On Representation Learning

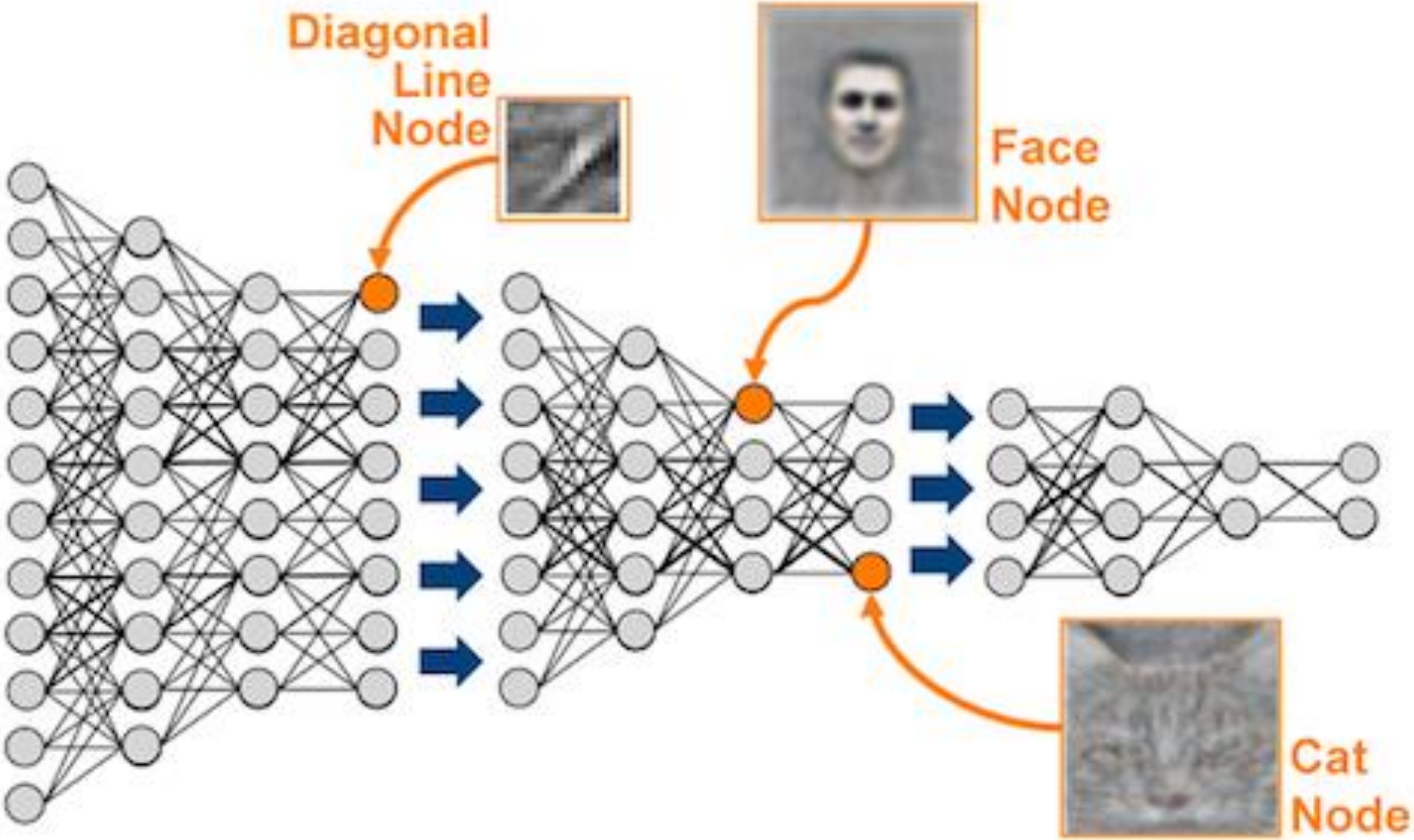
Alexander G. Ororbia II

Introduction to Machine Learning

CSCI-736

1/26/2023

Companion reading:
Chapter 6-8, & 15 of Deep Learning textbook


Diagonal
Line
Node
Face
Node
Cat
Node

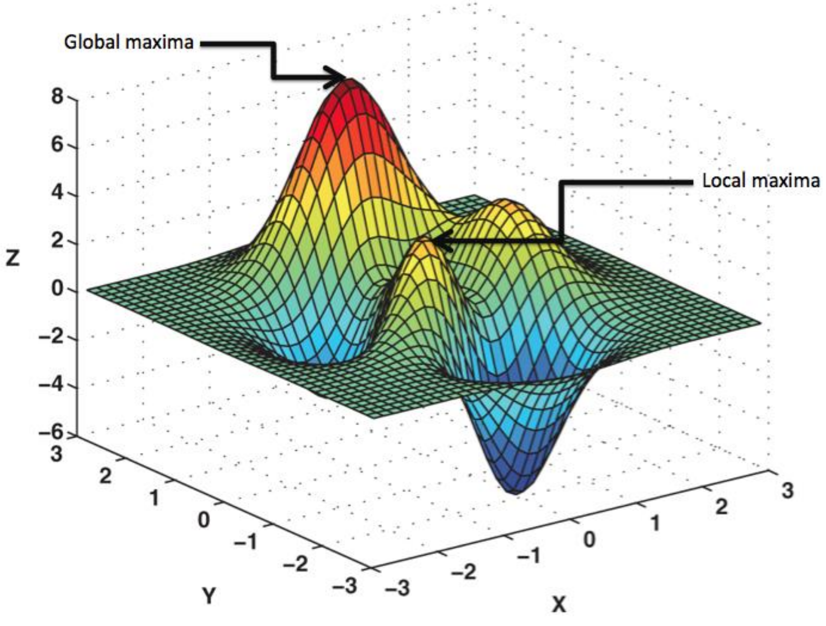
Backpropagation of Errors

White board time!

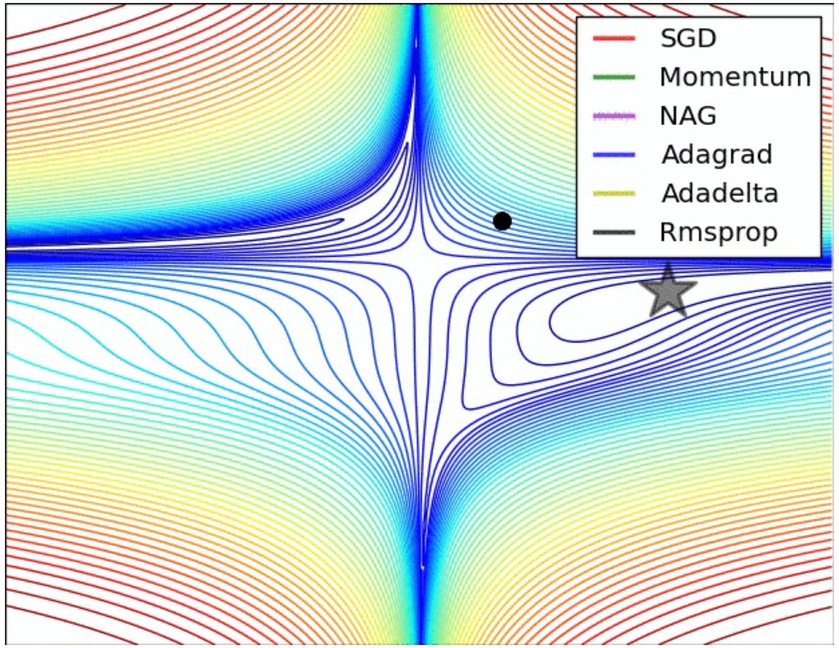
The Vanishing Gradient Problem

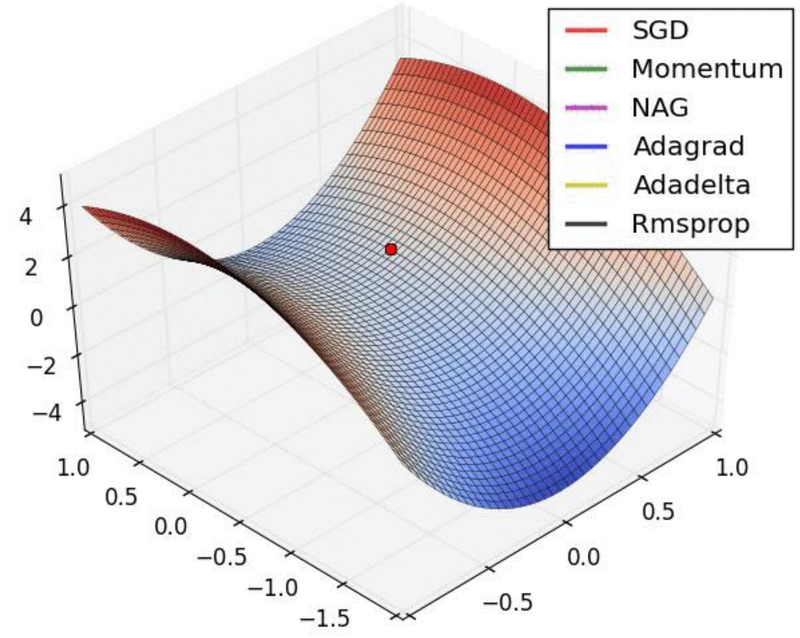
- Solving credit assignment problem with back-propagation too difficult
  - Difficult to know how much importance to accord to remote inputs (Bengio et al., 1994)
  - Information passed through a chain of multiplications back through network
    - Any value slightly less than 1 in hadamard product, and derivative signal quickly shrinks to useless values (near zero)
  - Learning long-term dependencies in temporal sequences becomes near impossible
- Complementary problem: Exploding gradients
  - Any value greater than 1 in hadamard, derivative signal increases dramatically (numerical overflow)

What do we do with the gradients?



- Use a method of parameter adjustment – "update rule", "optimizer"
- Gradient descent (GD), or mini-batch GD
  - Use estimator (i.e., backprop) to get gradient, then update parameters; online case = stochastic gradient descent (SGD)
- Alternative optimizers = shiny toys to make learning even faster
  - SGD + momentum, RMSprop, Adam, etc.





http://cs231n.github.io/neural-networks-3/#hyper

Random Parameter Initializations

- Classical approaches
  - Sample from *~U(-a, a)*, where is *a* small scalar
  - Sample from *~N(0, a)*, where is *a* small standard deviation
- Fan-in-Fan-out (number inputs, number output)
  - Calibrate by variances of neuronal activities
- Simple distributional schemes
  - Fan-in/Fan-out Uniform
  - Fan-in/Fan-out Gaussian (good for ReLU activations)
- Orthogonal Initialization
  - Use Singular Value Decomposition (SVD) to find initial weights
- Identity Initialization / Constraint (for RNNs)
  - Does not always work unless constraint is enforced
- Or other intelligent methods?
  - Greedy layer-wise pre-training (we will go over this later in the course!)

Why Do We Care How Parameters Are Initialized?

- Initialization affects final performance
  - Will put closer to some spots in function space and farther from others

- Where we end up in function space will often correlate w/ our error performance

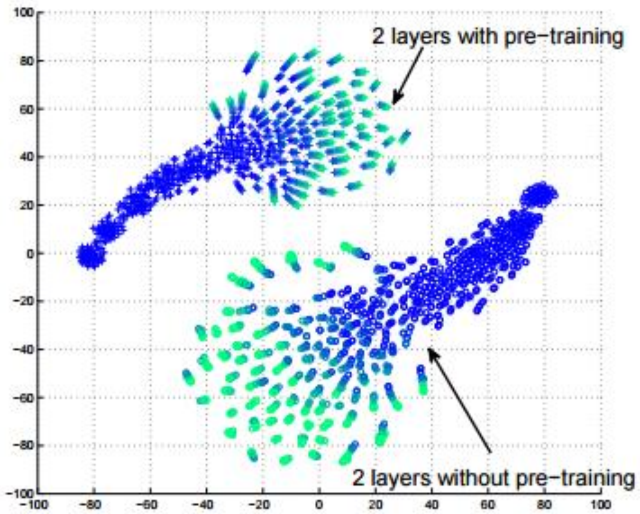


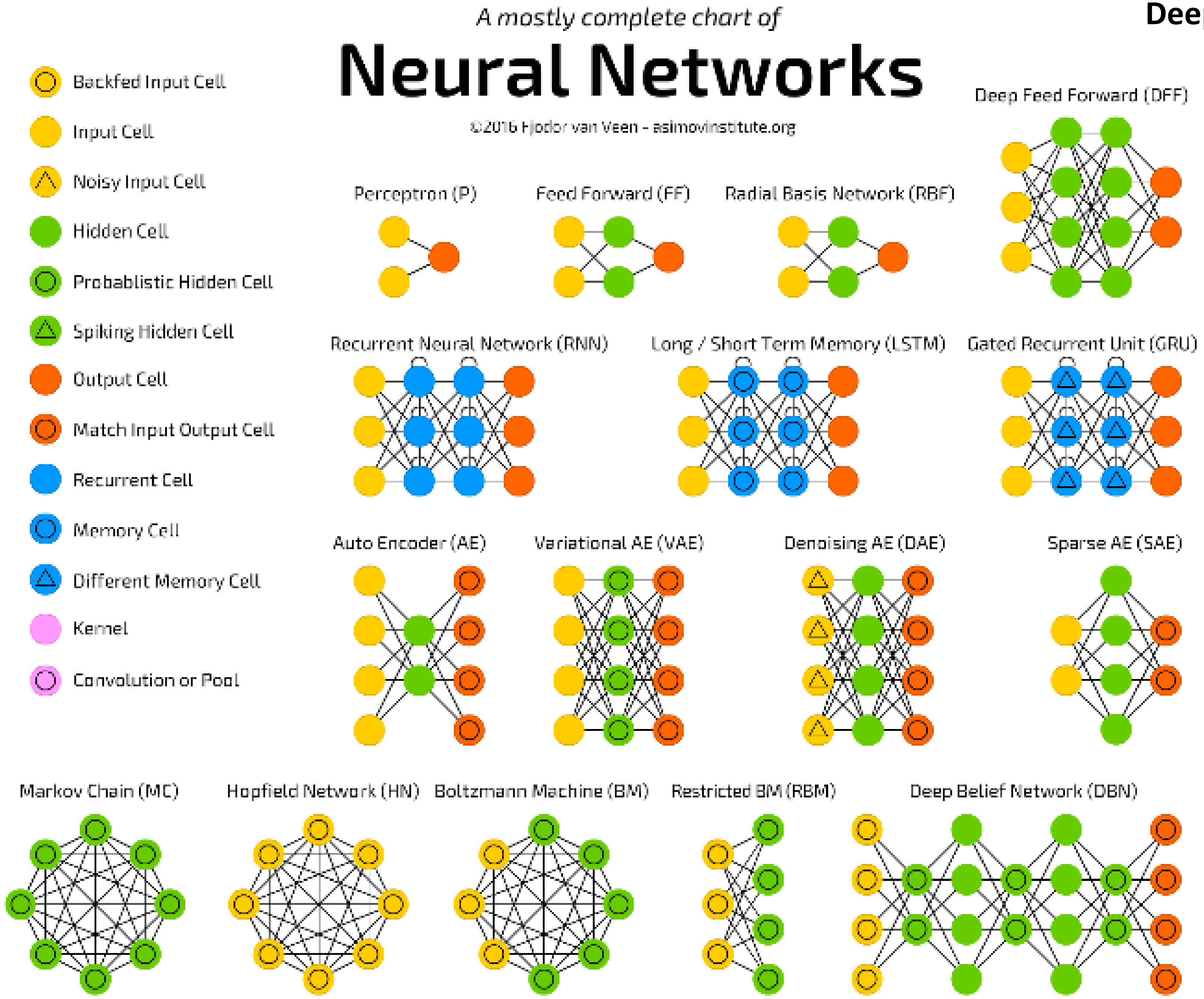
Figure 5: 2D visualizations with tSNE of the functions represented by 50 networks with and 50 networks without pre-training, as supervised training proceeds over MNIST. See Section 6.3 for an explanation. Color from dark blue to cyan and red indicates a progression in training iterations (training is longer without pre-training). The plot shows models with 2 hidden layers but results are similar with other depths.

“Why Does Unsupervised Pre-training Help Deep Learning?”, Erhan et al. 2010 http://jmlr.org/papers/volume11/erhan10a/erhan10a.pdf

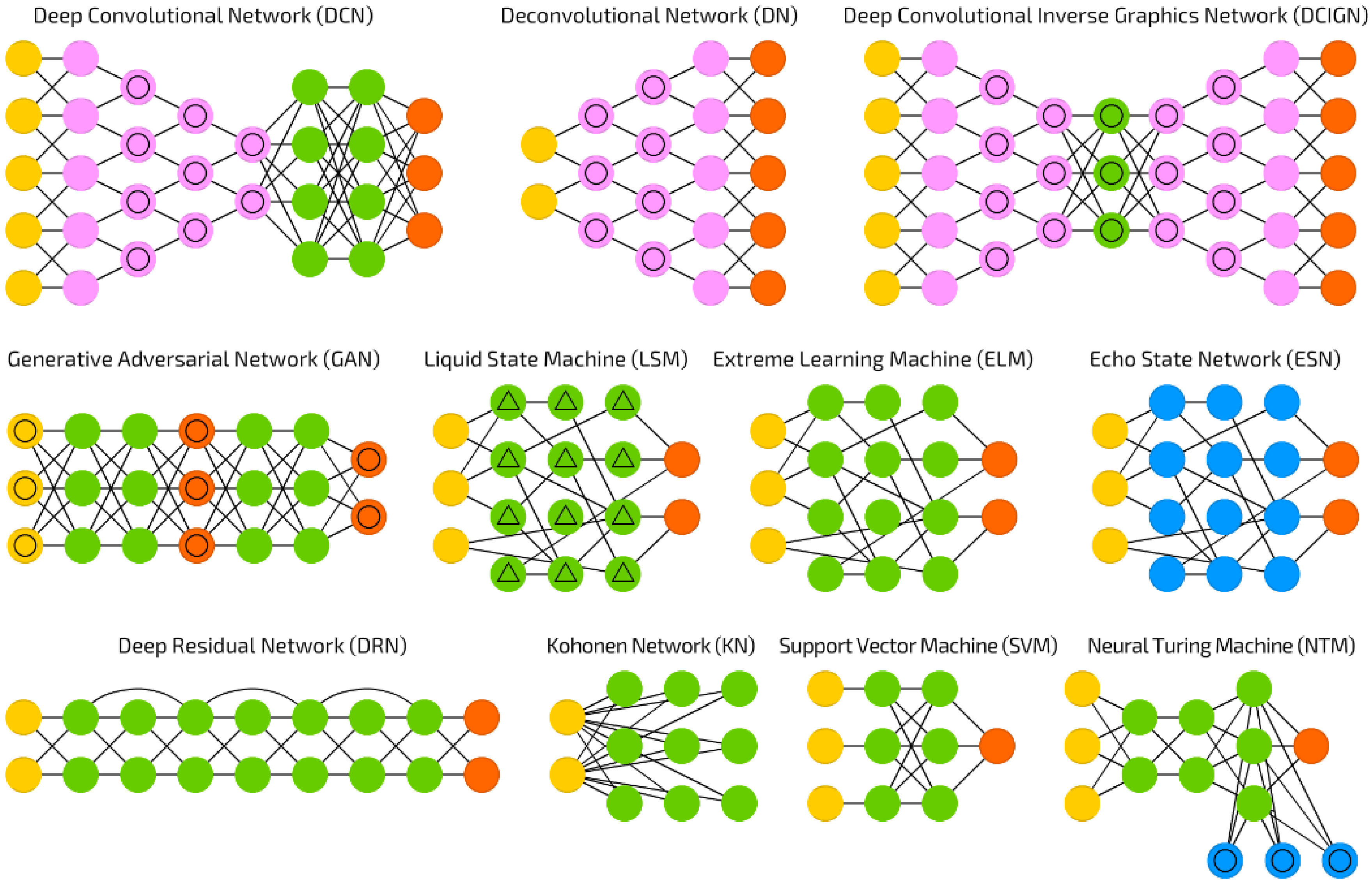
On the plethora of model structures…

THE SPACE OF NEURAL ARCHITECTURES

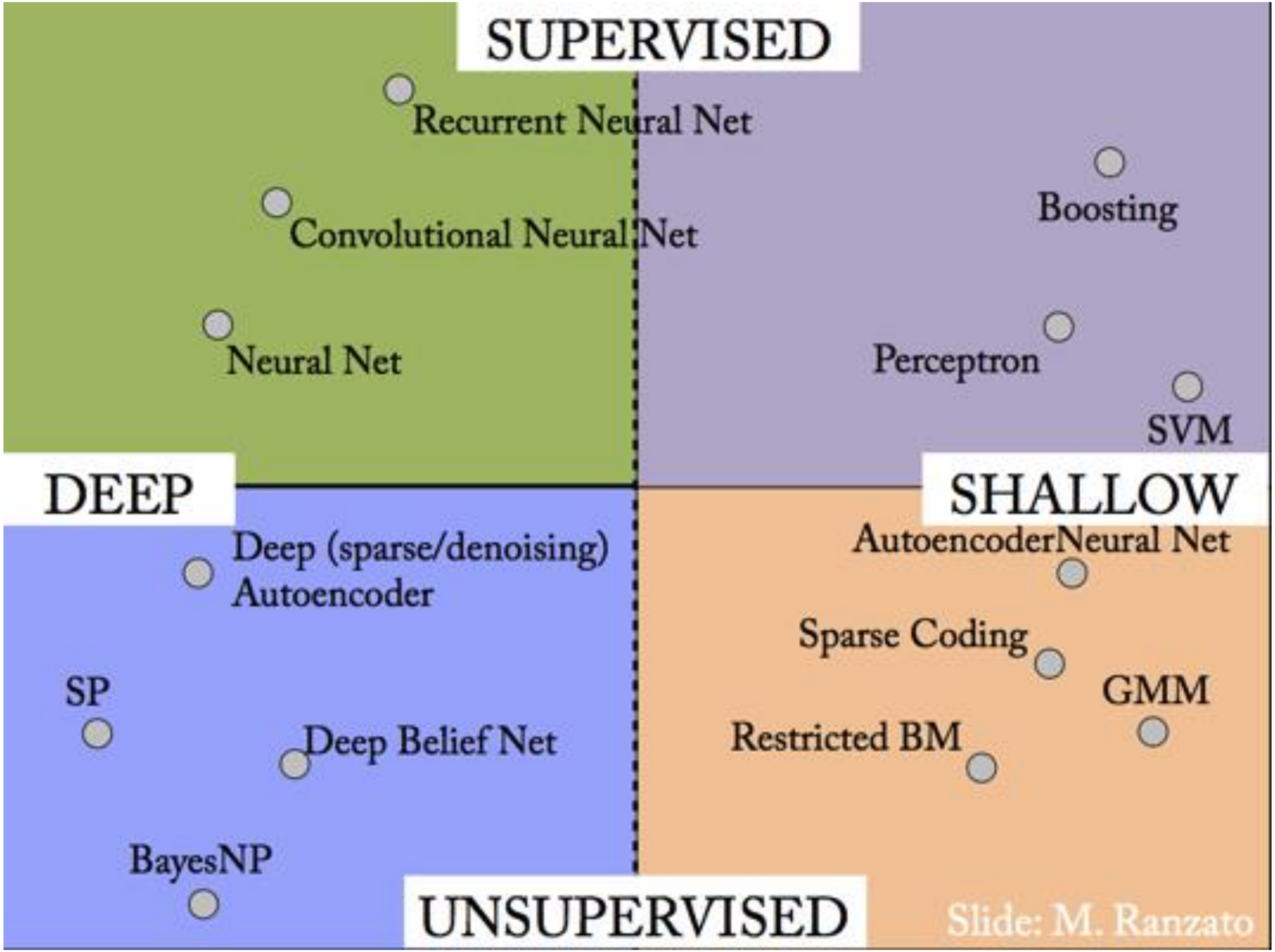
Deep zoo!



http://www.asimovinstitute.org/neural-network-zoo/



http://www.asimovinstitute.org/neural-network-zoo/

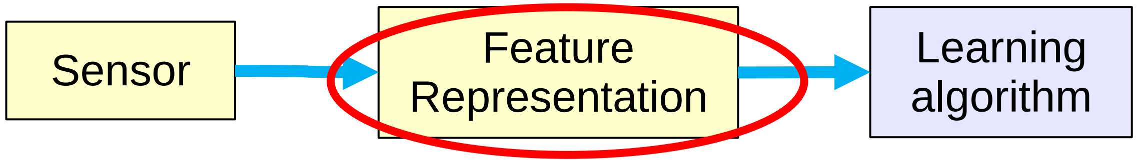

SUPERVISED
Recurrent Neural Net
Convolutional Neural Net
Neural Net
Boosting
Perceptron
SVM
DEEP
SHALLOW
Deep (sparse/denoising) Autoencoder
SP
Deep Belief Net
BayesNP
AutoencoderNeural Net
Sparse Coding
GMM
Restricted BM
UNSUPERVISED
Slide: M. Ranzato

Presentations Next Tuesday

- Topics: Representation Learning
- Paper announcements coming soon (once the two teams respond)

What is Representation Learning?

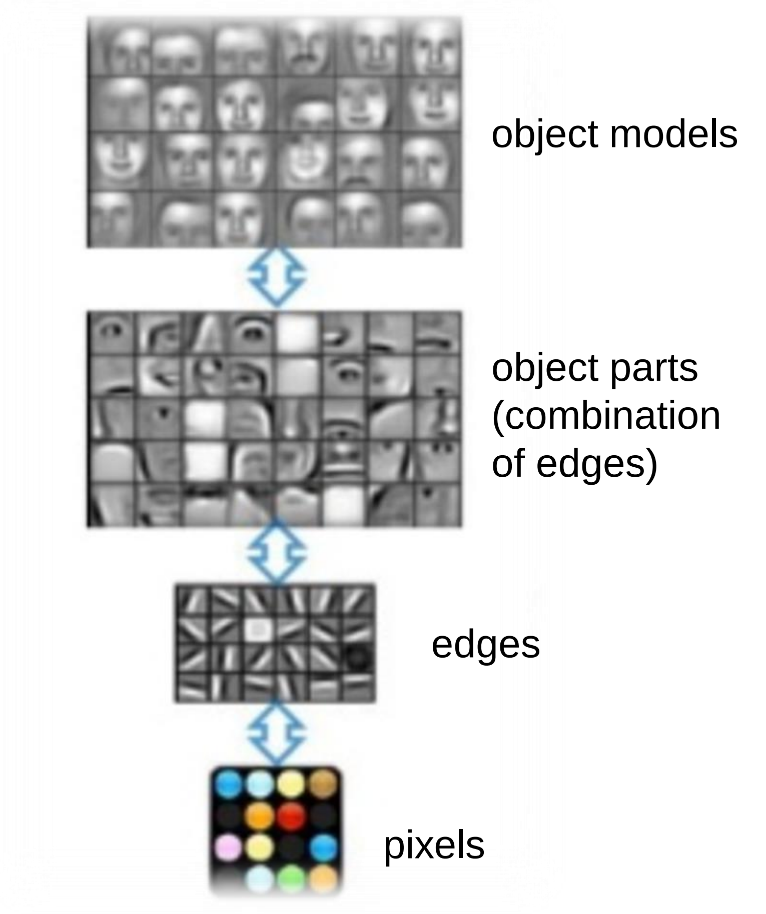
- Learn features automatically
  - Find a transformation of raw data input to a representation / space that can be effectively exploited in machine learning tasks
- Can be viewed as complementary to machine learning
  - "Automatic" pre-processing (or automated feature engineering)
- What makes one representation better than another?
  - ***Good representation*** = one that makes a subsequent learning task easier (choice of representation depends on the choice of subsequent learning task)



Why? Feature Abstraction

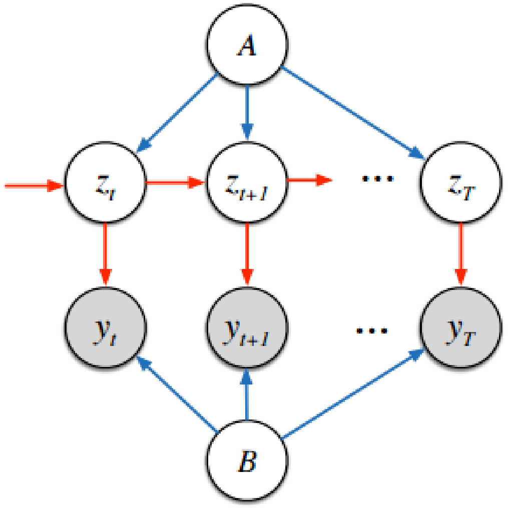
- Raw features, such as pixel values of image, viewed as "low-level" representation of data
  - Can be complex & high-dimensional
  - Observed variables ("nature", observed/recorded data)

- **Abstract representations** = layers of feature detectors
  - Unobserved variables that describe observed variables
    - Capture key aspects of data's underlying stochastic process
    - Many concepts can be represented as (strict) hierarchies (such as a taxonomy of species) → goal of model is to "learn" a plausible, structured unknown hierarchy
  - ***Goal***: extracting "structure" from"unstructured"/messy data



http://www.slideshare.net/roelofp/2014-1021-sicsdlnlpg

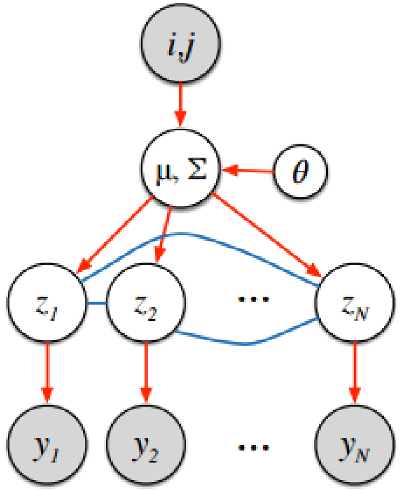
What might deep representational models look like?



Gaussian Linear State Space Model
Kalman Filter

$$z_t \sim \mathcal{N}(z_t | A z_{t-1}, \sigma_z^2 I)$$

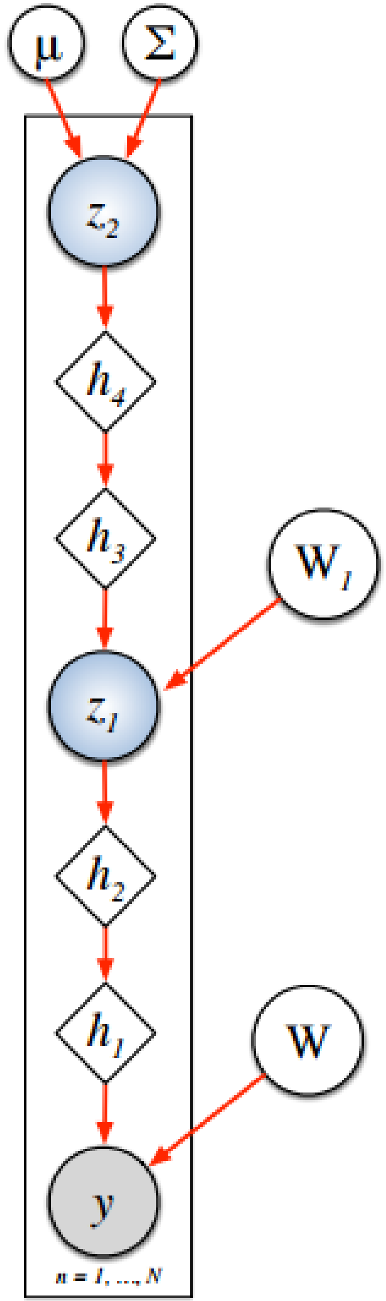
$$y_t \sim \mathcal{N}(y_t | B z_t, \sigma_y^2 I)$$



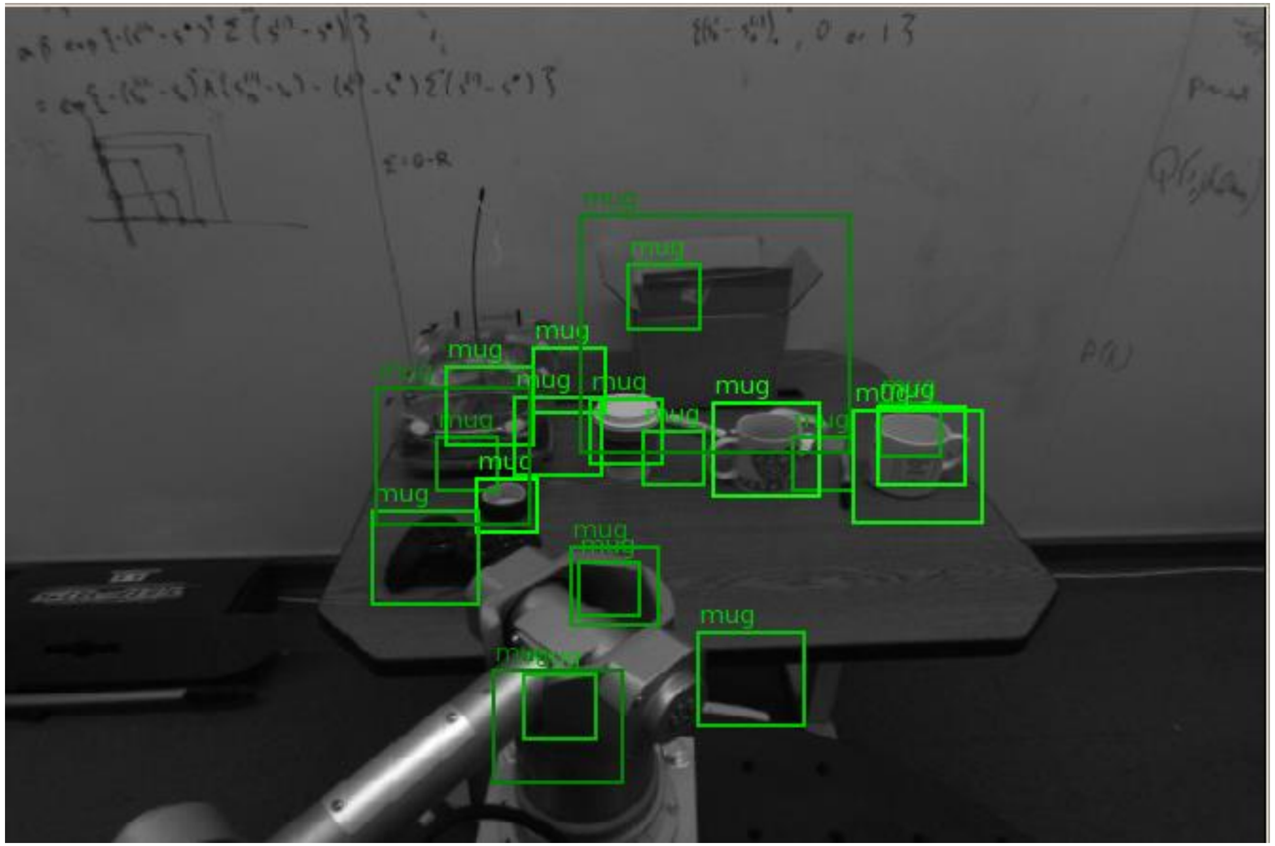
Latent Gaussian Cox Point Proces

$$x \sim \mathcal{N}(x | \mu(i,j), \Sigma(i,j))$$

$$y_{ij} \sim \mathcal{P}(c \exp(x_{ij}))$$



Computer Vision is Hard



How is computer perception done?

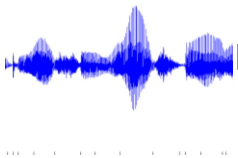
Object detection

Image

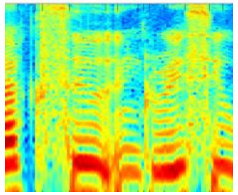
Low-level vision features

Recognition

Audio classification



Audio

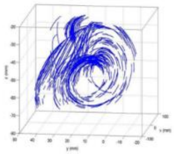


Low-level audio features

Speaker identification

Helicopter control

Helicopter



Low-level state features

Action

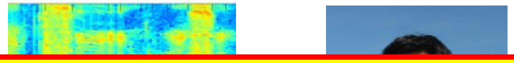
How is computer perception done?

Object detection

Image

Low-level vision features

Recognition



Au

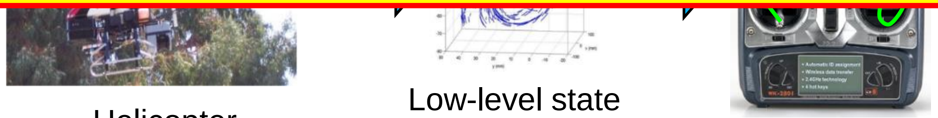
classi

Problems of hand-tuned features

1. Needs expert knowledge
2. Time-consuming and expensive
3. Does not generalize to other domains

Helic

control



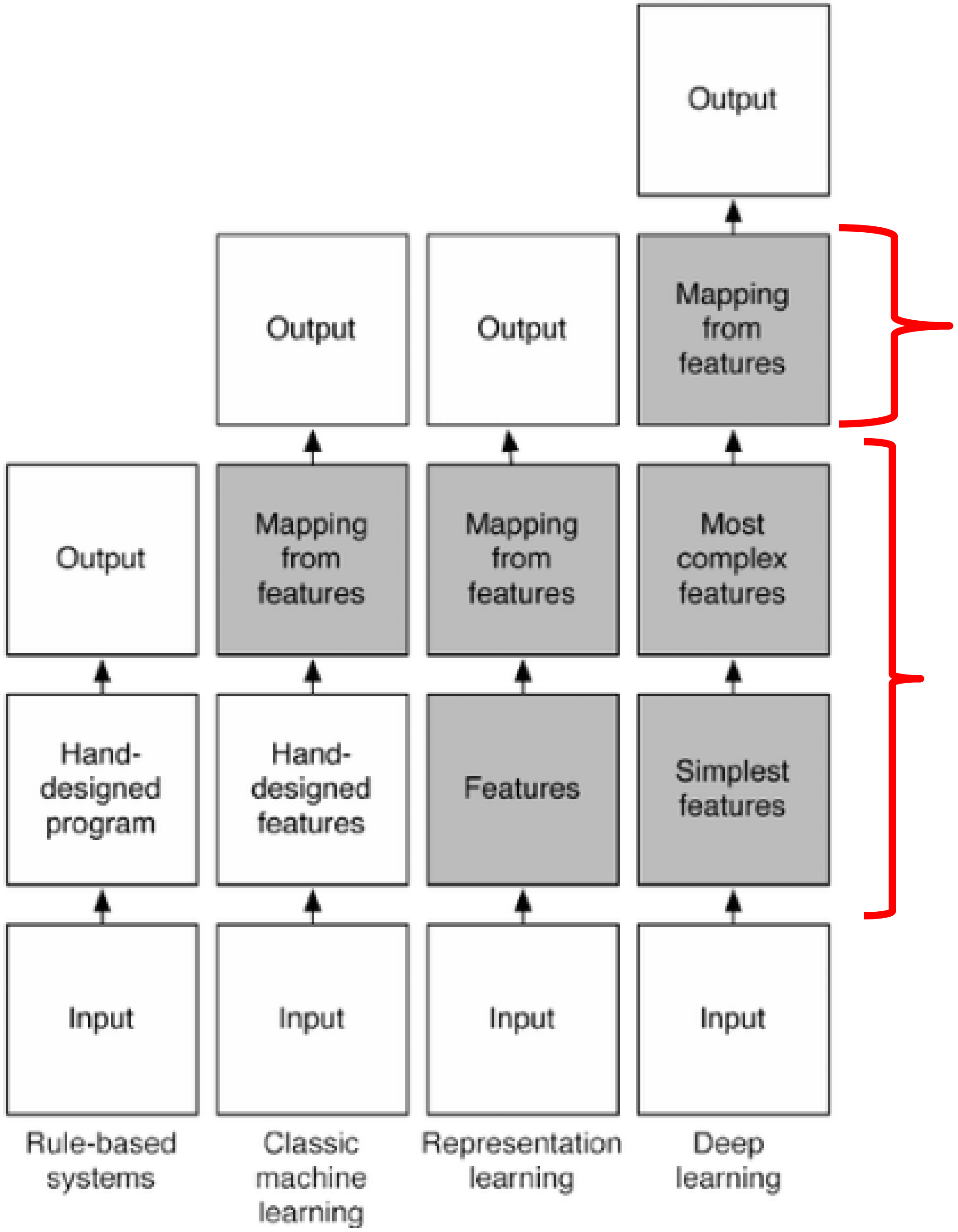
Helicopter

Low-level state features

Action

In context of a deep ANN:

Training w/ supervised criterion naturally leads to representation at every hidden layer (more so near top hidden layer) taking on properties that make discriminative/task-centric learning easier



Linear classifier/regressor

Representation = set of transforms w/in ANN

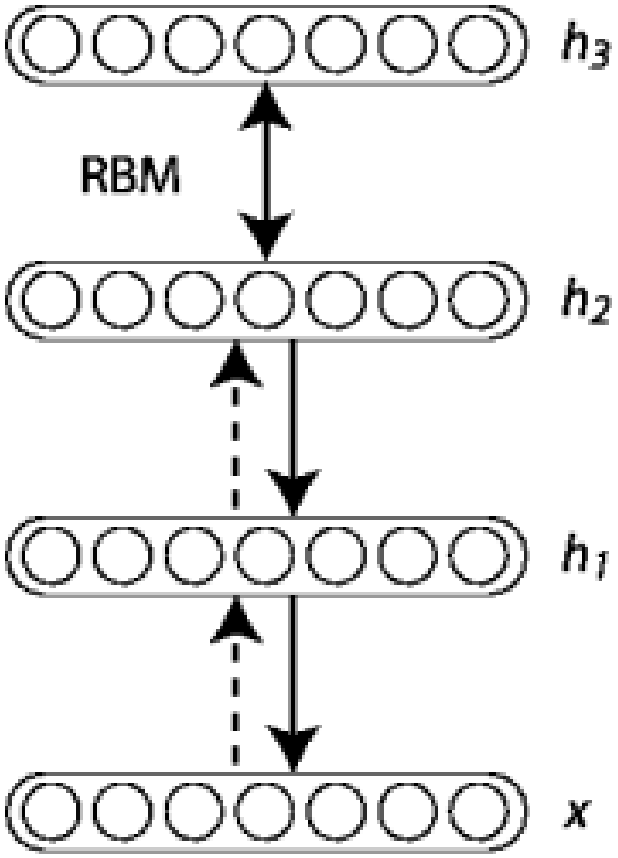
The Problem of Representation Learning

- Representation learning problems face trade-off between preserving as much information about input (as possible) and attaining nice properties, i.e., independence of detectors
  - Models (supervised or unsupervised) have main training objective but learn a representation as a "side effect"
- Often add constraints to shape representation in some wa
  - Density estimation – encourage elements of representation/latent vector **z** to be independent (distributions w/ more independences are easier to model)
- Offers a pathway to facilitate semi-supervised learning
  - ***Hypothesis***: unlabeled data can be used to learn a good representation

<u>Pre-Training</u>: Learning Your Initialization

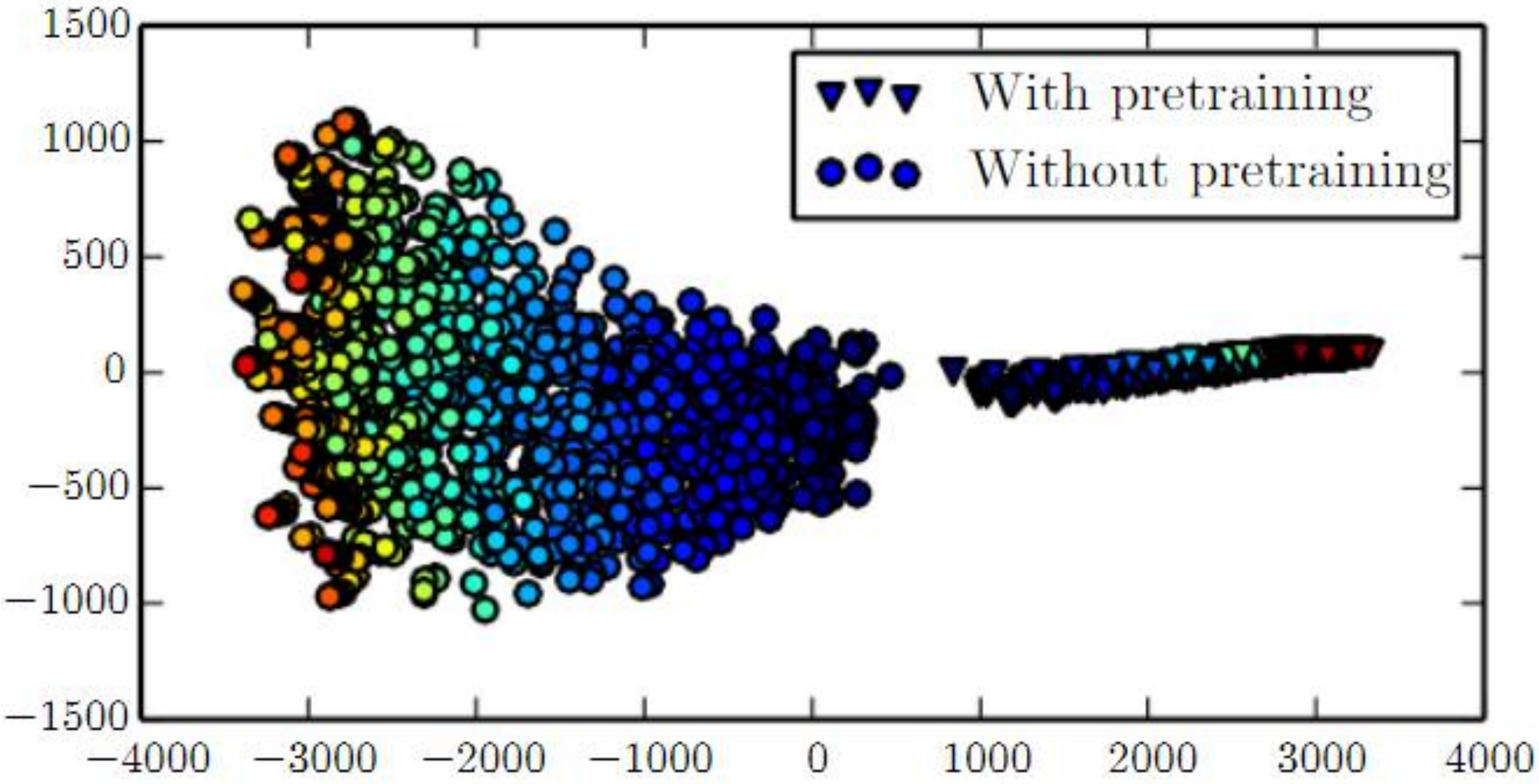
- General idea:
  - Train another model, i.e., deep belief network →
  - Dump its parameters into the one you care about
  - Fine-tune final model

- Unsupervised generative models were largely useful for this



$$P(x, h^1, \ldots, h^\ell) = \left(\prod_{k=0}^{\ell-2} P(h^k | h^{k+1}) \right) P(h^{\ell-1}, h^\ell)$$

<u>With:</u> $P(h^{k-1} | h^k)$, $P(h^{\ell-1}, h^\ell)$, $x = h^0$



1500
1000
500
0
−500
−1000
−1500
−4000
−3000
−2000
−1000
0
1000
2000
3000
4000
With pretraining
Without pretraining

QUESTIONS?