A FEW THINGS ABOUT

DEEP LEARNING

Giovani Chiachia
giovani.chiachia@gmail.com
Instituto de Computação
Unicamp

November 2013

WIRED

“The Man Behind the Google Brain: Andrew Ng and the Quest for the New AI”

www.wired.com/wiredenterprise/2013/05/neuro-artificial-intelligence/all/

WIRED

"The Man Behind the Google Brain: Andrew Ng and the Quest for the New AI"

www.wired.com/wiredenterprise/2013/05/neuro-artificial-intelligence/all/

The New York Times

How Many Computers to Identify a Cat? 16,000

www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html

WIRED

"The Man Behind the Google Brain: Andrew Ng and the Quest for the New AI"

www.wired.com/wiredenterprise/2013/05/neuro-artificial-intelligence/all/

The New York Times

How Many Computers to Identify a Cat? 16,000

www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html

MIT Technology Review

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

www.technologyreview.com/featuredstory/513696/deep-learning/

BREAKTHROUGH RESULTS

BREAKTHROUGH RESULTS

Object Recognition

Images from CIFAR-10 dataset: www.cs.toronto.edu/~kriz/cifar.html

BREAKTHROUGH RESULTS

Object Recognition

Why is it so hard?

Images from CIFAR-10 dataset: www.cs.toronto.edu/~kriz/cifar.html

BREAKTHROUGH RESULTS

Object Recognition

IMAGENET ILSVRC2012

Team name	Error (5 guesses)	Description
SuperVision	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	0.16422	Using only supplied training data
ISI	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.

www.image-net.org/challenges/LSVRC/2012/results.html

BREAKTHROUGH RESULTS

Object Recognition

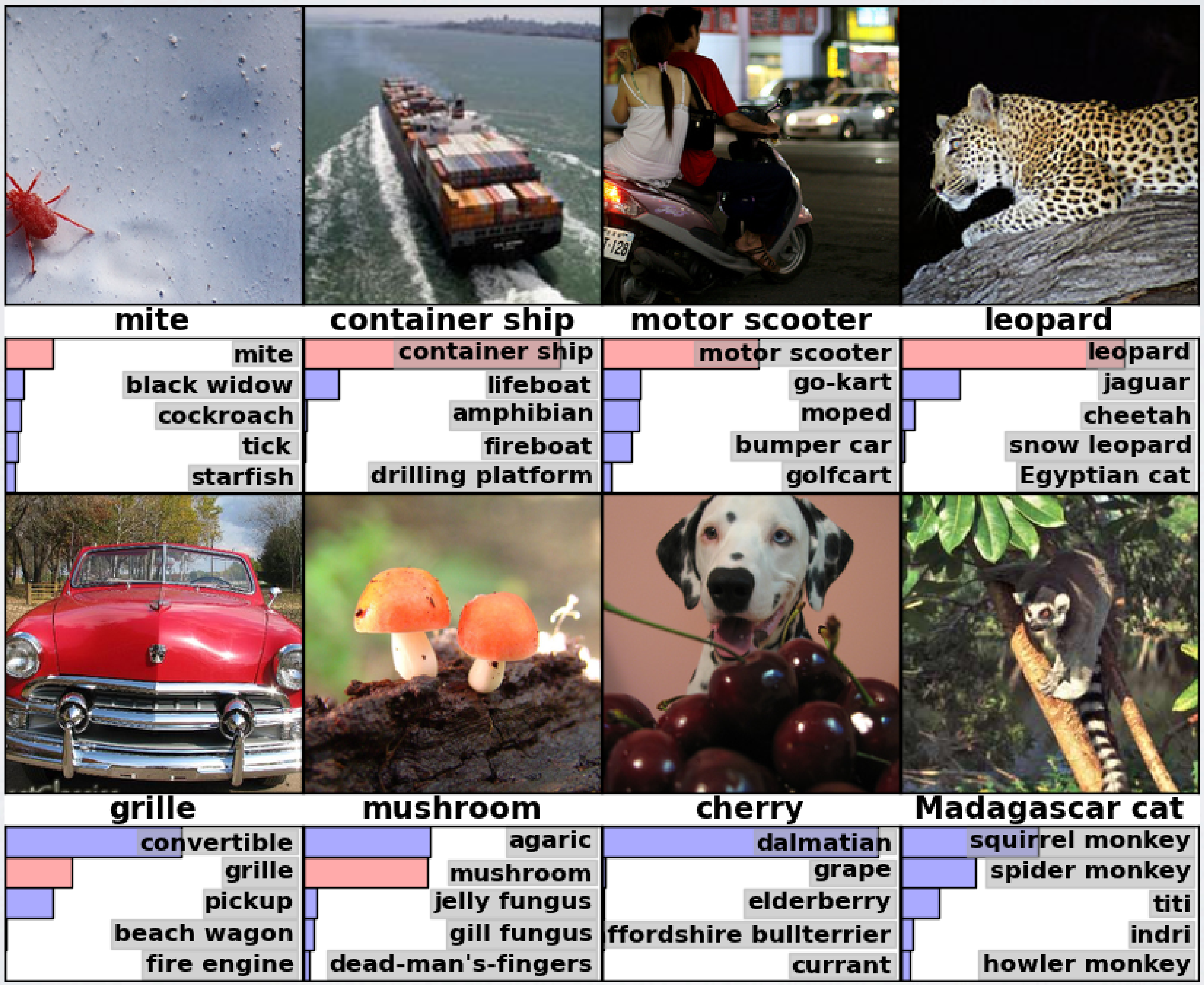
IMAGENET ILSVRC2012

Team name	Error (5 guesses)	Description
SuperVision	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	0.16422	Using only supplied training data
ISI	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.

www.image-net.org/challenges/LSVRC/2012/results.html

BREAKTHROUGH RESULTS

Object Recognition



www.cs.toronto.edu/~hinton/absps/imagenet.pdf

BREAKTHROUGH RESULTS

Traffic Sign Recognition

www.idsia.ch/~juergen/ijcnn2011.pdf

BREAKTHROUGH RESULTS

Traffic Sign Recognition

www.idsia.ch/~juergen/ijcnn2011.pdf

Rank	Team	Method	Correct recognition rate
1	IDSIA	Committee of CNNs	99.46 %
2	INI	Human Performance	98.84 %
3	sermanet	Multi-Scale CNNs	98.31 %
4	CAOR	Random Forests	96.14 %

benchmark.ini.rub.de

BREAKTHROUGH RESULTS

Merck Competition
Deep NN and GPUs come out to play
blog.kaggle.com/2012/10/31/merck-competition-results-deep-nn-and-gpus-come-out-to-play/

BREAKTHROUGH RESULTS

Merck Competition
Deep NN and GPUs come out to play
blog.kaggle.com/2012/10/31/merck-competition-results-deep-nn-and-gpus-come-out-to-play/

Microsoft Research
Speech Recognition Leaps Forward
research.microsoft.com/en-us/news/features/speechrecognition-082911.aspx

BREAKTHROUGH RESULTS

Merck Competition
Deep NN and GPUs come out to play
blog.kaggle.com/2012/10/31/merck-competition-results-deep-nn-and-gpus-come-out-to-play/

Microsoft Research
Speech Recognition Leaps Forward
research.microsoft.com/en-us/news/features/speechrecognition-082911.aspx

and more...

LARGE ADOPTION

just to mention a few big names

“artificial intelligence is finally getting smart”

www.technologyreview.com/featuredstory/513696/deep-learning/

MIT Technology Review

"artificial intelligence is finally getting smart"

www.technologyreview.com/featuredstory/513696/deep-learning/

DON'T TAKE IT THE WRONG WAY

MIT Technology Review

“artificial intelligence is finally getting smart”

www.technologyreview.com/featuredstory/513696/deep-learning/

DON’T TAKE IT THE WRONG WAY

"Biology is hiding secrets well. We just don’t have the right tools to grasp the complexity of what’s going on."

Bruno Olshausen

www.wired.com/wiredenterprise/2013/05/neuro-artificial-intelligence/all/

MIT Technology Review

"artificial intelligence is finally getting smart"

www.technologyreview.com/featuredstory/513696/deep-learning/

DON'T TAKE IT THE WRONG WAY

"Biology is hiding secrets well. We just don't have the right tools to grasp the complexity of what's going on."

Bruno Olshausen

"We clearly don't have the right algorithms yet. It's going to take decades. This is not going to be an easy one, but I think there's hope."

Andrew Ng

www.wired.com/wiredenterprise/2013/05/neuro-artificial-intelligence/all/

ILSVRC2012 WINNER

Object Recognition

IMAGENET

Team name	Error (5 guesses)	Description
SuperVision	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	0.16422	Using only supplied training data
ISI	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.

www.image-net.org/challenges/LSVRC/2012/results.html

ILSVRC2012 WINNER

"Our model is a large, deep convolutional neural network trained on raw RGB pixel values. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three globally-connected layers with a final 1000-way softmax. It was trained on two NVIDIA GPUs for about a week. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of convolutional nets. To reduce overfitting in the globally-connected layers we employed hidden-unit "dropout", a recently-developed regularization method that proved to be very effective."

www.image-net.org/challenges/LSVRC/2012/results.html

WHAT'S NEW?

convolutional neural networks
max-pooling layers
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

WHAT'S NEW?

convolutional neural networks	Lecun et al., 1989
max-pooling layers	Fukushima, 1980
60 million parameters	
non-saturating neurons	
efficient GPU implementation	
"dropout"	

NEURAL NETWORK RENAISSANCE

IN 2006

Hinton et al. showed that
a particular form of **autoencoder** can be trained and stacked in a **greedily manner**, so that a bound on the probability of representing well the training data is increased at each layer.

IN 2006

Hinton et al. showed that
a particular form of **autoencoder** can be trained and stacked in a **greedily manner**, so that a bound on the probability of representing well the training data is increased at each layer.

others paper followed soon after

IN 2006

a partic[...] trained and
stacked [...] und on the
probab[...] ng data is

autoencoder
is a neural network
whose aim is to learn a
compressed representation
of the input data
(unsupervised)

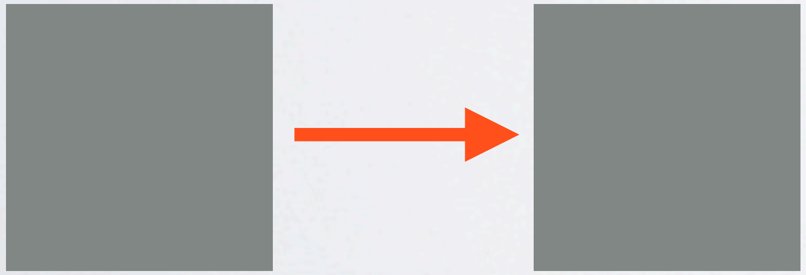
others paper followed soon after

KEY PRINCIPLES

unsupervised training of one layer at a time

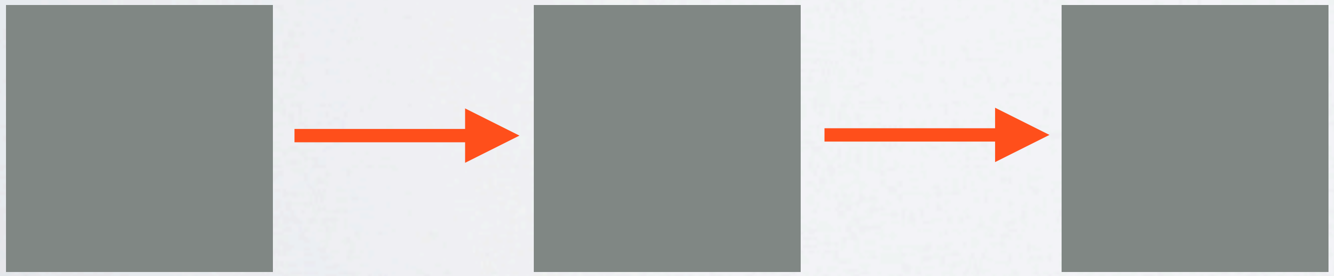
KEY PRINCIPLES

unsupervised training of one layer at a time



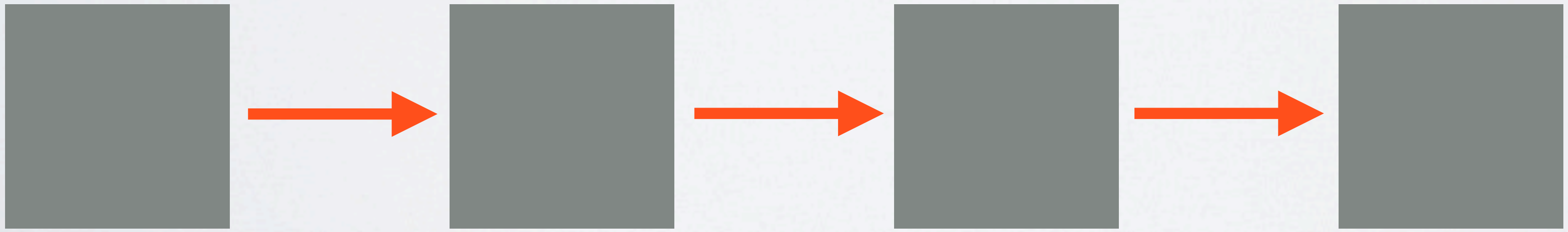
KEY PRINCIPLES

unsupervised training of one layer at a time



KEY PRINCIPLES

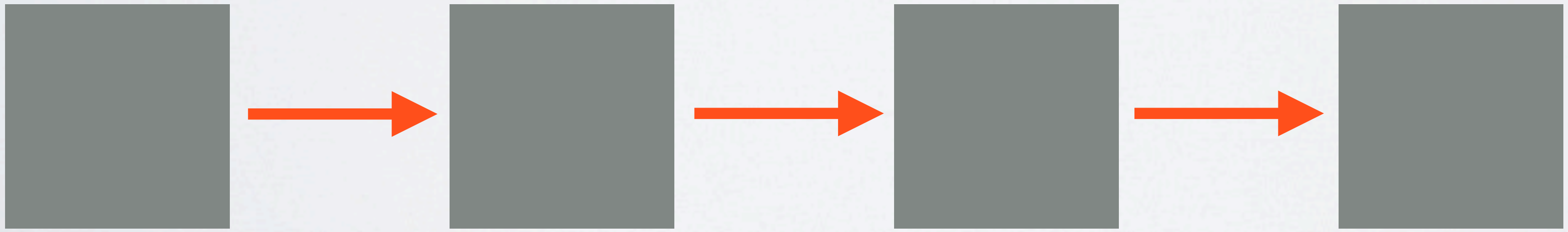
unsupervised training of one layer at a time



KEY PRINCIPLES

unsupervised training of one layer at a time

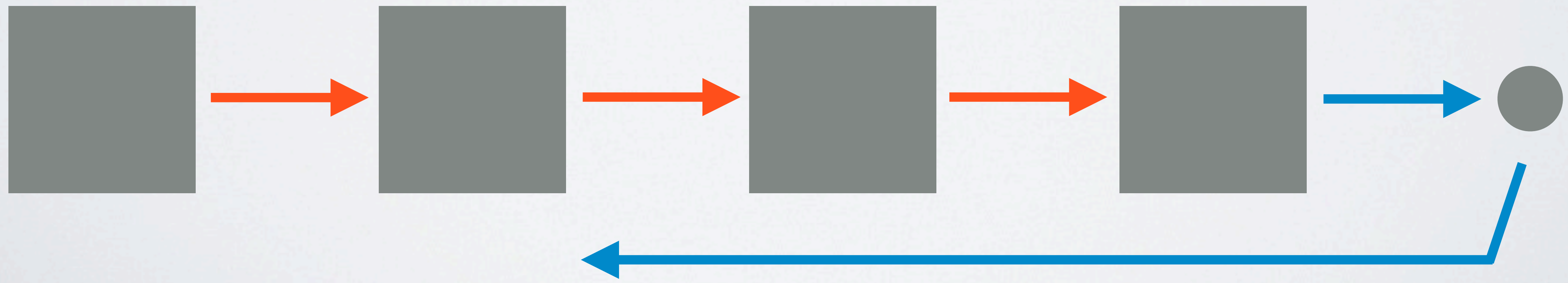
supervised training of all layers



KEY PRINCIPLES

unsupervised training of one layer at a time

supervised training of all layers

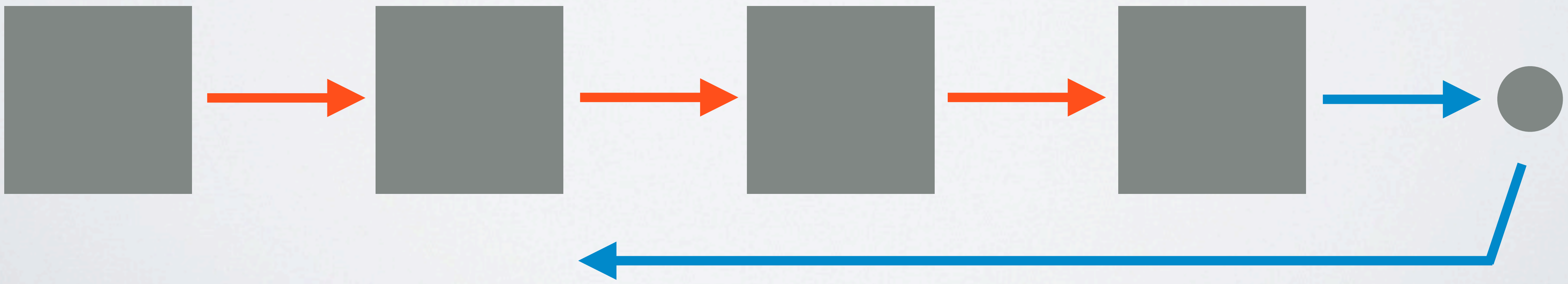


KEY PRINCIPLES

unsupervised training of one layer at a time

pre-training

supervised training of all layers



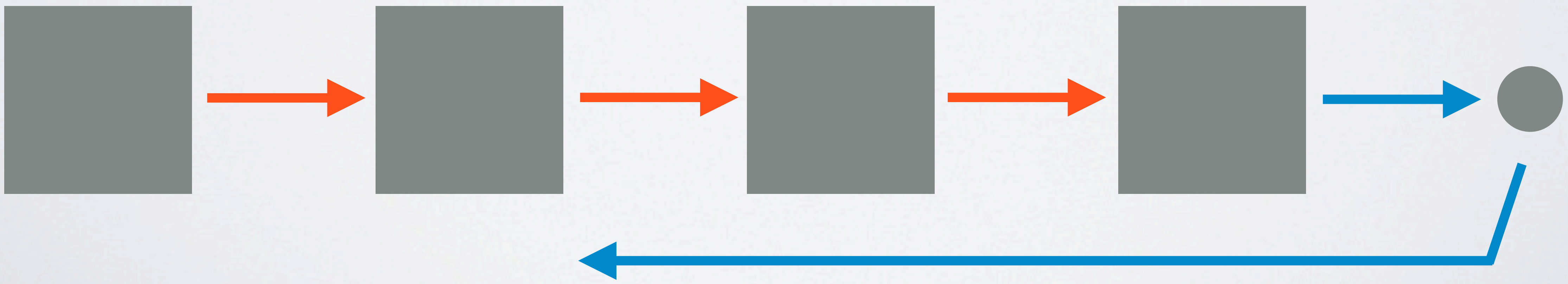
KEY PRINCIPLES

unsupervised training of one layer at a time

pre-training

supervised training of all layers

fine-tuning



UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

idea

learn one

layer of representation
at a time
on top of the previous one

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

idea

learn one
nonlinear
layer of representation
at a time
on top of the previous one

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

idea

learn one
nonlinear
layer of representation
at a time
on top of the previous one

learn one layer = learn neuron weights to extract one layer

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

before that (2006)

deep supervised
feedforward neural networks
tended to yield worse results then
shallow ones

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

hypothesis

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

hypothesis

learn **high-level abstractions** of the input

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

hypothesis

learn **high-level abstractions** of the input

helps fine-tuning to reach a **better local minimum**

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

hypothesis

learn **high-level abstractions** of the input

helps fine-tuning to reach a **better local minimum**

better **generalization**

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

motivation

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

motivation

in many problems, **high-level abstractions** are impossible to model with human ingenuity

UNSUPERVISED PRE-TRAINING

a.k.a. unsupervised feature learning

motivation

in many problems, **high-level abstractions** are impossible to model with human ingenuity

necessity to capture the **explanatory factors** (structure) of the data

WHY UNSUPERVISED?

WHY UNSUPERVISED?

supervised representation learning
in early layers tend to
discard information important
for higher concepts

Bengio et al, 2007

WHY UNSUPERVISED?

supervised representation learning
in early layers tend to
discard information important
for higher concepts
Bengio et al, 2007

it is more biologically plausible:

brain needs to learn 10^{14} synapses in 10^9 seconds

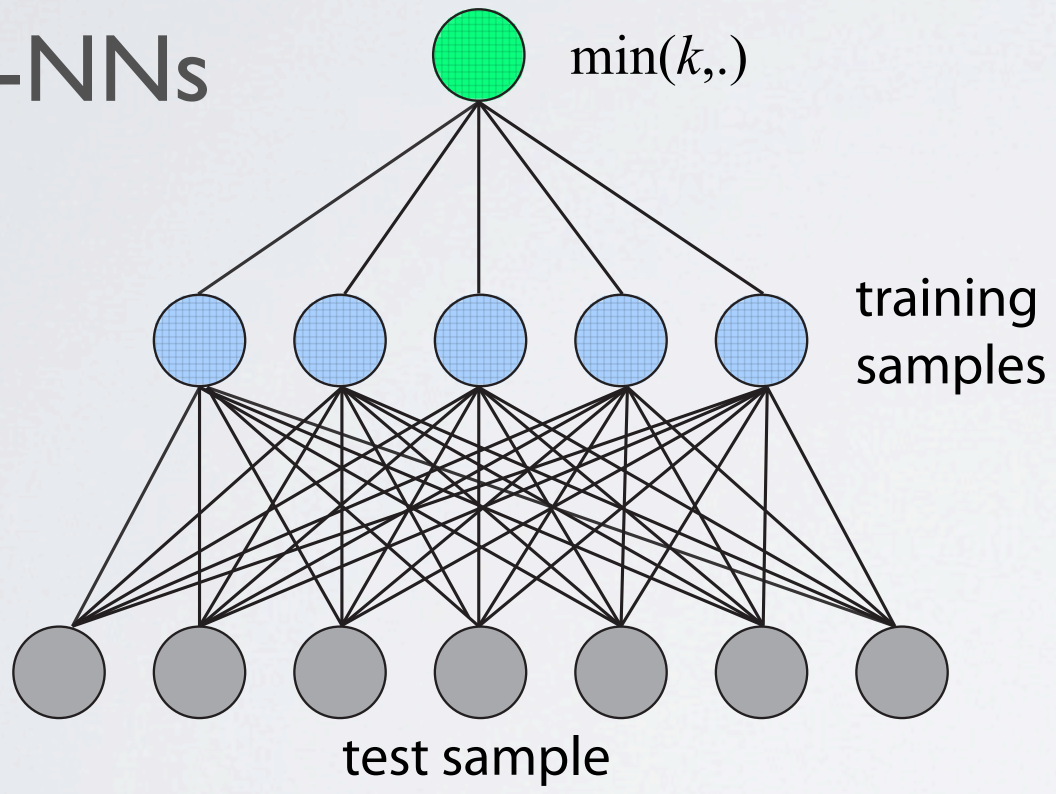
THE IMPORTANCE OF DEPTH

in many cases, depth 2 is enough to represent any function with a given target accuracy

THE IMPORTANCE OF DEPTH

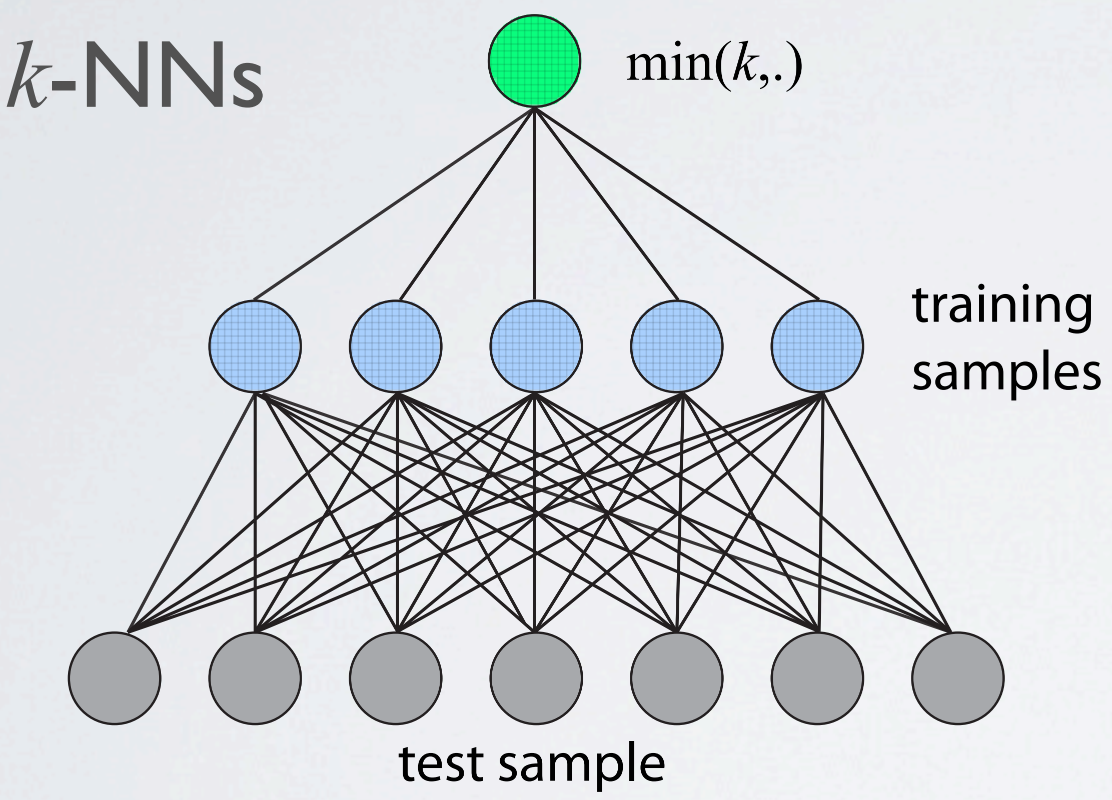
in many cases, depth 2 is enough to represent any function with a given target accuracy

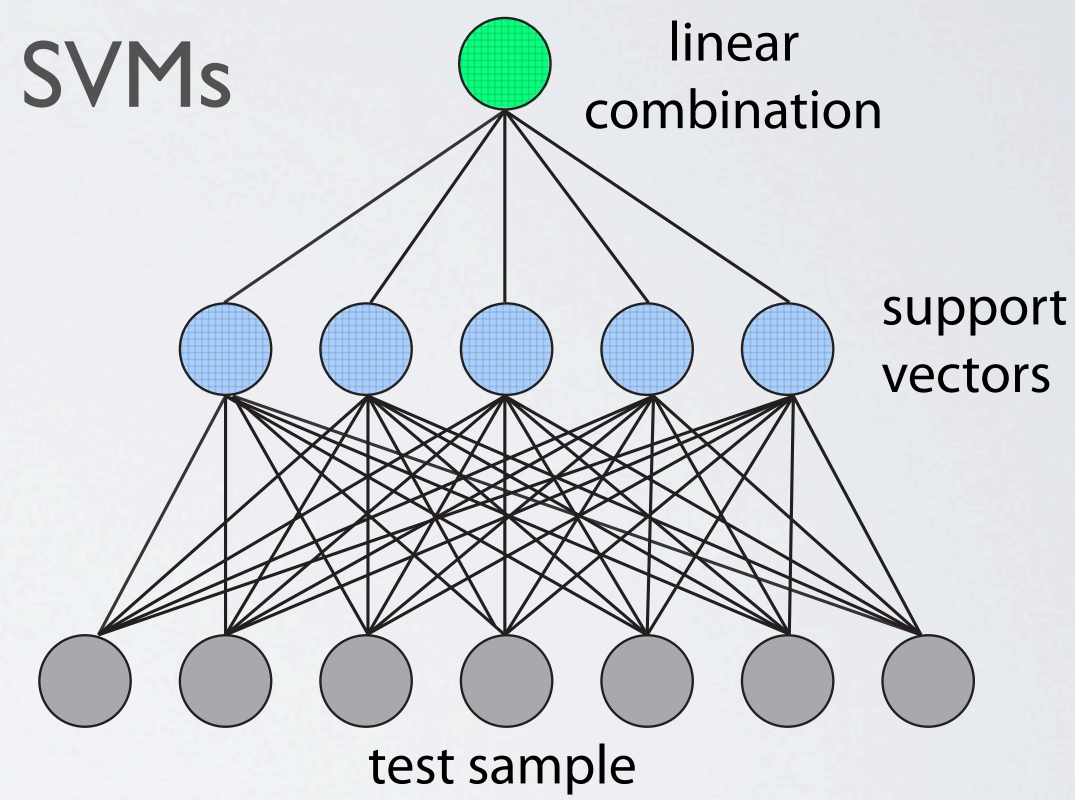
k-NNs



THE IMPORTANCE OF DEPTH

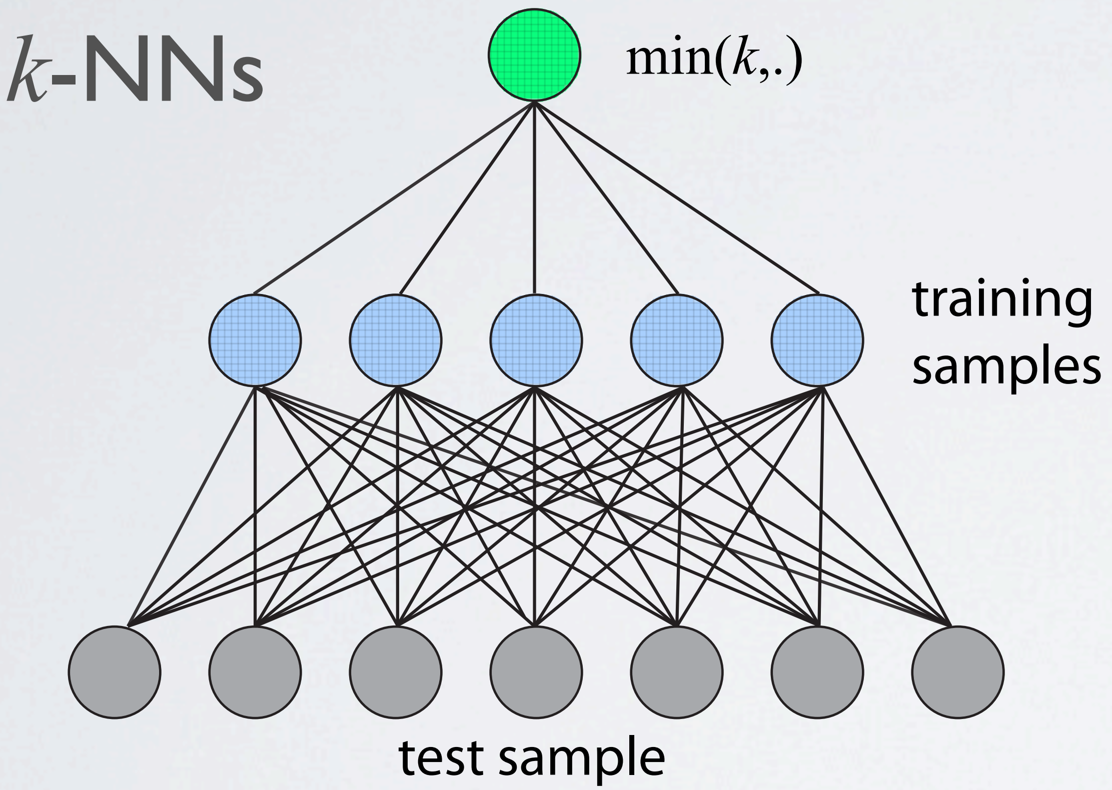
in many cases, depth 2 is enough to represent any function with a given target accuracy

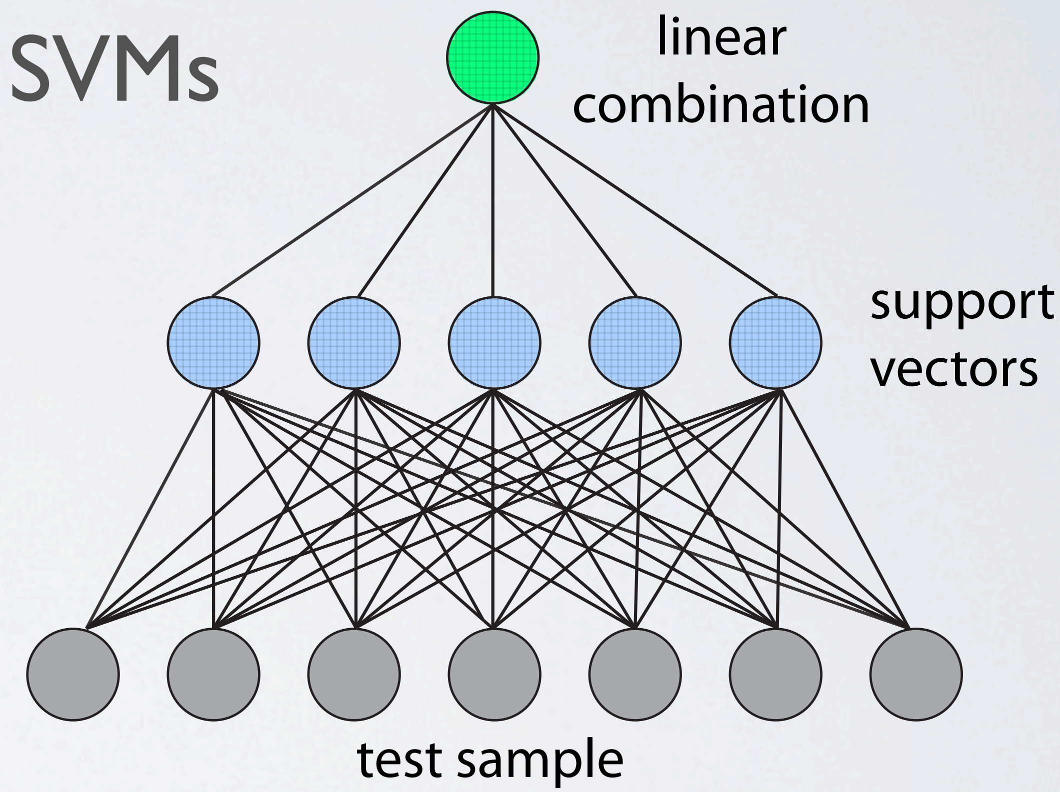




THE IMPORTANCE OF DEPTH

in many cases, depth 2 is enough to represent any function with a given target accuracy





but the required number of nodes in the graph may grow very large

THE IMPORTANCE OF DEPTH

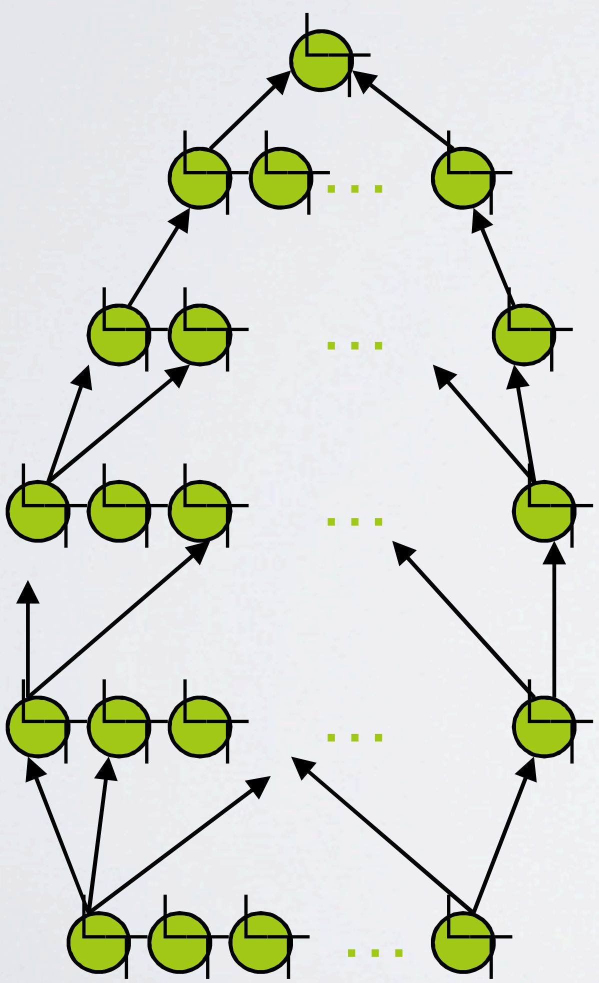
functions representable compactly with k layers may require exponential size with k-1 layers

Hastad et al 86, Hastad et al 91, Bengio et al 2007

THE IMPORTANCE OF DEPTH

functions representable compactly
with k layers may require exponential
size with k-1 layers

Hastad et al 86, Hastad et al 91, Bengio et al 2007

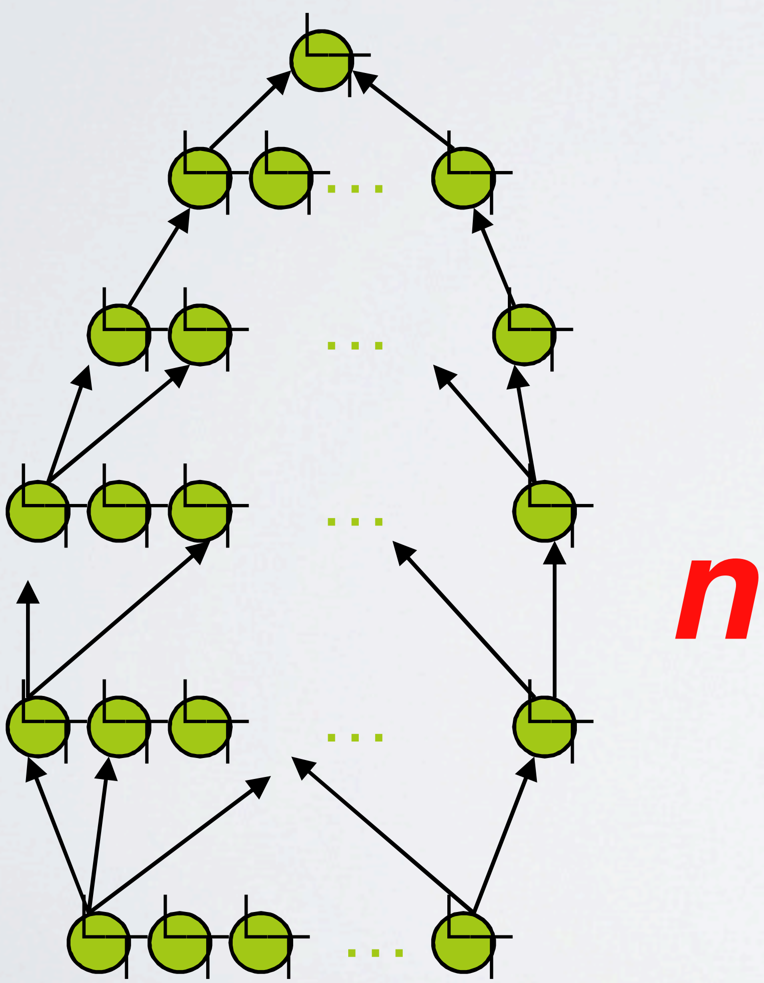


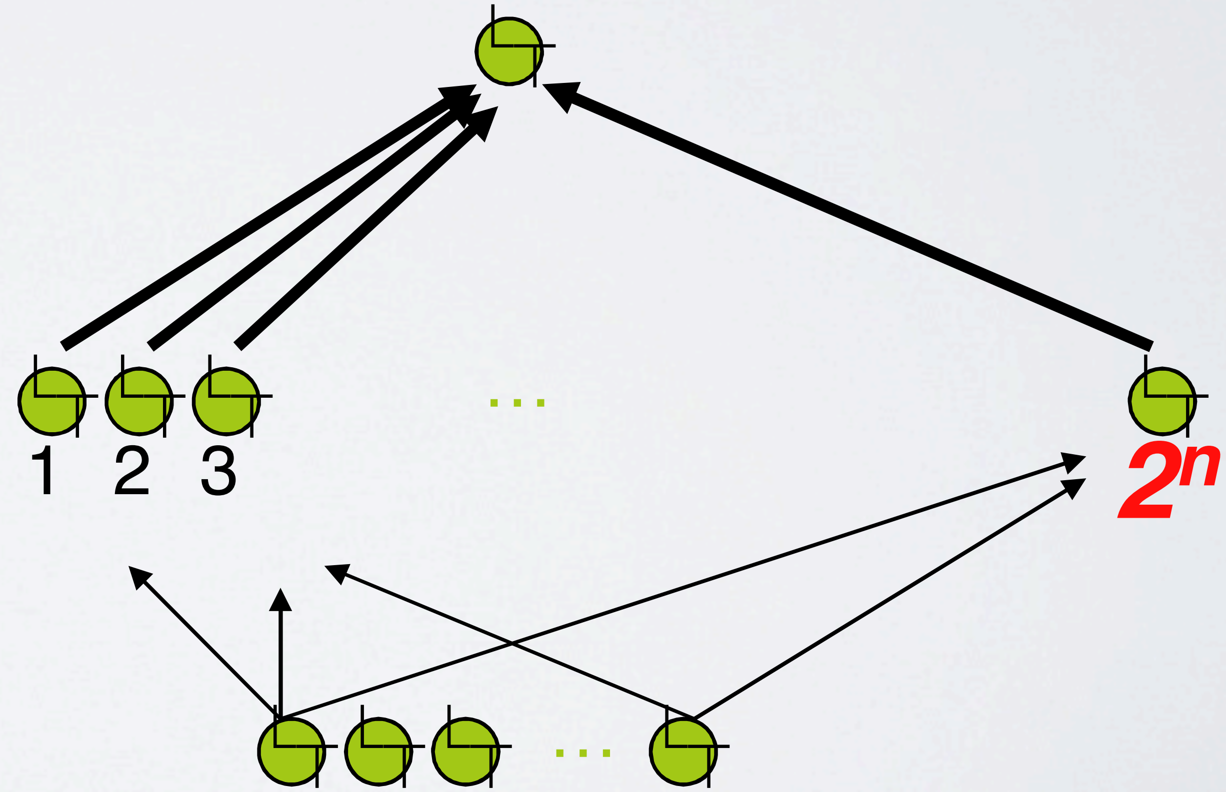
n

THE IMPORTANCE OF DEPTH

functions representable compactly
with k layers may require exponential
size with k-1 layers

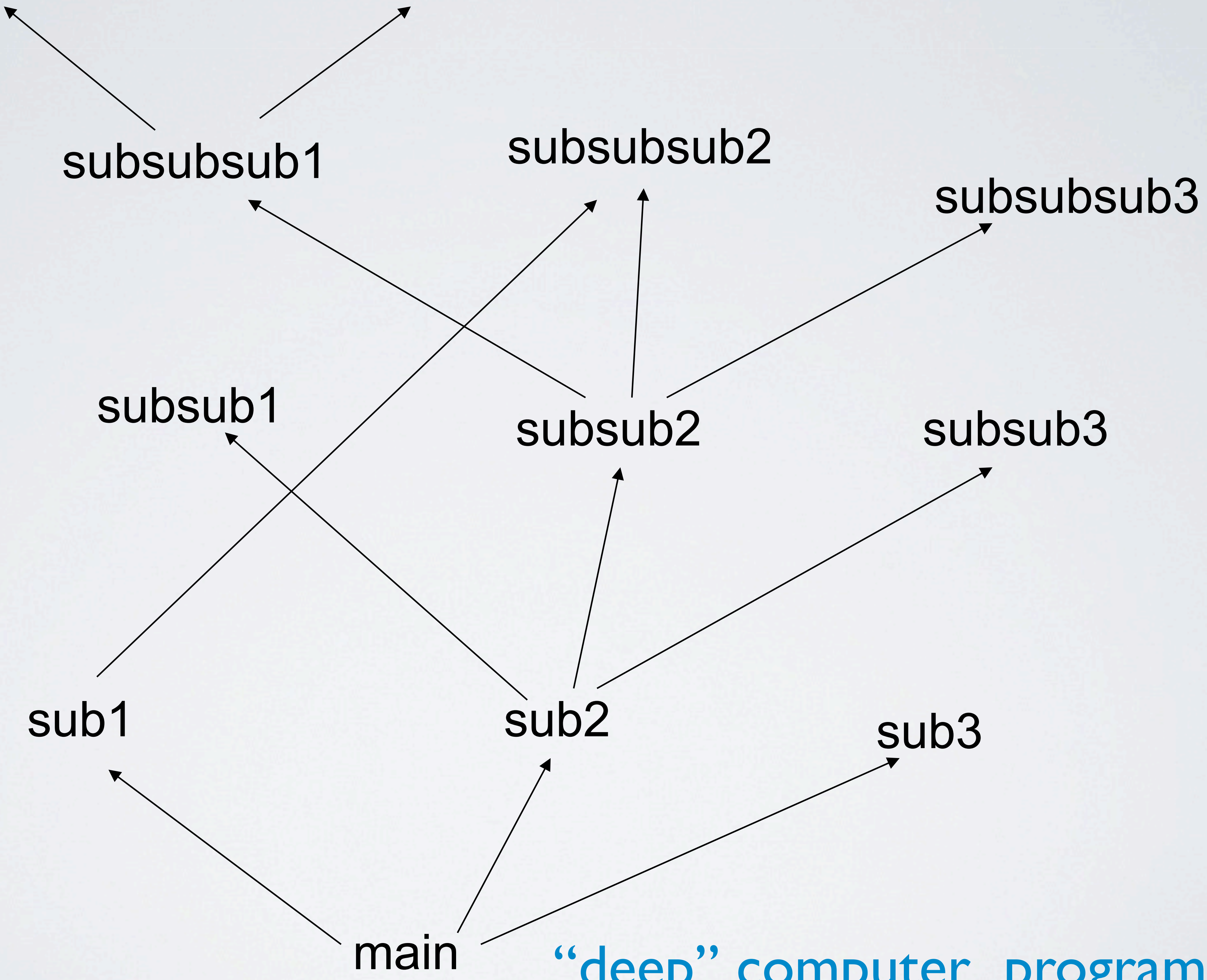
Hastad et al 86, Hastad et al 91, Bengio et al 2007





slide adapted from www.iro.umontreal.ca/~bengioy/talks/waterloo_may6.pdf | credit to Yoshua Bengio

INTUITION ON DEPTH



“deep” computer programs

INTUITION ON DEPTH

subroutine1 includes
subsub1 code and
subsub2 code and
subsubsub1 code

subroutine2 includes
subsub2 code and
subsub3 code and
subsubsub3 code and …

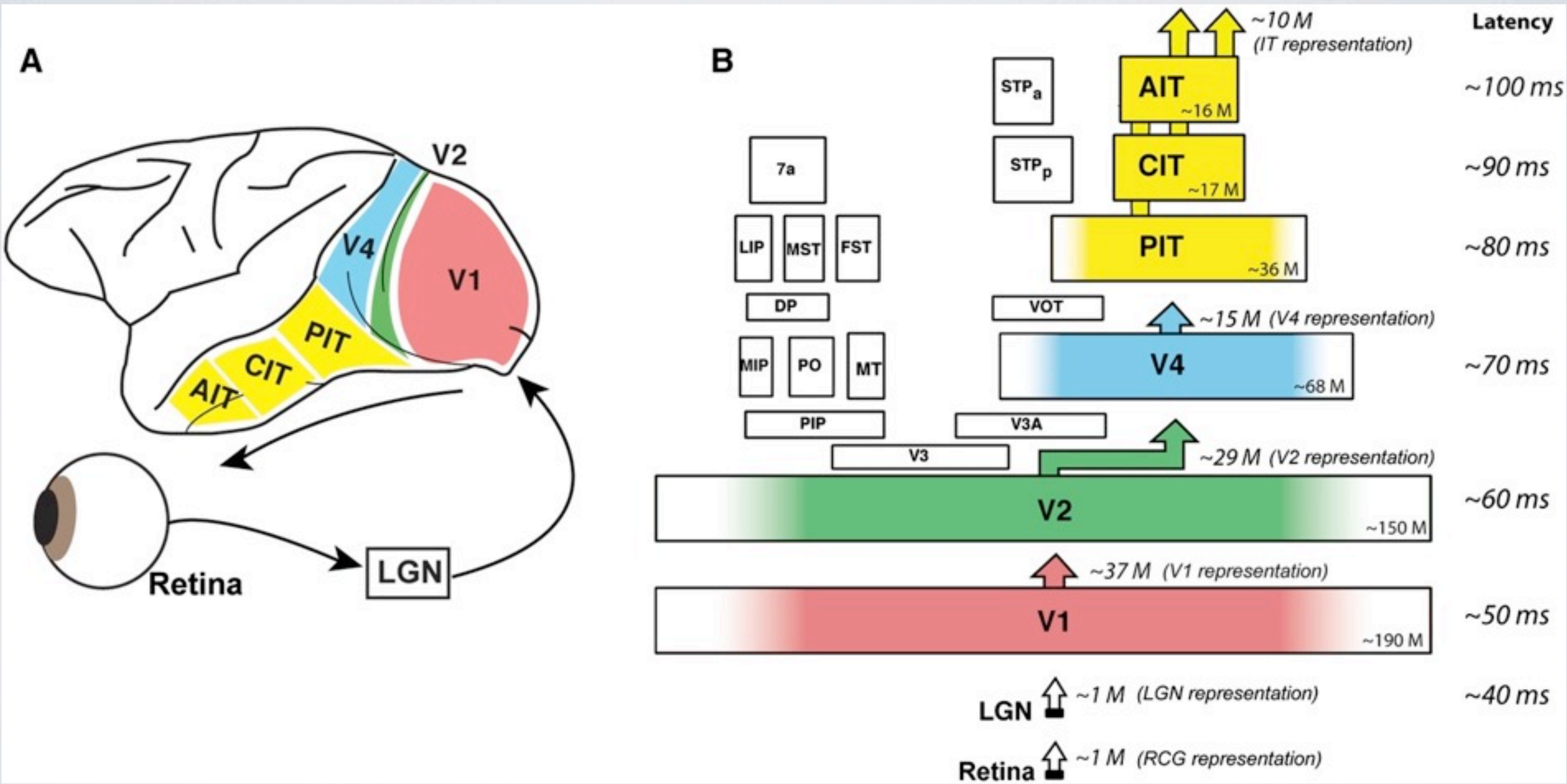
main

“shallow” computer programs

slide credit to Yoshua Bengio

THE IMPORTANCE OF DEPTH

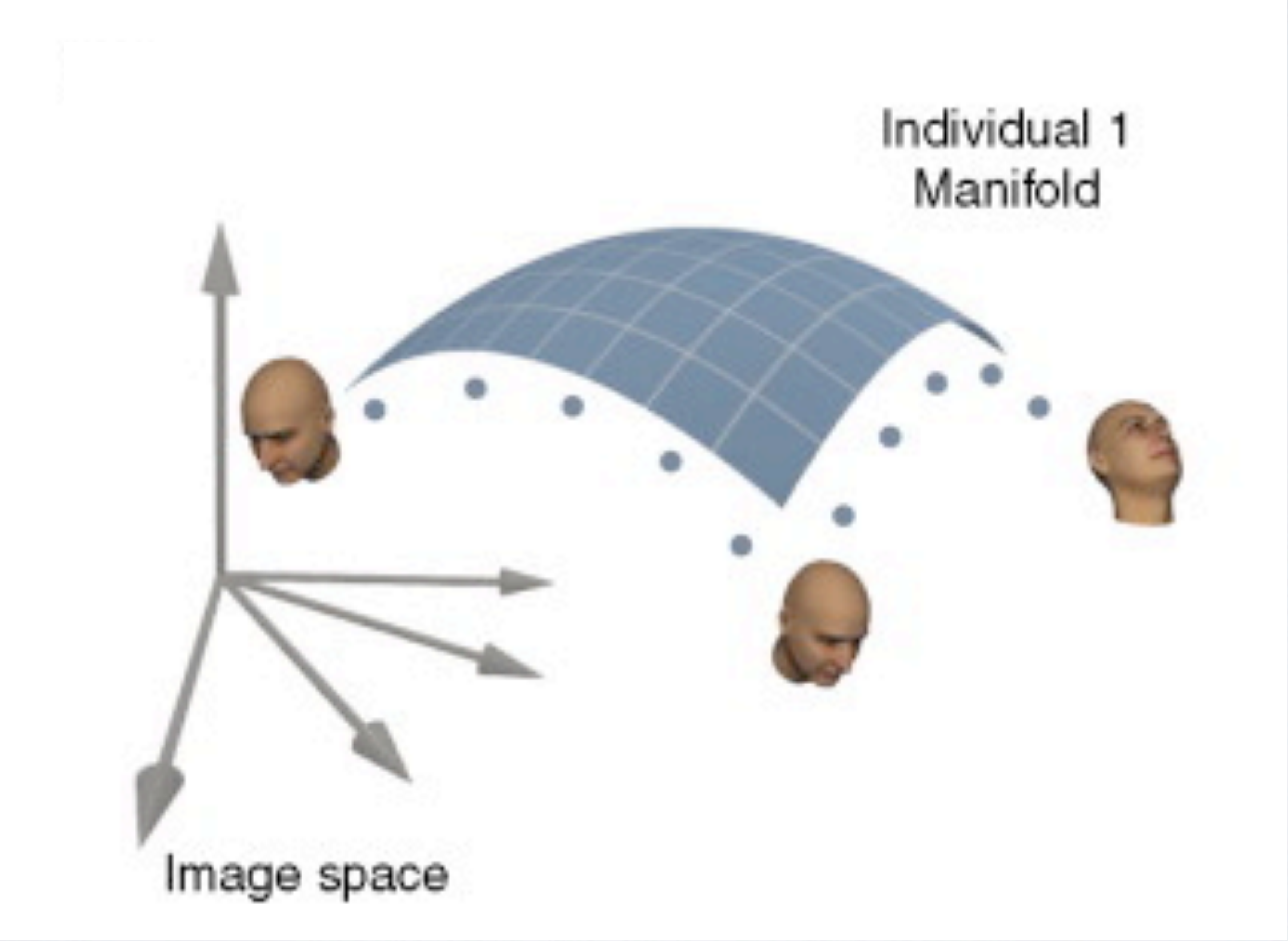
brain has a deep architecture



DiCarlo et al, 2012

THE IMPORTANCE OF DEPTH

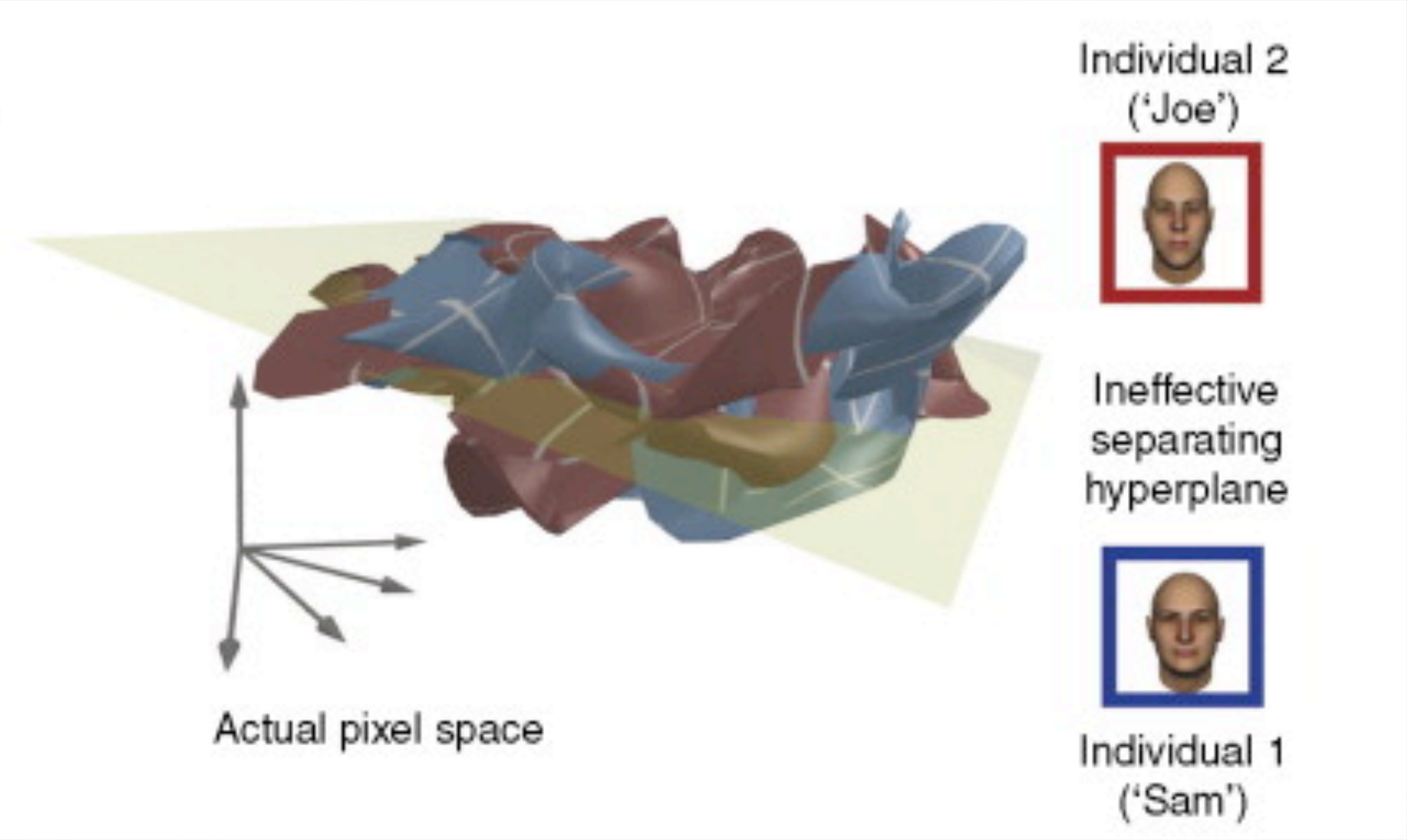
composing concepts | disentangling information



Cox and DiCarlo, 2007

THE IMPORTANCE OF DEPTH

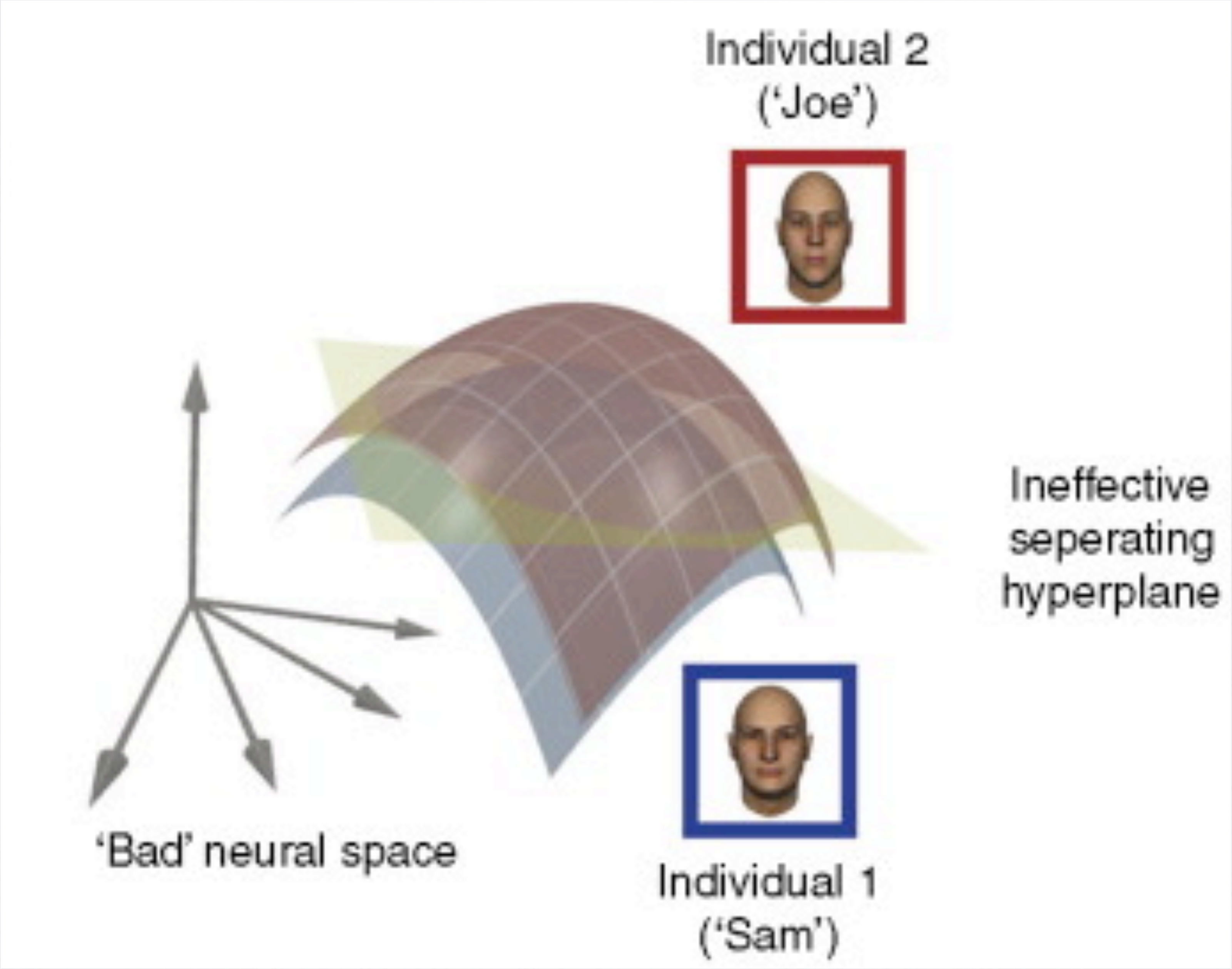
composing concepts | disentangling information



Cox and DiCarlo, 2007

THE IMPORTANCE OF DEPTH

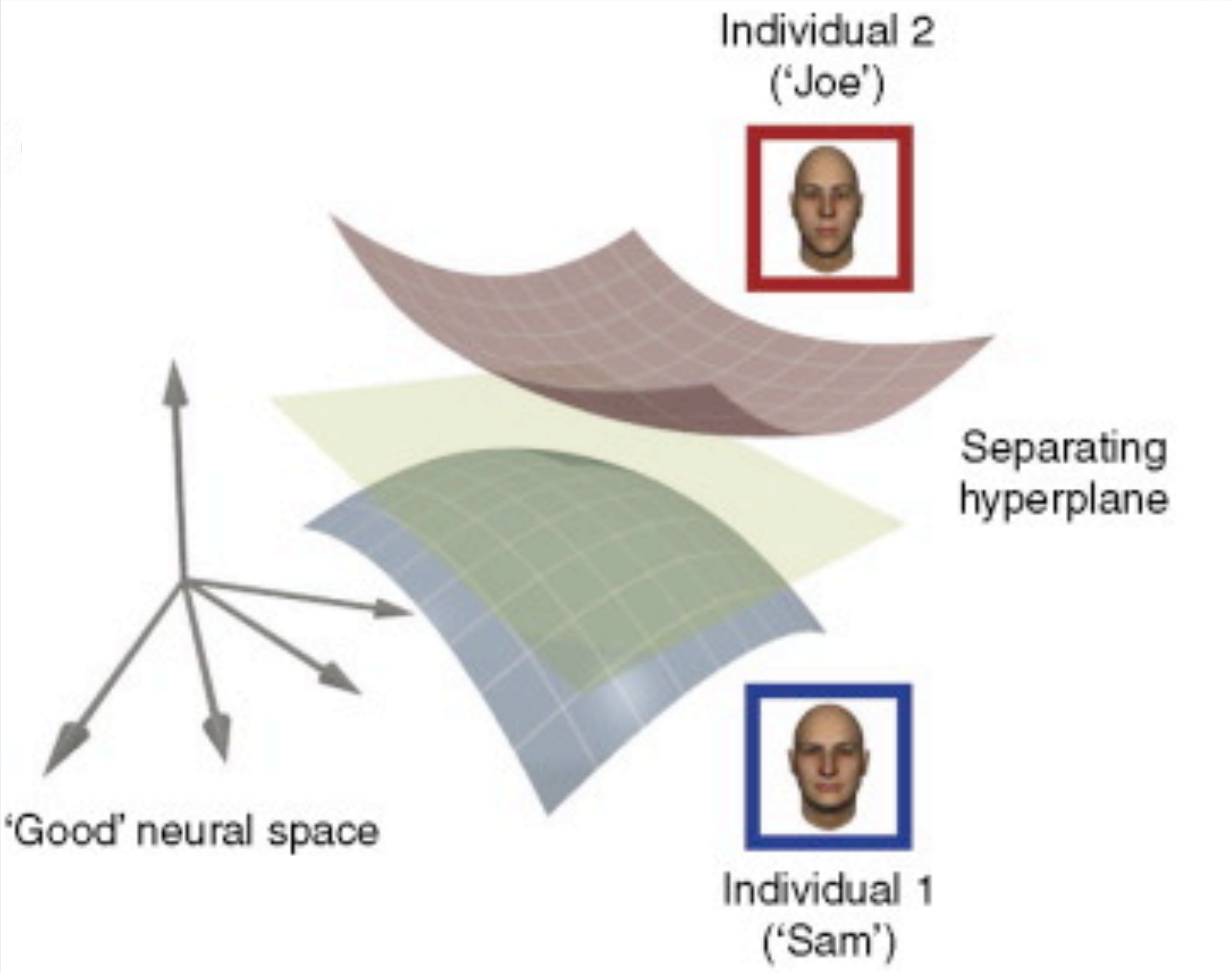
composing concepts | disentangling information



Cox and DiCarlo, 2007

THE IMPORTANCE OF DEPTH

composing concepts | disentangling information



Cox and DiCarlo, 2007

AFTER ALL

WHAT'S DEEP LEARNING?

AFTER ALL

WHAT'S DEEP LEARNING?

"When there is more than one hidden layer being learned, this is deep learning."

Geoffrey Hinton, Coursera class

AFTER ALL

WHAT'S DEEP LEARNING?

"When there is more than one hidden layer being learned, this is deep learning."

Geoffrey Hinton, Coursera class

HOW DEEP?

AFTER ALL

WHAT'S DEEP LEARNING?

"When there is more than one hidden layer being learned, this is deep learning."

Geoffrey Hinton, Coursera class

HOW DEEP?

"When the number of levels can be data selected, this is a deep architecture."

Yoshua Bengio, SSTiC 2013

NEURAL NETWORKS RENAISSANCE

In 2006...

NEURAL NETWORKS RENAISSANCE

In 2006...

autoencoders

NEURAL NETWORKS RENAISSANCE

In 2006...

autoencoders

pre-training

NEURAL NETWORKS RENAISSANCE

In 2006...

autoencoders

pre-training

unsupervised feature learning

NEURAL NETWORKS RENAISSANCE

In 2006...

autoencoders

pre-training

unsupervised feature learning

stacked in a greedily manner

NEURAL NETWORKS RENAISSANCE

In 2006...

autoencoders

pre-training

unsupervised feature learning

stacked in a greedily manner

...

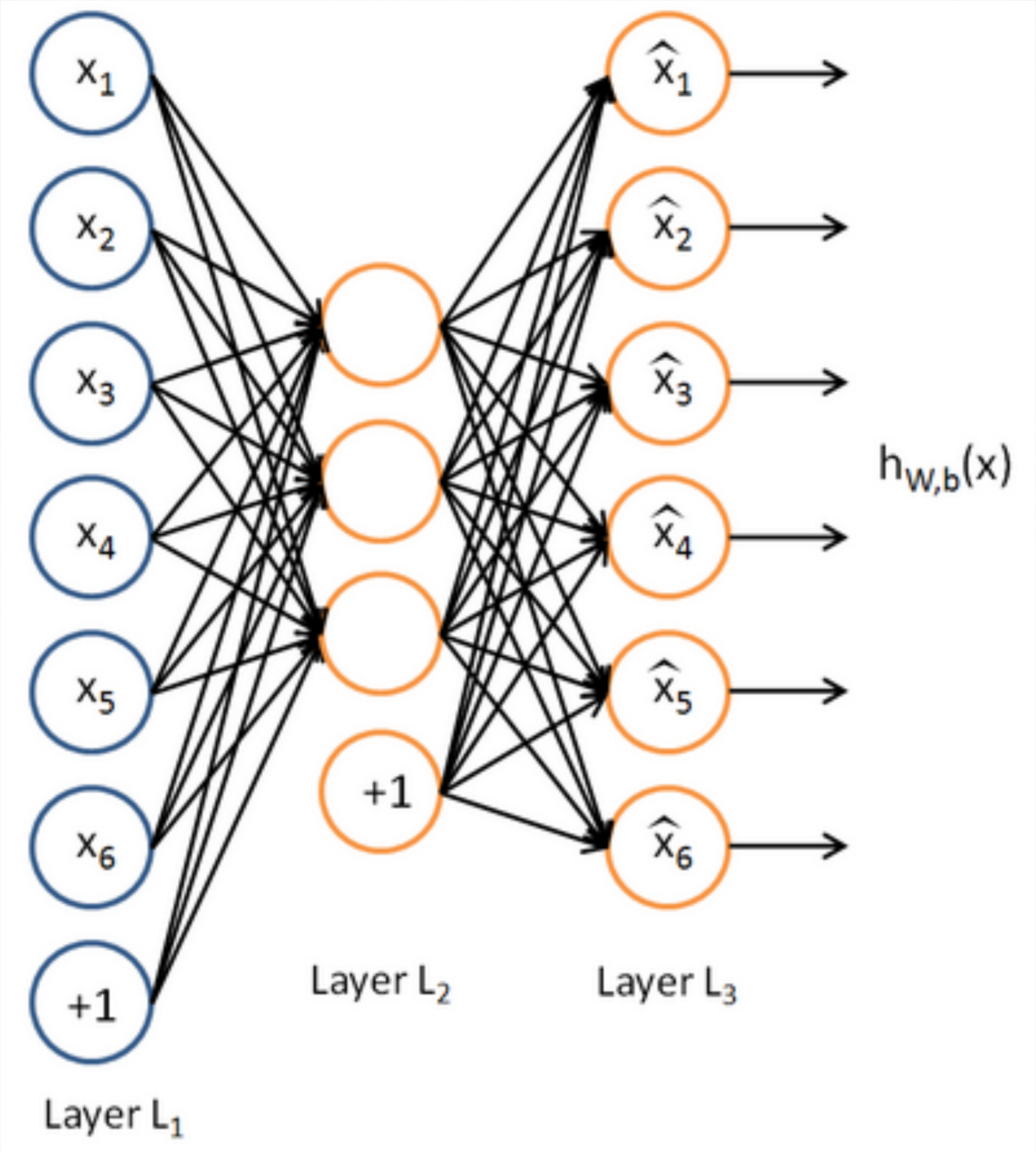
AUTOENCODER NEURAL NETS

AUTOENCODER NEURAL NETS

Is an unsupervised learning algorithm that applies **backpropagation**, setting the target values to be equal to the inputs.

$$\hat{f}_\theta(x) \approx (x)$$

AUTOENCODER NEURAL NETS



deeplearning.stanford.edu/wiki/images/f/f9/Autoencoder636.png

AUTOENCODER NEURAL NETS

tries to learn an approximation to the identity function

AUTOENCODER NEURAL NETS

tries to learn an approximation to the identity function

the network is usually forced to learn a **compressed** representation of the input

AUTOENCODER NEURAL NETS

tries to learn an approximation to the identity function

the network is usually forced to learn a **compressed** representation of the input

tries to discover structure in the data

AUTOENCODER NEURAL NETS

following the notation of previous lectures, we can back propagate the reconstruction error by setting

$$\delta_j^{(3)} = -(x_j - a_j^{(3)}) .* g'(z^{(3)})$$

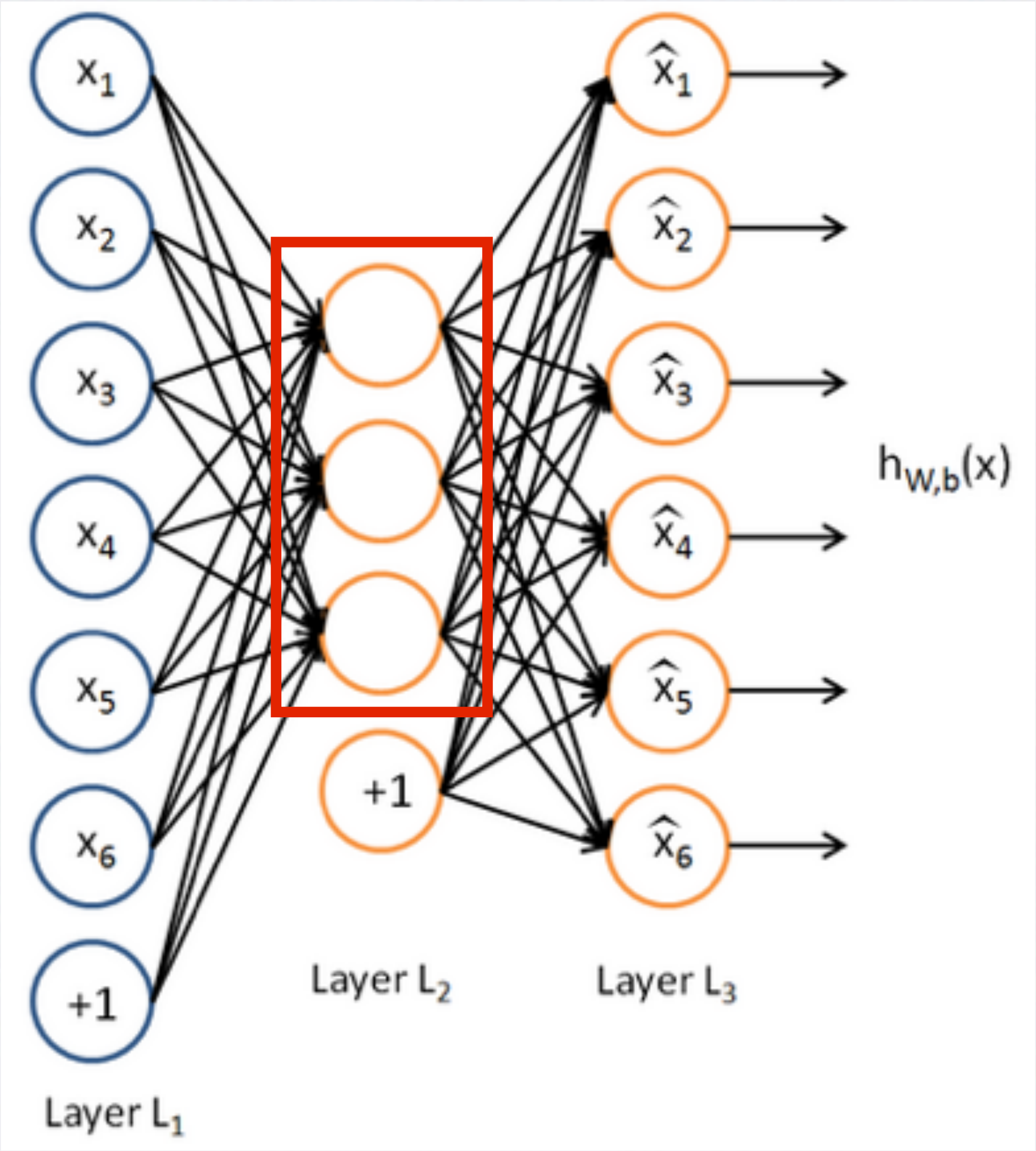
$$\delta^{(2)} = \left((\theta^{(2)})^T \delta^{(3)}\right) .* g'(z^{(2)})$$

AUTOENCODER NEURAL NETS

interesting structures can be discovered by placing constraints on the network such as **sparsity**

AUTOENCODER NEURAL NETS

interesting structures can be discovered by placing constraints on the network such as **sparsity**



AUTOENCODER NEURAL NETS

interesting structures can be discovered by placing constraints on the network such as **sparsity**

let

$$\hat{\rho} = \frac{1}{m} \sum_{i=1}^{m} \left[a_j^{(2)}(x^{(i)}) \right]$$

be the average activation of the hidden unit j (averaged over the training set)

AUTOENCODER NEURAL NETS

we would like to (approximately) enforce

$$\hat{\rho} = \rho$$

AUTOENCODER NEURAL NETS

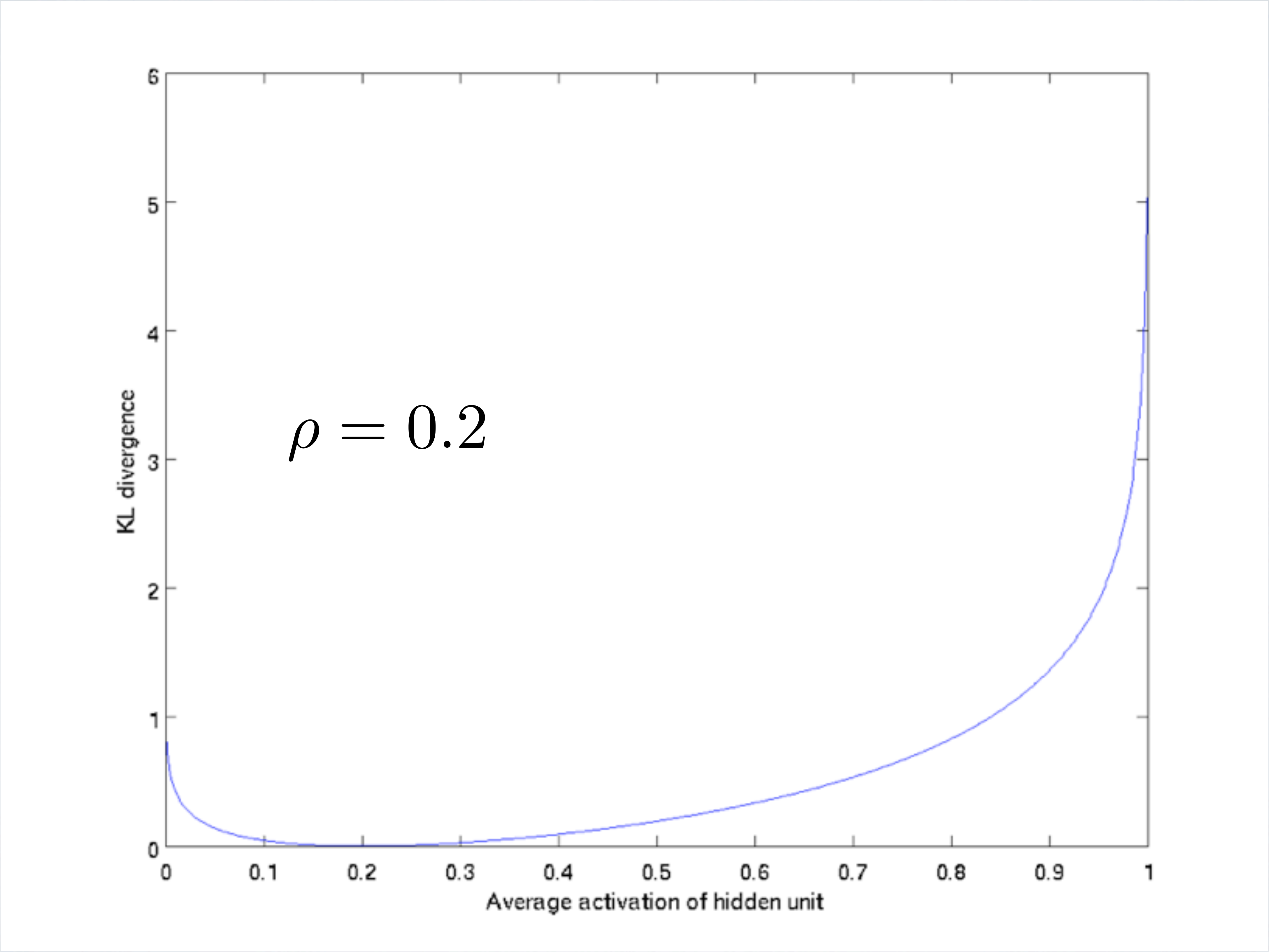
we would like to (approximately) enforce

$$\hat{\rho} = \rho$$

a possible choice of of penalty to add in the optimization objective is

$$\sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} = \sum_{j=1}^{s_2} \mathrm{KL}(\rho || \hat{\rho}_j)$$

AUTOENCODER NEURAL NETS



deeplearning.stanford.edu/wiki/images/4/48/KLPenaltyExample.png

AUTOENCODER NEURAL NETS

the objective function then becomes

$$J_{\text{sparse}}(\theta) = J(\theta) + \beta \sum_{j=1}^{s_2} \text{KL}(\rho || \hat{\rho}_j)$$

AUTOENCODER NEURAL NETS

the objective function then becomes

$$J_{\text{sparse}}(\theta) = J(\theta) + \beta \sum_{j=1}^{s_2} \text{KL}(\rho||\hat{\rho}_j)$$

and

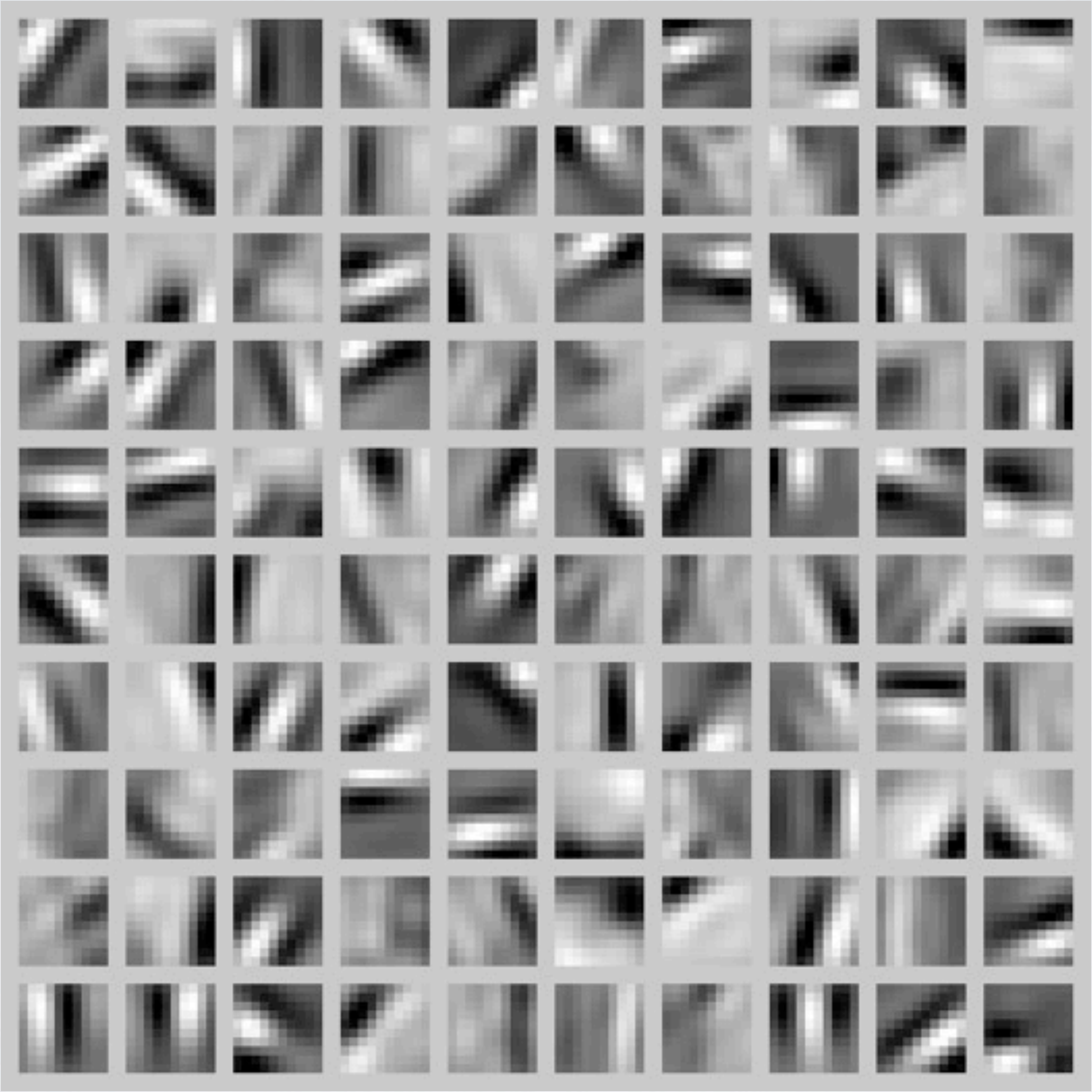
$$\delta_i^{(2)} = ((\theta_i^{(2)})^T \delta_i^{(3)}). * g'(z_i^{(2)}) + \beta \left(-\frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i} \right)$$

AUTOENCODER NEURAL NETS

visualizing the function learned from image patches

AUTOENCODER NEURAL NETS

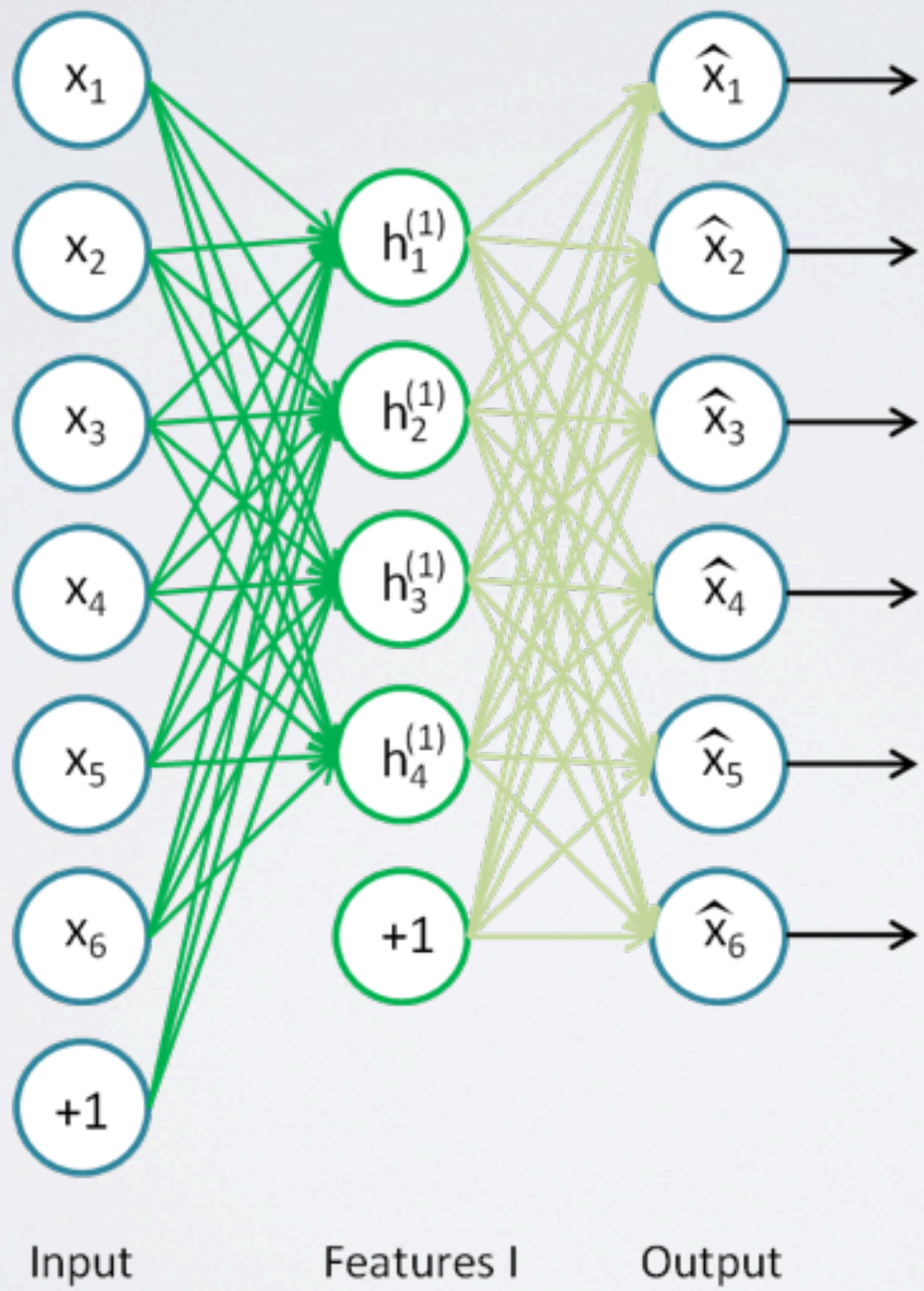
visualizing the function learned from image patches



deeplearning.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity

STACKED AUTOENCODERS

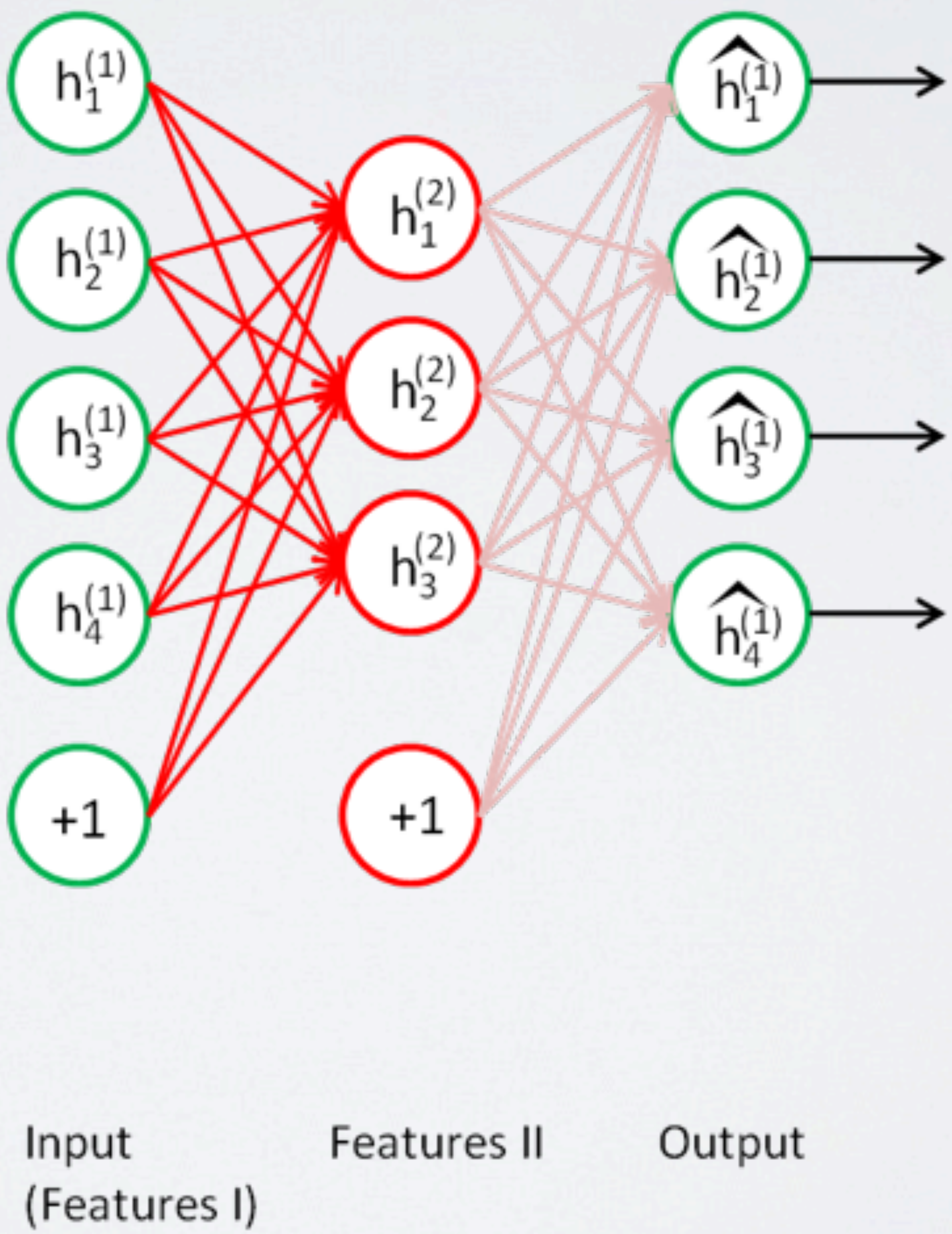
a NN consisting of multiple layers of autoencoders



deeplearning.stanford.edu/wiki/images/0/0e/Stacked_SparseAE_Features1.png

STACKED AUTOENCODERS

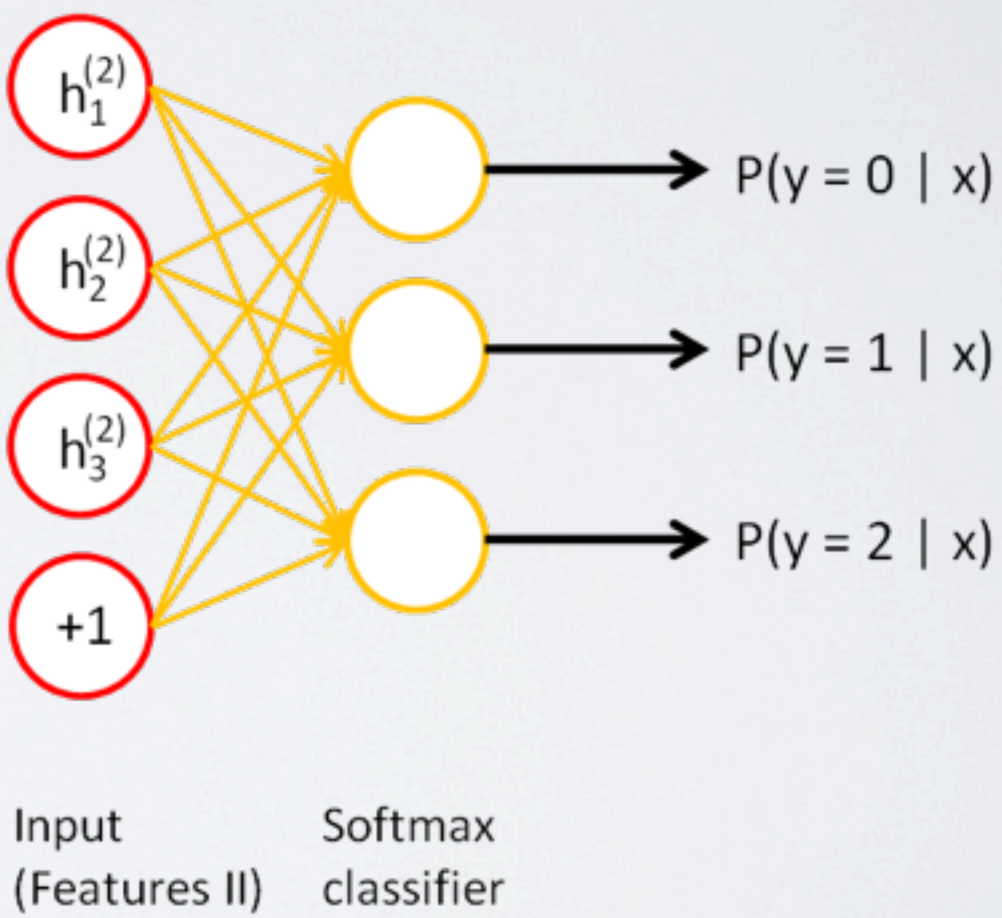
a NN consisting of multiple layers of autoencoders



deeplearning.stanford.edu/wiki/images/b/bf/Stacked_SparseAE_Features2.png

STACKED AUTOENCODERS

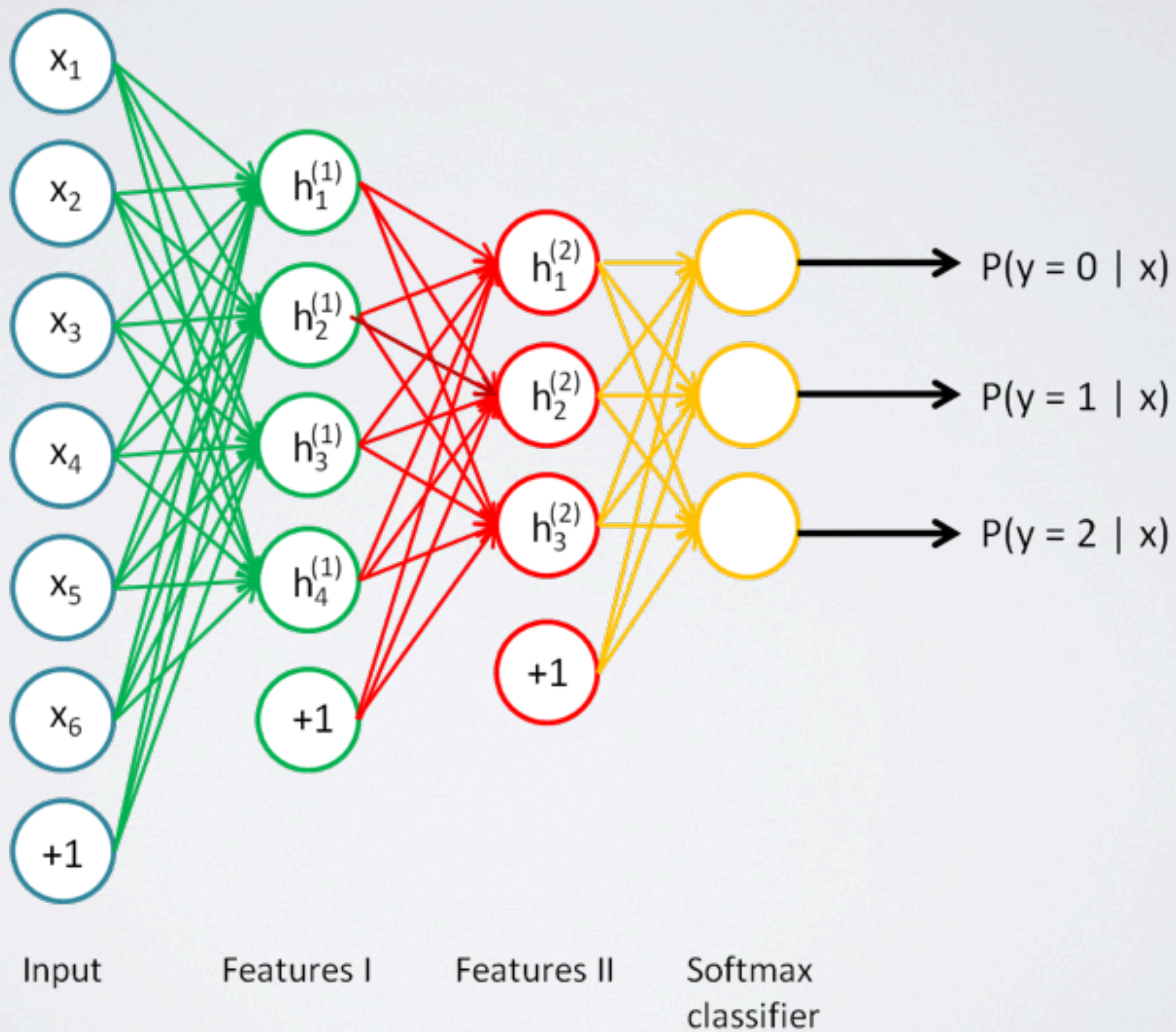
a NN consisting of multiple layers of autoencoders



deeplearning.stanford.edu/wiki/images/6/6b/Stacked_Softmax_Classifier.png

STACKED AUTOENCODERS

a NN consisting of multiple layers of autoencoders



deeplearning.stanford.edu/wiki/images/5/5c/Stacked_Combined.png

UNSUPERVISED PRE-TRANING

BEFORE

deep architectures performed poorly

UNSUPERVISED PRE-TRANING

BEFORE

deep architectures performed poorly

AFTER

state-of-the-art results

BUT...

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

NO PRE-TRAINING AT ALL!

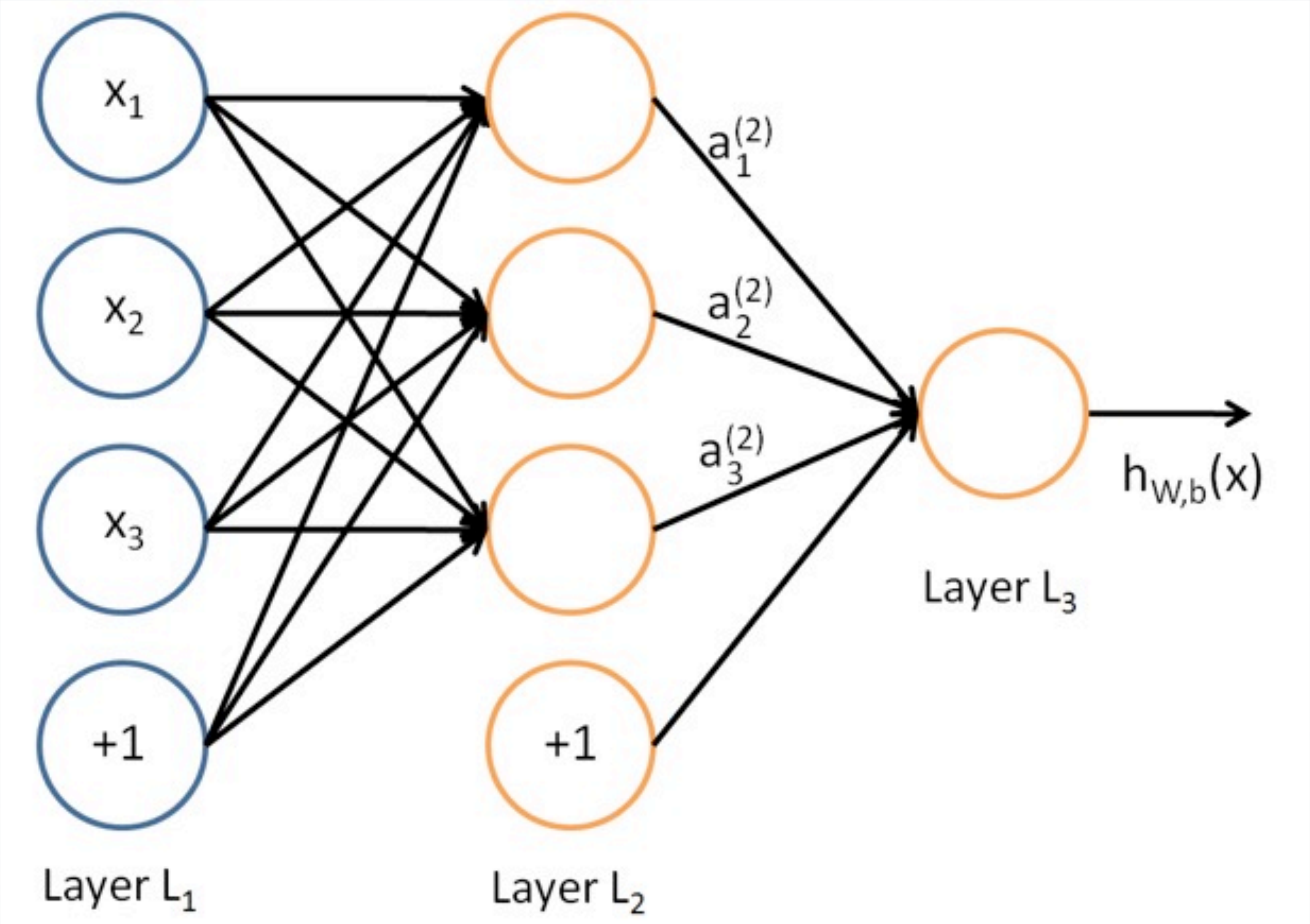
ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

NO PRE-TRAINING AT ALL!

CONVOLUTIONAL NEURAL NETWORKS

FULLY-CONNECTED NNS



ufldl.stanford.edu/wiki/images/9/99/Network331.png

CONVOLUTIONAL NNS

inspired by Hubel and Wiesel cells

CONVOLUTIONAL NNS

inspired by Hubel and Wiesel cells

simple

complex

CONVOLUTIONAL NNS

inspired by Hubel and Wiesel cells

simple

responds **maximally** to specific **local stimulus**

complex

CONVOLUTIONAL NNS

inspired by Hubel and Wiesel cells

simple

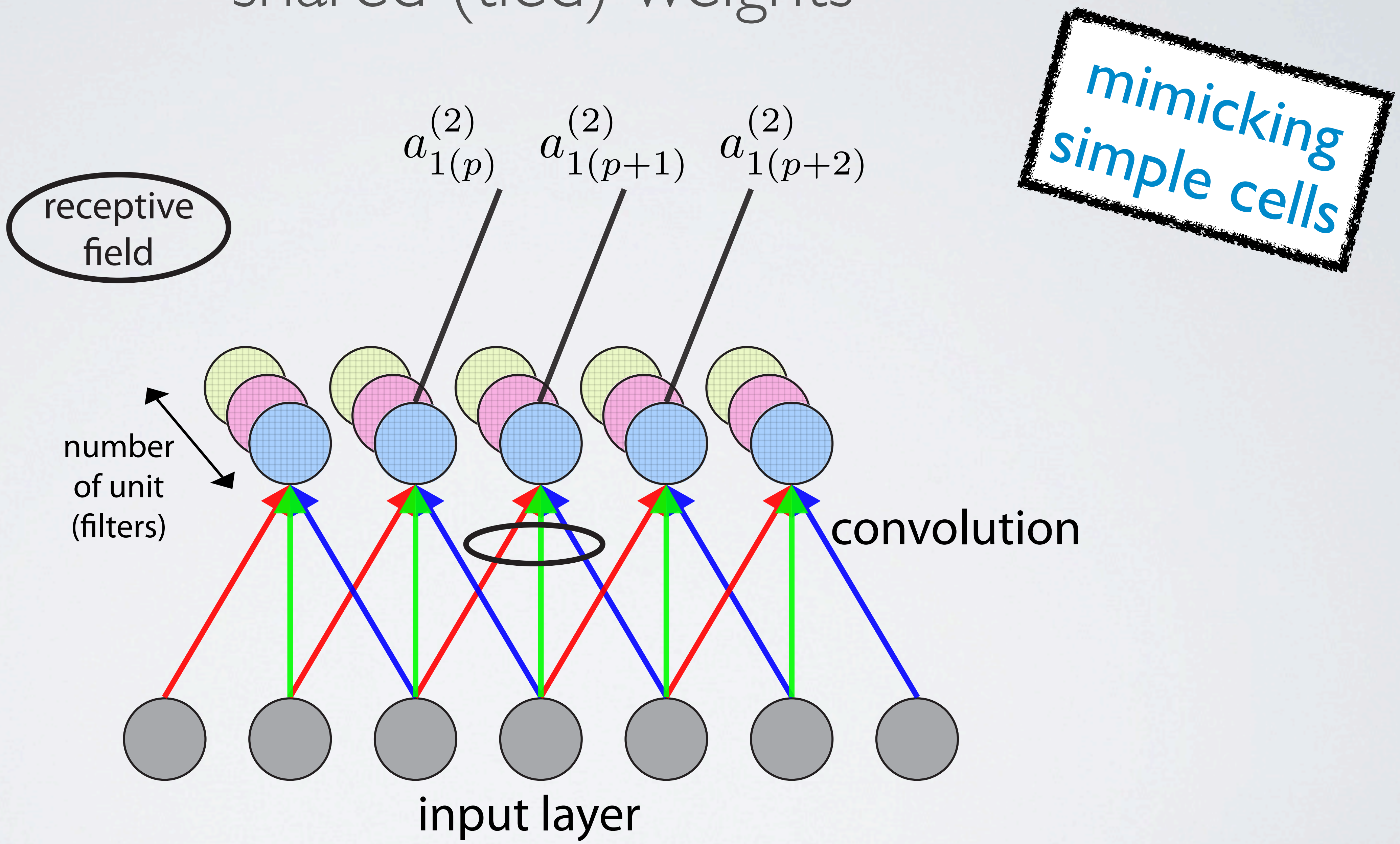
responds **maximally** to specific **local stimulus**

complex

local invariance to the exact position of stimulus

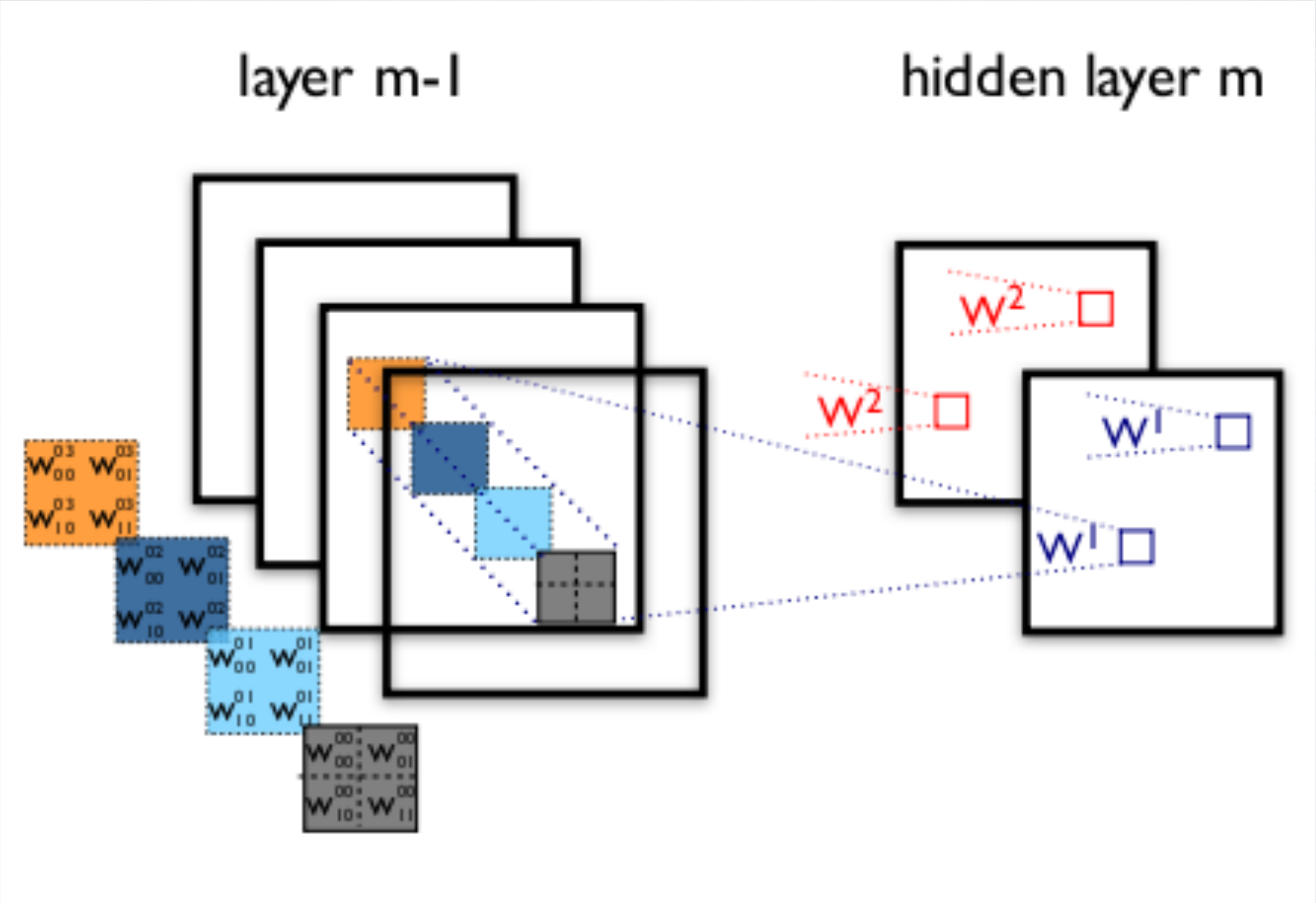
CONVOLUTIONAL NNS

shared (tied) weights



CONVOLUTIONAL NNS

shared (tied) weights



deeplearning.net/tutorial/_images/cnn_explained.png

CONVOLUTIONAL NNS

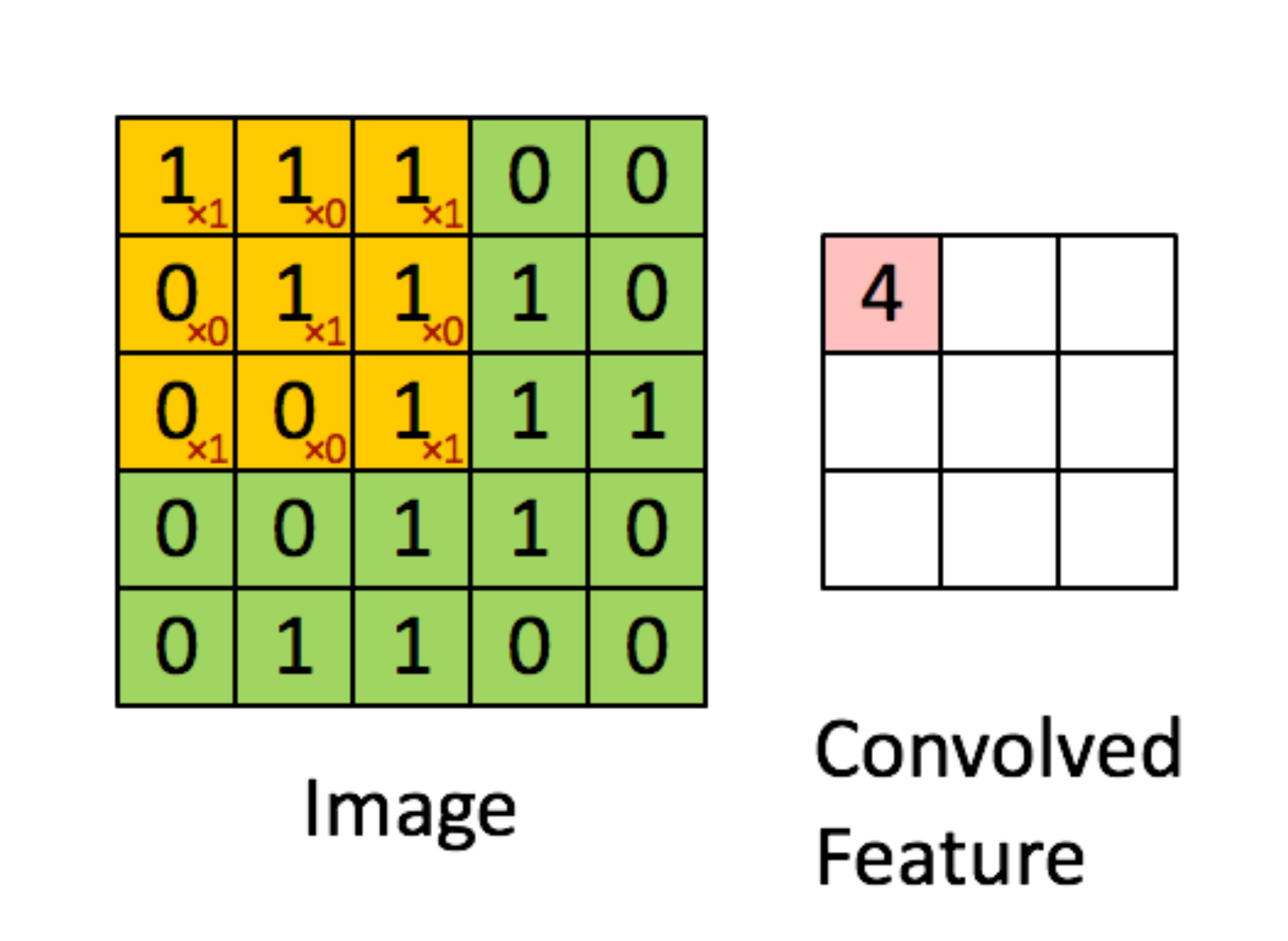
shared (tied) weights

$$\frac{\partial}{\partial \theta_{ij}} J(\theta) = \sum_{p \in \mathcal{P}} \left[a^{(l)}_{j(p)} \delta^{(l+1)}_{i} \right]$$

$\mathcal{P}$ is the set of all positions where θ_i is convolved

CONVOLUTIONAL NNS

shared (tied) weights



ufldl.stanford.edu/wiki/images/6/6c/Convolution_schematic.gif

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

CONVOLUTIONAL NNS

max (or average) pooling units

mimicking complex cells

$$g(a_j) = \max(a_{j,(p)}) \quad \forall p \in \mathcal{N}$$

where $\mathcal{N}$ defines the pooling regions that may or may not overlapped

CONVOLUTIONAL NNS

max (or average) pooling units

mimicking complex cells

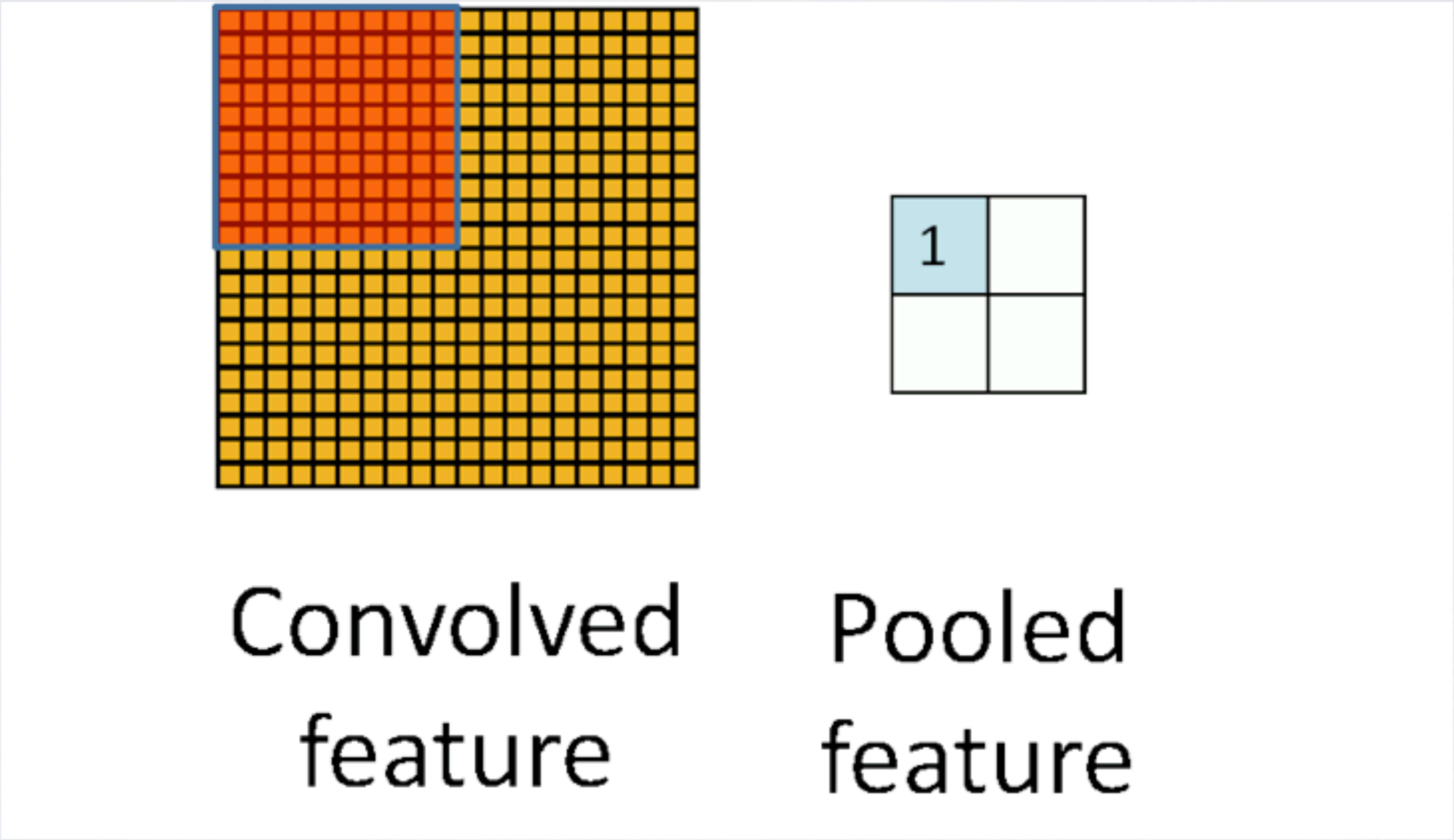
$$g(a_j) = \max(a_{j,(p)}) \quad \forall p \in \mathcal{N}$$

where $\mathcal{N}$ defines the $\boxed{\text{pooling regions}}$ that may or may not overlapped

receptive field

CONVOLUTIONAL NNS

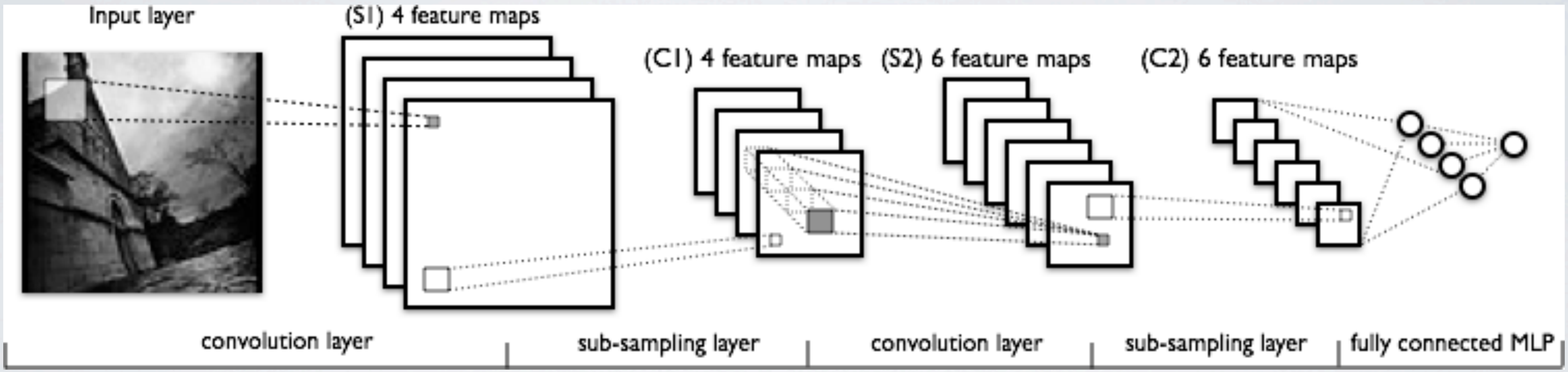
max (or average) pooling units



ufldl.stanford.edu/wiki/images/0/08/Pooling_schematic.gif

CONVOLUTIONAL NETS

convolution + pooling



http://deeplearning.net/tutorial/_images/mylenet.png

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

CONVOLUTIONAL NNS

non-saturating nonlinearity

rectified linear units

$$g(z^{(l)}) = \max(0, z^{(l)})$$

CONVOLUTIONAL NNS

non-saturating nonlinearity

rectified linear units

$$g(z^{(l)}) = \max(0, z^{(l)})$$

instead of

$$g(z^{(l)}) = \frac{1}{1 + e^{-z^{(l)}}}$$

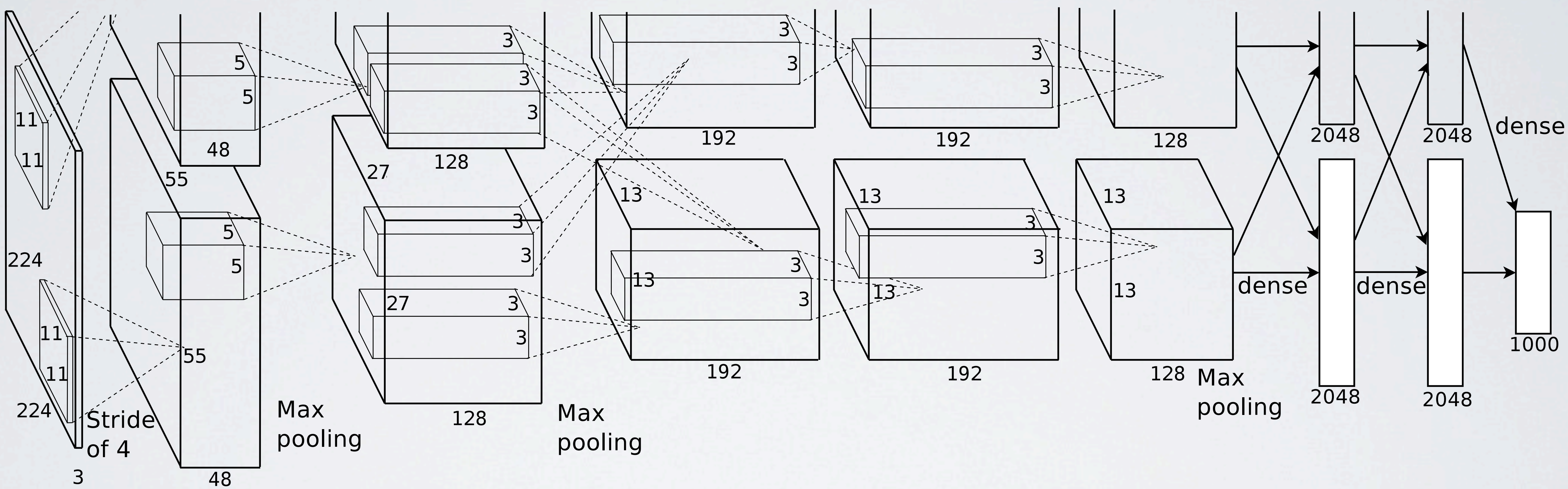
ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

THE 60 MILLION PARAMETER ARCHITECTURE



www.cs.toronto.edu/~hinton/absps/imagenet.pdf

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

CONVOLUTIONAL NNS

dropout regularization recipe

CONVOLUTIONAL NNS

dropout regularization recipe

set to zero the output of each
hidden neuron with probability 0.5

CONVOLUTIONAL NNS

dropout regularization recipe

set to zero the output of each
hidden neuron with probability 0.5

neurons "dropped out" contribute
neither in the forward pass nor in back-propagation

CONVOLUTIONAL NNS

dropout regularization recipe

set to zero the output of each
hidden neuron with probability 0.5

neurons "dropped out" contribute
neither in the forward pass nor in back-propagation

at test time, use all the neurons
but multiply their outputs by 0.5

CONVOLUTIONAL NNS

dropout regularization implications

CONVOLUTIONAL NNS

dropout regularization implications

every time an input is presented,
the neural network samples a different architecture

CONVOLUTIONAL NNS

dropout regularization implications

every time an input is presented,
the neural network samples a different architecture

all the sampled architectures share weights

CONVOLUTIONAL NNS

dropout regularization implications

every time an input is presented,
the neural network samples a different architecture

all the sampled architectures share weights

reduces complex co-adaptations of neurons

ILSVRC2012 WINNER

convolutional neural networks Lecun et al., 1989
max-pooling layers Fukushima, 1980
60 million parameters
non-saturating neurons
efficient GPU implementation
"dropout"

NO PRE-TRAINING AT ALL!

NO-PRETRAINING AT ALL?

NO-PRETRAINING AT ALL?

"if you **initialize the layers correctly**, you may not need pre-training at all, provided you have **enough labeled data**"

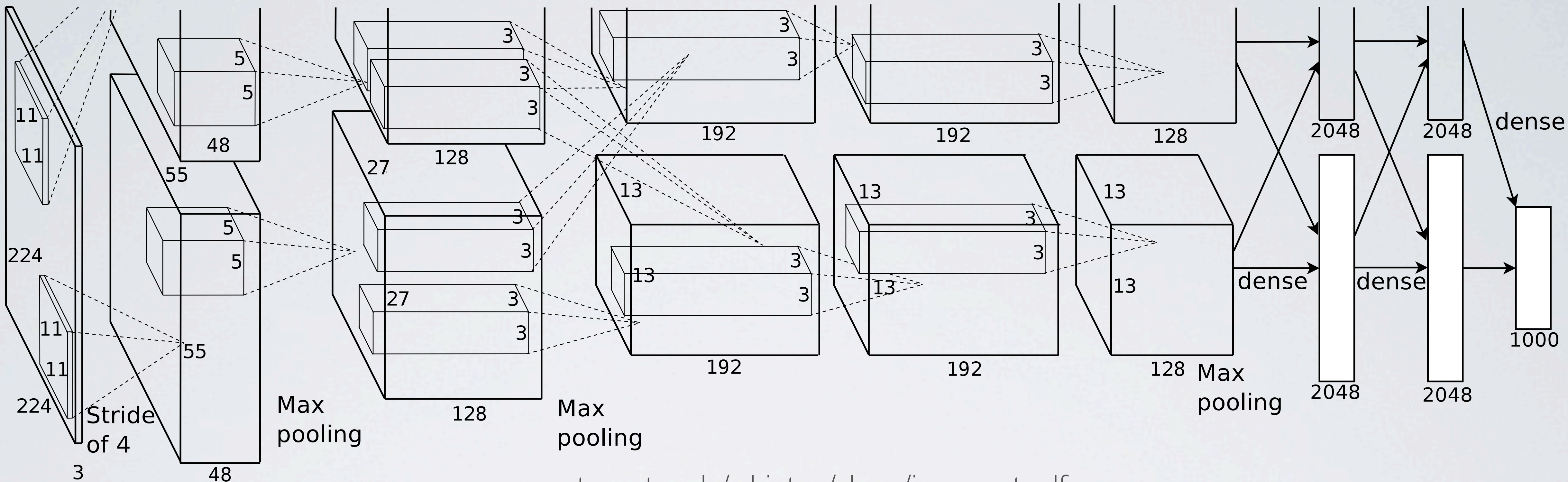
NO-PRETRAINING AT ALL?

"if you **initialize the layers correctly**, you may not need pre-training at all, provided you have **enough labeled data**"

"however, you can **always increase the size** of your neural net so that even a **huge amount of data is still not enough**"

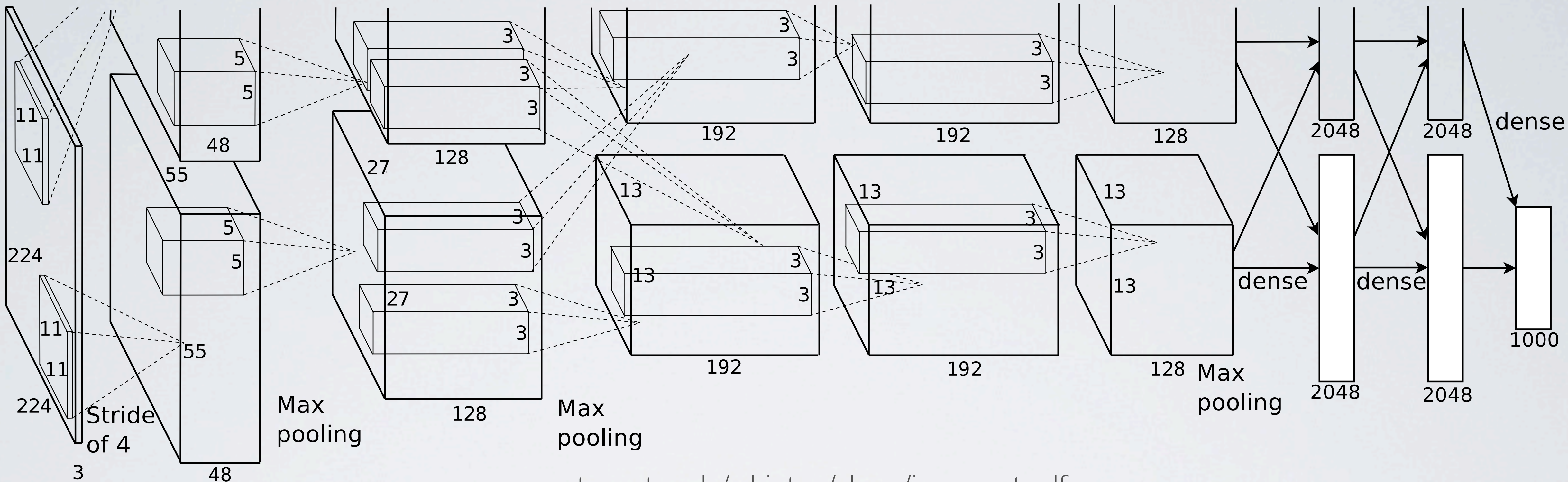
Geoffrey Hinton, Coursera class

ON THE ARCHITECTURE



www.cs.toronto.edu/~hinton/absps/imagenet.pdf

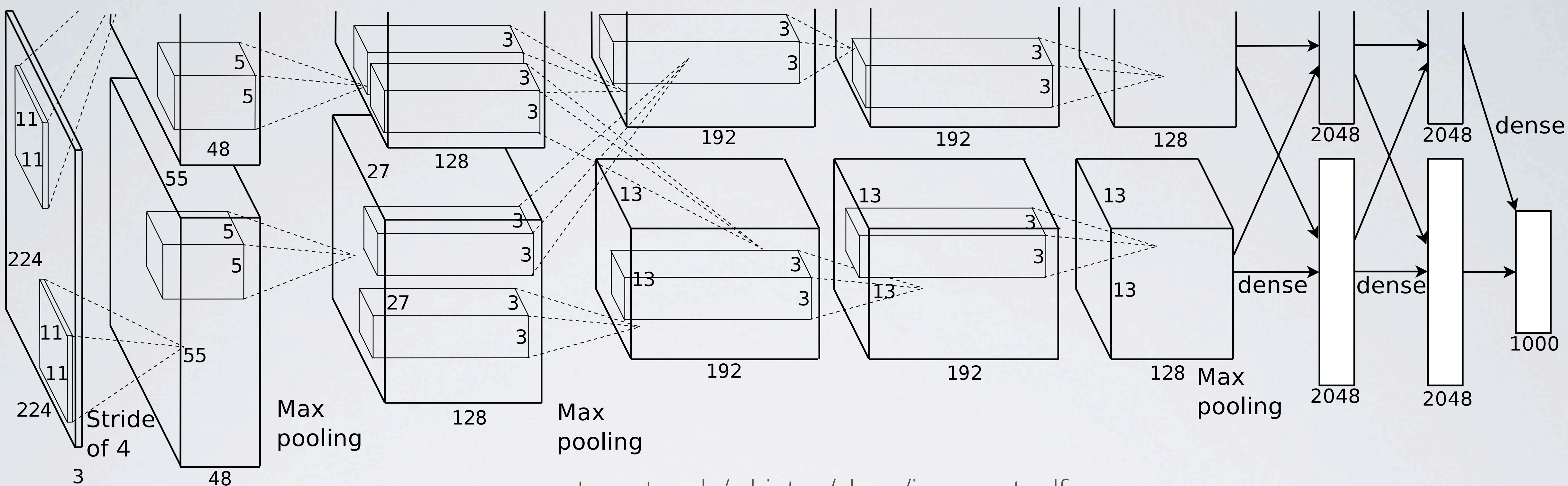
ON THE ARCHITECTURE



www.cs.toronto.edu/~hinton/absps/imagenet.pdf

typically hand-tuned

ON THE ARCHITECTURE

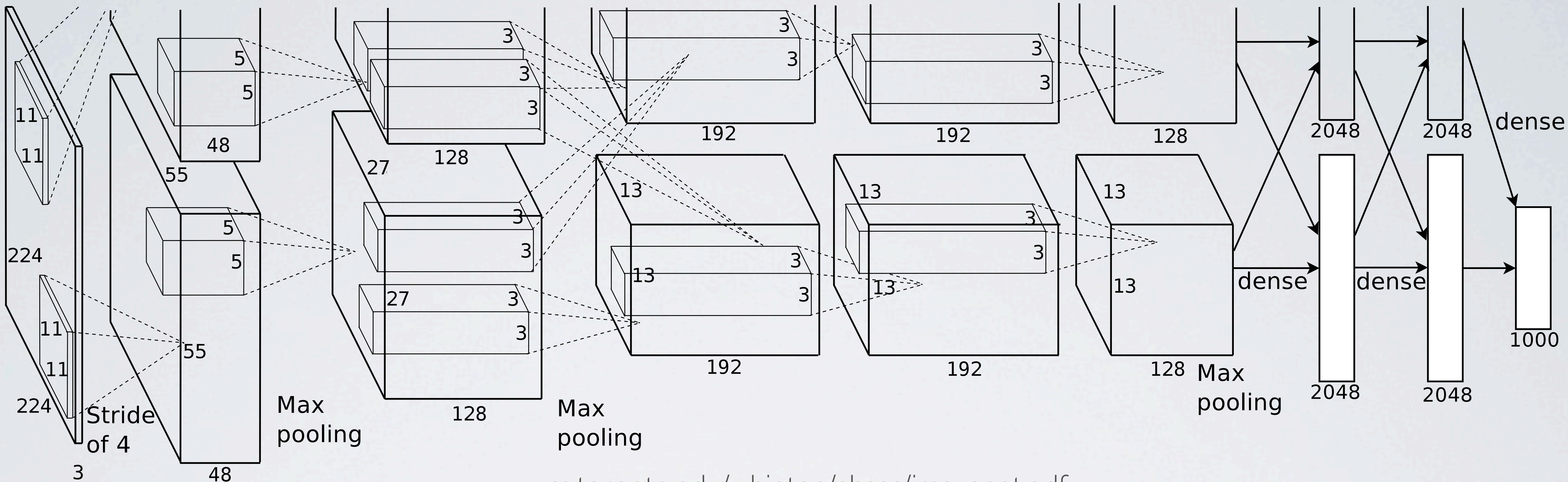


www.cs.toronto.edu/~hinton/absps/imagenet.pdf

typically hand-tuned

critical in the method's performance

ON THE ARCHITECTURE



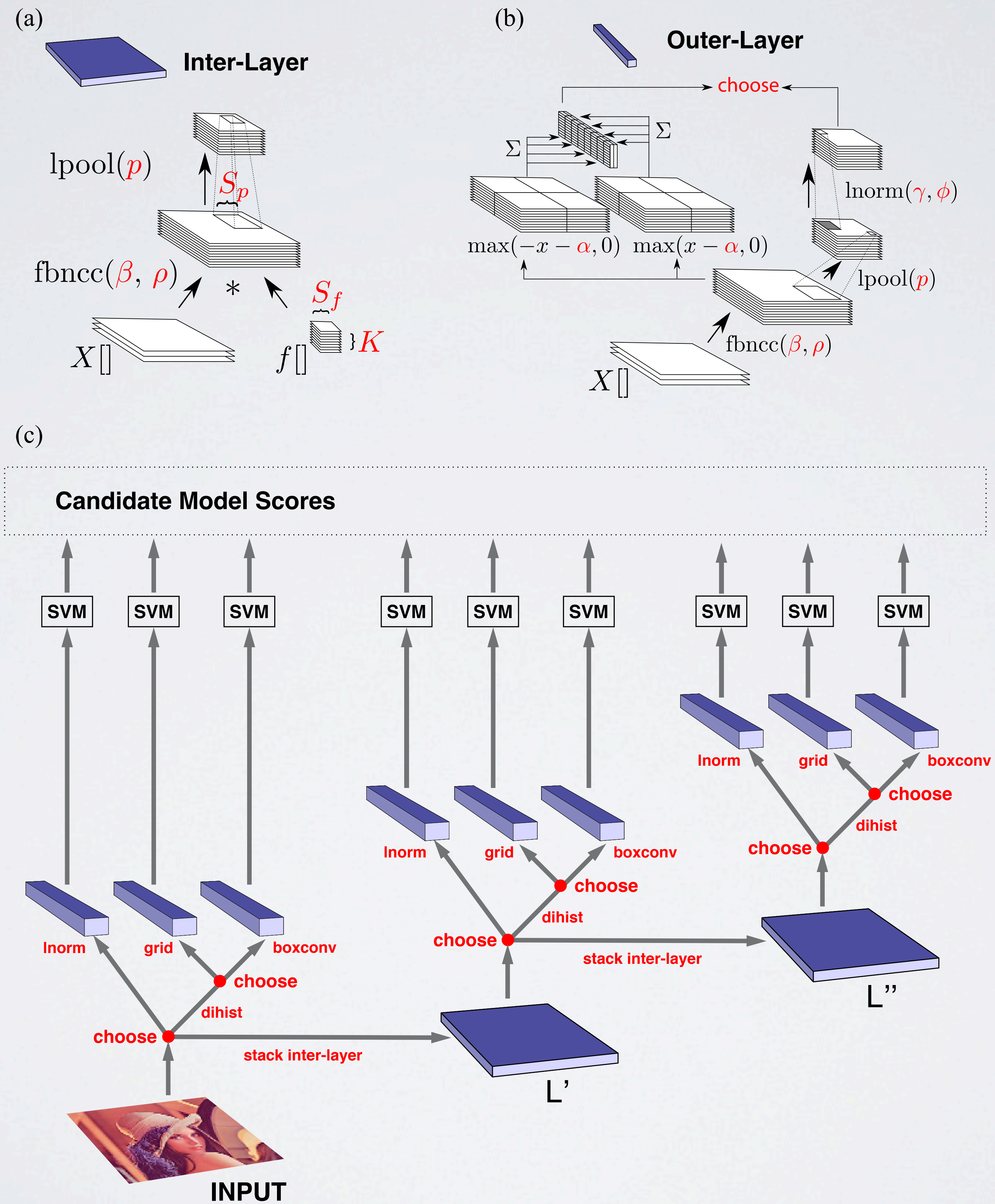
www.cs.toronto.edu/~hinton/absps/imagenet.pdf

typically hand-tuned

critical in the method's performance

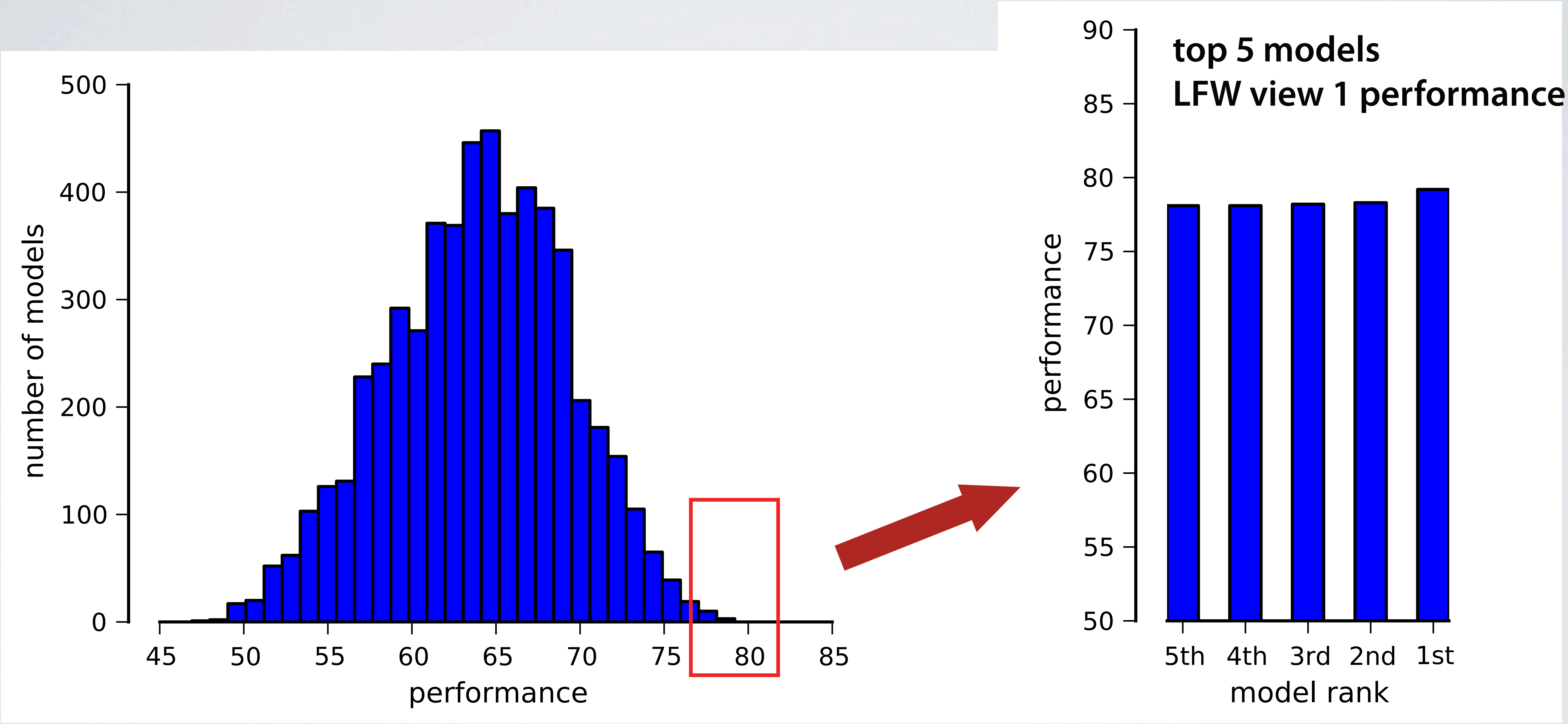
complicated search space

ON THE ARCHITECTURE



Bergstra et al, 2013

ON THE ARCHITECTURE



Pinto and Cox, 2013

QUICK LAB

using an alternative notation

CONVOLUTION+ACTIVATION

the filtering operation of an input $\mathbf{n}$ with a bank of k filters is

$$\mathbf{f}_i = \mathbf{n} \otimes \Phi_i \quad \forall i \in \{1, 2, \ldots, k\},$$

where $\otimes$ is a 3D convolution sliding over the first two dimensions, and $\Phi_i \in \mathbb{R}^{fh \times fw \times fd}$ is one such filter of our filter bank

and the rectified linear activation is

$$\mathbf{a}_i = \max(0, \mathbf{f}_i)$$

POOLING

the pooling operation with strength p and spatial downsampling of α is

$$\mathbf{p}_i = \text{downsample}_\alpha\left(\sqrt[p]{(\mathbf{a}_i)^p \odot \mathbf{1}_{ph \times pw}}\right),$$

where $\odot$ is a 2D convolution sliding over both dimensions and $ph \times pw$ is the pooling neighborhood

DIVISIVE NORMALIZATION

DIVISIVE NORMALIZATION

*"In biology, initial interests in DN focused on its ability to model dynamic gain control in **retina** [24] and the “masking” behavior in **perception** [11, 33], and to fit neural recordings from the **mammalian visual cortex** [12, 19]."*

Lyu, 2010

DIVISIVE NORMALIZATION

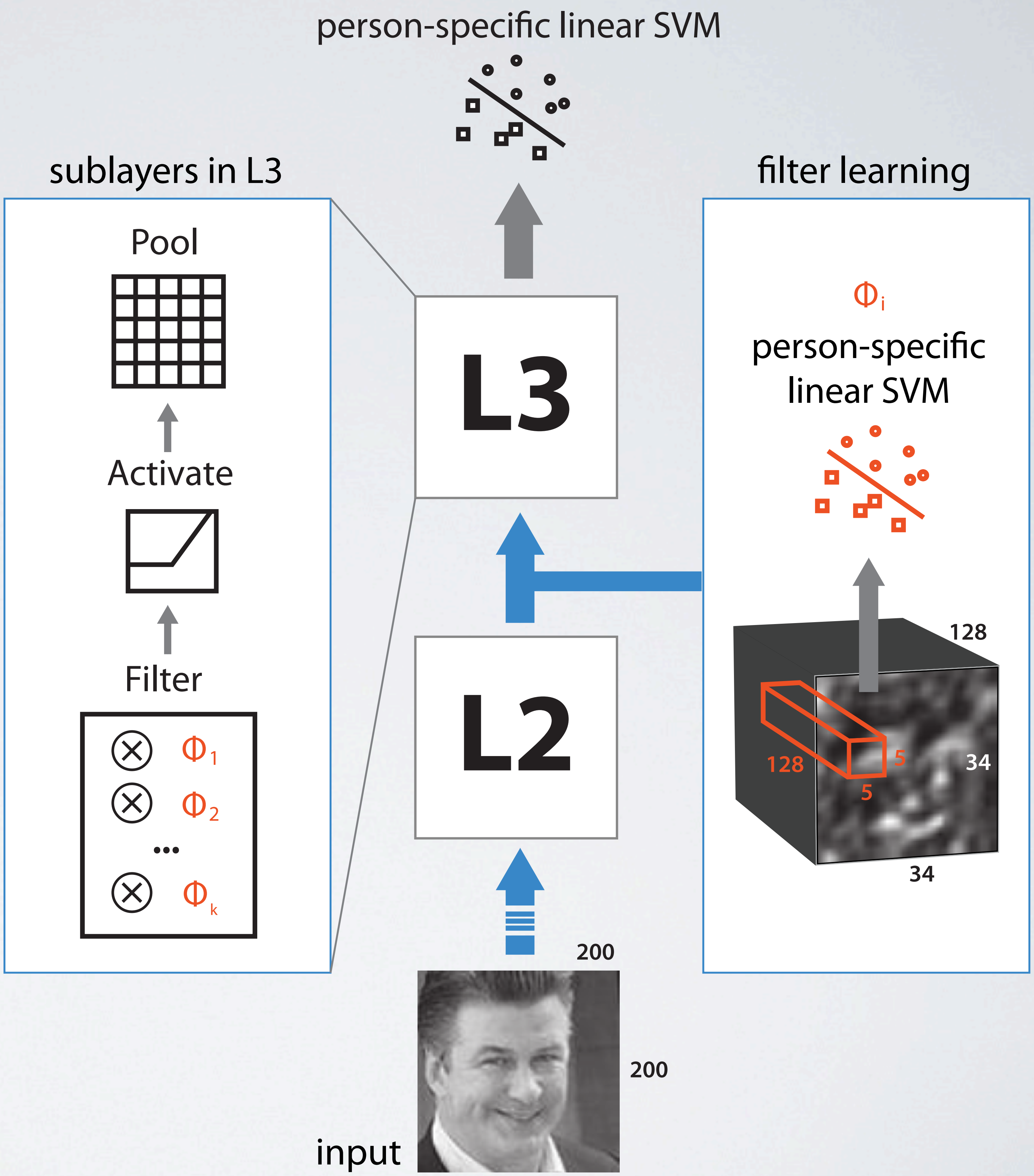
finally, the divisive normalization of an input $\mathbf{x} \in \mathbb{R}^{xh \times xw \times xd}$ is

$$\mathbf{n} = \frac{\mathbf{x}}{\sqrt{\mathbf{x}^2 \otimes \mathbf{1}_{nh \times nw \times nd}}},$$

where $\mathbf{1}_{nh \times nw \times xd}$ is a matrix of ones representing the normalization neighborhood let’s get our hands dirty!

DEEP PS

approach
from my
Ph.D. thesis



FACE IDENTIFICATION

questions?