Sparsity in Deep Learning

Abhinav Bhatele, Daniel Nichols

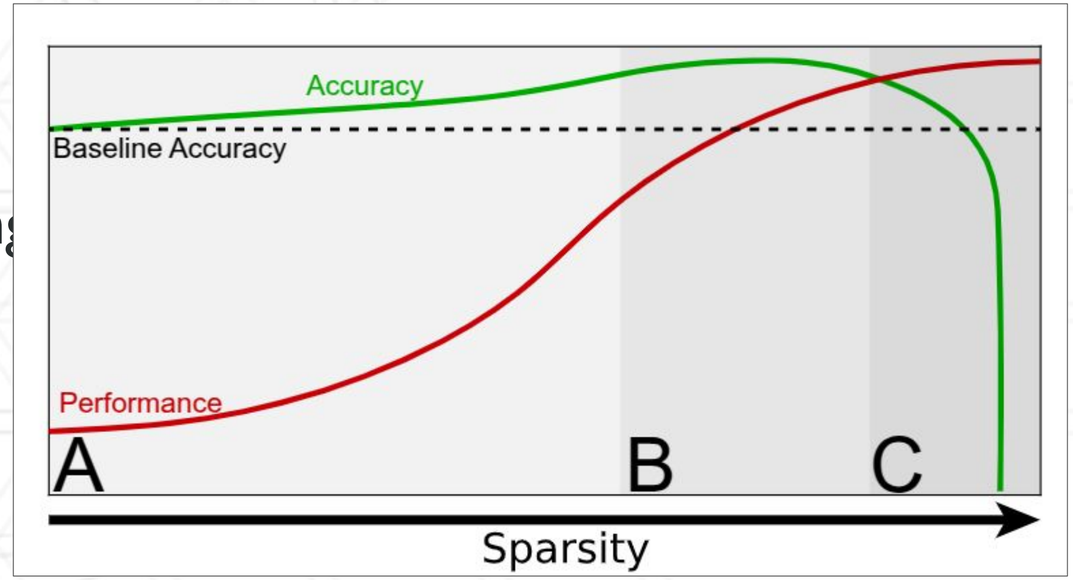
UNIVERSITY OF MARYLAND

Announcements

- Interim report for the project is due on April 17
- Midterm is on April 10

Why do we need sparsity?

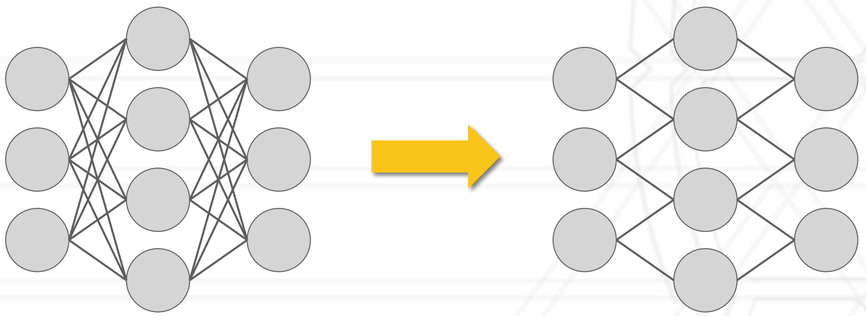
- Less parameters can mean less computation and memory
- A lot of parameters are not needed
  - Denil, et. al. "Predicting Parameters in Deep Learning" – 95% of parameters could be predicted from 5% in ConvNet
  - Many parameters are redundant
- Too many parameters can lead to overfittin
- Too few can lead to a loss an accuracy



T. Hoefler et. al. "Sparsity in Deep Learning"

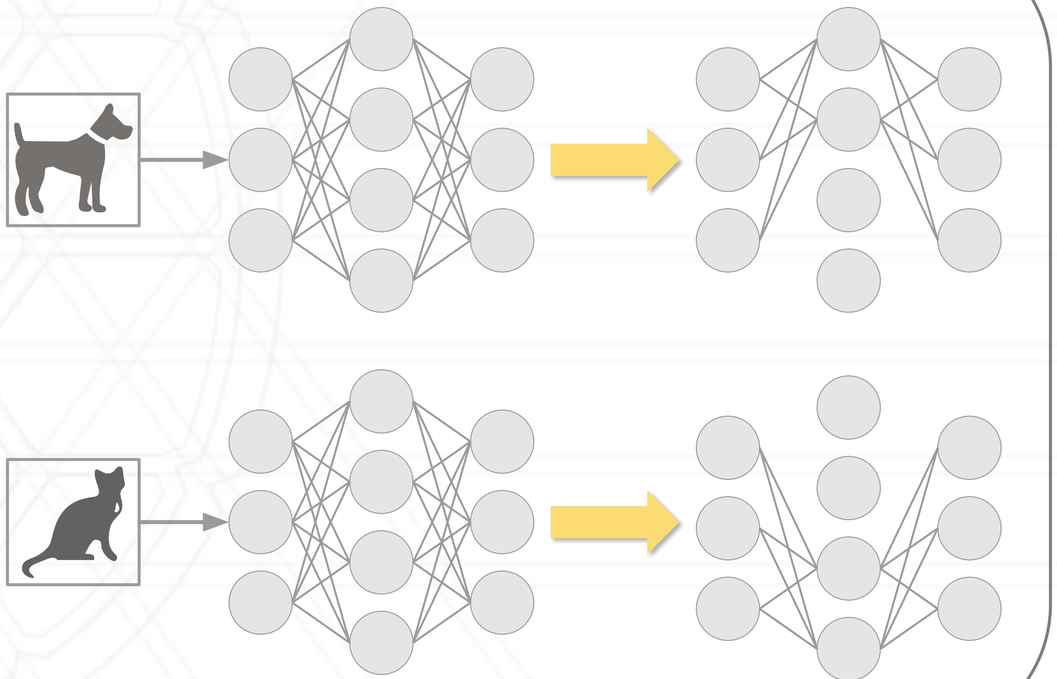
Types of Sparsity

Model/Structural



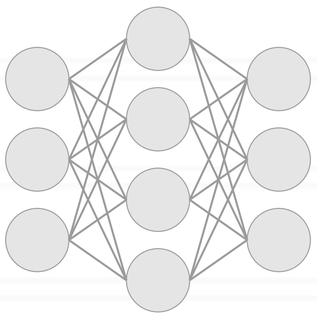
Architectural change; sparsity is independent of data during training/inference

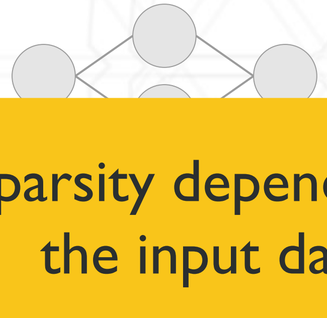
Ephemeral



Types of Sparsity

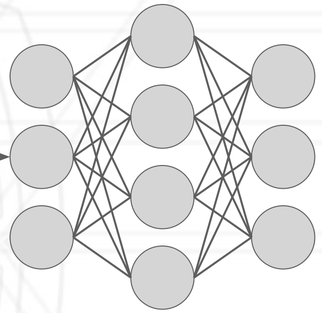
Model/Structural

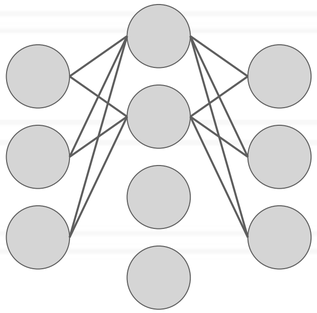


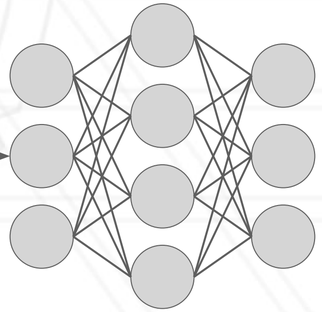


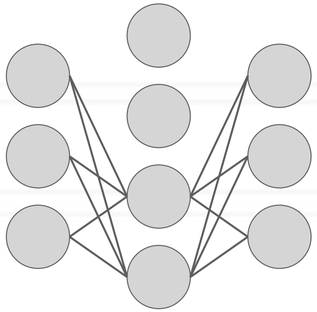
Sparsity depends on the input data

Ephemeral









Structural Sparsity

Model/Structural

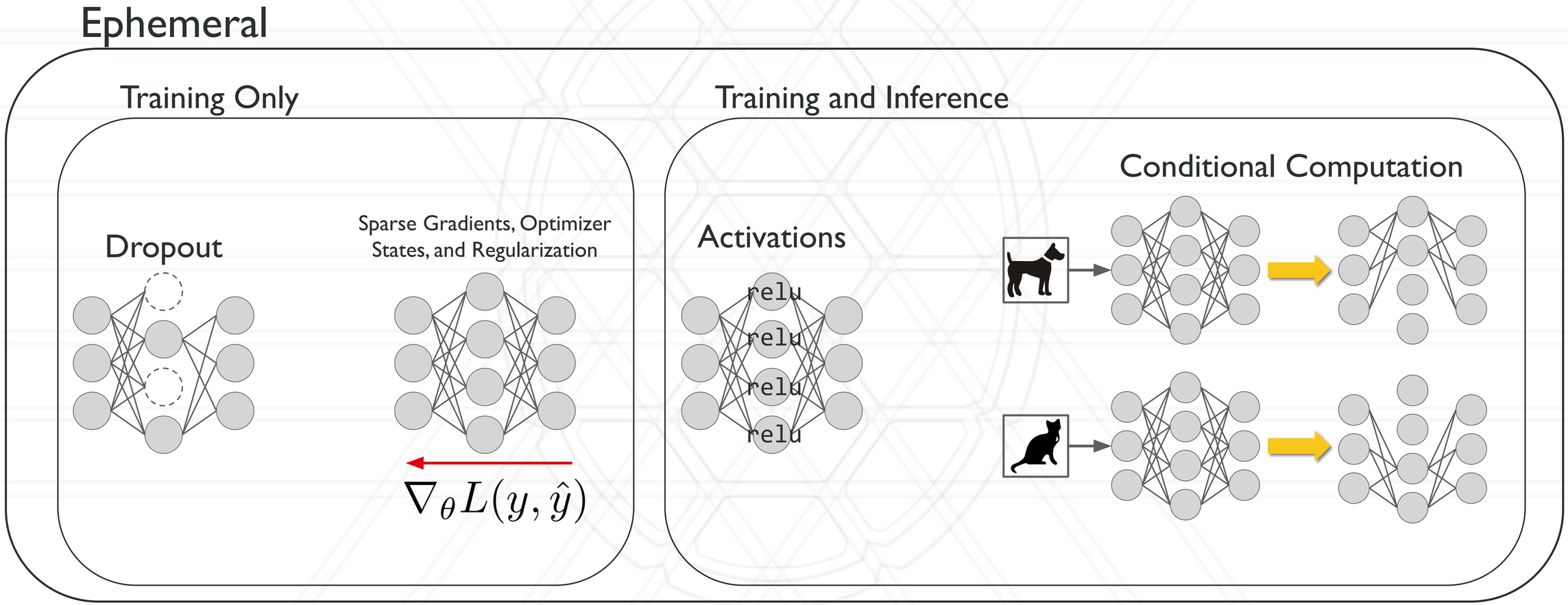
Weight Sparsity

Activation Sparsity

Pruning

- How do we know what weights/activations to remove?
- Data-Free Pruning
  - magnitude
  - merge-and-scale (combine activations with similar corresponding weights)
- Data-Dependent Pruning
  - “Trivial activations” – remove activations near zero for most data points
  - Output Sensitivity
  - Merge-and-scale
- Loss / Gradient Dependent Pruning
  - L_0 regularization
  - Remove weights with little changes

Ephemeral Sparsity

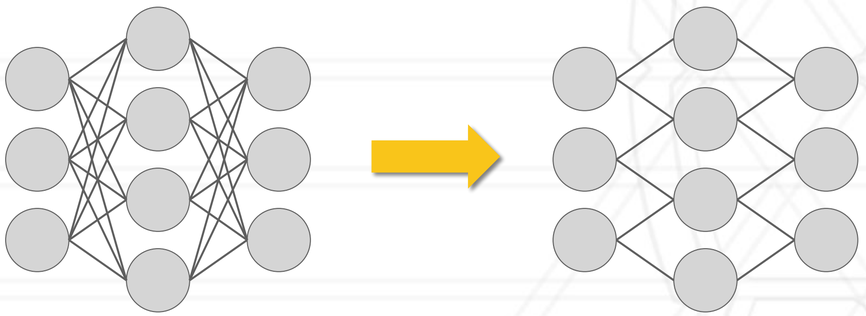


When to sparsify?

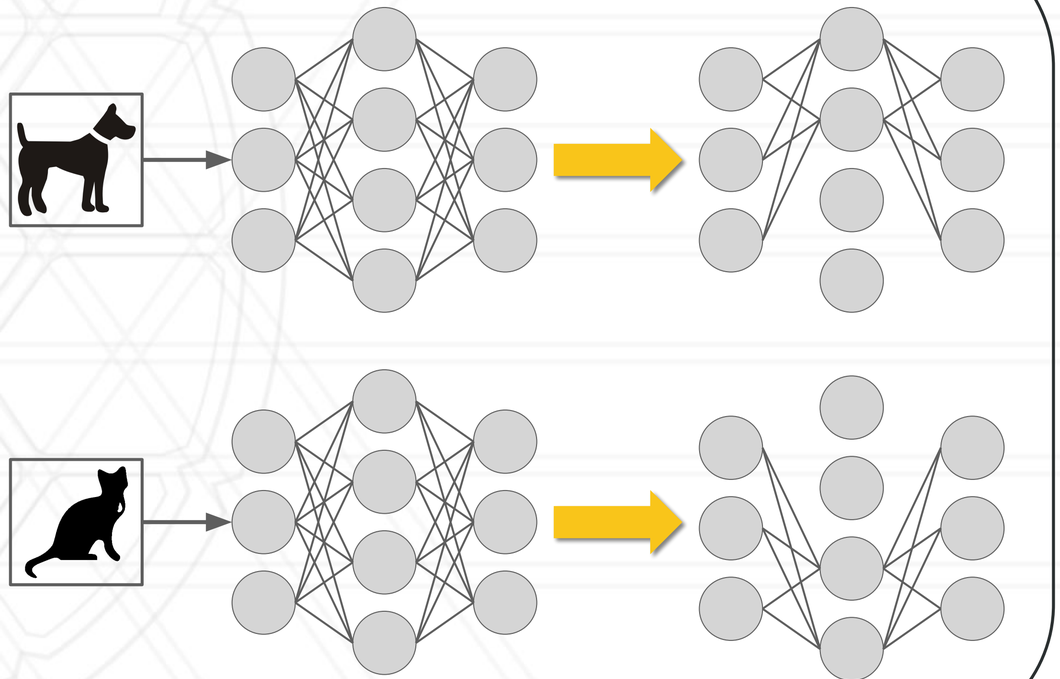
- Before training
-

Types of Sparsity

Model/Structural



Ephemeral



UNIVERSITY OF
MARYLAND