Carnegie Mellon University
Language Technologies Institute

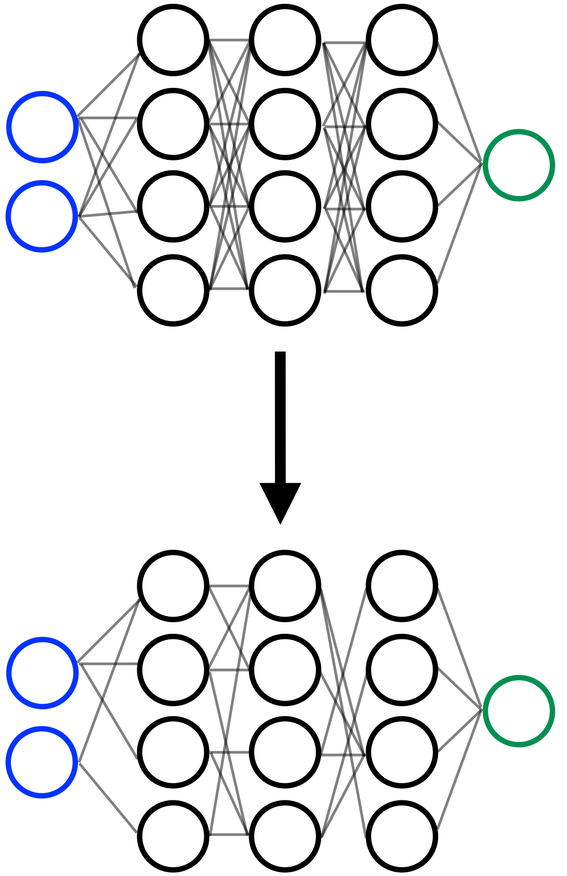
Lecture 7: Compression II

Knowledge distillation & Pruning

Yonatan Bisk & Emma Strubell

Model pruning: inducing sparsity

- **Key idea:** Take a large, accurate model, remove unimportant parameters.
- Modeling decisions: Which parameters to remove?
  - **Magnitude pruning:** Remove weights closest to 0.
  - Original idea: Magnitude ≠ saliency; use 2nd derivatives instead.



N(eur)IPS 1989

Optimal Brain Damage

Yann Le Cun, John S. Denker and Sara A. Solla
AT&T Bell Laboratories, Holmdel, N. J. 07733

Model pruning: inducing sparsity

- **Key idea:** Take a large, accurate model, remove unimportant parameters.
- Modeling decisions: Which parameters to remove?
  - **Magnitude pruning:** Remove weights closest to 0.
  - Original idea: Magnitude ≠ saliency; use 2nd derivatives instead.

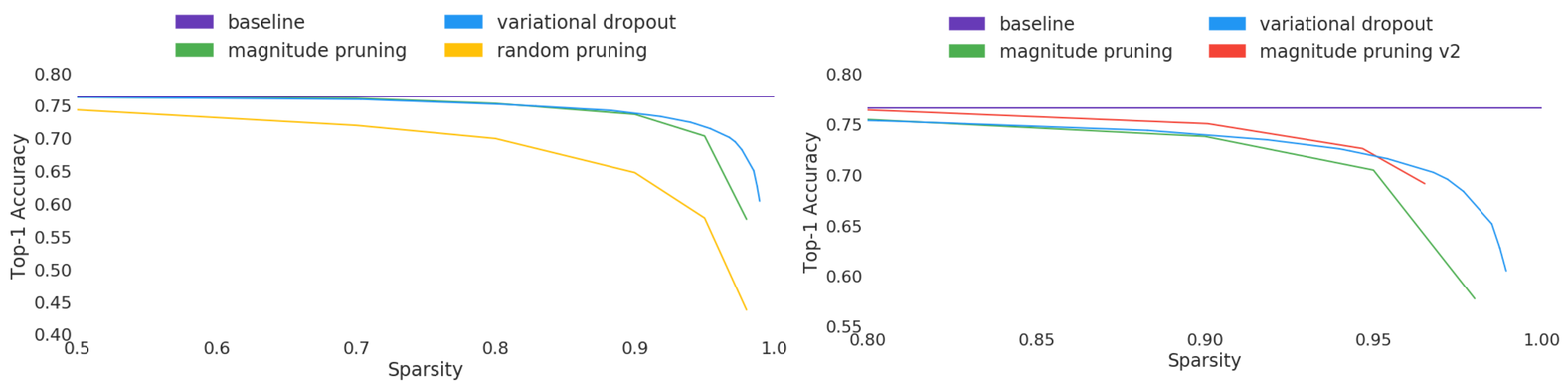


Figure 5. **Sparsity-accuracy trade-off curves for ResNet-50**

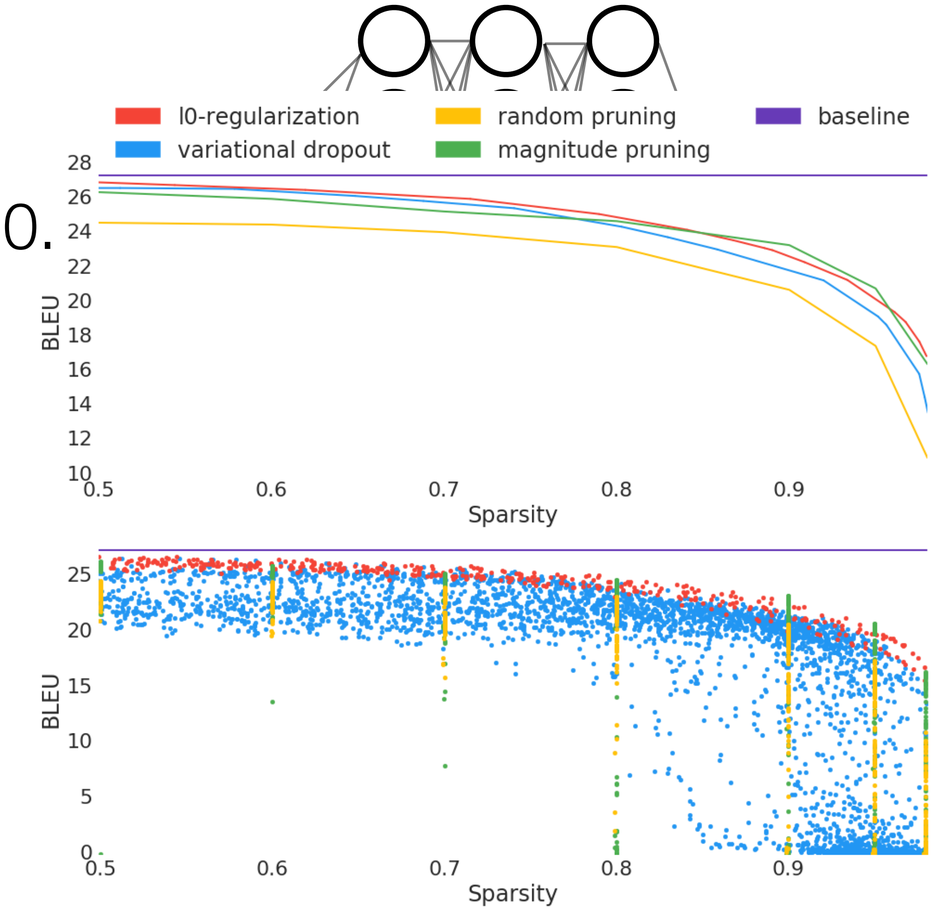
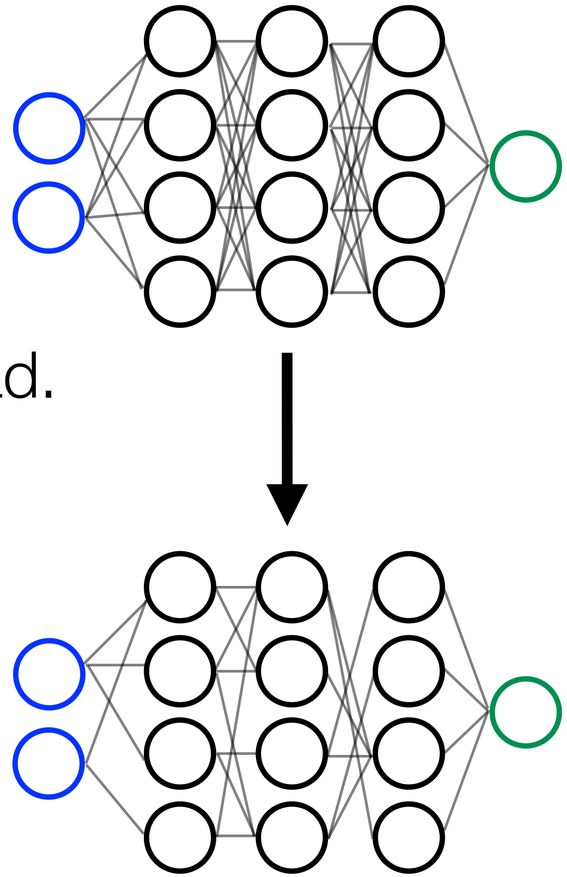


Figure 1. **Sparsity-BLEU trade-off curves for the Transformer.**

T. Gale, E. Elsen, S. Hooker. The State of Sparsity in Deep Neural Networks. ICML Joint Workshop on On-Device Machine Learning and Compact CNN Representations. 2019.

Model pruning: inducing sparsity

- **Key idea:** Take a large, accurate model, remove unimportant parameters.
- Modeling decisions: Which parameters to remove?
  - **Magnitude pruning:** Remove weights closest to 0.
  - Original idea: Magnitude ≠ saliency; use 2nd derivatives instead.
- Accuracy-efficiency trade-off:
  - Most accurate: Prune at the level of single parameters.
  - But, hardware determines efficiency.
  - **Structured pruning:** Take into account network structure & hardware considerations to prune entire rows, cols, or tensors.



Why does pruning work?

Why do Random Forests & Projections work?

High probability, $p \gg 0$ as number of dimensions increase, that any given separator will be within margin δ of correctly separating large regions of the input space.

Discovering Neural Wirings (Wortsman et al. 2019)

Instead of changing the weights of the network, you can simply change which weights of the random network to keep.

Holographic/Redundant Representations

Methods for robustness (e.g. dropout) create ensembled/redundant representations — only “one” is necessary for prediction (but may lose robustness).

Overparameterization makes training easier

Overparameterization makes optimization landscape convex-ish (Du et al. 2019, Haeffele and Vidal 2017).

Lottery Ticket Hypothesis

Dense networks contain sparse subnetworks that can match test accuracy of the original network.

Lottery Ticket Hypothesis (Frankle & Carbin, 2019)

- "*A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations*."
- Good subnetwork called a **winning ticket**.
- Larger networks more likely to contain winning tickets (more random subnetworks).
- **Iterative pruning:** Prune a bit at a time, not all at once, to achieve high pruning rate. Re-train after each iteration.
- **Rewinding:** When re-training a pruned network, *rewind* the weights to their randomly initialized values, or close to it.

Optimizing for compressible models

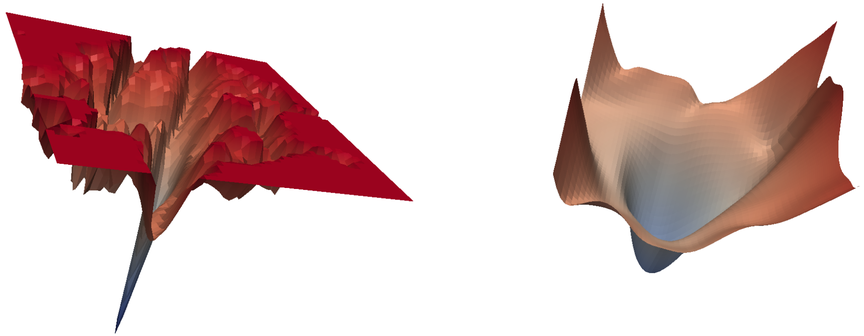
Can we improve compression by learning simpler functions in the first place?

Yes: Train **flat**, then compress.

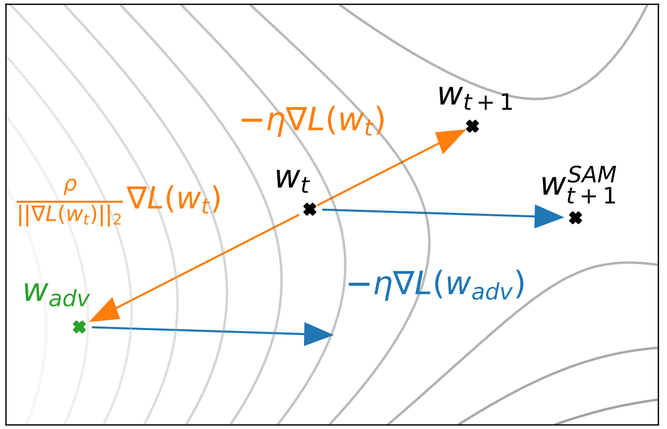
- **Flat minima**: *large connected regions in weight-space where error remains approximately constant* (Hochreiter & Schmidhuber, 1996)

- Low-complexity, high-generalization solutions (Wu et al. 2017; Keskar et al. 2017; Hao et al. 2019; Neyshabur et al. 2020)

- **How?** Sharpness-Aware Minimization (SAM) Foret et al. 2021



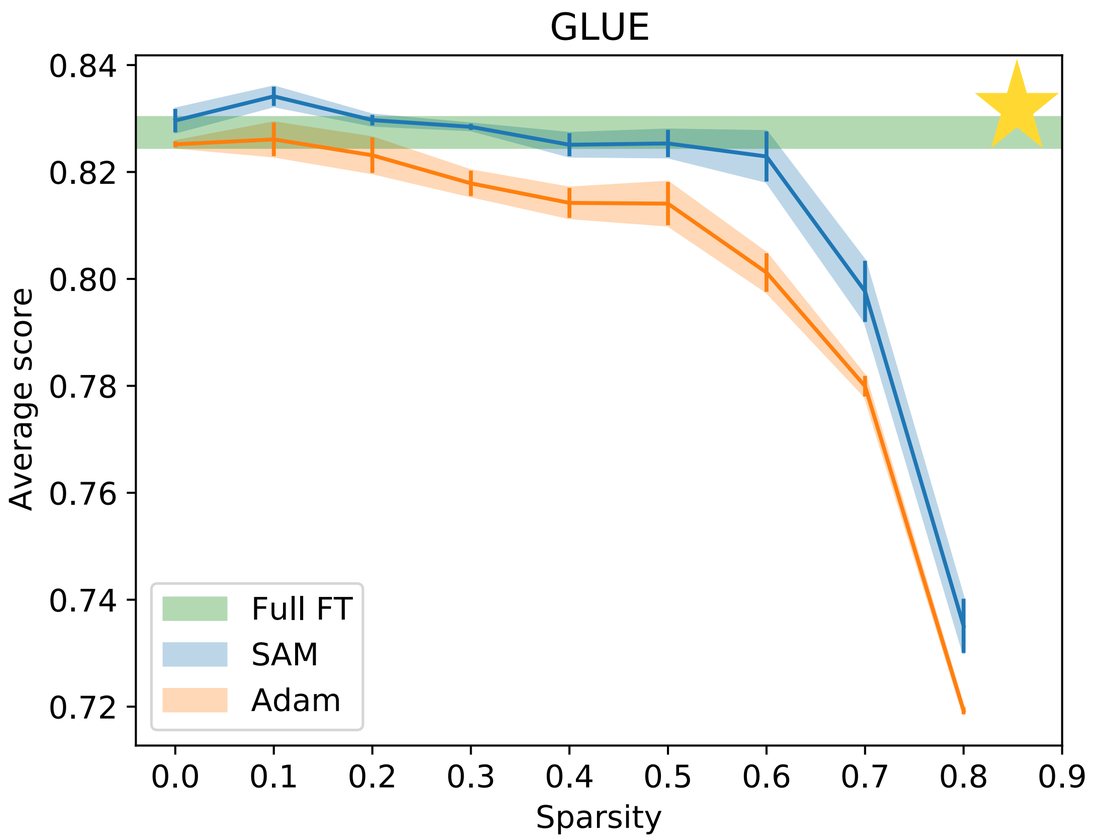
sharp **flat**



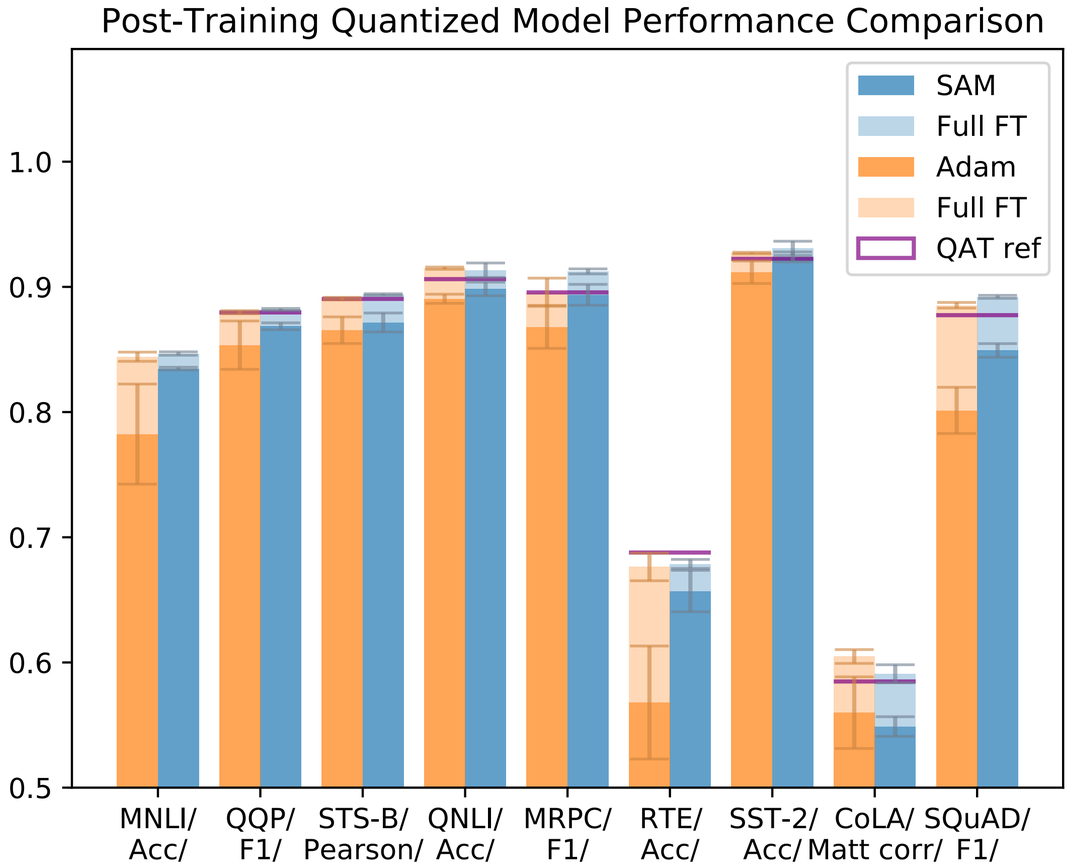
C. Na, S. V. Mehta, E. Strubell. Train Flat, Then Compress: Sharpness-Aware Minimization Learns More Compressible Models. Findings of EMNLP, 2022.

Models in flatter minima are more compressible

Model pruning



Model quantization



Does pruning work?

- Compression can destroy performance on a small subset of classes to preserve overall accuracy.
- Pruned networks are more sensitive to adversarial images and corruptions.
- All of the above is worse with higher levels of pruning.
- Pruning is worse than quantization.

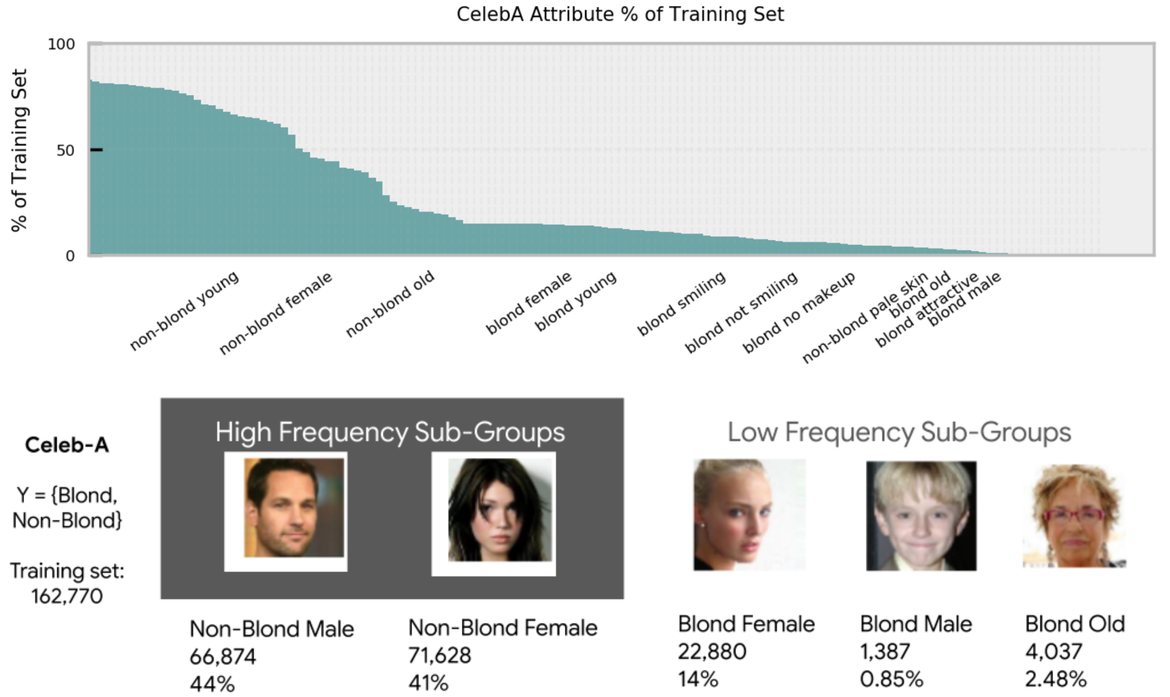
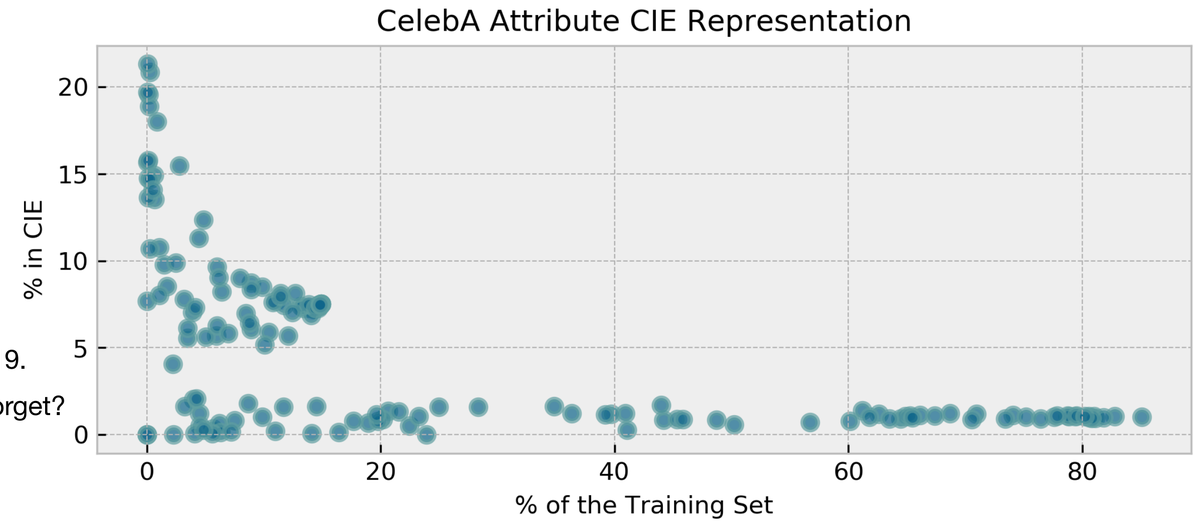


Figure 1: Most natural image datasets exhibit a long-tail distribution with an unequal frequency of attributes in the training data. Below each attribute sub-group in CelebA, we report the share of training set and total frequency count.



S. Hooker, A. Courville, G. Clark, Y. Dauphin, A. Frome. Characterizing Bias in Compressed Models. 2019.

S. Hooker, N. Moorosi, G. Clark, S. Bengio, E. Denton. What Do Compressed Deep Neural Networks Forget? ICML Workshop on Human Interpretability in Machine Learning. 2020

Knowledge distillation

- Method for **training** smaller models.
- Orig. idea: Train one model to predict outputs of ensemble.

Model Compression KDD 2006

Cristian Bucilă
Computer Science
Cornell University
cristi@cs.cornell.edu

Rich Caruana
Computer Science
Cornell University
caruana@cs.cornell.edu

Alexandru Niculescu-Mizil
Computer Science
Cornell University
alexn@cs.cornell.edu

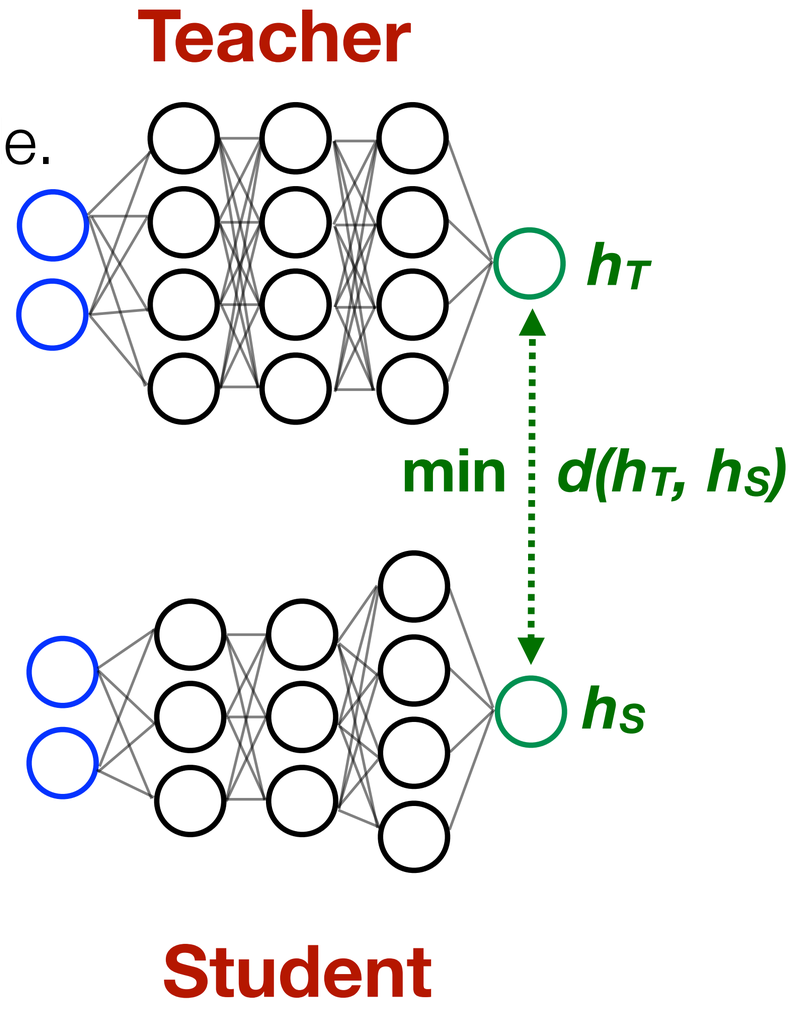
NeurIPS 2014 Workshop on Deep Learning

Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

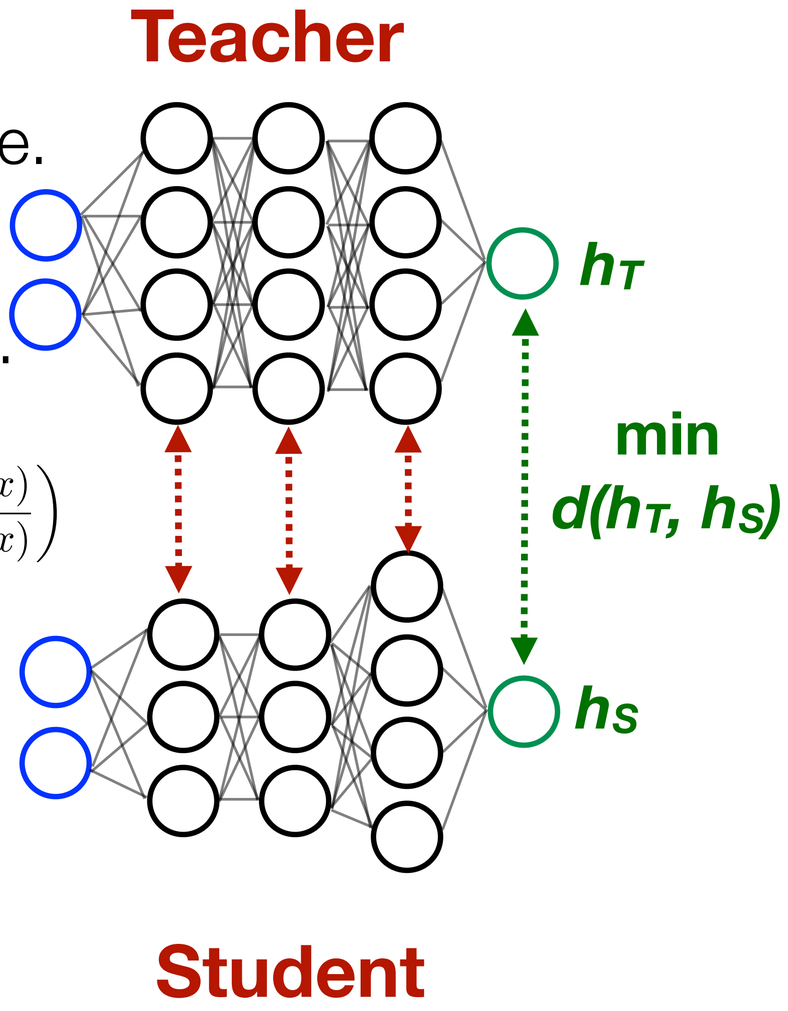
Jeff Dean
Google Inc.
Mountain View
jeff@google.com



Knowledge distillation

- Method for **training** smaller models.
- Orig. idea: Train one model to predict outputs of ensemble.
- **Key idea:**
  Train (smaller) model to produce outputs of (larger) model.
- Modeling decisions:
  - Hard or soft targets?
  - Where/how to mimic teacher?
  - How to optimize (distance metric ***d***)?
- Can be combined with any underlying architecture.

$$D_{\mathrm{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right)$$



Why does distillation work?

🤷‍♂️🤷‍♂️🤷‍♂️🤷‍♂️

Dense vs Sparse Labels

A cat is not a dog, but it's not not a dog. The representation should capture this

Understanding and Improving Knowledge Distillation (Tang et al. 2020)

Effects can be attributed to: 1) label smoothing 2) domain knowledge of class relationships 3) gradient rescaling based on difficulty.

Dataset Distillation (Wang et al. 2018)

The inverse problem — can the knowledge you learn from a dataset be "summarized" or distilled such that you can perform fast adaptation or few-shot learning from a key small set of examples.

Where does distillation work?

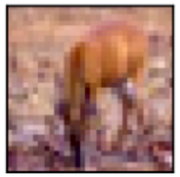
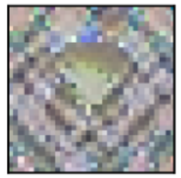
Does architecture matter?

Distillation has been performed with ~every architecture (CNNs, Transformers, …)

Are certain architectures better suited to distillation?

It's a reasonable hypothesis that model architecture expressivity may be related to applicability of distillation.

Dataset distillation

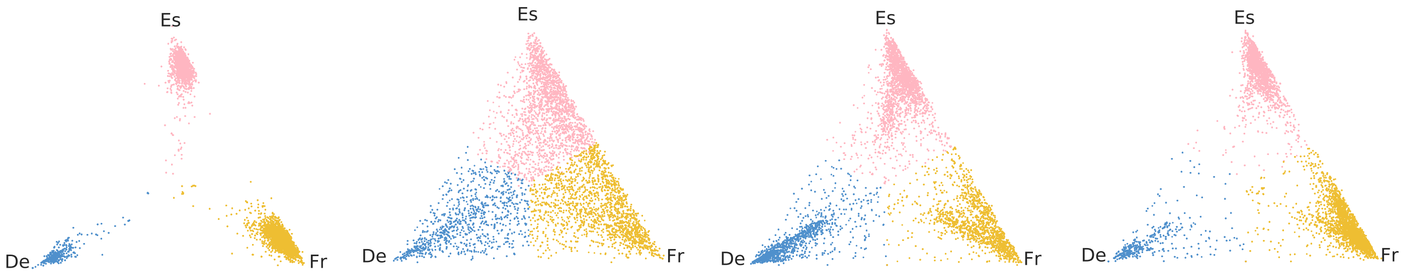
Entire training set (10 images)										Test accuracy
airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	
Images (init)											16.3%
Images (trained)											50.7%

https://blog.research.google/2021/12/training-machine-learning-models-more.html

Distillation Enables Novel Applications

Non-autoregressive Machine Translation

C. Zhou. J. Gu. G. Neubig. Understanding Knowledge Distillation in Non-autoregressive Machine Translation. ICLR 2020.



(a) AT Baseline (b) NAT Baseline (c) NAT Random Select (d) NAT Distill

Training CNNs with cheap convolutions and online distillation

Model	Convolution	Error %	#Params (M)	FLOPs (M)
ResNet-56 [26]	Standard	6.20	0.85	126.81
	Group-16	8.58	0.16	23.35
	Depthwise	9.55	0.13	20.14
	Shift	10.17	0.10	15.39

Beyond preserved accuracy: Loyalty & Robustness

- **Label loyalty:** How similar are the predicted labels compared to the full model?

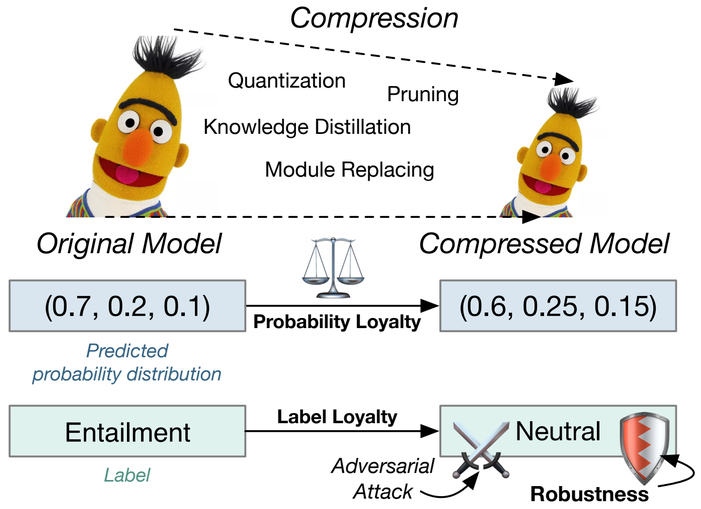
$$L_l = \text{Accuracy}(pred_t, pred_s)$$

- **Probability loyalty:** How similar are the compressed model's output probabilities compared to the full model?

$$D_{\text{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right) \qquad L_p(P\|Q) = 1 - \sqrt{D_{\text{JS}}(P\|Q)}$$

$$D_{\text{JS}}(P\|Q) = \frac{1}{2}D_{\text{KL}}(P\|M) + \frac{1}{2}D_{\text{KL}}(Q\|M)$$

- **Robustness:** How robust is the compressed model against an adversarial attack, compared to the full model?



Method	Speed	MNLI	L-L	P-L	AA	# Q
Teacher	1.0×	84.5 / 83.3	100	100	8.1	89.6
Head Prune	1.2×	80.9 / 80.6	87.8	85.5	9.1	90.5
+Finetune	1.2×	83.2 / 81.9	89.1	85.5	7.2	83.2
+KD	1.2×	**84.2 / 83.0**	**93.3**	**93.0**	8.3	90.5
+KD+PTQ	2.2×	80.8 / 80.4	89.6	86.3	**38.4**	**90.9**
Q8-QAT	1.8×	83.4 / 82.4	89.7	88.2	6.8	82.7
Q8-PTQ	1.8×	80.7 / 80.4	89.6	80.8	**40.2**	**91.6**
+Finetune	1.8×	82.9 / 81.9	89.7	84.8	7.1	84.5
+KD	1.8×	**84.1 / 83.5**	**94.0**	**93.9**	7.5	86.1
BERT-PKD	2.0×	81.3 / 81.1	88.9	89.0	6.4	81.9
Theseus	2.0×	81.8 / 80.7	88.1	82.5	8.3	89.7
+KD	2.0×	**82.6 / 81.7**	**91.2**	**91.4**	8.0	88.7
+KD+PTQ	3.6×	80.2 / 79.9	89.5	80.3	**36.5**	**91.3**

Xu, Zhou, Ge, Xu, McAuley, Wei. Beyond Preserved Accuracy: Evaluating Loyalty and Robustness of BERT Compression. EMNLP 2021.

Adaptive computation

- Original idea: Deploy smaller, simpler models for easier examples, larger, more complex models for difficult examples.

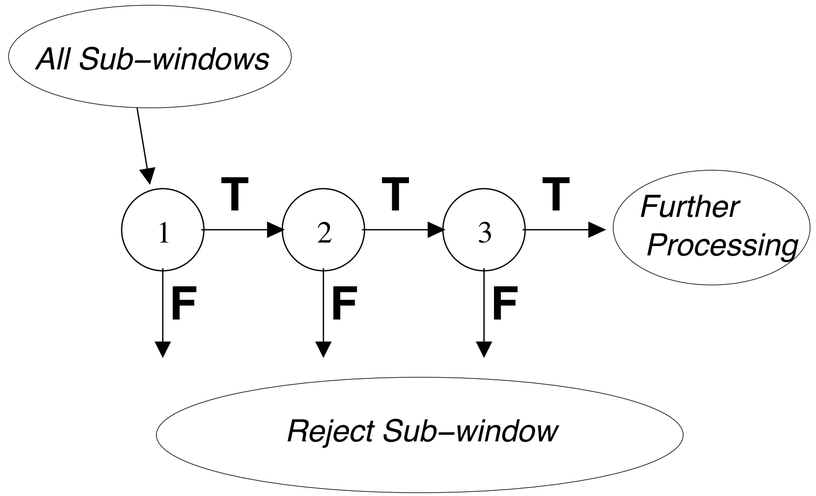
ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

CVPR 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

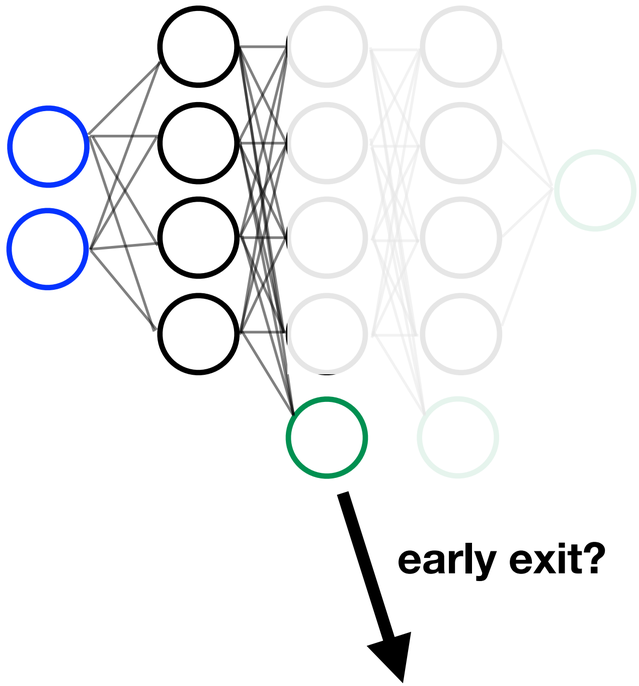
Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

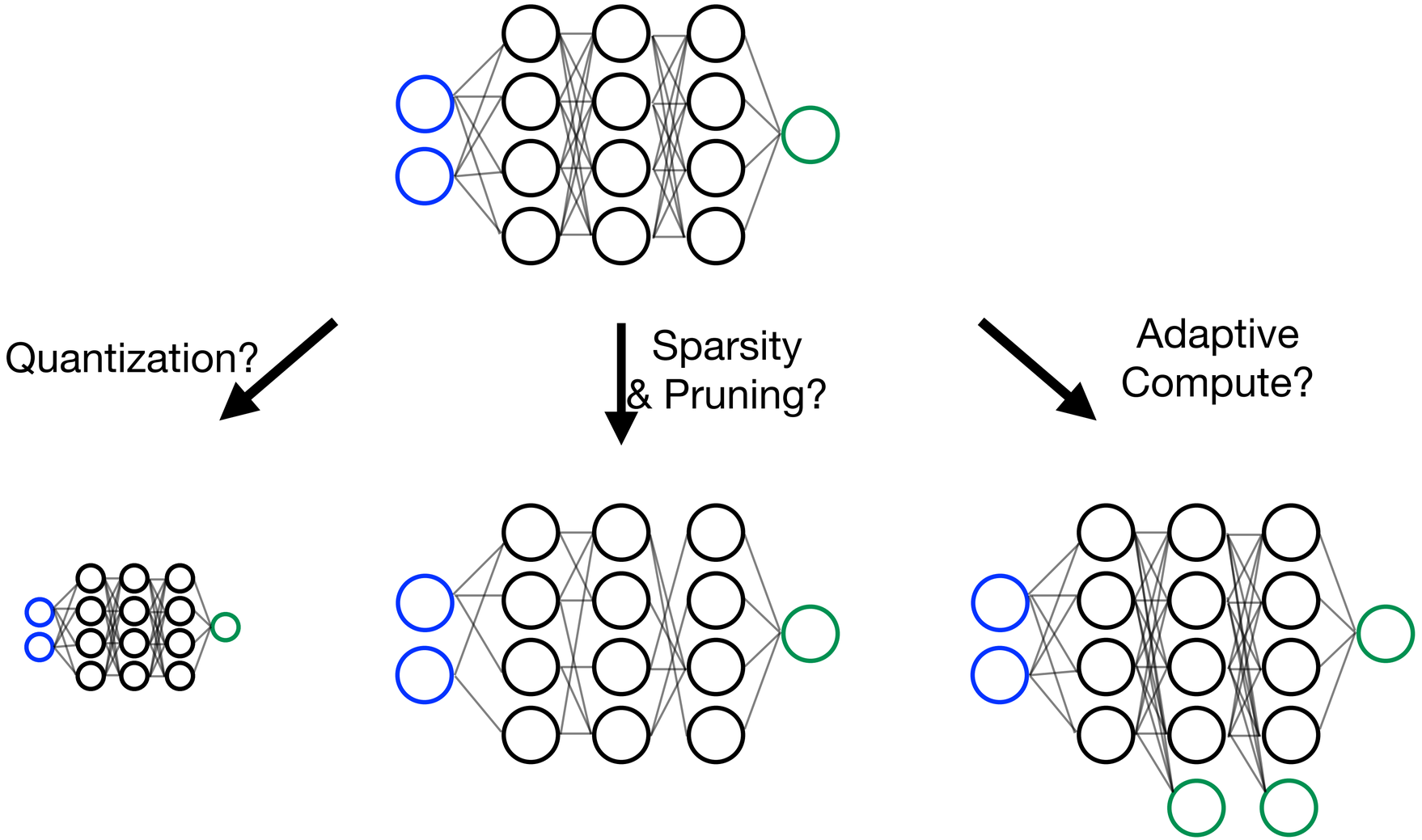


Adaptive computation

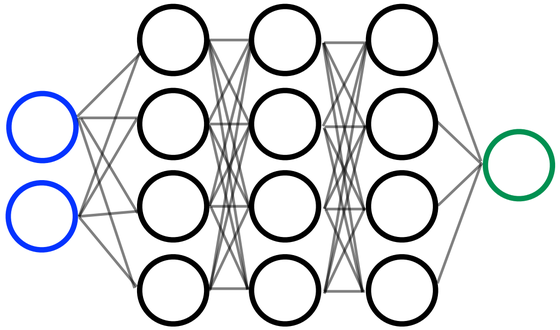
- Orig. idea: Deploy smaller, simpler models for easier examples, larger, more complex models for difficult examples.
- **Key idea:** Use more or fewer parameters depending on example difficulty.
- Ideally, parameters are shared (~self-distillation).
- Modeling decisions / challenges:
  - How to (efficiently) compute **confidence** / early stopping criterion?
  - **Batching** to maximize compute throughput on accelerators?



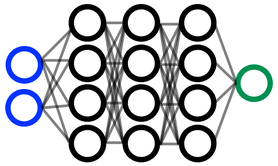
Model compression: Overview



Model compression: How to choose?

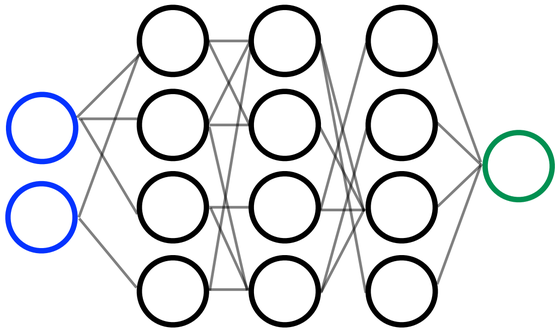


Quantization



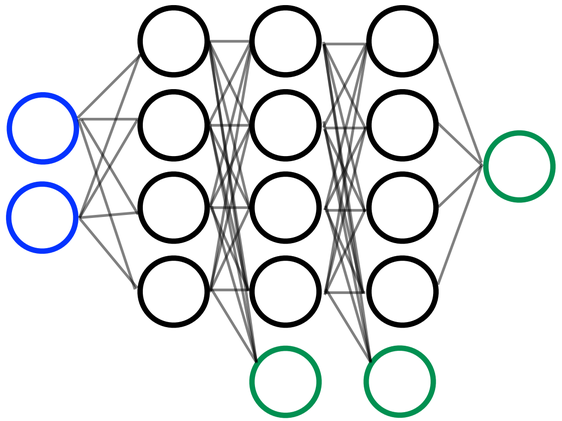
Pro: Fewer Bits/Memory
Con: Requires hardware support for efficiency, otherwise you're pretending which is probably slower.

Sparsity & Pruning



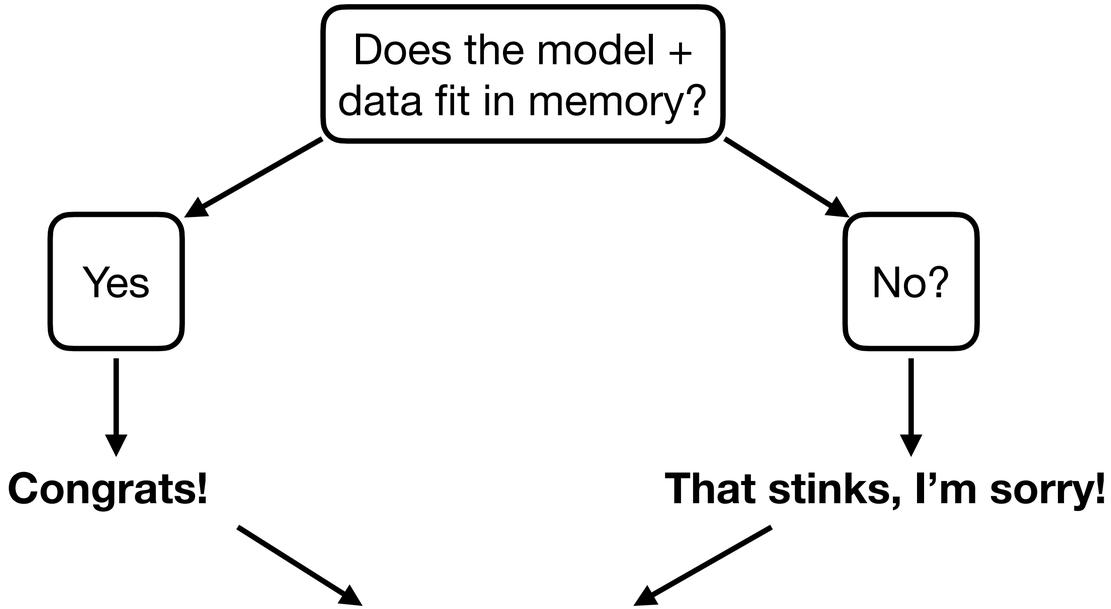
Pro: Substantially fewer parameters and operations
Con: Requires hardware support for Sparse Ops

Adaptive Compute



Pro: Early exit, process only if it changes the outcome
Con: Requires additional control flow logic. Better suited to CPU than GPU

Model compression: Is hardware the answer?



Now go worry about:
- Power consumption
- Wall clock
- Vector ops
- Competition with OS
- Model accuracy
- Other platforms …

CPUs:
- Designed for control flow
- Vector ops (very limited)
- Small cache, reliant on system RAM

GPUs:
- No control flow
- Large matrix multiplies
- Mixed precision support
- Large RAM
- Recent! devices have sparse ops

TPUs:
- No control flow
- Large matrix multiplies
- Large RAM
- Native mixed & low precision

FPGAs:
- Whatever you want boss