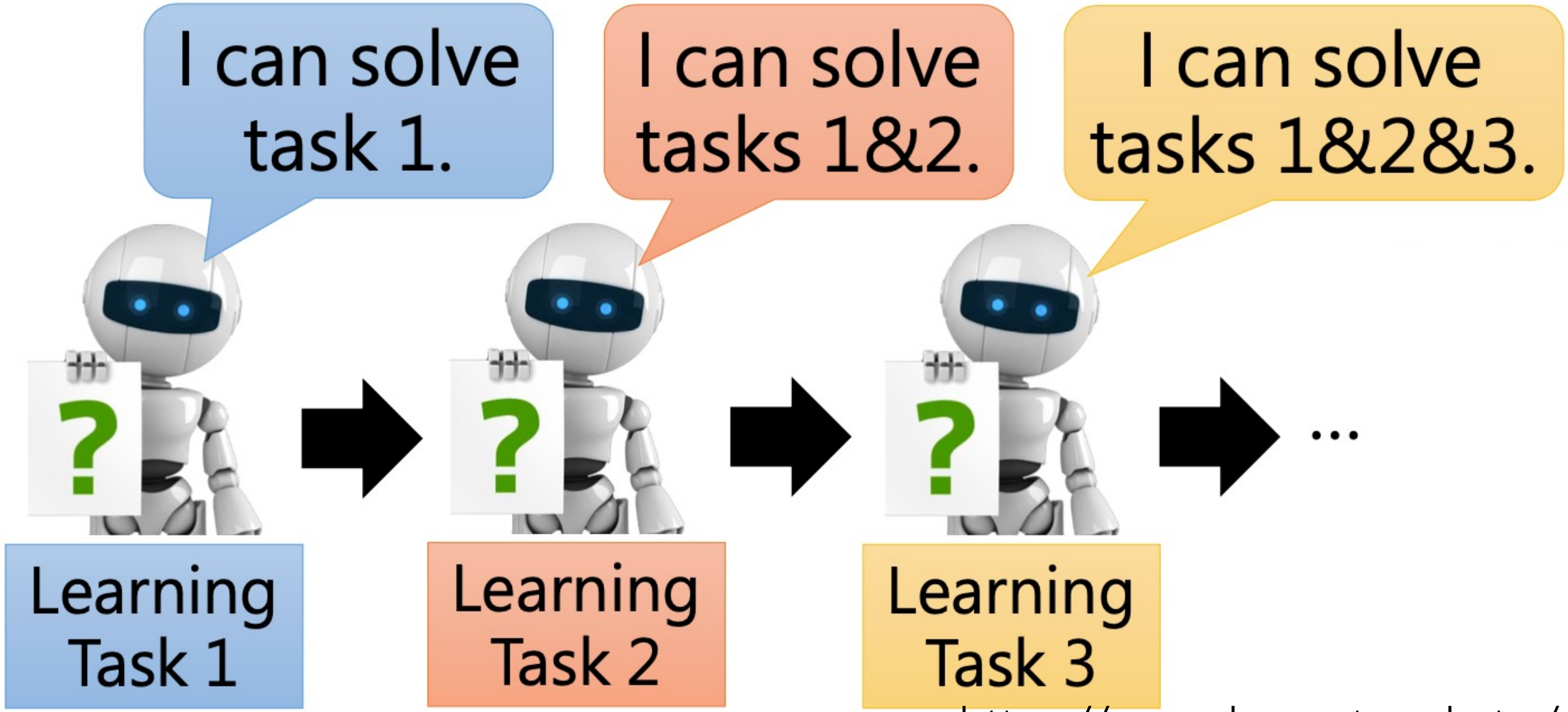
By 李想

2021.5.28

Background: Lifelong learning 终身学习

- **Lifelong learning** (Incremental learning, Continual Learning)：

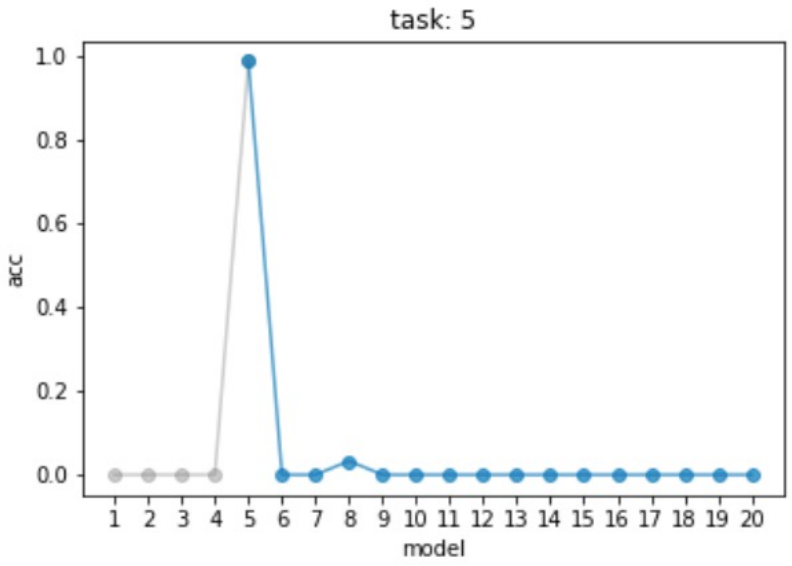
  The ability to sequentially learn new tasks without forgetting previous ones.



https://speech.ee.ntu.edu.tw/~hylee/ml/2020-spring.html

Background: Lifelong learning

- **Challenge:**
- Knowledge Retention: 不要遗忘过去所学
    - 捡了芝麻丢西瓜
- Knowledge Transfer: 知识迁移
- Model Expansion: 模型扩展



Catastrophic Forgetting

- **Multi-task *vs* Lifelong Learning** :
- Computation issue: 训练需要所有数据（e.g. 1000个task)
- Storage issue: 保存所有数据

Background: Lifelong learning

- **Evaluation**

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

Backward Transfer =

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

Forward Transfer =

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

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

- 由于遗忘，BWT一般为负，很少为正，越大越好

Background: Few-Shot Learning 少样本学习

- Learning from a small (single) number of labeled data points
- 对于测试中新的class，无需大量标注数据对模型进行重新训练，而是利用少量（几个）带标签数据使模型迅速适应到新的类别特征分类中。

- Mostly, using **Meta Learning:** Learn to learn
  - Metric-based: Relation network, Prototypical network, Induction network……
  - Optimization-based: MAML……

Dynamic Few-Shot Visual Learning without Forgetting

CVPR 2018

From few-shot learning point of view
Where the story begin……

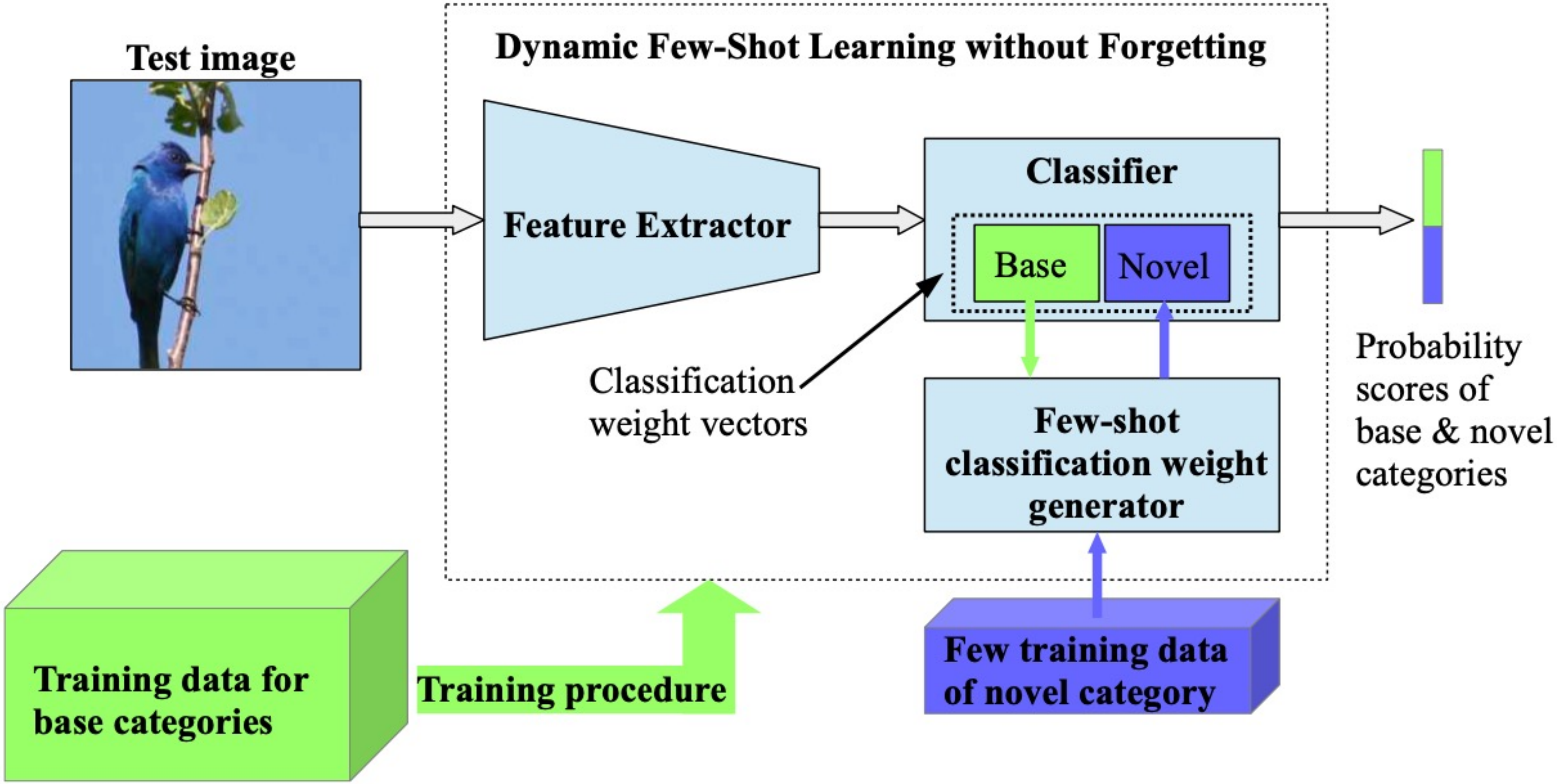
Introduction

- **Motivation**:
  - The learning of the novel categories needs to be fast
  - To not sacrifice any accuracy on the initial categories (不要遗忘)

- **Goal**: Not only is able to recognize base categories, but also learns to dynamically recognize novel categories from only a few training examples (provided only at test time) while also not forgetting the base ones or requiring to be re-trained on them

在动态地利用少样本学习新的category时，不遗忘训练时的category，不需要重新训练

Methodology



Methodology

- **Cosine-similarity based recognition model**
  - Use cosine similarity function between the feature representations and the classification weight vectors to get classification score.

- **Few-shot classification weight generator**
  - Feature averaging based weight inference

$$w'_{avg} = \frac{1}{N'} \sum_{i=1}^{N'} \overline{z}'_i$$

  - Attention-based weight inference: between new category and base category

$$w'_{att} = \frac{1}{N'} \sum_{i=1}^{N'} \sum_{b=1}^{K_{base}} Att(\phi_q \overline{z}'_i, k_b) \cdot \overline{w}_b$$

Experimental Results

Models	5-Shot learning – K_{novel}=5			1-Shot learning – K_{novel}=5		
	Novel	Base	Both	Novel	Base	Both
Matching-Nets [25]	68.87 ± 0.38%	-	-	55.53 ± 0.48%	-	-
Prototypical-Nets [22]	72.67 ± 0.37%	62.10%	32.70%	54.44 ± 0.48%	52.35%	26.68%
Ours						
Cosine Classifier	72.83 ± 0.35%	70.68%	51.89%	54.55 ± 0.44%	70.68%	39.17%
Cosine Classifier & Avg. Weight Gen	74.66 ± 0.35%	70.92%	60.26%	55.33 ± 0.46%	70.45%	48.56%
Cosine Classifier & Att. Weight Gen	**74.92 ± 0.36%**	70.88%	60.50%	**58.55 ± 0.50%**	70.73%	50.50%
Ablations						
Dot Product	64.58 ± 0.38%	63.59%	31.80%	46.09 ± 0.40%	63.59%	24.76%
Dot Product & Avg. Weight Gen	60.30 ± 0.39%	62.15%	46.41%	44.31 ± 0.40%	61.99%	39.05%
Dot Product & Att. Weight Gen	67.81 ± 0.37%	62.11%	48.70%	53.88 ± 0.48%	62.28%	42.41%
Ablations						
Cosine w/ ReLU.	71.04 ± 0.36%	**72.51%**	58.16%	52.91 ± 0.45%	**72.51%**	43.17%
Cosine w/ ReLU. & Avg. Weight Gen	71.30 ± 0.38%	72.47%	59.33%	53.19 ± 0.45%	71.70%	49.53%
Cosine w/ ReLU. & Att. Weight Gen	73.03 ± 0.38%	72.26%	**61.05%**	56.09 ± 0.54%	72.34%	**51.25%**

Few-Shot Lifelong Learning

AAAI 2021

Introduction

- **Motivation**:
  - Many real-world classification problems often have classes with very few labeled training samples —— Few-shot learning
  - All possible classes may not be initially available for training, and may be given incrementally —— Lifelong learning

- **Issues**:
  - Overfitting: Training the entire network on classes with very few samples
  - Catastrophic Forgetting: Model will not have access to old classes when new classes become available for training.

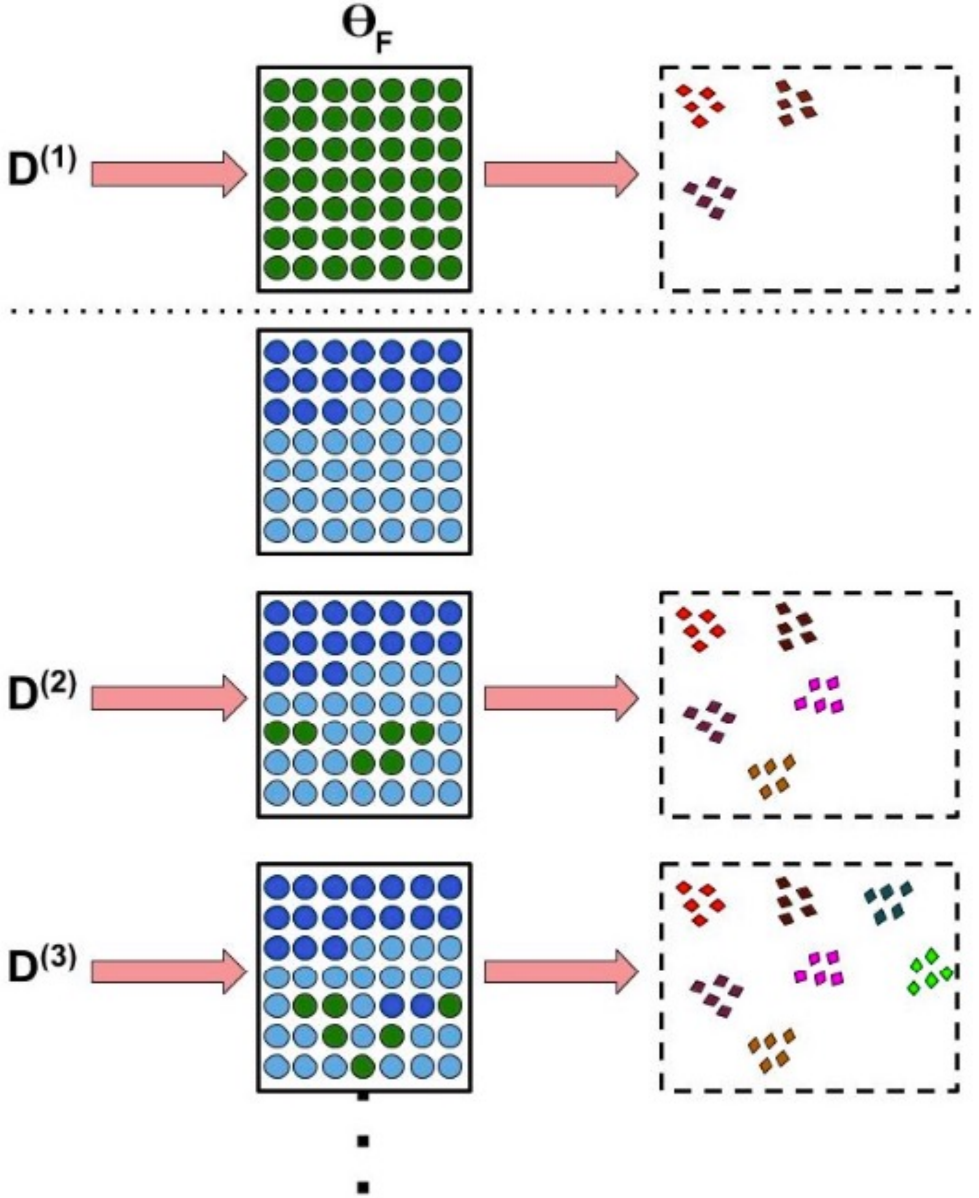
Problem Setting

- A sequence of labeled training sets: $D^1, D^2, \ldots, D^t = \{(x_j^t, y_j^t)\}_{j=1}^{|D^t|}$
- Each training set has a classes set : L^t ,where $L^i \cap L^j = \emptyset, i \neq j$
- The first training set D^1 consists of base classes (large number of training examples per class)
- The remaining training set $D^{t>1}$ as few-shot training set
    - C classes and K training examples per class (C-way K-shot setting)
- Incrementally trained on : $D^1, D^2, \ldots,$ and only D^t available at t^{th} training session
- After training session t^{th}, evaluate on all the encountered classes in $L^1, \ldots, L^t$

Methodology

- **Reduce overfitting**: Choose very few *unimportant* session trainable parameters to train on new classes.
- **Knowledge Retain**: The important parameters in the model are not affected.
- Unimportant para: All parameters in a layer having absolute value lower than the threshold 低绝对值

session trainable parameters
important parameters
unimportant parameters



Methodology: Base class

- Feature extractor Θ^F; Fully connected classifier Θ^C
- All parameters of the network are trainable: CE loss

$$L_{D^{(1)}}(\mathbf{x}, y) = F_{CE}(\Theta_C(\Theta_F(\mathbf{x})), y)$$

- Obtain the class prototypes: Average features of the same class

$$Pr[c] = \frac{1}{N_c} \sum_{k=1}^{N} \mathbb{I}_{(y_k=c)}(\Theta_F(\mathbf{x}_k))$$

- Self-Supervised Auxiliary Task: 旋转图片以增强数据做自监督辅助任务

  Rotation prediction network Θ^R in parallel with Θ^C

$$L_{D^{(1)}}(\mathbf{x}, y) = F_{CE}(\Theta_C(\Theta_F(\mathbf{x})), y) + F_{CE}(\Theta_R(\Theta_F(\mathbf{x})), y^r)$$

Methodology: New class

- Train session trainable para: Triplet loss 拉近同类，推远异类

$$L_{TL}(x_i, x_j, x_k) = max(d(\Theta_F(x_i), \Theta_F(x_j)) - d(\Theta_F(x_i), \Theta_F(x_k)), 0)$$

- Ensure not deviate far from previous values: $l1$-Regularization loss

$$L_{RL} = \sum_{i=1}^{N_p^t} ||w_i^t - w_i^{t-1}||_1$$

- Minimize similarity between prototypes of old and new: Cosine sim

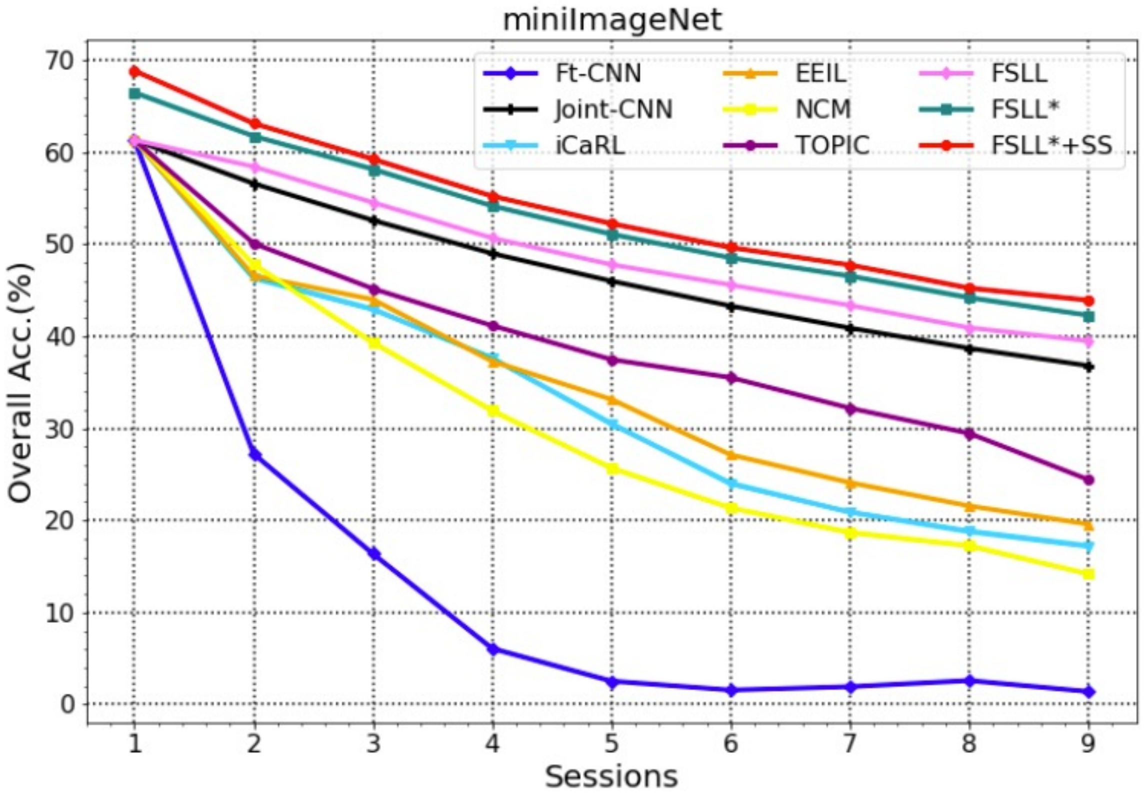
$$L_{CL} = \sum_{i=1}^{N_{Pr}^t} \sum_{j=1}^{N_{Pr}^{prev}} F_{cos}(Pr^t[i], Pr^{prev}[j])$$

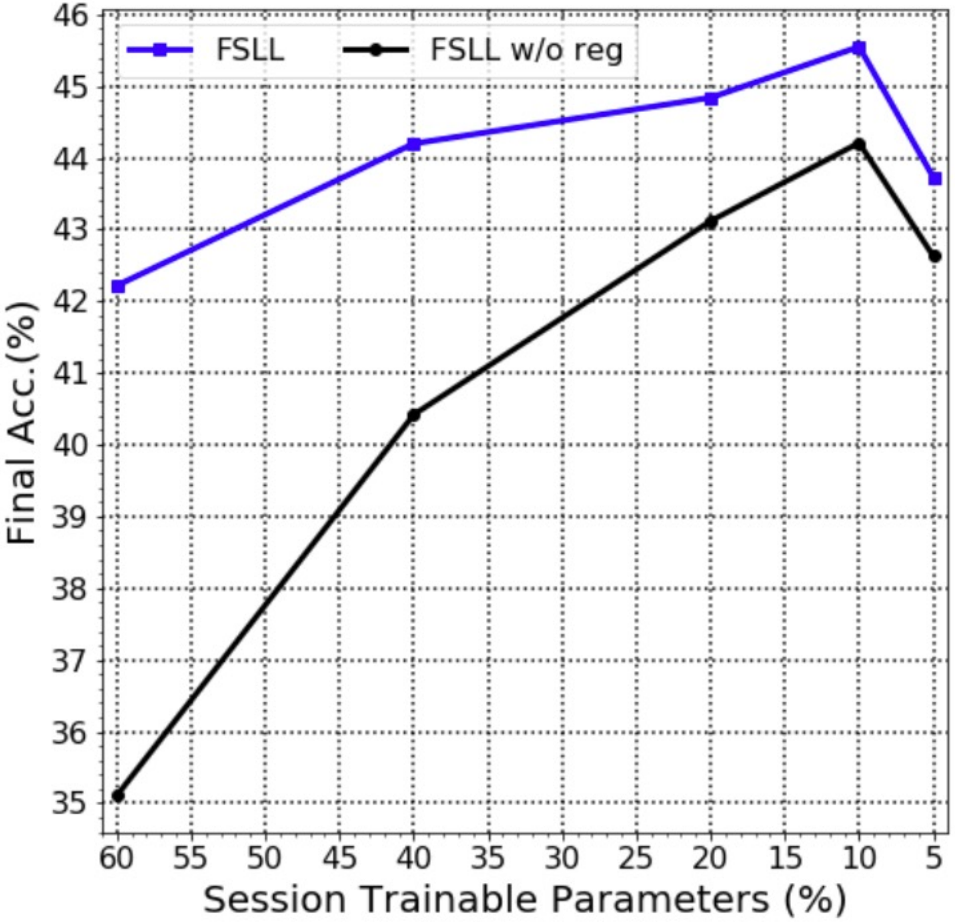
- Total loss: $L(D^{(t>1)}) = L_{TL} + L_{CL} + \lambda L_{RL}$

Experimental Results

Method	Sessions											Our Relative Improvements
	1	2	3	4	5	6	7	8	9	10	11	
Ft-CNN (Tao et al. 2020)	68.68	44.81	32.26	25.83	25.62	25.22	20.84	16.77	18.82	18.25	17.18	**+28.37**
Joint-CNN (Tao et al. 2020)	68.68	62.43	57.23	52.80	49.50	46.10	42.80	40.10	38.70	37.10	35.60	**+9.95**
iCaRL (Rebuffi et al. 2017)	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	**+24.39**
EEIL (Castro et al. 2018)	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	**+23.44**
NCM (Hou et al. 2019)	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	**+25.68**
TOPIC (Tao et al. 2020)	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	**+19.27**
FSLL (Ours)	**68.72**	**65.67**	**62.33**	**58.10**	**55.44**	**52.66**	**51.17**	**50.27**	**48.31**	**47.25**	**45.55**	**0**
FSLL* (Ours)	**72.77**	**69.33**	**65.51**	**62.66**	**61.10**	**58.65**	**57.78**	**57.26**	**55.59**	**55.39**	**54.21**	-
FSLL*+SS (Ours)	**75.63**	**71.81**	**68.16**	**64.32**	**62.61**	**60.10**	**58.82**	**58.70**	**56.45**	**56.41**	**55.82**	-

Experimental Results





Incremental Few-shot Text Classification with Multi-round New Classes: Formulation, Dataset and System

NAACL 2021

Story in NLP

Introduction

- **Challenges**:
  - For the learning process, the system should incrementally learn new classes round by round without re-training on the examples of preceding classes;
  - For the performance, the system should perform well on new classes without much loss on preceding classes.

- **Tasks**:
  - Intent classification: understanding the intents under user queries
  - Relation classification: determine the correct relation between two entities in a given sentence

Problem Formulation

- **Training data**:
  - Provided with m rounds of new classes sequentially $\{C_n^1, \dots, C_n^m\}$
  - Each round C_n^i has h new classes: $C_n^i = \{C_{n,1}^i, \dots, C_{n,h}^i\}$
  - Each new class only has k examples ($k \in [1,5]$)
  - k not fixed, varies for different new classes in the same round $k\_C_{n,s}^i \neq k\_C_{n,t}^i$ (更符合实际)
  - With base classes: $C_b = \left\{C_{b,1}, C_{b,2}, \cdots, C_{b,g}\right\}$
- **No Dev data**: 现实应用中没有dev data供我们选择best model
- **Testing data**:
  - Without base classes: $C_n^1 \cup \cdots \cup C_n^m \cup C_o$
  - With base classes: $C_b \cup C_n^1 \cup \cdots \cup C_n^m \cup C_o$

Methodology

- *Entailment* : casts the text classification problem into textual entailment
  - Positive pair: (x_i , y_i) 文本x_i 与其golden label y_i
  - Negative pair: (x_i , y_j) 文本x_i 与其他label $y_j \in C_n^i, y_j \neq y_i$
- **Training strategy**
  - RoBERT输入[CLS] x [SEP] y [SEP]做二分类，x与y是否为真
  - 首先在大量文本蕴含数据集上做fine-tune
- **Inference strategy**
  - 模型预测出概率>0.5的所有class中最大的作为预测结果
  - 若不存在，则预测为C_o

Datasets

	IFS-Intent			IFS-Relation		
	#class	#train	#test	#class	#train	#test
C_b	20	2088	800	10	5000	400
C_n^1	10	30	400	10	30	400
C_n^2	10	30	400	10	30	400
C_n^3	10	30	400	10	30	400
C_n^4	10	30	400	10	30	400
C_n^5	10	30	400	10	30	400
C_o	7	–	280	10	-	400

Single-domain Multi-domain

Experimental Results

		C_n^1	C_n^2	C_n^3	C_n^4	C_n^5	C_o
C_n^1	DNNC	55.50±2.27					72.29±0.20
	ENTAILMENT	65.17±1.36					75.43±0.41
	HYBRID	**70.08±0.77**					**78.25±0.19**
C_n^2	DNNC	64.58±0.42	77.75±1.08				61.72±0.90
	ENTAILMENT	64.08±2.04	76.33±1.01				**64.68±0.71**
	HYBRID	**74.25±1.34**	**86.67±1.01**				64.39±0.27
C_n^3	DNNC	65.25±1.67	79.58±1.50	64.67±1.93			50.25±0.52
	ENTAILMENT	**75.50±1.63**	83.83±0.62	75.25±1.24			**56.56±2.43**
	HYBRID	74.25±1.08	**85.92±1.05**	**76.58±1.05**			53.09±1.73
C_n^4	DNNC	66.75±0.54	79.08±0.51	60.50±2.35	62.25±1.08		42.56±0.76
	ENTAILMENT	68.33±1.16	72.67±0.77	68.58±1.90	69.50±1.34		**53.92±0.75**
	HYBRID	**73.75±1.41**	**85.50±1.06**	**71.67±1.53**	**75.83±2.44**		52.75±0.63
C_n^5	DNNC	65.33±0.62	76.75±1.59	62.83±3.17	59.75±2.83	57.25±2.32	36.66±1.07
	ENTAILMENT	67.58±0.82	73.50±1.24	67.83±0.47	71.83±0.66	**73.75±0.74**	**50.95±0.68**
	HYBRID	**70.75±1.27**	**82.50±1.27**	**72.42±0.96**	**76.67±1.05**	71.00±0.41	47.05±1.60

Table 2: System performance without base classes on the benchmark IFS-INTENT. Horizontal direction: different groups of testing classes (base classes C_b, five rounds of novel classes ($C_n^1, \cdots, C_n^5$) and the OOD classes C_o); vertical direction: timeline of incremental learning over new rounds of novel classes. Numbers are averaged over results of three random seeds.

		C_b	C_n^1	C_n^2	C_n^3	C_n^4	C_n^5	C_o
C_b	ProtoNet	87.25±0.10						53.4±10.68
	DyFewShot	81.04±1.91						55.01±2.52
	DNNC	95.96±0.68						61.89±4.78
	ENTAILMENT	**96.42±0.41**						**64.73±3.84**
	HYBRID	96.12±0.12						58.92±1.22
C_n^1	ProtoNet	85.83±1.94	31.67±1.48					43.66±3.08
	DyFewShot	81.29±1.56	00.00±0.00					39.33±1.25
	DNNC	**95.75±0.41**	74.83±1.64					**64.54±2.02**
	ENTAILMENT	94.42±0.21	75.42±1.56					56.38±5.29
	HYBRID	95.62±1.00	**77.75±0.25**					58.41±5.10
C_n^2	ProtoNet	83.92±0.33	24.92±5.54	38.83±3.43				31.14±9.83
	DyFewShot	81.29±1.56	00.00±0.00	00.50±0.71				33.94±1.42
	DNNC	95.42±0.62	72.92±4.37	75.08±3.30				**49.02±3.23**
	ENTAILMENT	94.29±0.16	71.92±1.45	**84.83±1.33**				48.12±3.20
	HYBRID	**96.44±0.19**	**76.75±2.75**	75.00±1.00				42.11±0.30
C_n^3	ProtoNet	81.08±2.06	24.33±5.54	30.67±6.17	22.50±1.34			23.62±6.99
	DyFewShot	81.29±1.56	00.00±0.00	00.50±0.71	00.00±0.00			27.48±1.24
	DNNC	**95.67±0.33**	68.17±2.37	66.33±5.02	71.25±3.78			**45.69±1.73**
	ENTAILMENT	92.71±0.41	70.75±0.54	**82.83±2.16**	**73.92±2.52**			29.34±3.31
	HYBRID	95.44±0.44	**73.62±0.62**	71.62±2.62	73.50±0.75			33.69±3.66
C_n^4	ProtoNet	81.17±2.52	17.83±2.58	31.75±0.94	24.92±1.90	22.25±3.19		28.19±4.78
	DyFewShot	81.54±1.71	00.25±0.35	00.17±0.24	00.00±0.00	00.00±0.00		23.52±1.51
	DNNC	95.29±0.16	68.75±2.35	66.75±3.82	67.00±3.40	57.75±1.41		42.09±3.72
	ENTAILMENT	91.67±0.36	65.92±2.18	**79.92±1.78**	**73.75±0.74**	69.08±0.12		**45.73±2.80**
	HYBRID	**95.69±0.06**	**72.12±0.62**	67.75±1.25	70.25±0.25	**72.62±1.38**		38.85±0.89
C_n^5	ProtoNet	80.00±2.65	21.83±5.45	29.17±3.70	24.67±3.12	23.17±3.60	30.33±4.17	29.24±2.96
	DyFewShot	81.50±1.27	00.08±0.12	00.83±0.62	00.00±0.00	00.00±0.00	00.50±0.71	21.23±1.34
	DNNC	95.12±0.47	67.50±0.89	67.92±4.70	64.42±4.17	52.42±1.20	53.33±2.09	30.46±5.92
	ENTAILMENT	89.17±0.60	65.08±2.45	**78.50±0.94**	**69.08±1.12**	**68.25±0.35**	**70.67±1.30**	**39.48±1.45**
	HYBRID	**95.56±0.06**	**68.75±2.75**	67.38±0.62	63.75±1.75	65.12±3.62	61.62±2.38	37.65±0.44

Table 4: System performance with base classes on the benchmark IFS-INTENT.

Comments

- Pro：
  - 提出无需base class的新任务
  - 任务设置更贴合实际应用情况
  - 根据新任务构建了新的dataset

- Con：
  - 方法上亮点不足
  - 更偏向传统地解决intention classification和relation classification任务，而没有针对incremental few-shot任务特点来设计method

Thanks

Q&A