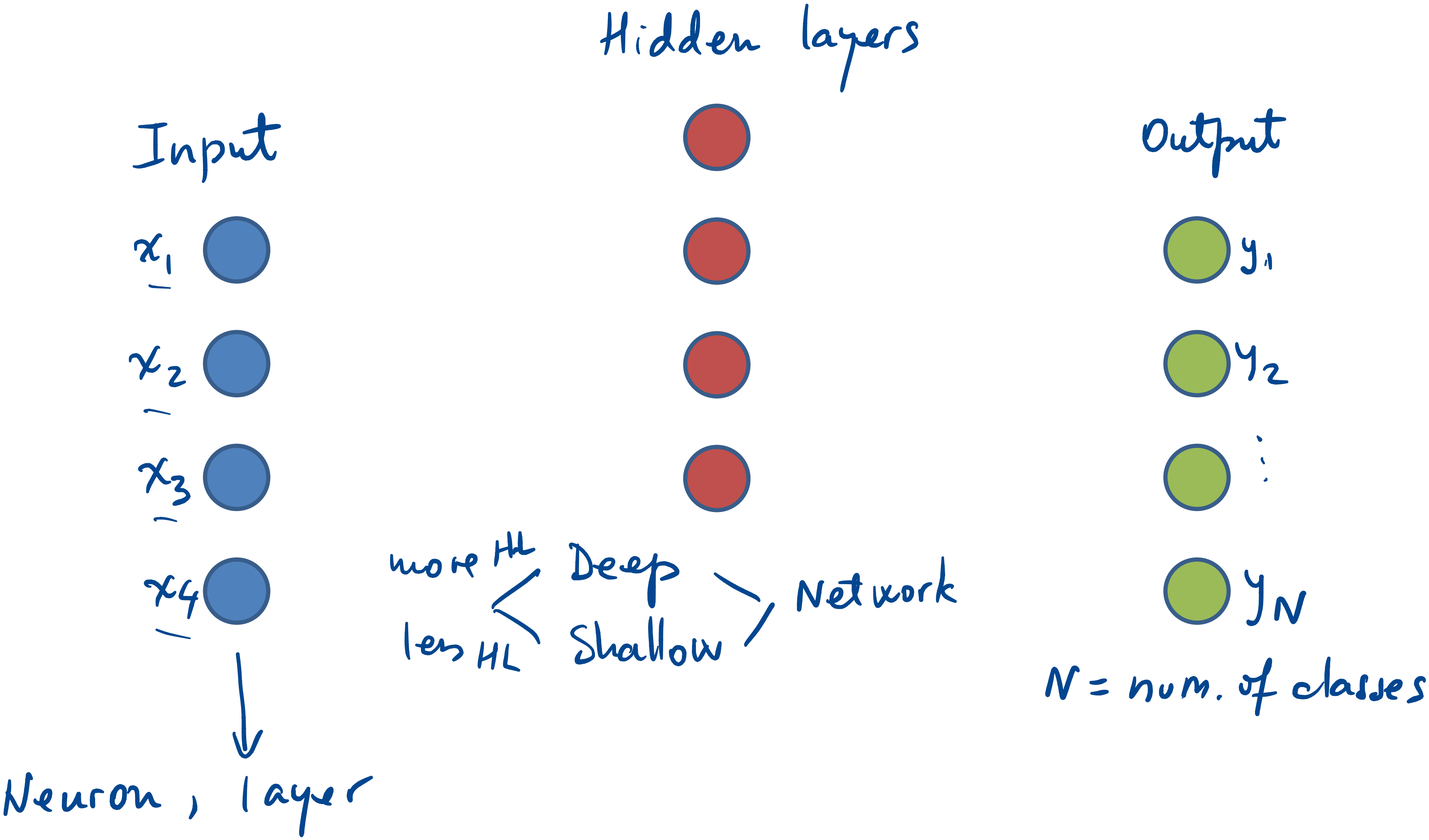
University of Colorado
Boulder

Deep Learning Applications for Computer Vision

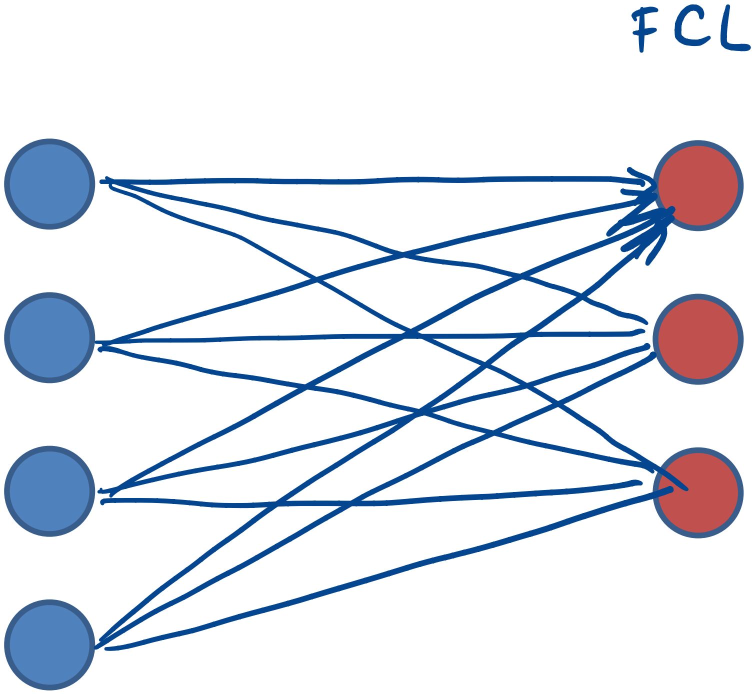
Lecture 12: Neural Networks for Image Classification

Neural Network basics



Fully connected layers

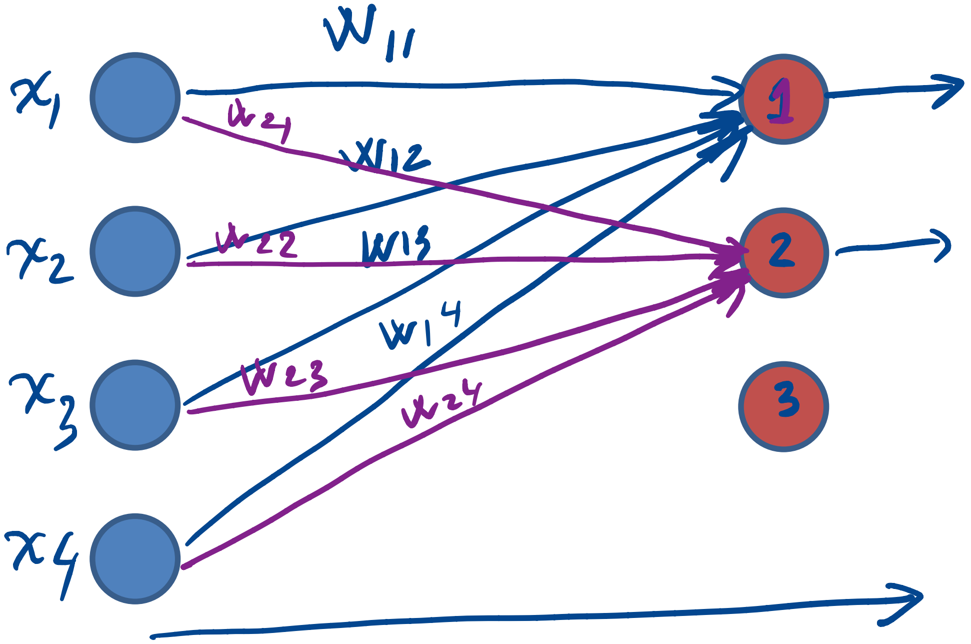
- Also known as a **dense** layers
- Def: every output from every neuron in the previous layer is an input for every neuron from the dense layer



Input, Output, Weights, Biases

- Every connection between two neurons has a **weight** associated with it
- Output = ……… (weighted sum of inputs + bias)

→ activation function

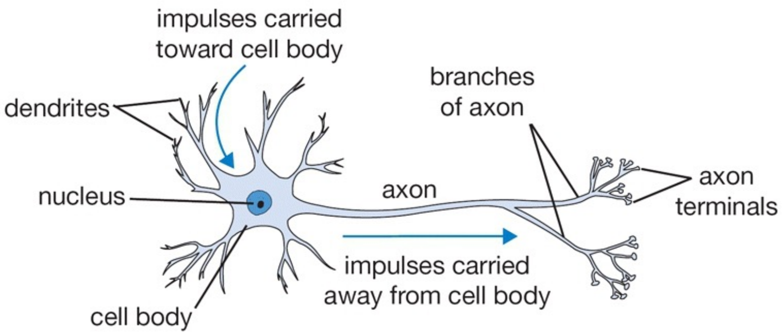


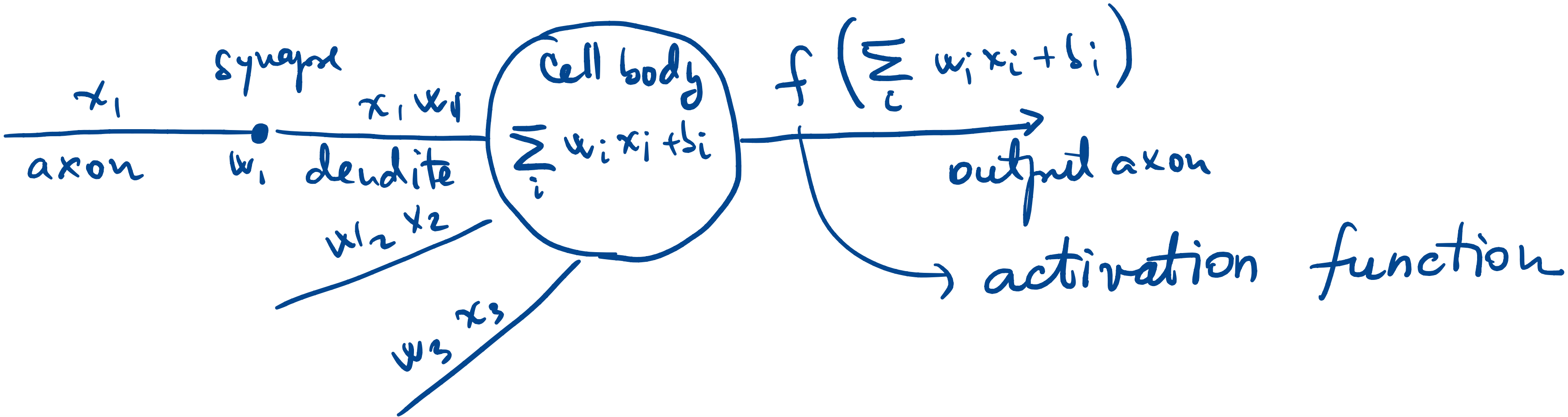
$$a_1 = \sum_i x_i w_{1i} + b_1$$

$$a_2 = \sum_i x_i w_{2i} + b_2$$

Inspiration: biological neural networks

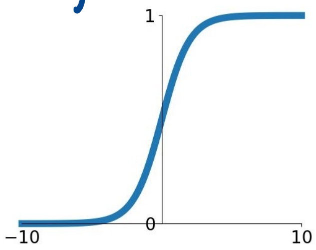
- Neuron cells
  - accept information from multiple inputs
  - transmit information to other neurons
  - connect a synapses
  - have thresholds that must be attained for the response to take place – the neuron "fires"


impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body

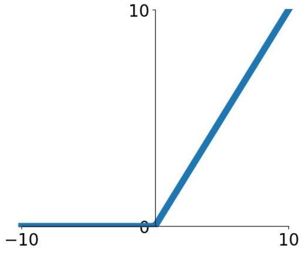


Activation functions

Sigmoid



ReLU



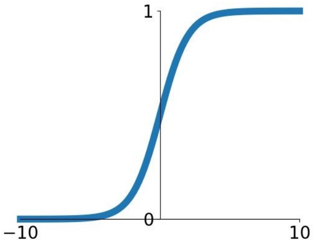
- Reasoning
  - if we only have linear mappings at every level/layer, the result will also be a linear mapping
  - image classification is more complex; linear mapping would not be a good fit
  - we need a non-linear function – done by activation functions
  - similar to the thresholding at synapses, in biological NN
- What happens?
  - sometimes output needs to be in a certain range (0-1) or
  - sometimes output needs to always be positive (>=0)

$0 \ldots \infty$

Activation functions

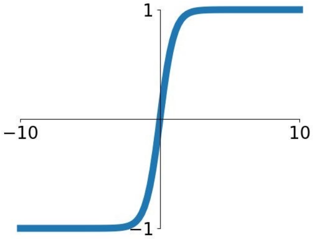
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



tanh

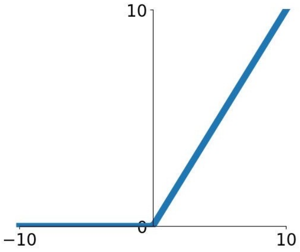
$$\tanh(x)$$



ReLU

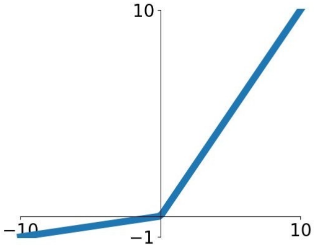
$$\max(0, x)$$

Rectified Linear Unit



Leaky ReLU

$$\max(0.1x, x)$$

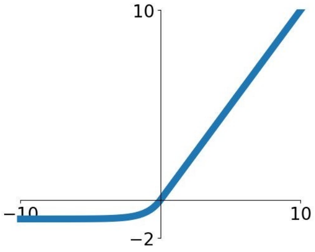


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

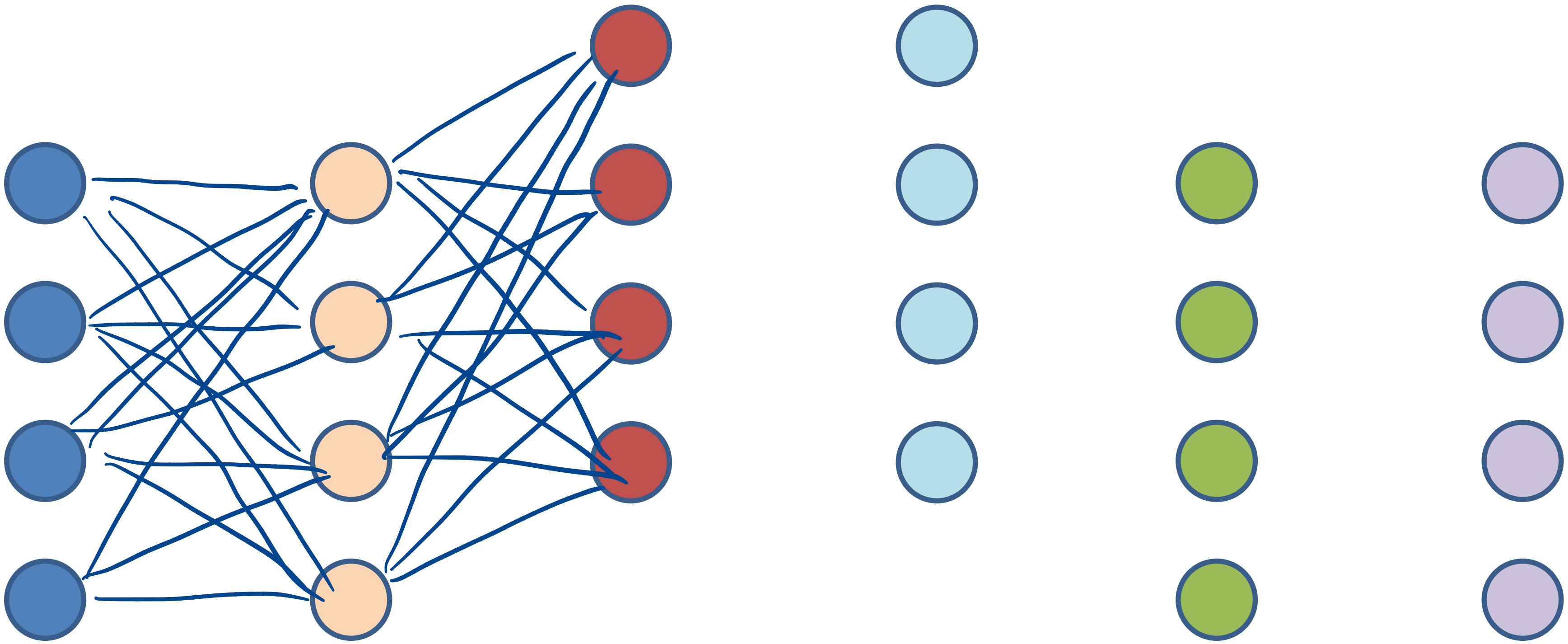
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Deep neural networks

- Many hidden layers
- Many weights to learn



How do we train the network?

- The goal is to iteratively find such a set of weights that allow the activations/outputs to match the desired output (the labels)
- We want to minimize a loss function; the loss function is a function of the weights and biases in the network

Next time: classification example when we have only a single layer of weights, and a 10-class image classification problem.