Stability Techniques in Differentially Private Machine Learning

By: Coleman DuPlessie, Aidan Gao
Mentored by: Hanshen Xiao

Background

01 What is machine learning?

02 What is CIFAR-10?

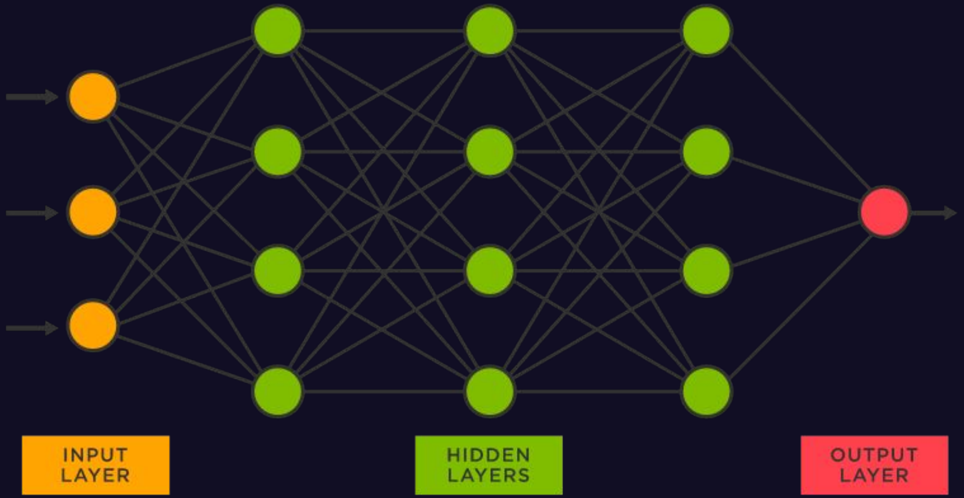
03 What is differential privacy?

04 How is differential privacy currently used in machine learning?

05 How is our method different?

What is Machine Learning?

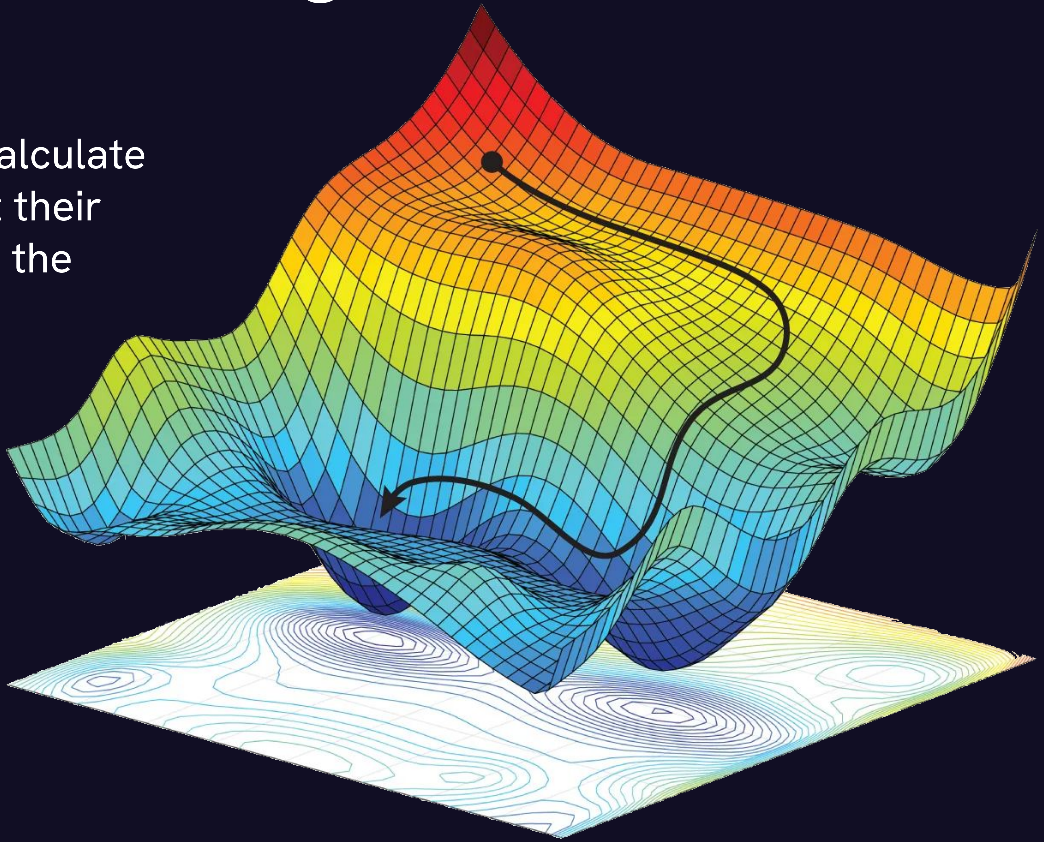
Machine Learning models learn by being *shown* instead of *told*



Left - a representation of a machine learning model with 3 inputs and 1 output. Each node has a value, which is calculated by taking a weighted average of the nodes in the previous layer.

What is Machine Learning?

Machine learning models calculate the slope of the gradient at their current point, then move in the direction with the greatest accuracy improvement

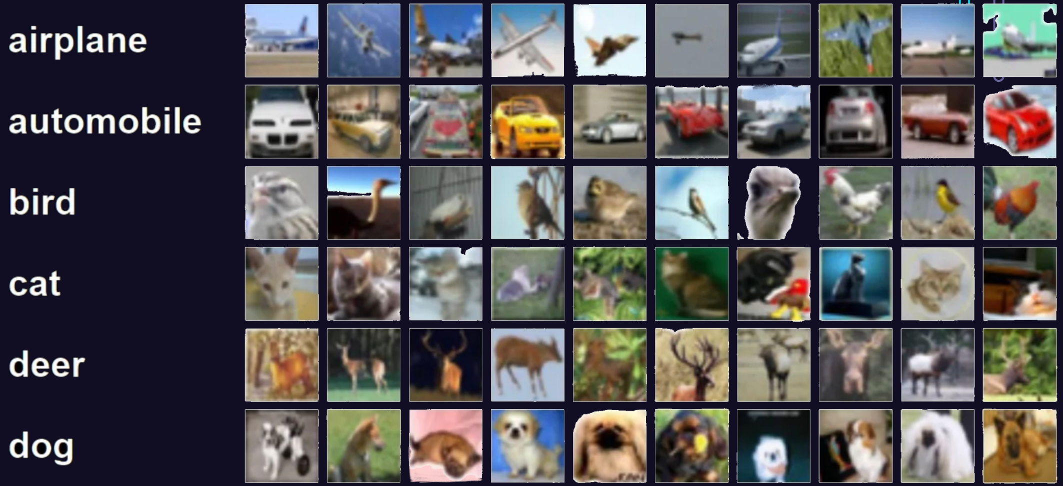


What is CIFAR-10?

A common machine learning benchmark that asks models to sort images into 10 categories

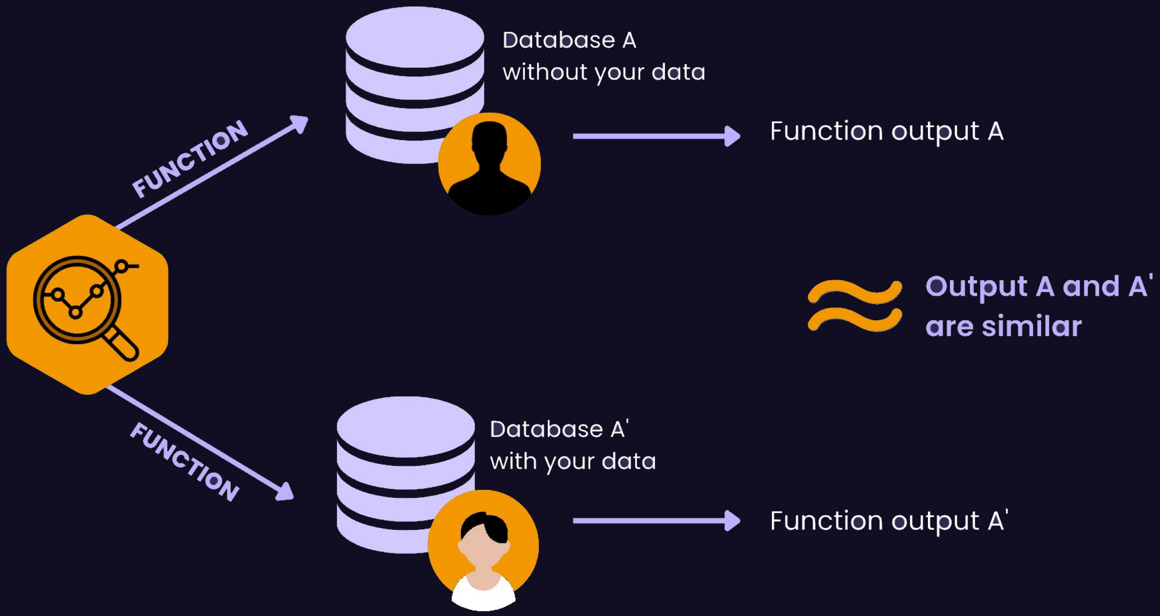
Advantages:

- Classification is a well-known, common task for ML models.
- Low-resolution images mean models aren’t too big.
- Large body of prior research to reference and benchmark against.



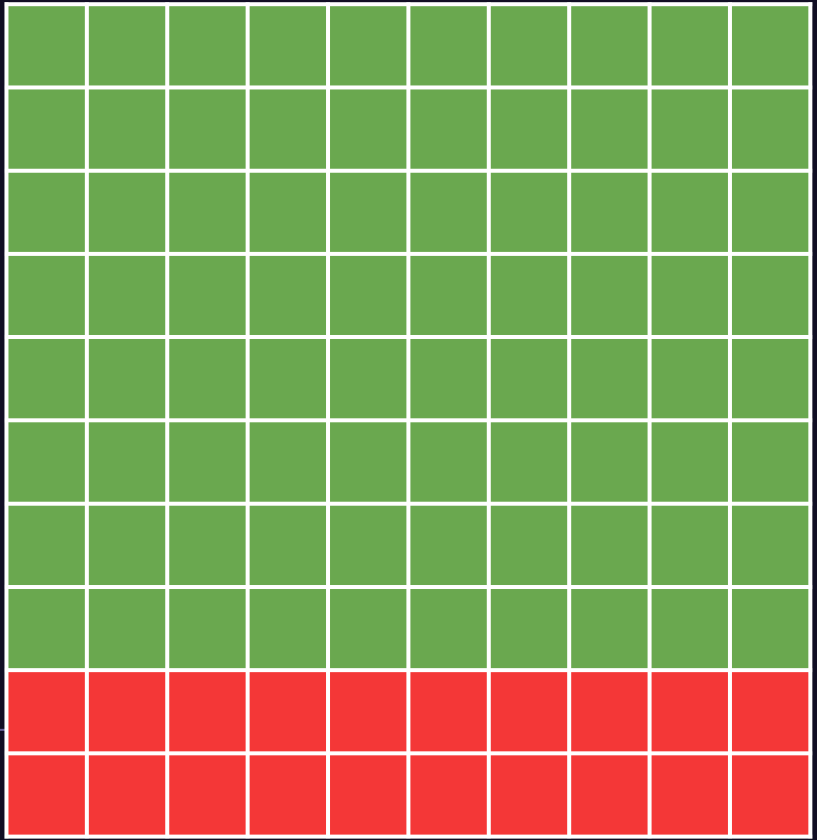
Some images from CIFAR-10

What is Differential Privacy?



Differential privacy guarantees that a small change in input (e.g. one less datapoint) will only lead to a bounded change in output.

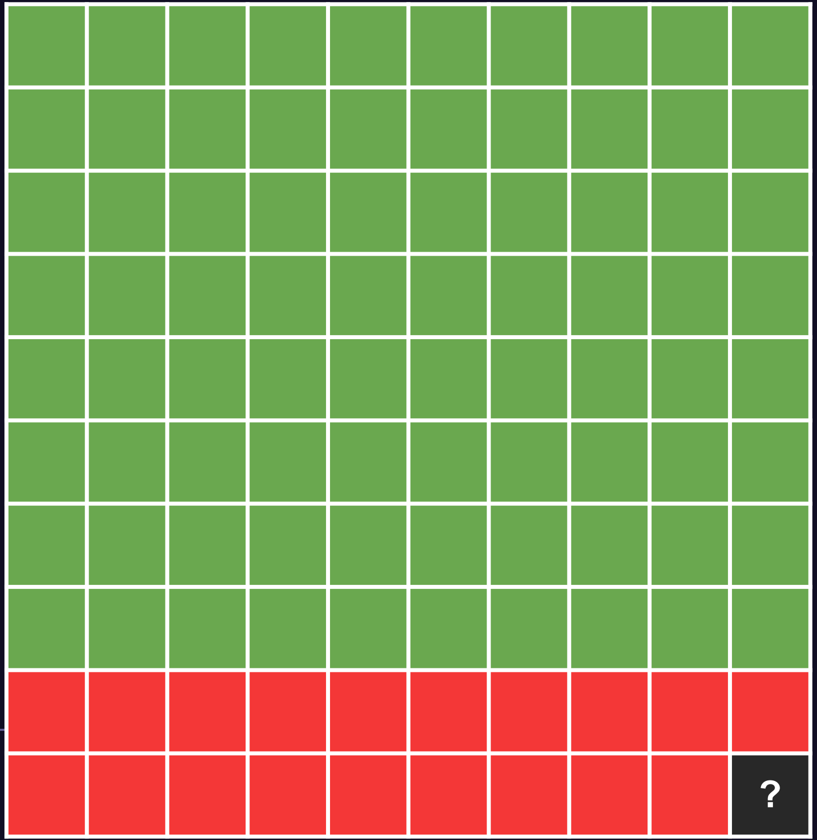
Differential Privacy



20
cheaters

100 students

Anonymization Falls Short

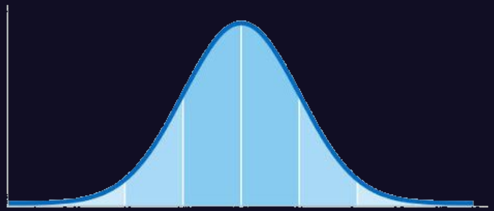


If you know the summary statistic and the other 99 students, you can figure out whether or not I cheated easily

Differential Privacy Works

~21
cheaters

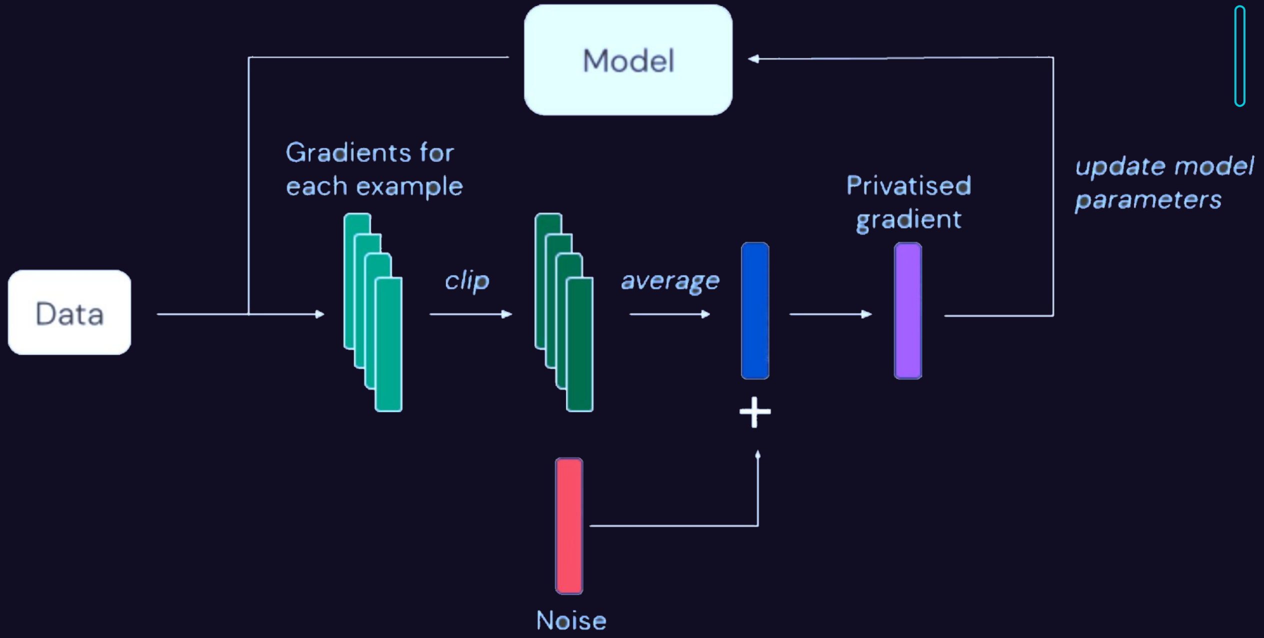
Even if you know about everyone else, you can't know (or even have a good guess) whether the difference was caused by my response or just random noise

20 +  = ~21

Real statistic + random noise = published "statistic"

Differential Privacy in Machine Learning

Differentially
Private
Stochastic
Gradient
Descent

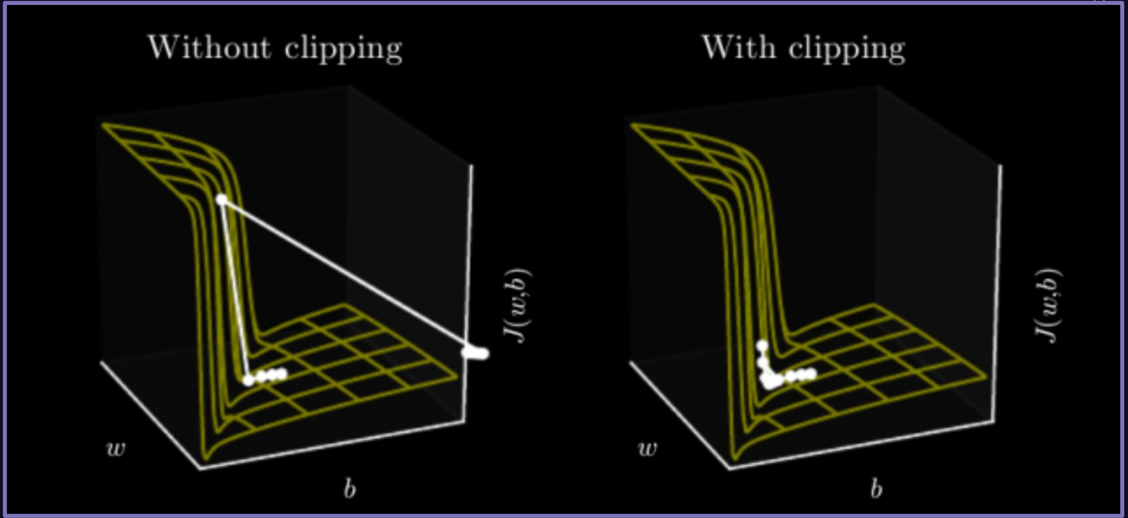


Differential Privacy in Machine Learning

Accuracy loss from:

- Adding random noise
- Gradient clipping

More clipping ➡ less noise needed



A New Method

Adding noise to the model after training!

Advantages

- Avoids gradient clipping entirely
- Allows us to add noise once at the end of training, instead of every batch

Disadvantages

- Relatively new, little prior work done on optimization
- No easy way to create stability guarantee, empirical estimations don't create privacy guarantees

Methods

Constants

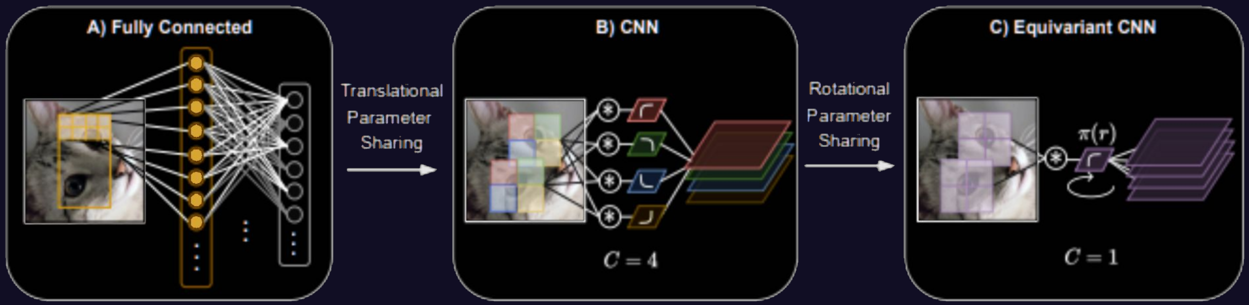
- Model selection
- Stability calculation

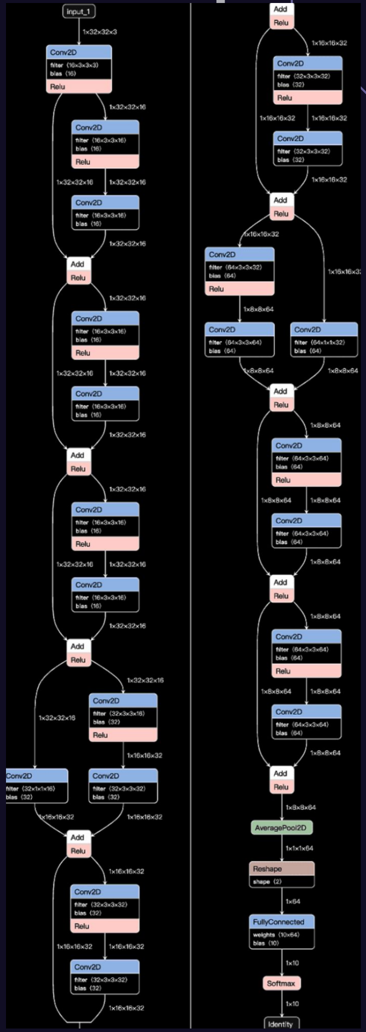
Experiments

- Full-Batch gradient and Pretraining
- Layer Freezing
- Pruning and Gradient Clipping
- Tree-net
- Linear regressions for post-training privatization

▶ Model selection

- ECNN
- VGG 19
- Resnet 20





Empirical Model Stability Calculation

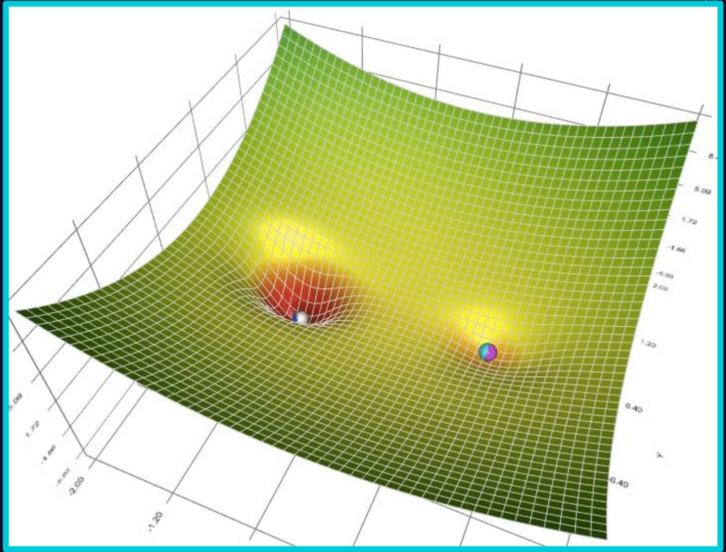
Two Main Methods

- L2 norm
- Square root sum of eigenvalues

- Models are trials

Full-Batch Gradient and Pretraining

- SGD vs. Full gradient: What's the difference?
- Pretraining to eliminate randomness
- Techniques here establish baseline



Results For Baseline

- For small CNN, l2 norm of deviation is 0.007 for baseline
- We test with a "public" and "private" simulation, using a 5k pretrain and 25k samples
- We use this to test full batch gradient vs. SGD for the resnet20
- We aim to find balance between stability and accuracy

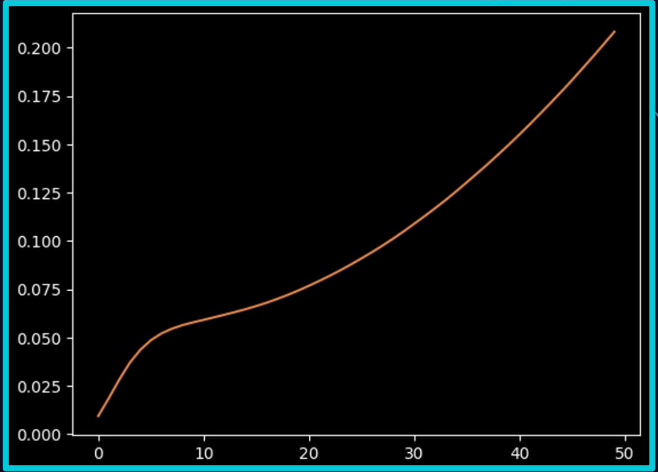


Fig 1: divergence l2 norm over 50 epochs for full batch

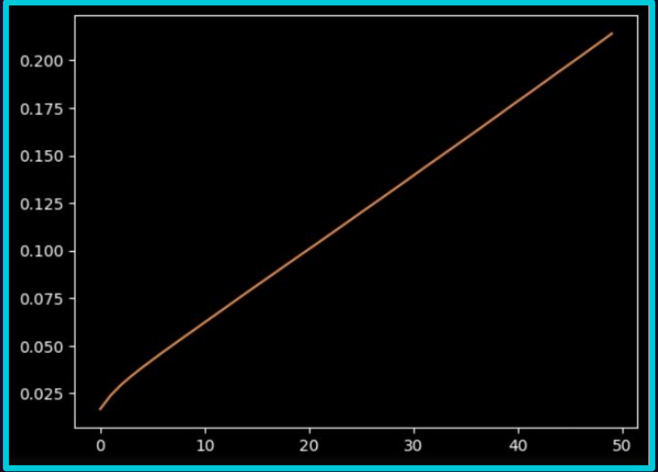
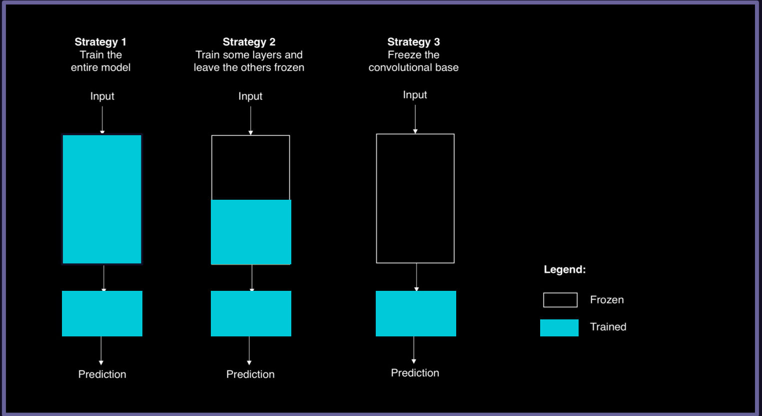


Fig 2: divergence l2 norm over 50 epochs for small batch

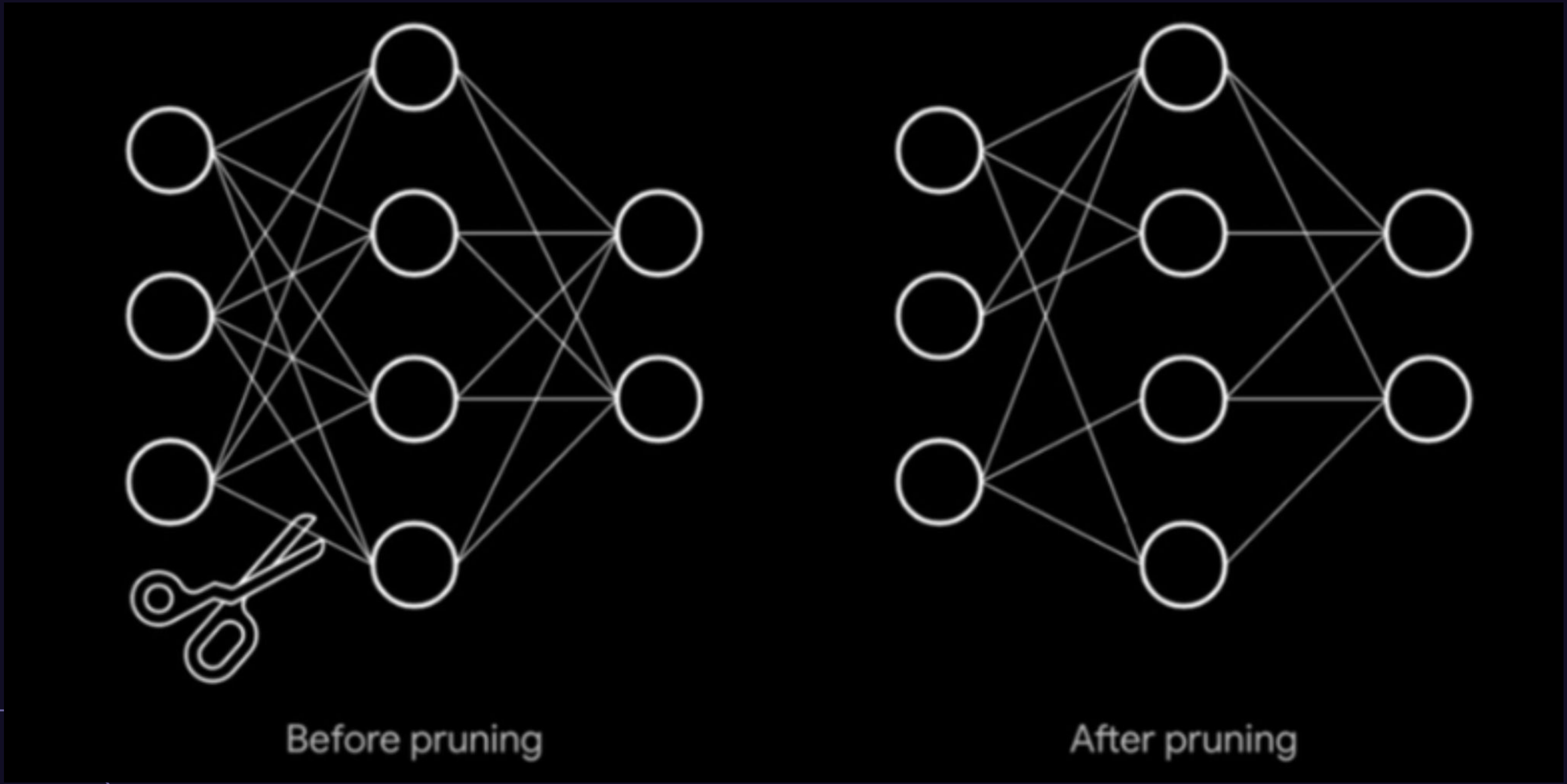
Layer Freezing

- Extension of Baseline, how is accuracy/stability affected if only last layers are trained
- How many layers do we need to freeze?
- Results oriented around noise addition and resistance to addition

- Deviation from pretrained model: 0.4%
- Accuracy loss from noise: 0.3%
- Accuracy increase: 1-2%



Pruning



Pruning and Gradient Clipping

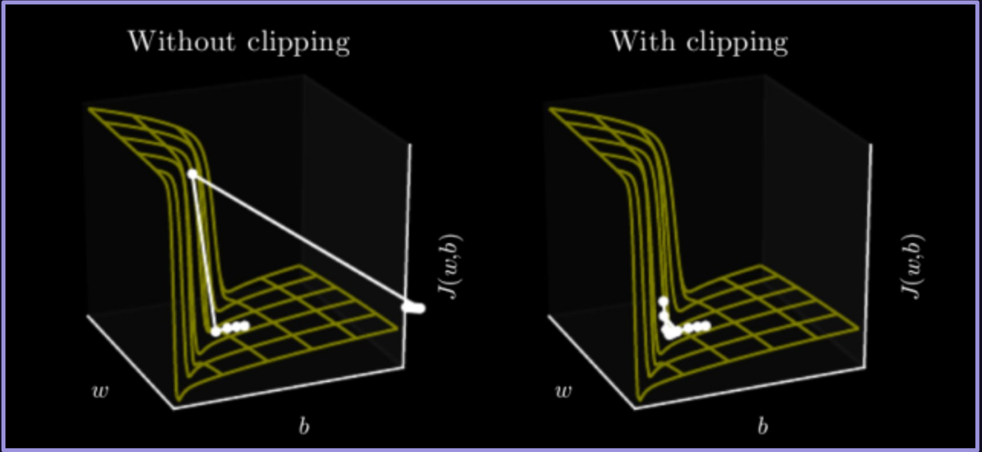
- Pruning nodes with low l1 norm reduces clutter
- Gradient clipping prevents large gradients from creating largely different models
- Combine for most important features with no extremes

Pruning the pretrained model:

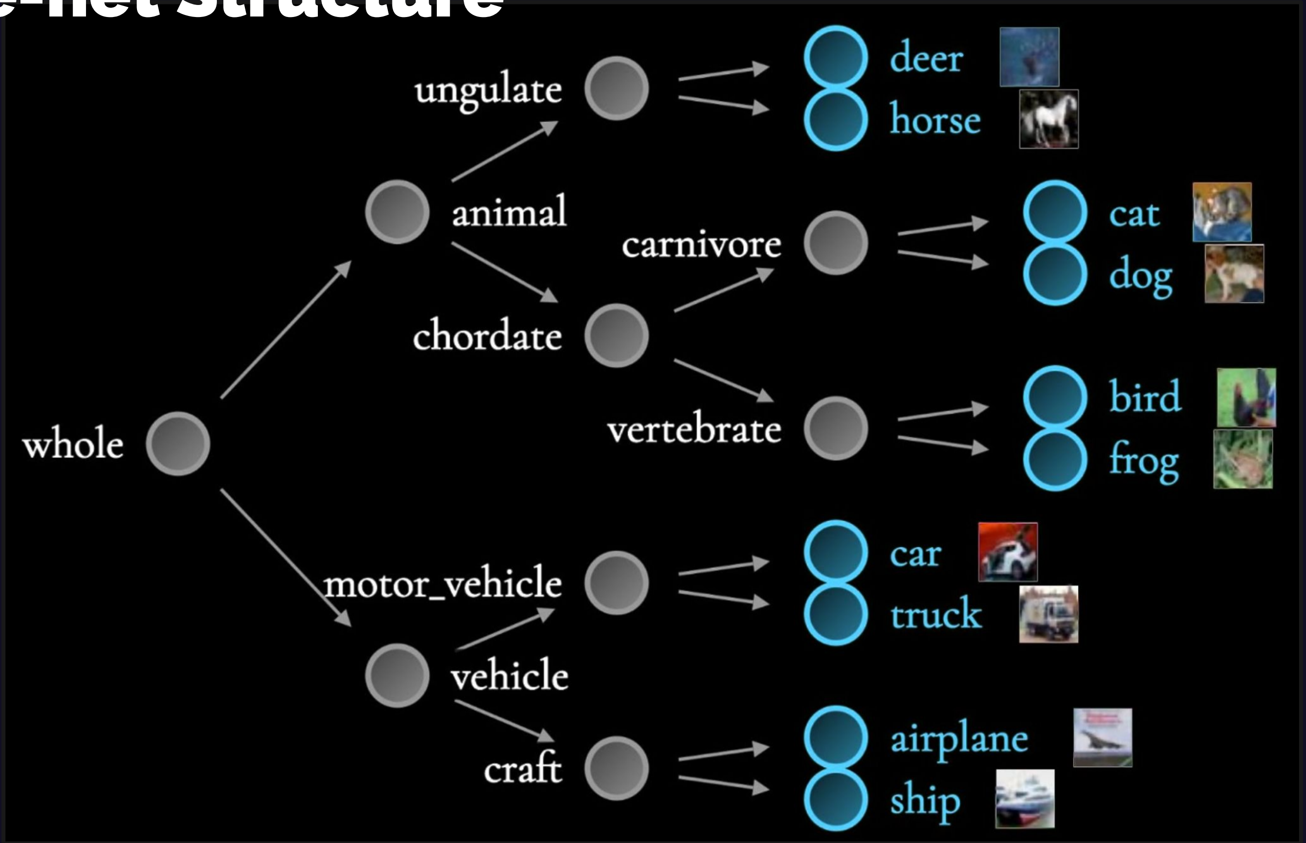
- L2 norm deviation: 55%
- Accuracy: 15% increase

Clipping:

- L2 norm deviation: 20%
- Accuracy: 13% increase



▶ Tree-net Structure



Tree-net Results

- Three models compared: resnet20, one layer tree-net, full tree-net
- Similar accuracies of ~70%
- Full tree-net more resistant to noise

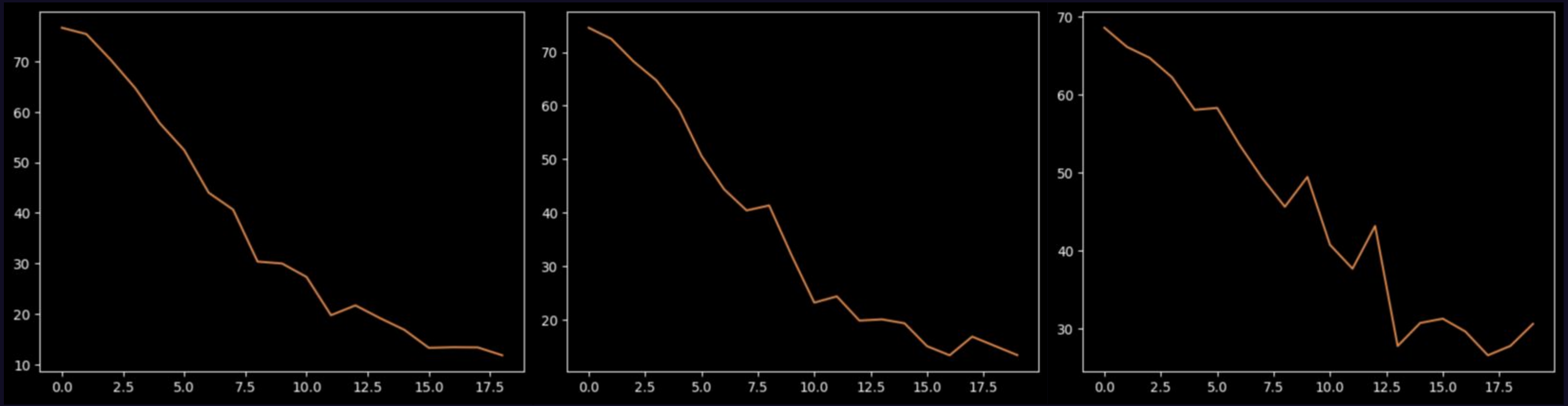


Fig 3:Accuracy graphs of Resnet20, One layer, and Tree-net

Linear regressions (Ongoing)

Linear regressions function as a proof of concept: simpler models that are easier to create and run tests on.

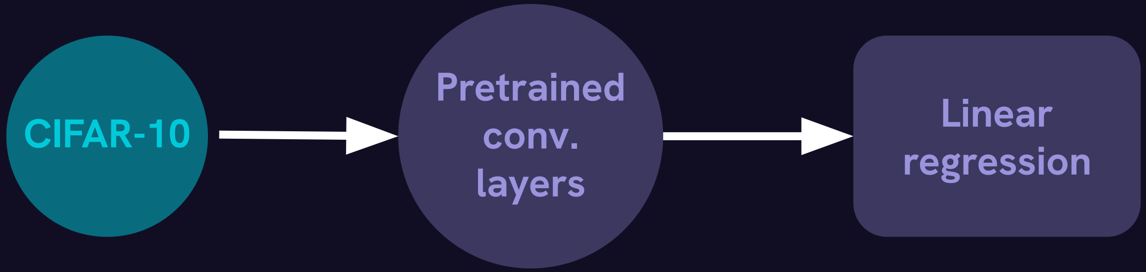
Two different (but similar) types:

- A single-layer neural network, trained with SGD.
- A linear regression fit with least-squares to the whole train set.

Trained on a small fraction of the 50,000 sample train set several times, then compared to itself to empirically measure stability.

Linear regressions (Ongoing)

We found that data preprocessing increases both accuracy and stability in linear regressions.



Acknowledgements

- Thank you to PRIMES for giving us this opportunity
- Thank you to our parents for supporting us
- Thank you to our mentor, Hanshen Xiao, for guiding and helping us during our research

Any Questions?