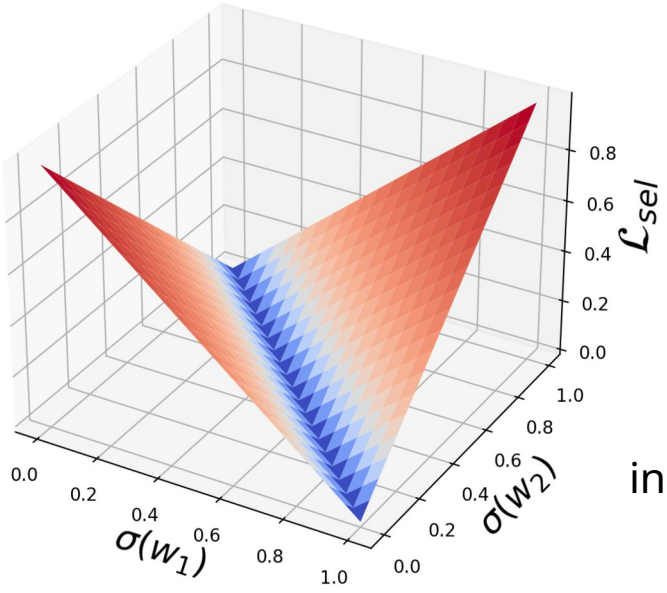
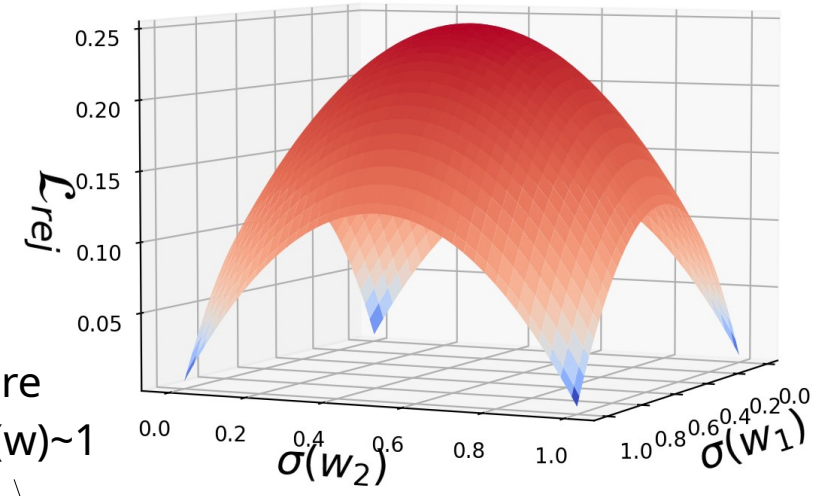
deepPP weekly meeting

12/05/2023

deeppp

Pruner loss functions

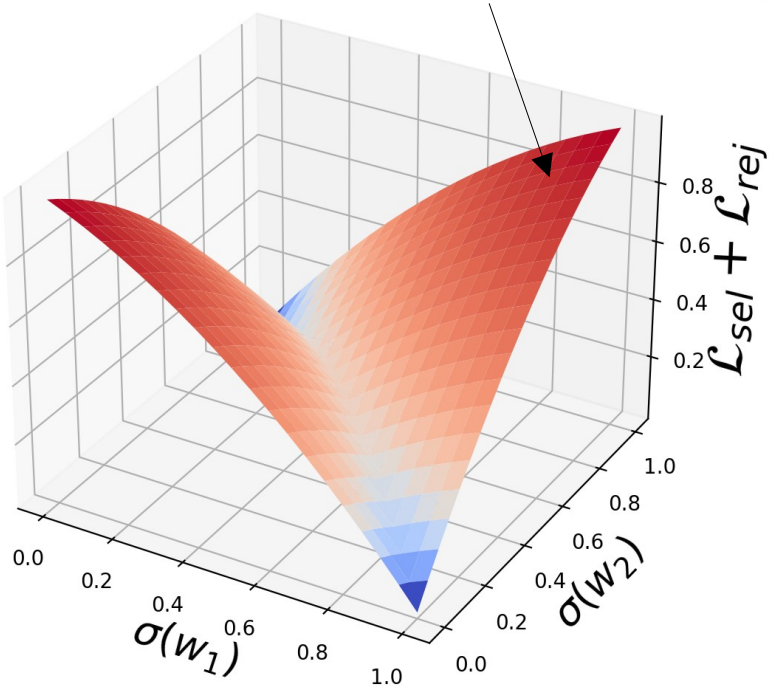




Pruner weights are initialized to have σ(w)~1

$$\mathcal{L}_{\text{sel}} = \frac{|\sum_{i=1}^{N} \sigma(w_i) - n|}{N}$$

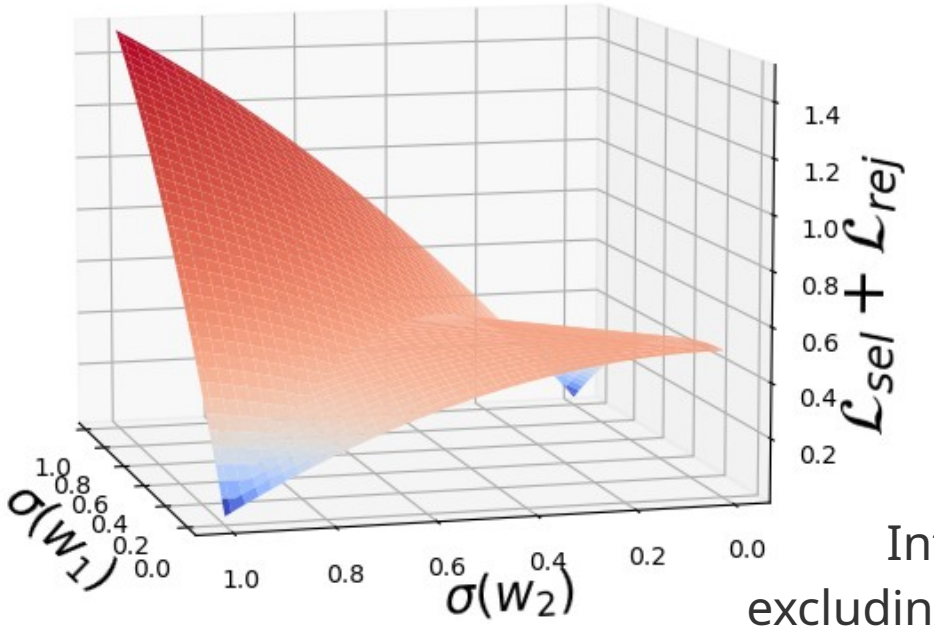
$$\mathcal{L}_{\text{rej}} = \frac{\sum_{i=1}^{N} \sigma(w_i)(1-\sigma(w_i))}{N}$$



➔ Is it possible to modify the loss in such a way

1) that the point (1,1,...,1) is no longer a minimum point for L_{rej}
2) as to increase the slope of L_{pru} in that region?

Modified rejection loss



$$f(x,y) = \frac{x(1-x)+y(1-y)+N\cdot xy}{N}$$

$$\mathcal{L}_{\text{rej}} = \frac{\sum_{i=1}^{N}\sigma(w_i)[1-\sigma(w_i)]+N\prod_{i=1}^{N}\sigma(w_i)}{N}$$

Introducing a term $\prod_{i=1}^{N}\sigma(w_i)$ has the effect of excluding the point (1,1,...,1) from the minima of L_{rej} but does not increase the slope of L_{pru} in that region because

$$\prod_{i=1}^{N}\sigma(w_i) \sim \sigma(w_i)^N \to 0$$ for big values of N and $\sigma(w_i) < 1$

➔ Result of training with this loss function: the problem of weights updated very slowly still persists, so the reason is not due to the fact that (1,1,...,1) is a minimum point for L_{rej} (therefore it is not a strong point of attraction)

Increasing the slope

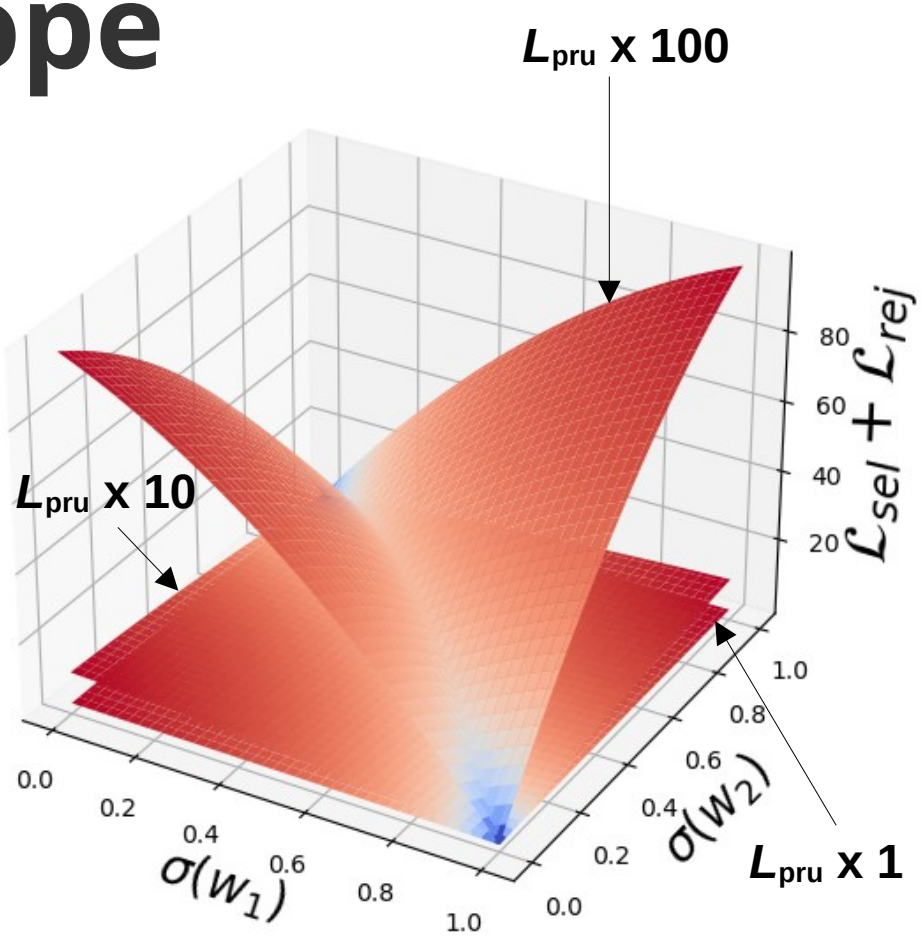
- Is increasing the slope of L_{pru} the key?

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

$\mu = \langle L_{cla} \rangle_{\text{last 10 epochs}} / (\lambda * \langle L_{pru} \rangle_{\text{last 10 epochs}})$

Tested with

- $\lambda=\lambda*10$ every 20 epochs, from $\lambda=1$ to $\lambda=10^7$
- $\lambda=10^5$, $\lambda=10^6$, $\lambda=10^7$



➔ Not working: weights still updated too little and too drastically

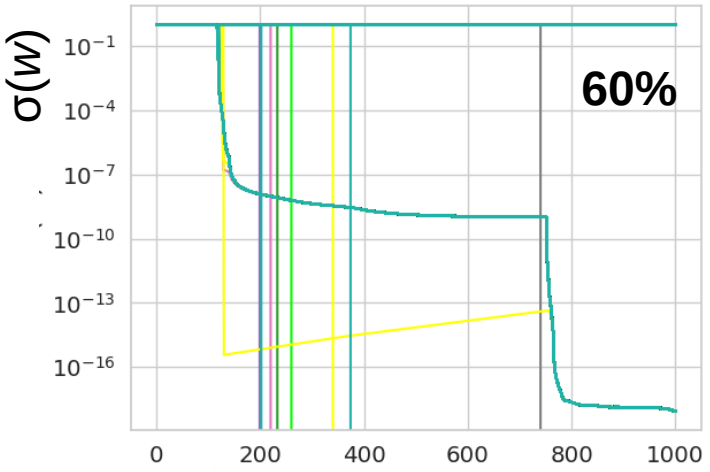
What really works

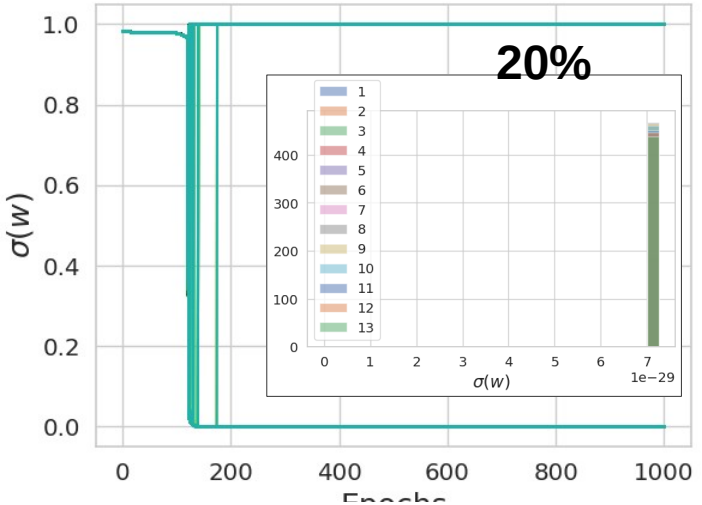
- In general, pruning occurs well and completely only if $L_{tot} = L_{cla} + \alpha * L_{pru}$ with α **increasing gradually**
  - When we set a fixed value for α the weights are updated too little or too drastically
  - Defining α as $\langle L_{cla} \rangle_{\text{last 10 epochs}} / \langle L_{pru} \rangle_{\text{last 10 epochs}}$ works fine only if L_{pru} decreases in such a way that α increases gradually
- That's why introducing $L_{tot} = L_{cla} + \mu * \lambda * L_{pru}$ with $\lambda = \lambda * 10$ every 20 epochs from $\lambda=1$ to $\lambda=10^7$ is working fine (see next slide)

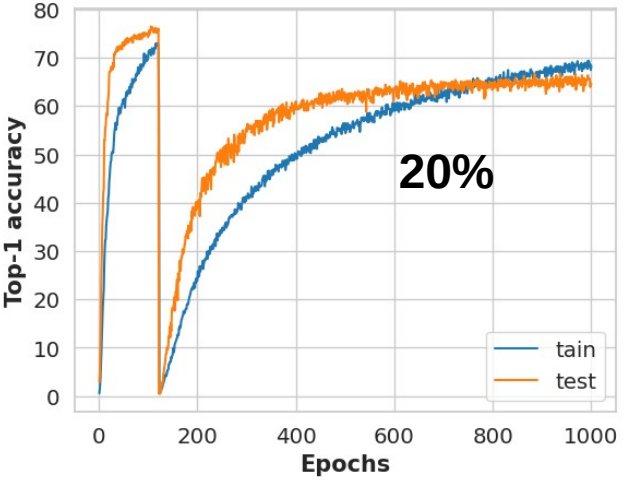
➔ IDEA: define λ in such a way that it increases gradually if the variance of the weights is not changing

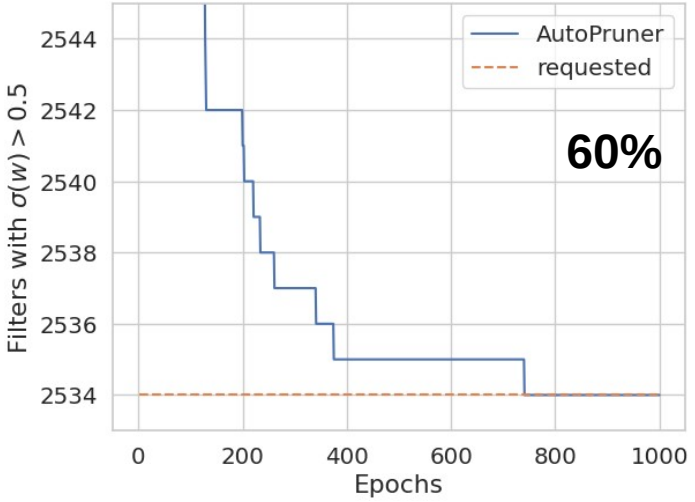
Results with increasing λ

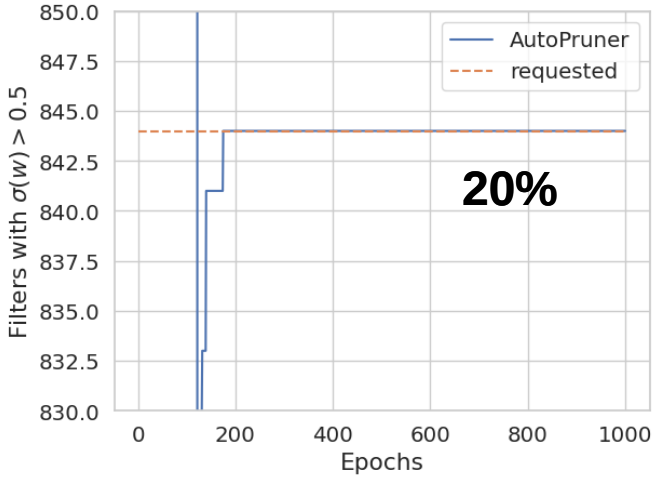
- $L_{tot} = L_{cla} + \mu * \lambda * L_{pru}$ with λ=λ*10 every 20 epochs, from λ=1 to λ=10^7

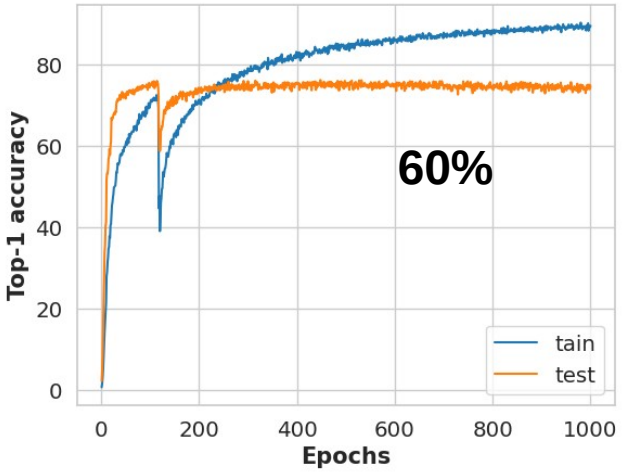












Alternative strategy

- Use Markov chain instead of back propagation to update pruner weights
  - ➔ $w_{new} = w_{old} + \delta\ rand[-1,1]$
  - ➔ Weights updated only if $L_{tot}^{new} < L_{tot}^{old}$

PLAN: investigate both strategies