End-to-End Optimization and Learning for Multiagent Ensembles

Extended Abstract

James Kotary*
Syracuse University
Syracuse, United States of America
jkotary@syr.edu

Vincenzo Di Vito*
Syracuse University
Syracuse, United States of America
vdivitof@syr.edu

Ferdinando Fioretto
Syracuse University
Syracuse, United States of America
ffiorett@syr.edu

ABSTRACT

Ensemble learning is an important class of algorithms aiming at creating accurate machine learning models by combining predictions from individual agents. A key challenge for the design of these models is to create effective rules to combine individual predictions for any particular input sample. This paper proposes a unique integration of constrained optimization and learning to derive specialized consensus rules. The paper shows how to derive the ensemble learning task as end-to-end training of a discrete subset selection module. Results over standard benchmarks demonstrate an ability to substantially outperform conventional consensus rules in a variety of settings.

KEYWORDS

Ensemble multi-agent learning; decision focused learning; integration of optimization and learning.

ACM Reference Format:
James Kotary*, Vincenzo Di Vito*, and Ferdinando Fioretto. 2023. End-to-End Optimization and Learning for Multiagent Ensembles: Extended Abstract . In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Deep neural networks (DNNs) represent the current state-of-the art in a variety of machine learning tasks, and are finding utility in new applications at a rapid pace. Despite their effectiveness in pattern recognition tasks such as image classification and language modeling, they are subject to unavoidable error in deployment, due to phenomena such as overfitting and convergence to local minima in training. Even models with very high accuracy produce faulty classifications and erroneous predictions on some input samples, but similarly constructed models may fail on different input samples depending on differences of network architecture, training algorithm, or distribution of training data. *Ensemble learning* [6] is a meta-algorithm which aims to create accurate and robust machine learning models by collecting and combining the outputs of individually pre-trained models. These individual *ensemble agents* or *members*, despite being trained to perform the same task, may exhibit error diversity, i.e., failure on separate samples so that their accuracy profiles complement each other across an overall distribution of test samples.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

2 SETTING AND GOALS

In the present setting, an *ensemble* consists of a collection of n agents' models or *base learners* represented by functions $f_i, 1 \leq i \leq n$, trained independently on separate (but possibly overlapping) datasets $(\mathcal{X}_i, \mathcal{Y}_i)$, all on the same intended *classification* task. On every task studied, it assumed that $(\mathcal{X}_i, \mathcal{Y}_i)$ are given, along with a prescription for training each agents' models, so that f_i are assumed to be pre-configured.

Let $n \in \mathbb{N}$ be the number of ensemble agents, $c \in \mathbb{N}$ the number of classes and $d \in \mathbb{N}$ the input feature size. Given a sample $z \in \mathbb{R}^d$, each base learner $f_j : \mathbb{R}^d \to \mathbb{R}^c$ computes $f_j(z) = \hat{y}_j$.

Each classifier $f_i(\phi_i, x)$ is trained with respect to its parameters ϕ_i to minimize a classification loss $\mathcal{L}$ as

$$\min_{\phi_i} \mathbb{E}_{(x,y)\sim(\mathcal{X}_i,\mathcal{Y}_i)} \left[\mathcal{L}(f_i(\phi_i, x), y)\right]. \tag{1}$$

The goal is then to combine the agents into a *smart ensemble*, whose aggregated classifier g performs the same task, but with greater overall accuracy on a *master dataset* $(\mathcal{X}, \mathcal{Y})$, where $\mathcal{X}_i \subset \mathcal{X}$ and $\mathcal{Y}_i \subset \mathcal{Y}$ for all i with $0 \leq i \leq n$:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim(\mathcal{X},\mathcal{Y})} \left[\mathcal{L}(g(\theta, x), y)\right]. \tag{2}$$

3 END-TO-END MULTIAGENT ENSEMBLE LEARNING (E2E-MEL)

To combine predictions of individual agents, conventional ensemble learning schemes resort to selection criteria such as majority voting. The end-to-end learning scheme in this work is motivated by the intuition that a more useful ensemble prediction could instead be informed by the samples' input features, and that voting over a well-selected sub-ensemble can provide more accurate inferences than voting over the entire master ensemble. Further, it can do so more reliably than a well-selected single base learner can. The sub-ensemble size k is treated as a hyperparameter. While a natural choice would be to select only the best predicted agent (thus, setting $k = 1$) for a given input sample, it is consistently observed (as the paper shows in Section 4) that optimal task performance is achieved for a value of k that is strictly larger than 1 but smaller than n.

An *end-to-end Smart Multi Agent Ensemble* (e2e-MEL), or *smart ensemble*, consists of an ensemble of agent models along with a *selection net* which is trained by e2e-MEL to select the *sub-ensemble*, of specified size k, which produces the most accurate combined prediction for a given input.

The e2e-MEL model is composed of three main steps:

(1) Predict a vector of scores $g_\theta(z) = \hat{c}$, estimating a notion of prediction accuracy for each base learner on sample z.

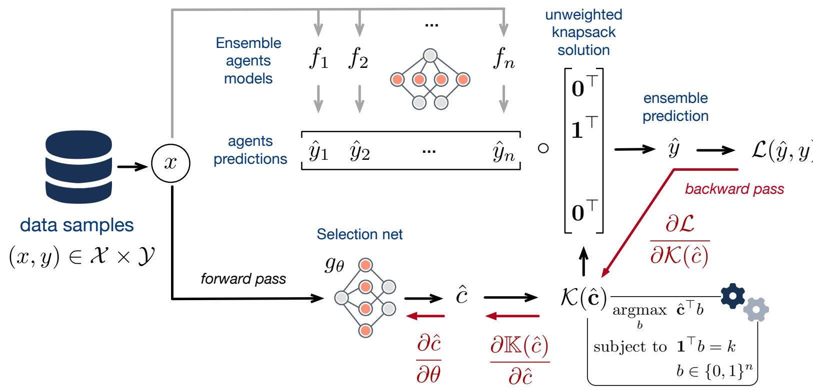

Figure 1: End-to-end Smart Multi Agent Ensemble learning scheme: Black and red arrows illustrate forward and backward operations, respectively.

(2) Identify the base learner indices $\mathcal{E} \subset [n]$ which correspond the the top k predicted scores.

(3) Collect the predictions of the selected sub-ensemble: $f_j(z), j \in \mathcal{E}$ and perform an approximate majority voting scheme over those predictions to determine the class of z.

Figure 1 illustrates the e2e-MEL model and training, in terms of its component operations. Backpropagation is illustrated in backward red arrows; note that only operations downstream from the selection net g are backpropagated, since the e2e-MEL is parameterized by the parameters of g alone.

3.1 Differentiable Model Selection

The e2e-MEL system is based on learning to select $k < n$ predictions from the master ensemble, given a set of input features. This can be done by way of a structured prediction of binary values, which are then used to mask the individual agent predictions. The unweighted knapsack problem

$$\mathcal{K}(\hat{c}) = \underset{b}{\text{argmax}} \quad \hat{c}^\top b \tag{3a}$$

$$\text{subject to} \quad \mathbf{1}^\top b = k, \tag{3b}$$

$$b \in \{0,1\}^n, \tag{3c}$$

can be viewed as a selection problem whose optimal solution assigns the value 1 to the elements of b associated to the top k values of $\hat{c}$. Relaxing the constraint (3c) to $0 \leq b \leq 1$ results in an equivalent linear programming problem with discrete optimal solutions $b \in \{0,1\}^n$, despite being both convex and composed of continuous functions. This useful property holds for any linear program with totally unimodular constraints and integer right-side-side coefficients [1]. The LP problem can be viewed as a mapping from $\hat{c}$ to a binary vector b indicating its top k values, and can be modeled differentiably by smoothing with random perturbations as proposed in Berthet et al. [2].

4 EVALUATION AND CONCLUSIONS

In order to maximize the agent's error diversity, for each task the agents are designed to be specialized for recognizing either one or two particular classes. To this end, the training set of each agent is partitioned to have a majority of samples belonging to a particular class, while the remaining part of the training dataset is uniformly distributed across all other classes by random sampling. Therefore the agents perform well only on a small subset of data, while on average they have low performances, as shown in 1.

The e2e-MEL training is evaluated on several vision classification tasks: digit classification on *MNIST dataset* [3], age estimation on *UTKFace dataset* , image classification on *CIFAR10 dataset* [4], and emotion detection on *FER2013* dataset [5]. The experiments are focused on the efficacy of e2e-MEL training as compared with the following widely adopted aggregation rules when paired with a pre-trained ensemble: *Unweighted Average*, which averages the softmax predictions over all agents; *Majority Voting*, which predicts the most-predicted class among agents; and *Random Selection*, which applies the unweighted average over k randomly selected agents.

Note in particular the increase of each ensemble's accuracy over the that of its constituent agents, which highlights the ability to form accurate predictions from weak base learners by filtering the best predictions for each sample. This demonstrates a noteworthy ability of e2e-MEL to leverage error diversity across ensemble models. This work demonstrates the integration of constrained optimization and machine learning models as a valuable toolset for not only enhancing but also combining machine learning models to improve performance on common tasks. We believe this is a very promising area and hope our work can motivate new solutions in which decision focused learning may be used to improve the capabilities of machine learning systems.

	Accuracy (%)				
Dataset	**e2e-MEL**	**UA**	**MV**	**RS**	**IA**
MNIST	**98.55**	96.91	95.99	96.93	89.6
UTKFACE	**90.97**	84.60	80.78	84.60	51.2
FER2013	**66.31**	63.89	63.15	63.89	47.8
CIFAR10	**64.09**	60.59	60.35	60.59	31.1

Table 1: e2e-MEL vs unweighted average (UA), majority voting (MV) and random selection (RS), along with the average among individual agents (IA).

ACKNOWLEDGMENTS

This research is partially supported by NSF grants 2007164 and 2232054, and NSF CAREER Award 2143706. Its views and conclusions are those of the authors only.

REFERENCES

[1] Mokhtar S Bazaraa, John J Jarvis, and Hanis D Sherali. 2008. *Linear programming and network flows*. John Wiley & Sons.

[2] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. 2020. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems* 33 (2020), 9508–9519.

[3] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142. https://doi.org/10.1109/MSP.2012.2211477

[4] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images.

[5] Kuang Liu, Mingmin Zhang, and Zhigeng Pan. 2016. Facial Expression Recognition with CNN Ensemble. In *2016 International Conference on Cyberworlds (CW)*. 163–166. https://doi.org/10.1109/CW.2016.34

[6] Ian H Witten, Eibe Frank, Mark A Hall, Christopher J Pal, and MINING DATA. 2005. Practical machine learning tools and techniques. In *Data Mining*, Vol. 2.