BUILDING LIGHT MODELS WITH COMPETITIVE PERFORMANCE FOR REMOTE SENSING

Olga Garces Ciemerozum
Satellogic & Universitat Oberta de Catalunya
olga.garces@satellogic.com

Javier Marín
Satellogic & Universitat Oberta de Catalunya
javier.marin@satellogic.com

ABSTRACT

The communication between ground stations and low earth orbit satellites is limited by a window of time as well as by the signal transmission speed. As a consequence, machine learning models for remote sensing need to be reasonably small in order to be transmitted and loaded to the device. Top performing deep learning models in the literature usually include millions of parameters, which limits their potential use on board once the satellite is in orbit. This paper is inspired by a previous work, PRANC, which explores the feasiblity of using a linear combination of multiple pseudo-randomly generated frozen models for classification purposes. We extend its use to semantic segmentation of building footprints. While this is not a reduction technique as such, results demonstrate that these type of models can be easily transmitted and reconstructed on board without compromising the model performance. In particular, the network reaches a competitive performance, while requiring only hundreds of kilobytes.

1 INTRODUCTION

The execution of machine learning models, such as deep neural networks, under storage, computational and connectivity constrains has led both the industry and the research community to explore and propose different techniques to overcome these challenges. In particular, a lot of interest has grown in the fields of edge computing, split computing, model efficiency, model compression and their combinations. Among the aforementioned techniques, we divide them into four different groups: 1) model compression, 2) designing lightweight efficient models, 3) update compression, and 4) random initialization.

Pruning is a compression method based on the assumption that not all the weights and connections of the neural network are equally important and relevant, so some of them could be removed without causing a significant degradation of the model's performance. Cai et al. (2022) survey comprises different pruning techniques according to their granularity. Quantization is a compression method that can be separated into weight sharing and representation of weights in fewer bits. Some relevant researches in this area are Mary Shanthi Rani et al. (2022); Martinez et al. (2021); Girish et al.; Lin et al. (2022); Xu et al.. Knowledge distillation is technique aims at training a small model using the knowledge acquired by a bigger model or an ensemble Hinton et al. (2015); Touvron et al.; Lin et al.; Chung et al. (2020); Crowley et al. (2021); Tan & Liu (2022); Ganesh et al. (2021). Other compression techniques can be found in Nie et al. (2022); Gabbay & Shomron (2021).

Another research line focuses on designing lightweight efficient models. Cai et al. (2022) names some of the most common strategies to design lighter models, these are using efficient convolution layers such as 1x1 convolutions, group convolutions or depthwise separable convolutions. Some of the models that can be considered edge friendly due to their reduced size are SqueezeNet, MobileNet or ShuffleNet.

The update compression is useful in the case when there is an unreliable connection between the server and the edge device while the training data is available on the server. Konečný et al. (2017) propose two communication efficient methods for reducing the uplink costs: structured and sketched updates. More recently, Chen et al. (2022) propose postdeployment updates, where only the updated parts of the model are compressed via matrix factorization and sent to the edge device, with the inconvenience that this method can be applied only to CNN or DNN.

Split computing is a framework that divides the DNN model into head, executed on the device and tail, executed on the server. The result is then sent back to the device. An exhaustive summary of the state of the art can be found in Matsubara et al. (2022).

Recently there are several researches that are related to randomly initializing models. In the paper Parameter-Efficient Masking Networks (Bai et al., 2022) the authors propose a new paradigm of model compression: expressing a model as one randomly initialized layer and a number of masks that will adjust the weights as needed. Nooralinejad et al. (2022) also propose random initialization of weights, in this case, all the weights in a model are initialized randomly and multiple random models are combined to give one final prediction. For this, each model receives a coefficient that determines its weight in the result.

One interesting use case consists in transmitting large models to low earth orbit (LEO) satellites. Due to communication constraints, it can be prohibitive to upload accurate deep learning models. Hence, it is of paramount importance to find alternatives to upload new models without compromising the performance. In this project, due to its simplicity and remarkable size in terms of parameters, we extent Nooralinejad et al. (2022) work, Pseudo RAndom Networks for Compacting deep models (PRANC), and adapt it for land use classification.

The rest of the paper is organized as follows. In Section 2, we explain in detail Nooralinejad et al. (2022) work. In Section 3, we describe our experimental setup. In Section 4, we present results from our experiments and conclude the paper in Section 5.

2 METHOD

This research builds upon Nooralinejad et al. (2022) work, PRANC, which uses a linear combination of a set of randomly initialized models (basis models), reducing the problem to finding the linear mixture coefficients that will provide the best classification result. The method looks for models that are functionally similar to pretrained models, rather than close in the parameter space (see Fig. 1).

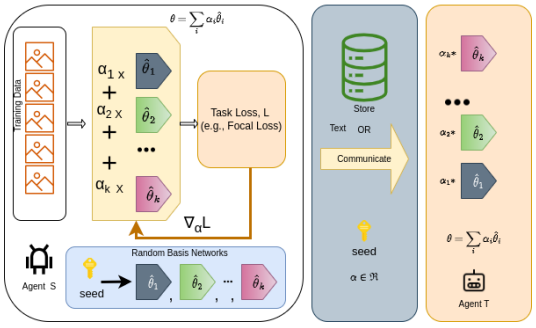

Figure 1: PRANC is a linear combination of α randomly generated models

Given the task of training a model $f(.;\theta)$ with the parameters $\theta \in \mathbb{R}^d$ so that $f(x_i;\theta)$ predicts y_i, the authors propose initializing the basis models with parameters $\{\hat{\theta}_j\}_{j=1}^{k}$ using a random seed. Then, the weights of the final model is defined as:

$$\theta := \sum_{j=1}^{k} \alpha_j \hat{\theta}_j$$

where α_j is a scalar weight of j's basis model. The authors argue that there are infinite solutions for θ, so the search is reduced to finding the one with the smallest residual error.

$$argmin_{\alpha} \sum_{i} L(f(x_i; \sum_{j=1}^{k} \alpha_j \hat{\theta}_j), y_i)$$

The model can be trained, reconstructed and stored efficiently in terms of computation and memory usage, with the limitation of having to transmit batch normalization layers as they are. More infor-

mation about the PRANC method, optimization, model storage, reconstruction and experiments is available in the original paper.

The authors of this work perform experiments for image classification on several datasets such as CIFAR10, CIFAR100, Imagenet and show that the PRANC method outperforms SOTA model compression methods such as knowledge distillation or pruning while using less parameters. Another important conclusion is that a larger number of base models leads to higher accuracy.

3 EXPERIMENTAL SETUP

To run our experiments we use the original repository created by the authors. We adapt the code accordingly to apply the PRANC method on semantic segmentation tasks. The dataset used in this work is the ISPRS Potsdam dataset[1], which contains six different labels: impervious surfaces, building, low vegetation, tree, car and clutter/background. We use for the semantic segmentation task DeeplabV3+ (Chen et al., 2018).

We first train the baseline model to ensure that we obtain similar results (Wang et al., 2022) on the Potsdam dataset. Then, the PRANC method is applied to train α models initializing their weights randomly and optimizing alpha coefficients. Since the model is meant to be run onboard, we reduce the number of randomly initialized models. More concretely, we train with 5, 10, 30, 100 and 240[2] models. The intention of the experiments is finding how much the performance changes with respect the number of participating models and comparing it with the baseline DeepLabv3+.

For training and validating the DeepLabv3+ models, we use six GPUs with 24GiB each. For the models composed of 5, 10, 30 and 100 alphas, we use a training batch size of 28 and a validation batch size of 10, while with the 240 alphas model the training batch size needs to be reduced to 20 in order to be able to load the data into the memory. Each type of model is trained five times for 100 epochs. All the models use an SGD optimizer with a 0.09 learning rate and a 0.9 momentum. The loss function used for training all the models is a Focal loss. The PRANC method uses a step scheduler, with a step of 50. For the baseline models, the learning rate scheduler is a cosine annealing scheduler with warm restarts. [3].

To measure each experiment performance overall classes, we use F1-score. The value obtained is the result of averaging the five runs. We also report the IoU, sensitivity and specificity for each case. The best performance is extracted using the validation partition.

4 RESULTS

Table 1 shows the results of each of the experiments we conduct in this paper. The models composed of fewer participants seem to perform nearly as good as the one with 240 baseline models. In particular, the difference in performance is below one percentage point. This is an interesting finding, since using a higher number of base models, as the authors suggest, may be prohibitive onboard due to energy consumption and time constraints.

Regarding the model size required to transmit the PRANC model and the baseline, while in the case of sending the full DeepLabV3+ model, we need to send 103 Mb, 26,678,870 parameters, it turns out that for the PRANC model we only need to send 456 Kb, this is 55,517 parameters. Those parameters include the random seed, the learned alpha coefficients, as well as the parameters representing the mean and the standard deviation for layers other than convolutional or fully connected.

Figure 2 shows a qualitative comparison between the baseline, DeepLabV3+, and the PRANC model with 240 (240A) random models. As can be seen from the results, the classes representing low vegetation and tree are the ones with the lowest performance, as it happens in other implementations to predict the Potsdam dataset Wang et al. (2022). Small differences seem to come from better defining the contours of some classes, or missing trees without leaves.

[1] https://www.isprs.org/education/benchmarks/UrbanSemLab/

[2] We did stop at 240, divisible by the maximum number of gpus we had, since we did not want to significantly increase the computational cost onboard.

[3] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingWarmRestarts.html

Table 1: Qualitative results averaging five runs per experiment.

Model	F1-score	IoU	Sensitivity	Specificity
5Alpha	0.8494	0.7484	0.8421	0.9617
10Alpha	0.8435	0.7416	0.8409	0.9599
30Alpha	0.8429	0.7398	0.842	0.9601
100Alpha	0.8485	0.7478	0.8416	0.9617
240Alpha	**0.8572**	**0.7604**	**0.8548**	**0.9633**
Baseline	0.8868	0.8049	0.884	0.9697

In general, none of the models, including the baseline is able to capture the red areas as background. Those instances are usually predicted as any other class, depending on the context: clutter around building gets labeled as building, while clutter around low vegetation gets low vegetation labels.

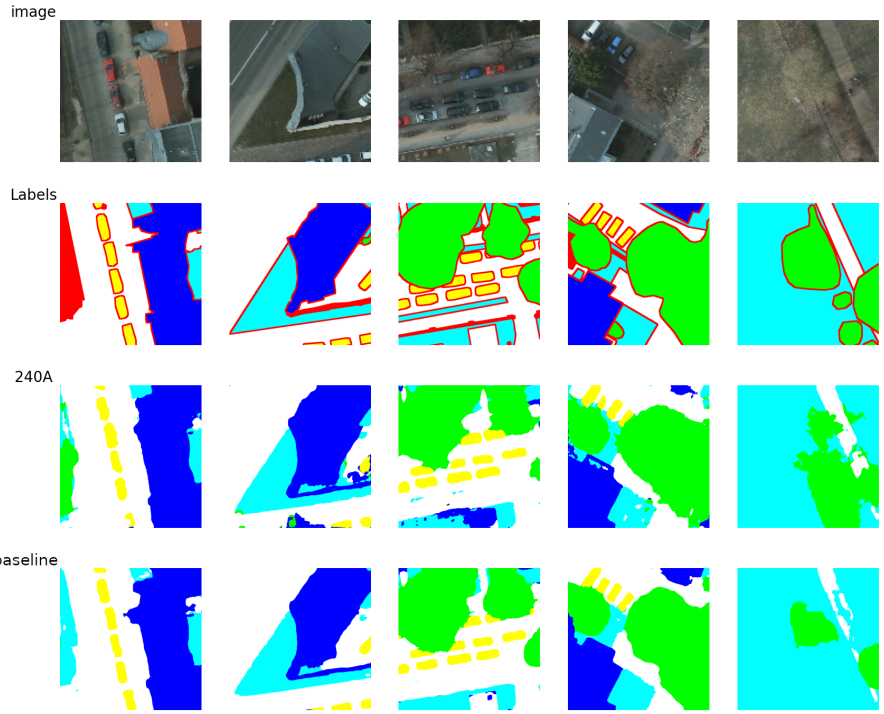

Figure 2: Comparison between the baseline and PRANC using 240 models.

5 Conclusions

This paper, inspired and based on PRANC, explores the feasibility of using this technique to create accurate models that can be easily transmitted to remote sensing satellites. In particular, we conduct multiple experiments with different numbers of base models and compare their performance to a baseline DeepLabv3+ model. In comparison to the original work, that used large numbers of baseline models (under 20,000), we use a relatively small number (under 240) and with minimal parameter tuning achieve a mean F1-score less than 3% short from the baseline's performance, and less than 4% when using a combination of five models. These results show that using randomly initialized models onboard permit transmitting effectively accurate models that only require 456 Kb. As an ongoing work, it would be interesting to explore how to further improve the model. For instance, instead of training the PRANC model from scratch, we could train it via distillation using the baseline or smaller models (to reduce the computational cost onboard) as the teacher. While this research focuses on transmitting models to orbiting satellites, these finding could be also used reduce the storage required on ground.

REFERENCES

Yue Bai, Huan Wang, Xu Ma, Yitian Zhang, Zhiqiang Tao, and Yun Fu. Parameter-Efficient Masking Networks, October 2022. URL http://arxiv.org/abs/2210.06699. arXiv:2210.06699 [cs].

Han Cai, Ji Lin, Yujun Lin, Zhijian Liu, Haotian Tang, Hanrui Wang, Ligeng Zhu, and Song Han. Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications. *ACM Transactions on Design Automation of Electronic Systems*, 27(3):1–50, May 2022. ISSN 1084-4309, 1557-7309. doi: 10.1145/3486618. URL http://arxiv.org/abs/2204.11786. arXiv:2204.11786 [cs].

Bo Chen, Ali Bakhshi, Gustavo Batista, Brian Ng, and Tat-Jun Chin. Update Compression for Deep Neural Networks on the Edge. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3075–3085, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-66548-739-9. doi: 10.1109/CVPRW56347.2022.00347. URL https://ieeexplore.ieee.org/document/9857002/.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, August 2018. URL http://arxiv.org/abs/1802.02611. arXiv:1802.02611 [cs].

Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level Online Adversarial Knowledge Distillation, June 2020. URL http://arxiv.org/abs/2002.01775. arXiv:2002.01775 [cs, stat].

Elliot J. Crowley, Gavin Gray, Jack Turner, and Amos Storkey. Substituting Convolutions for Neural Network Compression. *IEEE Access*, 9:83199–83213, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3086321. URL https://ieeexplore.ieee.org/document/9446890/.

Freddy Gabbay and Gil Shomron. Compression of Neural Networks for Specialized Tasks via Value Locality. *Mathematics*, 9(20):2612, October 2021. ISSN 2227-7390. doi: 10.3390/math9202612. URL https://www.mdpi.com/2227-7390/9/20/2612.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. Compressing Large-Scale Transformer-Based Models: A Case Study on BERT. *Transactions of the Association for Computational Linguistics*, 9:1061–1080, September 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00413. URL https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00413/107387/Compressing-Large-Scale-Transformer-Based-Models-A.

Sharath Girish, Saurabh Singh, Kamal Gupta, and Abhinav Shrivastava. LilNetX: Lightweight Networks with EXtreme Model Compression and Structured Sparsification. pp. 19.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv:1503.02531 [cs, stat].

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency, October 2017. URL http://arxiv.org/abs/1610.05492. arXiv:1610.05492 [cs].

Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge Distillation via the Target-aware Transformer. pp. 13.

Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. FQ-ViT: Post-Training Quantization for Fully Quantized Vision Transformer. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 1173–1179, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3. doi: 10.24963/ijcai.2022/164. URL https://www.ijcai.org/proceedings/2022/164.

Julieta Martinez, Jashan Shewakramani, Ting Wei Liu, Ioan Andrei Barsan, Wenyuan Zeng, and Raquel Urtasun. Permute, Quantize, and Fine-tune: Efficient Compression of Neural Networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15694–15703, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.01544. URL https://ieeexplore.ieee.org/document/9577701/.

M. Mary Shanthi Rani, P. Chitra, S. Lakshmanan, M. Kalpana Devi, R. Sangeetha, and S. Nithya. DeepCompNet: A Novel Neural Net Model Compression Architecture. *Computational Intelligence and Neuroscience*, 2022:1–13, February 2022. ISSN 1687-5273, 1687-5265. doi: 10.1155/2022/2213273. URL https://www.hindawi.com/journals/cin/2022/2213273/.

Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. Split Computing and Early Exiting for Deep Learning Applications: Survey and Research Challenges, March 2022. URL http://arxiv.org/abs/2103.04505. arXiv:2103.04505 [cs, eess].

Chang Nie, Huan Wang, and Lu Zhao. STN: Scalable Tensorizing Networks via Structure-Aware Training and Adaptive Compression, May 2022. URL http://arxiv.org/abs/2205.15198. arXiv:2205.15198 [cs].

Parsa Nooralinejad, Ali Abbasi, Soheil Kolouri, and Hamed Pirsiavash. PRANC: Pseudo RAndom Networks for Compacting deep models, June 2022. URL http://arxiv.org/abs/2206.08464. arXiv:2206.08464 [cs].

Chao Tan and Jie Liu. Improving Knowledge Distillation With a Customized Teacher. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2022. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2022.3189680. URL https://ieeexplore.ieee.org/document/9839519/.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. pp. 11.

Di Wang, Jing Zhang, Bo Du, Gui-Song Xia, and Dacheng Tao. An Empirical Study of Remote Sensing Pretraining, May 2022. URL http://arxiv.org/abs/2204.02825. arXiv:2204.02825 [cs].

Kunran Xu, Yishi Li, Huawei Zhang, Rui Lai, and Lin Gu. EtinyNet: Extremely Tiny Network for TinyML. pp. 9.