Carnegie Mellon University

18-649 Guest Lecture:
Machine Learning

Tianshu Huang — CMU WiseLab | LIONS

(1) When should I use ML / DL?
(2) How I develop and deploy ML to the edge?

This lecture brought to you by:

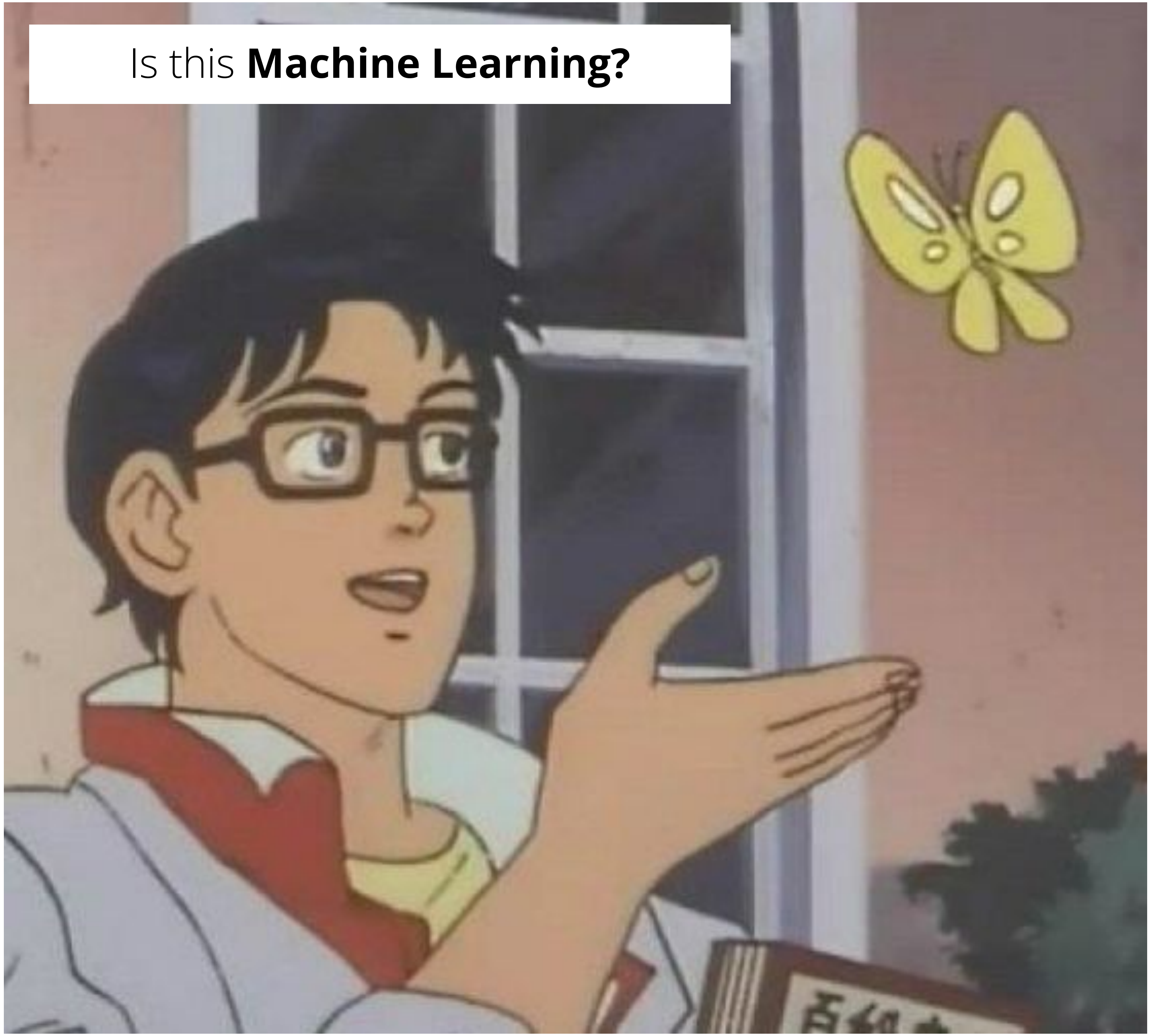
Learning for Systems

Have some background in Machine Learning and/or systems, and interested in **using Machine Learning to understand and improve systems?**

- Understanding program resource usage
- Edge orchestration
- Handling network dropout
- And more!

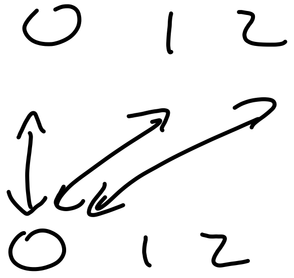
Contact: tianshu2@andrew.cmu.edu

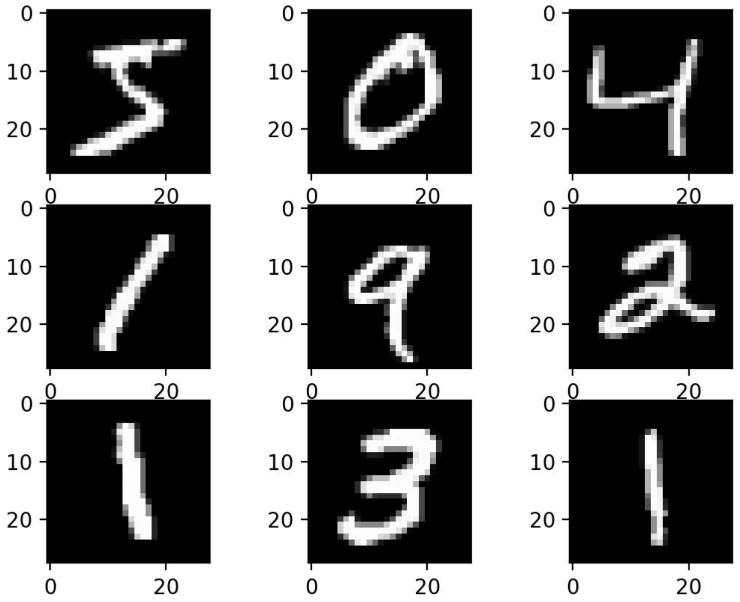
Machine Learning



Machine Learning

Example: Digit Classification





No Data

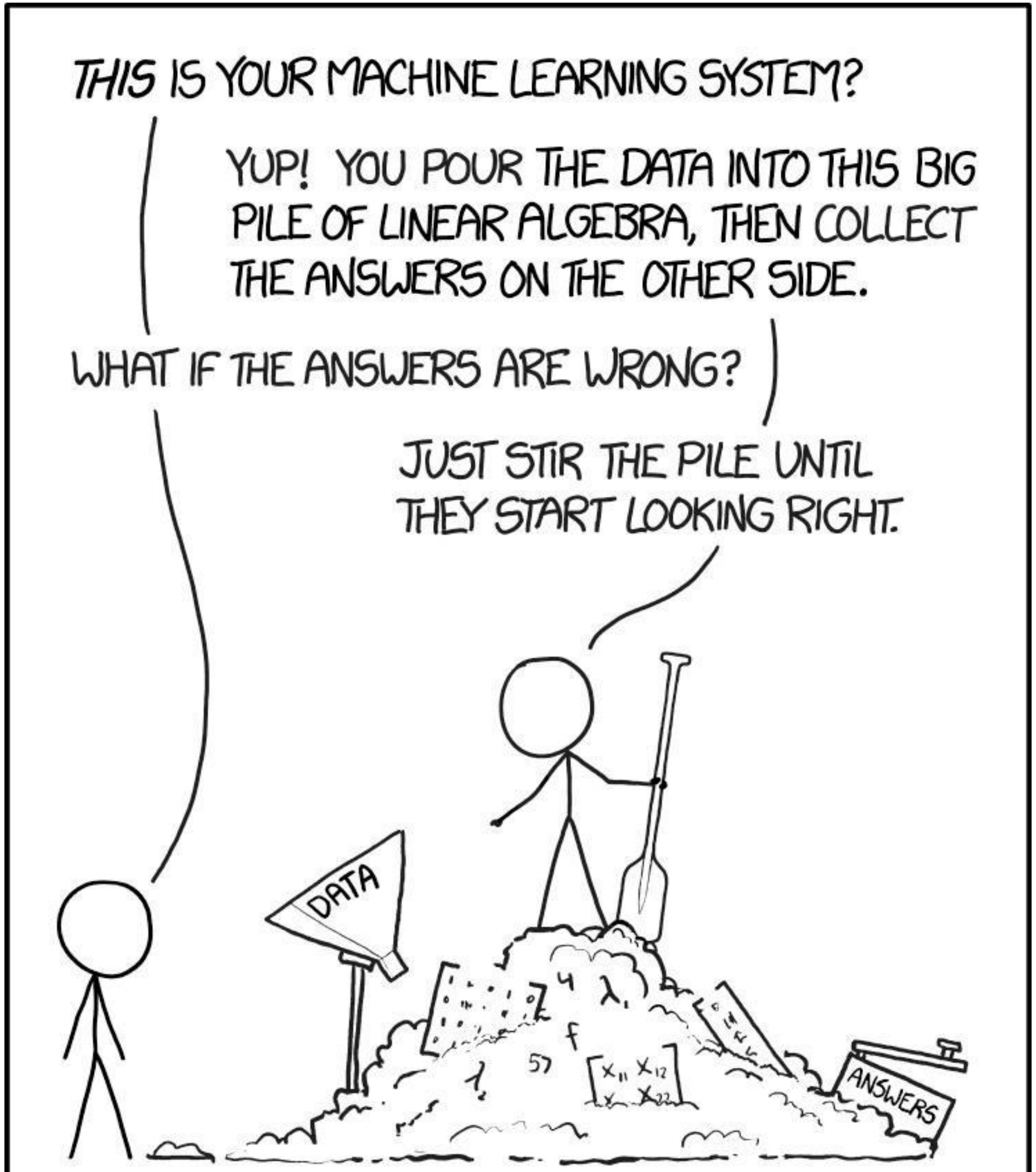
- Color distribution
- pixel neighborhood
- Hough transform

Some Data

- Compare images
- feature extraction + examples

Machine Learning

- NN
- Conv NN
- Transformers
- Perceptrons
- Naive Bayes
- Linear Regression
- PCA
- XGBoost
- Random Forest
- Decision Trees
- SVM



https://xkcd.com/1838/

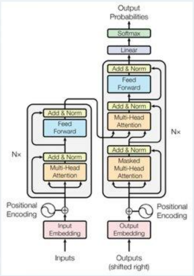
Deep Learning

STOP DOING DEEP LEARNING

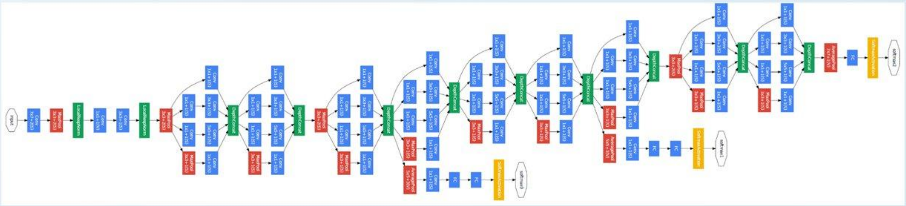
- PERCEPTRONS WERE ONLY EVER MEANT TO BE FULLY CONNECTED
- THOUSANDS OF PAPERS yet NO REAL-WORLD USE FOUND for going deeper than ONE LAYER
- Wanted to add more nonlinearity anyway for a laugh? We had a tool for that: It was called “KERNEL METHODS”
- “Yes please give me a network that can PAY ATTENTION TO ITSELF. Please give me PRETRAINED WEIGHTS for my YOLO-9000” - Statements dreamed up by the utterly Deranged

LOOK at what “Research Scientists” have been demanding your Respect for all this time, with the statistical methods & optimization algorithms we built for them

(This is REAL "Deep Learning”, done by REAL “ML Engineers”):



?????



????????????????????

“Hello I would like to learn **1.6 TRILLION** parameters please”

They have played us for absolute fools

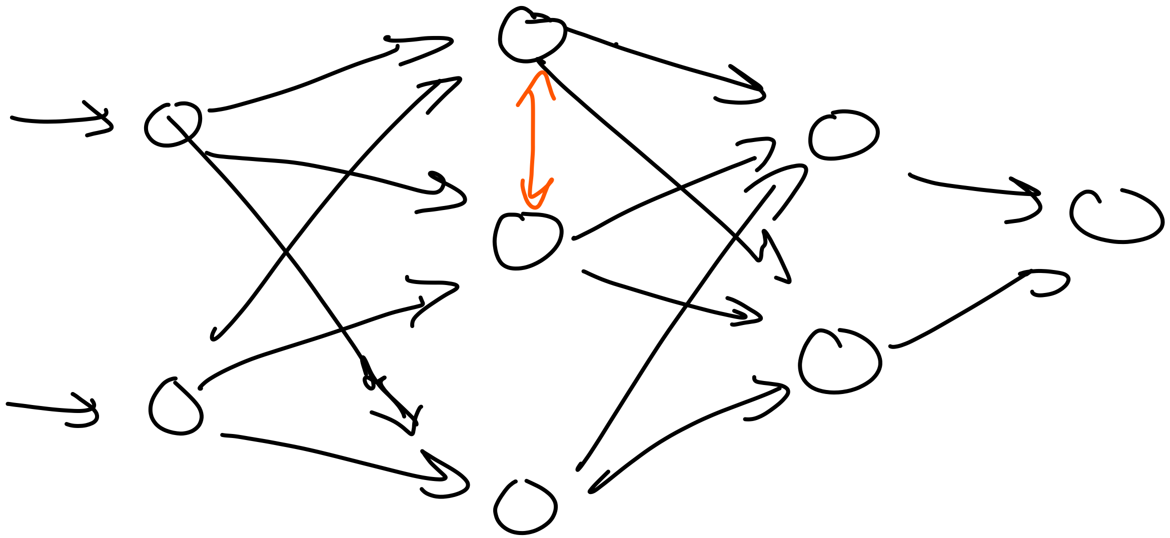
https://www.reddit.com/r/okbuddyphd/comments/n2m6vz/stop_doing_deep_learning/

Deep Learning

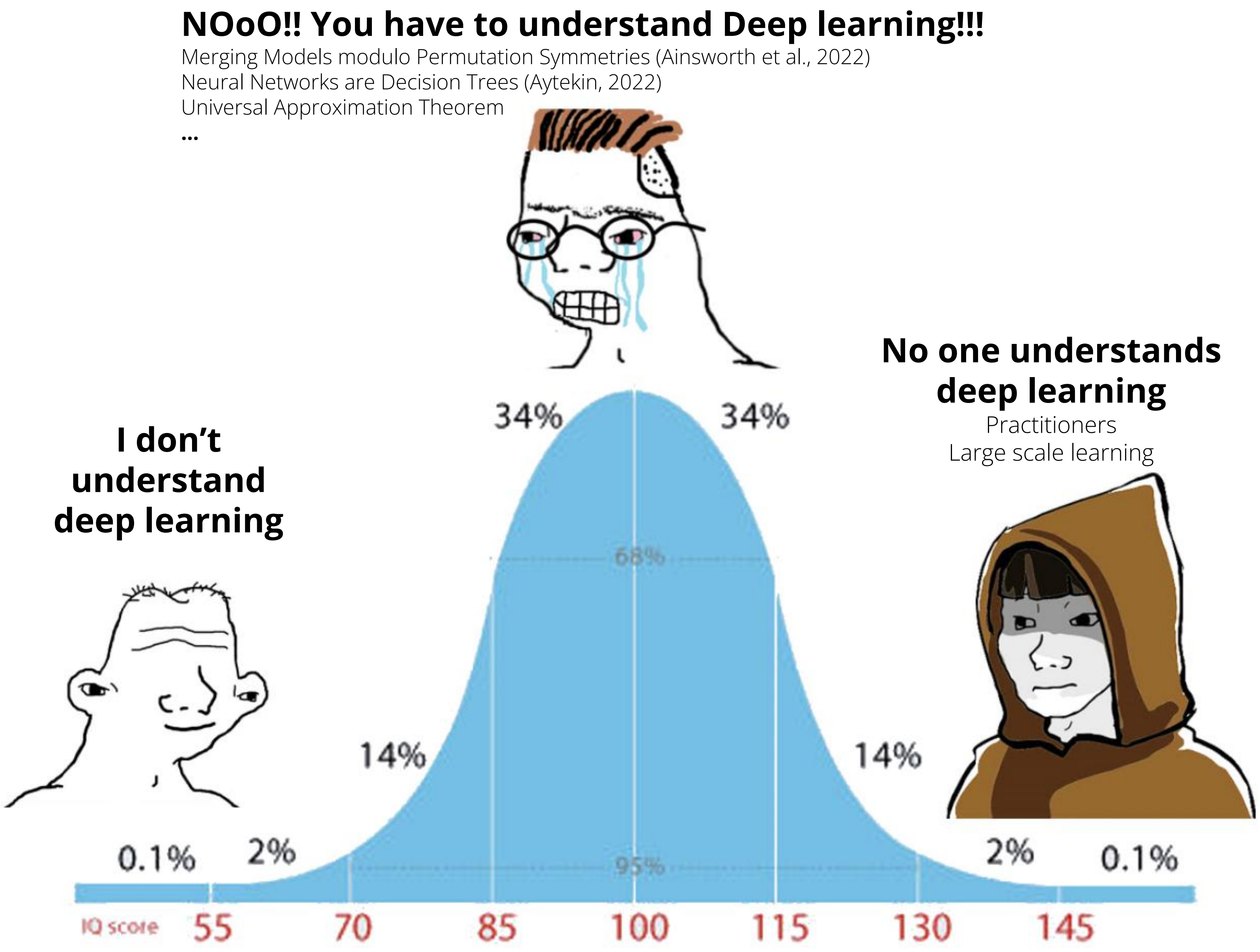
"Dense"

"MLP"

"Fully Connected"



The Mysteries of Deep Learning



The Natural Image Manifold

The **natural image manifold** is a surface embedded in a **high dimensional space** that is:

(1) **low-dimensional**,
(2) **highly nonlinear**, and
(3) **locally smooth**.

- high dim: 100 x 100px, grayscale

  10,000

- low-dim subspace:

  Unif[image] $\approx$ noise

- nonlinear

  $\frac{\text{img1} + \text{img2}}{2} \notin$ image

- locally smooth

  image $+ \varepsilon \approx$ image

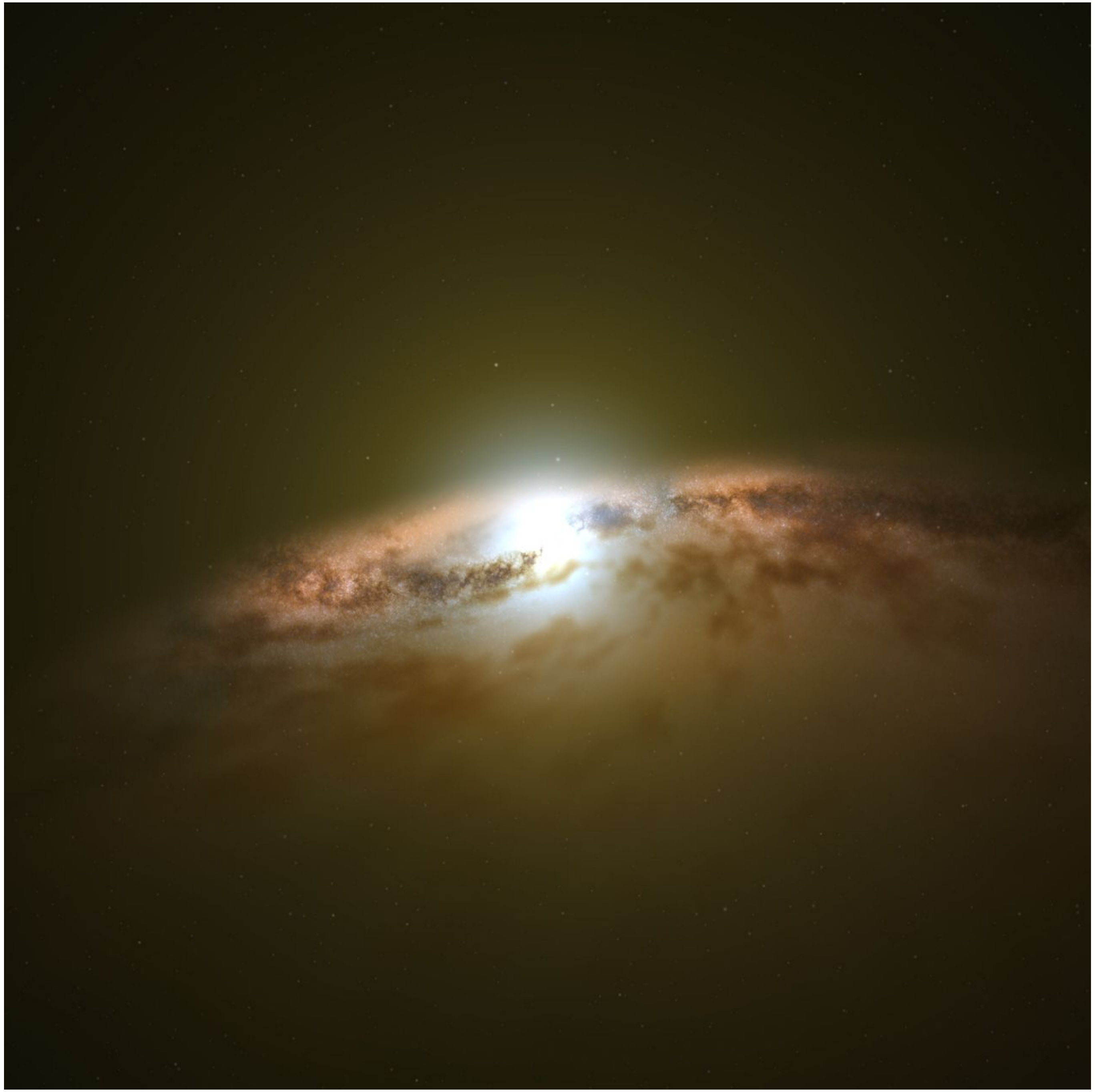
Carnegie Mellon University

The Natural Image Manifold

The **natural image manifold** is a surface embedded in a **high dimensional space** that is:

(1) **low-dimensional**,
(2) **highly nonlinear**, and
(3) **locally smooth**.

Analogy by the late Thomas S. Huang via Atlas Wang



The Natural Image Manifold

The **natural image manifold** is a surface embedded in a **high dimensional space** that is:

(1) **low-dimensional**,
(2) **highly nonlinear**, and
(3) **locally smooth**.

Good:

Computer Vision

Audio Processing

Graphics

Natural Language

↳ Text understanding

↳ Source code

Bad:

Sales

Anything with a simple analytical soln.

Tabular Data

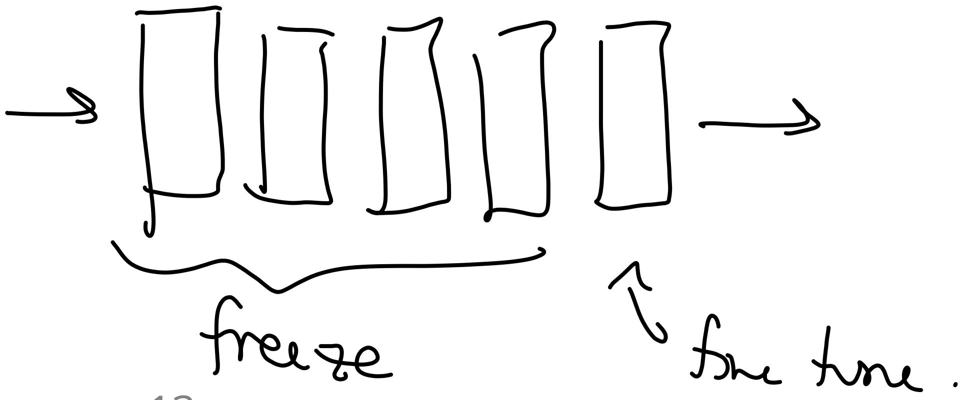
Practical Deep Learning

See **Adversarial Robustness**.
Image from "Robust Physical-World Attacks on Deep Learning Models," Eykholt et al, 2018.

Pre-Trained Models & Transfer Learning

Google

(ResNet, MobileNet, YOLO)



[https://www.reddit.com/r/machinelearningmemes/comments/g1mxz8/transfer_learning_101/]

Practical Deep Learning

Eleuther AI

- Train from scratch?
- Transfer learn?
- Fully pre-trained model?

Deep Learning on the Edge

Please inspect your power cables if you own a RTX 4090, even if you don't use it for deep learning.

Deep Learning on the Edge

- Trained neural network with a given architecture
- Weights are 32-bit float
- Given power budget

f32 → i8 (Quantization)

Pruning

Distillation

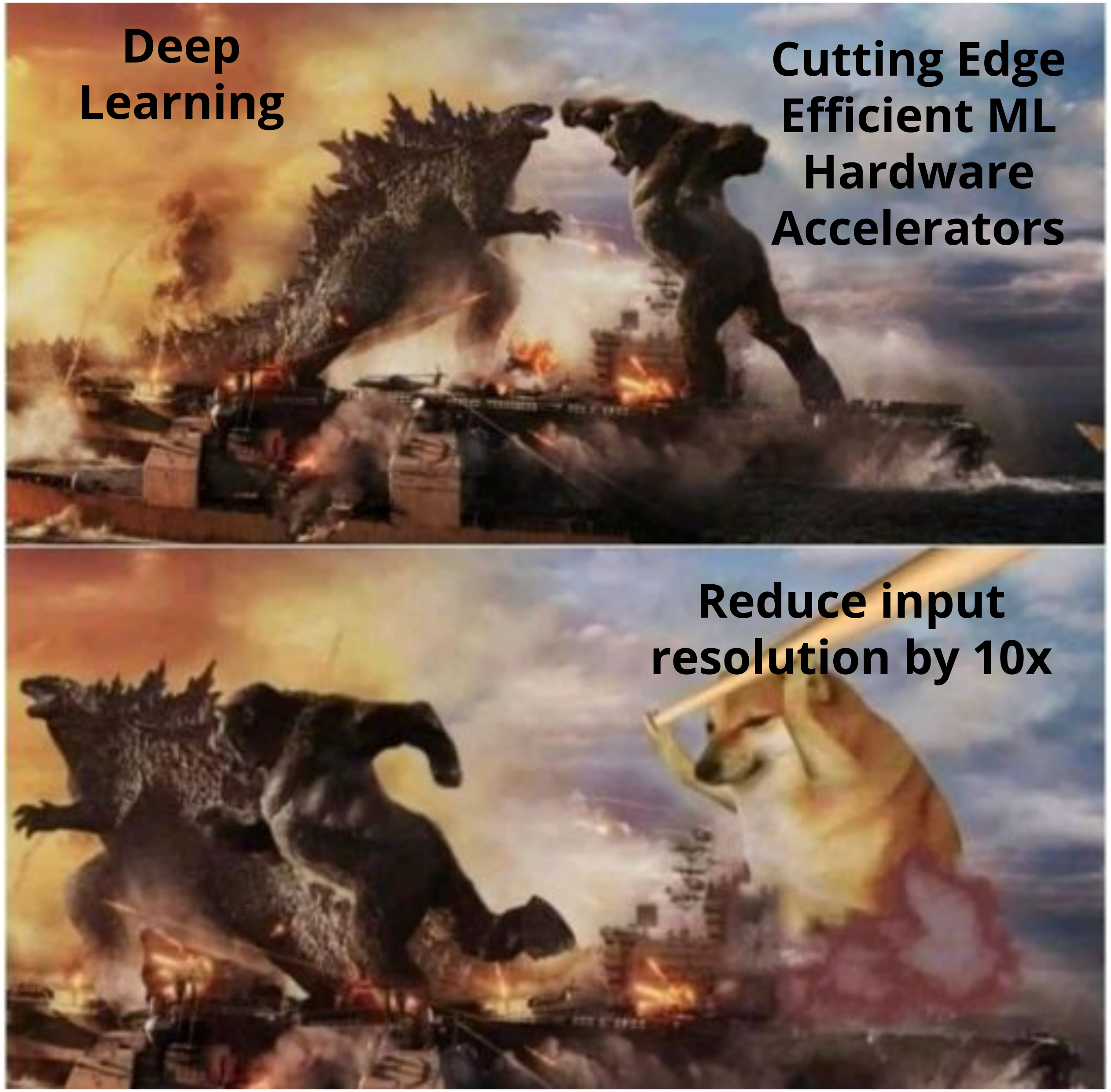
Resolution (space, time)

Offloading

More efficient

Simplest is Best

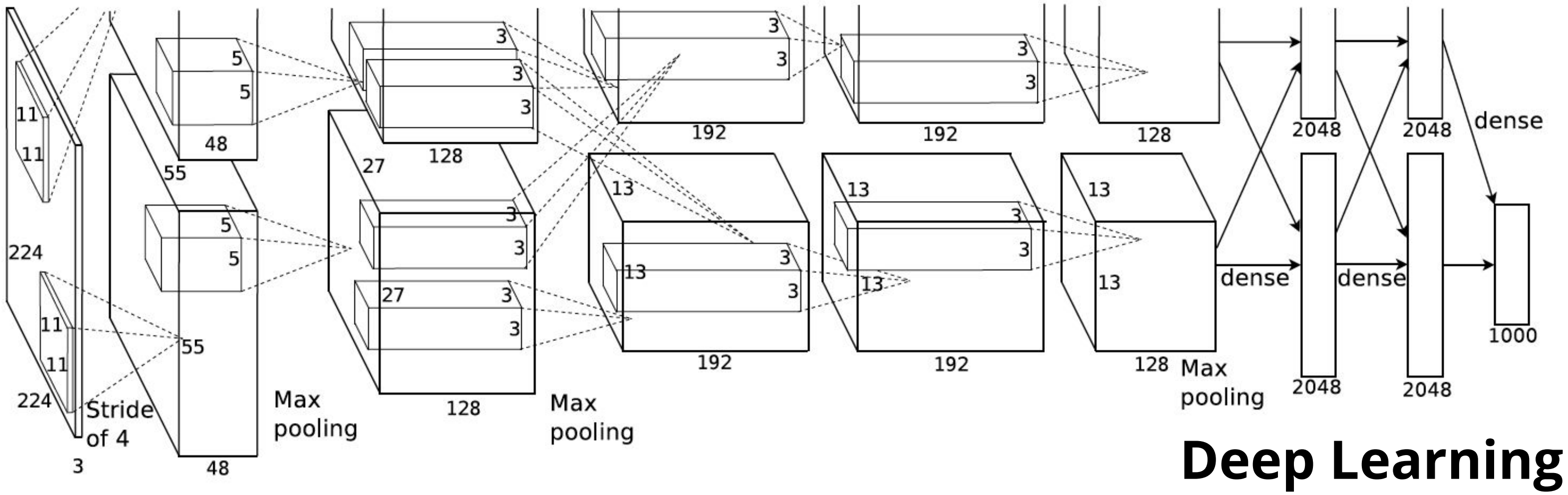
- Cloud Offloading
- Reduce input resolution
- Don't use deep learning



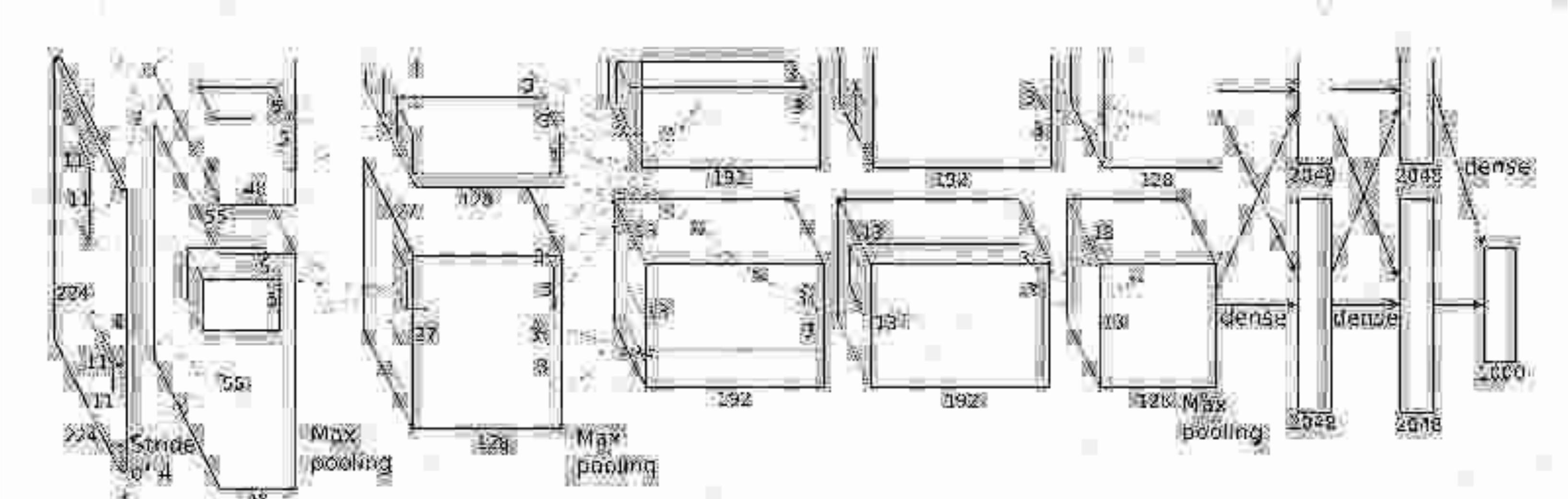
“ImageNet Classification with Deep Convolutional Neural Networks,” Krizhenvsky et al., 2012 (AlexNet - First “true” ConvNet)

Quantization

- float32 → float16
- Even better: int8
- Binary is even possible



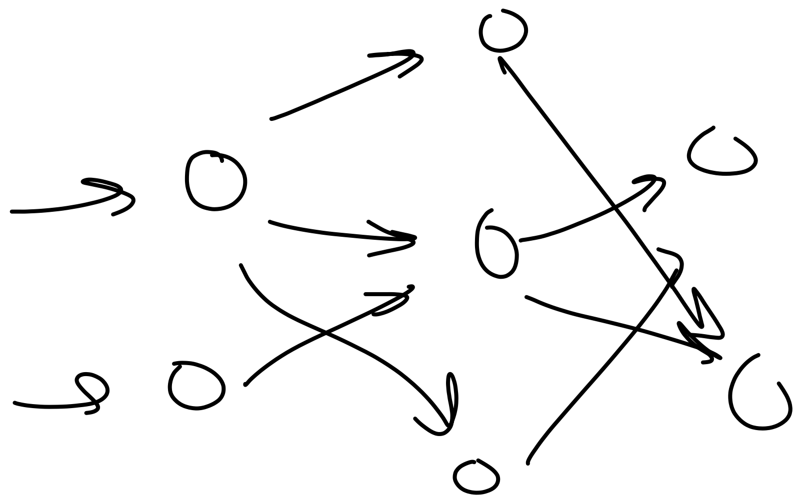
Deep Learning



Q u a n t i z e d Deep Learning

Pruning

- Pruning: Optimal Brain Damage (LeCun et al., 1989)
- Structured Sparsity
- Lottery Ticket Hypothesis



Pruning

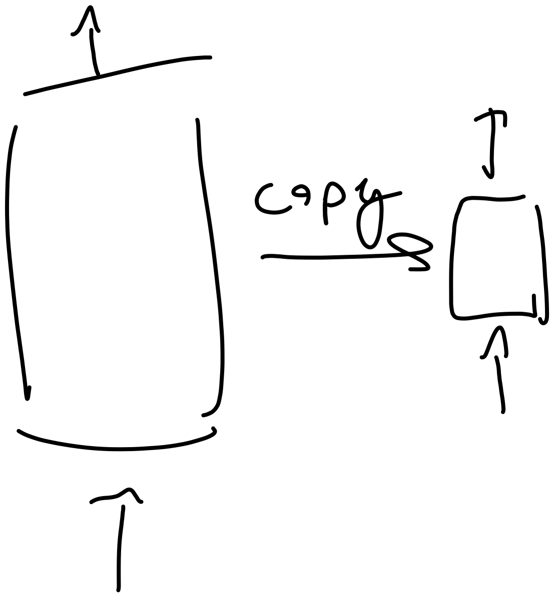
- Pruning
- Structured Sparsity
- Lottery Ticket Hypothesis

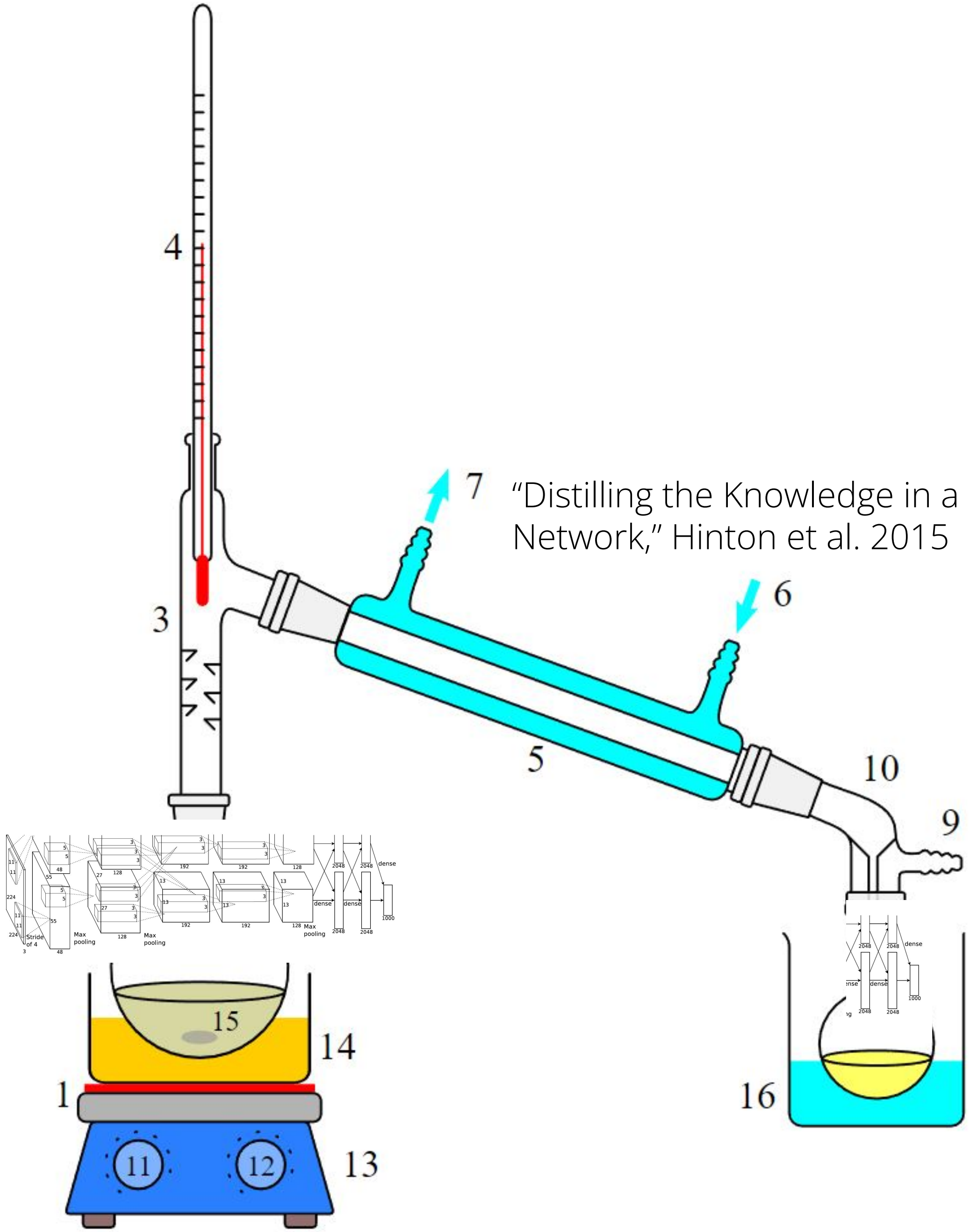
Baseline Model

Quantization Optimized Execution

Optimal Brain Damage
LeCun et al., 1989

Distillation



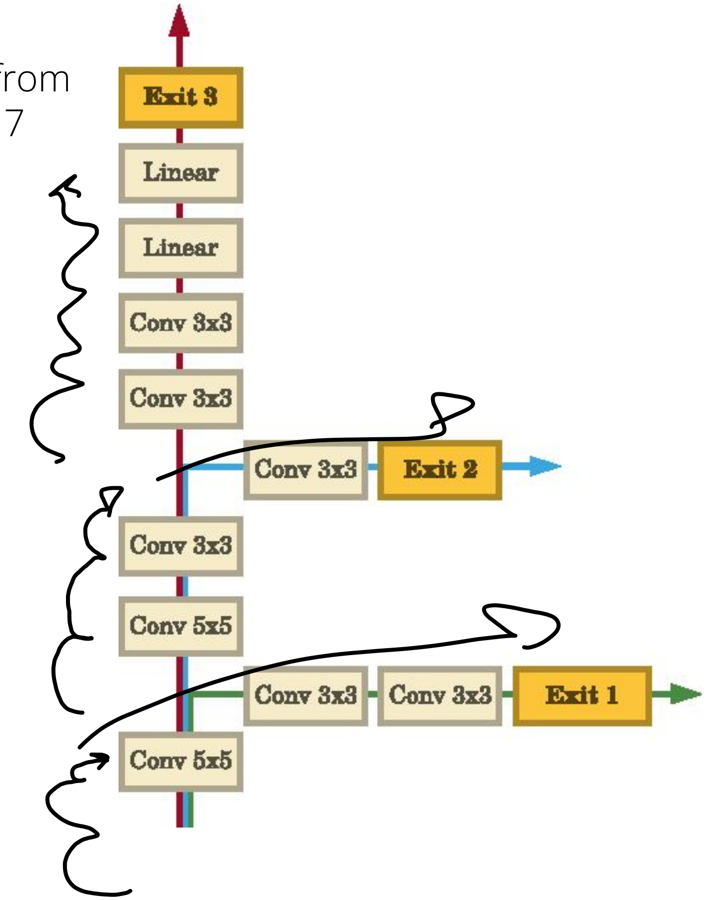


“Distilling the Knowledge in a Neural Network,” Hinton et al. 2015

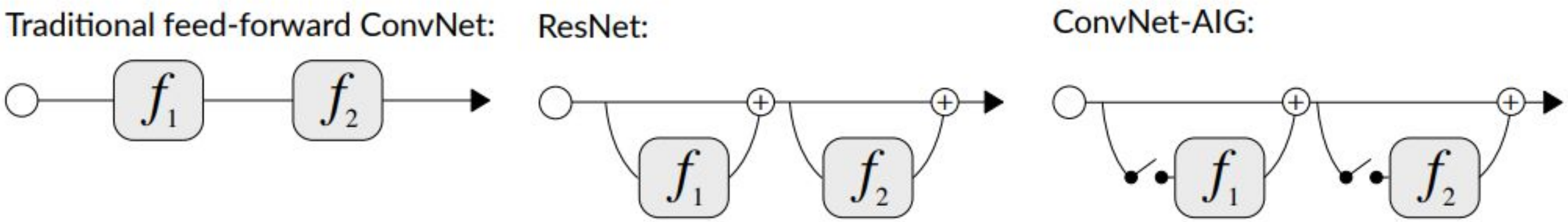
Adaptive Inference

- Different numbers of layers
- Different input resolution
- ...

“BranchyNet: Fast inference via early exiting from deep neural networks,” Teerapittayanon, 2017



“Convolutional Networks with Adaptive Inference Graphs,” Veit & Belongie, 2018



TL;DR

1. Don’t use machine learning
2. Don’t use deep learning
3. Don’t use deep learning on the edge (do it in the cloud)
4. Don’t train deep learning models (use a pre-trained model optimized for edge deployment)
5. Don’t train deep learning from scratch (use transfer learning)
6. Give up (or take a machine learning class)

Additional Topics

- Distributed / federated training
- Edge Training
- GPU vs CPU vs TPU
- Deep Learning Frameworks
- Edge Deployment Frameworks (TF Lite, Pytorch Mobile)
- ML Accelerator Design
- ML Compilers
- Efficient Architectures
- Anything ML Related