Optimizing Accuracy/Score Tradeoff: Final Presentation

Mehdi BERRADA & Selim ZIGHED

26/03/2025

Overview

Baseline Model

- The baseline model used in this project is the ***ResNet-18***.
- *Model Hyperparameters* :
  - *Epochs* : 200
  - *Batch Size* : 128
  - *Loss Function* : Cross-entropy
  - *Optimizer* : SGD
  - *Learning Rate* : 0.01, *Momentum* : 0.9,
  - *Weight Decay* : 5.10^{-4}, *Scheduler* : CosineAnnealingLR

Baseline Model

- The baseline model used in this project is the ***ResNet-18***.
- ***Model Hyperparameters*** :
  - *Epochs* : 200
  - *Batch Size* : 128
  - *Loss Function* : Cross-entropy
  - *Optimizer* : SGD
  - *Learning Rate* : 0.01, *Momentum* : 0.9,
  - *Weight Decay* : 5.10^{-4}, *Scheduler* : CosineAnnealingLR

Baseline Model's Performance

- ***Accuracy*** : Our ResNet-18 model achieves an accuracy of 93.60%
- ***Total Parameters*** : 11173962 params
- ***FLOPs*** : 556651520 operations
- ***Average Inference Latency*** : 0.002321 sec
- ***Score to be optimized :*** 3.98778

Optimising Strategy

- **Structured Pruning** → 9 models (*different pruning amounts*).
- **Unstructured Pruning** → Applied to each structured pruned model, generating **36 models**.
- **Quantization** → Applied **FP16** to each pruned model.

Results for Pruning and Quantization

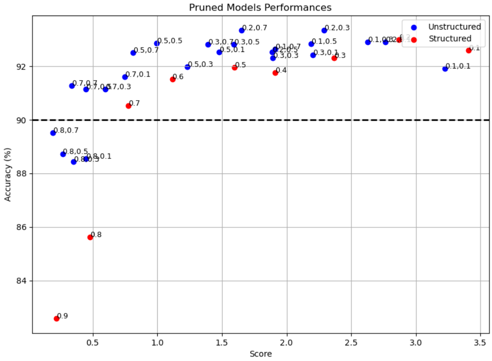

Figure – Trade-off Plot for Pruned and Quantized models

Depthwise Separable Convolutions

Depthwise Separable Convolutions decompose a standard convolution operation into two smaller steps :

- ***Depthwise Convolution :*** Applies a single filter to each input channel independently.
- ***Pointwise Convolution (1x1 Conv.) :*** Combines the outputs from the depthwise convolution across all channels.

Depthwise Separable Convolutions

Models	*Baseline*	*DSC Factorized*
Parameters	11173962	1439626
FLOPs	556651520	74277888
Accuracy	93.60%	91.93%
Score	3.98778	0.522354

Table – Before vs After Comparison

Grouped Convolutions

- Instead of using a single convolution across all input channels, the channels are divided into groups.
- Each group has its own set of filters, reducing the number of computations.

Grouped Convolutions : Varying Groups (G)

Models	*Baseline*	*G=2*	*G=8*
Parameters	11173962	5681226	1561674
FLOPs	556651520	282973184	77714432
Accuracy	93.60%	93.15%	90.31%
Score	3.98778	2.025123	0.556422

Table – Before vs After Comparison

After *fp16 Quantization*, the score drops by half with almost no impact on the precision, as the folowing plot shows.

Factorization Results

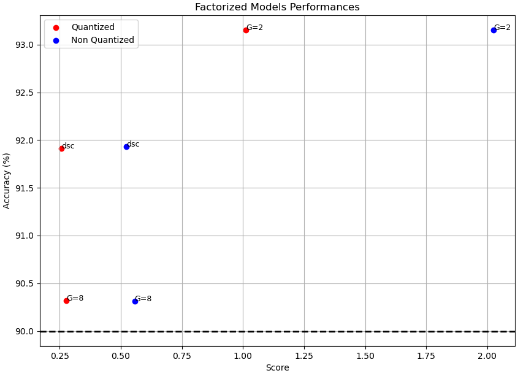

Figure – Trade-off Plot for Factorized Models

Knowledge Distillation

- Transfers useful knowledge from a powerful teacher to a lightweight student.
- Used the ResNet-18 baseline as the teacher and some of our models as students.

Knowledge Distillation Setup

- ***Teacher Model :*** Our Baseline ResNet18
  (The teacher is Pretrained)
- ***The input data*** are the same for both teacher and students
- ***Epochs :*** 30
- ***Loss :*** Cross Entropy + KL divergence

Further Optimization

Our main goal for now is to push some of our models closer to an ideal trade-off (90% accuracy while further reducing computational cost). The strategy is as follows :

- Distilling some already pruned models that don't reach 90% accuracy.
- Pruning and distilling when the accuracy is higher than 90%.

Pruned DSC Factorized Models

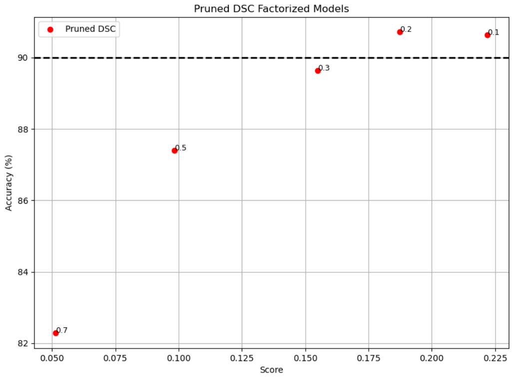

Figure – Pruned DSC Factorized Models Performances

Pruned DSC Factorized Models

Pruning Amount	Accuracy	Score
0.1	90.64%	0.221828
0.2	90.71%	0.18726
0.3	89.63%	0.155024
0.5	87.40%	0.098265
0.7	82.28%	0.051471

Table – Recap Performance Table of the Pruned DSC Factorized Models

Distillation Impact on Pruned DSC Factorized Models

- Given that the 0.1 and 0.2 times pruning already lead a good score/accuracy trade-off, and that the 0.7x pruned model is far from the condition of 90% accuracy, we will focus on the two remaining models.
- We use the baseline model to distill the 0.3 and the 0.5 times pruned ones, results are as follows :

Distillation Impact on Pruned DSC Factorized Models

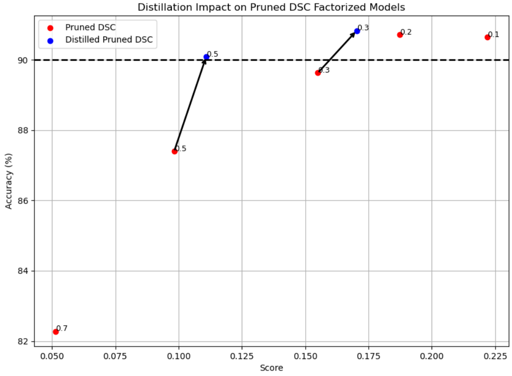

Figure – Distillation Impact on Pruned DSC Factorized Models

Distillation Impact on Pruned DSC Factorized Models

- We managed to reach the 90% accuracy barrier with distillation for both the two models.

Pruning	Accuracy	New Accuracy	Score	New Score
0.3	89.63%	90.83%	0.155024	0.170323
0.5	87.40%	90.09%	0.098265	0.110921

Table – Distillation Impact on the Pruned DSC Factorized Models

- The slight Score Increase can be explained by the readjustement of some pruned weights, yet the trade-off is still very good.

Pruning and Distillation at once

The below described process will be followed on different student models, mainly the ones that have undergone Different Factorization Techniques

- Load the teacher and the student models
- Prune the student model
- Start the training with the distillation loss
- Remove the pruned weights to make sure they are not readjusted

Pruning and Distillation Experiments

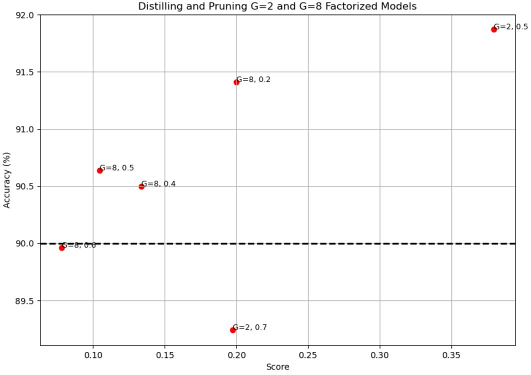

Figure – Distilling and Pruning factorized models

Trade-off Recap

Model	Accuracy	Score
$G = 8, 0.6$ pruning and distillation	89.96%	0.078058
$G = 8, 0.5$ pruning and distillation	90.64%	0.104438
Distilled $DSC, 0.5$ pruning	90.09%	0.110921
$G = 8, 0.4$ pruning and distillation	90.50%	0.133548
Distilled $DSC, 0.3$ pruning	90.83%	0.170323

Table – Generated Models Performance Table (from lowest score on)

New Test : Ensemble Method

How it works :

- Multiple optimized models make independent predictions.
- The final classification is determined by majority voting.
- Can be applied to a diverse set of models (e.g., pruned, quantized, factorized).

Ensemble Method

- **Advantages :**
  - Improves prediction stability and generalization.
  - Reduces reliance on a single highly compressed model.
  - Allows leveraging different trade-offs between accuracy and efficiency.
- **Project wise Limitation**
  - Calculation of the Score to optimize

Ensemble Method Test

- We used the five models in the last table
- The best of these models reaches 90.83% accuracy
- ***Results :***
  - 93.47% Accuracy
  - As per the score, the most logical hypothesis is to sum the scores of the involved models, in this case, we get 0.597288, which is a great trade-off.

Trade-off Plot

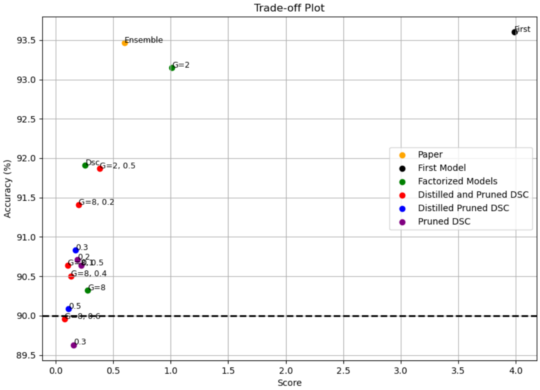

Figure – Trade-Off Plot