POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

A Journey to Improve Neural Architecture Search: Advancements in Neural Architecture Transfer and Once-For-All

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Simone Sarti

Advisor: Prof. Matteo Matteucci

Co-advisor: Eugenio Lomurno

Academic year: 2021-2022

1. Introduction

Neural Architecture Search (NAS) is a family of techniques for automatically designing the optimal neural network architecture for a given task. Classical NAS techniques require large amounts of computational resources and time to run, and they often produce extremely task-specific networks. To overcome these drawbacks, two NAS techniques have recently been developed: *Once-For-All* (OFA) [1] and *Neural Architecture Transfer* (NAT) [2].

NAT transforms a pretrained supernet into a task-specific supernet, from which subnets that achieve optimal tradeoffs between a set of objectives can be directly extracted without the need for retraining. A phase of transfer learning to optimise the supernet is alternated with a phase of predictor guided many-objective evolutionary search to find the subnets that are currently on the tradeoff front. Only those parts of the supernet whose structure can be sampled from the distribution of subnets on the tradeoff front are fine-tuned. The best architectures found are added to an archive and returned at the end of the process, along with the adapted supernet. NAT is partly based on OFA.

OFA is used to design high performance neural networks that are adaptable to different hardware configurations, while minimising power consumption. To achieve this, a single large model with a dynamic architecture, called *supernet*, is trained through the smaller architectures it contains, called *subnets*. The training algorithm is called Progressive Shrinking (PS) and consists of many phases to progressively fine-tune smaller and smaller subnets from within the supernet. PS uses elastic parameters to activate subnets, which are progressively unlocked during the execution of four elastic steps: Elastic Resolution, Elastic Kernel Size, Elastic Depth and Elastic Width. Elastic steps can consist of several training phases. The network obtained after the first step, which is the maximal network, is used as teacher network to perform knowledge distillation on the subnets sampled in the following phases. Once the supernet has been trained, the search step is responsible for finding the subnet that best adapts to a specific platform and resource constraints. Training and search are decoupled, thus the search process can be run many times without the need to repeat the training.

This thesis examines how the presence of new architectural components and other algorithmic

improvements affect NAT. The first step along this path is the study of neural networks with early exits. To maximise the performance of such networks, a technique called *Anticipate, Ensemble and Prune* (AEP) is proposed. The AEP training methodology is then incorporated into an extended version of OFA, called OFAv2, which also supports architectures with parallel blocks and dense skip connections on top of early exits. Finally, NATv2 is built on top of OFAv2 to allow its search and adaptation processes to work on all the new supernets.

2. AEP

AEP is a new technique for maximising the performance of networks using early exits. It relies on the weighted ensemble of the exits to jointly train the network and compute its output.

2.1. Method

First, the exit section of the basic single-exit network is identified and replicated. The copies, with the number of features proportionally rescaled, are attached at intermediate points within the network, one after each main network stage. Two weights are then assigned to each exit, one used for calculating the network loss and the other used for calculating the network output. The network loss is computed as the weighted sum of the categorical cross-entropy losses L_i of the exits. The accuracy of the network is computed as the Argmax of the weighted sum of the raw outputs O_i of the exits. The network weights are learned via the joint training of the weighted ensemble of all the exits. After training the full-ensemble network, a pruning step is performed. For each possible activation state of the exits, the corresponding subnetwork is evaluated on the validation set. The best one is then extracted and evaluated on the test set. The outline of the AEP technique is shown in Figure 1.

2.2. Experiments

Experiments on the AEP method were performed on a large variety of architectures (ResNet50, VGG16, DenseNet169, MobileNetV3Small, EfficientNetB5) and datasets (Tiny ImageNet, CIFAR-10, CIFAR-100, EuroSAT, FashionMNIST, GTSRB) to ensure generality. The networks were trained from scratch as well as fine-tuned, using both large (224x224) and small (64x64) images. The patterns followed to assign weights to the exits were: DESC (decreasing for both losses and outputs), ASC (increasing for both losses and outputs), MIX (decreasing for losses, increasing for outputs) and UNIF (uniform for both losses and outputs). Weights were chosen to be always positive and sum to one. Pruned networks are indicated by the symbol '*'.

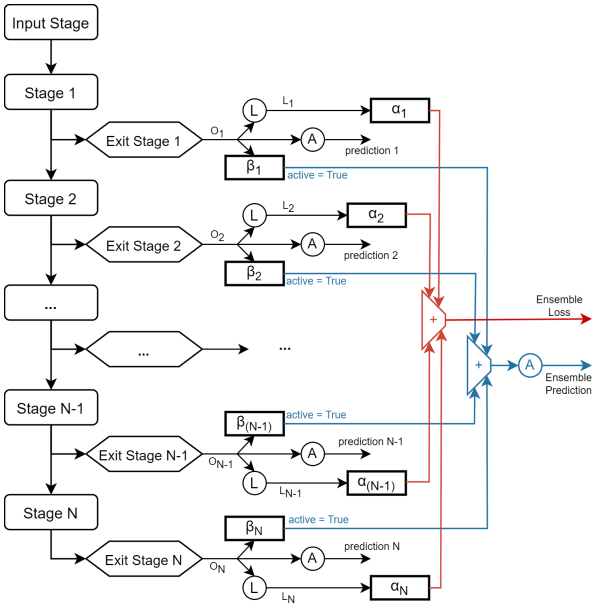

Figure 1: Outline of the AEP technique. Ⓛ represents the loss function, while Ⓐ represents the Argmax function. α_i and β_i represent the weights assigned to the loss and the output of exit stage i, respectively.

2.3. Results

Figure 2 summarises the results of the AEP experiments. The results were averaged over both networks and datasets, and are expressed as a percentage change over those of the baseline single-exit networks. In most cases, the networks obtained using AEP significantly outperform the single-exit networks in terms of average accuracy and, thanks to pruning, also in terms of network complexity, expressed in terms of number of parameters, operations (MACs) and latency. The gains are particularly relevant when training from scratch and when using smaller images. In the case of training from scratch, the best weight assignments are those with descending weights for loss (DESC, MIX), while the ASC strategy is the best for fine-tuning. In

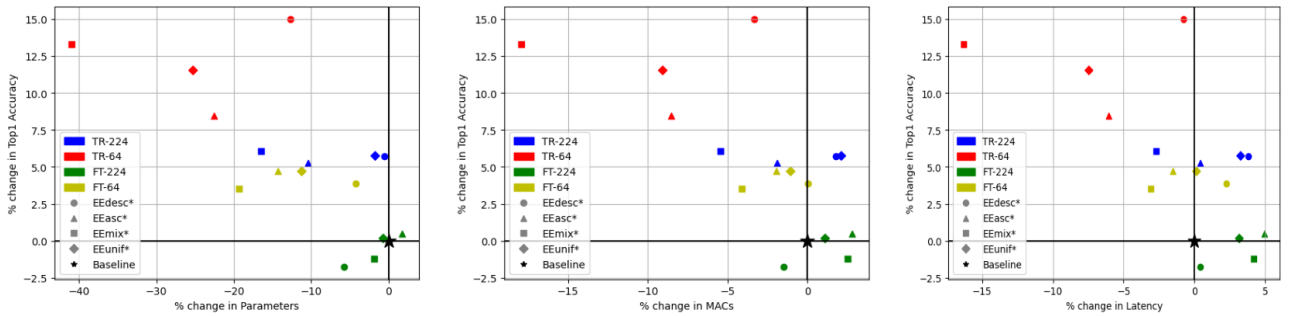

Figure 2: Results obtained using the complete AEP technique compared to those obtained using single-exit networks (baseline). The average variation in accuracy, on the y-axis, is represented as a function of the average variation in another network metric, on the x-axis. Each data point identifies a learning scenario/weighting strategy pair, the former represented by the colour of the data point, the latter by its shape. Better results will be closer to the top left-hand corner of the plot.

addition, AEP training often produces intermediate exits with higher prediction accuracy than the final classifier of the baseline network.

3. OFAv2

OFAv2 extends OFA to support supernets with new architectural components and designs, such as early exits, dense skip connections, and parallel blocks.

3.1. Method

OFAv2 extends the PS algorithm with two new steps: Elastic Level and Elastic Exit. They allow Extended Progressive Shrinking (EPS) to support progressive training of supernets containing parallel blocks and early exits, respectively. Elastic Level allows subnets to be sampled and trained with a progressively smaller number of active blocks at the same level in the network. Elastic Exit allows to sample and train subnets terminating at exits that are progressively closer to the input. As in OFA, the sampled subnets are single-exit networks. The additional skip connections are automatically activated and deactivated as a consequence of Elastic Depth. The EPS pipeline is schematised in Figure 3. To train the maximal early-exit networks, OFAv2 uses a technique based on the AEP training. EPS also takes advantage of AEP to create a new knowledge distillation technique in which the prediction of the teacher network, from which the subnets learn, is the result of the ensemble of the exits of the maximal early-exit network (ENS-KD). Finally, EPS proposes to progressively extract the maximal network to be used as a teacher from the supernet obtained after each training phase, instead of continuing to use the one obtained after the first step in the training pipeline. The following naming convention is used for the supernets: The first prefix can be either SE or EE, representing single-exit and early-exit networks respectively, the second prefix refers to inter-stage modifications, here D stands for dense skip connections, P for parallel blocks, and B (base) for their absence.

3.2. Experiments

Experiments on OFAv2 were conducted on all eight supernets generated by the combinations of the three architectural additions, for both width multipliers 1.0 and 1.2. Networks were also tested with and without the progressively extracted teacher network. All experiments were performed on the Tiny ImageNet dataset. After each training phase, the subnets corresponding to the possible combinations of elastic parameters, each being held constant throughout the network, were tested.

3.3. Results

Figure 4 shows the results of the OFAv2 experiments when a width multiplier of 1.0 was used. For each supernet, the results are shown for both cases where the teacher network was fixed and where it was progressively extracted (“pet”). For each elastic step and supernet, the accuracy of the best subnet tested is shown. For non-executed optional elastic steps, the results of the last executed step is shown. In the first step, with only Elastic Resolution active, single-exit parallel networks perform best, while early-exit

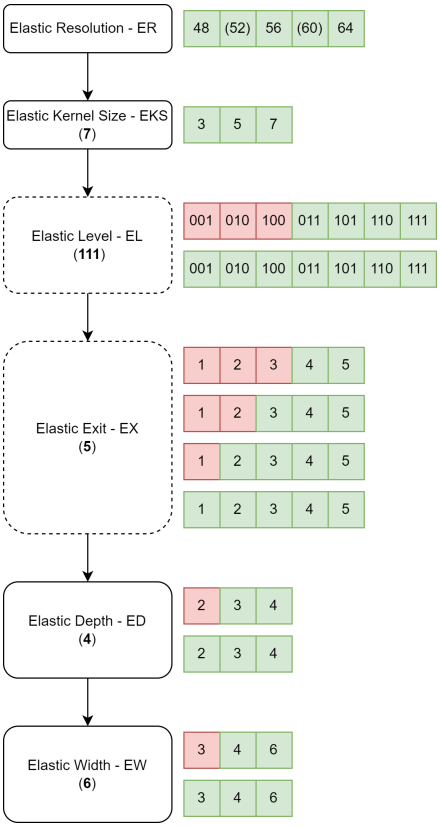

Figure 3: The Extended Progressive Shrinking algorithm. The available options for a given phase are shown in green, while the unavailable ones are reported in red. When a step has not been reached yet, the value for the corresponding modifier is set to the maximum one, reported under the name of the step. Dashed steps are performed only on supernets supporting the architectural modifications touched by that step.

networks lag behind, probably due to the difficulty of optimising for multiple exits and image sizes simultaneously. However, from the Elastic Kernel Size step on, early-exit networks close the gap and surpass the performance of single-exit networks, providing a significant improvement in accuracy at the end of training. The presence of dense skip connections consistently results in a modest advantage over their absence, and these improvements stack with those provided by early exits. For what concerns the teacher network, the progressive extraction strategy is always beneficial. These same observations are also valid in the case when a width multiplier of 1.2 is used. In conclusion, the best OFAv2 supernet is the one supporting both early exits and skip connections, trained with ENS-KD and the progressively extracted teacher network.

4. NATv2

NATv2 is an extension of NAT that builds on and complements OFAv2. It can be used in place of OFAv2's search step, allowing to search for optimal subnets also on datasets other than the one used to train the supernet, for all OFAv2 supernets.

4.1. Method

First, new encodings are defined for OFAv2 networks to provide information about which exit to take or which parallel blocks to activate, so that the evolutionary search can be run on the new supernets. The activation state of dense skip connections can be inferred. The way the archive is managed has been reworked, it is now initialised by sampling subnets from the architecture space rather than the search space, allowing a fairer distribution of network architectures also in terms of stage sizes. Complementary, a pre-processing step has been introduced: instead of directly sampling the architectures that will make up the initial archive, a large number of architectures are extracted, evaluated and compared, and only the best ones are kept to be part of the initial archive. What's more, the archive no longer grows by adding the architectures found by the search process; instead, these architectures replace worse ones within an archive of larger but constant size. Two alternative two-phase post-processing steps are also added to maximise the accuracy of the subnets returned by NATv2. The first fine-tuning phase uses only the training set to find the optimal number of epochs for which to train the subnet, this knowledge is used in the second phase where the subnet is fine-tuned on both the training and validation sets together, before being tested. The first post-processing method fine-tunes a subnet as it is returned by NATv2, i.e. with a single exit The second method is based on AEP and can be used only on subnets derived from supernets with early exits. In the latter case, the exits above the selected one are retrieved from the supernet and reconnected, then fine-tuning is performed jointly. Finally, a number of algorithms have been reimplemented to make better use of available computing resources, speeding up the execution of NATv2.

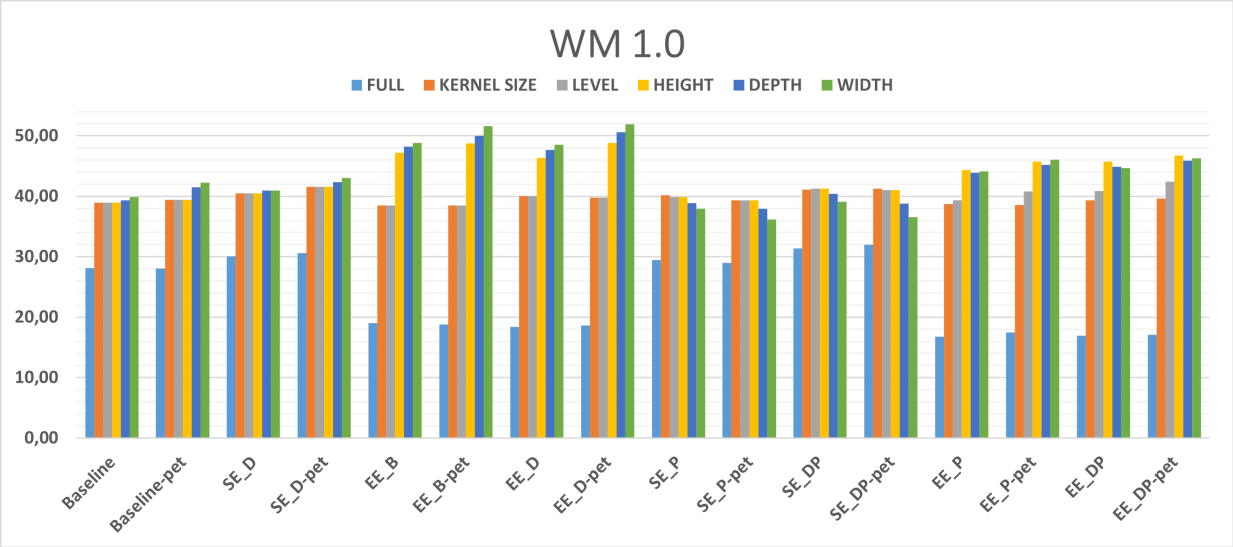

Figure 4: Top1 Accuracy achieved by the best subnet of each supernet after each EPS step. Results shown are for supernets trained with the width multiplier set to 1.0. OFAMobileNetV3 is the baseline and is omitted in the following supernet names, and "pet" means that the teacher network has been progressively extracted. For non-executed optional elastic steps, the value of the last executed step is shown.

4.2. Experiments

The performance of many error predictors was investigated for different numbers of input architectures and feature encodings (one-hot or integer), recording RMSE, time and correlation with Pearson's, Spearman's and Kendall's coefficients. Moreover, a study was also conducted to identify the best optimisation strategy to use in the post-processing step. Four NATv2 experimental phases were carried out: In the first, using the new search space encodings, NATv2 was tested on the OFAv2 supernets for a wide range of datasets. The LGBM error predictor was used. From the second phase, integer encodings of the architectures were fed to the error predictor, the new archive management strategy was used and the time-saving measures were applied. In the third phase, the pre-processing step was introduced. In the fourth and final phase, the post-processing steps were added, and OFAv2 supernets trained with the progressively extracted teacher network (which was a later addition) were used as a starting point. Due to the long running times, after the first phase, experiments were only carried out on CIFAR-10, CIFAR-100 and Tiny ImageNet.

4.3. Results

In terms of error predictors, the best correlations between true errors and predicted errors were achieved by CatBoost, closely followed by LGBM. As the latter was faster to fit, it was the model ultimately chosen. The best input feature encoding to use, between integer or one-hot, turned out to be very dependent on the specific predictor model. In the case of LGBM, integer encoding proved to be superior. In general, the higher the number of training samples, the better the correlations. For the standard post-processing method, the optimisation strategy that resulted in larger average accuracy improvements was to use the SGD optimizer with initial learning rate set to 10^{-4} when the subnets to be fine-tuned were found within a single-exit supernet. SGD should be replaced by AdamW for subnets derived from early-exit supernets. For the AEP-based post-processing method, the best approach was to use the AdamW optimizer with the initial learning rate set to 10^{-4} and the UNIF exit weighting strategy. Between the architectural changes to the supernets and the algorithmic changes, the architectural changes provided the greatest improvements over the baseline. Subnets derived from early-exit supernets significantly outperformed those derived from the baseline OFAMobileNetV3 supernet, while reducing the number of parameters by up to 10 times, the number of MACs by up to 5 times, and latency by more than half. The presence of dense skip connec-

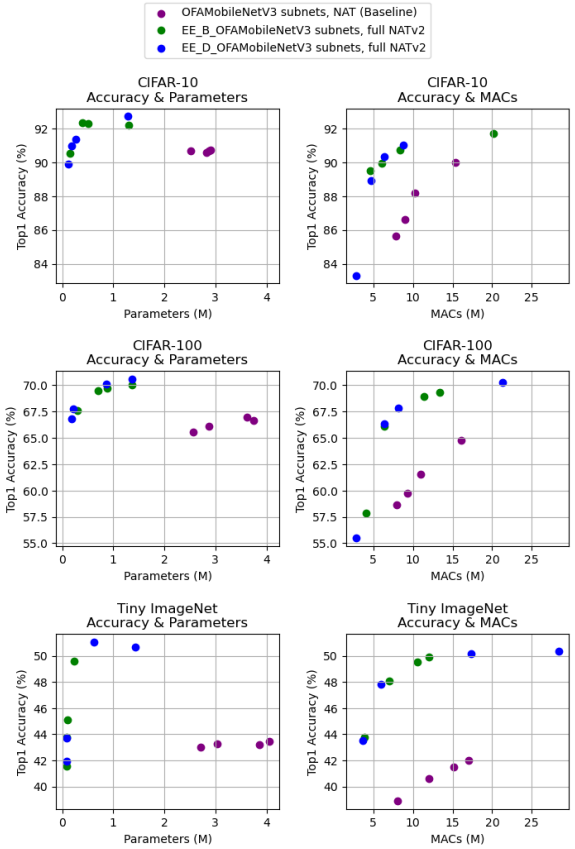

Figure 5: Results obtained by the baseline subnets after the first NATv2 experimental phase, similar to the original NAT, compared to the results of the best subnets obtained using the full version of NATv2. Different rows show the results obtained on different datasets. Different columns show the results obtained for different additional target objectives. Better results will be closer to the top left-hand corner of the plot.

tions into the networks also led to small accuracy gains, regardless of the number of exits. Parallel subnets were quickly abandoned as they didn't meet expectations. The use of better supernets as a starting point in the fourth phase of the experiment also provided a performance boost. Of the algorithmic changes, the addition of the post-processing step was by far the most helpful. Figure 5 compares, for the "accuracy & parameters" and "accuracy & MACs" bi-objective optimisations, the results obtained by the subnets found within OFAMobileNetV3 after the first set of NATv2 experiments, i.e. the scenario most similar to NAT, and the best subnets found (those derived from EE_B_OFAMobileNetV3 and EE_D_OFAMobileNetV3) using the full version of NATv2, using the standard post-processing so as not to alter the second objective scores. The superiority of the new subnets is obvious. If accuracy is the main concern, and thus the AEP-based post-processing can be used, comparing only the most accurate networks from the baseline subnets group and the full NATv2 subnets group, the full version of NATv2 leads to jumps in accuracy from 90.73% to 93.06% on CIFAR-10, from 66.93% to 72.03% on CIFAR-100, and from 43.45% to 54.31% on Tiny ImageNet.

5. Conclusion

This thesis journey, which started with AEP and progressed through OFAv2, has concluded with NATv2. In it, we demonstrated how the use of early exits in convolutional neural networks, via the the weighted ensemble and pruning step of the newly proposed AEP technique, resulted in smaller and more accurate image classification networks. OFAv2 introduced early exits, dense skip connections and parallel blocks into the OFAMobileNetV3 supernet, as well as a number of other improvements and extensions to the OFA algorithm. As a result, some of the newly created supernets were found to be significantly better than the baseline supernet. Finally, NATv2 combined the techniques and results of the previous two projects with modifications to the original NAT algorithm. In the end, better subnets were found for all datasets.
In terms of future developments, AEP could be improved by jointly optimising network and losses weights, searching for optimal outputs weights, and performing a multi-objective evaluation in the pruning step. Regarding OFAv2, a more in-depth study on the impact of different blocks in parallel networks could prove useful. Lastly, NATv2 could be extended to allow the search of subnets with multiple exits.

References

[1] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," *arXiv preprint arXiv:1908.09791*, 2019.

[2] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, and V. N. Boddeti, "Neural architecture transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 2971–2989, 2021.