HAL open science

Aggregated f -average Neural Network for Interpretable Ensembling

Mathieu Vu, Emilie Chouzenoux, Jean-Christophe Pesquet, Ismail Ben Ayed

To cite this version:

Mathieu Vu, Emilie Chouzenoux, Jean-Christophe Pesquet, Ismail Ben Ayed. Aggregated f -average Neural Network for Interpretable Ensembling. 2023. hal-04237149v2

HAL Id: hal-04237149

https://inria.hal.science/hal-04237149v2

Preprint submitted on 12 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Aggregated f-average Neural Network for Interpretable Ensembling

Mathieu Vu, Émilie Chouzenoux, *IEEE Senior Member*, Jean-Christophe Pesquet, *IEEE Fellow*, and Ismail Ben Ayed

***Abstract*—Ensemble learning leverages multiple models (i.e., weak learners) on a common machine learning task to enhance prediction performance. Basic ensembling approaches average the weak learners outputs, while more sophisticated ones stack a machine learning model in between the weak learners outputs and the final prediction. This work fuses both aforementioned frameworks. We introduce an *aggregated f-average* (AFA) shallow neural network which models and combines different types of averages to perform an optimal aggregation of the weak learners predictions. We emphasise its interpretable architecture and simple training strategy, and illustrate its good performance on the problem of few-shot class incremental learning.**

***Index Terms*—ensemble learning, estimator aggregation, weakly supervised learning, few-shot learning, incremental learning.**

I. Introduction

A. Ensemble learning

Ensemble learning (or ensembling) is a set of methods which leverage an ensemble of models (also called weak learners), instead of relying on a single learner to perform a given machine learning task (e.g., classification). While ensembling is obviously more demanding in terms of computing resources, it can achieve better accuracy and generalisation, improve overall stability, and reduce prediction variance and bias. Two main phases are identified in the process of building an ensemble model, namely the training of weak learners and the fusion of outputs [1]. The former focuses on producing an ensemble of diverse models, which is a crucial step in ensemble learning [2], [3], [4], [5]. For example, bootstrap aggregating (or bagging) [6] trains each model on a different subset of the training data to produce diverse weak learners. Output fusion gathers outputs from each weak learner of the ensemble and combines them to produce the final prediction [7], [8], [9].

One could distinguish two categories of methods for output fusion in ensemble learning [1]. The most basic one is to average weak learners outputs or, in the case of classification, to use a majority voting scheme [10]. Different types of averages could be used (e.g., arithmetic, geometric, harmonic, etc) and weights could be included to further refine results. Those weights can be set using various kinds of criteria, for example based on weak learners isolated performance [11]. The second category of methods uses meta-learners. They consist in plugging an additional model, responsible for taking advantage of the weak learners. Mixture of experts is a popular variant of meta-learners, where a gating network selects the weak learner that is most suited to produce the correct prediction given a certain input [12], [13]. A more straightforward output fusion based on meta-learners is the stacking of an additional learning model. Taking weak learners output as input, it learns the best combination to assemble the unified prediction [14], [15].

B. Contribution

Our contribution, in this work, is to introduce *aggregated f-average* (AFA) neural networks (NNs), based on a novel architecture for the output fusion phase of ensemble learning. It consists of a shallow neural network modelling different types of averages (arithmetic, geometric, harmonic, etc.) and is able, through supervised learning, to combine and/or select them optimally. Thanks to a specific architecture including original nonlinear activations and constrained weights, it is easily interpretable. To illustrate the performance of AFA neural networks upon the state-of-the-art, we describe their application and implementation in the currently popular setting of few-shot class incremental learning (FSCIL).

The paper is organised as follows. First, in section II, we present the architecture of our AFA model along with its training process. We then introduce, in section III, the FSCIL problem and describe our ensembling approach in this context. Experiments on several datasets highlight the benefits of our model, when compared with other ensemble output fusion methods.

II. Methodology

A. Ensembling through averaging

Let K machine learning models trained for a common task (e.g., classification), produce K outputs $(x_k)_{1\leq k\leq K}$, assumed to be vectors in $\mathbb{R}^N$. In ensemble learning, those K outputs are combined during an output fusion phase in order to produce a single, expectably better, prediction for the task at hand. A naive method is to average the outputs. We summarize in Table I common expressions for weighted averages, with $(\omega_k)_{1\leq k\leq K}$ nonnegative reals such that $\sum_{k=1}^{K}\omega_k = 1$.

M. Vu and E. Chouzenoux acknowledge support from the European Research Council under Starting Grant MAJORIS ERC-2019-STG-850925.

M. Vu, E. Chouzenoux, and J-C. Pesquet are with OPIS of Inria Saclay (Palaiseau, France) and CVN at CentraleSupélec, Université Paris-Saclay (Saclay, France). I. Ben Ayed is with LIVIA at ÉTS Montréal (Montréal, Canada). Corresponding author: Mathieu Vu (mathieu.vu@inria.fr).

TABLE I
EXAMPLES OF WEIGHTED AVERAGES ALONG WITH THEIR VALIDITY DOMAINS.

Mean	Arithmetic	Geometric	Harmonic	Power-q
Formula	$\sum_{k=1}^{K} \omega_k x_k$	$\prod_{k=1}^{K} x_k^{\omega_k}$	$\left(\sum_{k=1}^{K} \frac{\omega_k}{x_k}\right)^{-1}$	$\left(\sum_{k=1}^{K} \omega_k x_k^q\right)^{1/q}$
Validity	$x_k \in \mathbb{R}^N$	$x_k \in [0,+\infty)^N$	$x_k \in (0,+\infty)^N$	$x_k \in [0,+\infty)^N$, $q > 0$

B. f-average

Following Kolmogorov's mean framework [16], we rewrite the above examples under the generalised form:

$$\tilde{x} = f^{-1}\Big(\sum_{k=1}^{K} \omega_k f(x_k)\Big). \tag{1}$$

Hereabove, f is a bijective function from $[0,+\infty)^N$ to some convex C of $\mathbb{R}^N$, and f^{-1} is its inverse function from C to $[0,+\infty)^N$. We will subsequently assume that f operates componentwise, in the sense that it consists of the application of the same scalar function to each of the components of its argument. Let us now express functions f and f^{-1} to retrieve the popular averaging rules from Table I. We denote $(\xi_n)_{1\leq n\leq N}$ the components of $x \in \mathbb{R}^N$. We set $\epsilon \in (0,+\infty)$ and, to circumvent the indefiniteness of the harmonic mean for vectors with a zero component, we define the *leaky hyperbolic* function h_ϵ as

$$(\forall \xi \in \mathbb{R}) \quad h_\epsilon(\xi) = \begin{cases} \dfrac{1}{\xi+\epsilon} - \epsilon & \text{if } \xi \in [0, 1/\epsilon - \epsilon] \\ -\dfrac{\xi}{\epsilon^2} + \dfrac{1}{\epsilon} - \epsilon & \text{if } \xi < 0 \\ -\epsilon^2\Big(\xi - \dfrac{1}{\epsilon} + \epsilon\Big) & \text{if } \xi > 1/\epsilon - \epsilon. \end{cases} \tag{2}$$

Table II summarizes the expression for f, f^{-1}, as well as their associated definition domains, recovering the averaging rules from Table I. Exact geometric and harmonic means formula are retrieved when ϵ goes to zero.

We now propose to extend the generalised average framework (1) to the case when scalars $(\omega_k)_{1\leq k\leq K}$ are replaced by matrices $(\Omega_k)_{1\leq k\leq K}$ in $\mathbb{R}^{N\times N}$, so as to allow a full mixing of the weak learners. Given some functions (f, f^{-1}) defined as previously, the f-average output $\tilde{x} \in \mathbb{R}^N$ is obtained as

$$\tilde{x} = f^{-1}\left(\sum_{k=1}^{K} \Omega_k f(x_k)\right) = f^{-1}\big(W\boldsymbol{f}(\boldsymbol{x})\big). \tag{3}$$

Hereabove,

$$W = [\Omega_1, \ldots, \Omega_K] \in \mathbb{R}^{N\times KN}, \tag{4}$$

and

$$\boldsymbol{f}\colon \quad [0,+\infty)^{KN} \to C^K \tag{5}$$

$$\boldsymbol{x} = (x_k)_{1\leq k\leq K} \mapsto \big(f(x_k)\big)_{1\leq k\leq K} \tag{6}$$

applies f in a parallel manner to the vector inputs $(x_k)_{1\leq k\leq K}$.

In order to ensure the interpretability of the averaging operation in (3), W is chosen such that

$$\begin{cases} W \in [0,+\infty)^{N\times KN}, \\ W\mathbb{1}_{KN} = \mathbb{1}_N. \end{cases} \tag{7}$$

This guarantees, in particular, that $W\boldsymbol{f}(\boldsymbol{x})$ belongs to the definition domain of f^{-1} (since this domain has been assumed to be convex). For instance, if $\boldsymbol{x} \in [0,1]^{KN}$ (e.g., in a classification context), the constraint on W ensures that the output $\tilde{x}$ also belongs to $[0,1]^N$.

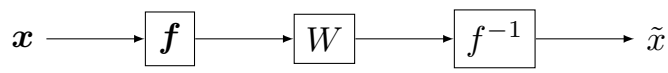

Fig. 1. Structure of a neural network that performs an f-average for ensembling

Remarkably, operation (3), that we call an f-average, can be represented as the application, on $\boldsymbol{x}$, of a two-layer neural network whose structure is drawn in Figure 1. This neural network is parametrized by the choice of f (along with its inverse function f^{-1}) and by the weight matrix W. The former can be set by the user, while the latter can be determined through supervised learning, by minimizing a loss associated to the task at hand. The f-average network is interpretable, as the contribution of each output in the final prediction can easily be retrieved using the weights in matrix W, and the averaging operation is determined by the choice of f.

C. Aggregated f-averages

The previous approach requires the prior choice for the average rule (i.e., f). To wave this restriction, we suggest aggregating $J > 1$ f-averages, associated to different functions $(f_j)_{1\leq j\leq J}$. Resorting to the same structure as the one presented in the previous section, we define

$$(\forall j \in \{1,\ldots,J\}) \quad \tilde{x}_j = f_j^{-1}\Big(W_j \boldsymbol{f}_j(\boldsymbol{x})\Big), \tag{8}$$

where, for every $j \in \{1,\ldots,J\}$,

$$\begin{cases} W_j \in [0,+\infty)^{N\times KN} \\ W_j\mathbb{1}_{KN} = \mathbb{1}_N, \end{cases} \tag{9}$$

and $\boldsymbol{f}_j$ is a function operating component-wise from $[0,+\infty)^{KN}$ to C_j, associated to a given mean function f_j, f_j^{-1} is its inverse function operating component-wise from some convex set $C_j \subset \mathbb{R}^N$ to $[0,+\infty)^N$, and $\boldsymbol{x}$ concatenates

TABLE II
EXAMPLES FOR f, f^{-1}, AND DEFINITION DOMAINS, FOR $x = (\xi_n)_{1\leq n\leq N}$. GEOMETRIC AND HARMONIC FORMULAS ARE RETRIEVED WHEN $\epsilon \to 0$.

Mean	$f(x)$	f domain	$f^{-1}(x)$	f^{-1} domain
Arithmetic	Id	$[0,+\infty)^N$	Id	$[0,+\infty)^N$
Geometric	$\big(\ln(\xi_n+\epsilon)\big)_{1\leq n\leq N}$	$[0,+\infty)^N$	$\big(\exp(\xi_n)-\epsilon\big)_{1\leq n\leq N}$	$[\ln\epsilon,+\infty)^N$
Harmonic	$\big(h_\epsilon(\xi_n)\big)_{1\leq n\leq N}$	$[0,+\infty)^N$	$\big(h_\epsilon(\xi_n)\big)_{1\leq n\leq N}$	$(-\infty,\epsilon^{-1}-\epsilon]^N$
Power-q	$(\xi_n^q)_{1\leq n\leq N}$	$[0,+\infty)^N$	$(\xi_n^{1/q})_{1\leq n\leq N}$	$[0,+\infty)^N$

columnwise the inputs $(x_k)_{1\leq k\leq K}$. The resulting joint aggregate estimate of the J outputs $(\tilde{x}_k)_{1\leq j\leq J}$ is defined as

$$\widehat{x} = \sum_{j=1}^{J} A_j \tilde{x}_j = A \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_J \end{bmatrix}, \tag{10}$$

with, for every $j \in \{1,\dots,J\}$, $A_j \in [0,+\infty)^{N\times N}$, and $A \in [0,+\infty)^{N\times NJ}$ is the rowwise stacking of $(A_j)_{1\leq j\leq J}$ matrices. Operations (8)-(10) are equivalent to plug $\boldsymbol{x}$ as the input of a neural network with J sub-networks of the form presented in Figure 1, operating in parallel, followed by a linear layer involving the weight matrix A. We further add a final activation function, $g\colon \mathbb{R}^N \to \mathbb{R}^N$, to control the domain of the output. For instance, a softmax activation can be used to get nonnegative outputs summing to one, in a classification context. The resulting network, called *aggregated f-average*, is displayed in Figure 2. It has a limited number $JN^2(K+1)$ of linear parameters, namely the entries of matrices $W_1,\dots,W_J$, and A. The training of these parameters can follow a classical supervised learning approach. Given a sample and its ground truth, the task model loss is computed (e.g., a cross-entropy loss for classification), before updating the weights from all layers using a backpropagation algorithm (e.g., Adam optimizer [17]). Constraints on weight matrices $(W_j)_{1\leq j\leq J}$ can simply be imposed through a projection step of each row of these matrices on the convex unit simplex set [18], performed after each backpropagation update. Let us remark that this model maintains the interpretability properties of the J f-average sub-models it contains. Furthermore, weights from matrix A can be viewed as the contribution level of each type of average model.

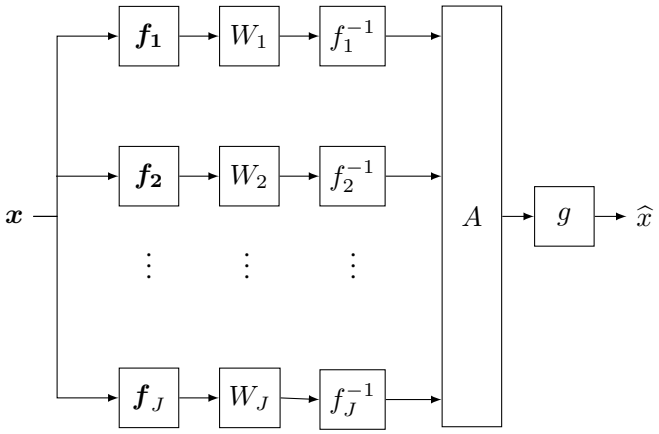

Fig. 2. Structure of the proposed aggregated f-average neural network. It aggregates J f-averages for ensembling, with $A \in [0,+\infty)^{N\times NJ}$. The activation function $g : \mathbb{R}^N \to \mathbb{R}^N$ is selected according to the task (e.g, softmax for classification, linear for regression).

III. AGGREGATED f-AVERAGE FOR FEW SHOT INCREMENTAL LEARNING

A. FSCIL framework and datasets

We now illustrate the potential of the proposed AFA framework by applying it to the problem of few-shot class incremental learning. FSCIL, recently introduced in [19], focuses on designing machine learning methods that can deal with both incremental [20], [21] and few-shot situations [22], [23], [24], [25]. Specifically, FSCIL aims at including an increasing number of categories in a classification problem with the extra constraint that only a small number of training samples are available for upcoming classes. This is motivated by the frequent practical situation in computer vision when a model, built to classify certain categories seen during training phase, must be adjusted to classify images belonging to new classes with only (very) few annotations. The main difficulty revolves around the ability to learn new classes while preventing *catastrophic forgetting* of classes previously learned, yielding poor performance of standard incremental learning methods.

The FSCIL setting considered here consists of K successive sessions that incrementally provide new categories of images to be classified. First session ($k=1$), also called base session, is a standard classification problem with a large number n_{train} of training samples for each of the $n_{\text{class_base}}$ base classes. In subsequent sessions ($k \in \{2,\dots,K\}$), only a small additional number n_{shots} of training samples is provided for a limited number n_{way} of novel classes. At each session k, the number of classes to predict is

$$N_k = n_{\text{class_base}} + (k-1)n_{\text{way}}. \tag{11}$$

Our experiments focus on four typical FSCIL datasets, which characteristics are summarized in Table III.

TABLE III
FSCIL SETTINGS FOR EACH DATASET.

Dataset	$n_{\text{class_base}}$	n_{way}	n_{shots}	n_{train}
mini-ImageNet [26]	60	5	5	480
tiered-ImageNet [27]	100	10	5	850
FGVC-Aircraft [28]	50	5	5	70
CUB-200 [29]	100	10	5	60

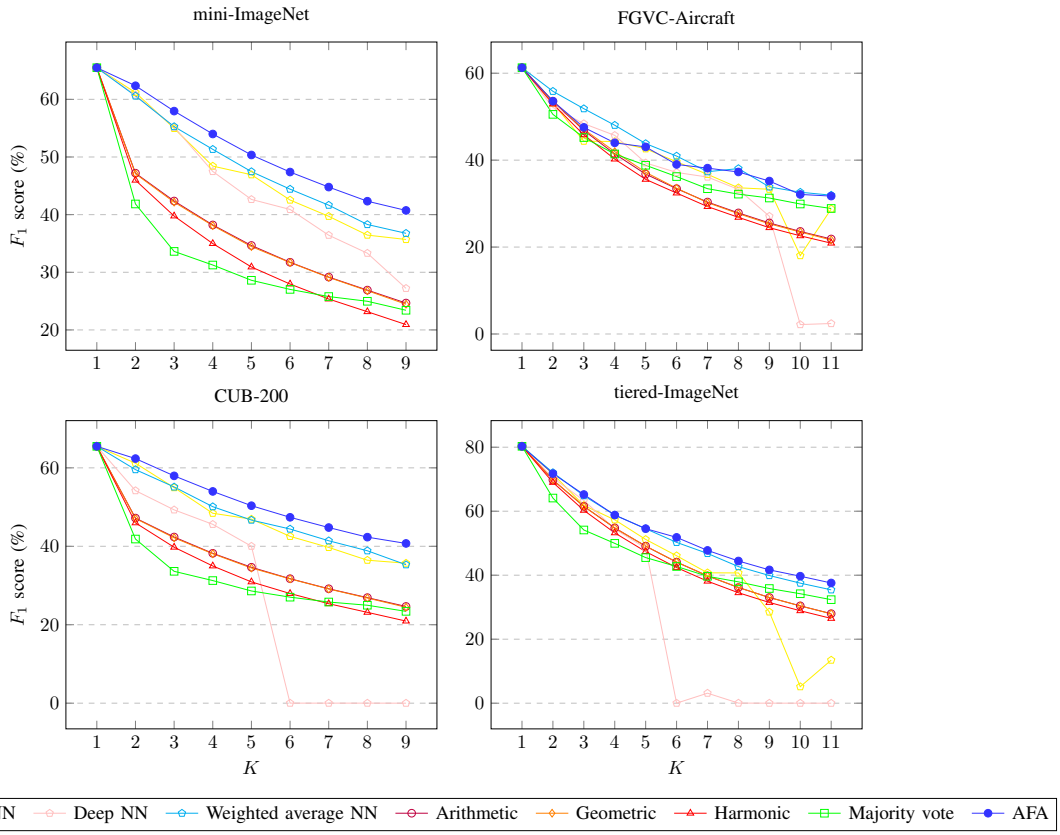

Fig. 3. Comparison of ensemble learning output fusion methods, on FSCIL datasets, in terms of averaged F_1 score over all classes, for various sessions k. The closer the F_1 score is to 1, the better. As sessions are incrementally added, the classification task becomes increasingly more difficult because of the additional categories to be handled.

B. Application of aggregated f-average

We address FSCIL as the ensembling of successive few-shot learning problems [30]. For each session, we train a few-shot learning classifier that serves as a weak classifier (i.e., weak learner) specialised on its own session in our ensemble model. During base session, the weak classifier is a standard classifier with a ResNet-18 architecture [31] trained thanks to the larger training set. Then, for the remaining sessions ($k \geq 2$), its backbone (i.e., all layers except the last fully connected layer) is frozen and used as feature extractor. For these sessions, that last fully connected layer (specialised on base session) is replaced by a few-shot learning method. We use a state-of-the-art nearest-neighbour classifier that determines the estimated class of a test image by retrieving the closest mean centroid in the feature space [32]. It is trained to classify the n_{way} new classes provided by the current session.

As described in Figure 5, for a given K, our aggregated f-average model takes as input the outputs from sessions $k \in \{1, \ldots, K\}$, of size $N = N_K$ (i.e., the total number of classes, defined in (11)), to form a vector $\boldsymbol{x} \in \mathbb{R}^{KN_K}$. As the weak classifiers from earlier sessions are trained to predict a fewer number of classes, a zero-padding is performed for its prediction vector in order to reach output size N_K. Our AFA model is trained for 100 epochs, minimising a cross-entropy loss with the Adam algorithm [17], using an initial learning rate of 10^{-1} decreased by a factor 10 at epochs 40 and 70. The training uses a *prototype rehearsal* strategy [33] which consists of collecting predictions from all previous weak classifiers on training sets from all previous sessions to form the training set of the ensemble learning model. It is then fed to the model as mini-batches of size 64. We set $J = 4$ and $(f_j)_{1 \leq j \leq J}$ chosen so as to aggregate four different types of means, namely arithmetic, geometric, harmonic, and quadratic (power-2), using the activation functions presented in Table II with $\epsilon = 10^{-4}$. Weights of matrices $(W_j)_{1 \leq j \leq J}$ and matrix A are initialised with entries sampled randomly following a uniform distribution between 0 and $\frac{1}{KN_K}$ before being projected onto the simplex to comply with the sum-to-one constraint. Our model and all experiments were implemented using Tensorflow [34].

C. Results

We compare our proposed model performance, for increasing values of K, against classic ensembling methods for classification performing output fusion. We try out different types of averages (namely, arithmetic, geometric, harmonic), as well as a majority vote scheme. Performance were also compared against three different types of ensembling neural network: 1) a shallow neural network with similar number of parameters and layers as our AFA model, 2) a deeper neural network with five fully connected layers, and 3) a neural network specifically designed for ensembling including

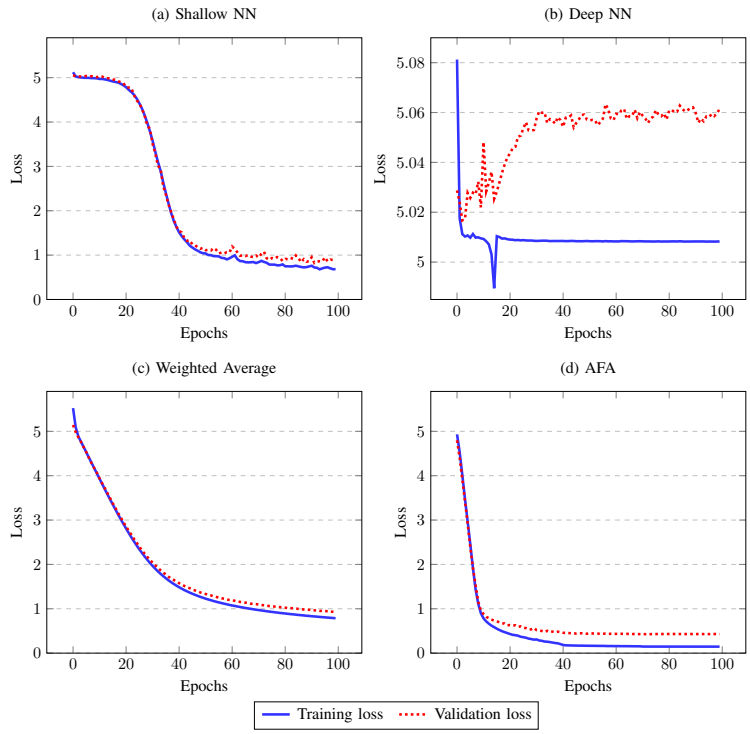

Fig. 4. Training and validation losses of different stacked NN models for ensembling. Models are trained on the same data, at the same session (K=5) on CUB-200, and with the same training parameters, except for learning rates adapted to each architecture. The AFA model presents a loss that converges faster to its plateau, at a lower value than other models. AFA and weighted average losses are smooth and decrease regularly, while the deep and shallow NN models present more instability on the validation loss. Especially, the deep NN shows a divergent validation loss.

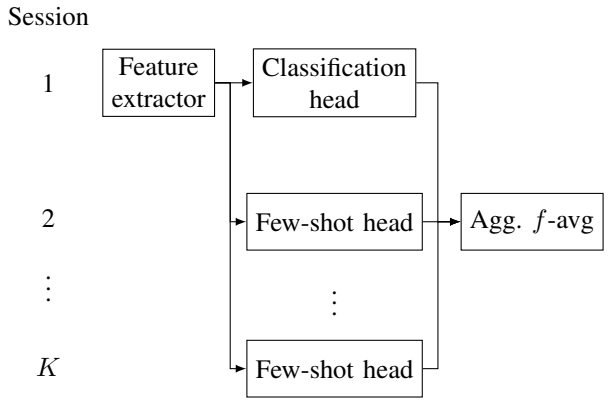

Fig. 5. Proposed AFA method for FSCIL. The feature extractor is frozen from base session onward. Weak classifiers are trained at each session so that the ensembling model combines their output to compute the best prediction for every session number K.

a weighted average layer followed by a fully connected layer for the output [35]. All neural network models, including the AFA model, were trained with the same process with only slight adjustments on learning rate parameters to adapt to each model architecture. The results, in terms of accuracy and F_1 scores (the larger, the better), are summarized in Table IV and Figure 3. Note that the mean accuracy over all classes might hide performance discrepancies between base and new classes [36], as can be seen in Table IV. Typically, a model trained on base session only (called *No Fine-tuning* in Table IV), reaches a decent mean accuracy thanks to its performance on base session classes ($k = 1$), while it yields very poor performance on new classes ($k > 1$). In contrast, our approach AFA performs an effective incremental learning, with good accuracy on both base and new classes.

TABLE IV
DETAILED METRICS COMPARISON ON THE LAST SESSION ($K = 9$) OF MINI-IMAGENET DATASET. FOR ALL METRICS, THE HIGHER THE BETTER.

Metrics	No Fine-tuning	Shallow NN	Deep NN	Weighted Avg.	AFA
Accuracy	39.26	32.69	27.04	38.63	**41.29**
Base Accuracy	**65.43**	47.38	38.80	55.23	55.12
New Accuracy	00.00	10.65	09.40	13.73	**20.55**
$F_1(\%)$	30.01	32.11	25.24	37.06	**40.73**

Classic averaging shows inconsistent results (see Figure 3), as the optimal type of average depends on the session number and on the dataset. The majority vote scheme seems to produce worse results in early sessions (low K), when few voters are available, and a gain in performance with an increasing number of voters. By design, the proposed ensemble method models those different types of average, and learns their optimal combination, thus producing significantly better performance in every session.

AFA also outperforms other NN approaches: among the three models compared with the AFA model, best results were achieved by the weighted average neural network model. Despite its dedicated design, its performance drop more significantly than our AFA model with more sessions incrementally added. Because of its generic architecture, the shallow NN produces even worse results while still improving with respect to classic averages. Finally, the deep NN shows the most

inconsistent results; its performance catastrophically drop in later sessions (higher K) showing training issues. Indeed, these later sessions drastically increase its architecture size due to the higher number K of models to ensemble and the higher number of classes N_K to predict, while not providing a significantly larger training set because of the few-shot constraint of our task at hand.

Figure 4 illustrates these behaviours. It presents training losses of the different stacked neural network models, including the proposed AFA, on a specific example. The plots demonstrate our model' stability and fast convergence to a lower loss value than other methods which exhibit a plateau at a higher value. The deep NN loss behavior shows the inability to train a large number of parameters on such limited amount of data, with a validation loss diverging.

IV. Conclusion

In this article, we presented a novel output fusion method for ensemble learning. Inspired by basic methods, namely averaging and model stacking, our method relies on an original neural network architecture that models multiple types of averages. It is trained to learn an optimal and interpretable selection and/or combination of the multiple weak learners inputs. We illustrated its operability and efficiency on the problem of few-shot class incremental learning where it significantly outperforms standard ensemble learning output fusion methods.

References

[1] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[2] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[3] Z. Chen, J. Duan, L. Kang, and G. Qiu, "Class-imbalanced deep learning via a class-balanced ensemble," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5626–5640, 2022.

[4] V. Cheplygina, D. M. J. Tax, and M. Loog, "Dissimilarity-based ensembles for multiple instance learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1379–1391, 2016.

[5] M. M. Hassan, M. S. AlRakhami, A. Z. Elhendi, and S. A. AlQahtani, "A distributed ensemble of diverse deep learning models for predicting covid-19 cases," in *IEEE EUROCON 2023 - 20th International Conference on Smart Technologies*, pp. 585–589, 2023.

[6] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[7] O. Kviman, H. Melin, H. Koptagel, V. Elvira, and J. Lagergren, "Multiple importance sampling elbo and deep ensembles of variational approximations," in *International Conference on Artificial Intelligence and Statistics*, pp. 10687–10702, PMLR, 2022.

[8] D. Luengo, L. Martino, V. Elvira, and M. Bugallo, "Efficient linear fusion of partial estimators," *Digital Signal Processing*, vol. 78, pp. 265–283, 2018.

[9] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017.

[10] F. Biggs, V. Zantedeschi, and B. Guedj, "On margins and generalisation for voting classifiers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9713–9726, 2022.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[12] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[13] Y. Bian, Q. Song, M. Du, J. Yao, H. Chen, and X. Hu, "Subarchitecture ensemble pruning in neural architecture search," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7928–7936, 2022.

[14] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[15] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, "An ensemble of fine-tuned convolutional neural networks for medical image classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 31–40, 2016.

[16] A. Kolmogorov, *Mathematics and Mechanics*. Kluwer, 1930.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Mathematical Programming Series A*, vol. 158, no. 1, pp. 575–585, 2016.

[19] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, pp. 12183–12192, 2020.

[20] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, pp. 233–248, 2018.

[21] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pp. 831–839, 2019.

[22] S. Singh, A. Majumdar, E. Chouzenoux, and G. Chierchia, "Semi-supervised deep convolutional transform learning for hyperspectral image classification," in *IEEE International Conference on Image Processing (ICIP 2022)*, pp. 206–210, 2022.

[23] W. Tang, A. Panahi, and H. Krim, "Joint concept matching based learning for zero-shot recognition," *arXiv preprint arXiv:1906.05879*, 2019.

[24] S. T. Martin, M. Boudiaf, E. Chouzenoux, J.-C. Pesquet, and I. B. Ayed, "Towards practical few-shot query sets: Transductive minimum description length inference," in *Advances in Neural Information Processing Systems* (A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds.), 2022.

[25] M. Boudiaf, I. Ziko, J. Rony, J. Dolz, P. Piantanida, and I. Ben Ayed, "Information maximization for few-shot learning," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 2445–2457, Curran Associates, Inc., 2020.

[26] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[27] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," *arXiv preprint arXiv:1803.00676*, 2018.

[28] S. Maji *et al.*, "Fine-grained visual classification of aircraft," tech. rep., Oxford University, 2013.

[29] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.

[30] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 770–778, 2016.

[32] Y. Wang, W. Chao, K. Q. Weinberger, and L. van der Maaten, "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning," *CoRR*, vol. abs/1911.04623, 2019.

[33] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pp. 2001–2010, 2017.

[34] M. Abadi, A. Agarwal, P. Barham, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[35] S. Ozechi, "Ensembling neural network models with tensorflow." https://blog.paperspace.com/ensembling-neural-network-models/. Accessed: 2023-09-05.

[36] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–50, 2016.