Deep Learning

Методы глубокого обучения

Agenda

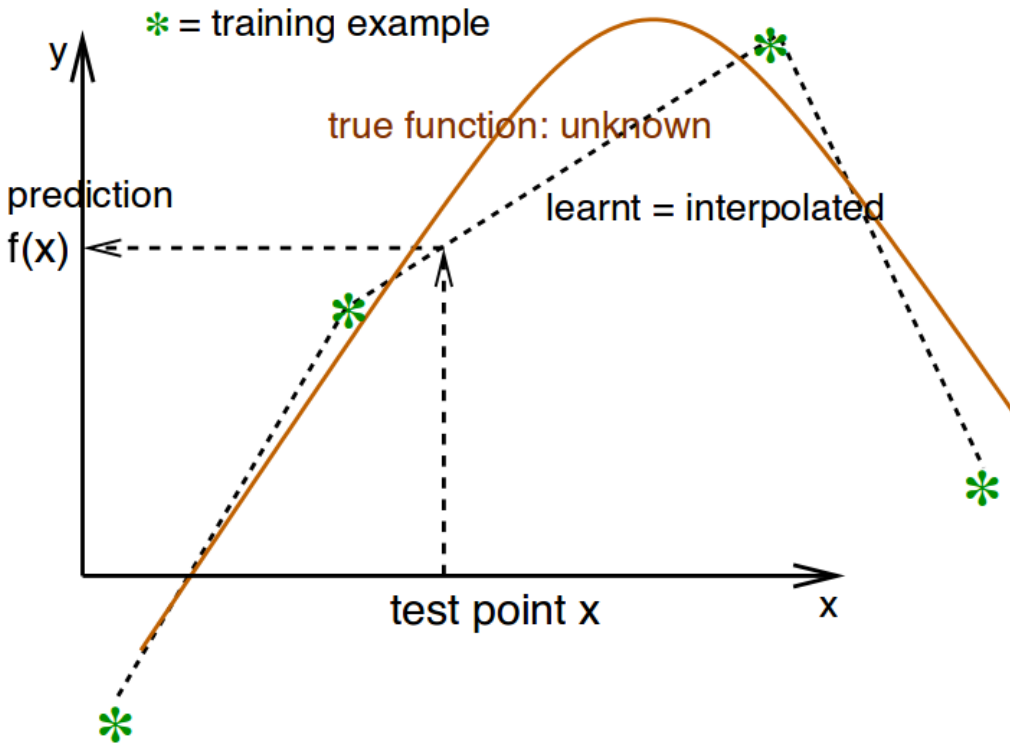
- Motivation & History
- Supervised, unsupervised and RL
- Sparse coding
- Autoencoders
- RBM, DBN, DBM
- CNN
- RNN & LSTM
- Reinforcement Learning
- Need more!

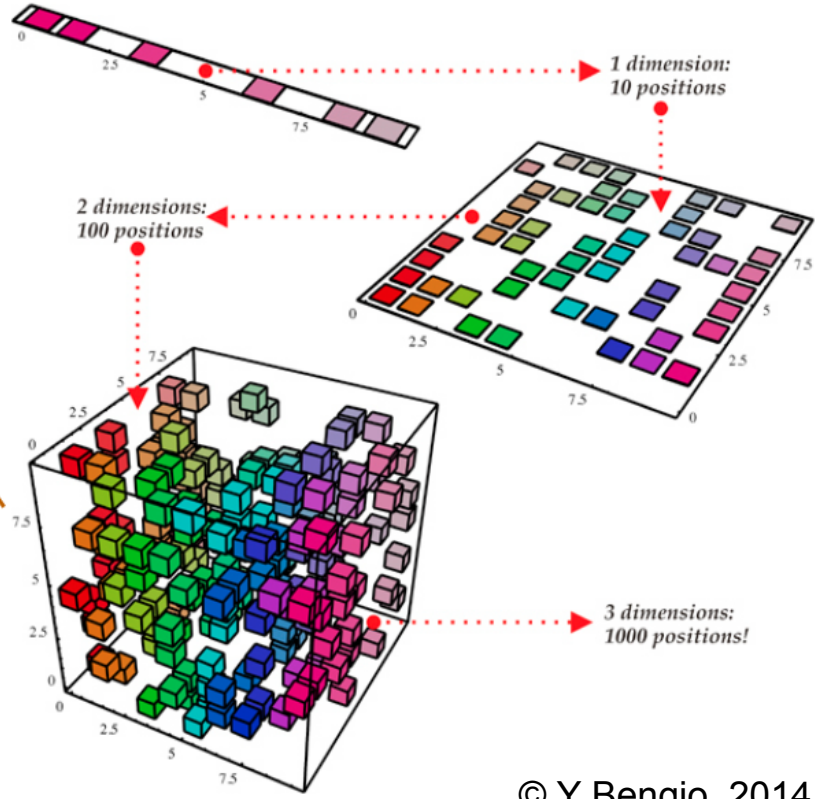
Current results

- Image recognition (try it at http://deeplearning.cs.toronto.edu)
- Speech recognition (Android)
- NLP (translation, Siri, Google Now)
- Multi--modal models
  - Image captioning
  - Image translation (try Google translate mobile)
  - Speech translation (Skype)
  - Emotion recognition
- Reinforcement learning (robotics)
- …

Motivation / The curse of dimensionality

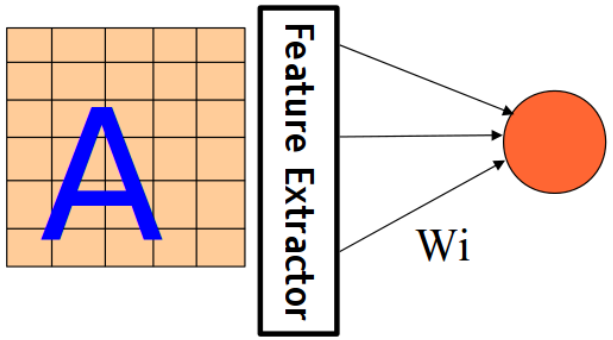
- Learning
- Generalization

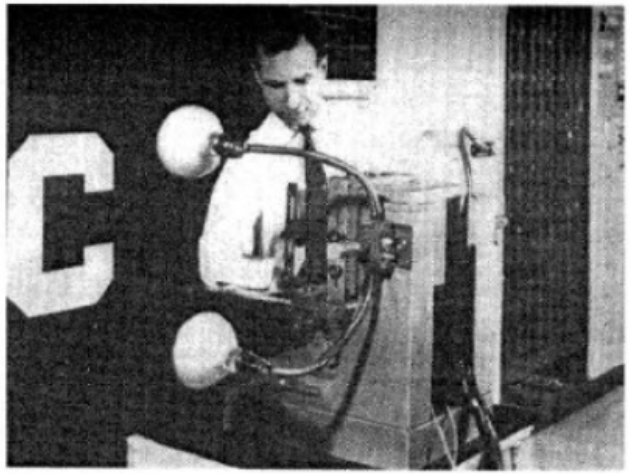


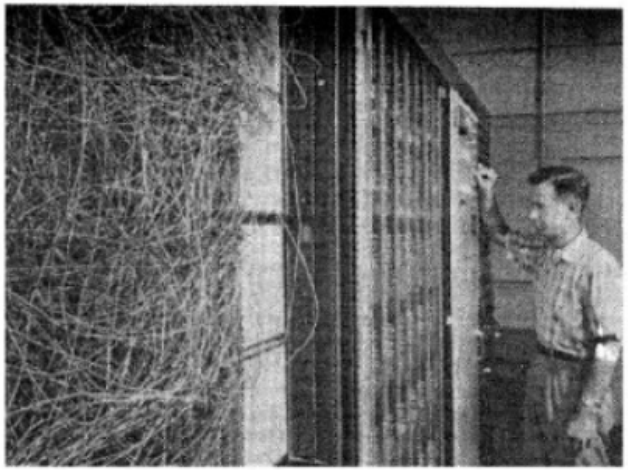


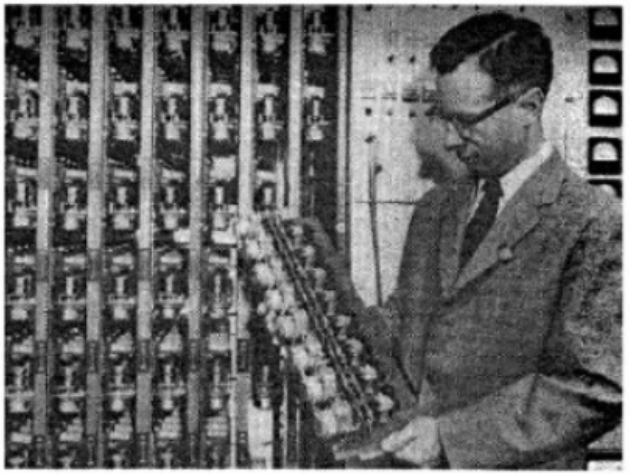
History / Early ML (Since the late 50's)

- Hand-crafted feature extractor + Trainable classifier
  - Perceptron (1957): 1-layer ANN

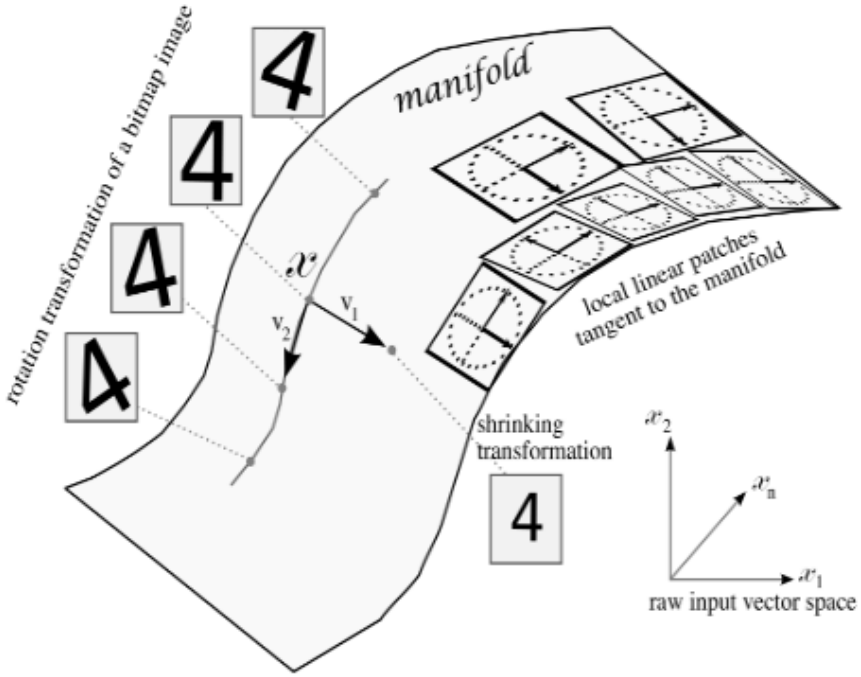


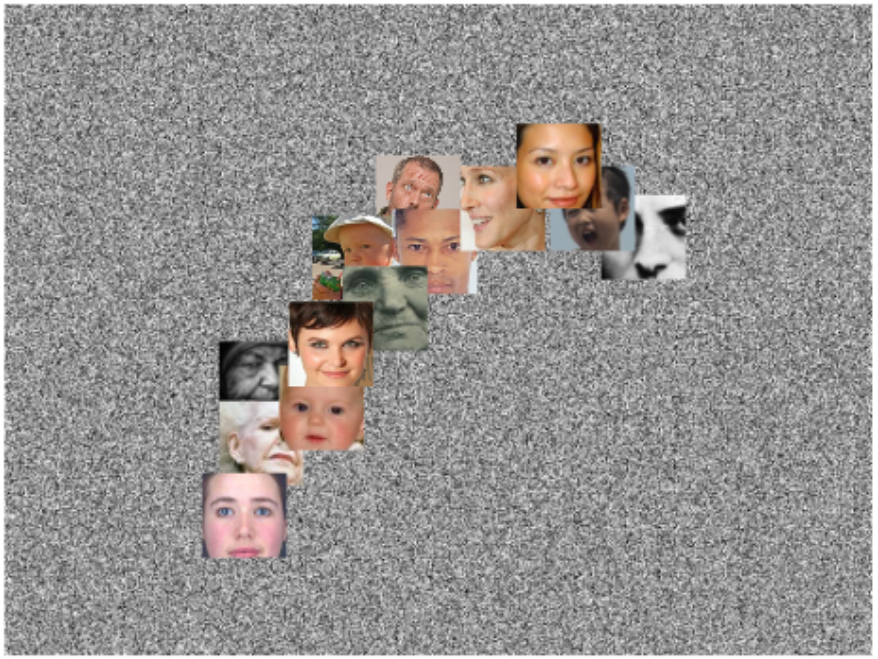






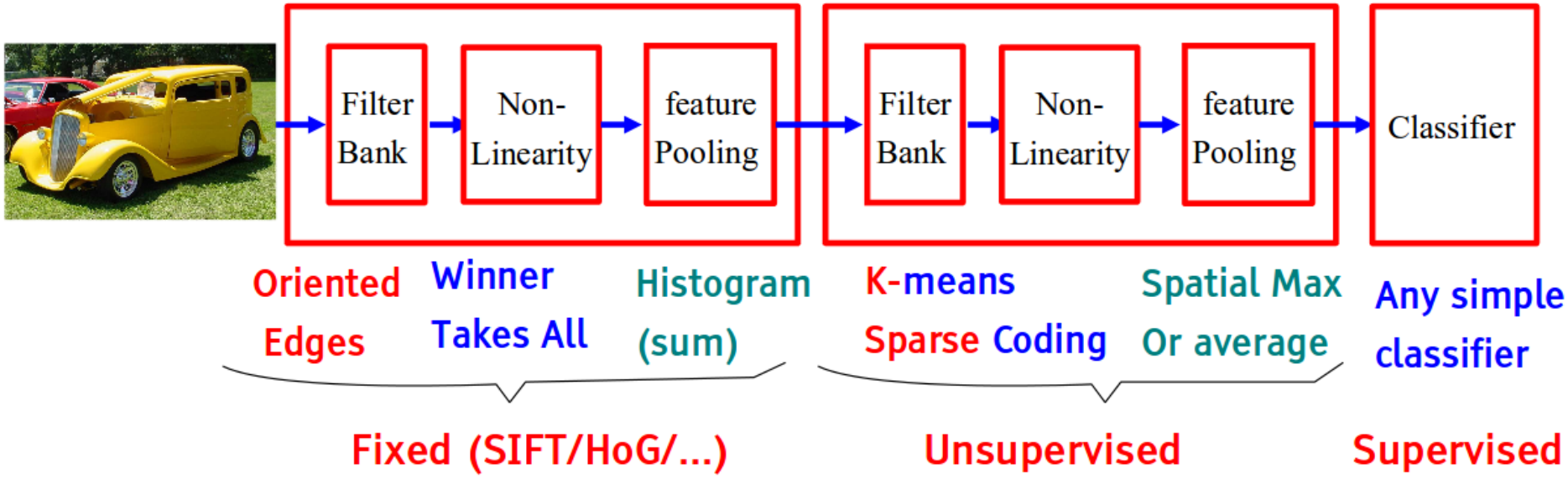
Manifold





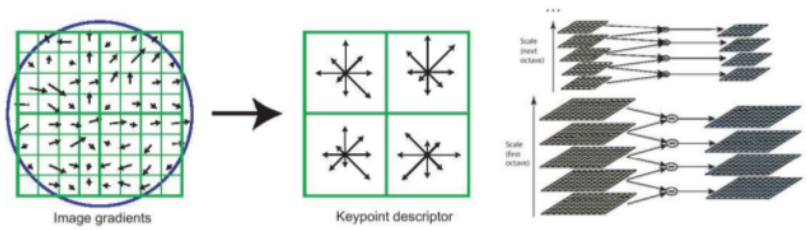
History / "Mainstream" ML (90's – 2011)

- Hand-crafted low-level feature extractor + Trainable (USL) mid-level feature extractor + Trainable (SL) classifier

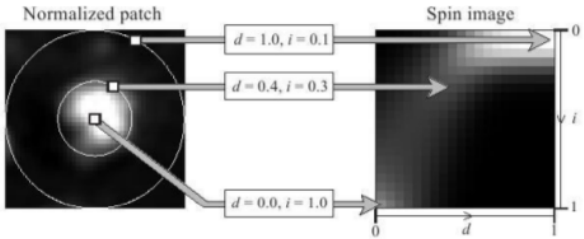


History / “Mainstream” ML

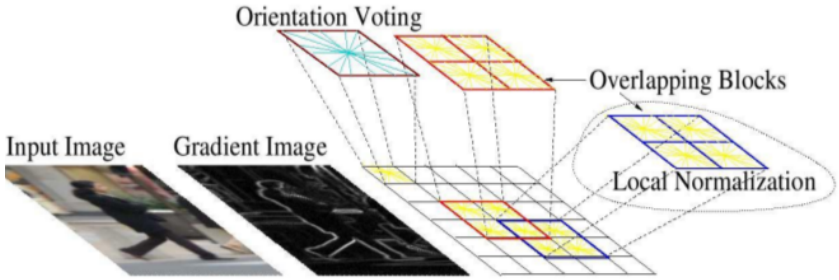
- CV features



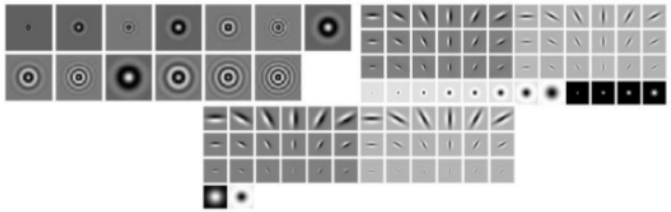
SIFT



Spin image



HoG



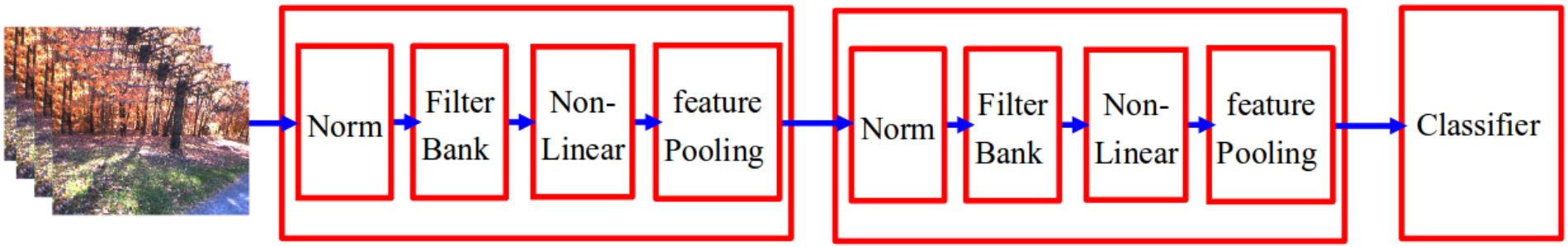
Textons

and many others:

SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, …..

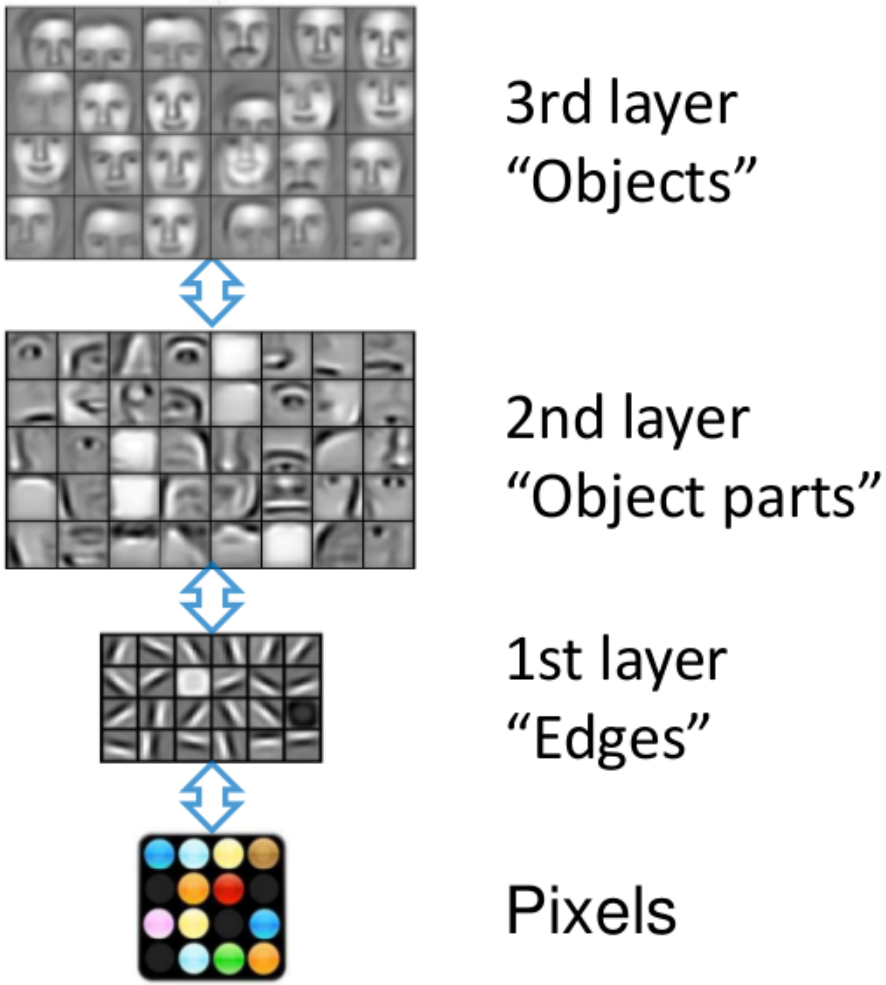
History / Deep ML

- Filter Bank + Non-Linearity = Non-linear embedding in high dimension
- Pooling = contraction, dimensionality reduction, smoothing
- Creating a hierarchy of features

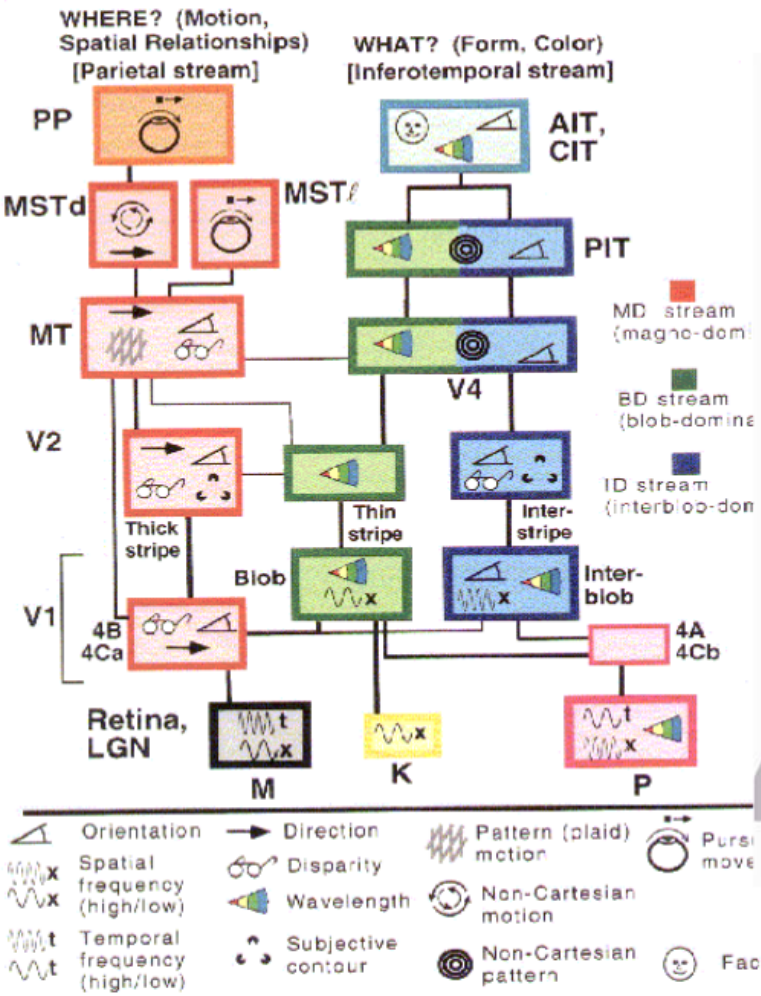


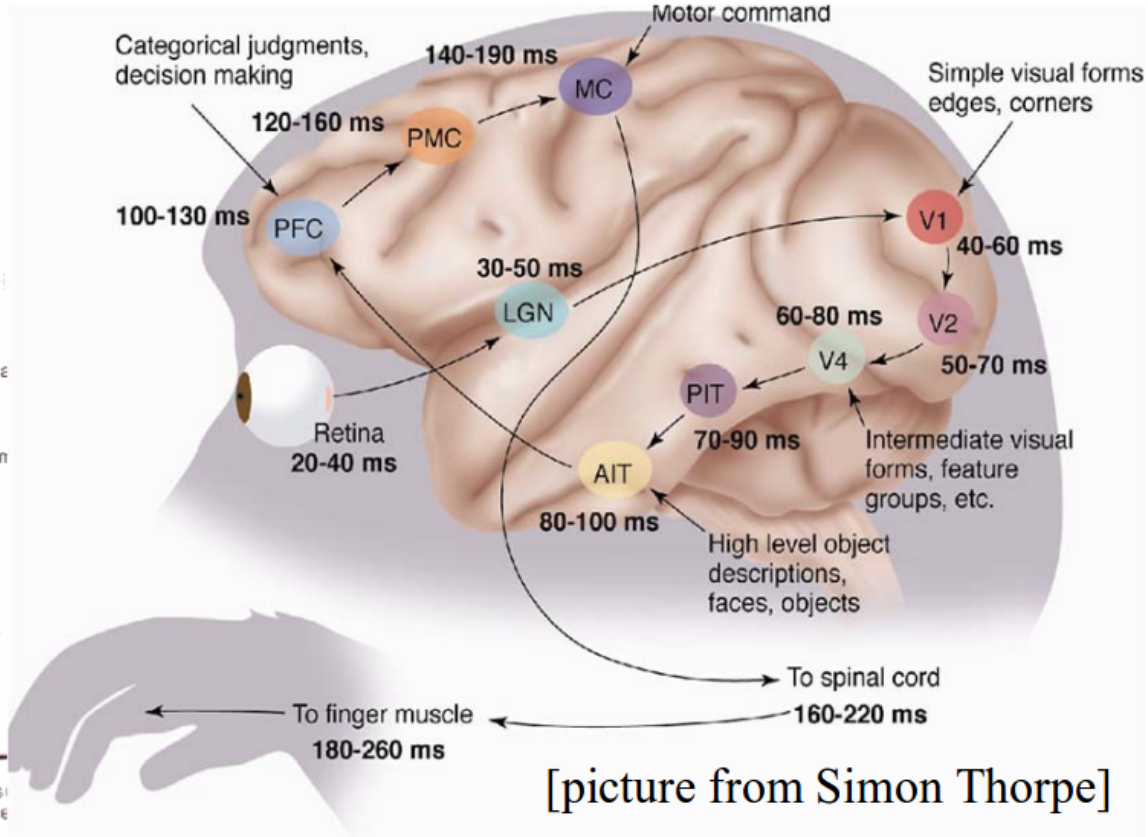
History / Deep ML

- Deep learning (end-to-end learning, feature learning): Trainable nonlinear feature extractor + Trainable classifier
- Deep = Hierarchical representation



Motivation / Biology inspired



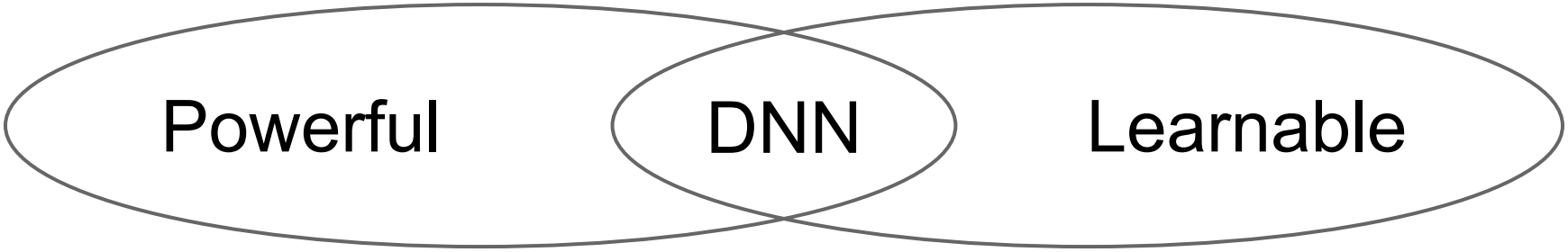


[picture from Simon Thorpe]

[Gallant & Van Essen]

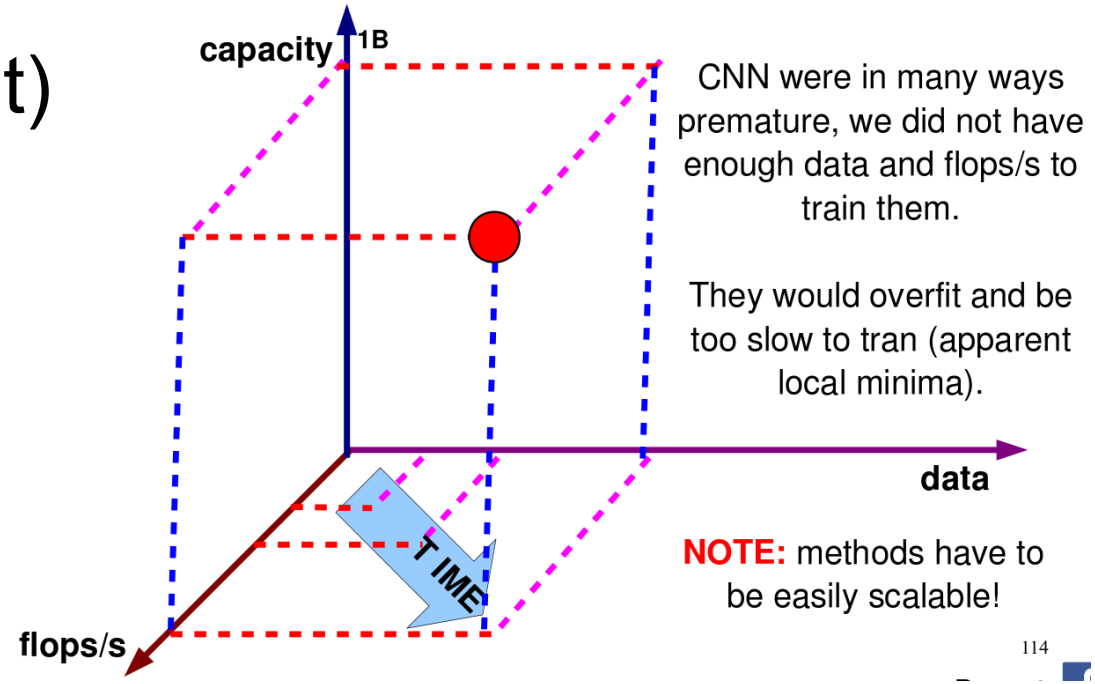
Motivation / Deep learning

- Y. Bengio: *Deep Learning: machine learning algorithms inspired by brains, based on learning multiple levels of representation / abstraction.*
- Deep Neural Networks (DNN)
  - A single neuron can implement boolean logic
  - 2-layers ANN = universal approximator
  - k layers = 2^N size 2-layers ANN (in some cases)



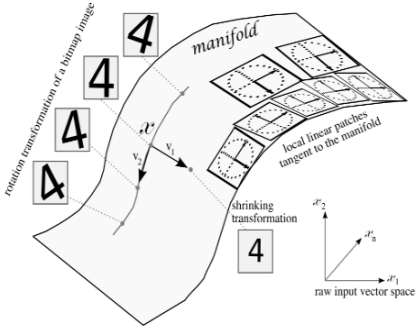
History / DNN

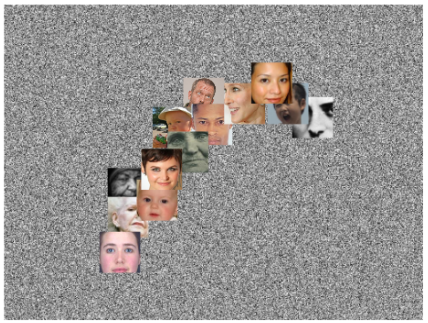
- Network capacity (underfit)
- CPU / GPU
- RAM
- Datasets (overfit)



How to train (find manifolds)?

- Supervised learning
  - Labeled data
  - Knowledge transfer
- Unsupervised learning
  - Weak labeled data
  - Unlabeled data
- Reinforcement learning
  - Unknown target





Sparse coding

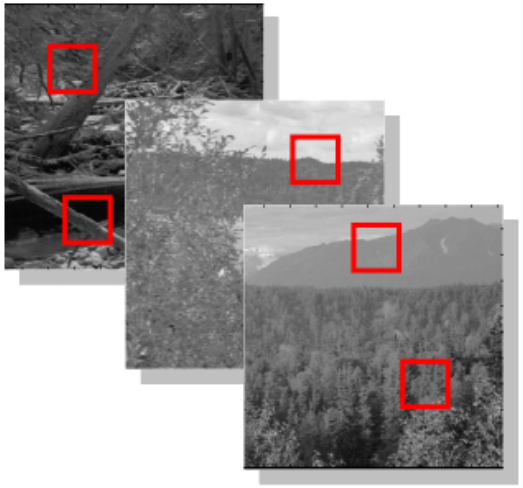
- Olshausen & Field, 1996. Originally developed to explain early visual processing in the brain (edge detection)
- Objective: Given a set of input data vectors learn a dictionary of bases such that:

$$\mathbf{x}_n = \sum_{k=1}^{K} a_{nk} \boldsymbol{\phi}_k,$$

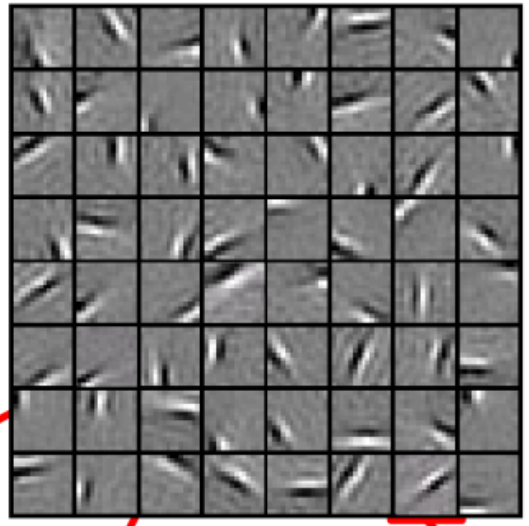
Sparse: mostly zeros

Sparse coding

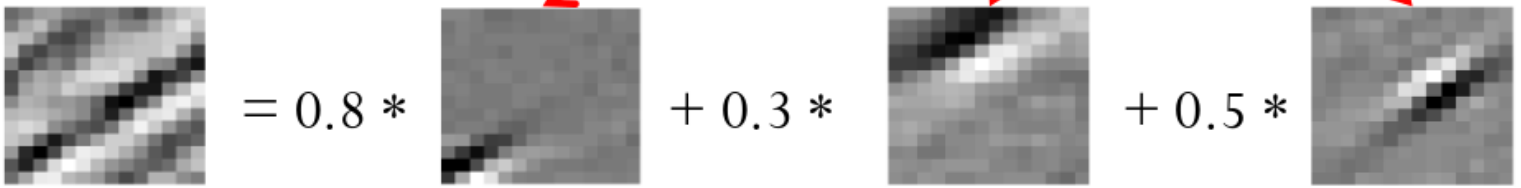
Natural Images



Learned bases: "Edges"



New example

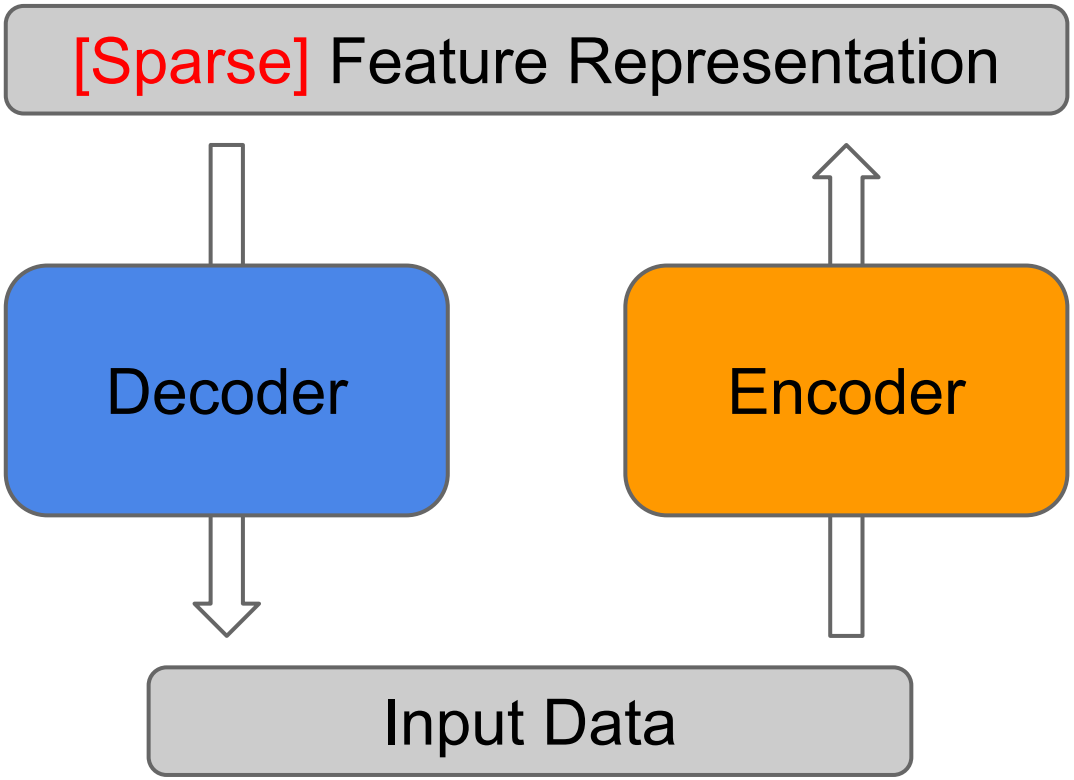


$$x = 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{65}$$

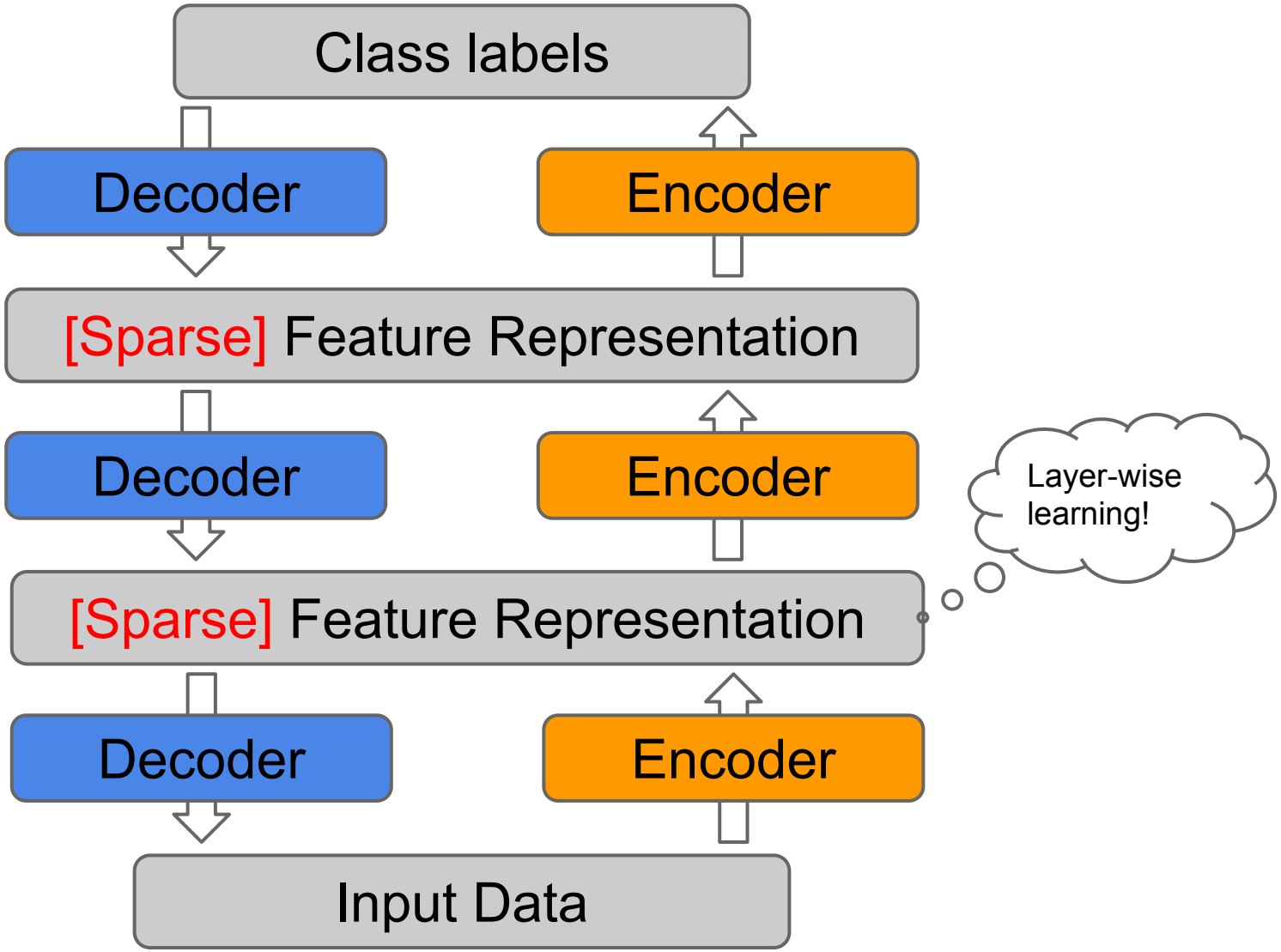
[0, 0, … **0.8**, …, **0.3**, …, **0.5**, …] = coefficients (feature representation)

Slide Credit: Honglak Lee

Autoencoders



Stacked Autoencoders

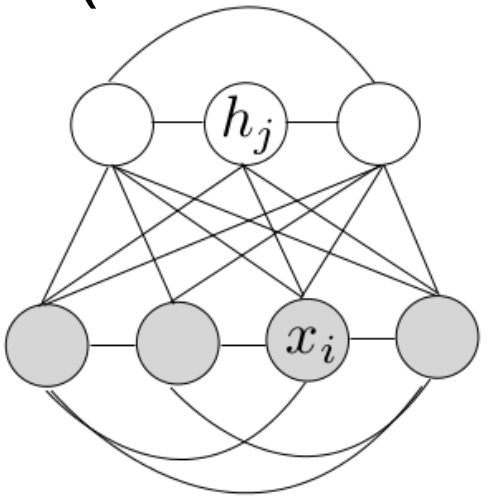


Boltzmann Machines (BM)

- Stochastic Hopfield Networks with hidden units
- Both visible and hidden units are binary
  (can be generalized with Gaussian--Bernoulli model)
- The states of the hidden units form interpretations of the input (visible units)

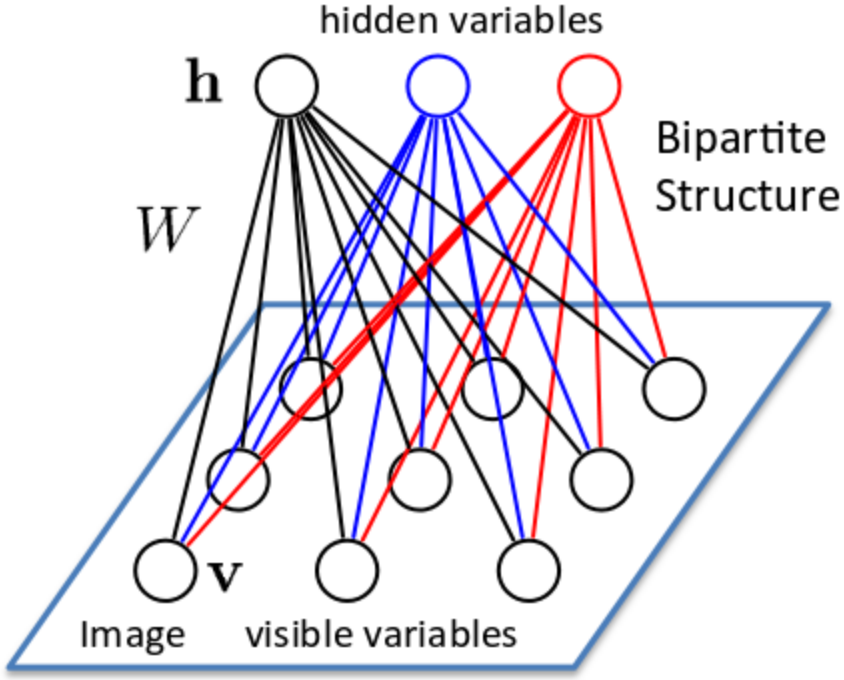
Probability of the joint configuration is given by the Boltzmann distribution:

$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\big(-E(\mathbf{v}, \mathbf{h}; \theta)\big)$$



Restricted Boltzmann Machines (RBM)

- Only one layer of hidden units
- No connections between hidden units
- In an RBM it only takes one step to reach thermal equilibrium when the visible units are clamped



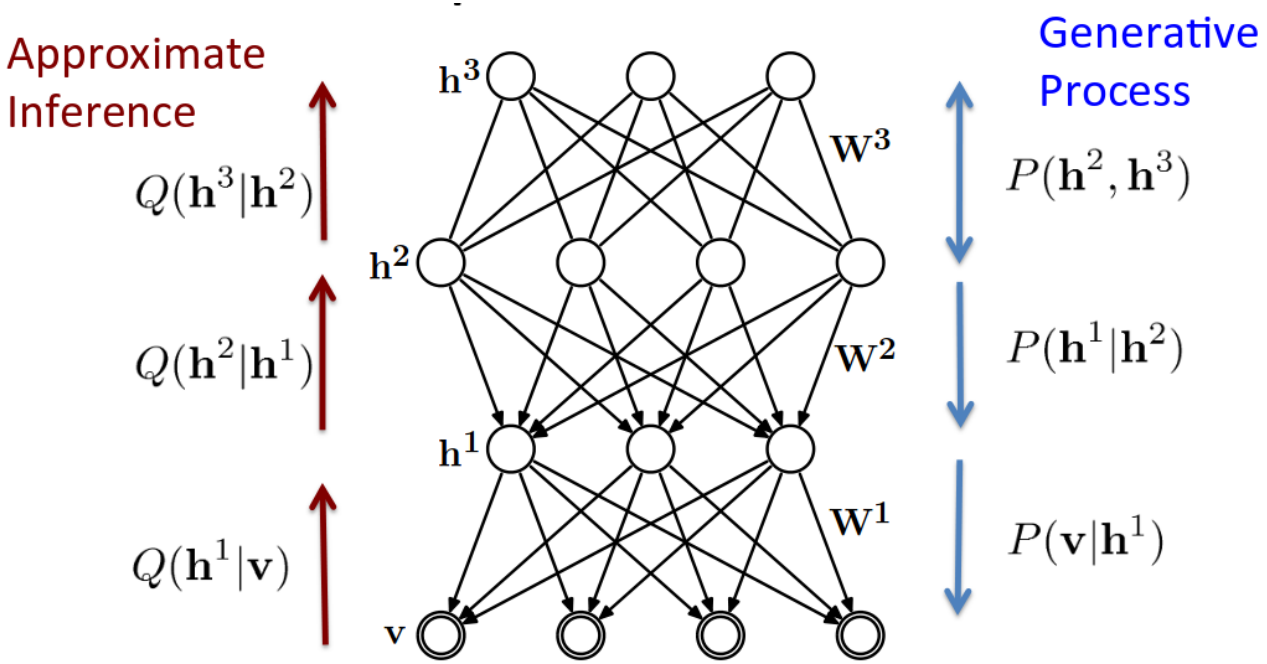
The energy of the joint configuration:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

$\theta = \{W, a, b\}$ model parameters.

Deep Belief Network (DBN)

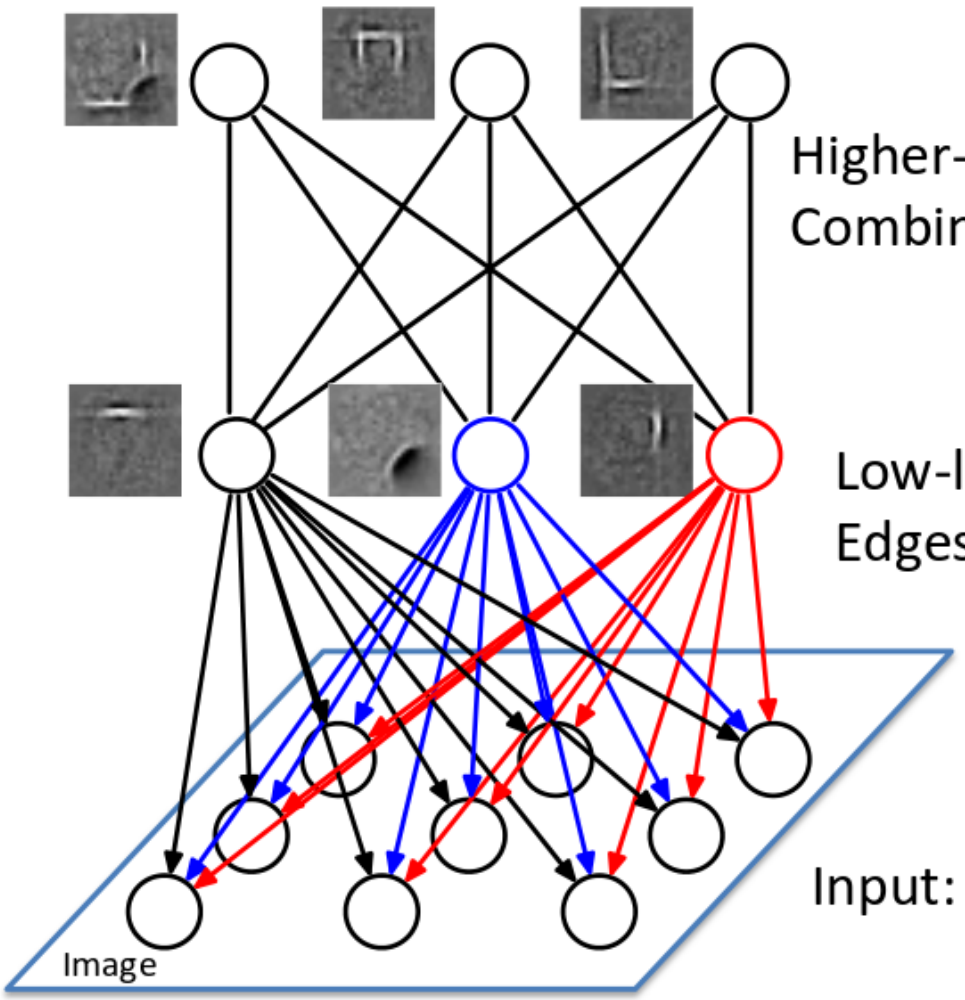
- Stacking RBMs
- Unsupervised layer-wise training



$$Q(\mathbf{h}^t|\mathbf{h}^{t-1}) = \prod_j \sigma\left(\sum_i W^t h_i^{t-1}\right) \qquad P(\mathbf{h}^{t-1}|\mathbf{h}^t) = \prod_j \sigma\left(\sum_i W^t h_i^t\right)$$

Deep Belief Network (DBN)

Internal representations capture higher-order statistical structure



Higher-level features:
Combination of edges

Low-level features:
Edges

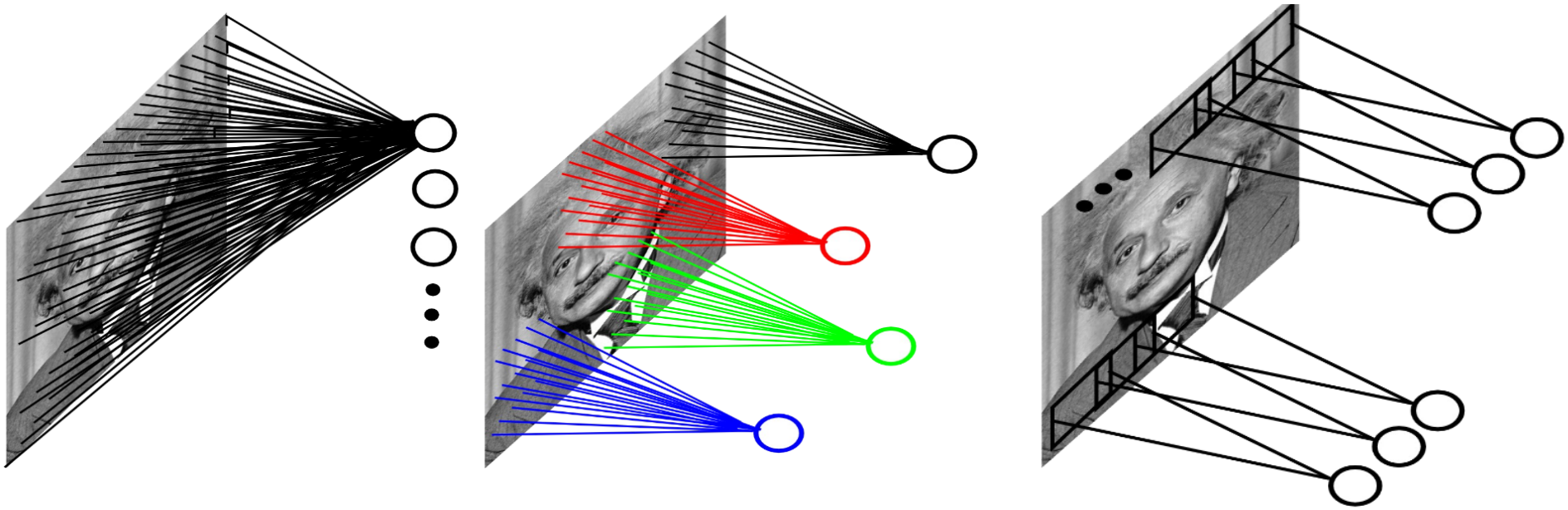
Built from **unlabeled** inputs.

Input: Pixels

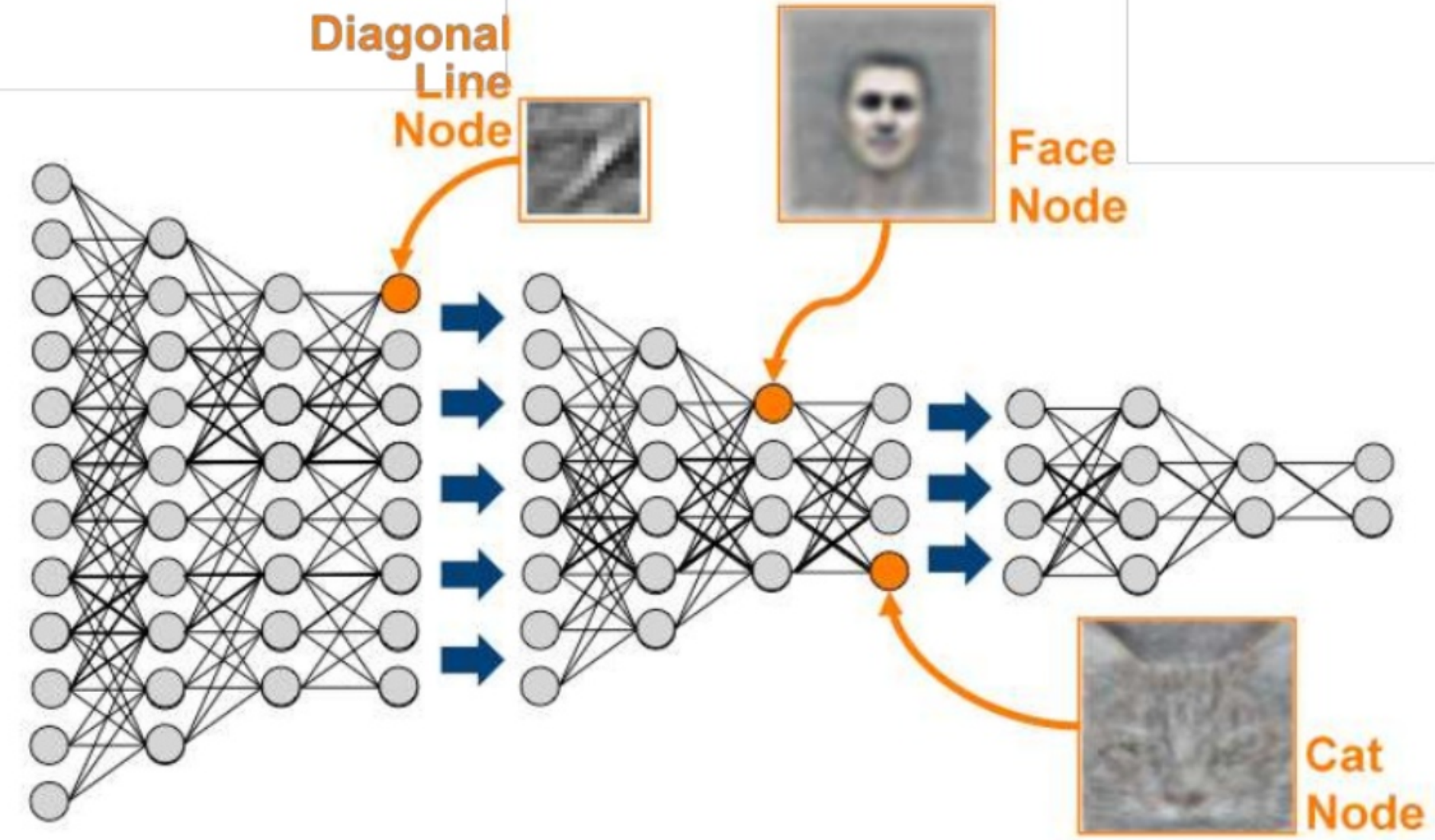
(Hinton et.al. Neural Computation 2006)

Convolutional network

- Fully connected
- Locally connected
- Convolutional = sparse connectivity + parameter sharing

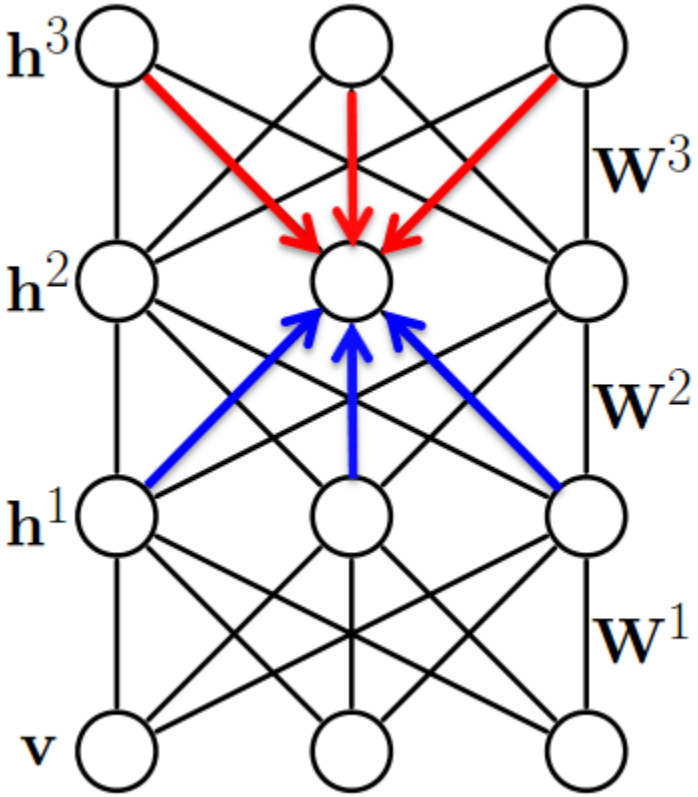


Google Brain (convolutional DBN)



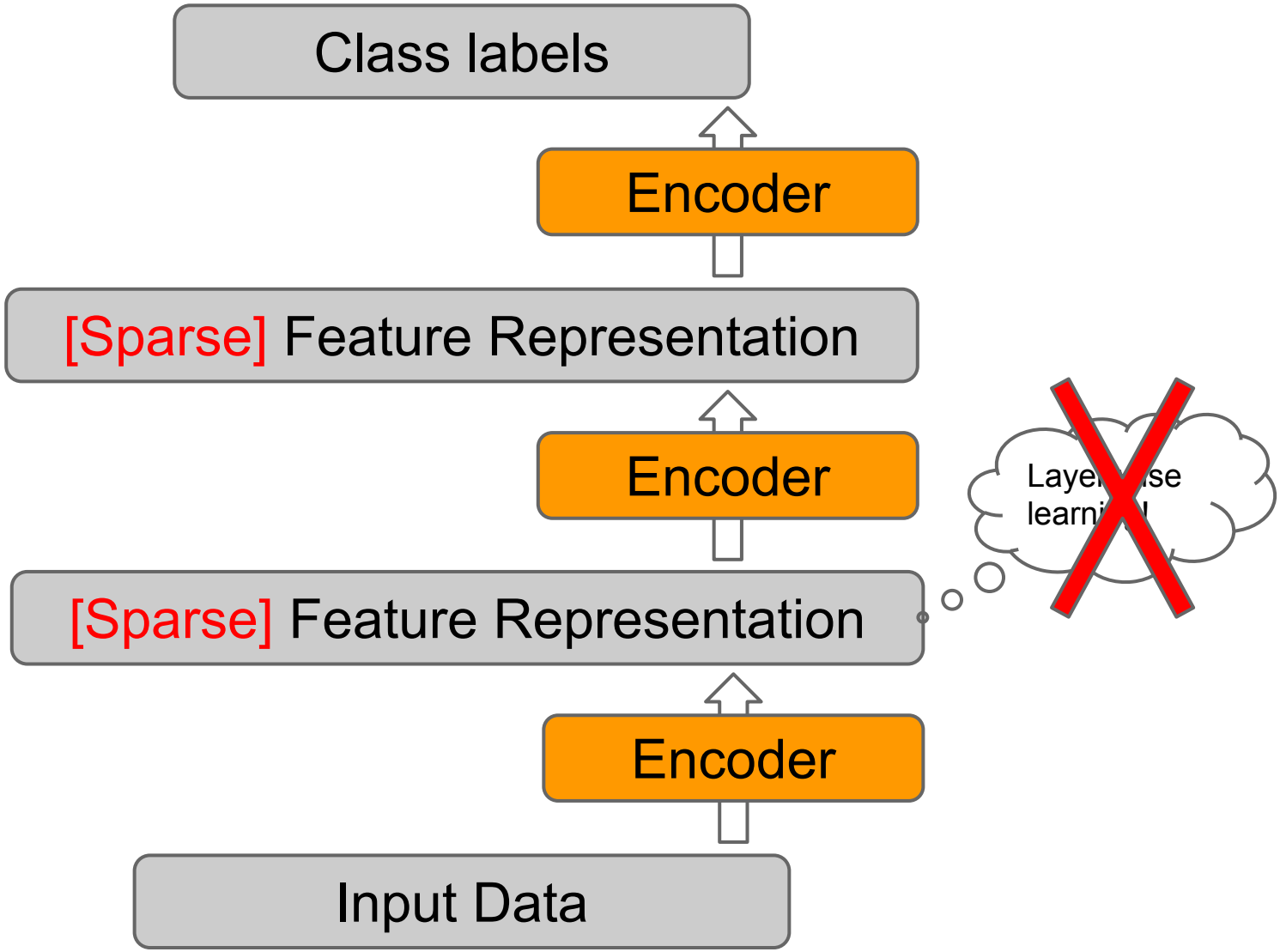
Deep Boltzmann Machines (DBM)

- All connections are undirected (dependencies between hidden variables)
- Joint optimization
- Approximate learning



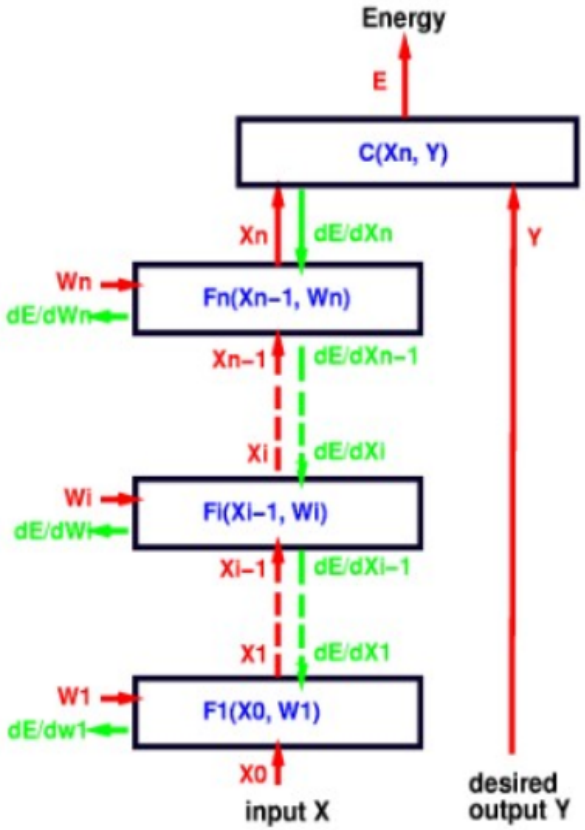
$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

Feed-forward NN



Back-propagation

To compute all the derivatives, we use a backward sweep called the **back-propagation algorithm** that uses the recurrence equation for $\frac{\partial E}{\partial X_i}$



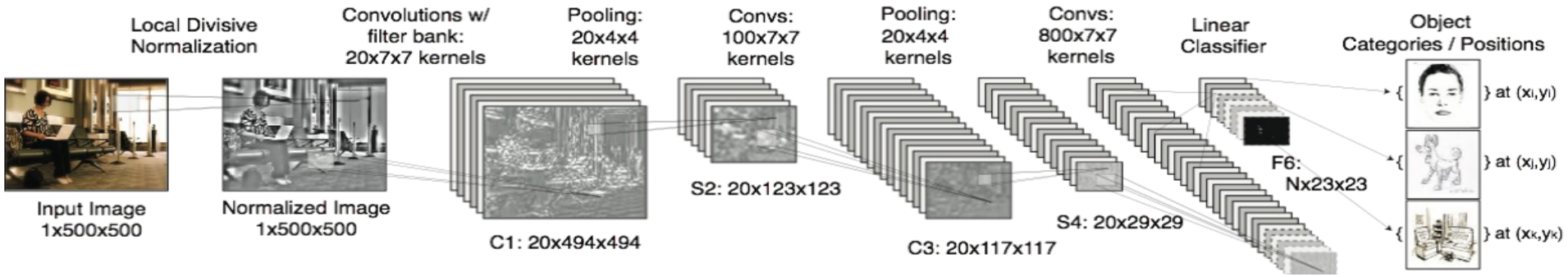
- $\frac{\partial E}{\partial X_n} = \frac{\partial C(X_n, Y)}{\partial X_n}$
- $\frac{\partial E}{\partial X_{n-1}} = \frac{\partial E}{\partial X_n} \frac{\partial F_n(X_{n-1}, W_n)}{\partial X_{n-1}}$
- $\frac{\partial E}{\partial W_n} = \frac{\partial E}{\partial X_n} \frac{\partial F_n(X_{n-1}, W_n)}{\partial W_n}$
- $\frac{\partial E}{\partial X_{n-2}} = \frac{\partial E}{\partial X_{n-1}} \frac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial X_{n-2}}$
- $\frac{\partial E}{\partial W_{n-1}} = \frac{\partial E}{\partial X_{n-1}} \frac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial W_{n-1}}$
- ....etc, until we reach the first module.
- we now have all the $\frac{\partial E}{\partial W_i}$ for $i \in [1, n]$.

Back-propagation

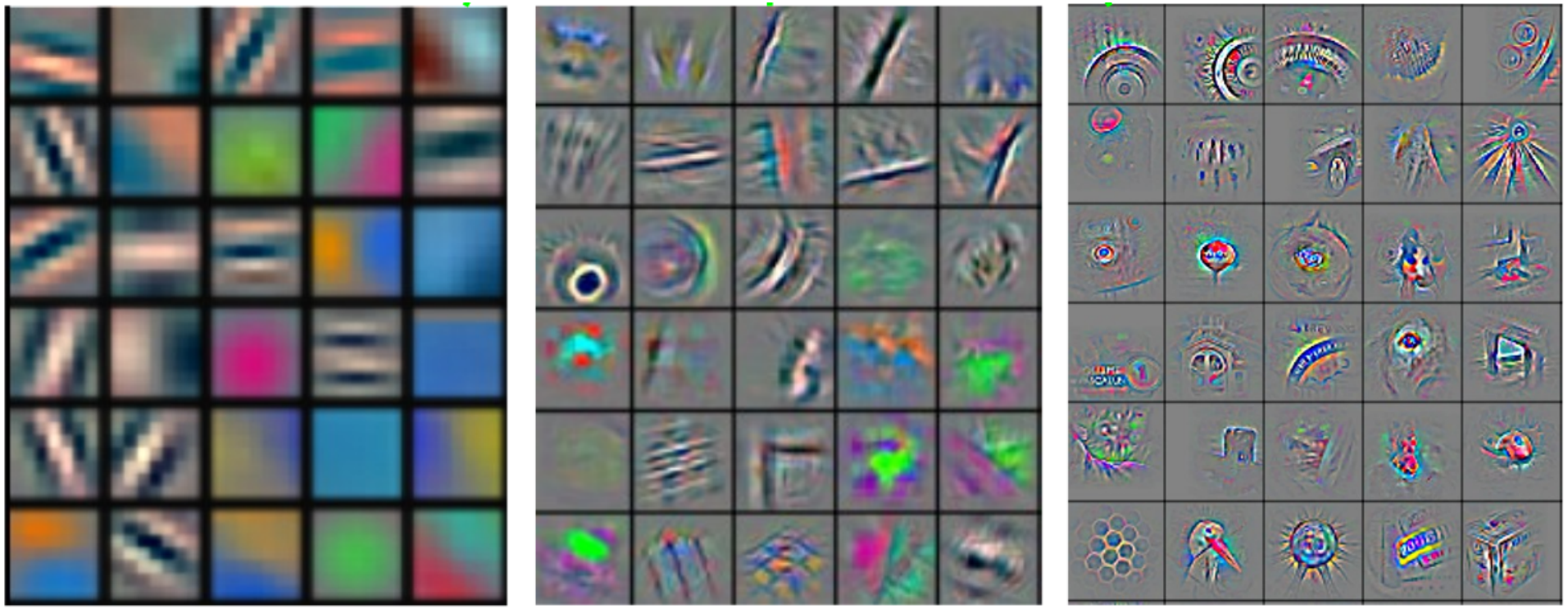
- Any connection is permissible
  - Networks with loops must be “unfolded in time”.
- Any module is permissible
  - As long as it is continuous and differentiable almost everywhere with respect to the parameters, and with respect to non-terminal inputs.
- Supervised learning is non-convex
  - Local minimas.
  - Saddle points.
- Back Propagation + SGD properties

Convolutional NN (CNN)

- Hubel & Wiesel 1962: simple cells detect local features + complex cells “pool” the outputs of simple cells within a retinotopic neighborhood
- Fukushima 1974-1982: Cognitron & Neocognitron
- LeCun et al. 1989-1998: LeNet



CNN / Hierarchical representation



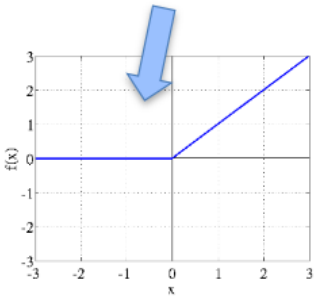
Convolutional NN / Sparse coding

Deep Sparse Rectifier Neural Networks

(Glorot,Bordes and Bengio AISTATS 2011), following up on (Nair & Hinton 2010) softplus RBMs

Neuroscience motivations

Leaky integrate-and-fire model

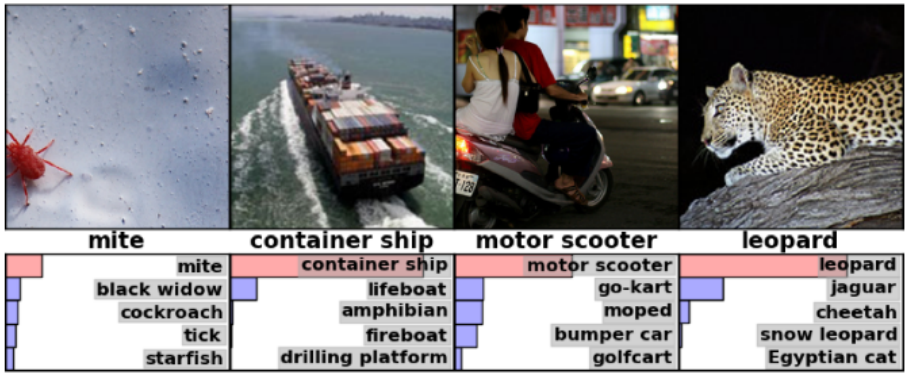


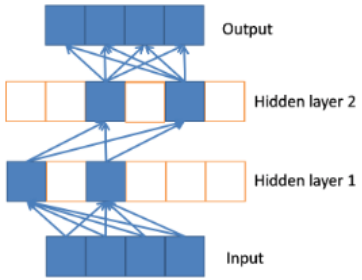
Rectifier

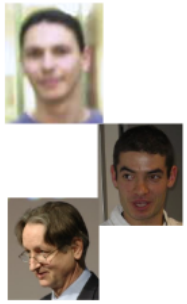
f(x)=max(0,x)

Machine learning motivations

- Sparse representations
- Sparse gradients
- *Trains deep nets even w/o pretraining*





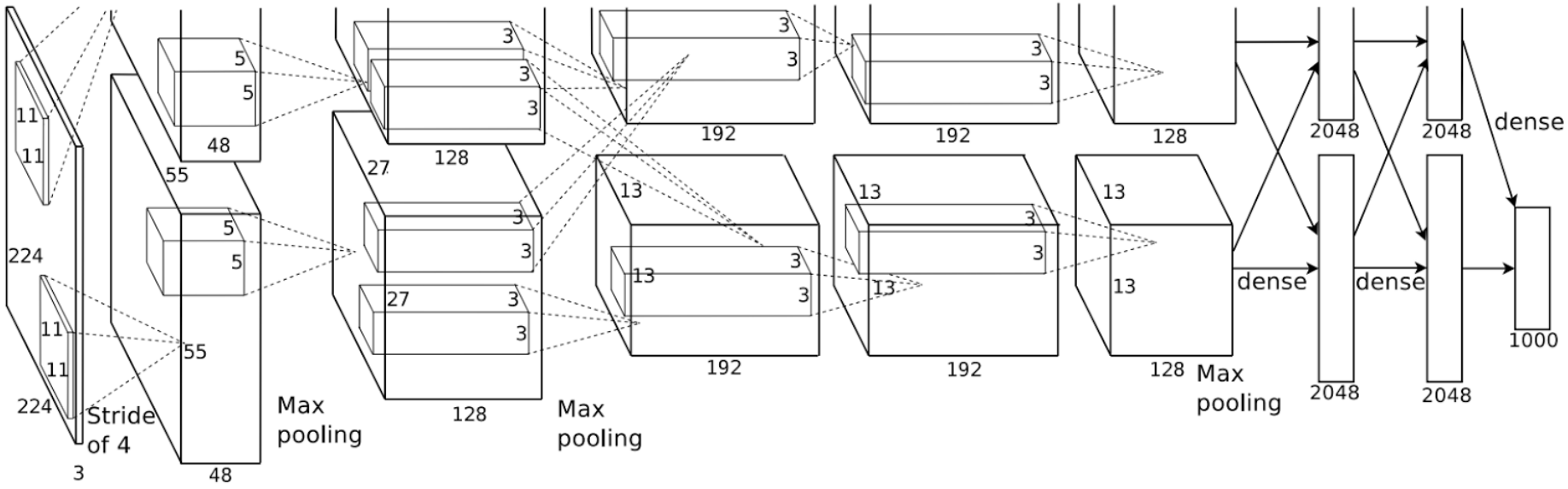


Outstanding results by Krizhevsky et al 2012 killing the state-of-the-art on ImageNet 1000:

	1st choice	Top-5
2nd best		27% err
Previous SOTA	45% err	26% err
Krizhevsky et al	37% err	15% err

Convolutional NN / AlexNet

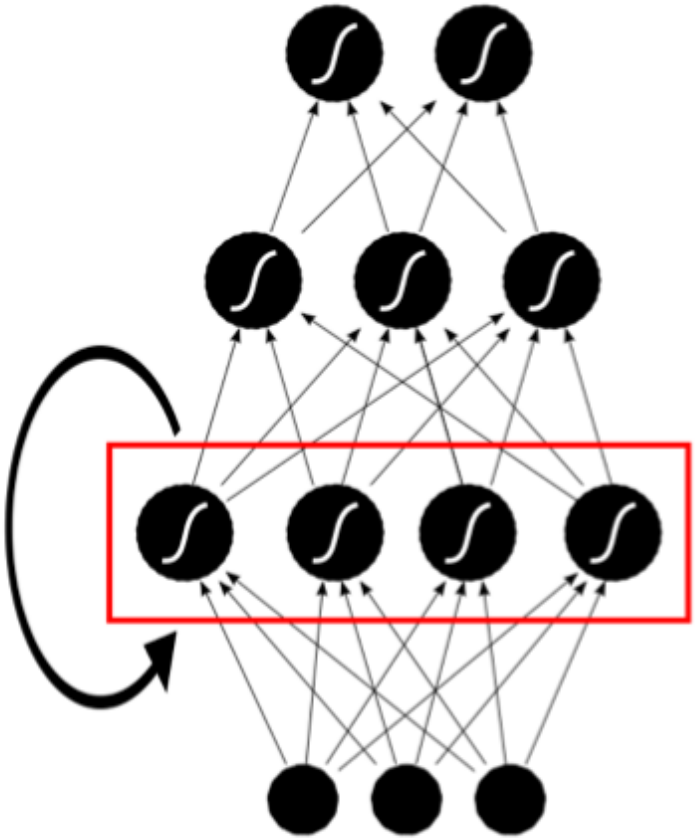
- Won the 2012 ImageNet LSVRC [Krizhevsky, Sutskever, Hinton 2012]
  - Method: large convolutional net
  - 650K neurons, 832M synapses, 60M parameters
  - Trained with backprop on GPU
  - Error rate: 15% (top 5), previous state of the art: 25% error



Convolutional NN / ConvNet

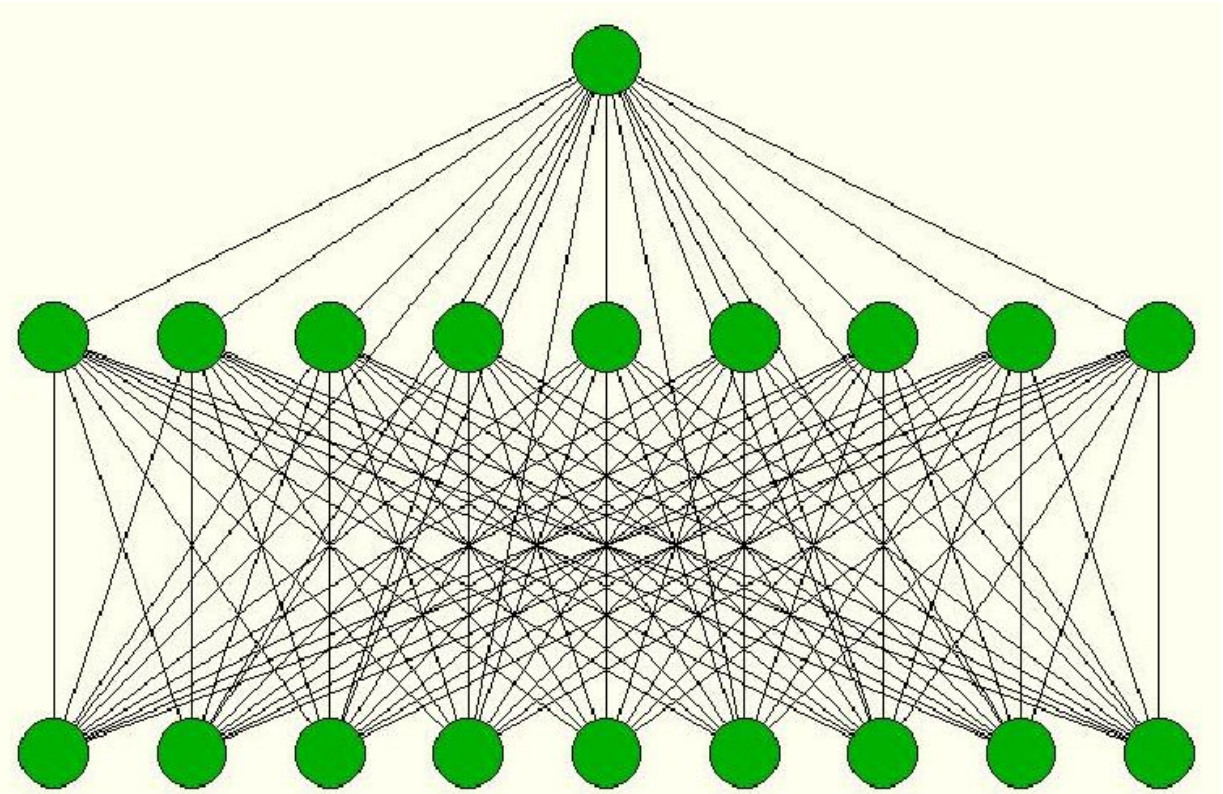
- Scene parse (Farabet et al. ICML 2012, PAMI 2013)

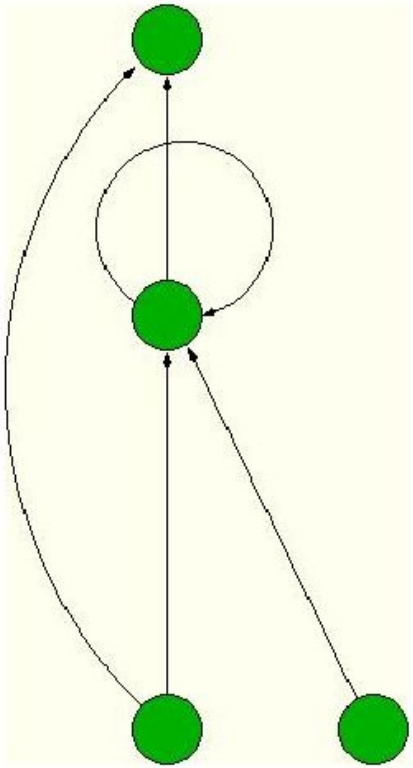
RNN



- Like feedforward networks except that one or more layers is connected to itself
- Self connections allow the network to build an internal representation of past inputs
- In effect they give the network memory

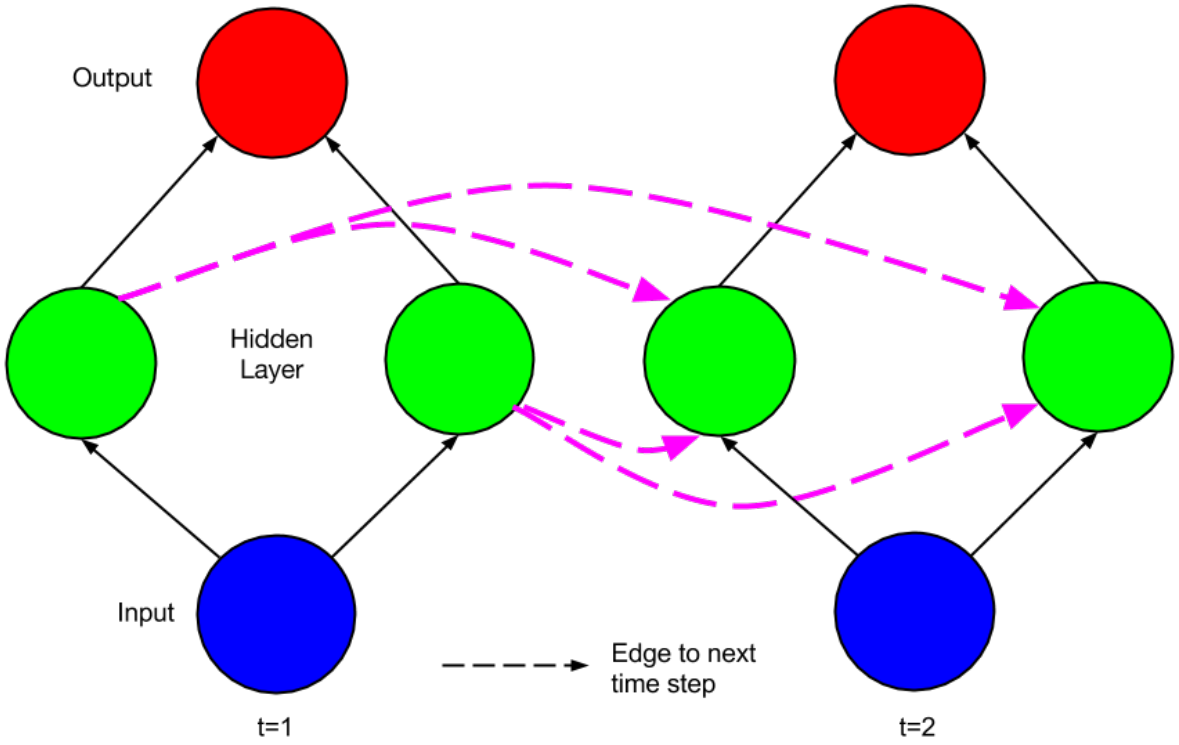
RNN / Parity problem





RNN / Training

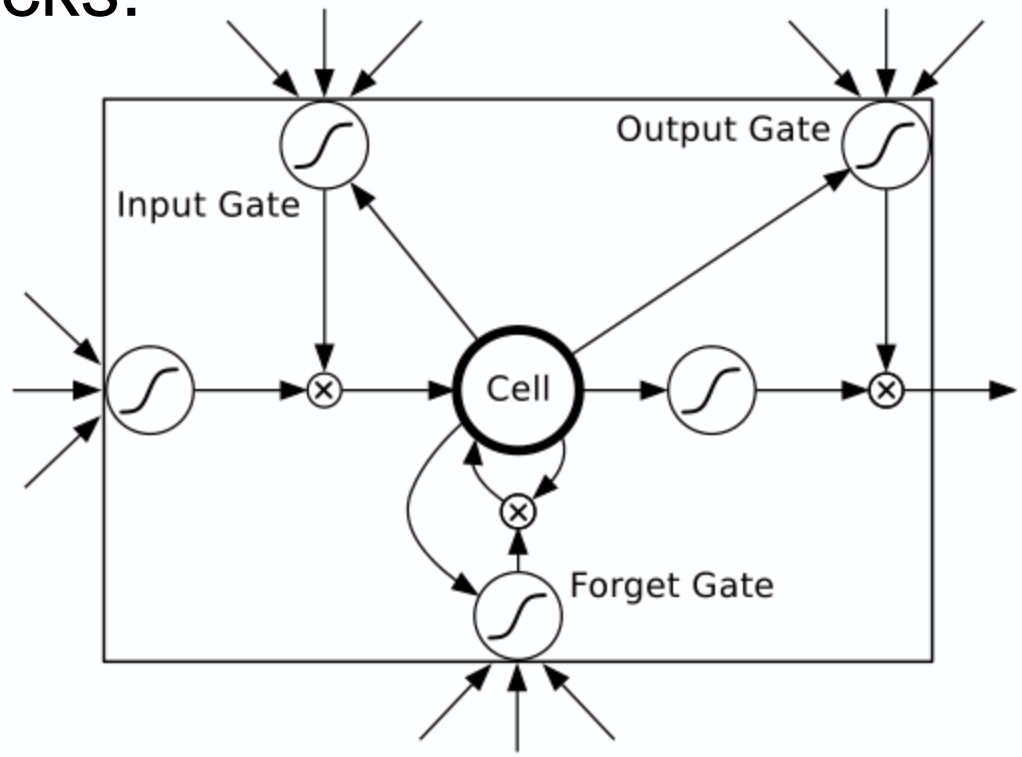
- Unfolding (shared weights) + Back-prop
- Vanishing / exploding gradients



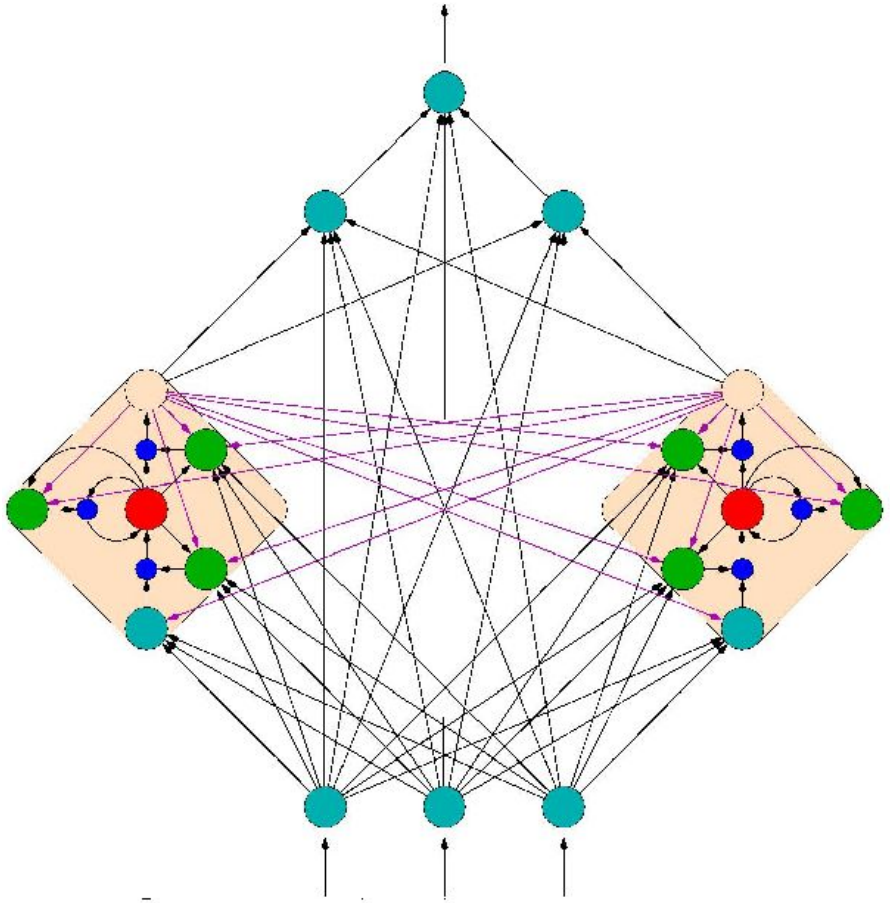
Long Short-Term Memory (LSTM)

S. Hochreiter and J. Schmidhuber, “Long Short-term Memory” Neural Computation 1997

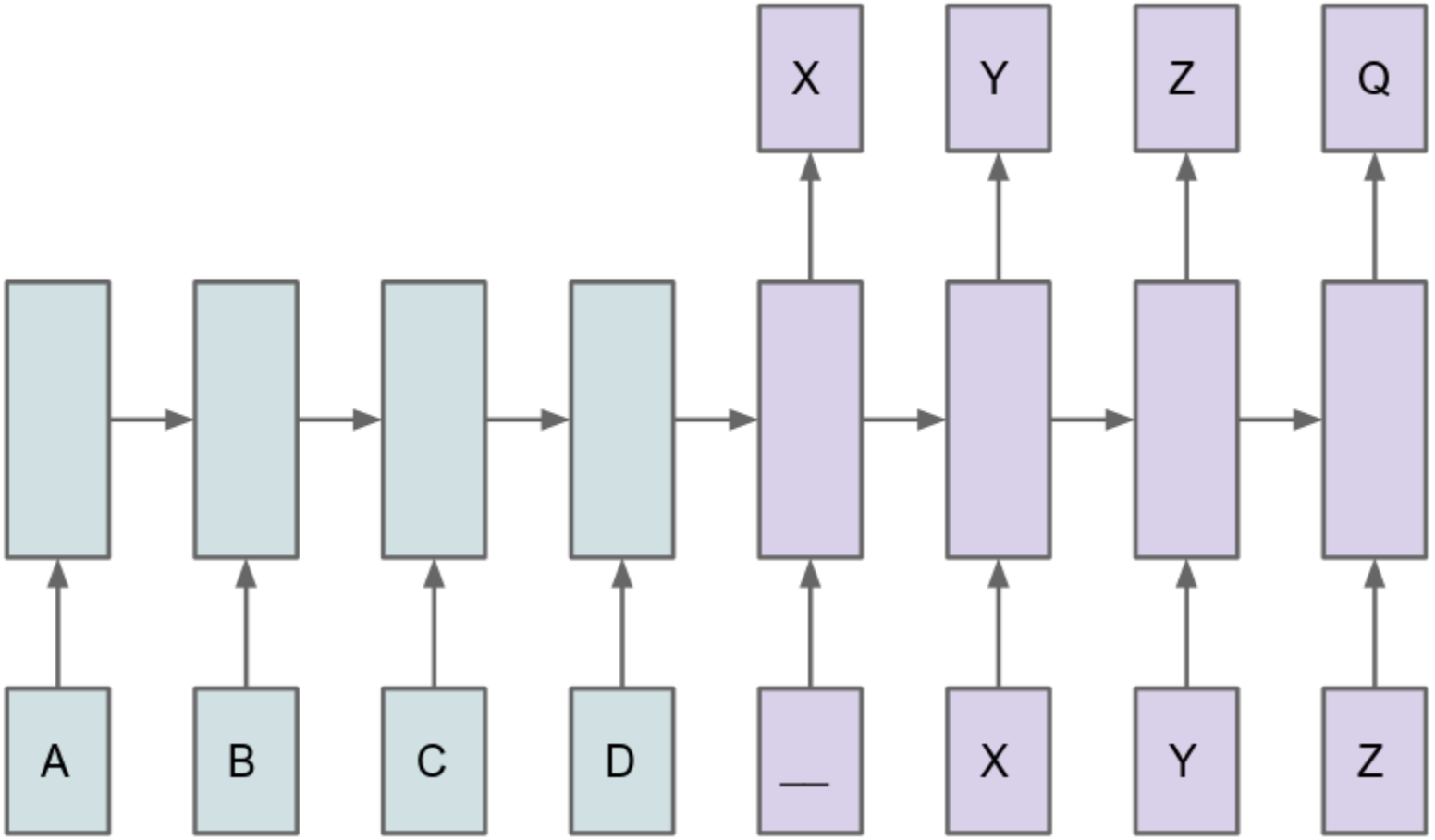
- Constant error flow
- LSTM Memory blocks: shared gates



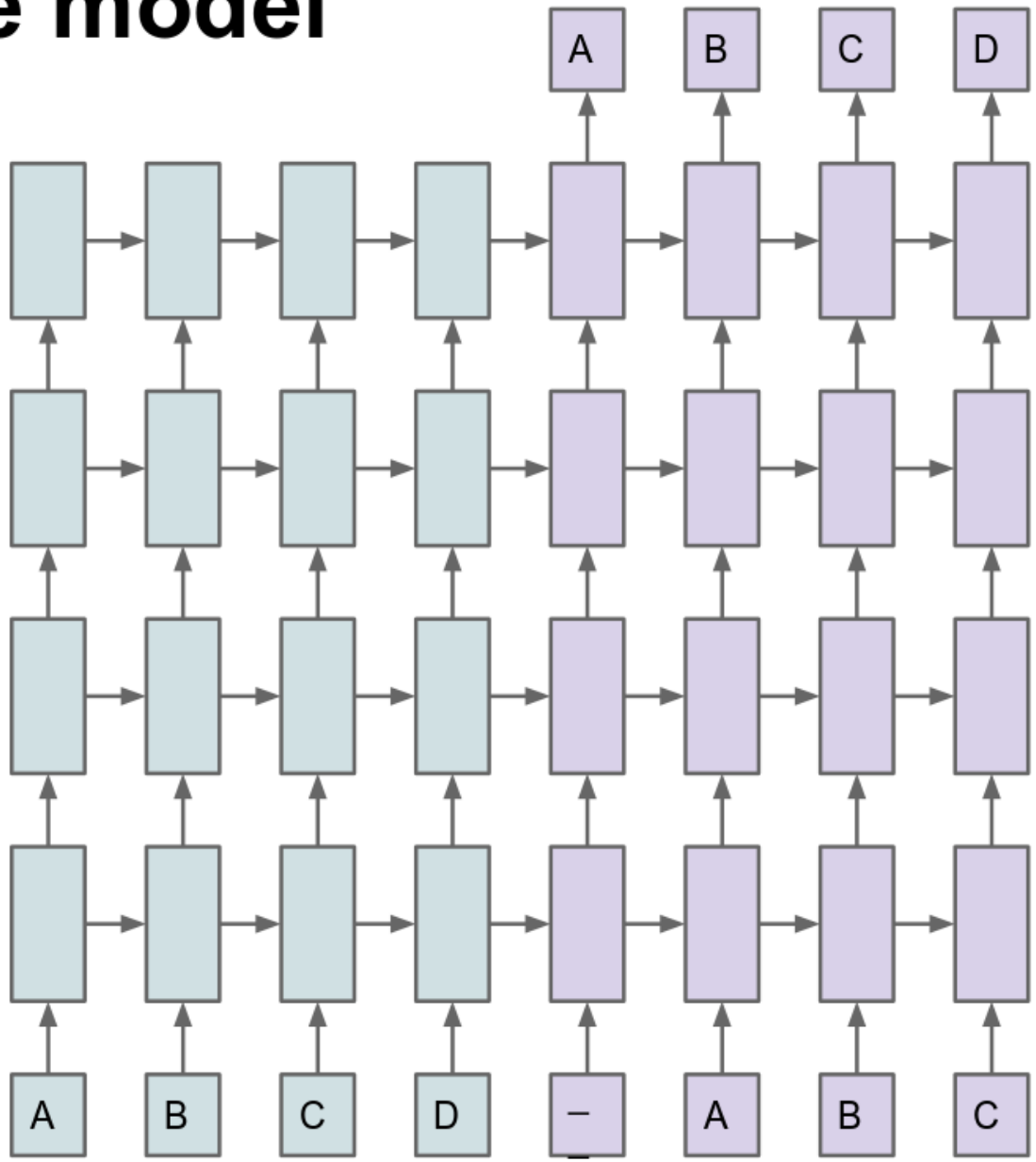
Long Short-Term Memory (LSTM)



LSTM / NLP



The model



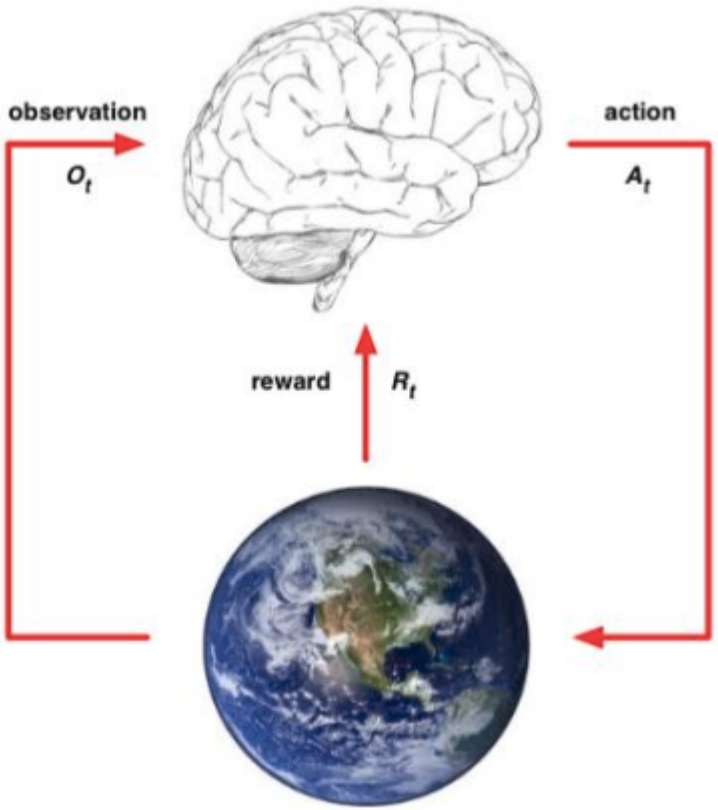
80k softmax by
1000 dims
This is very big!

1000 LSTM cells
2000 dims per
timestep

2000 x 4 =
8k dims per
sentence

160k vocab in
input language

Reinforcement Learning (RL)



RL / Multi-modal models

Images

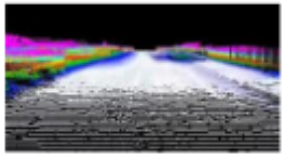
Text & Language

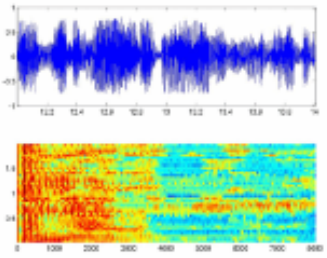
REUTERS

AP Associated Press

Video

Laser scans



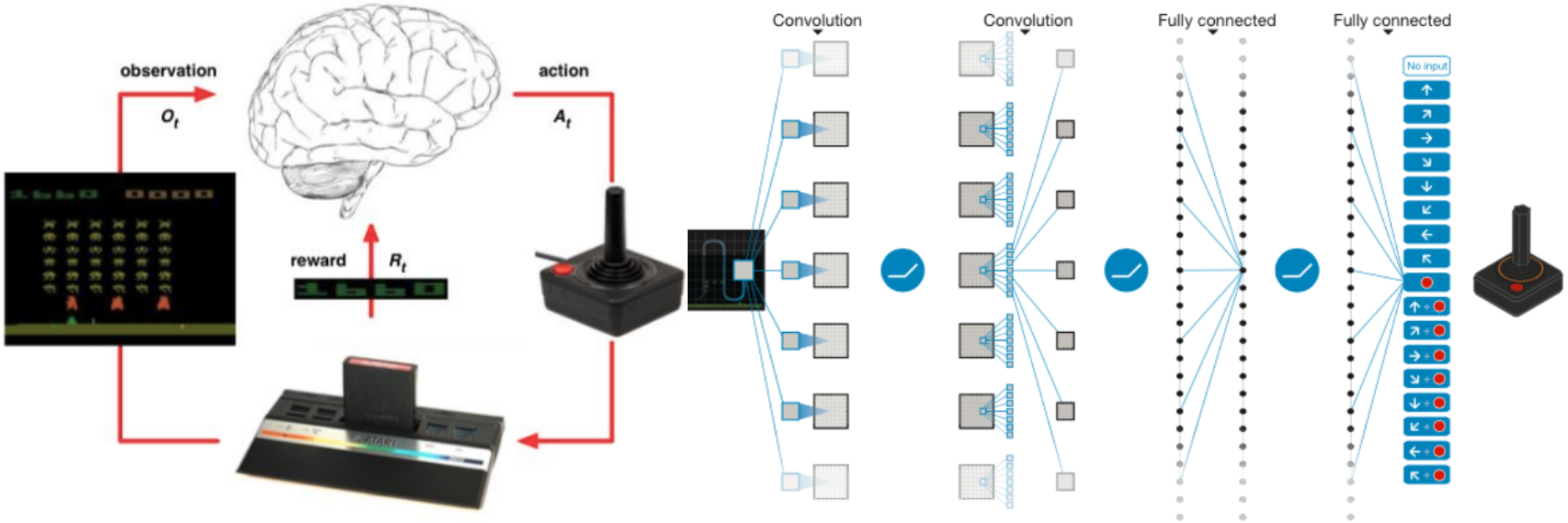


Speech & Audio

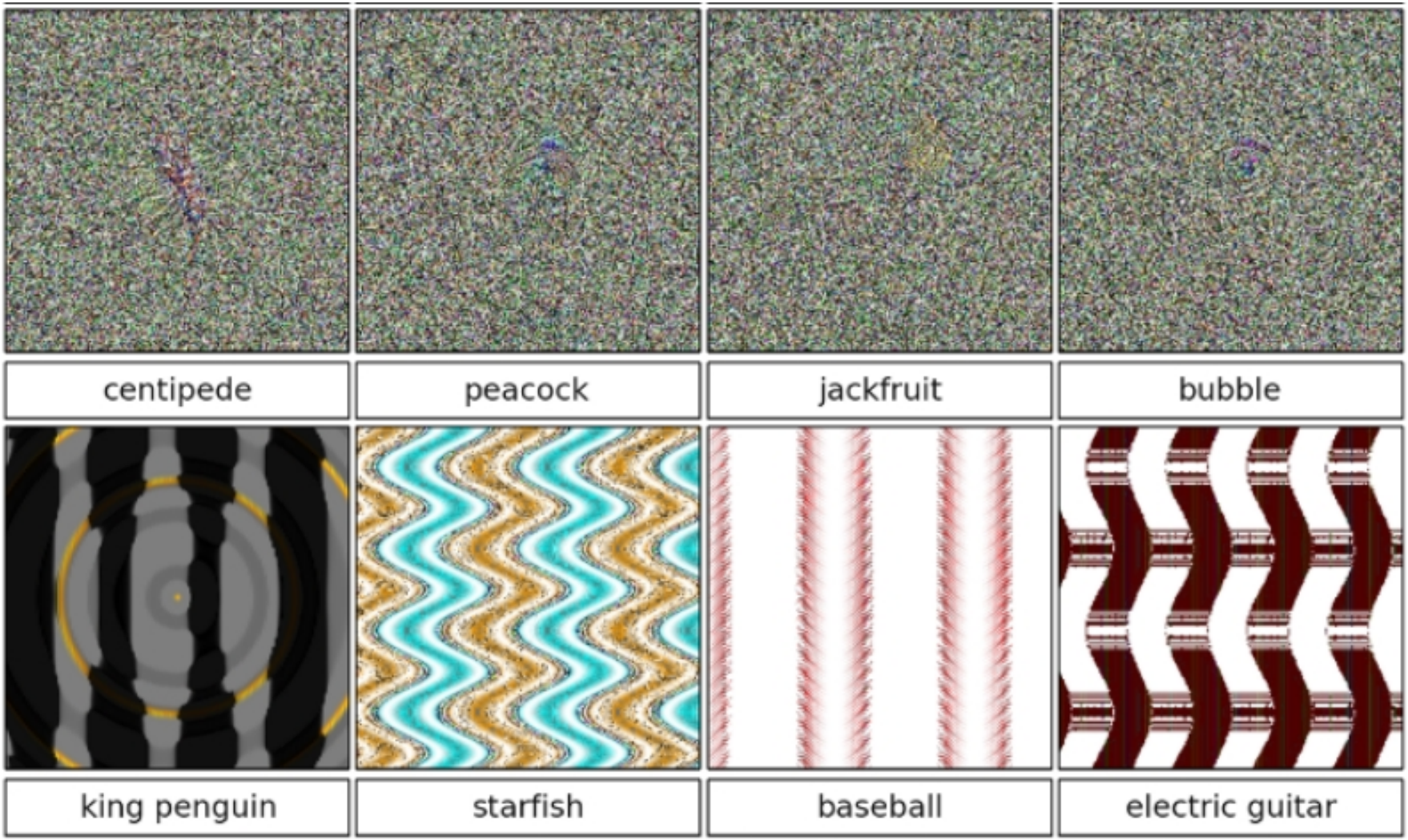
Time series data

RL / Atari Experiment

- Q-learning (reinforcement learning)
- CNN



There is no silver bullet

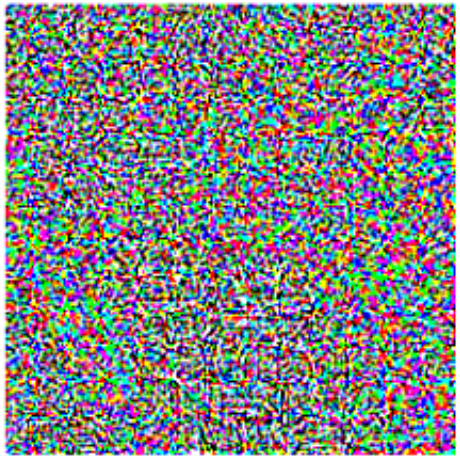


There is no silver bullet

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

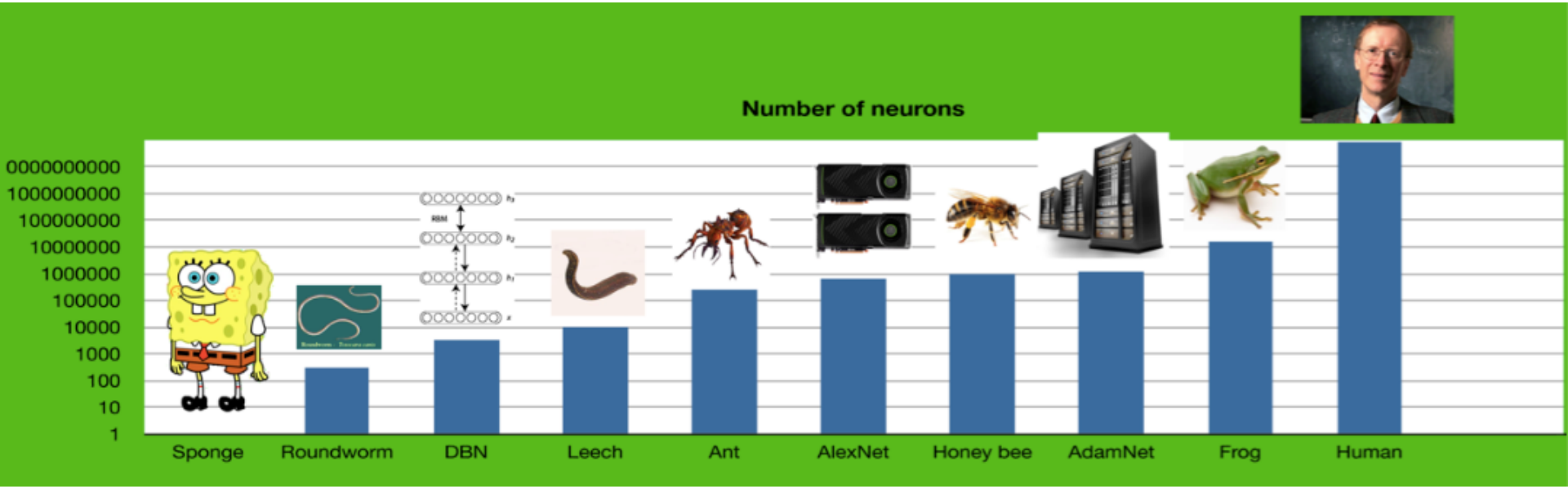
"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Long way to go...

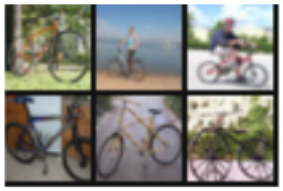


Learning to learn

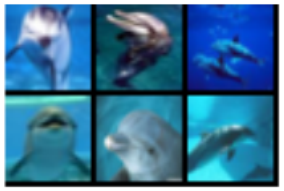
Background Knowledge

Millions of unlabeled images

Some labeled images



Bicycle



Dolphin

Elephant

Tractor

Learn to Transfer Knowledge

Learn novel concept from one example

Test:

Need more info! / Names

Alex Graves

Google DEEPMIND

UNIVERSITY OF TORONTO

Geoffrey E. Hinton

Yann LeCun

Microsoft

NETFLIX

Université de Montréal

Baidu 百度

NYU

TECHNISCHE UNIVERSITÄT MÜNCHEN

Stanford

Andrew Ng

Jürgen Schmidhuber

Yoshua Bengio

Need more info! / URLs

- Neural Networks and Deep Learning http://neuralnetworksanddeeplearning.com/index.html
- Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition http://cs231n.github.io/
- Caffe deep learning framework http://caffe.berkeleyvision.org/
- Deep Learning for Computer Vision https://sites.google.com/site/deeplearningcvpr2014/
- Unsupervised Feature Learning and Deep Learning Tutorial http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial
- Recurrent Neural Networks http://people.idsia.ch/~juergen/rnn.html

Questions?

Thank you!