Knowledge Distillation for Robotics Time Series Classification

Javidan Abdullayev[1], Maxime Devanne[1], Jonathan Weber[1], and Germain Forestier[1]

IRIMAS, Université de Haute-Alsace
`javidan.abdullayevauha.fr`

1 Abstract

Recently, deep learning models have shown great success in a variety of fields, especially computer vision, speech recognition and natural language processing. The success of deep learning models motivated researchers to apply them to time series analysis, especially in Time Series Classification (TSC). A trend which we witness in deep learning field, state-of-the-art deep learning models become more complex over time. It is often impractical to deploy very complex deep learning models to embedded systems (edge devices, mobile phones), robots or a production enviorenment due to resource constraints. In deep learning context, knowledge distillation is a model compression technique which is used to transfer knowledge from a heavy model (deep) to a lightweight model. As a result, the lightweight model will require less resources in terms of memory and computation but will deliver competitive performance compared to the heavy model. The purpose of this paper is to introduce and explore the concept of Knowledge Distillation (KD) for time series classification with specific focus on robotics time series classification using state-of-the-art Inception architecture. In light of the fact that deep learning models are employed in the classification of time series, we believe using knowledge distillation is a viable research direction for the future.

2 Introduction

In recent years deep learning revolutionized the field of machine learning. Deep learning models have shown excellent performance in a wide range of applications, particularly computer vision [1], document retrieval [2] and speech recognition [3]. The success of deep learning architectures motivated researchers to examine them for time series analysis, in particular for the task of Time Series Classification (TSC). However, as proposed deep architectures for TSC have become increasingly complex, it often becomes impossible or at least impractical to deploy such cumbersome deep models to robots with limited resources (memory constraints, computational power, etc.).

Hence, the development of shallower models maintaining good performances is required. In this context, one of the model compression techniques is called knowledge distillation which allows for knowledge to be transferred from a larger deep model or ensemble of models (the *teacher*) to a smaller model with fewer parameters (the *student*). As a consequence, the resulting smaller model will require less computing power and less memory consumption while performing competitively with the larger model. The purpose of this paper is to introduce and explore the concept of Knowledge Distillation (KD) with specific focus on robotics time series classification. We analyze the effects of KD with the state-of-the-art deep learning model for TSC, the Inception architecture, by assessing its impact on shallower models by reducing the number of layers.

This idea of Knowledge Distillation was first proposed by Bucilia et al. [4]. Later, Hinton et al. 2015 [5], further developed this concept and proposed to train a *student* model using the output of the softmax function so that softmax outputs of *student* and *teacher* networks are close to each other. This concept has been mainly evaluated in Computer Vision for image classification.

In the domain of time series analysis, Deep Learning has also been shown to be very effective for TSC. Several Convolutional Neural Network (CNN) approaches have been proposed and adapted for TSC, such as multi-scale CNN [6], Fully Convolutional Network (FCN) [7] or Residual Network (ResNet) [7]. Later, in a review paper by Ismail Fawaz et al.[8], these approaches were compared with other deep learning approaches. A more recent approach adapting the Inception architecture for TSC, namely Inception-Time [9], demonstrated that considering various sizes of convolutional filters results in better classification accuracies.

3 Proposed Approach

The core idea of Knowledge Distillation implies two neural networks, a *teacher* (generally a deep model) and a *student* (generally a shallow model), as illustrated in Figure 1. The goal is to train a *student* model on a time series dataset $\mathbf{D}$ by leveraging the knowledge acquired by a pre-trained *teacher* model. The knowledge generally refers to the last layer's output of the *teacher* model. To consider Knowledge Distillation, during training, the *student* model is optimized to mimic *teacher* final predictions through a distillation loss (the Kullback-Leibler divergence) measuring the similarity between both models probability distributions. The knowledge obtained from a teacher model during the training of the student model is recorded in the distillation loss. Moreover, the *student* model is also optimized to minimize the classification error through a student loss (cross-entropy). Thus, the final Knowledge Distillation loss $\mathcal{L}_{KD}$ is defined as:

$$\mathcal{L}_{KD} = \lambda \times \mathcal{L}_{CE}(\mathbf{Y}, \hat{\mathbf{Y}}_S) + (1-\lambda) \times \tau^2 \times \mathcal{L}_{KL}(\hat{\mathbf{Y}}_T^{\tau}, \hat{\mathbf{Y}}_S^{\tau}), \tag{1}$$

where λ controls the weight of both distillation loss $\mathcal{L}_{KL}$ and student loss $\mathcal{L}_{CE}$. The student loss $\mathcal{L}_{CE}$ corresponds to the classification loss defined as the cross-entropy between student predictions $\hat{\mathbf{Y}}_S$ and true labels $\mathbf{Y}$.

For the backbone architecture in our Knowledge Distillation framework, we choose the Inception architecture [9] as the pre-trained *teacher*. As shown in Figure 1, the *teacher* Inception network contains two residual blocks, each including three Inception modules.

For the *student* model, we follow the same Inception architecture but with fewer of Inception modules (layers). Hence, we build five different *student* models, denoted as Model_#M, by varying the number of Inception modules from 1 to 5. Detailed information about configuration of student models is given in table 1.

	Teacher	Students				
Model name	Model_6M	Model_5M	Model_4M	Model_3M	Model_2M	Model_1M
# Inception modules	6	5	4	3	2	1
Total # parameters	422 627	325 347	244 963	164 579	83 555	3 171

Table 1. Configurations of our *teacher* Inception and *student* Inception architectures

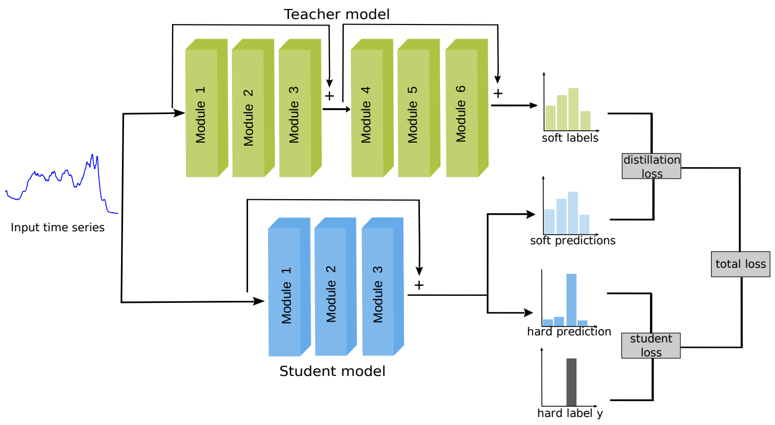

Fig. 1. Overview of our knowledge distillation architecture for time series classification

4 Experimental Evaluation

4.1 Experimental Setup

A large repository of time series datasets known as UCR Archive [10] is used to evaluate student models. The repository contains 128 datasets, however datasets containing unequal lengths or missing values are discarded. Consequently, we consider 112 univariate time series in our experiments.

The process of knowledge distillation start by training teacher model five times for each dataset. Among the five runs, the model with the lowest training loss is selected for knowledge distillation in order to achieve reproducible results. Then the selected teacher model is used to guide training process for the student model as described in Figure 1. In parallel, we also train the same student model, which we denote as studentAlone model, without taking into account knowledge distillation. In our experiments, aim is to assess impact of Knowledge Distillation on performance of shallower *student* models for the case of TSC. For that reason we compare the performances of *student* models benefiting from Knowledge Distillation against the *studentAlone* models trained only without the distillation loss. We train each student and studentAlone moels five times to reduce dependence on random initialization of Inception models. In order to assess each model's performance, we average the accuracy of five runs per dataset

4.2 Experimental Results

UCR Archive 2018 We first consider 112 time series datasets from the UCR Archive [10]. For each time series dataset, we compare the classification performances of both *student_#M* and *studentAlone_#M* models with various number of Inception modules. We then count the number of wins (*student* accuracy greater than *studentAlone* accuracy), losses (*student* accuracy lower than *studentAlone* accuracy) and ties (equal accuracies) for each *student_#M* and *studentAlone_#M* models. Comparative results are reported in Table 2. Furthermore, Figure 2 depicts number of wins for each model configuration. We can observe that Knowledge Distillation is particularly interesting in the case of intermediately complex models (*Model_4M*), where the *student_4M* model obtains higher accuracies than the *studentAlone_4M* on 57 time series datasets.

Model_1M			Model_2M			Model_3M			Model_4M			Model_5M		
Win	**Tie**	**Loss**	**Win**	**Tie**	**Loss**	**Win**	**Tie**	**Loss**	**Win**	**Tie**	**Loss**	**Win**	**Tie**	**Loss**
19	6	87	38	12	62	42	19	51	57	13	42	46	18	48

Table 2. Win/Tie/Loss comparison of *student_#M* and *studentAlone_#M* models with various numbers of layers. These results are based on *student* performance.

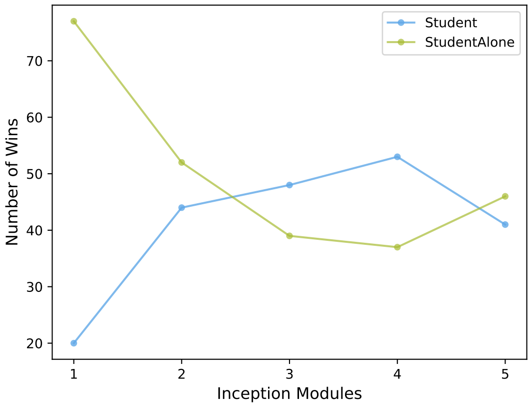

Fig. 2. Number of wins considering classification accuracy according to various numbers of layers of student models.

Based on all the results, we can conclude that in case of complex student model (student_5M) knowledge distillation does not help to improve the performance. This can be explained by the fact that as student models have similar complexity respect to a teacher model they will be able to capture discriminative patterns from the data without considering knowledge distillation thus demonstrate competitive performance with the teacher model. In case of intermediate complexity student model (student_4M), knowledge distillation helps to improve the performance significantly. The reason for this case is that since the student models have enough complexity, they can leverage and capture knowledge from a more complex teacher model. For relatively less complex student models knowledge distillation degrades performance of the student models (student_3M, student_2M, student_1M). This is due to the fact that less complex student models do not have enough capacity to leverage knowledge from a more complex student model.

Robotics Time Series Knowledge distillation is particularly interesting in the field of robotics in order to embed large deep neural networks in resource-constraint robotics systems. Hence, we focus especially on robotics time series by analyzing the results obtained for two related datasets, included in the UCR Archive, *SonyAIBORobotSurface1* and *SonyAIBORobotSurface2*. These datasets include use cases for surface detection using accelerometer data. Time series examples from SonyAIBORobotSurface1 dataset are illustrated in Figure 3. The robot has 3 accelerometers: roll, pitch and yaw. The examples from the Figure 3, represent accelerometer values that recorded in X axis. The task is to detect whether the surface being walked on is carpet or cement. The SonyAIBORobot-Surface2 dataset contains same kind of examples as SonyAIBORobotSurface1 dataset

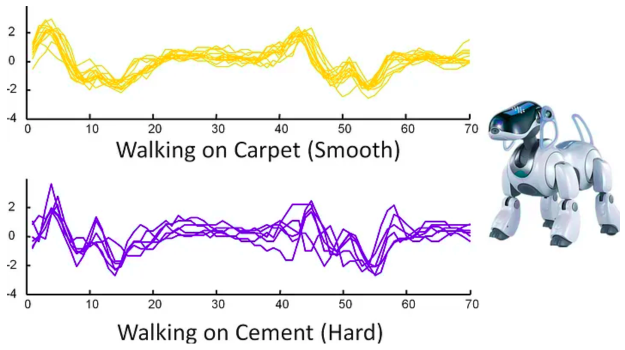

Fig. 3. Example time series from SonyAIBORobotSurface1 dataset.

but the task in that case is to detect whether the surface being walked on is field or cement.

Figure 4 and 5 demonstrate performance of all designed *student_#M*, *studentAlone_#M* and *teacher* models with different number of Inception modules for SonyAIBORobotSurface1 and SonyAIBORobotSurface2 datasets respectively. From the figure 4, we can notice that all distilled student models except student_5M model demonstrate better performance than the studentAlone models. The results that we discussed in previous section is valid for the intermediate and complex student networks but not for smaller student architectures. In case of sonyAIBORobotSurface1 dataset we can observe that smaller student models perform better than studentAlone models. The distilled student model even with only one inception module, student_1M, demonstrate better performance than the studentAlone_1M model. The accuracy of student_1M, studentAlone_1M and teacher models are 87.4%, 85.4% and 86.9% From the results, it is obvious that knowledge distillation improves the performance of the studentAlone_1M model by 2%. To find an explanation for this behaviour we visited the dataset source [1], as written there the dataset has only 20 training examples. It means that the dataset is too small thus even smaller student models can capture discriminative features from the data as well as leverage knowledge from the teacher model.

We also analyze performance of knowledge distillation on another robotics dataset SonyAIBORobotSurface2. From the figure 5, we can observe similar patterns that we observed for SonyAIBORobotSurface1 dataset. Smaller and intermediate distilled student models demonstrate slightly better performance over the corresponding studentAlone models. Based on the source[2], the dataset contains only 27 training examples which shows the reason why smaller distilled student models performs better than studentAlone models. For the complex distilled student_5M model knowledge distillation does not improve the performance. From the Figure 5 we can notice an unexpected pattern in which performance of studentAlone models do not improve as we increase number of inception modules after studentAlone_3M model. We believe that this pattern is due to overfitting issue with this dataset as it contains only 27 training samples.

[1] https://www.timeseriesclassification.com/description.php?Dataset=SonyAIBORobotSurface1
[2] https://www.timeseriesclassification.com/description.php?Dataset=SonyAIBORobotSurface2

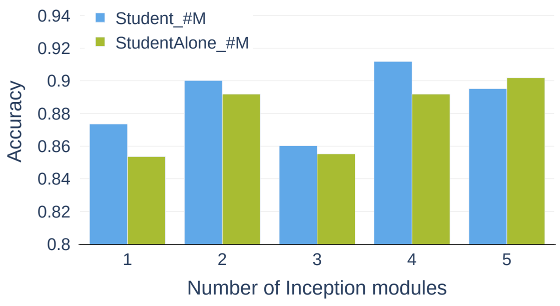

Fig. 4. Performance of *student_#M*, *sudentAlone_#M* and *teacher* models with different number of Inception modules on SonyAIBORobotSurface1 dataset.

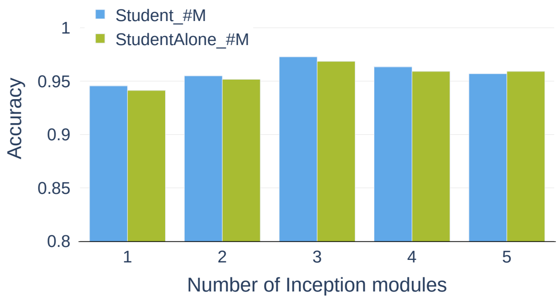

Fig. 5. Performance of *student_#M*, *sudentAlone_#M* and *teacher* models with different number of Inception modules on SonyAIBORobotSurface2 dataset.

5 Conclusion

In this paper, we analyzed the success of knowledge distillation on Inception network in the case of TSC, particularly for two robotics datasets. We assessed the impact of knowledge distillation by reducing the complexity of student models while leveraging teacher knowledge. In particular, we evaluated the impact of knowledge distillation by reducing the number of inception modules in student architectures. The experimental results conducted on the UCR Archive 2018 datasets which is the largest repository of time series datasets available and it suggests that knowledge distillation boost the performance when applied on intermediate complexity student models. This results is inline with the one that observed in [11] for image classification problem. The *student_4M* Inception model has 2 more wins over the *teacher* model. Considering that the number of parameters for the *student_4M* model is 42% less than the *teacher* Inception model then the *student_4M* model can be regarded as a very promising architecture to replace the *teacher* model.

The results that we observed on robotics dataset was a bit different. For robotics datasets knowledge distillation improved performance for even smaller student models.

We strongly believe that this is due to less number of training examples in those datasets. According to the experiments on robotics datasets, even student_1M model demonstrate slightly better performance than the more complex teacher model. It is worth to note that the number of student_1M model parameters is equal to about 1% of the teacher model parameters. Based on these results, we can say that knowledge distillation is effective also for robotic datasets since it allows significant reduction in model size while maintaining good performance. Contrary to image classification problems in which knowledge distillation evaluated on few datasets, in our study we considered 112 time series datasets, demonstrating that finding an appropriate architecture for all type of datasets is not a straightforward task.

Acknowledgment

The authors would like to thank the creators and providers of the UCR Archive.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
2. Li, H., Lu, Z.: Deep learning for information retrieval. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. (2016) 1203–1206
3. Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H.G., Ogata, T.: Audio-visual speech recognition using deep learning. Applied intelligence **42** (2015) 722–737
4. Ba, J., Caruana, R.: Do deep nets really need to be deep? Advances in neural information processing systems **27** (2014)
5. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
6. Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv:1603.06995 (2016)
7. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International joint conference on neural networks (IJCNN), IEEE (2017) 1578–1585
8. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data mining and knowledge discovery **33**(4) (2019) 917–963
9. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**(6) (2020) 1936–1962
10. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6) (2019) 1293–1305
11. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: Proceedings of the AAAI conference on artificial intelligence. Volume 34. (2020) 5191–5198