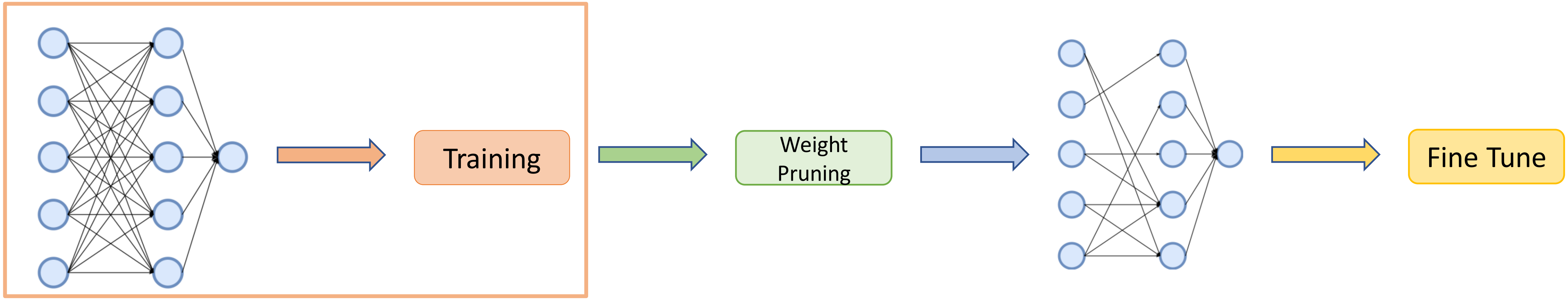
Winning the Lottery Ahead of Time: Efficient Early Network Pruning

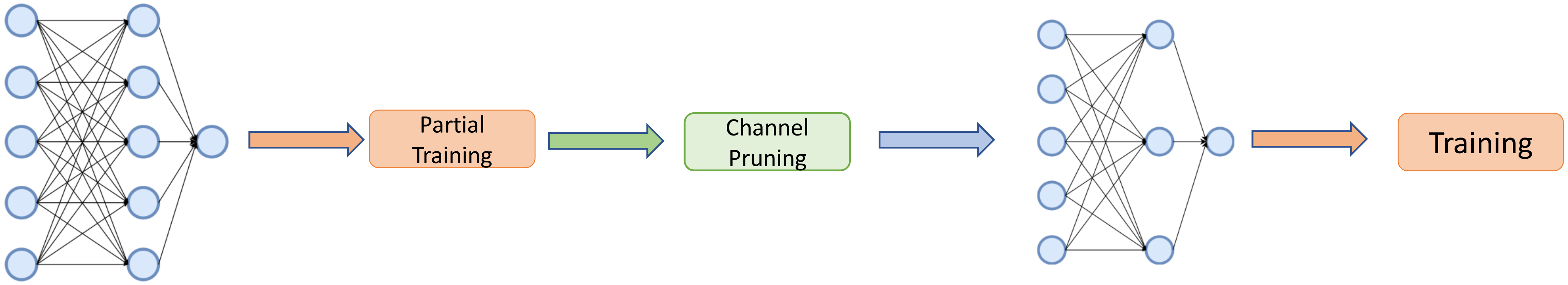
John Rachwan, Daniel Zügner, Bertrand Charpentier, Simon Geisler,
Morgane Ayle, Stephan Günnemann

Network Pruning – Previous Works



Drawbacks:
- Training phase is as expensive as training the Dense model
- Weight pruning does not practically make weight matrices smaller

Network Pruning – Our Work



Contributions:

- We remove parameters that least affect the Neural Tangent Kernel
- We extract subnetworks when the Neural Tangent Kernel transitions to a stable state
- We remove entire channels in order to practically reduce weight matrices

Background: NTK, GF

- In the infinite width regime, NNs simplify to linear models with a kernel called **Neural Tangent Kernel** (NTK):

$$NTK(\theta) = g_Y(\Theta_t)^T g_Y(\Theta_t)$$

- Under the same regime, the NTK is **constant** throughout training

- **Gradient flow** (GF) is used to study optimization dynamics and is typically approximated by taking the L2 norm of the gradients of the network:

$$GF(\theta) = g_L(\Theta_t)^T g_L(\Theta_t)$$

How to Prune?

- We present the following relationship between the NTK and GF:

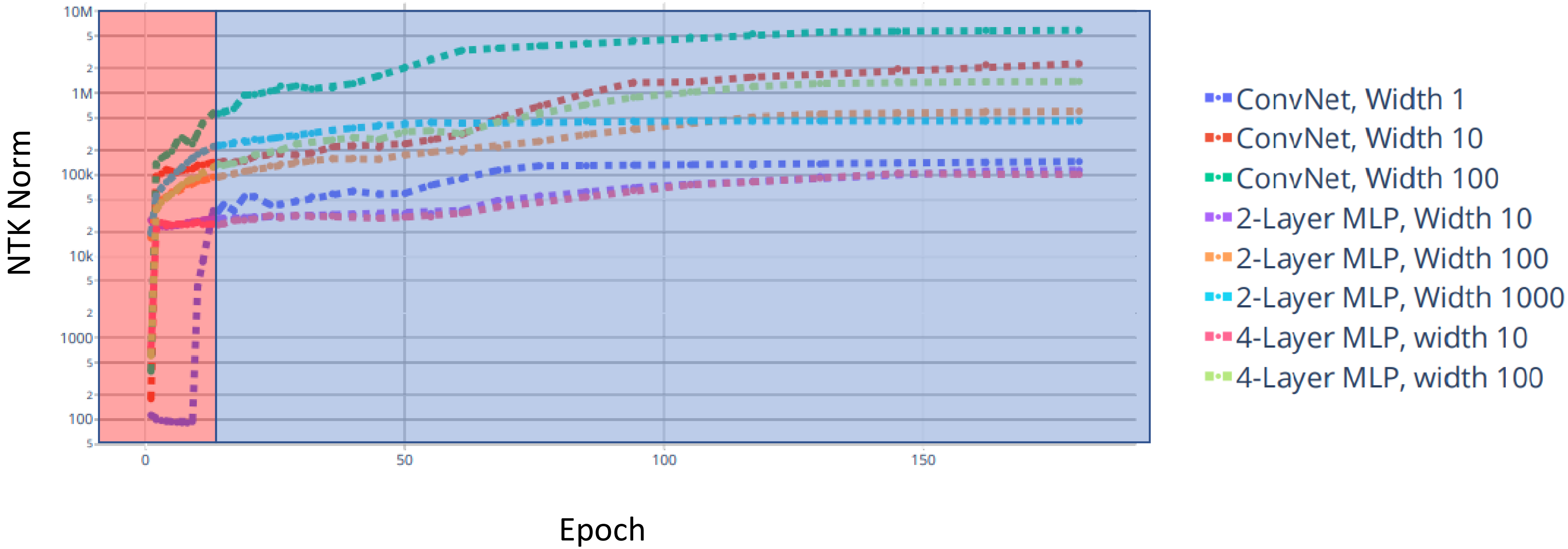
$$\begin{aligned} GF &= g_L(\Theta_t)^T g_L(\Theta_t) \\ &= g_L(Y)^T g_Y(\Theta_t)^T g_Y(\Theta_t) g_L(Y) \\ &= g_L(Y)^T NTK\, g_L(Y) \end{aligned}$$

- Knowing that preserving the GF also preserves gradient of the loss w.r.t. the prediction $g_L(Y)$. We can conclude that preserving the GF also preserves the NTK.

- In order to preserve the GF, we can use the following importance score:

$$I(\Theta_l) = \left|\Theta_l^T H_L(\Theta_l) g_L(\Theta_l)\right|$$

Lazy Kernel Regime

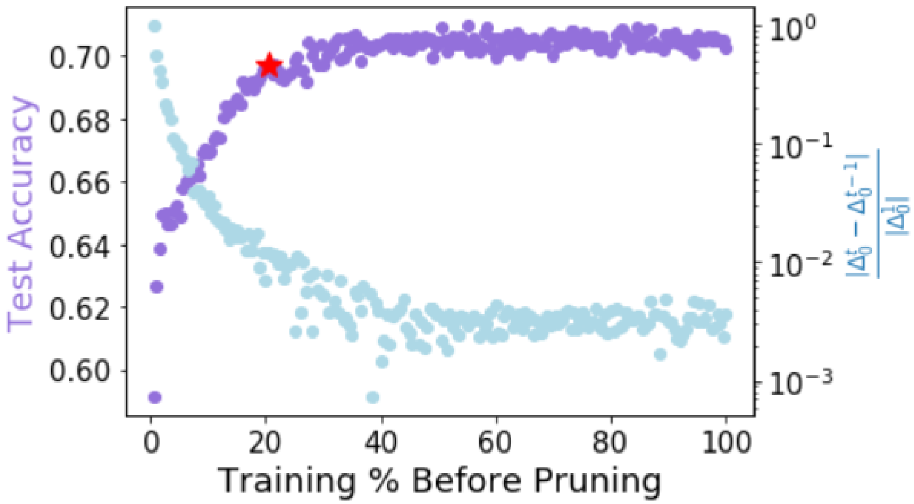


Goldblum, Micah & Geiping, Jonas & Schwarzschild, Avi & Moeller, Michael & Goldstein, Tom. (2019). Truth or Backpropaganda? An Empirical Investigation of Deep Learning Theory.

When to Prune?

- Constancy of the NTK is a consequence of a **constant weight norm during training**
- Hence, we can detect the transition to the *lazy kernel regime* when:

$$\frac{|\Delta_0^t - \Delta_0^{t-1}|}{|\Delta_0^1|} \simeq 0 \quad / \quad \Delta_0^t = \frac{||\Theta(t) - \Theta(0)||^2}{||\Theta(0)||^2}$$



Why to Prune?

- In order to perform structured pruning, we need to score the layer channels instead of weights
- This can be done by introducing learnable gates c=1 to the output of each layer whose gradients would represent the channel's:

$$f_l(\Theta_l * x + b_l) = f_l(c_l(\Theta_l * x + b_l))$$

- We then score the importance of each activation using:

$$I(f_l) = |H_L(c_l) g_L(c_l)|$$

Results – Image Classification

	Method	Test accuracy ↑	Weight sparsity	Node sparsity	Training time (h) ↓	Batch time (ms) ↓	GPU RAM (GB) ↓	Disk (MB) ↓	Emissions (g) ↓
-	Dense	62.1%	-	-	0.77	114	1.03	1745	88
Structured	Random-S	53.9%	98.0%	86.0%	**0.59**	53	0.23	35	29
	SNAP	49.3%	98.0%	89.0%	0.67	54	0.16	36	33
	CroP-S	57.4%	98.0%	89.0%	0.61	46	0.23	36	35
	CroPit-S	56.5%	98.1%	89.0%	0.62	**44**	0.23	33	30
	EarlyBird	60.7%	98.0%	89.0%	0.56	68	0.20	36	62
	EarlyCroP-S	**62.2%**	97.9%	88.0%	0.64	69	0.23	36	58
	GateDecorators	55.0%	97.9%	87.0%	0.61	78	0.23	36	68
	EfficientConvNets	29.5%	98.0%	86.0%	0.72	55	0.24	36	83
Unstructured	Random-U	55.8%	98.0%	-	0.74	118	1.23	35	99
	SNIP	61.9%	98.0%	-	0.79	109	1.24	35	90
	GRASP	63.4%	98.0%	-	0.79	113	1.24	35	91
	CroP-U	63.8%	98.0%	-	**0.74**	109	1.23	35	94
	CroPit-U	56.3%	98.0%	-	**0.74**	111	1.23	35	91
	EarlyCroP-U	**65.1%**	98.0%	-	**0.74**	109	1.23	35	91
	LTR	64.7%	98.0%	-	3.44	109	1.28	35	301

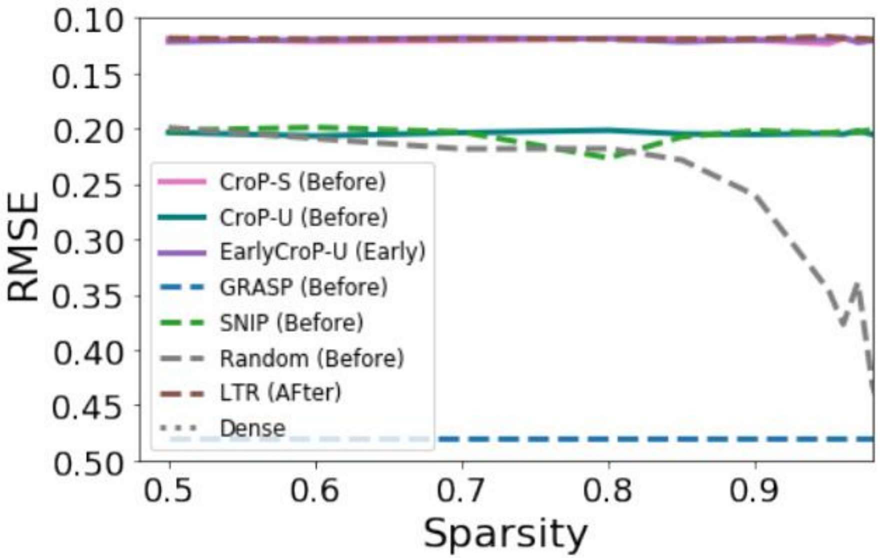
Results – Large Models

Model	Test acc.	Weight sparsity	Node sparsity	Epochs	Training time (h)	VRAM (GB)	Emissi-ons (g)
RN48	92.4%	-	-	30	4.60	18.84	634
RN16	92.1%	-	-	30	4.02	3.89	445
RN48-S	**92.5%**	98.5%	89.9%	30	**0.64**	3.56	47
RN48-S	93.2%	98.5%	89.9%	80	2.60	3.56	194

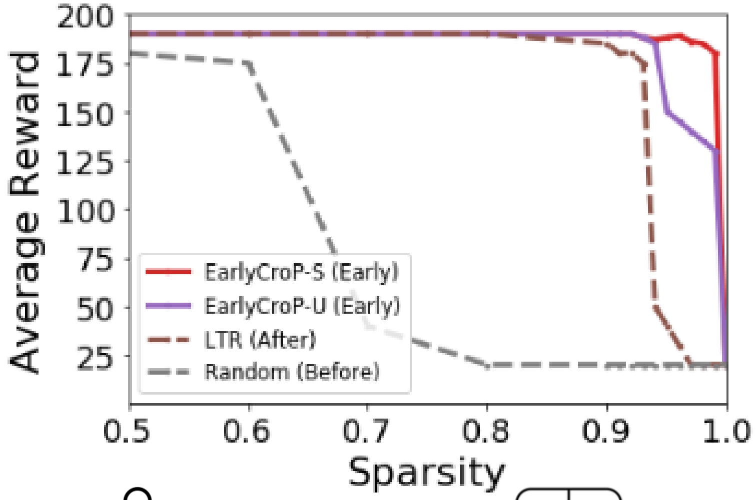
ResNext

CIFAR10

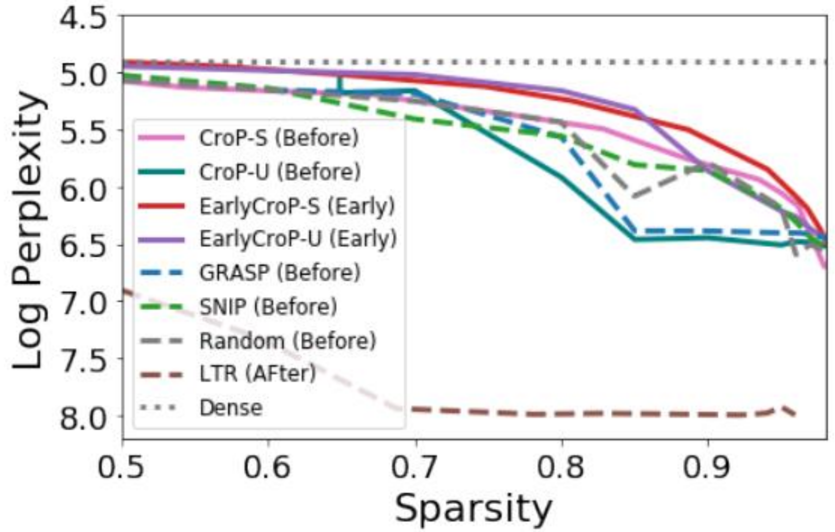
Results – Other tasks



FCRN NYU- DE



MLP CartPole-V0



PSMM PTB

Winning the Lottery Ahead of Time: Efficient Early Network Pruning

John Rachwan, Daniel Zügner, Bertrand Charpentier, Simon Geisler,
Morgane Ayle, Stephan Günnemann