deepPP weekly meeting

17/03/2023

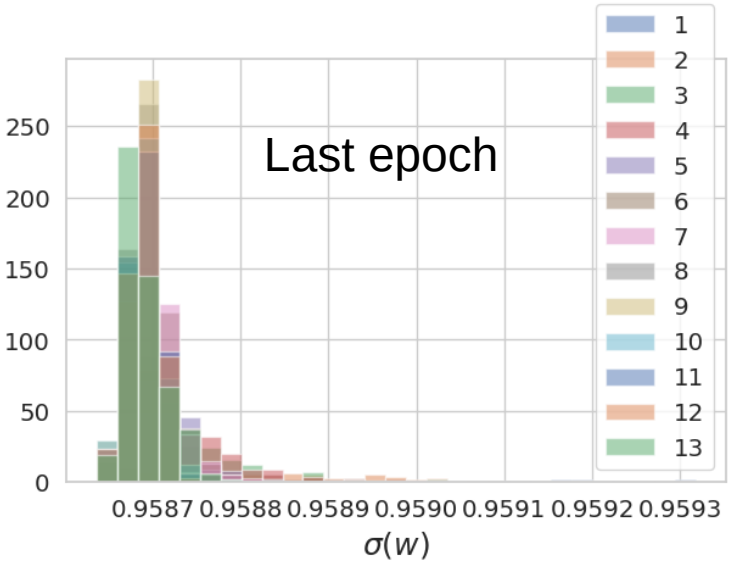
deeppp

AutoPruner on VGG16

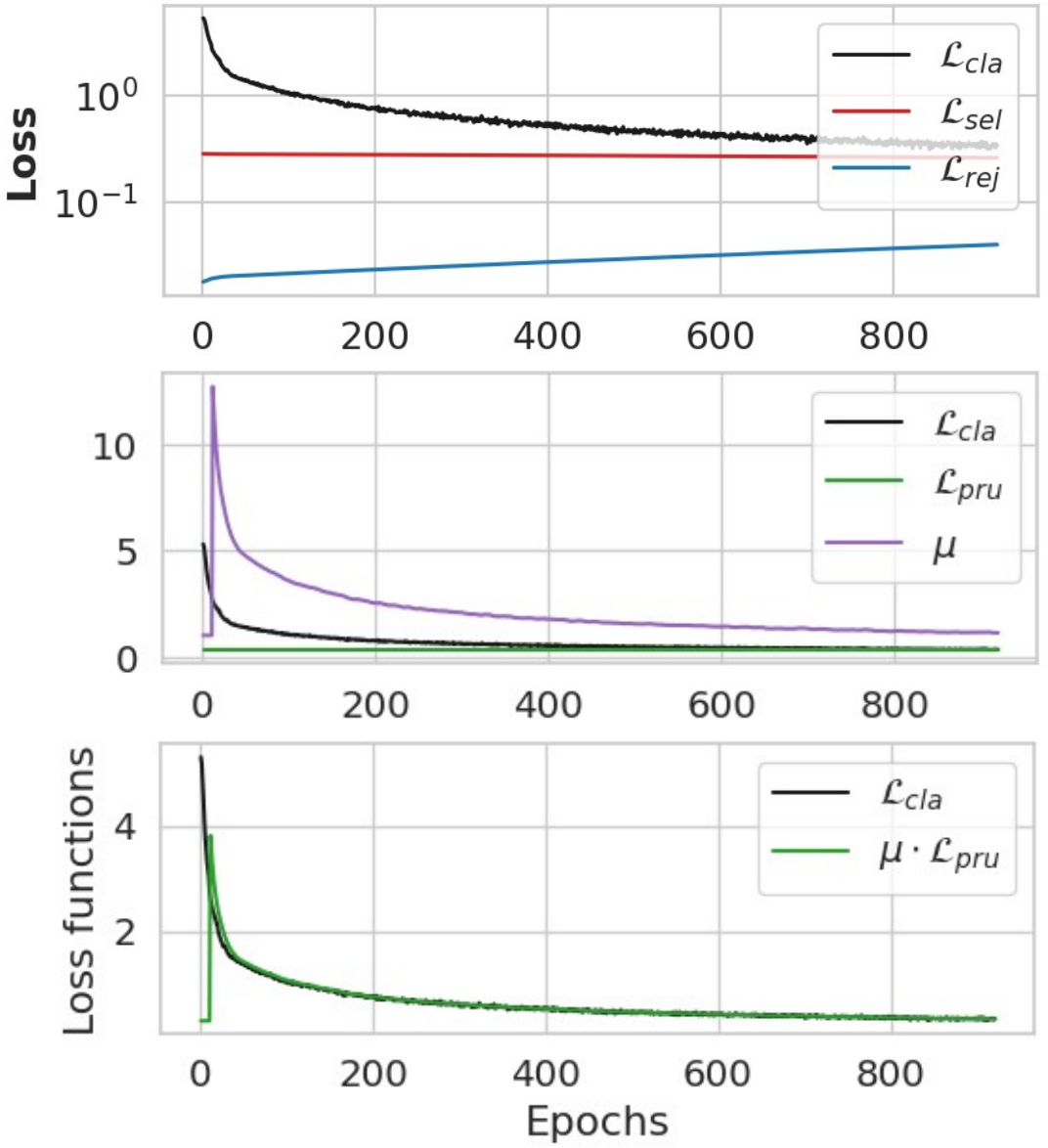
- Network architecture & implementation details:
  - 13 convolutional layers + 3 fc layers adapted for the dataset used (200 classes), no avarage pooling after the convolutional block, use of dropout for the fc block
  - AutoPruner after each convolutional layer to prune filters, early activation
  - Implementation of the running parameter μ to balance loss terms

Results

- 70% filters required, 920 training epochs



L_{pru} is never > L_{cla} and the pruning never happens

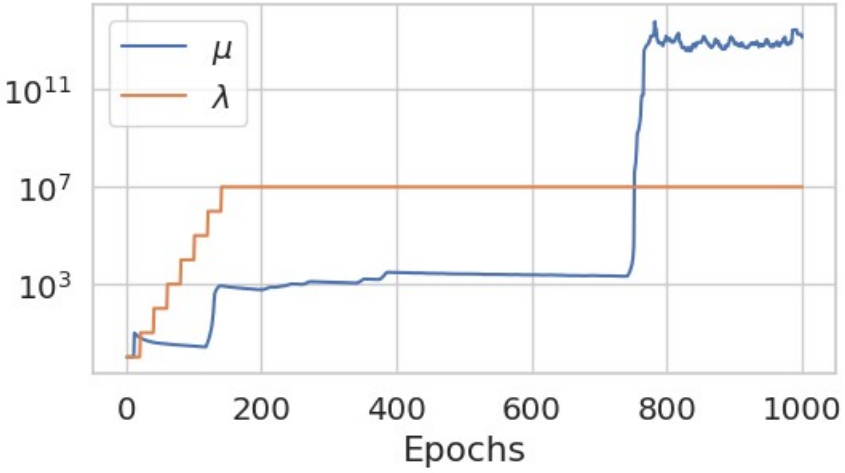


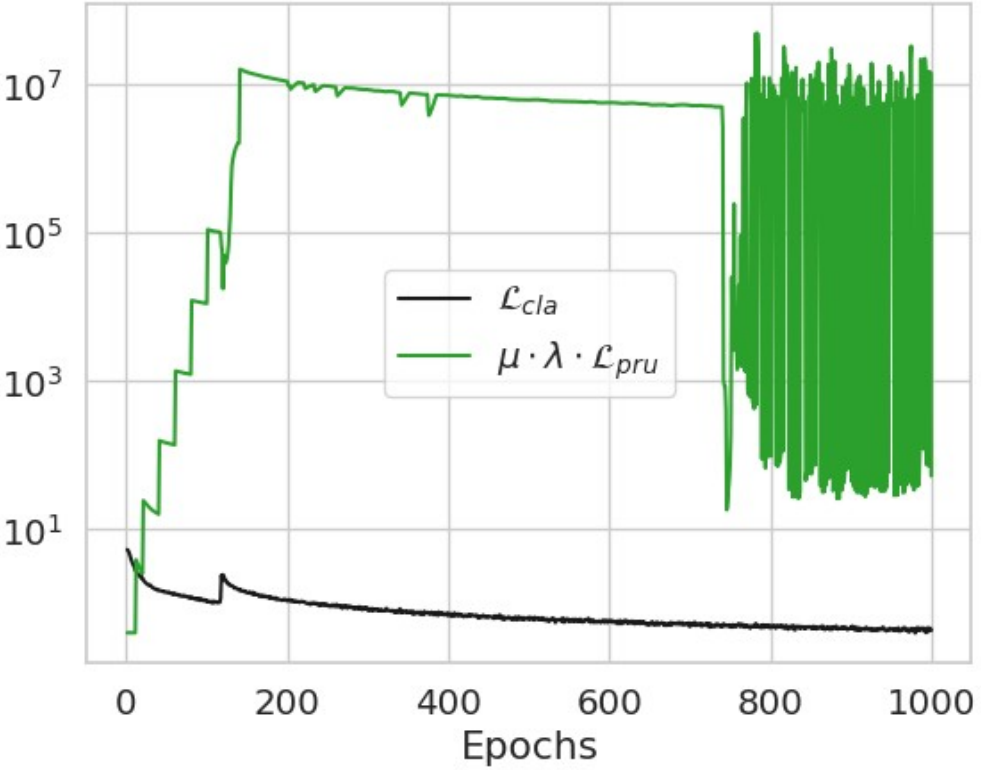
Solution

- $L_{tot} = L_{cla} + \mu * \lambda * L_{pru}$

  ➔ λ=λ*10 every 20 epochs, from λ=1 to λ=10^7

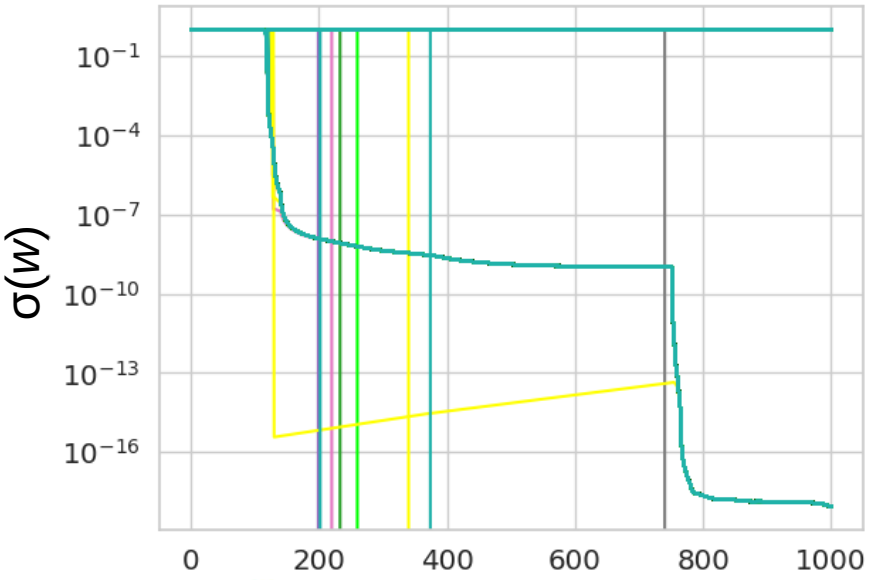
  Tested for 60% filters required, 1000 epochs

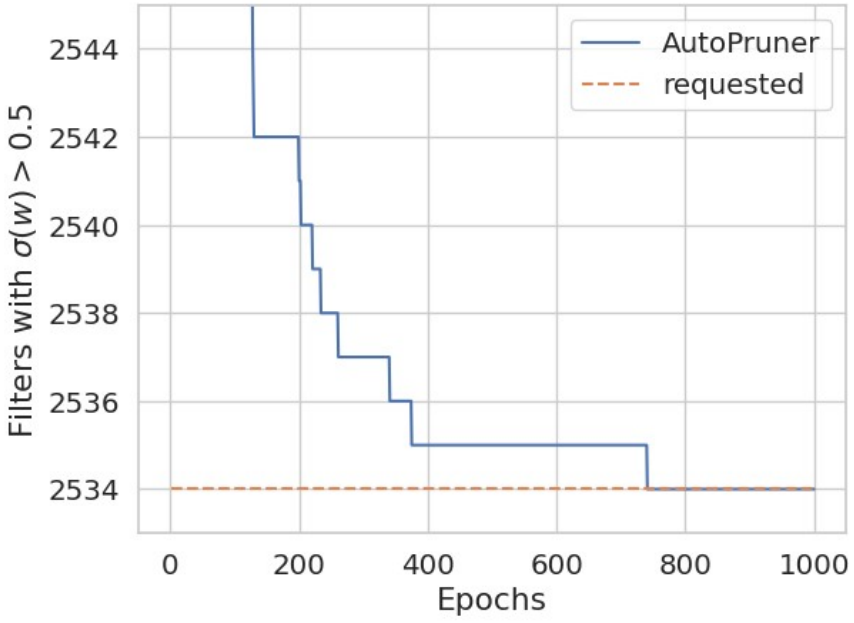


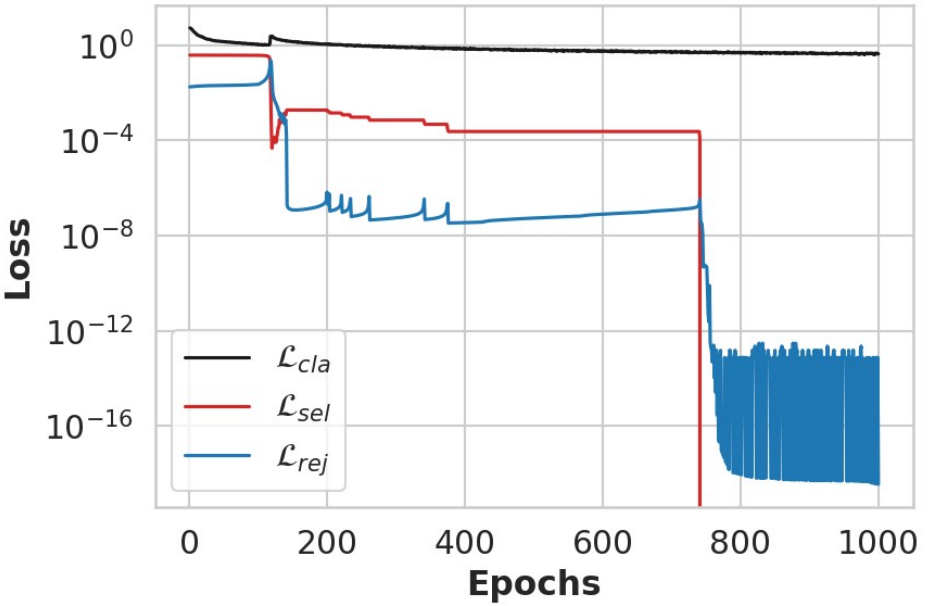


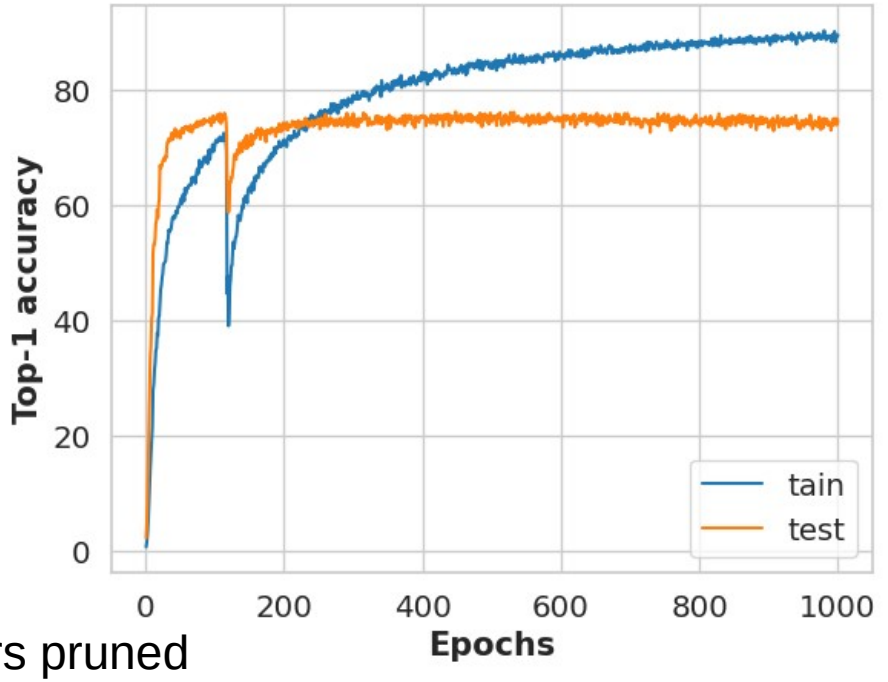
The pruning term becomes dominant at a certain point and the pruning can take place

Results





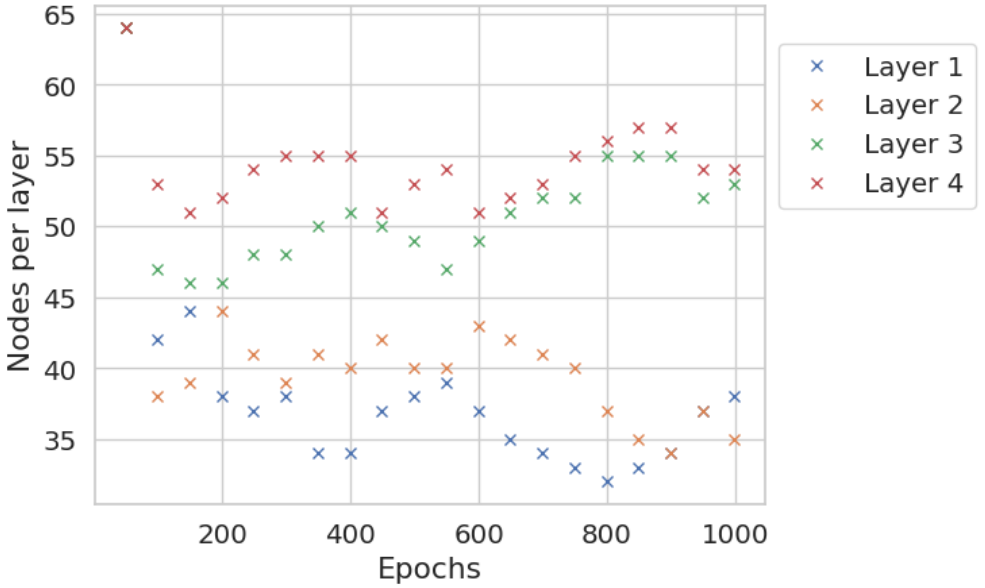




- ✔ Filters pruned
- ✔ Constraint satisfied
- ✔ Good accuracy (but comparison with the full model still missing)

What’s missing

- Still need to look at performance of these architectures



- Performance comparison between models with architectures determined by pruning and uniformely reduced models

What's missing

- Comparison with full VGG16 model
- Comparison with VGG16 trained with only the filters selected by AutoPruner
- Look at results with different compression rates
- Improve the implementation of this new parameter λ (need to justify it, need to think of a start&stop criterion)
- Look at the results of FC-DNNs: what should we do when AP does not select nodes in one or more layers? Still under investigation

Work in progress & todo list

- ACAT proceeding (deadline: 19/03/2023)
- Run GN1/SALT
- Try to reproduce Greta’s steps with Marco’s framework for data/MC comparison