Neural Architecture for Online Ensemble Continual Learning

Mateusz Wójcik[1,2] **Witold Kościukiewicz**[1,2] **Tomasz Kajdanowicz**[2] **Adam Gonczarek**[1]

[1] Alphamoon Ltd., Grabarska 1, 50-072 Wrocław [2]Wroclaw University of Science and Technology

Contact: mateusz.wojcik@alphamoon.ai

Introduction

Continual learning with an **increasing number of classes** (Figure 1) is a challenging task. The difficulty rises when each example is presented exactly once (online learning) and when a memory buffer is unavailable.

We propose the **fully differentiable** ensemble method called **DE&E** that allows us to efficiently train an **ensemble of neural networks** in the end-to-end regime. The presented architecture (Figure 2) is inspired by an Encoders and Ensembles (hereafter referred to as E&E) [1] and adapted to the most challenging **task-free online class incremental** setup.

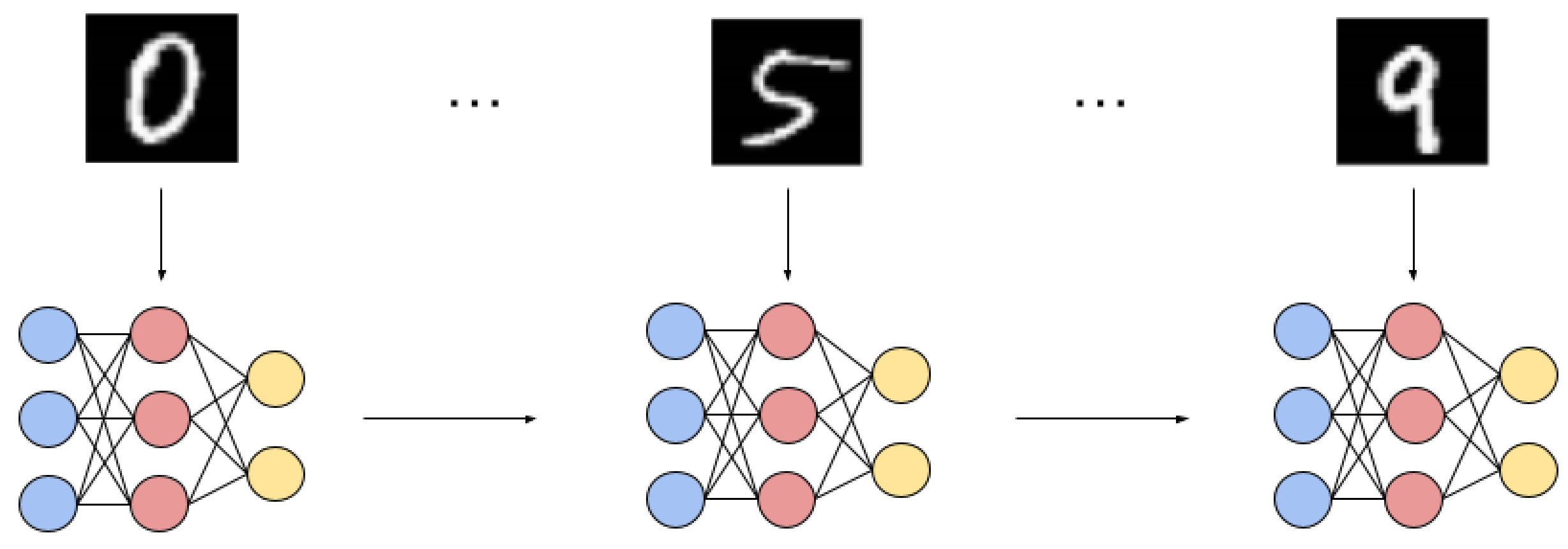

Figure 1: Class incremental continual learning setup.

Architecture

We extend the E&E architecture and **improve it by**:

- introducing a **differentiable KNN layer** (*soft KNN*) [2]
- proposing **a novel approach to aggregate ensemble predictions**

Input image is processed by the feature extractor. Each classifier has an associated key, which is a random vector of the same length as the feature extractor output. Obtained embeddings are used to find the most relevant classifiers according to their keys. The *soft KNN* layer approximates the *soft KNN* scores. Predictions are weighted in the voting layer by both cosine similarity and *soft KNN* scores. Final output is the class with the highest voting score.

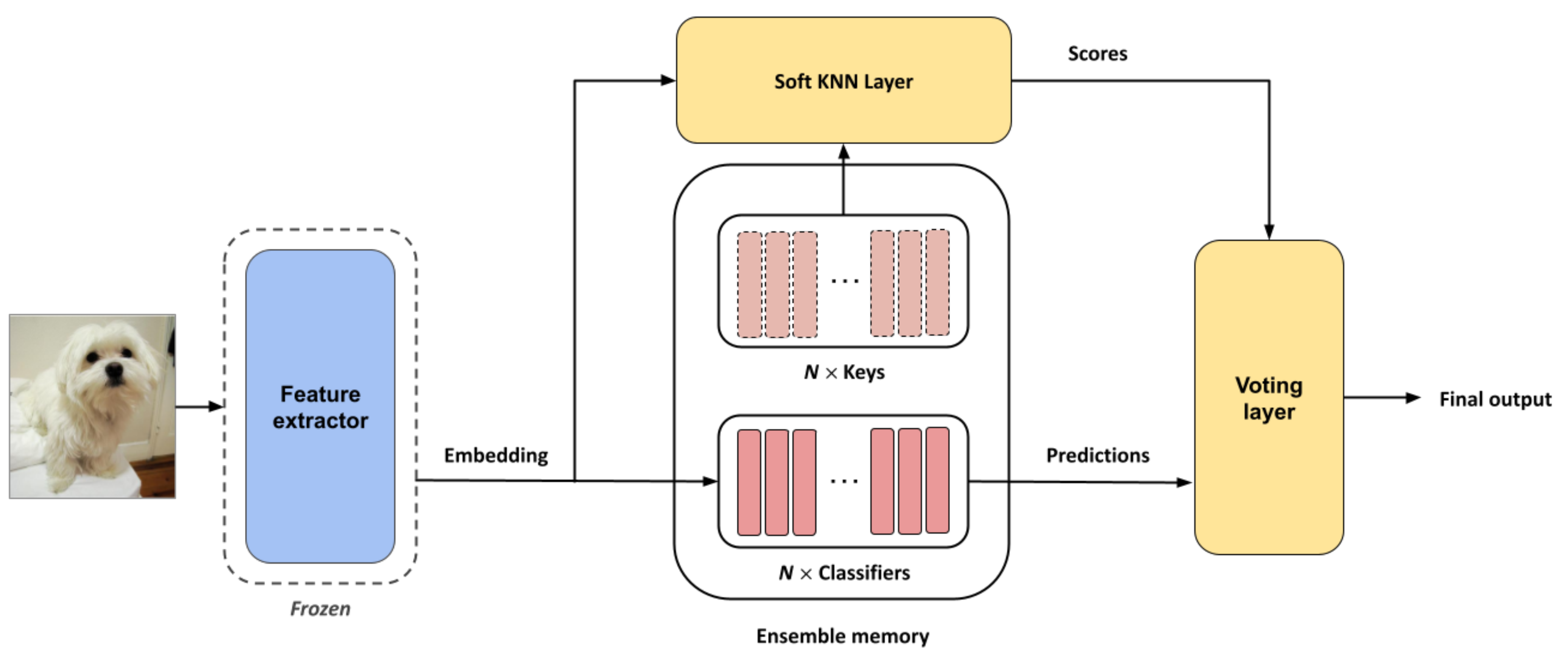

Figure 2: Proposed model architecture.

Soft KNN and voting procedure

The standard KNN algorithm is often implemented using ordinary sorting operations that make it non-differentiable. However, it is possible to obtain **a differentiable approximation of the KNN model** by solving the Optimal Transport Problem [3]. Based on this concept, we add a *soft KNN* layer to the model architecture.

We use both cosine similarity and *soft KNN* approximation to weight the predictions by giving the higher impact for classifiers with keys similar to extracted features. **The final prediction is obtained as follows:**

$$\hat{\mathbf{y}} = \frac{\sum_{n=1}^{N} \gamma_n c_n \hat{\mathbf{y}}_n}{\sum_{n=1}^{N} c_n}$$

where N is the number of classifiers, c_n is the cosine similarity between the input and classifier keys, γ_n is the *soft KNN* approximation and the $\hat{\mathbf{y}}_n$ is the classifier output.

Experiments result

For all setups evaluated, our model performed best improving results of main reference method (E&E) up to 6%. We can also see a significant difference in achieved accuracy between the DE&E approach and baselines. Furthermore, it achieved this results without replaying training examples seen in the past. For the ensemble of 128 classifiers and MNIST dataset, our architecture achieved accuracy more than **18% better than the best method with a memory buffer.**

	MNIST (10 splits)		**CIFAR-10 (5 splits)**	
	$N = 16$	$N = 128$	$N = 64$	$N = 128$
Naive	14.41 ±5.99	11.63 ±2.22	19.65 ±0.33	19.70 ±0.36
LwF	12.38 ±3.99	9.88 ±0.55	19.48 ±0.55	19.62 ±0.60
EWC	14.33 ±4.44	10.97 ±2.32	19.52 ±0.29	19.88 ±0.50
SI	10.18 ±1.00	17.22 ±4.64	17.97 ±2.40	21.32 ±5.76
CWR*	16.41 ±5.42	10.38 ±0.79	18.92 ±2.97	22.41 ±2.00
GEM (10 / exp)	67.81 ±2.61	58.92 ±6.34	30.75 ±1.41	29.27 ±1.46
A-GEM (10 / exp)	53.59 ±5.21	21.31 ±15.90	39.86 ±14.25	36.12 ±6.19
Replay (10 / exp)	74.49 ±3.84	69.02 ±4.90	44.03 ±3.72	43.82 ±7.10
E&E	78.16 ±1.85	85.60 ±0.52	46.34 ±1.98	56.24 ±1.41
DE&E (ours)	**84.19** ±1.00	**87.54** ±0.24	**48.78** ±1.34	**59.36** ±0.73

Architecture advantages

The proposed method achieves **higher accuracy having the same number of parameters.** The smaller the ensemble, the higher the gain in accuracy. For an ensemble of 1024 classifiers, the accuracy is already very close, suggesting that the gain decreases with large ensembles (Figure 2).

We also observed **significantly reduced forgetting** relative to the reference method (Figure 3). The larger the ensemble the relatively less knowledge is forgotten. Stronger specialization amplified by the introduced voting method makes classifiers less likely to lose acquired knowledge.

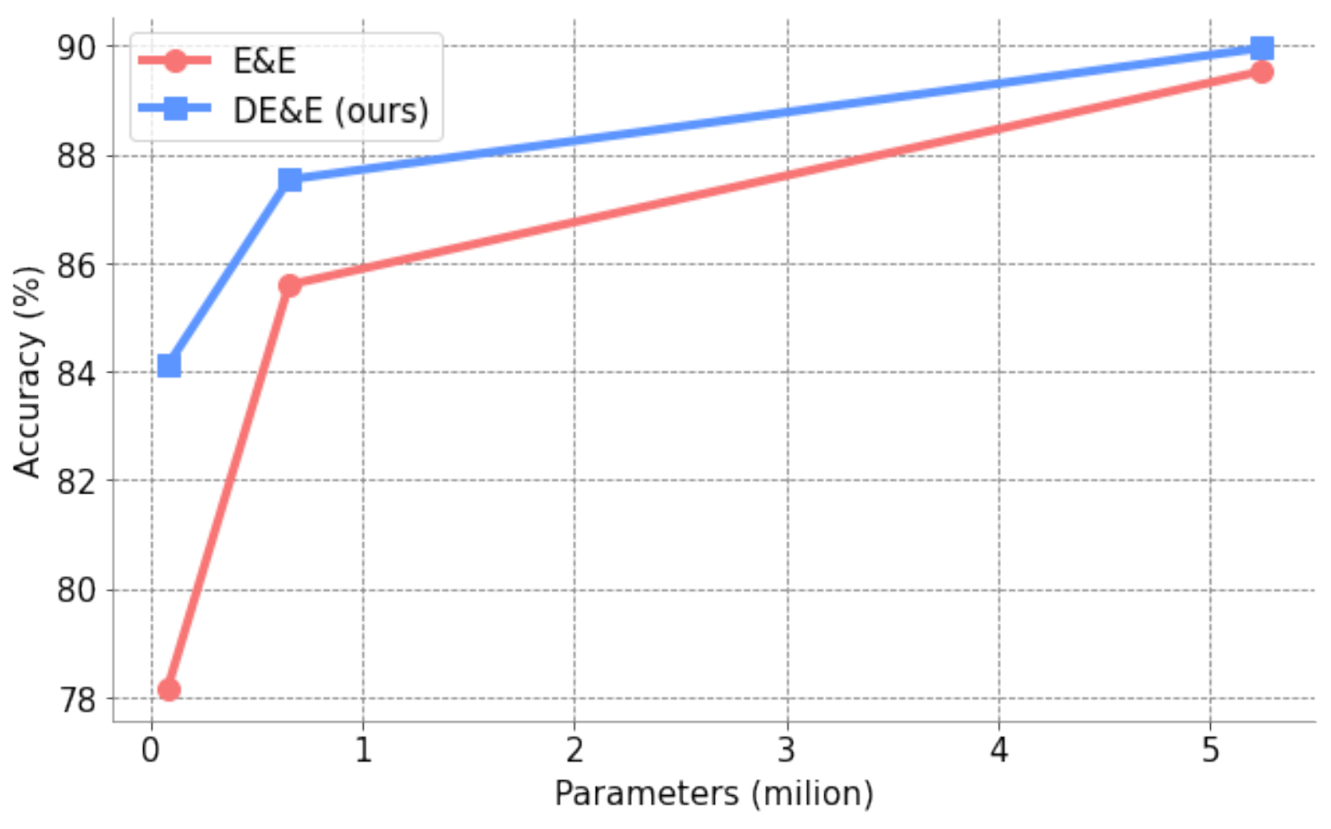

Figure 3: Number of weights in ensembles (16, 128, 1024 classifiers) and achieved accuracy (%) on 10-split MNIST.

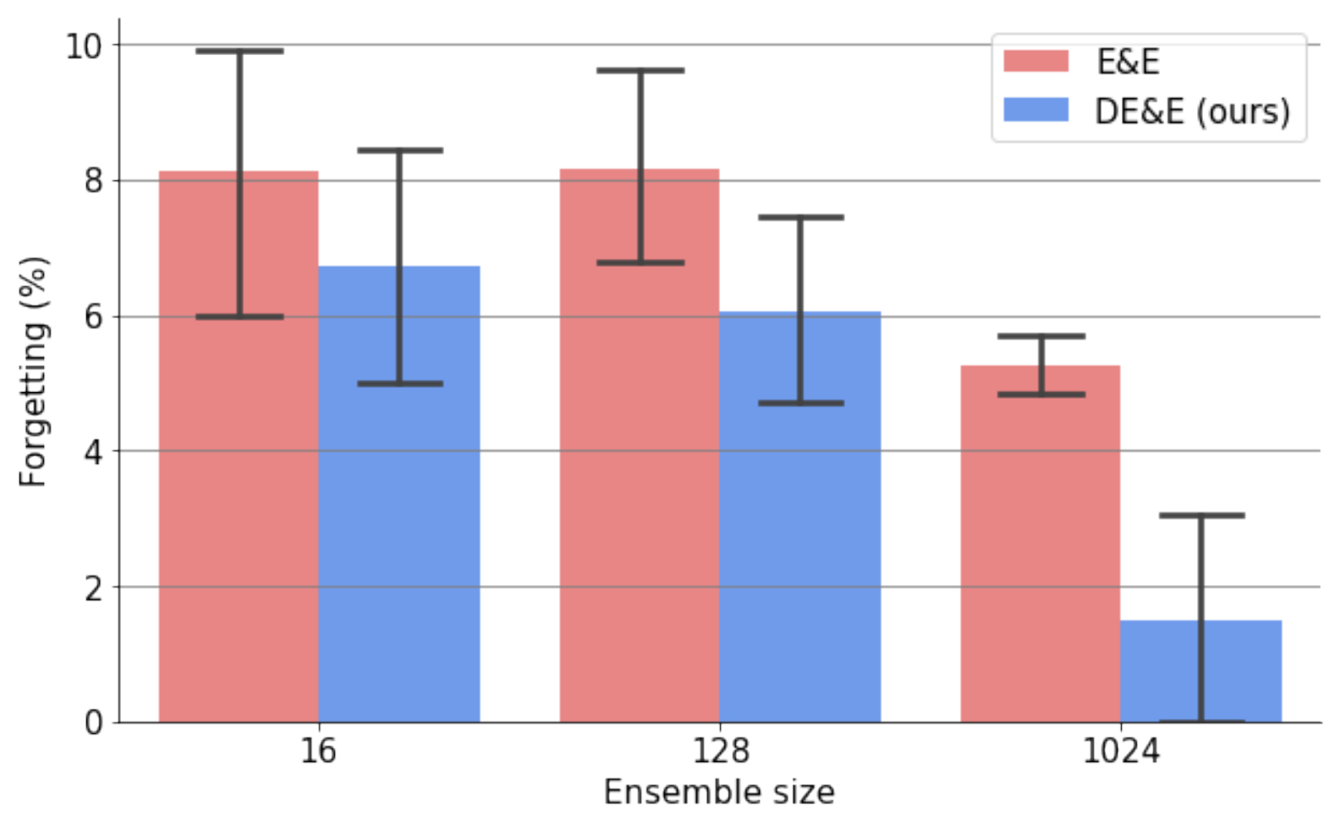

Figure 4: Averaged forgetting rate (the lower the better) for ensembles evaluated on 10-split MNIST.

Summary

We showed improved accuracy for all of the cases studied and achieved SOTA results. We have shown that it is possible to noticeably improve the quality of classification and reduce forgetting rate using the ensemble with the same number of parameters. This effect is observed especially in small ensembles that gained significantly higher performance. The presented architecture outperforms methods with a memory buffer and enables researchers to make further steps towards overrun the current SOTA in class incremental problems. Undoubtedly, the field of continual learning using ensemble methods needs more attention due to its vast potential.

References

[1] Murray Shanahan, Christos Kaplanis, and Jovana Mitrovic. Encoders and ensembles for task-free continual learning. *CoRR*, abs/2105.13327, 2021.

[2] Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20520–20531. Curran Associates, Inc., 2020.

[3] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.