Deep Neural Network Classification under Resource Constraints

***Abstract*—This study proposes a novel approach to multi-class classification using neural networks under resource constraints, with applications in domains such as medical diagnostics, job allocation, and targeted marketing. This research develops an adaptive, resource-aware framework that employs a customized loss function, enabling decision-focused learning with resource constraints. We introduce an Integrated Prediction and Optimization (IPO) framework, designed to improve predictive accuracy while adhering to practical resource constraints, contributing a more effective solution for constrained classification tasks. Through a medical resource allocation use case, we demonstrate that the proposed approach effectively addresses challenges related to the efficient distribution of scarce resources in constrained decision-making scenarios, making it particularly relevant for real-world deep learning applications.**

***Index Terms*—Constrained-Based Learning, Adaptive Algorithm, Resource Constraints.**

I. Introduction

Classification tasks are fundamental to machine learning, playing a crucial role in decision-making processes across diverse domains such as healthcare, finance, and industrial manufacturing [1, 2]. These tasks often involve allocating limited resources based on predicted classifications, where errors can lead to severe consequences, such as the misallocation of critical medical resources or significant financial losses. This challenge is especially significant in multi-class classification problems, where resource constraints are linked to specific classes, requiring a careful balance between prediction accuracy and resource allocation.

Traditionally, classification and resource allocation are treated as separate stages within the *predict-then-optimize* (PO) framework [3, 4]. In this approach, a predictive model first generates class probabilities, which are then fed into an optimization model that allocates resources based on external constraints. While widely adopted, the PO framework inherently suffers from a disconnect between prediction accuracy and resource allocation efficiency, often leading to suboptimal overall performance [5]. Recently, several studies have introduced *Integrated Prediction and Optimization* (IPO) frameworks that integrate optimization constraints directly into the model's learning process [6, 7]. However, existing methods predominantly focus on binary classification or simple cost-sensitive scenarios, lacking the capability to effectively handle multi-class problems with complex constraints [8, 9]. Moreover, these methods often introduce unnecessary penalization, distorting the loss landscape and leading to inefficient model training [10].

Several studies apply the IPO approach by incorporating sensitive parameters into the learning process to address financial, operational, or societal impacts [11]. In many real-world scenarios, these costs can vary significantly depending on the type of error. For instance, in medical diagnostics, the cost of a false negative (failing to identify a diseased patient) is often much higher than that of a false positive [12]. Similarly, in fraud detection, missing a fraudulent transaction results in greater losses than mistakenly flagging a legitimate one [13]. Integrating these costs helps models minimize the most critical errors, aligning predictions with real-world priorities [14, 12]. However, a major limitation of these studies is their reduced effectiveness when misclassification costs are not precisely known or are expensive to obtain.

Another significant challenge in classification tasks, especially in multi-class scenarios, is data imbalance. Imbalanced datasets, where certain classes are underrepresented, can lead to biased models that perform poorly on minority classes. Weighted Cross Entropy (WCE) has emerged as a common technique to address this issue by influencing the learning process through the assignment of higher weights to minority classes [12]. WCE modifies the loss function to penalize misclassifications of minority classes more heavily, encouraging the model to pay closer attention to these underrepresented categories. However, studies employing the WCE technique to tackle imbalance often assume that class imbalance is a static weighting problem, with a constant ratio between the appearances of different classes as observed in the training data.

Whereas most existing studies integrate sensitive parameters into the learning process, applying the decision-focused learning approach, to address imbalance and misclassification costs in classification problems, only a few attempt to influence the learning process using sensitive parameters to manage constraints on the number of instances classified into specific classes, often arising from resource availability [14, 15]. However, these studies primarily address binary classification with classifiers designed for tabular datasets, such as decision trees and XGBoost. Moreover, they do not integrate resource constraints and classification decisions from the test dataset as part of the learning process. For instance, Shifman et al. [14] projected the resource constraints onto the training dataset, disregarding the classification decisions made on the test dataset and their alignment with the resource constraints.

In our research, we propose an approach that integrates resource constraints into multi-class classification problems directly within the training process of neural networks. This is achieved by iteratively adjusting the loss function based on real-time evaluations of resource constraints and classification decisions from the test dataset during model training. This innovative methodology diverges from existing approaches by embedding resource limitations, test dataset classification decisions, and the discrepancies between them as intrinsic components of the learning process. The iterative training and adjustment process allows the model to dynamically adapt to resource constraints, providing a robust and efficient solution for scenarios involving dynamically changing test datasets. The proposed framework is particularly well-suited for applications where limited resources must be carefully allocated across multiple classes, such as in healthcare prioritization or portfolio management scenarios [16, 14]. Through this innovative integration of prediction and optimization, our research contributes a significant advancement in the field of resource-constrained classification, enhancing the practical applicability and effectiveness of machine learning models in real-world settings. The main contributions include:

- Methodologically, we adopt the IPO approach to incorporate resource constraints into the learning process in a multi-class setting. Unlike existing methods, which primarily focus on single-class constraints or binary scenarios, our framework uniquely supports simultaneous constraints across multiple classes. This capability enables efficient allocation of limited resources in complex, real-world scenarios where multiple classes compete for constrained resources, making it a significant advancement over the current methods.
- In terms of training data enrichment, we utilize the classification decisions from the test set, without relying on the labels in the test dataset, to adjust the model's learning process to accommodate resource constraints. By leveraging only the predicted outcomes of the test samples, we maintain the integrity of the test data while enabling the model to adjust its parameters dynamically in response to resource constraints.
- From a modeling perspective, we apply the proposed methodology to neural networks by dynamically adjusting the loss function based on real-time evaluations of resource constraints and classification decisions from the test dataset. Leveraging neural networks allows us to address multi-class classification problems under resource constraints, particularly for image-based datasets.

These contributions collectively advance the literature on resource-constrained classification, offering a robust and effective solution for applications requiring optimal resource allocation across multiple classes.

II. Definitions and Problem Formulation

In our research, the goal is to develop an on-demand classifier that minimizes prediction errors while adhering to constraints on the number of classified instances in specific classes within the test dataset.. Assume a training dataset $D = \{x^{(i)}, y^{(i)}, i = 1, ..., N\}$, where $x^{(i)}$, denotes an observed sample i and $y^{(i)} \in \{0, 1\}^{1 \times K}$ is the true class label, subject to $\sum_{k=1}^{K} y_k^{(i)} := 1$, ensuring that each sample receives a single label among K possible classes. Assuming $\mathcal{M}$ is a classification model trained on the training dataset. Given observed samples in a test dataset $\Delta = \{x^{(j)}, y^{(j)}, j = 1, ..., M\}$, where the ground truth $y^{(j)}$ is unknown, the trained model returns a probability matrix $\hat{Y} \in (0, 1)^{|M| \times |K|}$, with an entry $\hat{y_t}^{(j)}$ representing the probability that a sample $x^{(j)}$ belongs to class t. Thus, the constraint regarding the number of classified instances for every class k that has a limited resources N_k can be written as follows:

$$\sum_{\forall j} \tilde{N_k}^{(j)} \leq N_k, \tag{1}$$

$$\text{where, } \tilde{N_k}^{(j)} = \begin{cases} 1, \text{if} \operatorname*{argmax}_t \hat{y_t}^{(j)} = k \\ 0, \text{otherwise} \end{cases}$$

The obtained $\tilde{N_k}^{(j)}, k = 1, ..., K$ make up the predicted class label vector for sample j, $\tilde{N}^{(j)} \in \{0, 1\}^{1 \times K}$, where $\sum_{k=1}^{K} \tilde{N}_k^{(j)} := 1$. Note that the classification decision is made in favor of the class with the maximum score. The resulting $\tilde{N}_i$ will be the amount of samples that the model predicted as class i.

In the next Section, we develop a framework which use the predicted class-dependent cost vector for minimizing the loss of the multi-class classification problem, while addressing the resource-constraints as an integrated part of model's learning phase.

III. Methods

A. Cost-Sensitive Loss Function for Multi-Class Classification Problems Under Resource Constraints

This section introduces a framework for minimizing the loss of the multi-class classification problem, while addressing the resource-constraint as an integrated part of model's learning phase. We propose an adaptive learning mechanism that iteratively adjusts a loss function using a predicted class-dependent cost matrix $C \in \mathbb{R}^{K \times K}$, where an entry $c_{l,k}$ represents the cost of labeling instance $x^{(j)}$ as class l when the actual class is k. We will utilize these costs to balance the classification decisions according to the resource limits of the constrained classes. The costs will be incorporated into the loss function, typically a continuous function of the predicted probabilities (e.g., hinge loss, exponential loss, or cross-entropy loss).

For this purpose, we will apply the following properties related to the predicted class-dependent cost matrix.

1) All costs are equal to or greater than zero, i.e., $c_{l,k} \geq 0$.
2) For each *constrained predicted class* l, the cost for correct classification will be equal to or lower than the misclassification costs, i.e., $c_{l,l} \leq c_{l,k}$, where $k \neq l$. This property essentially ensures that the cost of a true positive (TP) is smaller than or equal to the cost of a false positive (FP), aligning with the goal of minimizing the penalty for correct predictions while penalizing misclassifications appropriately.
3) For each *unconstrained predicted class* l, an all-ones vector $\mathbf{1}^{1 \times K}$ will be used to apply conventional continuous loss functions to the predicted probabilities. In these cases, the cost of correct classification will be equal to misclassification costs.
4) For each predicted class l, the misclassification costs are equal.

Consider an example of the class-dependent cost matrix C for a problem with three classes (1, 2, 3):

$$C = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0.5 & 2 \\ 1 & 1 & 1 \end{bmatrix}.$$

From this matrix, it can be inferred that class 2 is constrained, while classes 1 and 3 are unconstrained. Specifically, the diagonal value for class 2, $c_{2,2} = 0.5$, is smaller than the off-diagonal values $c_{2,1} = 2$

and $c_{2,3} = 2$, indicating that the cost of a true positive (TP) for class 2 is lower than the cost of a false positive (FP). This configuration suggests a focus on limiting misclassifications into class 2, aligning with the resource-constrained nature of this class. On the other hand, for classes 1 and 3, the values in their corresponding rows are uniform, with $c_{1,1} = c_{1,2} = c_{1,3} = 1$ and $c_{3,1} = c_{3,2} = c_{3,3} = 1$, indicating that these classes are unconstrained and follow the standard cost-sensitive framework without prioritization or restrictions.

During the training process, the classifier weights and biases θ are modified in order to change the class probabilities such that the loss function will be minimized. We formulate the mean loss over the training set by:

$$E(\theta, C) = \frac{1}{N}\sum_{i=1}^{N} l(y^{(i)}, \hat{y}^{(i)}). \tag{2}$$

We propose illustrating the modification of the loss function by integrating class-dependent costs into the commonly used Cross Entropy, as follows:

$$l(y^{(i)}, \hat{y}^{(i)}) = -\sum_{l=1}^{K}\sum_{k=1}^{K}\Big(y_k \cdot \log\big[1 + (\hat{y}_k^{c_{l,k}} - 1) \tag{3}$$
$$\cdot\big\{1 - \tanh\big(\alpha \cdot \big(\sum_{t=1}^{K} \hat{y}_t \cdot \frac{e^{\beta \hat{y}_t}}{\sum_{j=1}^{K} e^{\beta \hat{y}_j}} - \hat{y}_l\big)\big)\big\}\big]\Big).$$

For simplicity of representation, the notations $\hat{y}_k^{(i)}$ and $y_k^{(i)}$ were replaced by $\hat{y}_k$ and y_k in eq. 3. The term $\sum_{t=1}^{K} \hat{y}_t \cdot \frac{e^{\beta \hat{y}_t}}{\sum_{j=1}^{K} e^{\beta \hat{y}_j}}$ is a soft, continuous and differentiable approximation of the function $\max_{t \in \{1,\ldots,K\}} \hat{y}_t$. As the parameter β increases, the expression converges towards the behavior of the maximum function. We use the hyperbolic tangent $f(x) = \tanh(\alpha \cdot x)$, where $x \in [0, 1]$, to obtain $f(0) = 0$ and $f(x) \longrightarrow 1, \forall x > \epsilon$, where ϵ is close to 0. ϵ directly depends on α and β parameters.

For simplicity, we assume a single resource constraint for a specific class λ, i.e., $\sum_{\forall j} \tilde{N}_{\lambda}^{(j)} = \tilde{N}_{\lambda} \leq N_{\lambda}$. This assumption illustrates a situation where a decision-maker is looking to classify instances for a specific class under a limited number of classifications. This choice often reflects a realistic situation in which a decision-maker seeks the optimal resource allocation under their authority. For example, in a task of job allocation, assuming a specific group manager of a company will seek to allocate candidates, according to their background and skills, to a limited number of open positions in his group, with a high potential to succeed (e.g., qualification and match).

The loss function can be simplified and observed as:

$$l(y^{(i)}, \hat{y}^{(i)}) \approx \begin{cases} -\sum_{k=1}^{K} y_k \cdot c_{\lambda,k} \log(\hat{y}_k), & \text{if } \max_t \hat{y}_t = \hat{y}_{\lambda}, \\ -\sum_{k=1}^{K} y_k \cdot \log(\hat{y}_k), & \text{if } \max_t \hat{y}_t \neq \hat{y}_{\lambda}. \end{cases}$$

It should be observed that the utilization of hyperbolic tangent functions allows for a separation between the expressions for the predicted constrained class and the predicted unconstrained classes in the loss function, $l(y^{(i)}, \hat{y}^{(i)})$. The proposed loss function is a modified version of the Cross-Entropy that incorporates a cost-sensitive approach only for the predicted constrained classes. It is important to note that this modification allows us to penalize only the misclassifications of the predicted constrained classes at the expense of the correct classifications.

B. Explication of the Binary Case in the Loss Function

In this section, we illustrate the application of the previously described loss function in binary classification scenarios in which the positive class is the resource-constrained class. Specifically, this loss function can be divided into four key elements: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

As depicted in Figure 1, the plot consists of four subplots that clarify the relationship between the loss function and the model's probabilistic prediction of the actual class label, parameterized by various values of $c_{1,0}$ and $c_{1,1}$. Note that $c_{0,0}$ and $c_{0,1}$ is 1. Specifically, for predictive probabilities less than 0.5, the function represents the loss obtained from FP or FN classifications. Conversely, for probabilities exceeding 0.5, the function represents the loss of TP and TN classifications. The magnitude of the change in the loss values during the transition from FP to TP is influenced by the size of the difference between the cost of $c_{1,0}$ and $c_{1,1}$. By increasing the

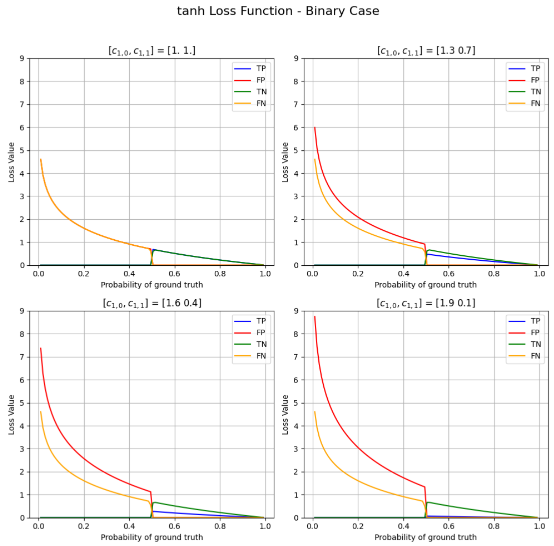

Fig. 1: Loss function values as a function of the predicted probability of the ground truth label for different cost values $[c_{1,0},\, c_{1,1}]$.

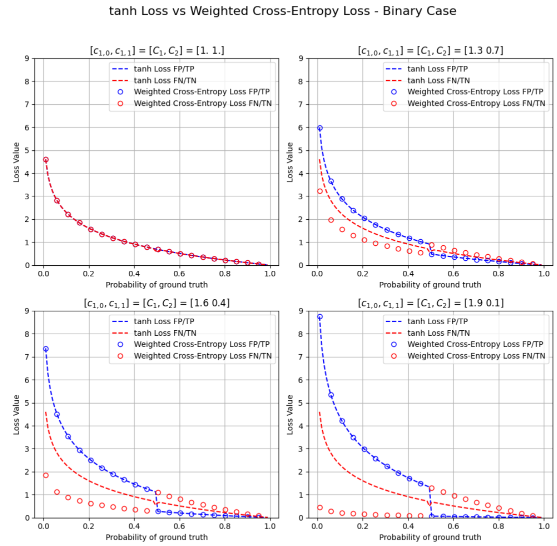

Fig. 2: Comparative Analysis of Weighted Cross Entropy and the Proposed Loss Function in a Binary Context. $[C_1, C_2]$ = [Unconstrained Class, Constrained class].

penalty cost of false positives while simultaneously decreasing the cost of true positives, it becomes possible to control the number of classifications for the constrained class, thereby meeting the resource constraint. The transition from FN to TN exhibits a near-continuous behavior around the probability of the ground truth of 0.5.

From Figure 1, it can be observed that the proposed loss function behaves similarly to the conventional cross-entropy loss function when predicting the unconstrained class and similarly to the Weighted Cross-Entropy (WCE) [12] when predicting the constrainted class as follows:

$$WCE = -C_1 \cdot y_1 \cdot \log(\hat{y_1}) - C_2 \cdot y_2 \cdot \log(\hat{y_2}) \quad (4)$$

In Figure 2 we compare our proposed loss function with the weighted cross-entropy (WCE) loss in a binary classification setting. Each subplot shows a different configuration of the cost vector, $[C_1, C_2]$, highlighting the effects of adjusting weights for constrained versus unconstrained classes.

One key observation is that in the FP and TP regions, both our loss function and WCE behave similarly. However, a noticeable difference arises in the FN and TN regions.

In WCE, increasing the penalty for the constrained class by modifying C_2 inadvertently distorts the loss for the non-constrained class in the FN/TN regions. Specifically, this adjustment decreases the penalty for False Negatives and increases the penalty for True Negatives, leading to an imbalance. This distortion undermines the loss function's effectiveness, as it reduces the penalty for FN cases (where higher penalization might be needed) and raises the penalty for TN cases (which ideally should be lower or unaffected for accurate classification).

This unintended imbalance in WCE can lead to suboptimal model behavior, especially in scenarios with resource constraints, as it forces trade-offs between classes that should ideally remain independent. In contrast, our proposed loss function eliminates this issue by preserving the standard cross-entropy behavior for the non-constrained class in the FN/TN regions. This selective adjustment ensures that we can penalize the constrained class without impacting the loss values for the unconstrained class, thus providing a more targeted and stable approach for handling resource-limited settings.

C. Adaptive Learning Algorithm for Satisfying Resource Constraints

To integrate resource constraints into training, we design an adaptive mechanism that iteratively adjusts the class-dependent cost matrix C. Initially, all elements of C are set to 1, aligning the loss function with standard cross-entropy for conventional training. If resource constraints are met after this initial training, no further adjustments to C are needed.

In this context, each "iteration" refers to a complete training process that continues until convergence. The pseudo-code of the adaptive learning algorithm for satisfying resource constraints (under the assumption of single constrained class in a multiclass classification task) is presented in Algorithm 1. The following function,

$$F(N_\lambda, \tilde{N}_\lambda) = (N_\lambda - \tilde{N}_\lambda)^2 \cdot (\tanh(\gamma(\tilde{N}_\lambda - N_\lambda)) + 1),$$

illustrated in Figure 3, adjusts C based on the model's behavior with respect to the resource-constrained class. The graph also shows the gradient of F, illustrating how the penalty increases as the count $\tilde{N}_\lambda$ approaches or exceeds the constraint N_λ. The tuning parameter γ controls the steepness of this penalty, allowing for fine adjustments to the sensitivity of the cost vector update.

This iterative algorithm ensures that the model can be trained under resource constraints effectively, allowing penalization only when necessary and targeting the constrained class specifically without distorting the behavior for other classes.

IV. Experiments

This section examines the behavior of the proposed custom loss function and presents the findings from the evaluation of the IPO method, comparing them with the results obtained using the conventional PO approach. In the PO method, classification in the second phase is based on predicted probabilities generated by EfficientNet with cross-entropy loss, ensuring compliance with the constraints presented in Eq. 1. The classification process in this phase is performed using a generalized version of the heuristic

---

Algorithm 1: Adaptive Training with Resource Constraints

Input: Allowed resource constraints N_λ, learning rate μ, convergence criterion (e.g. meeting constraints)
Output: A model that is meeting resource constraints

Initialize Training:;
Set C as a matrix of ones;
Train the model using the custom loss function, which at this stage is equivalent to the cross-entropy loss, and continue training until the model reaches convergence. ;
Evaluate Resource Constraints:;
Test the model on a test dataset;
Estimate the predicted count $\tilde{N}_\lambda$ of samples classified into the resource-constrained class λ;
Compare $\tilde{N}_\lambda$ with the allowed count N_λ;
Adjust Costs if Necessary:;
if $\tilde{N}_\lambda > N_\lambda$ **then**
    Compute the adjustment function

$$F(N_\lambda, \tilde{N}_\lambda);$$

Update C **Using Gradient Descent:**;
Update the cost $c_{\lambda,k}$ for the non-constrained classes using:

$$c_{\lambda,k} = c_{\lambda,k} - \mu \cdot \frac{\partial F}{\partial \tilde{N}_\lambda}, \forall k \neq \lambda$$

Repeat if Constraints Are Not Met:;
while *resource constraints are not met* **do**
    Retrain the model with the updated cost matrix C;
    Reevaluate $\tilde{N}_\lambda$ and N_λ;

---

approach proposed in [14] for binary classification, extended to the multi-class classification problem. The generalization follows a straightforward rule: a sample is assigned to the class with the highest predicted probability unless that class has already met its constraint, in which case the next highest probability class is considered instead.

A. Dataset

The dataset used in this study consists of 4,130 pairs of knee joint X-ray images provided by [17], who reprocessed the original data from the Osteoarthritis Initiative (OAI), available at https:

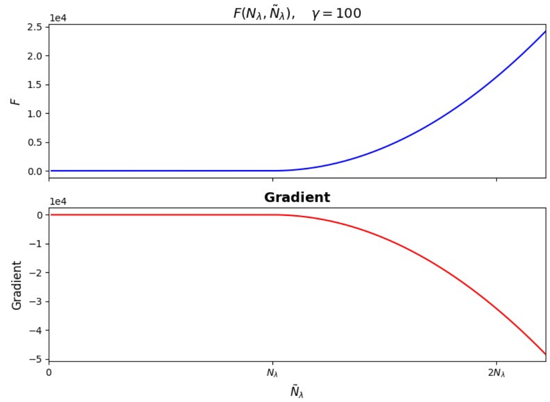

Fig. 3: Function and Gradient Illustration for Integrating Resource Constraints into Training.

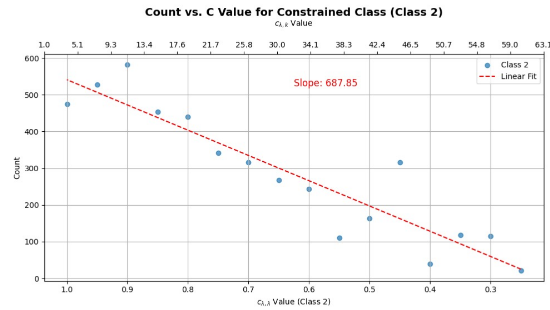

Fig. 4: Relationship between constrained and unconstrained weights ($c_{\lambda,\lambda}$, $c_{\lambda,k}$ respectively) and their impact on the predicted count for Class 2. In this example $\lambda = 2$. The primary x-axis represents the constrained weight values, while the secondary x-axis corresponds to the unconstrained weight scale.

//nda.nih.gov/oai/. Knee osteoarthritis (OA) is a prevalent condition among the elderly, often leading to reduced mobility and physical disability. Early detection and intervention can help slow disease progression, and enable a more effective allocation of medical resources. To support this goal, we classified patients into five severity levels based on OA progression and used these classifications to guide the distribution of limited treatment resources. The dataset was randomly split into training, validation, and test sets in a 7:1:2 ratio.

B. Illustration of our custom loss function behavior

Figure 4 illustrates the impact of adjusting class weights on the model's classifications for the constrained class. As the weight of the unconstrained classes increases (upper X-axis), the number of samples predicted for the constrained class decreases due to the increasing false positives penalty, as shown in Figure 1. These results support the hypothesis that modifying class weights effectively influences the model's prediction behavior for constrained classes.

C. Setup and Results

In our experiments, we evaluated the classification performance of the proposed IPO method in comparison to the PO approach under varying constraint levels for Class 3. Figure 5 illustrates the accuracy trends for Class 3 across varying constraint levels, ranging from 40% to 90% of patients with severity level 3 in the test dataset. The blue curve represents the accuracy obtained using the conventional PO method, while the red curve shows the results of our IPO approach. Each data point corresponds to an accuracy measurement under a specific constraint level, with numerical values indicating the constraints for Class 3 under that setting. The last point at each constraint level represents the convergence of the final solution based on the stopping criterion. It can be observed that the IPO method achieves higher accuracy under less restrictive constraints (e.g., 70%–90%), while attaining higher accuracy at the second-to-last point before convergence under stricter constraints (e.g., 40%–60%).

V. Conclusion

This paper introduces an advanced computational learning method for multi-class classification problems under resource constraints in deep neural networks. The approach incorporates custom loss functions to integrate constraints directly into the learning process. The mathematical formulation validates the suitability of the proposed custom loss function's structure for the problem at hand. Experiments validate the expected theoretical behavior of

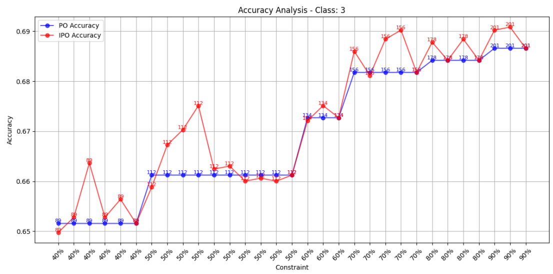

Fig. 5: Accuracy comparison of PO (blue) and IPO (red) methods for Class 3 under varying constraint levels.

the proposed loss function, showing that adjusting class weights effectively influences predictions for constrained classes. Furthermore, the results show that the proposed IPO method outperforms the conventional PO method across different constraints.

While our study presents promising results, several limitations and directions for future research should be considered. First, the stopping criterion in the adaptive algorithm is designed to terminate iterations once the constraint is satisfied. However, exploring alternative stopping criteria based on the difference between actual classifications and constraints may improve performance. Additionally, our experiments focus on a single-class constraint scenario. A comprehensive analysis across datasets and constraints is needed for better generalizability.

References

[1] D. Pessach, G. Singer, D. Avrahami, H. C. Ben-Gal, E. Shmueli, and I. Ben-Gal. “Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming”. In: *Decision Support Systems* 1132 (2020).

[2] Hao Wu, Xi Zhang, and Jufeng Yang. “Deep Learning-Based Encrypted Network Traffic Classification and Resource Allocation in SDN”. In: *Journal of Web Engineering* (2021), pp. 2319–2334.

[3] Xuecheng Tian, Ran Yan, Yannick Liu, and Shuaian Wang. “A smart predict-then-optimize method for targeted and cost-effective maritime transportation”. In: *Transportation Research Part B: Methodological* 172 (2023), pp. 32–52.

[4] Kaiwen Xia, Li Lin, Shuai Wang, Haotian Wang, Desheng Zhang, and Tian He. “A Predict-Then-Optimize Couriers Allocation Framework for Emergency Last-mile Logistics”. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023, pp. 5237–5248.

[5] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. “Rebalancing bike sharing systems: A multi-source data smart optimization”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1005–1014.

[6] Adam N Elmachtoub and Paul Grigas. “Smart “predict, then optimize””. In: *Management Science* 68.1 (2022), pp. 9–26.

[7] Othman El Balghiti, Adam N Elmachtoub, Paul Grigas, and Ambuj Tewari. “Generalization bounds in the predict-then-optimize framework”. In: *Advances in neural information processing systems* 32 (2019).

[8] Bryan Wilder, Bistra Dilkina, and Milind Tambe. “Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 1658–1665.

[9] Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. “Differentiation of blackbox combinatorial solvers”. In: *International Conference on Learning Representations*. 2019.

[10] Hao Zhou, Shaoming Li, Guibin Jiang, Jiaqi Zheng, and Dong Wang. “Direct heterogeneous causal learning for resource allocation problems in marketing”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 4. 2023, pp. 5446–5454.

[11] Yixin Guo et al. “Real-World Weight Cross-Entropy Loss: A New Perspective for Classification”. In: *arXiv preprint arXiv:2001.00570* (2020). URL: https://arxiv.org/abs/2001.00570.

[12] Yuri Sousa Aurelio, Gustavo Matheus De Almeida, Cristiano Leite de Castro, and Antonio Padua Braga. “Learning from imbalanced data sets with weighted cross-entropy function”. In: *Neural processing letters* 50 (2019), pp. 1937–1949.

[13] Yusuf Sahin, Serol Bulkan, and Ekrem Duman. “A cost-sensitive decision tree approach for fraud detection”. In: *Expert Systems with Applications* 40.15 (2013), pp. 5916–5923.

[14] Danit Abukasis Shifman, Izack Cohen, Kejun Huang, Xiaochen Xian, and Gonen Singer. “An adaptive machine learning algorithm for the resource-constrained classification problem”. In: *Engineering Applications of Artificial Intelligence* 119 (2023), p. 105741.

[15] Toon Vanderschueren, Bart Baesens, Tim Verdonck, and Wouter Verbeke. “A new perspective on classification: optimally allocating limited resources to uncertain tasks”. In: *Decision Support Systems* 179 (2024), p. 114151.

[16] Yu Chen, Wenbing Jin, Limin Zheng, Xiaojie Xiu, Yudong Huang, Weijie Qiu, and Yudong Li. “Closed-Loop Dynamic Blending Optimization Based on Variational Bayesian and its Application in Industry”. In: *IEEE Access* 11 (2022), pp. 494–505.

[17] Pingjun Chen, Linlin Gao, Xiaoshuang Shi, Kyle Allen, and Lin Yang. “Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss”. In: *Computerized Medical Imaging and Graphics* 75 (2019), pp. 84–92.