Knowledge Distillation-driven Communication Framework for Neural Networks: Enabling Efficient Student-Teacher Interactions

Abstract

This paper presents a novel framework for facilitating communication and knowledge exchange among neural networks, leveraging the roles of both students and teachers. In our proposed framework, each node represents a neural network, capable of acting as either a student or a teacher. When new data is introduced and a network has not been trained on it, the node assumes the role of a student, initiating a communication process. The student node communicates with potential teachers, identifying those networks that have already been trained on the incoming data. Subsequently, the student node employs knowledge distillation techniques to learn from the teachers and gain insights from their accumulated knowledge. This approach enables efficient and effective knowledge transfer within the neural network ecosystem, enhancing learning capabilities and fostering collaboration among diverse networks. Experimental results demonstrate the efficacy of our framework in improving overall network performance and knowledge utilization.

INTRODUCTION

Knowledge forms the foundation of intelligence, enabling AI systems to comprehend and navigate the complexities of the world. In the realm of AI, knowledge extends beyond mere data to encompass the collective insights and expertise acquired by neural networks. This paper addresses the challenge of optimizing knowledge utilization and communication among neural networks through a novel framework. By adopting the roles of students and teachers, the neural networks within the proposed framework engage in a collaborative exchange of knowledge. When confronted with new data, a network assumes the role of

a student, seeking guidance from teachers who have already been trained on the incoming information. Leveraging knowledge distillation techniques, the student network absorbs the distilled wisdom of its teachers, effectively augmenting its own knowledge base. This framework facilitates efficient knowledge transfer, empowering AI systems to leverage the collective intelligence of the network ecosystem for enhanced learning and improved performance.

Existing frameworks for teacher-student networks commonly rely on DNN retraining and online adaptation as their foundation. These frameworks can be further enhanced by integrating techniques like Knowledge Distillation (KD) [1], or Lifelong Learning (LLL) [2]. KD methods, focus on transferring knowledge from a teacher network to a student network. LLL algorithms enable the retraining of pre-trained neural networks on new tasks while preserving knowledge from previously learned tasks. The evaluation of knowledge in these frameworks typically revolves around performance metrics, such as accuracy and success rates for each dataset class. However, measuring knowledge primarily relies on assessing the performance of the teacher or student agents, often using the entire test dataset as a rough indicator of the knowledge's reliability. This approach may not provide a comprehensive understanding of the encoded knowledge within the models and its true reliability.

The motivation behind developing a network of nodes, where each node represents a neural network, stems from the desire to enhance the learning capabilities and overall performance of individual networks. In traditional isolated neural networks, the learning process is limited to the specific data they have been trained on. However, by establishing a network of interconnected nodes, the potential for knowledge sharing and collaboration emerges. Each node within this network can act as both a student and a teacher, enabling the exchange of information and experiences among the neural networks. The communication between these nodes serves the purpose of facilitating the learning of new tasks from one another. When a node encounters a task for which it has not been trained, it can assume the role of a student. By sending the new data to other nodes in the network, the student node seeks guidance from those nodes that have already acquired knowledge in the given task. This knowledge transfer is crucial as it allows the student node to leverage

the collective expertise of the network, accelerating its learning process and enhancing its performance.

Moreover, the use of knowledge distillation techniques in this communication framework further augments the learning process. By distilling the knowledge from the teacher nodes, the student node can absorb and integrate the distilled wisdom, benefiting from the accumulated experiences and insights of the network. Overall, the development of a network of nodes with defined communications fosters a collaborative learning environment for neural networks. It harnesses the power of collective intelligence, enabling networks to effectively learn new tasks from one another. This approach has the potential to advance the capabilities of individual networks, leading to improved performance, enhanced knowledge retention, and the ability to tackle complex tasks more efficiently.

This paper introduces a novel framework that creates an unconstrained environment where multiple neural networks can operate as both students and teachers. This approach allows for dynamic interactions among networks, enabling knowledge sharing and collaboration. By breaking away from the traditional isolated network paradigm, the framework opens up new possibilities for enhancing learning capabilities and performance. The paper demonstrates that the overall knowledge within the framework continually increases as new data flow to random networks. By leveraging the framework's communication structure, the knowledge dissemination process is facilitated, allowing networks to learn from each other's experiences. This continuous flow of new data and knowledge contributes to the collective intelligence of the network ecosystem, resulting in improved performance and enhanced knowledge retention across the entire framework. The paper provides a clear and strict definition of the messages sent by different nodes within the framework. By precisely defining the communication protocols, the paper ensures efficient and effective knowledge transfer between networks. This clarity and specificity in message exchange facilitate seamless interactions and minimize potential ambiguity or misinterpretation, thereby improving the overall performance and reliability of the framework. In summary, the paper's contributions include the introduction of an unconstrained environment for multiple networks to function as students or teachers, the establishment of a framework where the overall knowledge continually grows with the influx of new data, and the definition

of strict message protocols to enable efficient communication between nodes. These contributions advance the understanding and utilization of knowledge in neural network frameworks, paving the way for improved learning, collaboration, and performance in AI systems.

I. Related work

Teacher-Student Learning

The concept of teacher-student learning emerged from the idea of compressing the knowledge held by one or a group of deep neural networks (DNNs) into a single DNN student network [3]. Teacher-student learning strikes a balance between efficiency and performance [4]. In a study [5], a large-size pre-trained network acts as a teacher, providing labels for unlabeled data. Notably, [1] demonstrated remarkable results by distilling knowledge from an ensemble of models into a single student model. Building upon the research of [5] and [1], subsequent studies have explored teacher-student interactions to enhance the process of knowledge distillation, resulting in students exhibiting strong performances [6], [7], [8], [9], [4], [10].

In a study by [9], a novel framework is proposed that compresses wide and deep networks into thinner and deeper ones. This framework, referred to as FitNets, leverages the outputs and intermediate representations learned by the teacher network to train a deeper and narrower student network, resulting in improved training process and student performance. [6] define the distilled knowledge as the flow of the solving procedure learned through intermediate layer feature representation. By considering this, the student DNN outperforms the teacher DNN. [7] demonstrate significant performance improvement in the student network by training it to learn and mimic the attention maps of a powerful teacher network. This approach enforces the student network to focus on relevant features and enhances its performance. [8] introduce a framework where a teacher network is employed to train a lightweight student network for prediction, leading to improved performance of the student network. [4] propose a redundant teacher-student framework consisting of a static teacher network, a static student network, and a continuously adapting student network. This framework enhances student robustness against adversarial attacks, demonstrating its effectiveness in improving network

resilience. More recently, [10] present a Teacher-Student network framework based on lifelong learning. In this framework, the Teacher network provides the Student with information learned in the past, while the Student is simultaneously trained with new data, enabling knowledge retention and adaptation to new tasks. In summary, these studies contribute to the field by introducing innovative frameworks and techniques such as FitNets for compressing networks, attention map learning, teacher-student collaboration, and leveraging lifelong learning. These approaches enhance the training process, improve student performance, and increase network robustness in the face of adversarial attacks, demonstrating the potential of teacher-student frameworks in advancing the capabilities of neural networks.

Last but not least, efforts are being made on integrating these techniques to multiple teacher learning. [11] proposes a novel framework, where the option to use multiple teachers is offered. They provide an analytical work that summarises some of the most frequently used distillation techniques and the integrate the framework for multiple teachers and different architectures.

Out Of Distribution Detection

Deep learning models often struggle to perform well on real-world tasks when faced with training and test data distributions that differ significantly. Typically, a deep neural network (DNN) is trained on labeled data and tested on unknown test data during inference, assuming both datasets share the same label space. However, when the DNN encounters data points during inference that contain classes not encountered in training, it tends to provide incorrect predictions and conclusions. To address this issue, Out-Of-Distribution (OOD) detection algorithms aim to predict whether a test example belongs to a different distribution than the training data or if it belongs to the same distribution. This problem can be formulated as a binary classification task, where the input $\mathbf{x} \in \mathcal{X}$ and label $y \in \mathcal{Y} = \{1, ..., K\}$ are both random variables following a joint data distribution $P_{in}(\mathbf{x}, y) = P_{in}(y|\mathbf{x})P_{in}(\mathbf{x})$. A classifier $P_\theta(y|\mathbf{x})$ is assumed to be trained on a dataset drawn from $P_{in}(\mathbf{x}, y)$, where θ represents the model parameters. The terms $P_{out}(\mathbf{x})$ and $P_{in}(\mathbf{x})$ represent the out-of-distribution and in-distribution, respectively. The objective is to determine whether an input $\mathbf{x}$ belongs in P_{in} or P_{out} and design a detector $g(\mathbf{x}) : \mathcal{X} \rightarrow \{0, 1\}$,, which assigns the label 1 if the

data is from the in-distribution and the label 0 otherwise [12]. In recent years, specialized OOD detection algorithms have been developed to identify whether test data points are out-of-distribution during inference. A common baseline approach for OOD detection is to gather statistics on the maximum softmax output probabilities for correctly and incorrectly classified predictions using a validation dataset. These statistics are then used to define a threshold, which is applied during the model inference process ([13]). Typically, examples that are correctly classified tend to have higher maximum softmax probabilities compared to misclassified examples, allowing for their identification.

Another approach involves utilizing Generative Adversarial Networks (GANs) to generate samples from the out-of-distribution P_{out}, referred to as "boundary" samples that lie in the low-density region of the in-distribution P_{in}. Simultaneously, a DNN classifier is trained to produce a low softmax value for the expected class when presented with outlier data during training or out-of-distribution data during inference. To detect OOD data, the softmax class prediction probability is compared against a threshold [12]. In some cases, a neural network classifier is trained to provide confidence estimates for each input, which are then utilized to differentiate between in-distribution and out-of-distribution examples [14]. If the confidence estimate is less than or equal to a certain detection threshold, the input is considered out-of-distribution. A one-class classifier is employed as a detector, trained on the output of an early layer of the original pre-trained DNN classifier, using its original training set to discriminate between in-distribution and out-of-distribution data [15]. Additionally, Likelihood Regret is proposed as an efficient OOD score for generative models based on Variational Auto-encoders (VAE). It is defined as the logarithmic ratio between the likelihood obtained by the posterior distribution optimized separately for a given input and the likelihood approximated by the VAE [16]. Overall, these various approaches contribute to improving OOD detection in deep learning models by leveraging different techniques such as thresholding on softmax probabilities, GAN-generated samples, confidence estimates, one-class classifiers, and likelihood-based scores. These advancements aim to enhance the robustness and reliability of deep learning models in handling out-of-distribution data scenarios.

NEURAL NETWORK COMMUNICATION FRAMEWORK DESCRIPTION

The Neural Network Communication Framework (NNCF) defines the protocol via the Framework's nodes cooperate together to learn tasks from other nodes. Each node contains an out of distribution detector, as described in [16] and a classifier. So, every node is aware of the tasks it has learned. Out of distribution detectors can automatically answer the question of which are the possible teachers when facing a new task.

Node

Our framework consists of nodes, each node contains two discrete parts, an out of distribution detector and a classifier. All nodes can be either students or teachers, depending on their knowledge on a given task. Figure 1 represents the structure of a node. Each node is treated as a unique entity and by using teacher-student learning techniques it interacts with others. Obviously, this framework has the possibilities of a valuable asset in the deep learning field as it emulates a human-like environment, where a stimulus of unknown data trigger the urge to search for teachers, experts on the specific task, and learn from them.

Distillation-driven Communication Framework for Neural Networks: Enabling Efficient Student-Teacher Interactions

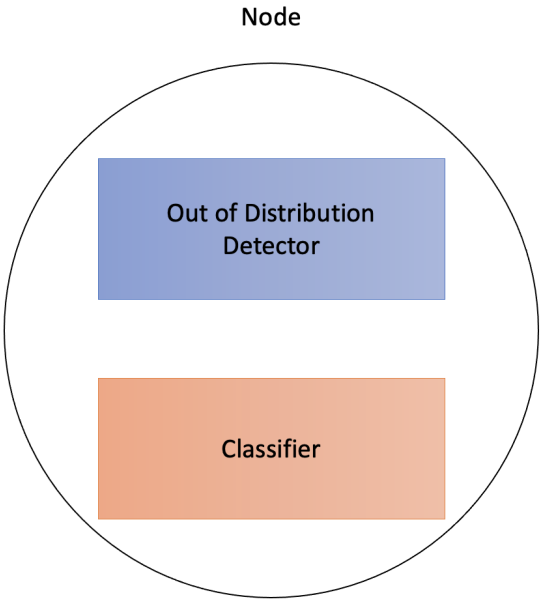

Fig. 1. The representation of a framework's node, containing an out of distribution detector and a classifier

Learning a new Task

As mentioned earlier, every time a stimulus is given to a node, the first question that needs to be answered is if the node is aware of the current data-stream. This question is answered through the node's out of distribution detector. The percentage of data needed and also the in or out of distribution threshold are given as hyperparameters and should be defined for each dataset. If the node has already learned the task, no action is taken. On the other hand, if the task is proven to be unknown, the current node should automatically try to search for a teacher within the Framework. Data are automatically sent to all nodes, each node should reply if this set of data is in or out of its distribution. All nodes that replied positively, are considered possible teachers from this point. The framework do not support multiple teacher learning yet, so out of all possible teachers the node chooses the one with higher accuracy score on the given data.

The framework aims to support a plethora of teacher-student learning techniques such as, knowledge distillation, continual learning and multi-teacher learning. For the time being, the work in [11] is utilised in order to distill the teacher's knowledge. To this end, four options are offered to the student:

1) **Training Data**: The data used to train the teacher are sent to the student.
2) **Soft-Output Knowledge Distillation**: The teacher provides the activation of the output, when passing the given data.
3) **Feature Knowledge Distillation**: The teacher provides the activation of the output and also some pre-defined intermediate layers, when passing the given data.
4) **Copy Teacher's Weights**: The teacher provides the classifier's weights and architecture.

Figure 2 illustrates the process described that enables a node to learn a new task when triggered by a stimulus of unknown data. As can be seen, the pipeline emulates the human behaviour. The self-awareness of each node's knowledge is the key to trigger learning a new task, or teaching an already known task to others. To this end, each node can operate in four states:

1) **Stable state**: The node does not exchange information with other nodes or takes place in a learning process.

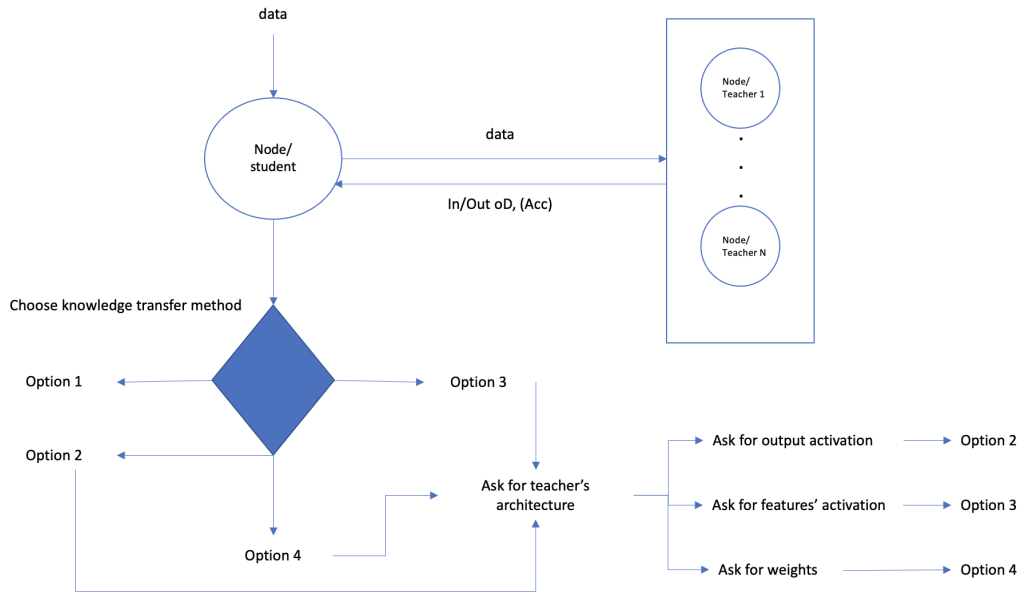

Fig. 2. The illustration of the process for learning new tasks

2) **Awareness state**: The node checks if it has been trained on a given dataset. Here, out of distribution detectors are used.
3) **Student state**: The node has recognised it is not aware of the new data and initiates the process to learn the new task.
4) **Teacher state**: The node is aware of the dataset sent from another node (in student state), and has been chosen to teach the current task.

Figure 3 illustrates the transitions between the different states for a single node, according to the different framework triggers. Here we can see the difference between a trigger form the framework and a trigger from another node. When the framework triggers a node, it automatically switches to the awareness state, if the node already knows this dataset it switches back to stable state. Whereas, if the nodes is not aware of this dataset it switches to student state. On the other hand, if a node receives a trigger from another node, it enters awareness state, but it switches to Teacher state if the dataset is known and to Stable otherwise.

The framework should have a strict communication protocol in order to function properly. To this end, for each type of communication the files exchanged should have specific format. Regarding the exchange of

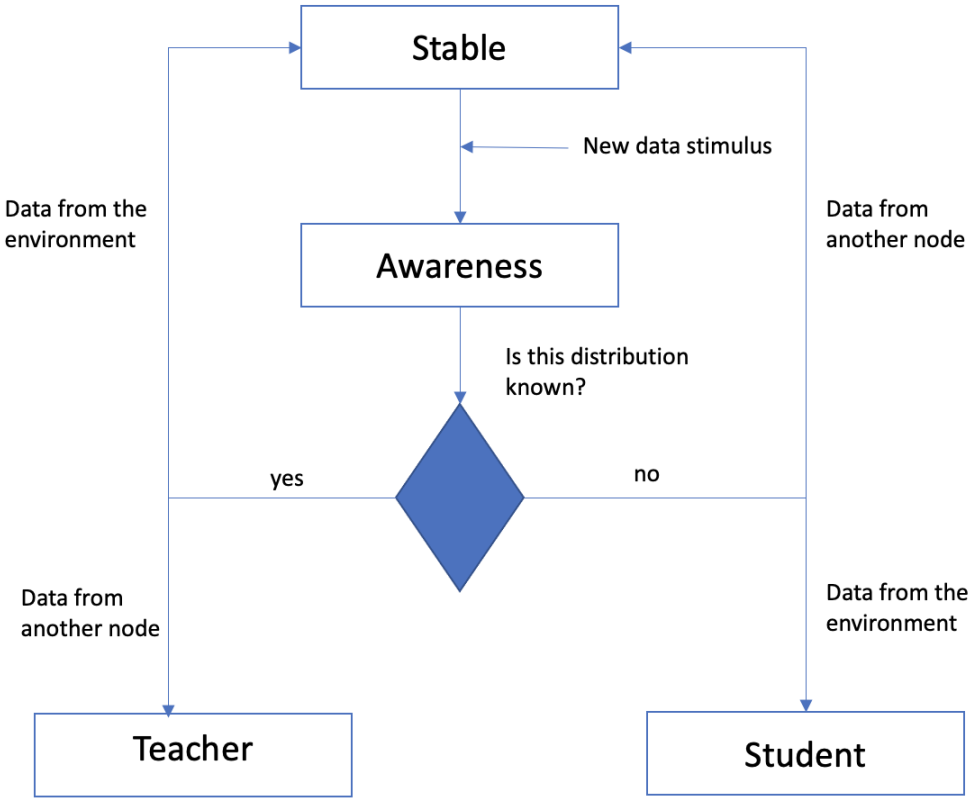

Fig. 3. The illustration of the transitions between states

data, a folder contains all pictures, which are part of the dataset and a CSV file, where each row represents a sample (image name and label) is sufficient. The response of the out of distribution detector and the accuracy score is implemented using a CSV file, where the first column is True or False (in or out of distribution) and the second column is filled with the accuracy score. The request of the learning option by the student is implemented using a CSV file containing a single element (the id of the requested option 1-4). The architecture of the teacher is adopted by the student, retrieved by an architecture library. The teacher's weights are sent via a PTH file and can be applied directly as the architecture is known, when option 4 is applied. Regarding option 2, where the output activation needs to be sent, the PT file contains a $M \times C$ matrix, where M is the total number of samples and C is the total number of classes. Concerning

the feature distillation option 3, the number of PT files needed is proportional to the number of feature layers used for the distillation process. Each one of these files, named after the respective feature layer, contains the layer's activation as a $M \times K$ tensor matrix, where K is the size of the layer's activation flattened. Thus, the knowledge can be distilled offline and does not require any additional messages.

Nodes' Communications

The nodes' communications are achieved using specific files and folders. Supposing that every time a node sends information to another node one message is sent, the most expensive option requires $2(N-1)+4$ messages, where N is the total number of nodes in the framework. The most expensive options are the Knowledge Distillation ones. When a new stimulus arrives, and the node attempts to learn the specific task, $2(N-1)$ messages are needed for the data transmission to all nodes and their responses (in or out of distribution and accuracy scores). Two messages are required for the request and the reception of the teachers' architecture and finally, two messages needed for the request and the reception of the teacher's activation outputs.

EXPERIMENTAL RESULTS

The experiments conducted, regarding the framework's functionalities, aim to prove the proper framework operation and present its potential. For the first experiment needs, five nodes were initiated:

1) Node with out of distribution detector (FMNIST), classifier (FMNIST) with 0.7137% accuracy.
2) Node with out of distribution detector (MNIST), classifier (MNIST) with 0.8562% accuracy.
3) Node with no out of distribution detector or classifier.
4) Node with out of distribution detector (FMNIST), classifier (FMNIST) with 0.6929% accuracy.
5) Node with no out of distribution detector or classifier, that will be triggered with new data (student).

As the human-like functionality of the framework suggests, a stimulus of FMNIST data is given to node 5. Node 5, having no out of distribution detector, tempts to search for a teacher. Two possible teachers are detected, node 1 and node 4, and as expected node 1 is chosen as teacher due to its higher accuracy score.

Options	Option 2	Option 3	Option 4
Accuracy	0.8659	0.8823	0.7137

TABLE I. NODE 5 CLASSIFIER ACCURACY SCORES AFTER IMPLEMENTING THE DIFFERENT LEARNING OPTIONS ON THE FMNIST DATASET

For the purposes of this experiment, all four options were implemented for this framework. Option 1 resulted in a complete dataset transfer, from node 1 to node 5. For the rest of the options the relevant procedures were followed and the results are shown in Table I. As can be seen, the network exploits the benefits of knowledge distillation that is able to guide the student during training and achieve higher accuracy score than the teacher. In this example, an MLP with limited layers was used, in order to illustrate the potential of our framework and of course knowledge distillation techniques. The question that arises is whether many small classifiers cooperating in our framework can surpass a deep neural network classifier, in terms of accuracy.

In order to show the framework's possibilities, we conducted the same experiment for the datasets CIFAR10 and SVHN. For this example all initiated nodes included a ResNet18 classifier. So the five nodes initiated here are:

1) Node with out of distribution detector (CIFAR10), classifier (CIFAR10) with 0.7865% accuracy.
2) Node with out of distribution detector (SVHN), classifier (SVHN) with 0.9595% accuracy.
3) Node with no out of distribution detector or classifier.
4) Node with out of distribution detector (CIFAR10), classifier (CIFAR10) with 0.7417% accuracy
5) Node with no out of distribution detector or classifier, that will be given new data (student)

As expected, node 1 was selected as teacher, resulting in the same conclusions as before when using the different learning options (Table II). If nodes 1 and 4 were never initialised, no nodes would have responded positively to node 5, thus it would return to Stable state.

It is noted here, that the functionality experiments are only aiming to prove the proper operations of the proposed framework. The potential experiments and improvements are unlimited, and the study on the

Options	Option 2	Option 3	Option 4
Accuracy	0.8206	0.8543	0.7865

TABLE II. NODE 5 CLASSIFIER ACCURACY SCORES AFTER IMPLEMENTING THE DIFFERENT LEARNING OPTIONS ON THE CIFAR10 DATASET

framework's properties is scheduled for future work. Although, the self-awareness feature this framework is introducing constitutes a novelty, which is inspired by the human nature. The awareness that a human does not poses a knowledge on a given subject, urges the search of a teacher or the knowledge itself.

CONCLUSION

In conclusion, this paper introduces a novel framework that leverages teacher-student interactions and knowledge distillation techniques to enhance communication and learning among neural networks. By establishing a network of nodes, the framework facilitates efficient knowledge transfer and collaboration. The proposed framework addresses the challenges of handling diverse data distributions and the limitations of individual networks by enabling the exchange of knowledge and experiences among nodes. It provides a systematic approach for students to learn from teachers, incorporating knowledge distillation to enhance learning outcomes. The framework's communication protocols ensure effective message exchange, promoting seamless interactions and optimizing knowledge utilization. The contributions of this paper advance the field of neural network communication and knowledge distillation, offering valuable insights and practical techniques for developing more intelligent and collaborative AI systems. The proposed framework opens up new avenues for research and applications, with the potential to revolutionize the way neural networks learn and interact in complex environments.

For the aforementioned reasons, we are eager to continue the experiments on the framework, but also trying to improve its functionality and offer various settings. The future work should be focused on using the work of paper [11] to integrate multiple teacher learning and the ability to use various architectures for teachers and students. Furthermore, it should be tested on different tasks, in order to discover the situations that a node forgets or decreases its performance on a task. Continual learning solutions could be applied

to solve this issue. The experiments should focus on the framework's ability to increase given knowledge, on the nodes' ability to remember all the tasks learned sequentially and the ability to retrieve the overall knowledge, when each node knows only part of it. To conclude the framework's impact is visible through the variety of research areas the future work will cover.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 951911 (AI4Media). This publication reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. "Distilling the Knowledge in a Neural Network". In: *arXiv preprint arXiv:1503.02531* vol. 2.no. 7 (2015).

[2] Davide Maltoni and Vincenzo Lomonaco. "Continuous learning in single-incremental-task scenarios". In: *Neural Networks* 116 (2019), pp. 56–73.

[3] C Bucilua, R Caruana, and A Niculescu-Mizil. "Model compression, in proceedings of the 12 th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining". In: *New York, NY, USA* 3 (2006).

[4] Andreas Bar, Fabian Huger, Peter Schlicht, and Tim Fingscheidt. "On the robustness of redundant teacher-student frameworks for semantic segmentation". In: (2019), pp. 0–0.

[5] Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifan Gong. "Learning small-size DNN with output-distribution-based criteria". In: *Fifteenth annual conference of the international speech communication association*. 2014.

[6] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4133–4141.

[7] Nikos Komodakis and Sergey Zagoruyko. “Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer”. In: *ICLR*. 2017.

[8] Guorui Zhou, Ying Fan, Runpeng Cui, Weijie Bian, Xiaoqiang Zhu, and Kun Gai. “Rocket launching: A universal and efficient framework for training well-performing light net”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[9] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. “Fitnets: Hints for thin deep nets”. In: *arXiv preprint arXiv:1412.6550* (2014).

[10] Fei Ye and Adrian G Bors. “Lifelong teacher-student network learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2021), pp. 6280–6296.

[11] Yuang Liu, Wei Zhang, and Jun Wang. “Adaptive multi-teacher multi-level knowledge distillation”. In: *Neurocomputing* 415 (2020), pp. 106–113.

[12] K. Lee, H. Lee, K. Lee, and J. Shin. “Training confidence-calibrated classifiers for detecting out-of-distribution samples”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2018.

[13] D. Hendrycks and K. Gimpel. “A baseline for detecting misclassified and out-of-distribution examples in neural networks”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017.

[14] T. DeVries and G. W. Taylor. “Learning confidence for out-of-distribution detection in neural networks”. In: *arXiv preprint arXiv:1802.04865* (2018).

[15] Vahdat Abdelzad, Krzysztof Czarnecki, Rick Salay, Taylor Denounden, Sachin Vernekar, and Buu Phan. “Detecting out-of-distribution inputs in deep neural networks using an early-layer output”. In: *arXiv preprint arXiv:1910.10307* (2019).

[16] Zhisheng Xiao, Qing Yan, and Yali Amit. “Likelihood regret: An out-of-distribution detection score for variational auto-encoder”. In: *Advances in neural information processing systems* 33 (2020), pp. 20685–20696.