Neural networks

Feedforward neural network - artificial neuron

ARTIFICIAL NEURON

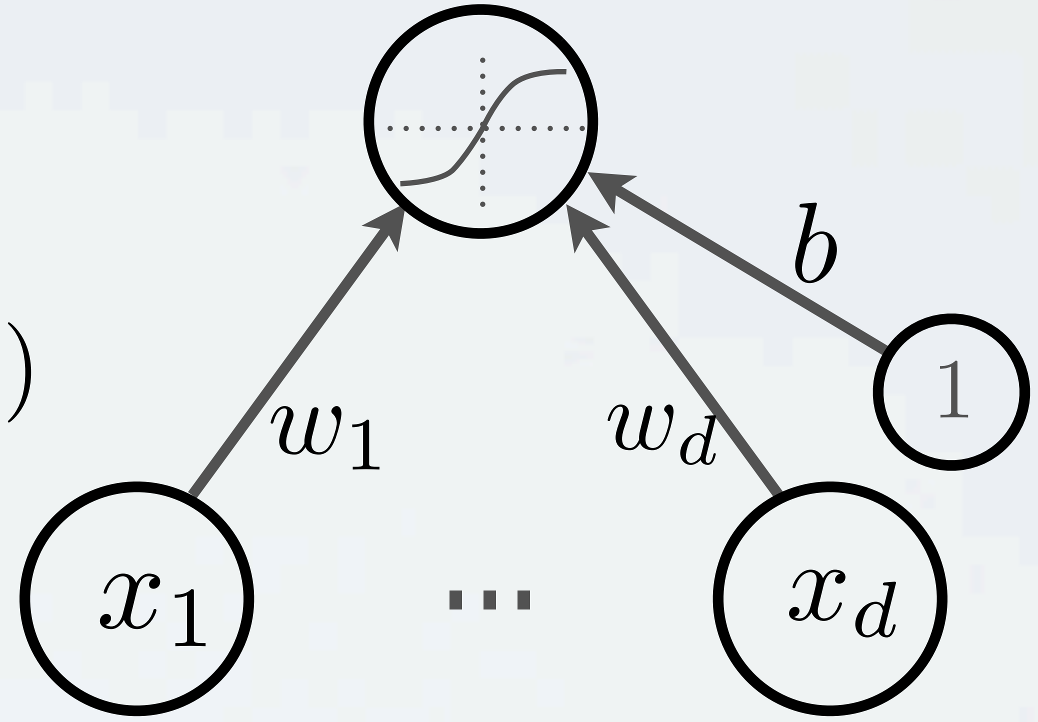
Topics: connection weights, bias, activation function

- Neuron pre-activation (or input activation):

$$a(\mathbf{x}) = b + \sum_i w_i x_i = b + \mathbf{w}^\top \mathbf{x}$$

- Neuron (output) activation

$$h(\mathbf{x}) = g(a(\mathbf{x})) = g(b + \sum_i w_i x_i)$$

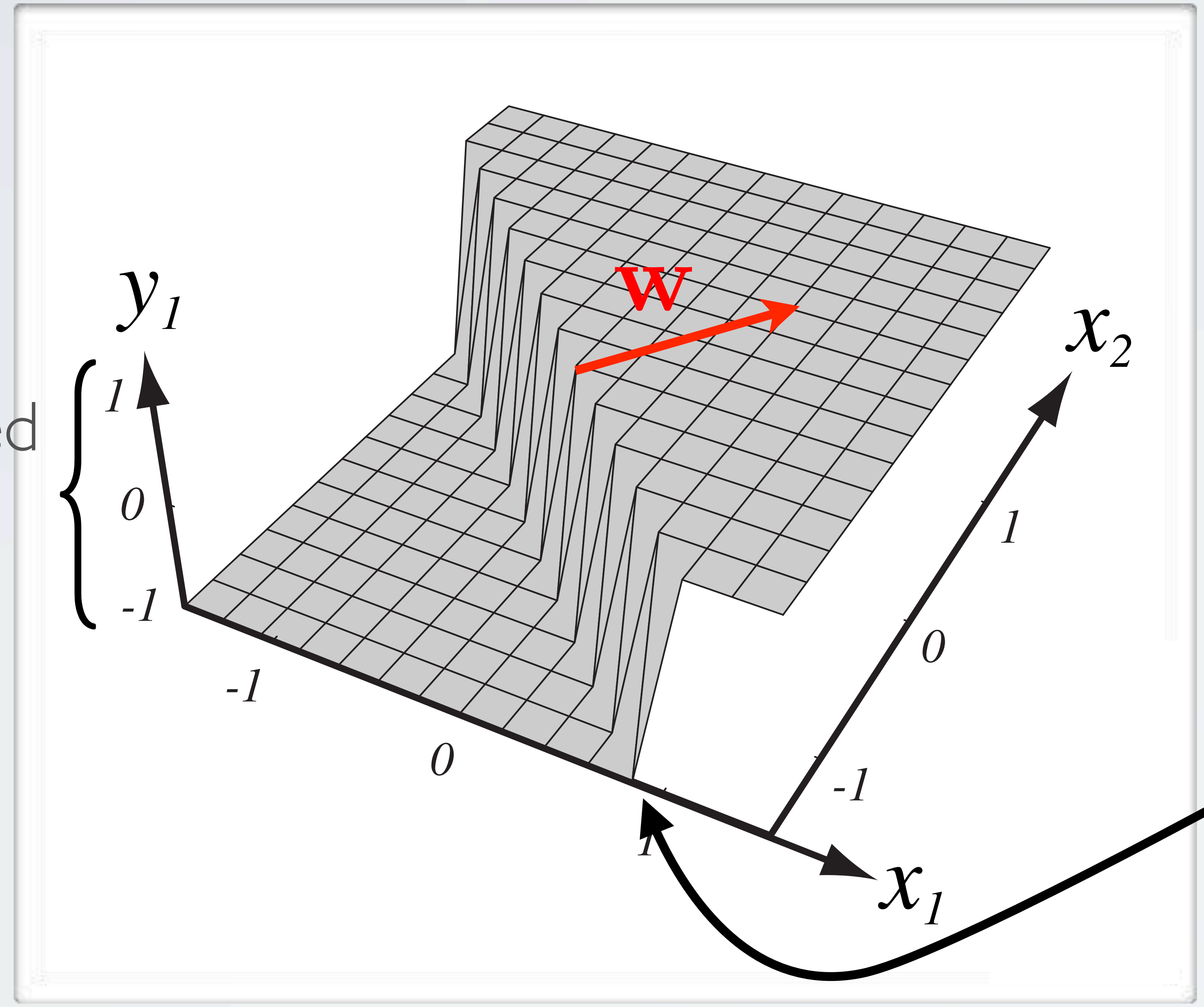


- $\mathbf{w}$ are the connection weights
- b is the neuron bias
- $g(\cdot)$ is called the activation function

ARTIFICIAL NEURON

Topics: connection weights, bias, activation function

range determined by $g(\cdot)$



bias b only changes the position of the riff

(from Pascal Vincent's slides)

Neural networks

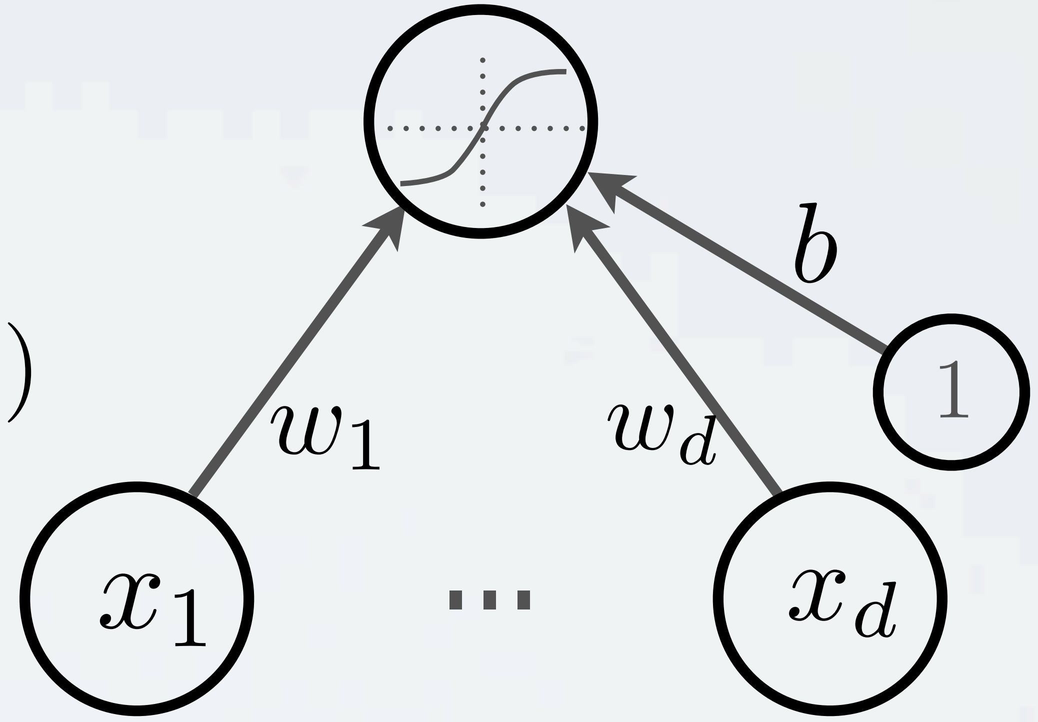
Feedforward neural network - activation function

ARTIFICIAL NEURON

Topics: connection weights, bias, activation function

- Neuron pre-activation (or input activation):

$$a(\mathbf{x}) = b + \sum_i w_i x_i = b + \mathbf{w}^\top \mathbf{x}$$



- Neuron (output) activation

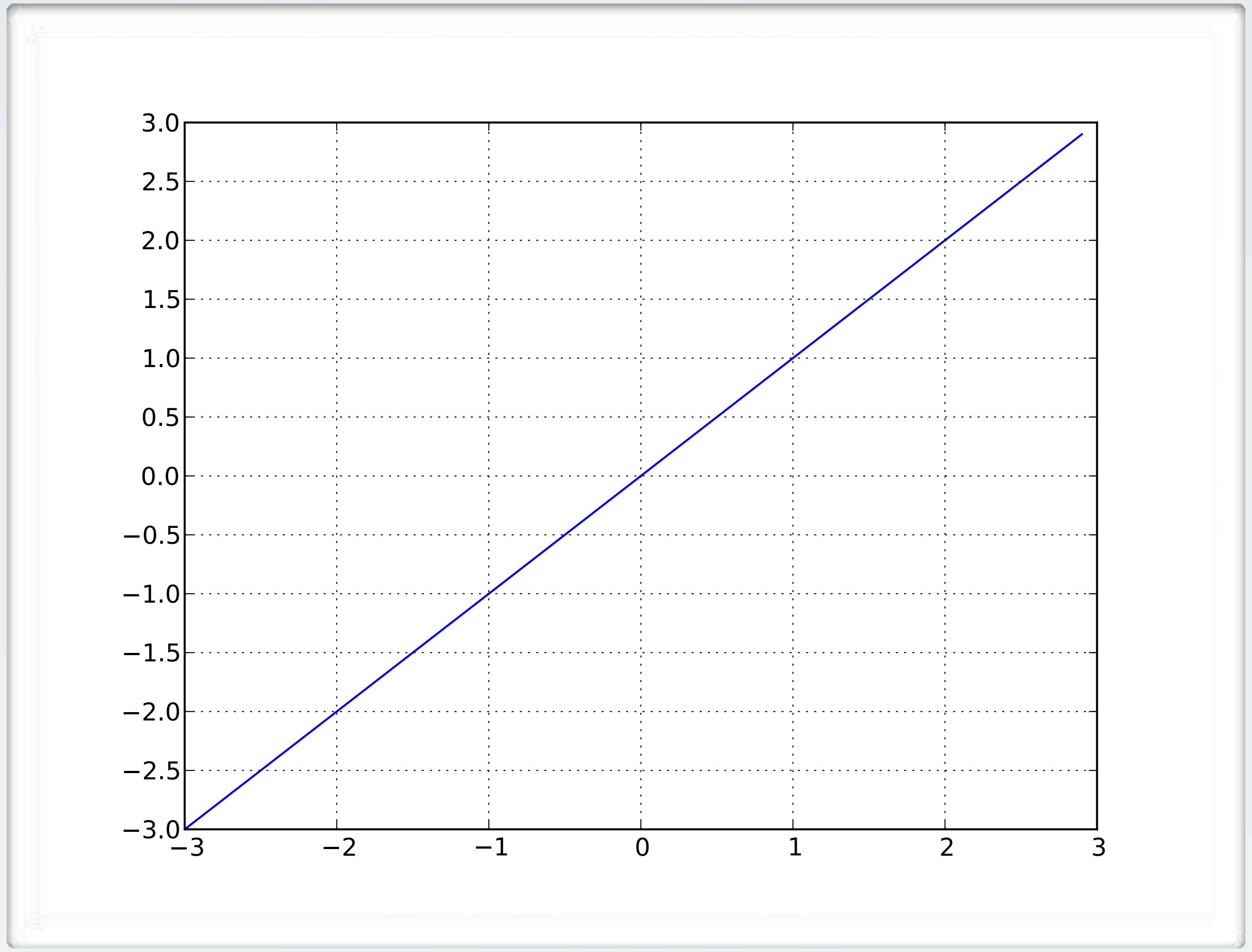
$$h(\mathbf{x}) = g(a(\mathbf{x})) = g(b + \sum_i w_i x_i)$$

- $\mathbf{w}$ are the connection weights
- b is the neuron bias
- $g(\cdot)$ is called the activation function

ACTIVATION FUNCTION

Topics: linear activation function

- Performs no input squashing
- Not very interesting...

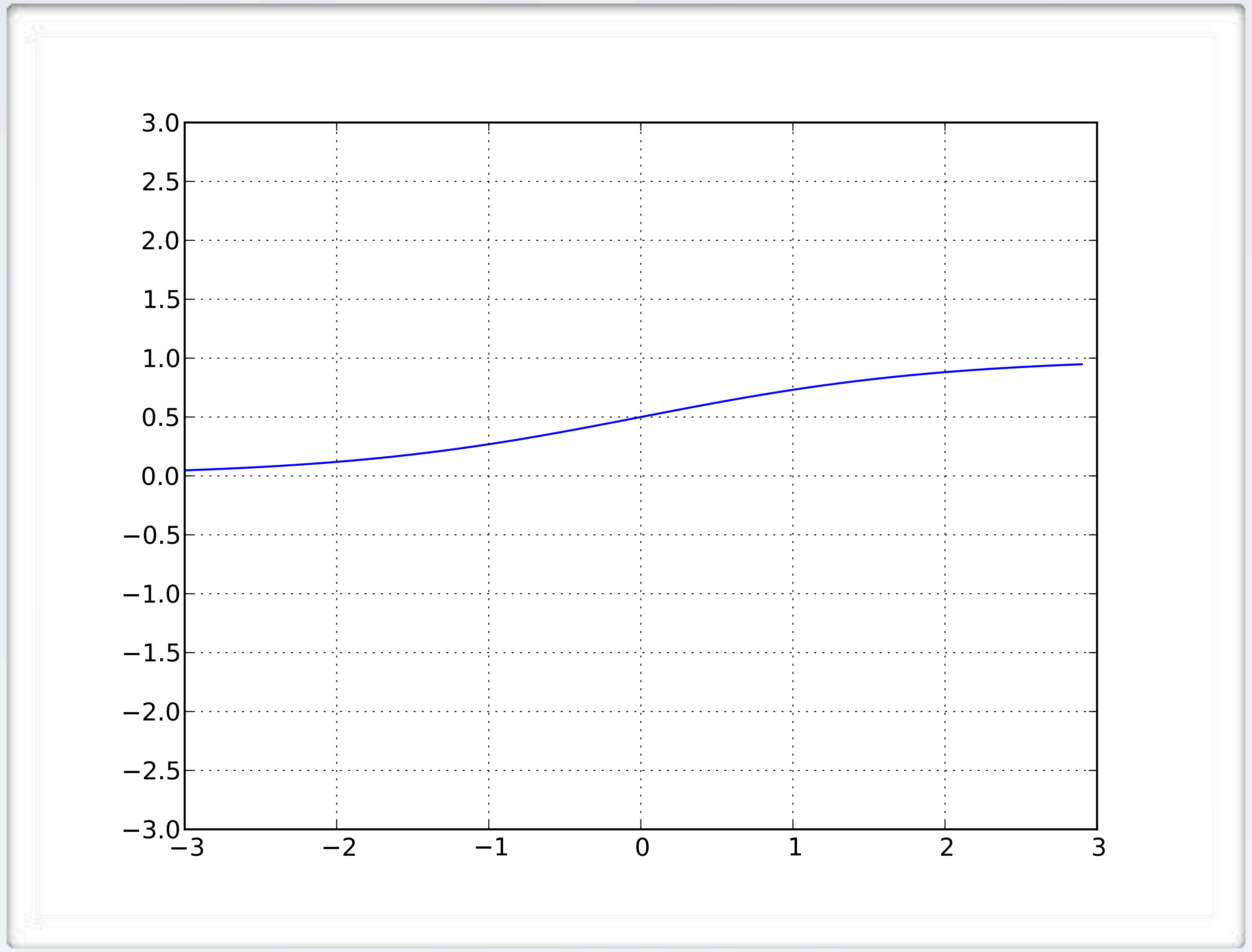


$$g(a) = a$$

ACTIVATION FUNCTION

Topics: sigmoid activation function

- Squashes the neuron's pre-activation between 0 and 1
- Always positive
- Bounded
- Strictly increasing

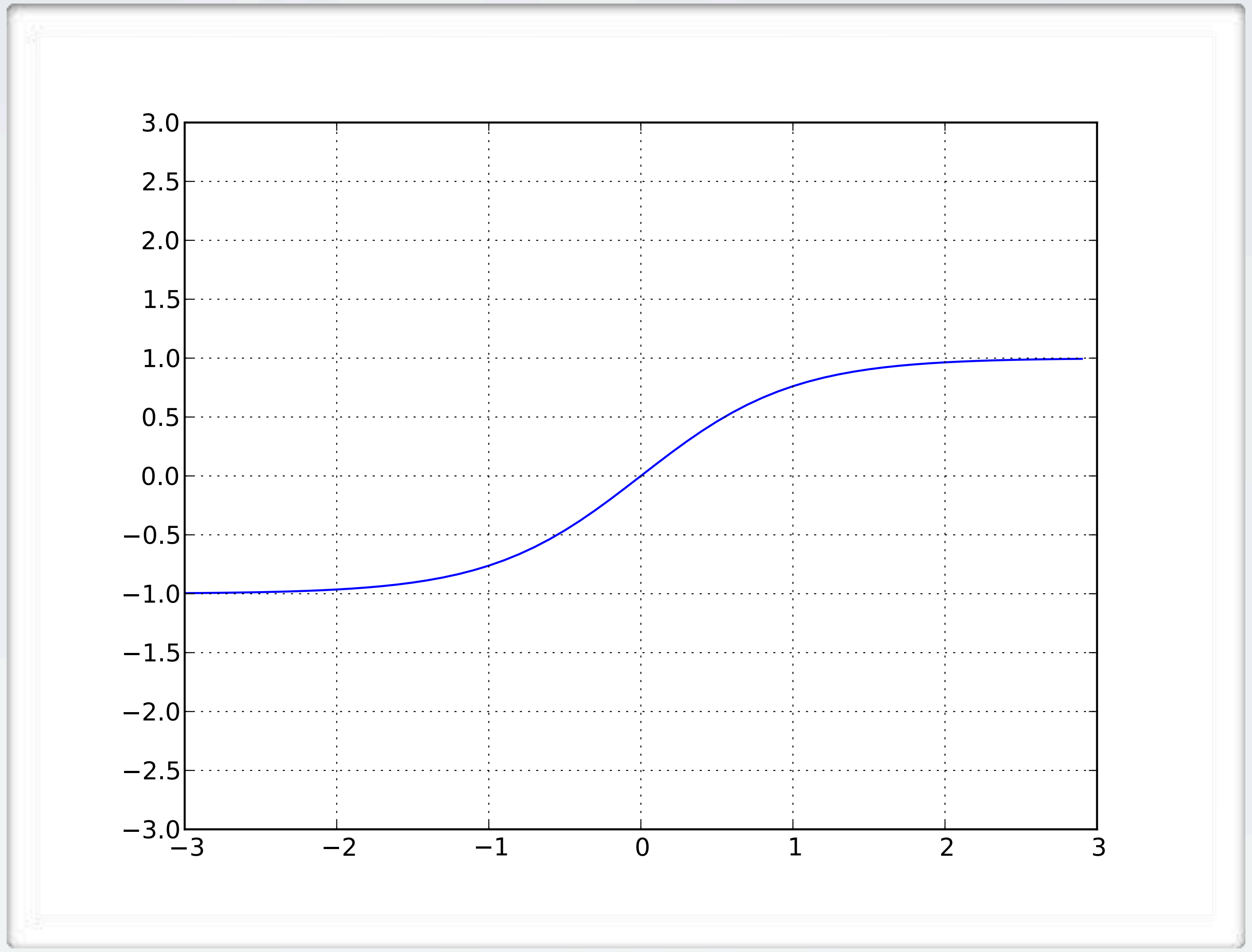


$$g(a) = \mathrm{sigm}(a) = \frac{1}{1+\exp(-a)}$$

ACTIVATION FUNCTION

Topics: hyperbolic tangent ("tanh") activation function

- Squashes the neuron's pre-activation between -1 and 1
- Can be positive or negative
- Bounded
- Strictly increasing

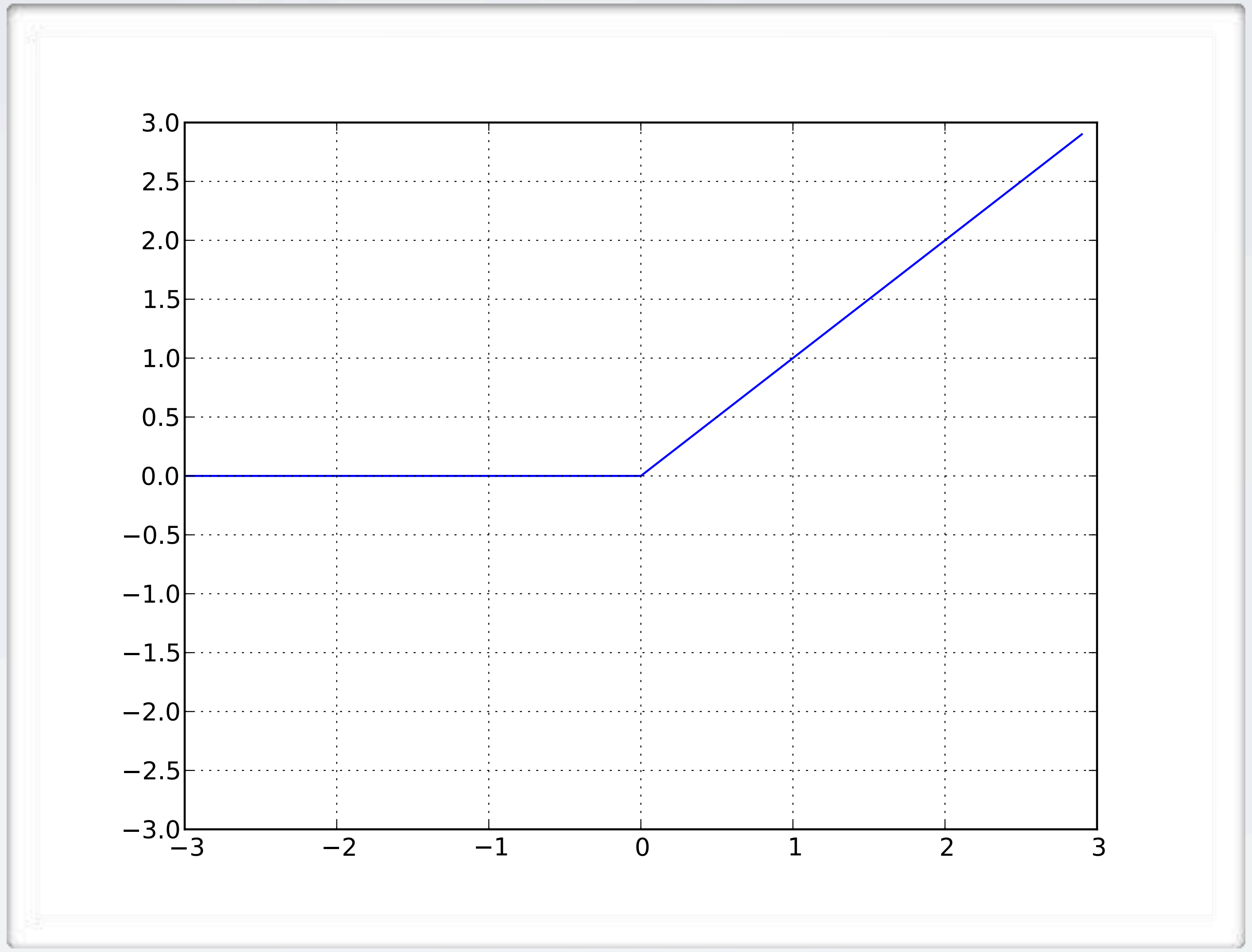


$$g(a) = \tanh(a) = \frac{\exp(a)-\exp(-a)}{\exp(a)+\exp(-a)} = \frac{\exp(2a)-1}{\exp(2a)+1}$$

ACTIVATION FUNCTION

Topics: rectified linear activation function

- Bounded below by 0 (always non-negative)
- Not upper bounded
- Strictly increasing
- Tends to give neurons with sparse activities



$$g(a) = \mathrm{reclin}(a) = \max(0, a)$$

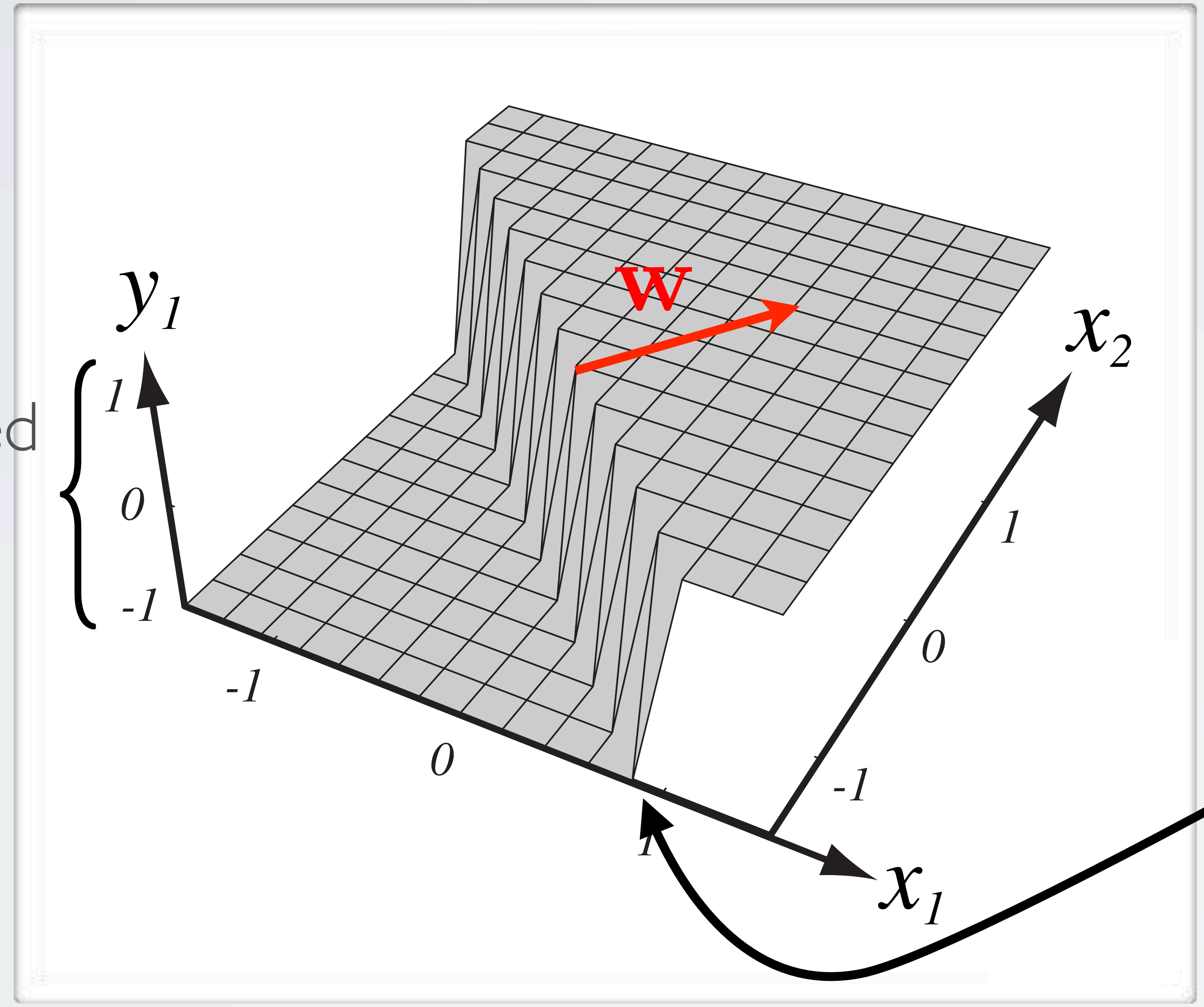
Neural networks

Feedforward neural network - capacity of single neuron

ARTIFICIAL NEURON

Topics: connection weights, bias, activation function

range determined by $g(\cdot)$



bias b only changes the position of the riff

(from Pascal Vincent's slides)

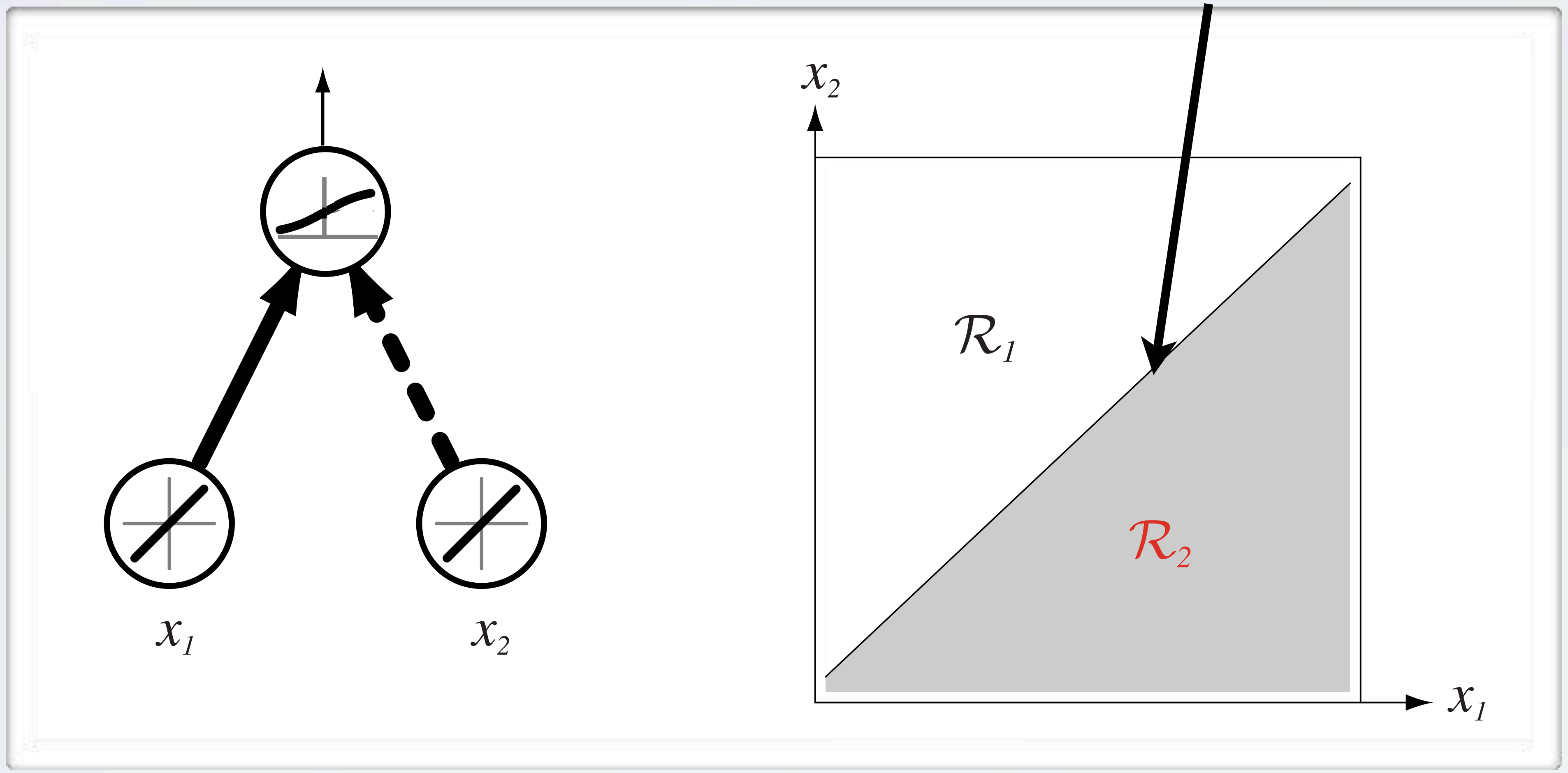
ARTIFICIAL NEURON

Topics: capacity, decision boundary of neuron

- Could do binary classification:
  - with sigmoid, can interpret neuron as estimating $p(y = 1|\mathbf{x})$
  - also known as logistic regression classifier
  - if greater than 0.5, predict class 1
  - otherwise, predict class 0

(similar idea can apply with tanh)

decision boundary is linear

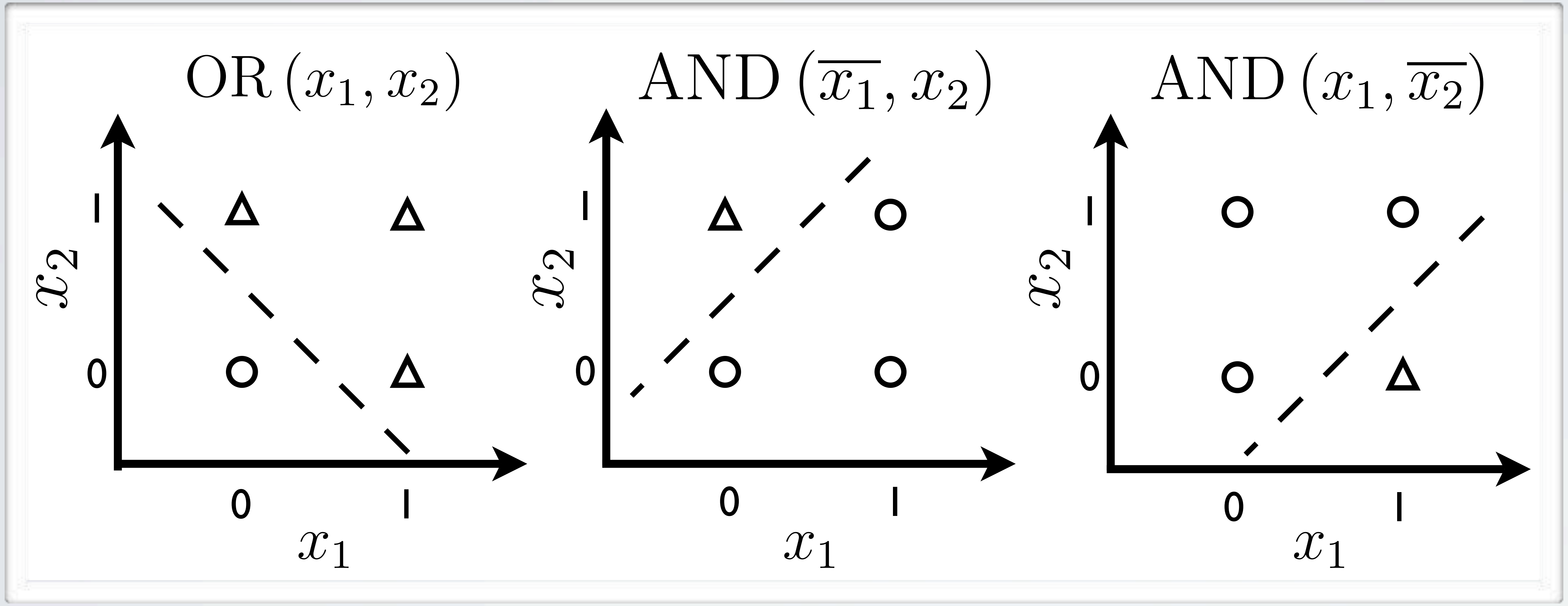


(from Pascal Vincent's slides)

ARTIFICIAL NEURON

Topics: capacity of single neuron

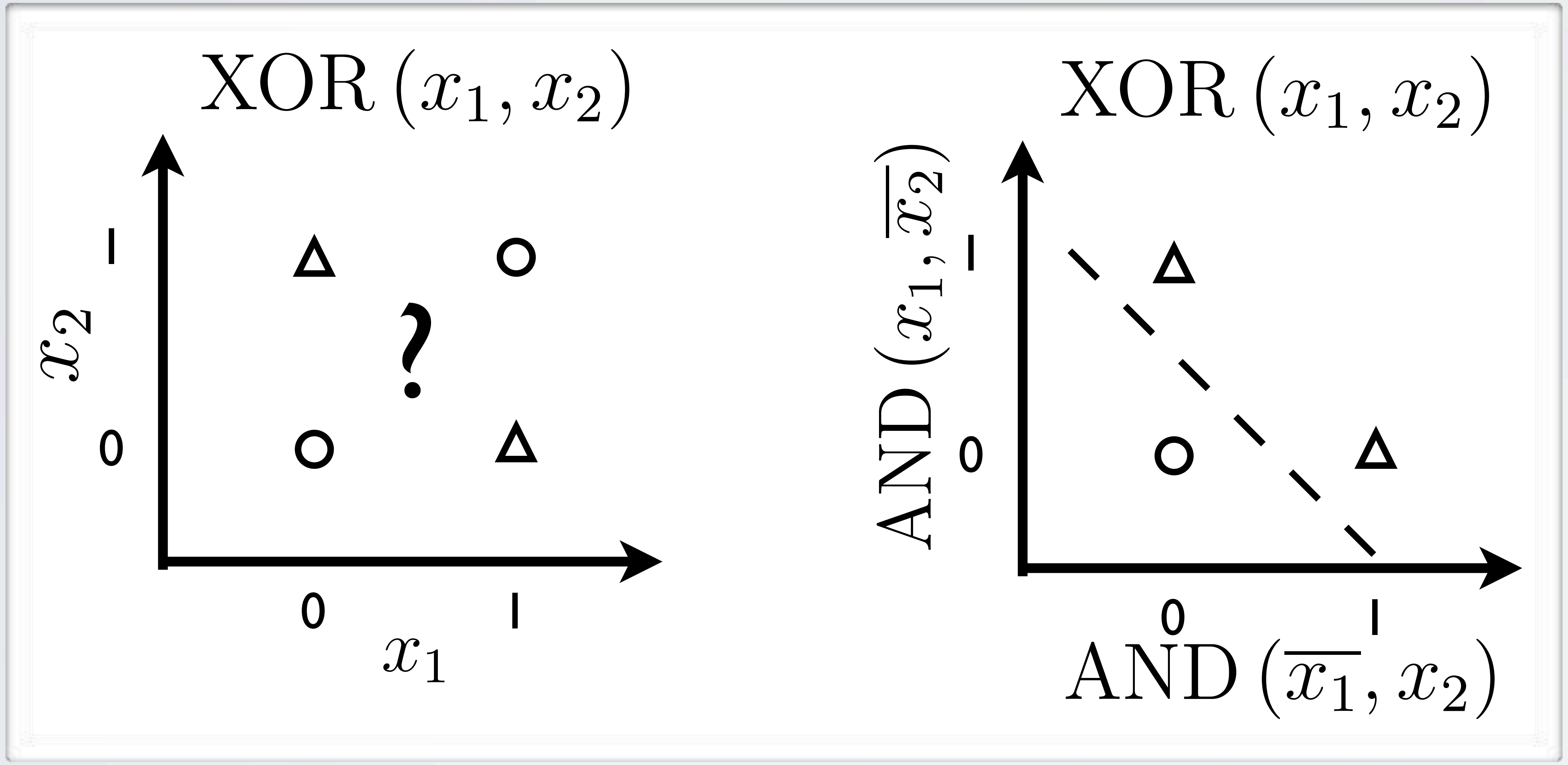
- Can solve linearly separable problems



ARTIFICIAL NEURON

Topics: capacity of single neuron

- Can't solve non linearly separable problems...



- ... unless the input is transformed in a better representation

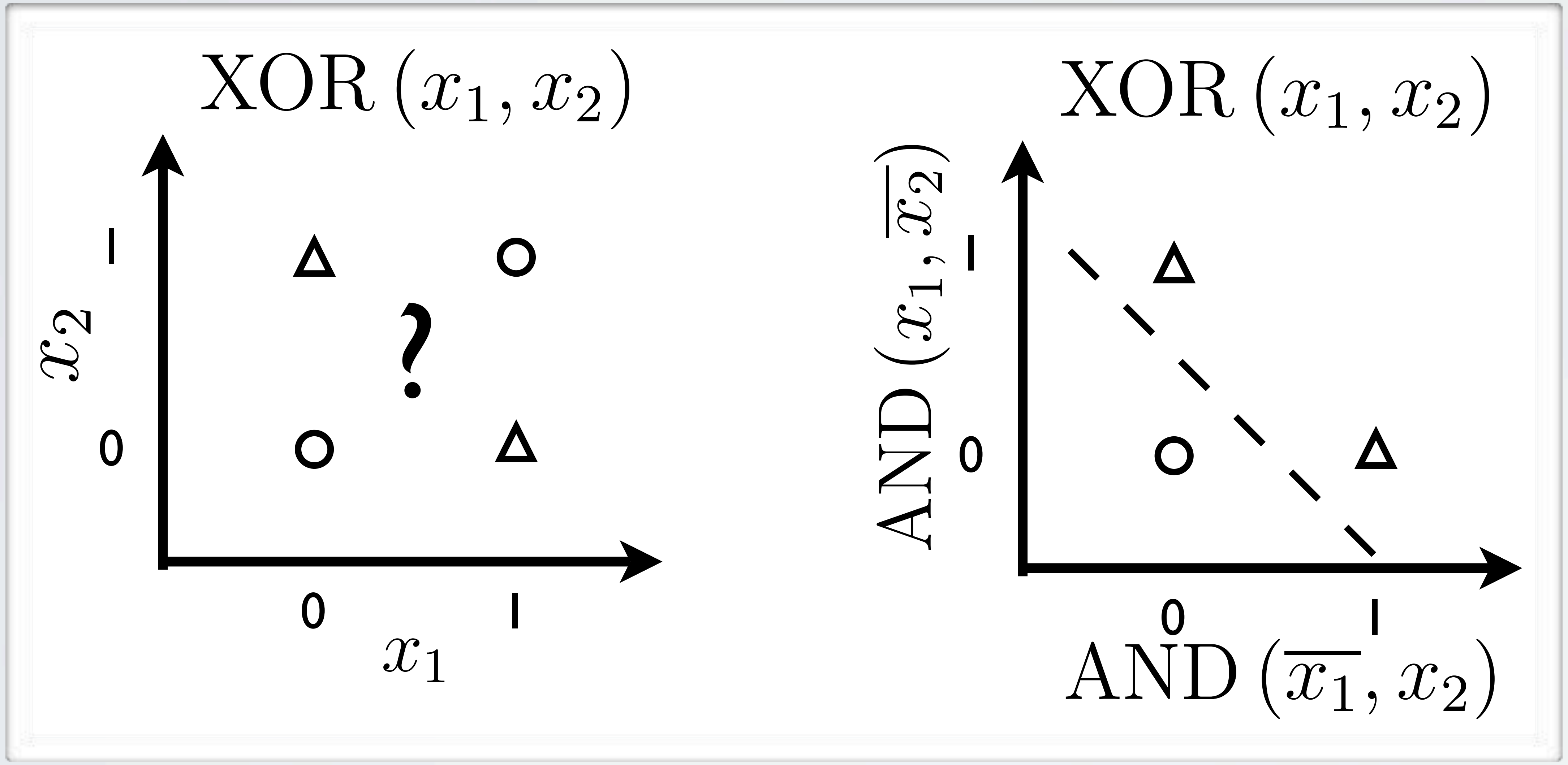
Neural networks

Feedforward neural network - multilayer neural network

ARTIFICIAL NEURON

Topics: capacity of single neuron

- Can't solve non linearly separable problems...



- ... unless the input is transformed in a better representation

NEURAL NETWORK

Topics: single hidden layer neural network

- Hidden layer pre-activation:

$$\mathbf{a}(\mathbf{x}) = \mathbf{b}^{(1)} + \mathbf{W}^{(1)}\mathbf{x}$$

$$\left(a(\mathbf{x})_i = b_i^{(1)} + \sum_j W_{i,j}^{(1)} x_j \right)$$

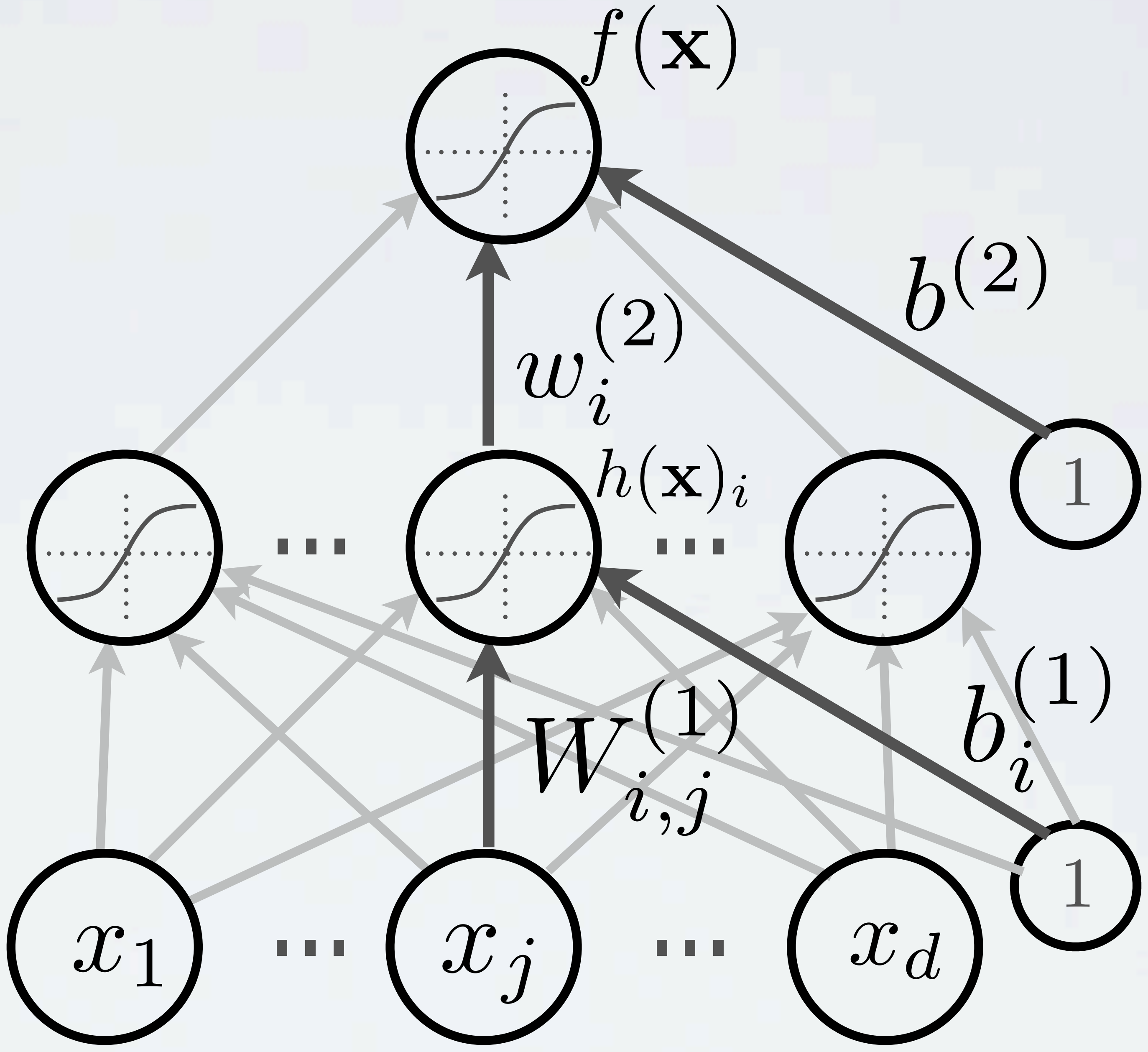
- Hidden layer activation:

$$\mathbf{h}(\mathbf{x}) = \mathbf{g}(\mathbf{a}(\mathbf{x}))$$

- Output layer activation:

$$f(\mathbf{x}) = o\left(b^{(2)} + \mathbf{w}^{(2)\top} \mathbf{h}^{(1)} \mathbf{x} \right)$$

output activation function



NEURAL NETWORK

Topics: softmax activation function

- For multi-class classification:
  - we need multiple outputs (1 output per class)
  - we would like to estimate the conditional probability $p(y = c|\mathbf{x})$

- We use the softmax activation function at the output:

$$\mathbf{o}(\mathbf{a}) = \mathrm{softmax}(\mathbf{a}) = \left[\frac{\exp(a_1)}{\sum_c \exp(a_c)} \ldots \frac{\exp(a_C)}{\sum_c \exp(a_c)}\right]^\top$$

  - strictly positive
  - sums to one

- Predicted class is the one with highest estimated probability

NEURAL NETWORK

Topics: multilayer neural network

- Could have L hidden layers:
  - layer pre-activation for $k{>}0$ $\quad(\mathbf{h}^{(0)}(\mathbf{x}) = \mathbf{x})$

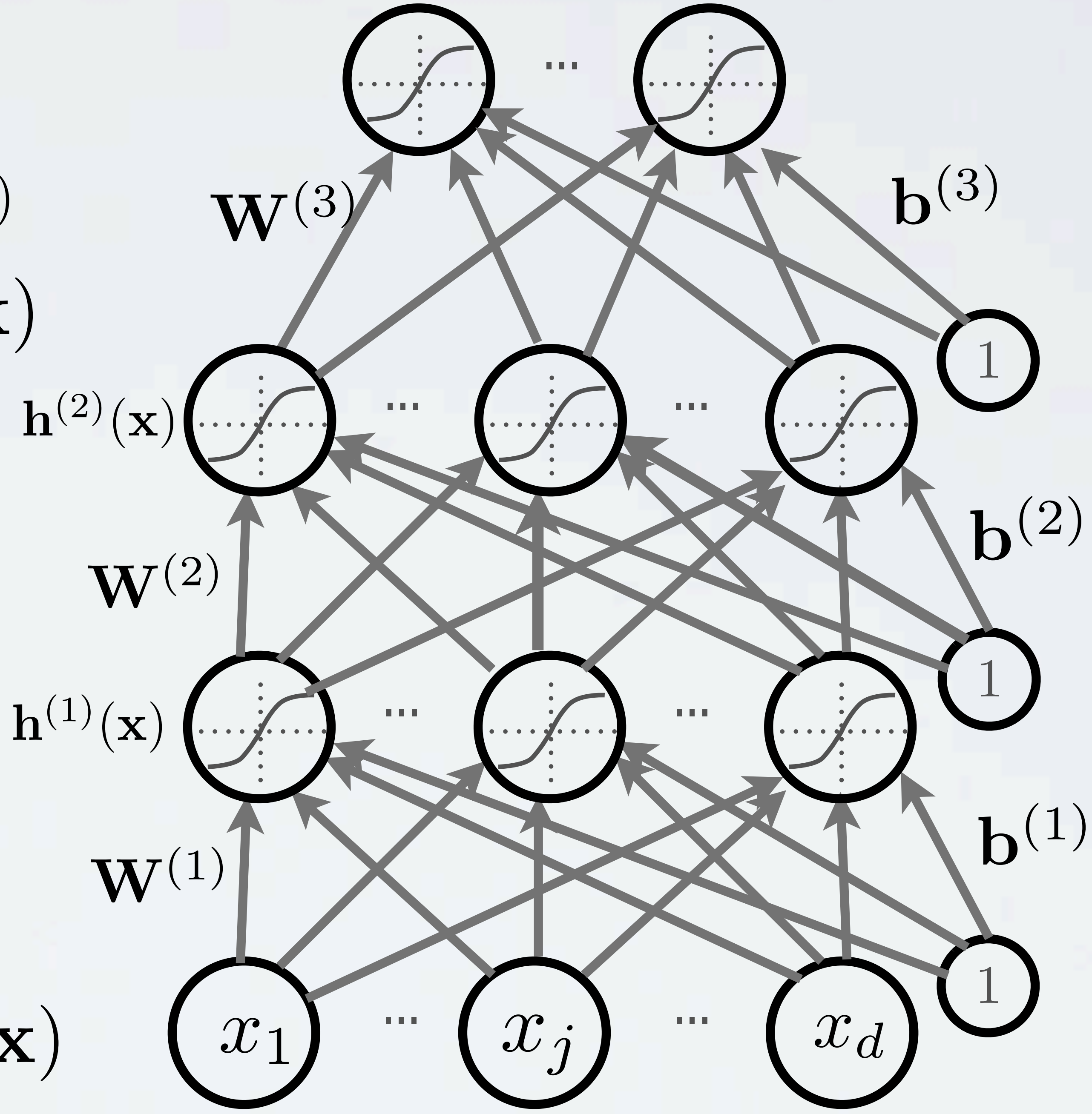
$$\mathbf{a}^{(k)}(\mathbf{x}) = \mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{h}^{(k-1)}(\mathbf{x})$$

  - hidden layer activation (k from 1 to L):

$$\mathbf{h}^{(k)}(\mathbf{x}) = \mathbf{g}(\mathbf{a}^{(k)}(\mathbf{x}))$$

  - output layer activation ($k{=}L{+}1$):

$$\mathbf{h}^{(L+1)}(\mathbf{x}) = \mathbf{o}(\mathbf{a}^{(L+1)}(\mathbf{x})) = \mathbf{f}(\mathbf{x})$$



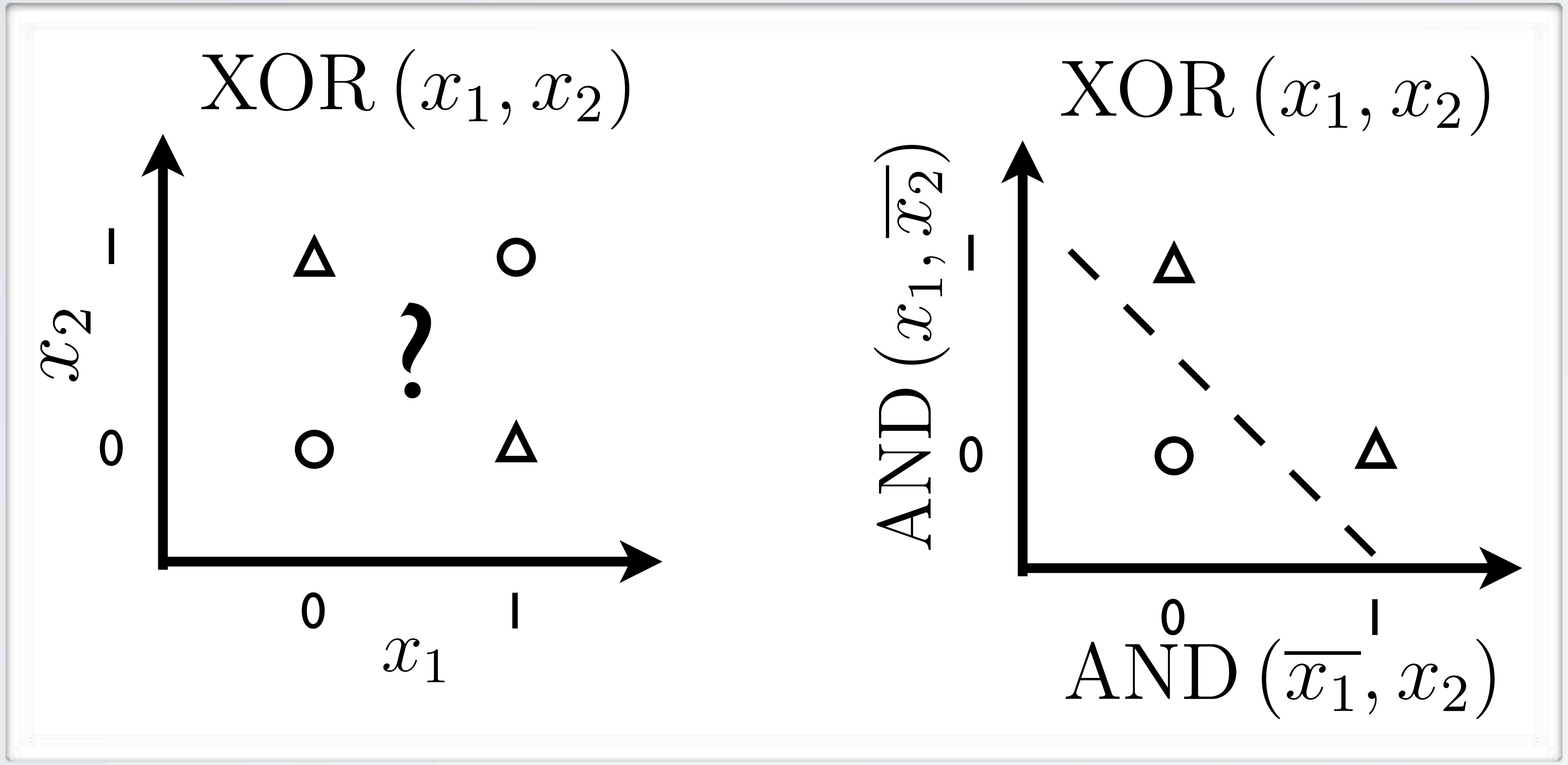
Neural networks

Feedforward neural network - capacity of neural network

ARTIFICIAL NEURON

Topics: capacity of single neuron

- Can't solve non linearly separable problems...



- ... unless the input is transformed in a better representation

NEURAL NETWORK

Topics: single hidden layer neural network

- Hidden layer pre-activation:

$$\mathbf{a}(\mathbf{x}) = \mathbf{b}^{(1)} + \mathbf{W}^{(1)}\mathbf{x}$$

$$\left(a(\mathbf{x})_i = b_i^{(1)} + \textstyle\sum_j W_{i,j}^{(1)} x_j\right)$$

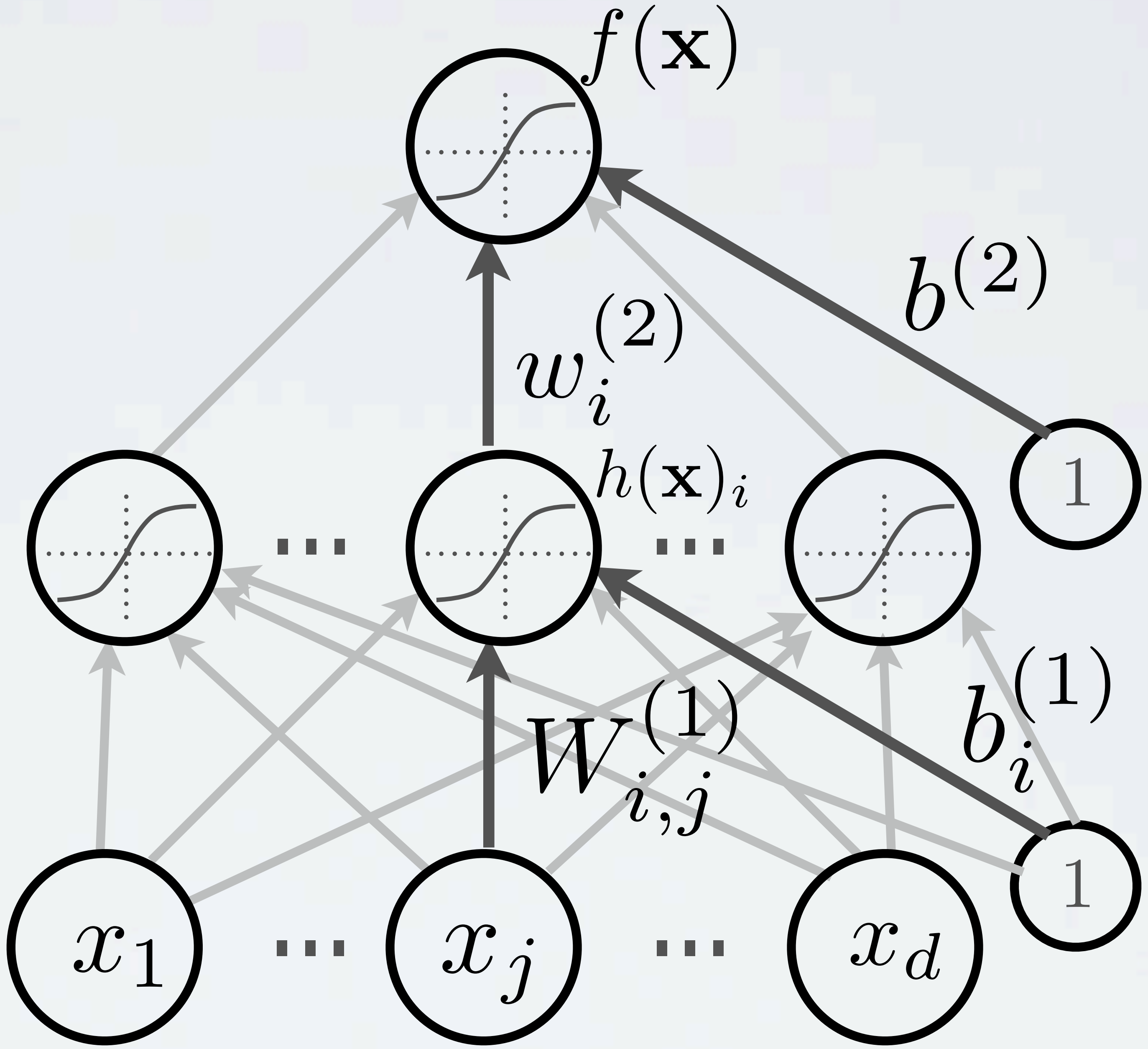
- Hidden layer activation:

$$\mathbf{h}(\mathbf{x}) = \mathbf{g}(\mathbf{a}(\mathbf{x}))$$

- Output layer activation:

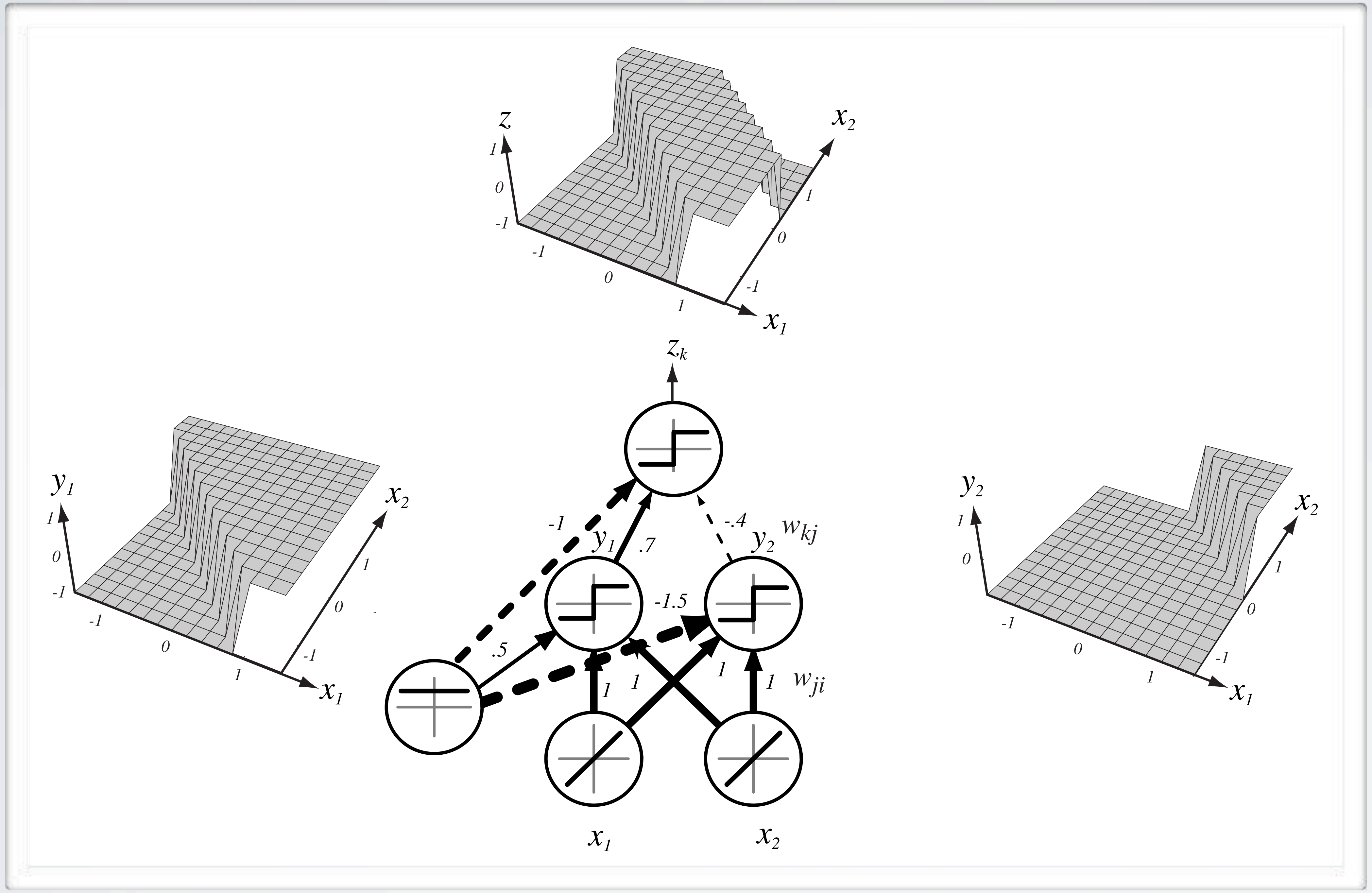
$$f(\mathbf{x}) = o\left(b^{(2)} + \mathbf{w}^{(2)}{}^{\top}\mathbf{h}^{(1)}\mathbf{x}\right)$$

output activation function



CAPACITY OF NEURAL NETWORK

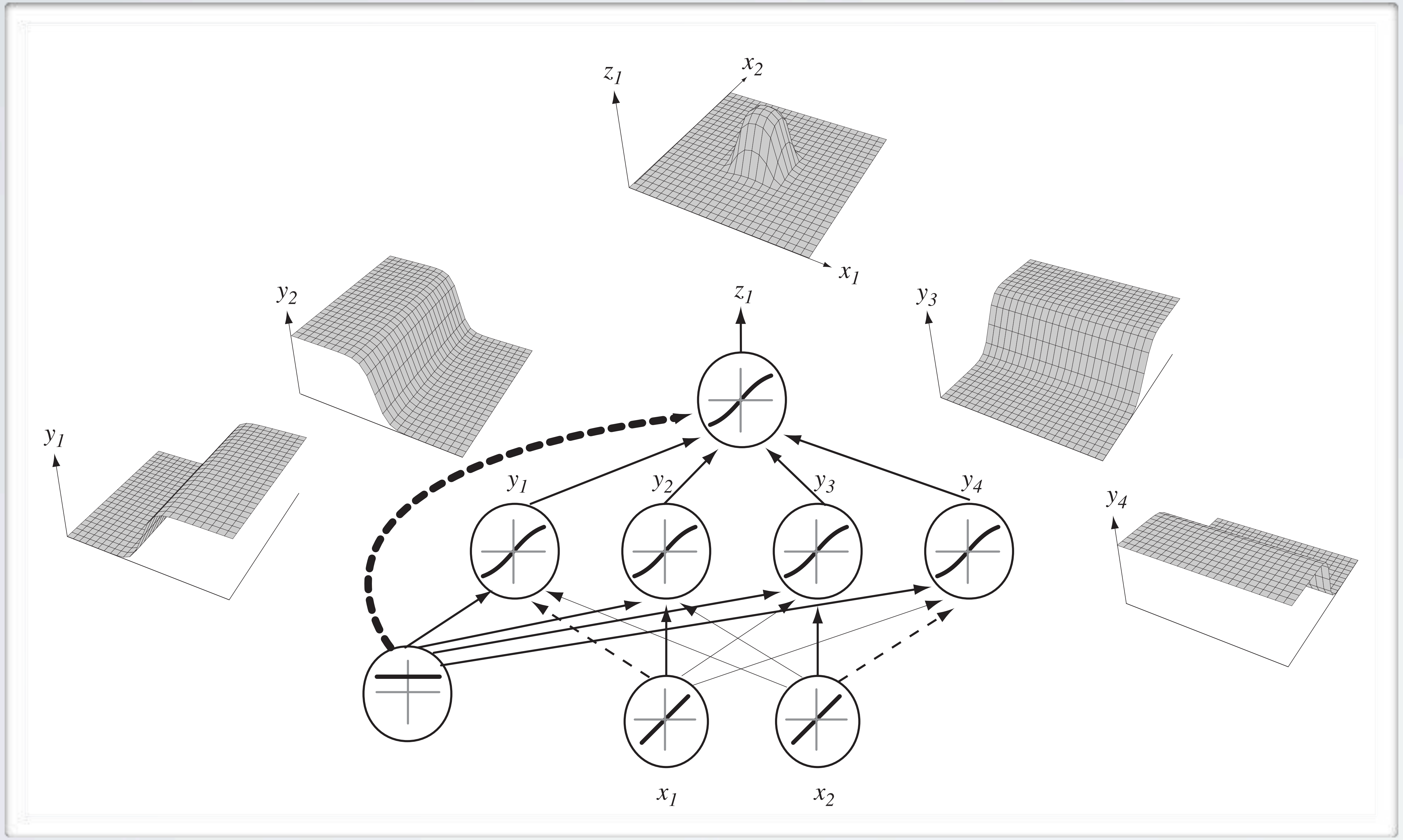
Topics: single hidden layer neural network



(from Pascal Vincent's slides)

CAPACITY OF NEURAL NETWORK

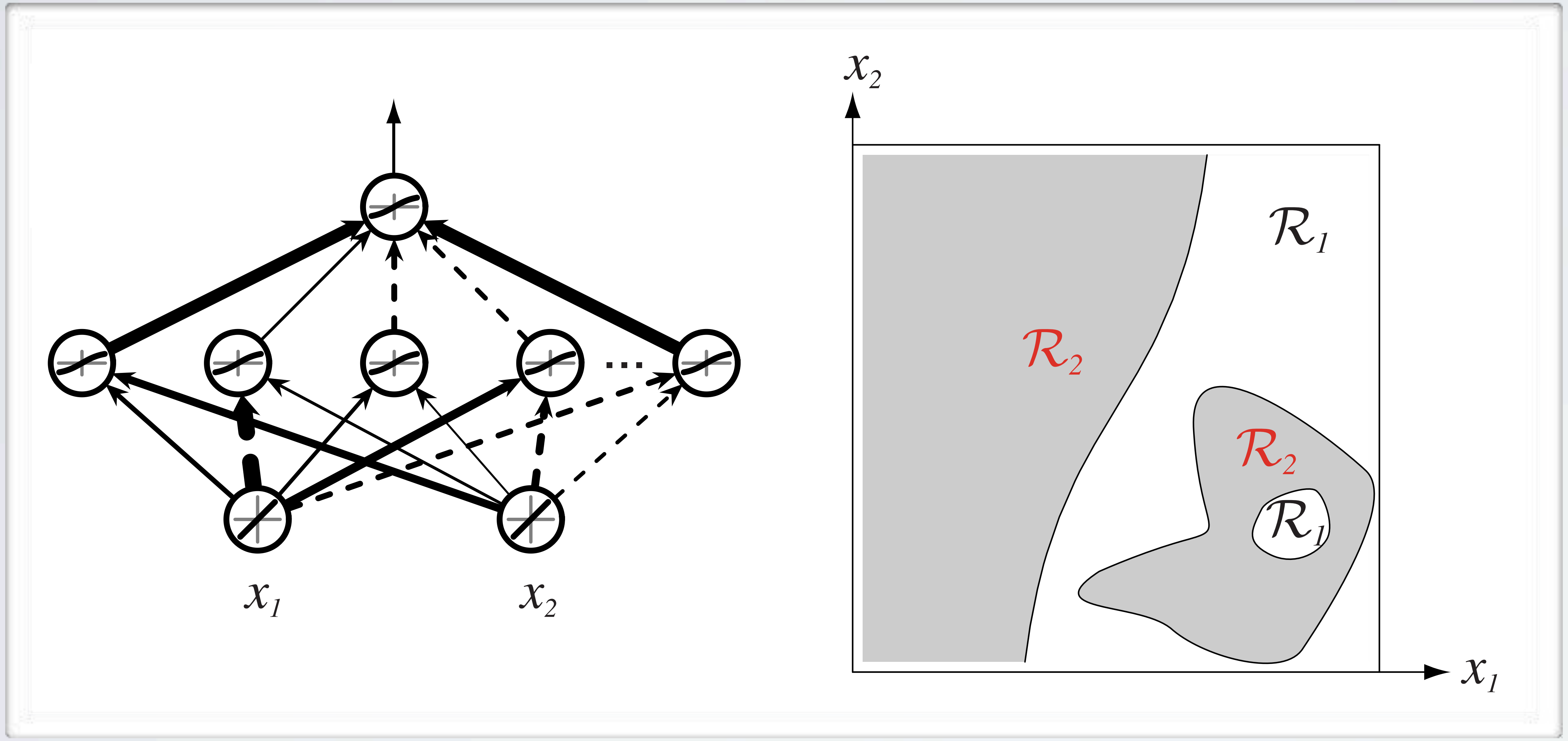
Topics: single hidden layer neural network



(from Pascal Vincent's slides)

CAPACITY OF NEURAL NETWORK

Topics: single hidden layer neural network



(from Pascal Vincent's slides)

CAPACITY OF NEURAL NETWORK

Topics: universal approximation

- Universal approximation theorem (Hornik, 1991):
  - "a single hidden layer neural network with a linear output unit can approximate any continuous function arbitrarily well, given enough hidden units"
- The result applies for sigmoid, tanh and many other hidden layer activation functions
- This is a good result, but it doesn't mean there is a learning algorithm that can find the necessary parameter values!

Neural networks

Feedforward neural network - biological inspiration

NEURAL NETWORK

Topics: multilayer neural network

- Could have L hidden layers:
  - layer pre-activation for $k>0$ $(\mathbf{h}^{(0)}(\mathbf{x}) = \mathbf{x})$

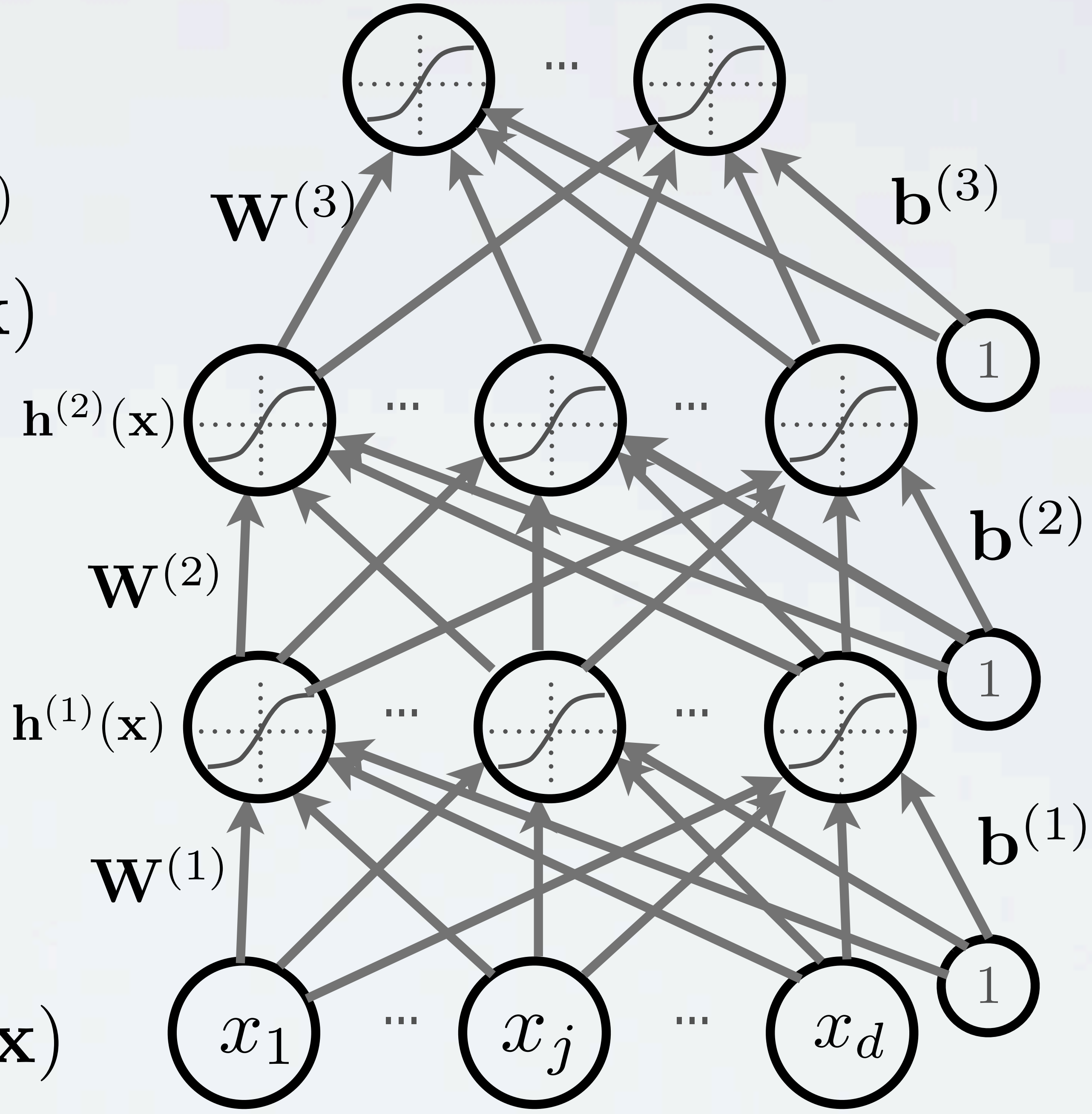
    $$\mathbf{a}^{(k)}(\mathbf{x}) = \mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{h}^{(k-1)}(\mathbf{x})$$

  - hidden layer activation (k from 1 to L):

    $$\mathbf{h}^{(k)}(\mathbf{x}) = \mathbf{g}(\mathbf{a}^{(k)}(\mathbf{x}))$$

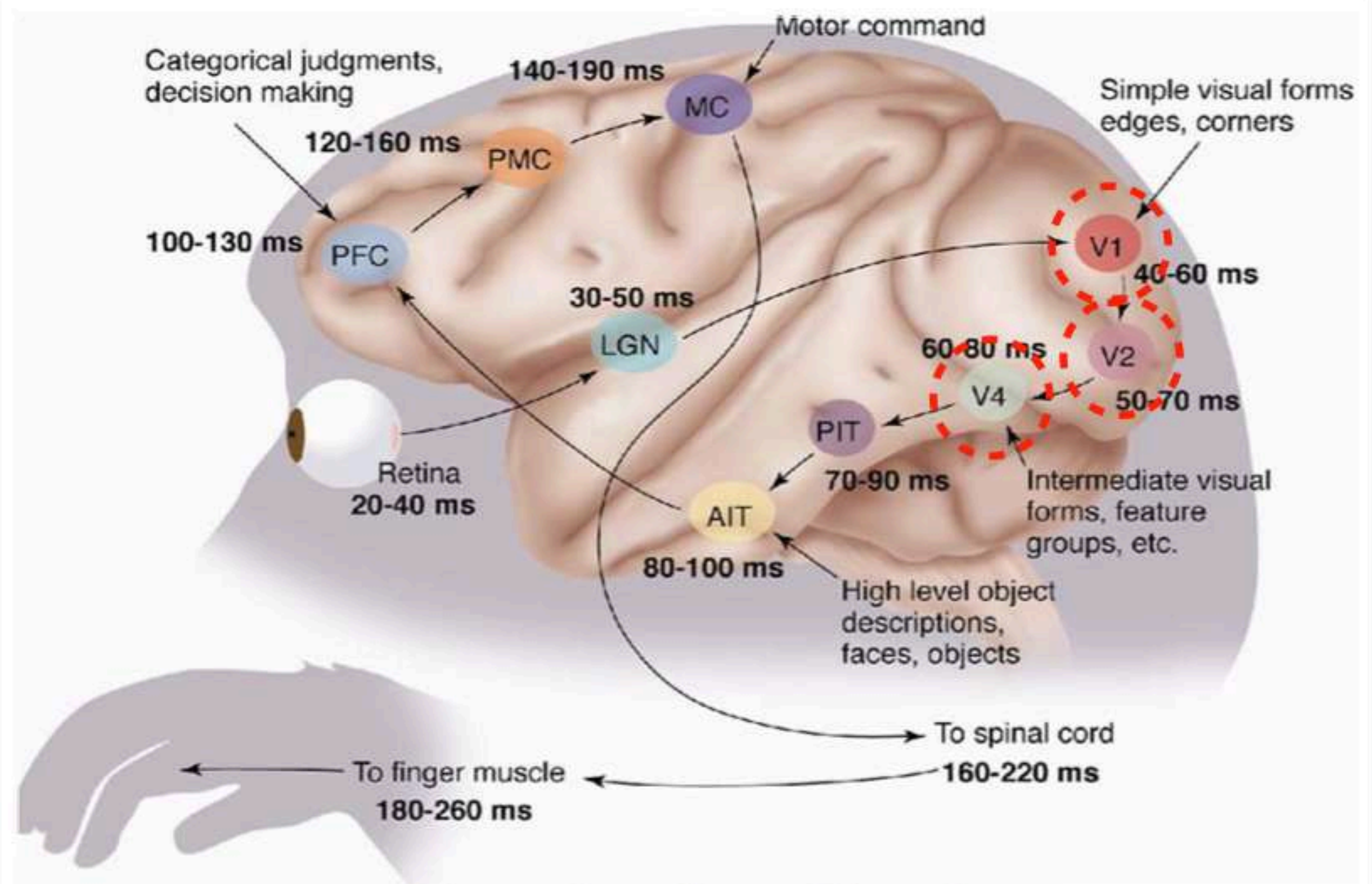
  - output layer activation ($k=L+1$):

    $$\mathbf{h}^{(L+1)}(\mathbf{x}) = \mathbf{o}(\mathbf{a}^{(L+1)}(\mathbf{x})) = \mathbf{f}(\mathbf{x})$$



NEURAL NETWORK

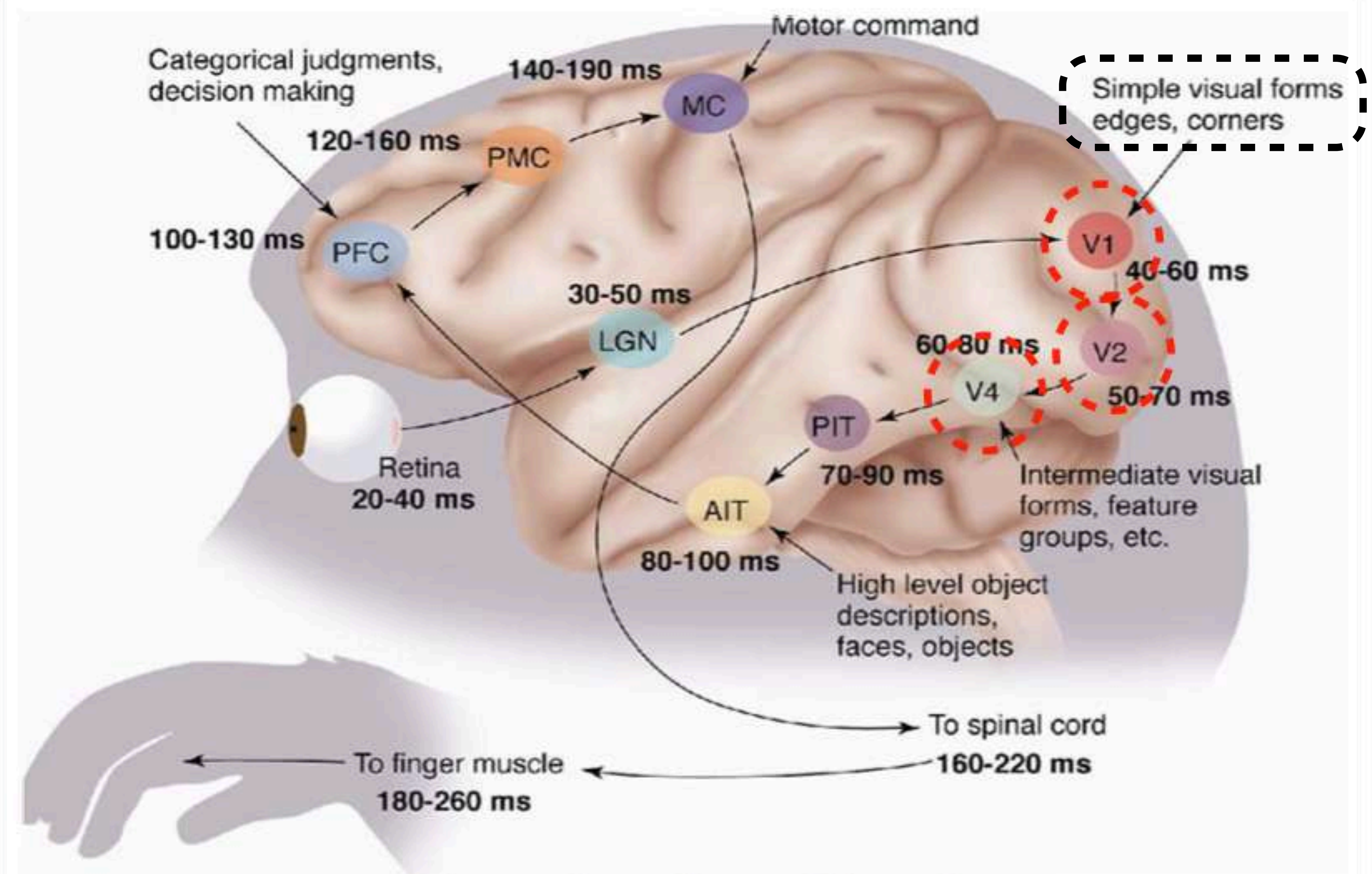
Topics: parallel with the visual cortex



[picture from Simon Thorpe]

NEURAL NETWORK

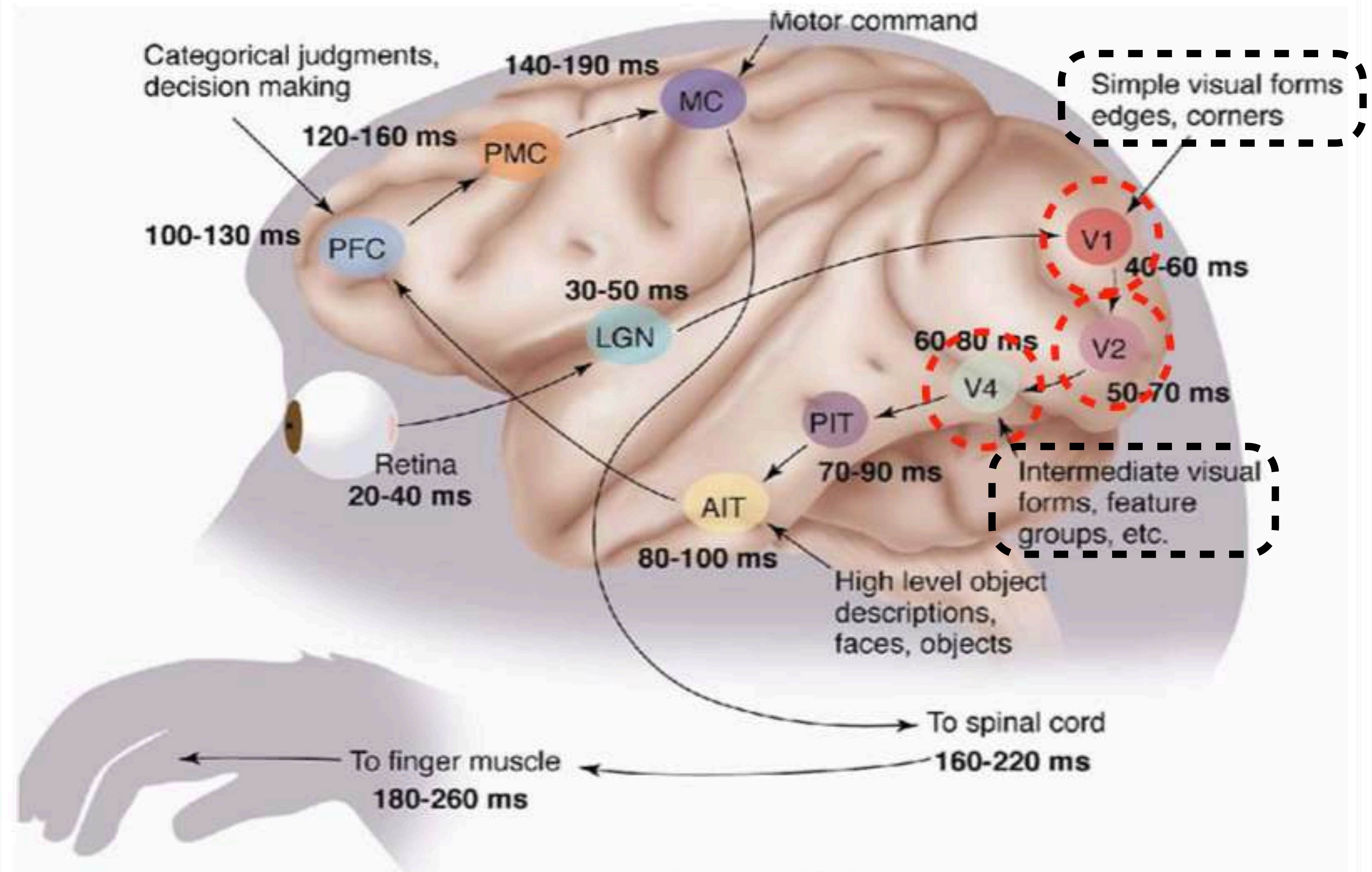
Topics: parallel with the visual cortex



[picture from Simon Thorpe]

NEURAL NETWORK

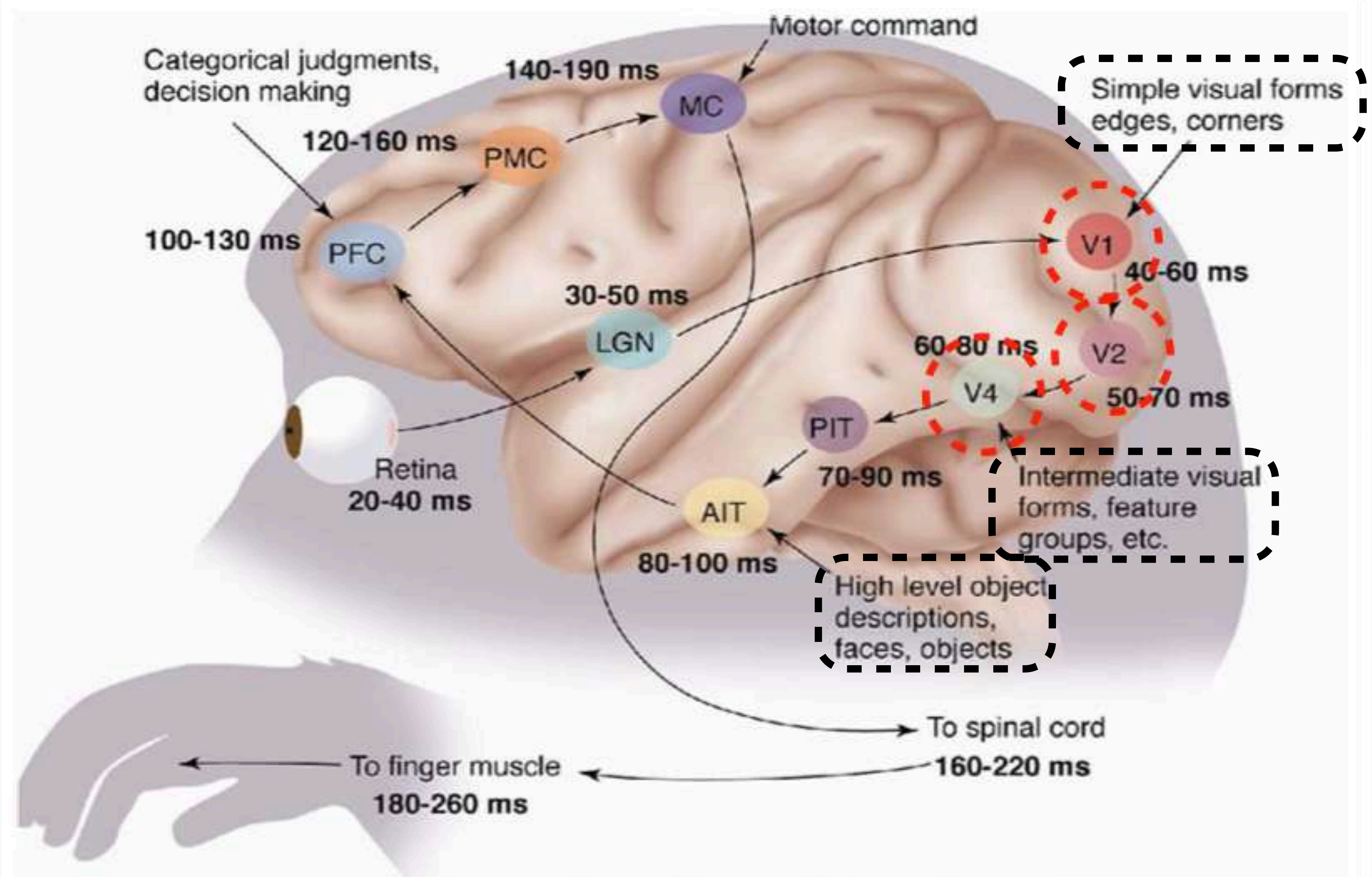
Topics: parallel with the visual cortex



[picture from Simon Thorpe]

NEURAL NETWORK

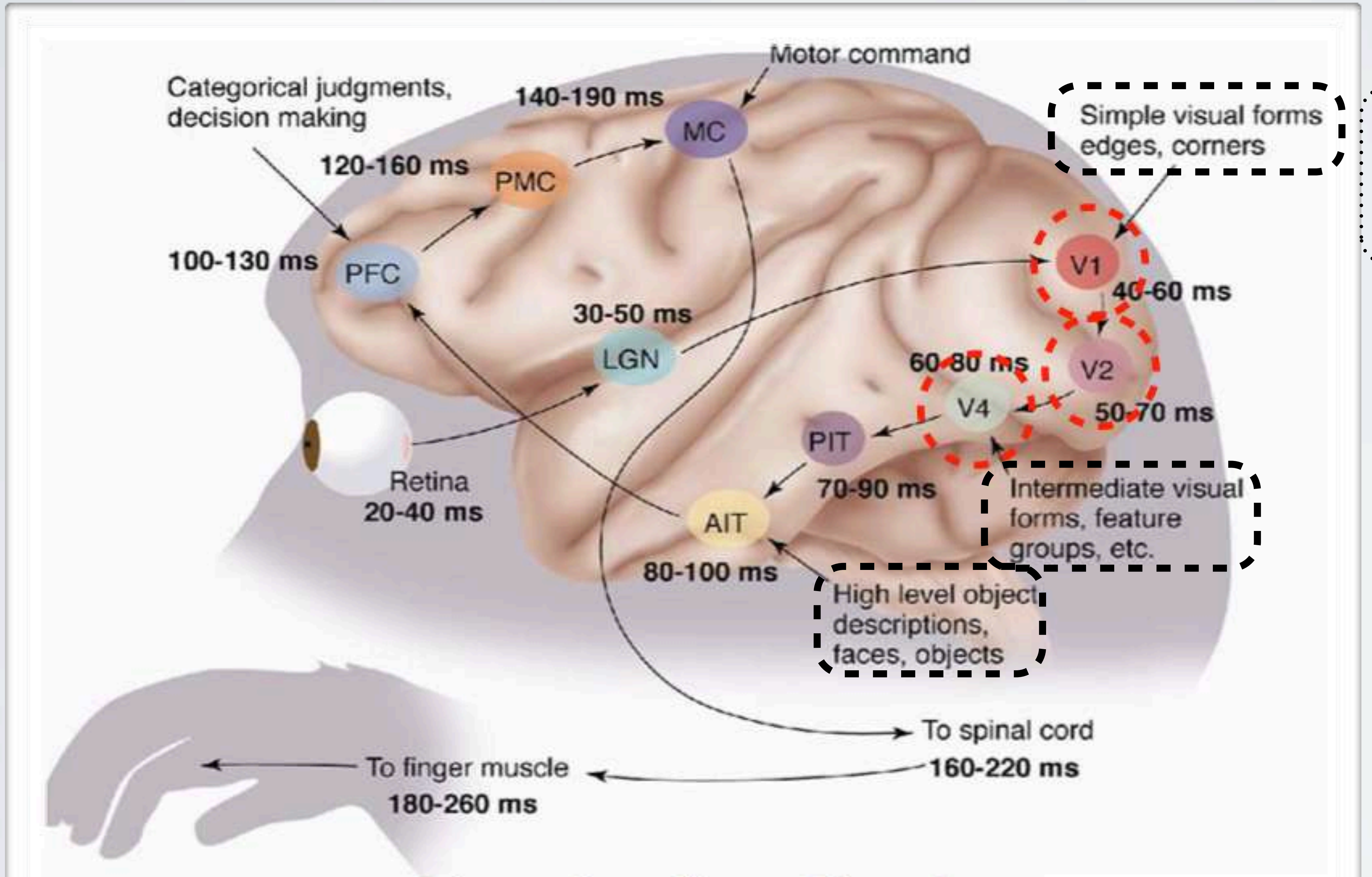
Topics: parallel with the visual cortex



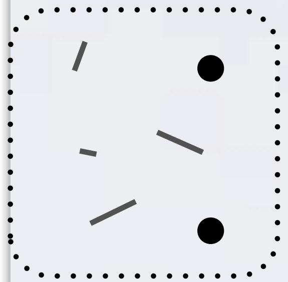
[picture from Simon Thorpe]

NEURAL NETWORK

Topics: parallel with the visual cortex



[picture from Simon Thorpe]

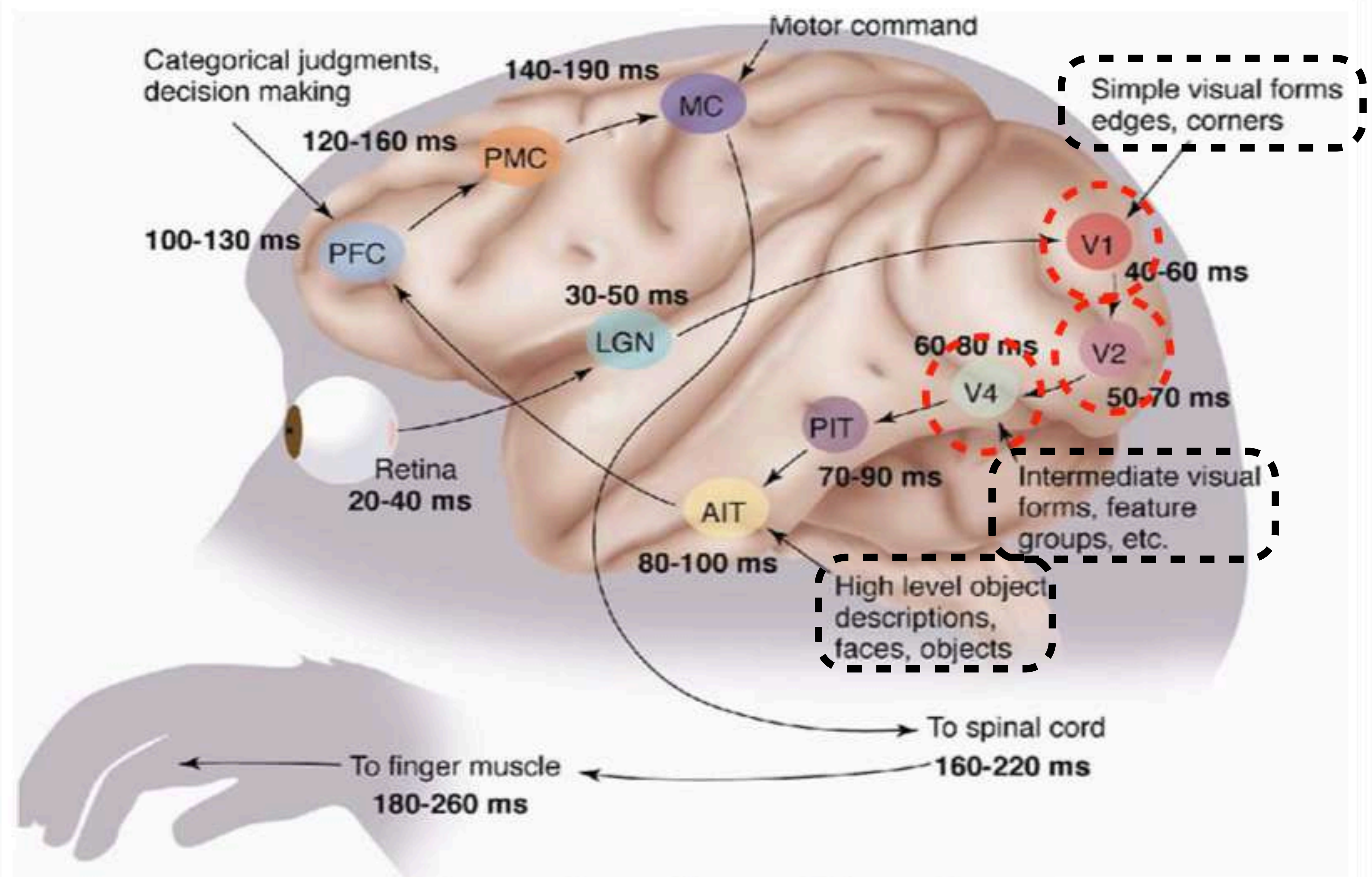


edges

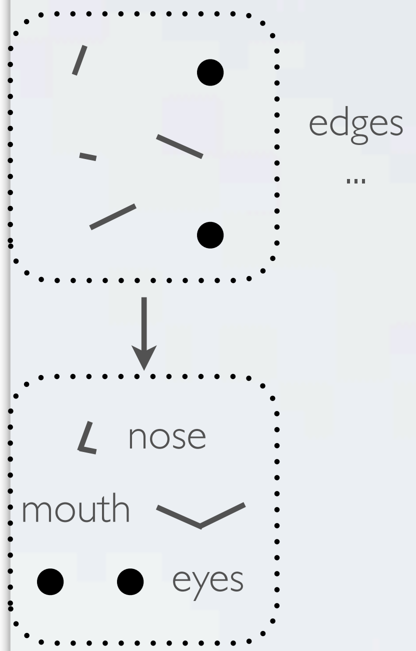
...

NEURAL NETWORK

Topics: parallel with the visual cortex

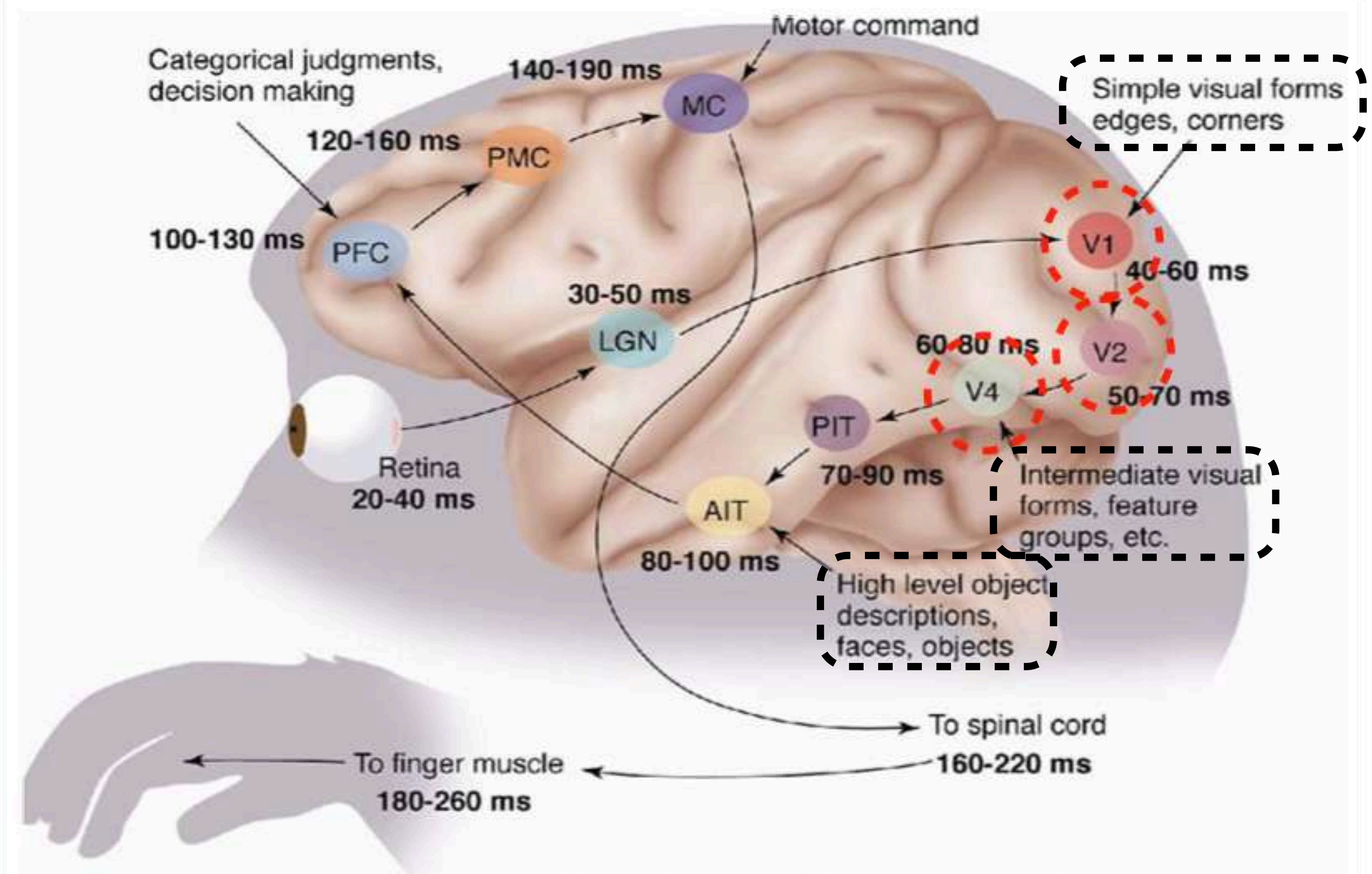


[picture from Simon Thorpe]



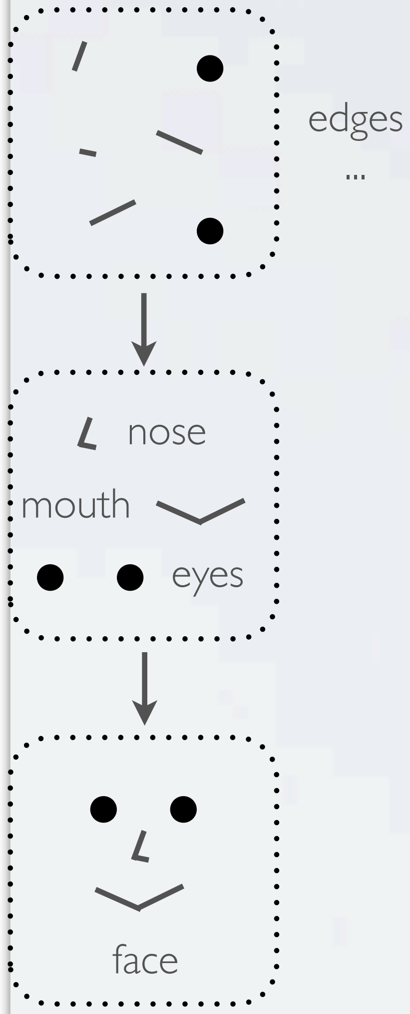
NEURAL NETWORK

Topics: parallel with the visual cortex



[picture from Simon Thorpe]

18



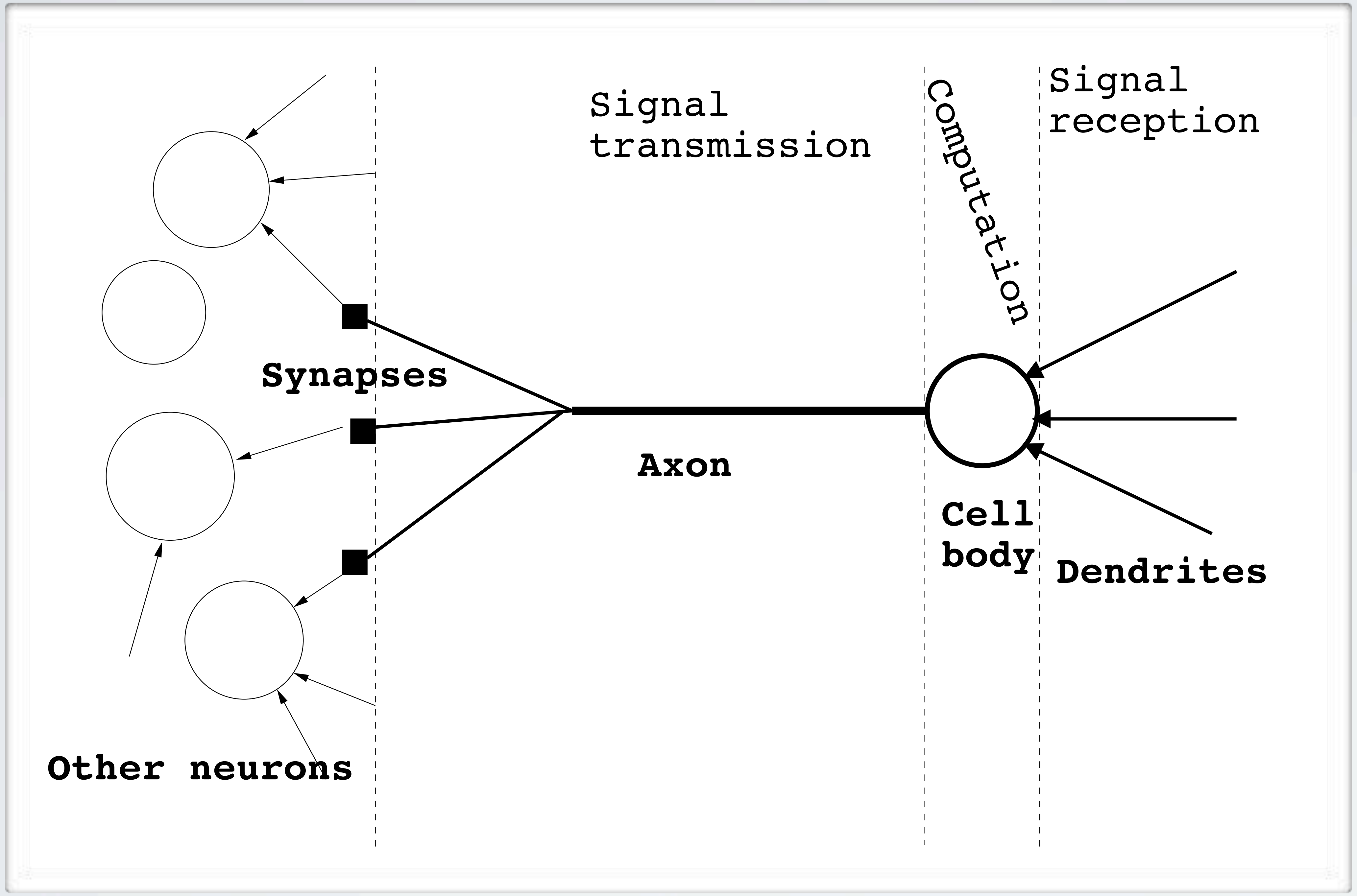
BIOLOGICAL NEURONS

Topics: synapse, axon, dendrite

- We estimate around 10^{10} and 10^{11} the number of neurons in the human brain:
  - they receive information from other neurons through their dendrites
  - the "process" the information in their cell body (soma)
  - they send information through a "cable" called an axon
  - the point of connection between the axon branches and other neurons' dendrites are called synapses

BIOLOGICAL NEURONS

Topics: synapse, axon, dendrite



(from Hyvärinen, Hurri and Hoyer's book)

BIOLOGICAL NEURONS

Topics: action potential, firing rate

- An action potential is an electrical impulse that travels through the axon:
  - this is how neurons communicate
  - it generates a "spike" in the electric potential (voltage) of the axon
  - an action potential is generated at neuron only if it receives enough (over some threshold) of the "right" pattern of spikes from other neurons

- Neurons can generate several such spikes every seconds:
  - the frequency of the spikes, called firing rate, is what characterizes the activity of a neuron
    - neurons are always firing a little bit, (spontaneous firing rate), but they will fire more, given the right stimulus

BIOLOGICAL NEURONS

Topics: action potential, firing rate

- Firing rates of different input neurons combine to influence the firing rate of other neurons:
  - depending on the dendrite and axon, a neuron can either work to increase (excite) or descrease (inhibit) the firing rate of another neuron

- This is what artificial neurons approximate:
  - the activation corresponds to a "sort of" firing rate
  - the weights between neurons model whether neurons excite or inhibit each other
  - the activation function and bias model the thresholded behavior of action potentials

BIOLOGICAL NEURONS

Hubel & Wiesel experiment

http://www.youtube.com/watch?v=8VdFf3egwfg&feature=related