Efficient On-Device Incremental Learning by Weight Freezing

Ze-Han Wang, Zhenli He, Hui Fang, Yi-Xiong Huang, Ying Sun,

Yu Yang, Zhi-Yuan Zhang, Di Liu

School of Software, Yunnan University

ASP-DAC2022

AI-IOT

- Ai-Iot is a combination of artificial intelligence and the Internet of Things
- It collects large amounts of data selectively and connects everything through AI analytics
- It is not a new technology, but a new form of IoT application
- This requires our edge devices to be smarter
- This can not only reduce the pressure on the data terminal, but also better serve the actual application scenarios

AI

- AI is a branch of computer science that seeks to produce a new kind of intelligent machine that can react in a manner similar to human intelligence
- Previous artificial intelligence has been dominated by machine learning, which is based on statistical learning
- With the advent of AlphaGo, deep learning has gradually replaced traditional machine learning
- The main neural network of deep learning is deep neural network(DNNs)

Edge Computing

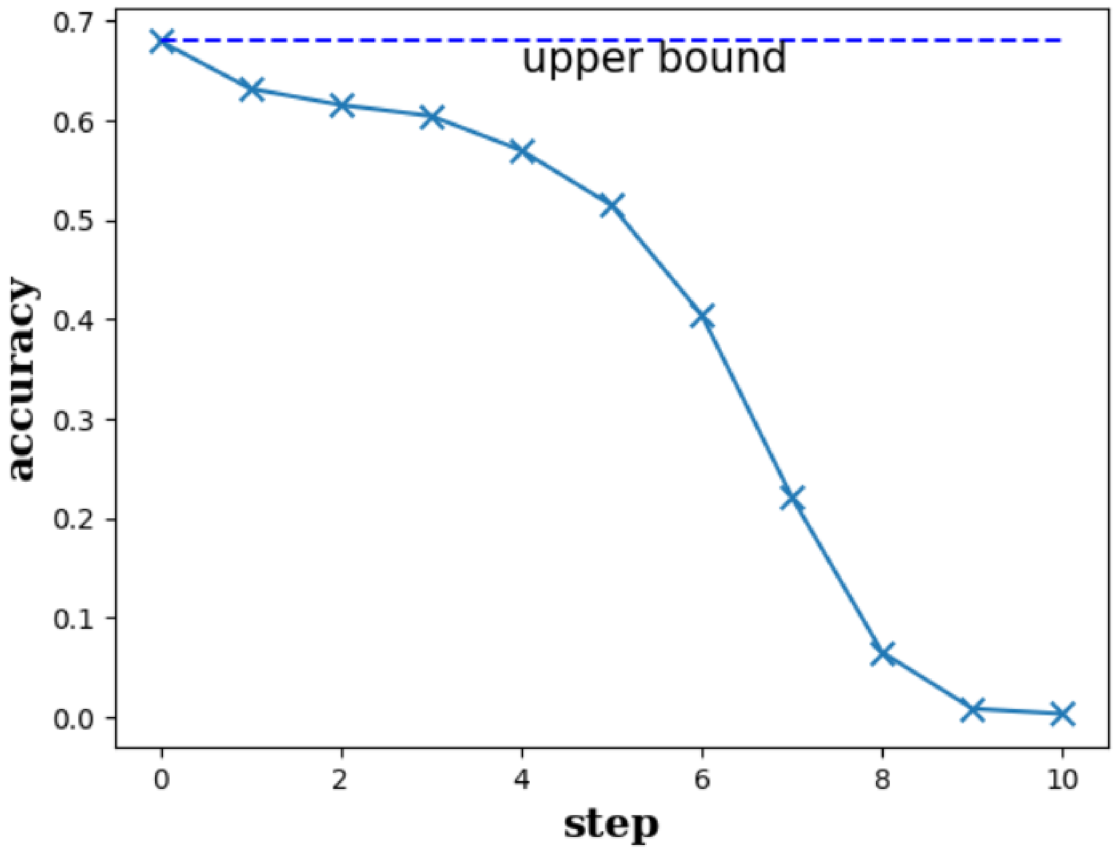
- Deep neural networks (DNNs) have become the ‘brain’ of many AI applications
- DNNs makes edge devices smarter and prevents data leaks
- Edge equipment has poor computing capability and high energy consumption
- DNNs is usually computationally intensive and time-consuming, and does not meet the requirements of edge computing

On-device Incremental Learning

- What is On-device Incremental Learning (ODIL)
  - To learn new knowledge for edge devices
    - New classes are needed
    - Data cannot be exchanged

- The challenges of ODIL
  - Catastrophic forgetting
  - Limited computation for edge systems

Catastrophic Forgetting



Catastrophic Forgetting

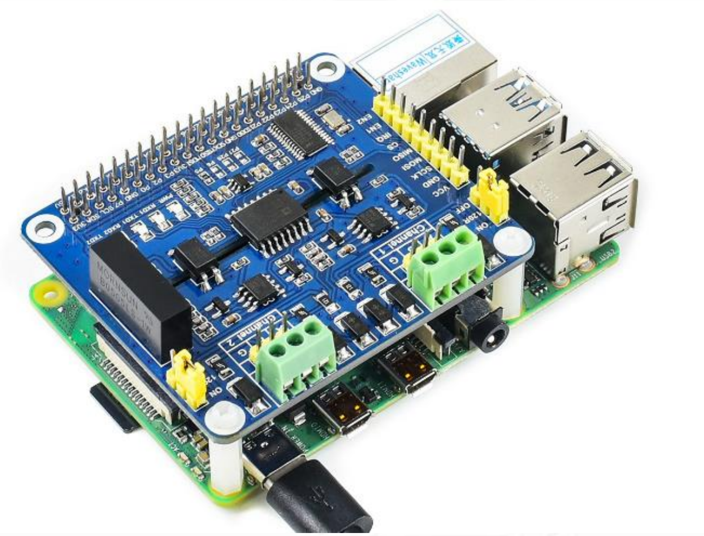
- Catastrophic forgetting : Learning new classes for the original model significantly sacrifices the accuracy for the old classes

- The existing incremental learning methods, e.g., Replay, GAN:
  - Replay: Preserve the images for old classes
  - GAN: Generate the images for old classes

- Issues of the existing incremental learning methods:
  - Memory consumption
  - Training overhead

On-device Incremental Learning

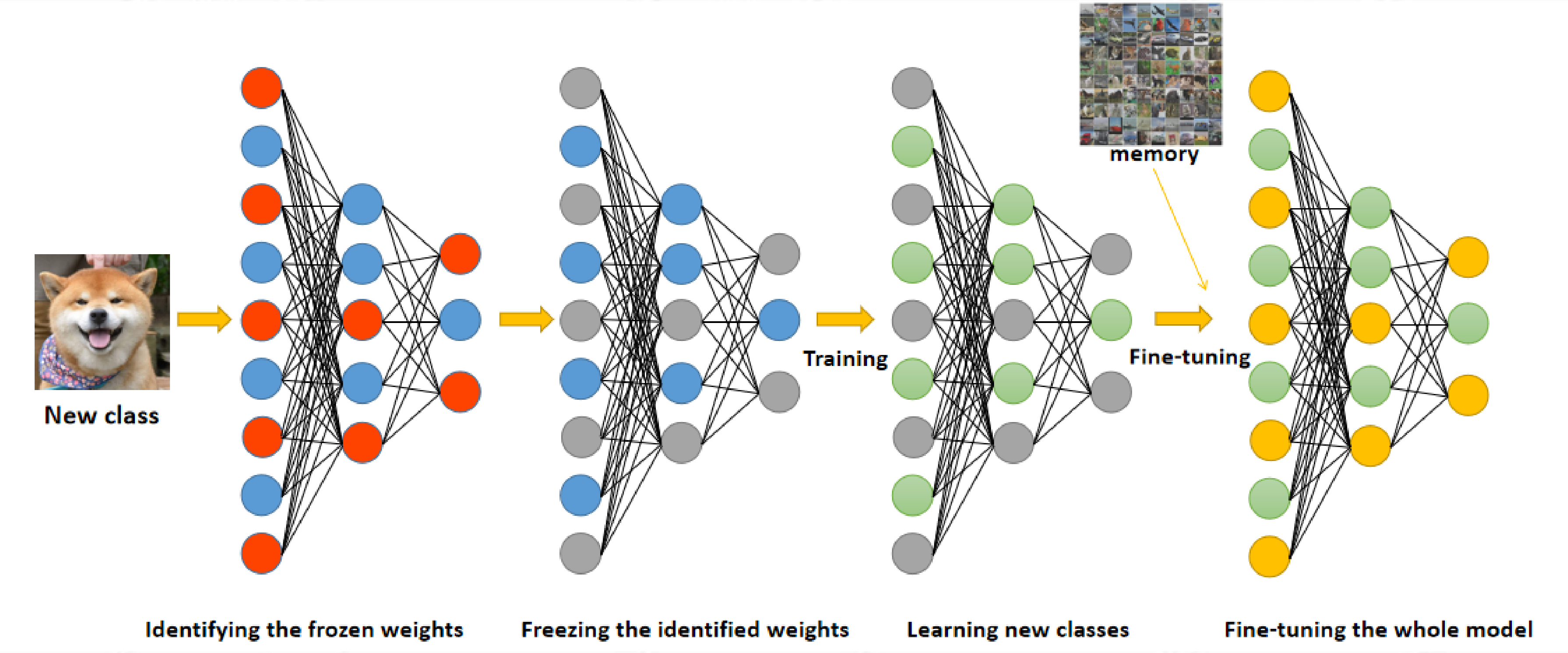
- Features of a successful ODIL method
  - Low training cost
  - Low memory load
  - Sustainable learning ability



Contribution

- Proposed a new method for efficient device-side learning
  - Weight freezing
  - New class of learning
  - Fine-tuning with replay

The overview


memory
New class
Training
Fine-tuning
Identifying the frozen weights
Freezing the identified weights
Learning new classes
Fine-tuning the whole model

Weight Freezing

- **Find out the network weights that are important to the old knowledge**
  - **Similar to weight pruning, identifying the important weights for the old classes**
  - **Frozen network is more efficient to train**

$$I_m = (E(D, W) - E(D, W | w_m = 0))^2$$

$$I_m(W) = (g_m w_m)^2$$

Learning New Classes

- **Weight update indication**

$$W_{new} = W_f \bigcup (W_{nf} + \alpha \cdot \frac{\partial E}{\partial W_f})$$

- **Training settings**
  - **Very few epochs required**

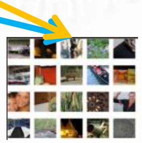
Fine-tuning With Replay

$$L(\omega) = L_C(\omega) + L_{D_f}(\omega)$$

$$L_C(\omega) = -\frac{1}{N}\sum_{i=1}^{N} p_i \log q_i$$

$$L_{D_f}(\omega) = -\frac{1}{N}\sum_{i=1}^{N} \hat{p_i} \log \hat{q}_i$$

new classes



old classes

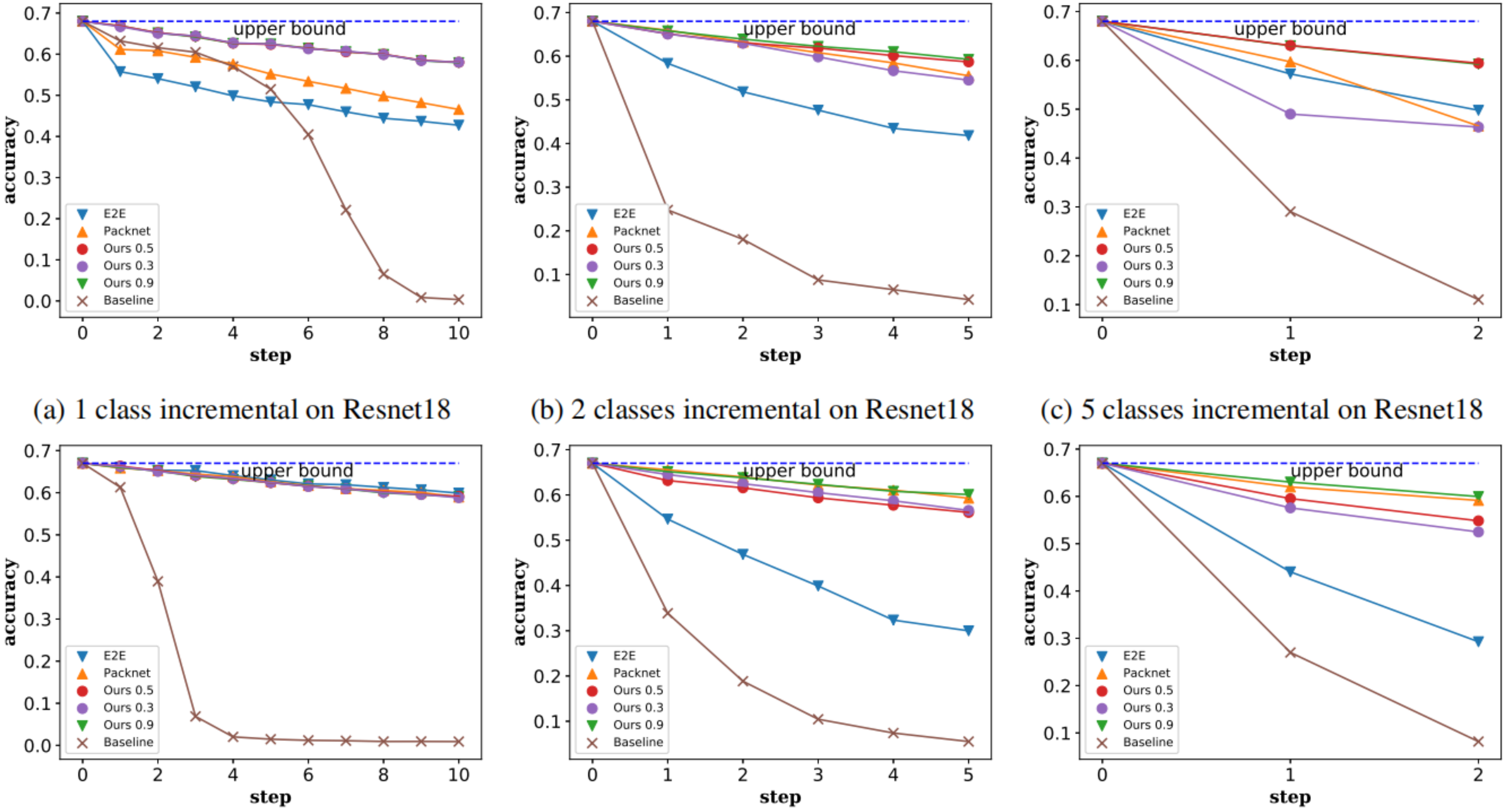
$$\hat{q}_i = \frac{exp(v_i/T)}{\sum_k^N exp(v_k/T)}$$

$$\hat{p_i} = \frac{exp(z_i/T)}{\sum_k^N exp(z_k/T)}$$

Experimental Ideas

- Tested our method on NVIDIA Jeston Nano and GeForce RTX 2060
- Tested the performance of Resnet and MobileNet on the CIFAR100 data set
- Conducted various ablation experiments
  - Different freezing rates
  - Incremental stride of different sizes
  - Results in different backbone networks

Experimental Results



(a) 1 class incremental on Resnet18

(b) 2 classes incremental on Resnet18

(c) 5 classes incremental on Resnet18

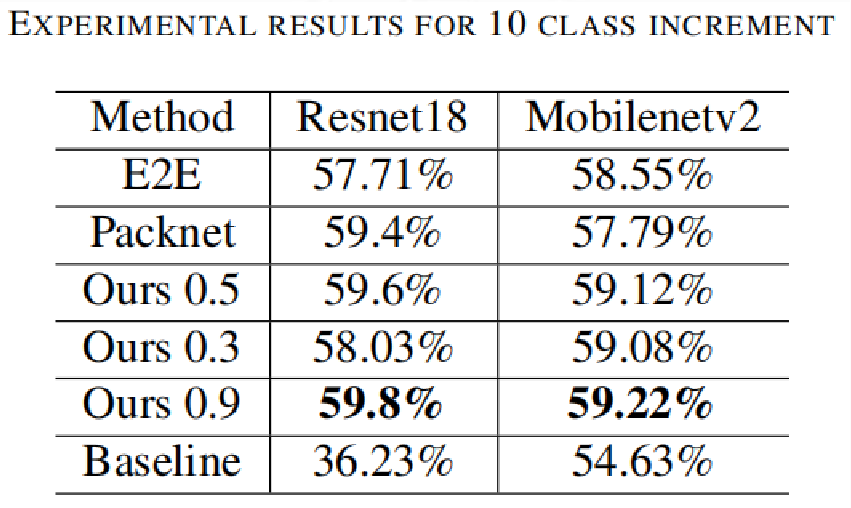
(d) 1 class incremental on MobileNetv2

(e) 2 class incremental on MobileNetv2

(f) 5 class incremental on MobileNetv2

Experimental Results

In order to test the applicability of the method, we tested the performance of the traditional 10 types of large increments, and we can find that our performance is similar or even better

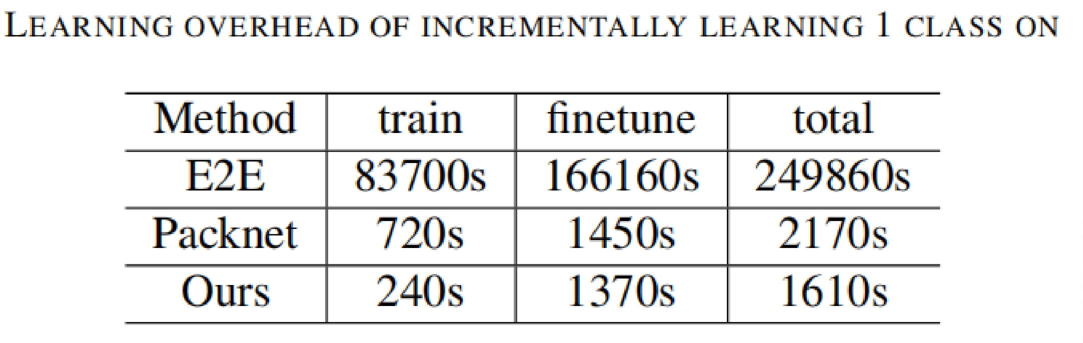


EXPERIMENTAL RESULTS FOR 10 CLASS INCREMENT

Method	Resnet18	Mobilenetv2
E2E	57.71%	58.55%
Packnet	59.4%	57.79%
Ours 0.5	59.6%	59.12%
Ours 0.3	58.03%	59.08%
Ours 0.9	**59.8%**	**59.22%**
Baseline	36.23%	54.63%

Experimental Results

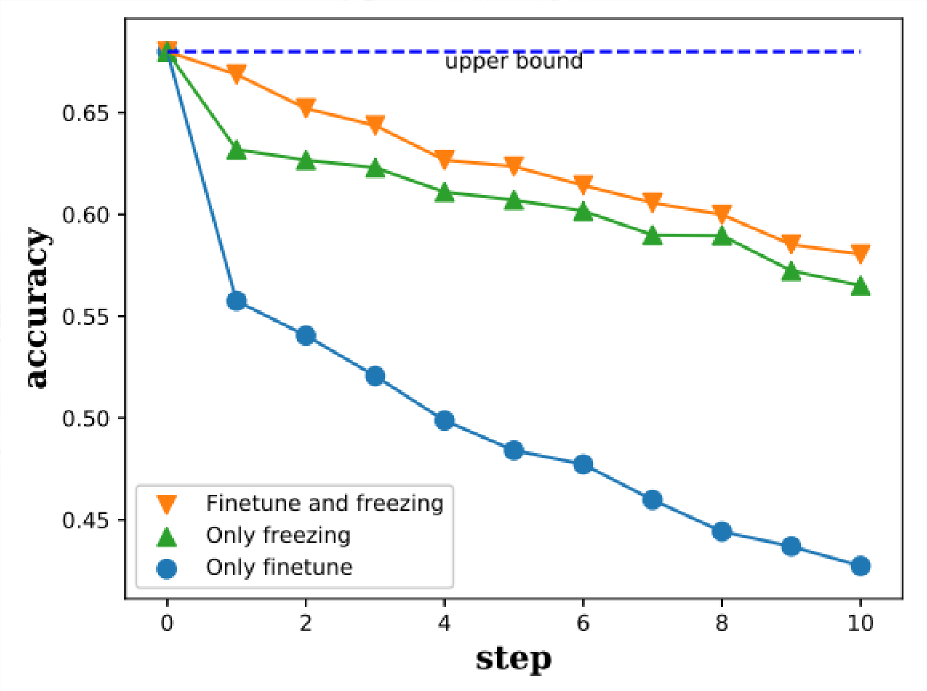
We also tested the related training time consumption on NVIDIA Jeston Nano. Obviously our method requires less time, which also means that less energy needs to be consumed, so that edge devices have better battery life.



LEARNING OVERHEAD OF INCREMENTALLY LEARNING 1 CLASS ON

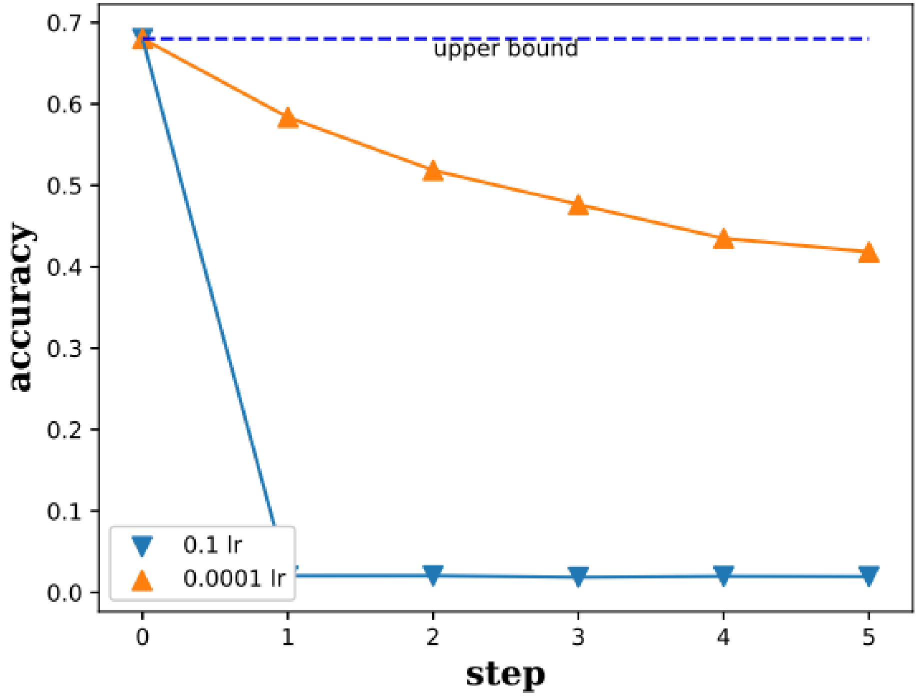
Method	train	finetune	total
E2E	83700s	166160s	249860s
Packnet	720s	1450s	2170s
Ours	240s	1370s	1610s

Experimental Results



. Ablation study of weight freezing and fine-tuning

Experimental Results



The influence of learning rate

Conclusion

- We propose an efficient on-device incremental learning method

- Resolve catastrophic forgetting and limited training on edge equipment

- The comprehensive training methods of freezing weight, replay fine tuning and knowledge distillation are proposed

- The experimental results demonstrate the efficacy and efficiency of our proposed approach.

Thank you for your attention!

感谢聆听！