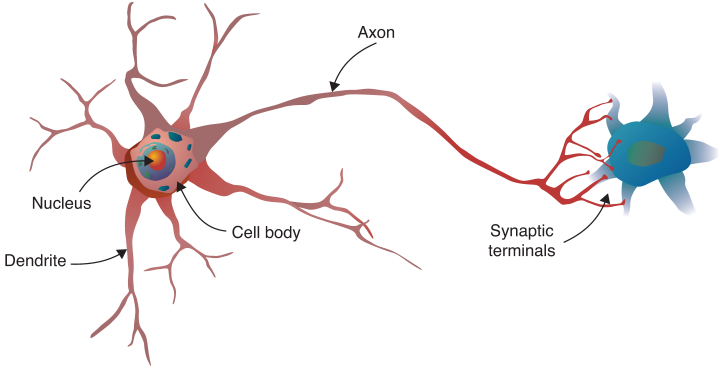
Neural Networks

CS 4277 Deep Learning

Kennesaw State University

KENNESAW STATE UNIVERSITY

Biological Neurons and Brains



[1]

- *Dendrites* receive input signals
- *Cell body* "sums" the input signals, applies some function (e.g. threshold) to sum, and places output signal on *axon*
- Axons connect to dendrites of other neurons via *synapses*
- Brain has 10^{11} neurons, each connected to 10^4 other neurons
- Much more complex than ANNs – excitatory and inhibitory signals, "backward" connections, neurotransmitters, etc.

[1] https://github.com/blueberrymusic/Deep-Learning-A-Visual-Approach

KENNESAW STATE UNIVERSITY

Linear Algebra Interlude

Vector: list/array of numbers.

$\mathbb{R}^D$ means an D-dimensional vectors of real numbers.

Column vector in $\mathbb{R}^3$:

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \mathbf{w} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Transpose of a column vector is a row vector:

$$\mathbf{x}^T = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

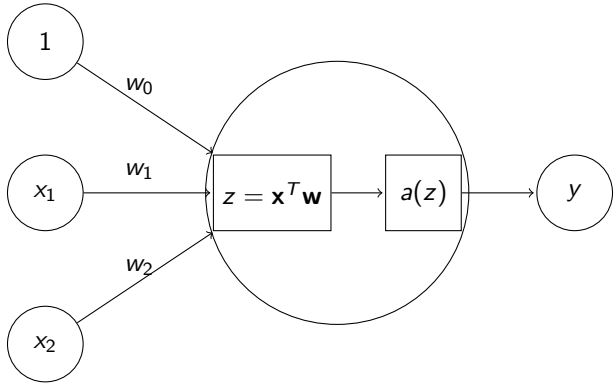
Inner product, a.k.a. "dot" product of two vectors:

$$\mathbf{x}^T\mathbf{w} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \sum_{i=1}^{D} x_i w_i = (1)(4) + (2)(5) + (3)(6) = 32$$

KENNESAW STATE UNIVERSITY

Artificial Neurons

General model of an artificial neuron:



- The summation and ReLU activation functions are implicit in the book's depictions of units (represented by the big outer circle here).
- The book uses θ instead of w for input weights and adds ϕ for output weights. We'll adopt the same naming conventions when we depict neural networks going forward.

Early History

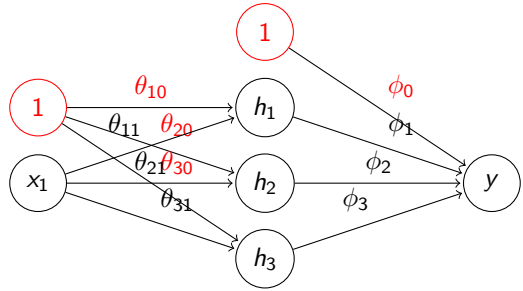
- McCulloch and Pitts (1943): first mathematical model of a biological neuron. Threshold activation function to simulate synaptic firing.
- Rosenblatt (1958, 1962) coined the term *perceptron* and came up with a perceptron learning algorithm.
- Minsky and Papert (*Perceptrons*, 1969) analyzed perceptrons in detail, focusing on their limitations and killing neural network research in CS departments until the 1990s.
- Rummelhart, Hinton and Williams (1986) published the backpropagation algorithm, the key ingredient needed to make MLPs practical.

Simple Shallow Network

A *shallow network* has an input layer, a single hidden layer, and an output layer. A neural network with one input, x, one output, y, and 10 paramters, ϕ can be written mathematically as

$$y = f(x, \phi) = \phi_0 + \phi_1 a(\theta_{10} + \theta_{11}x) + \phi_2 a(\theta_{20} + \theta_{21}x) + \phi_3 a(\theta_{30} + \theta_{31}x)$$

and depicted diagrammatically as



Here we show the bias inputs (in red) and all the weights. In future diagrams, those will be left off.

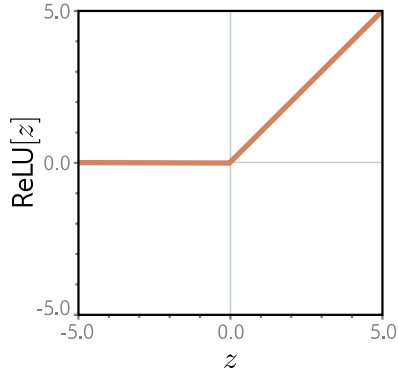
KENNESAW STATE UNIVERSITY

Rectified Linear Units

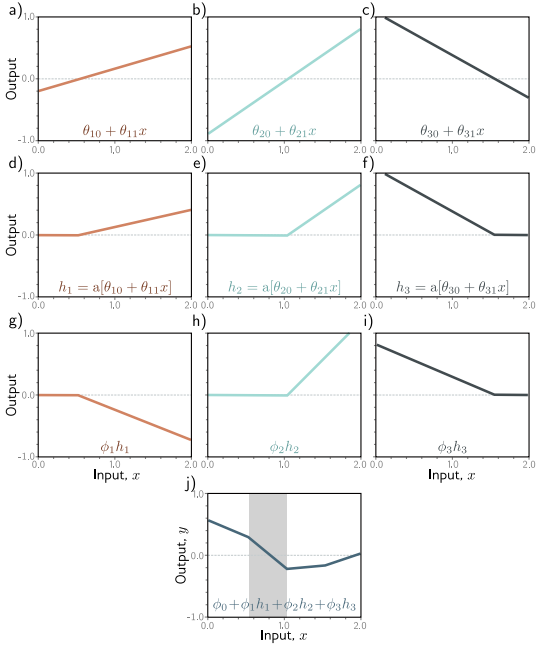
The activation function in the previous neural networkj (and in future ones) is the *rectified linear unit*, or ReLU:

$$a(z) = RelU(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases}$$

The plot of $ReLU(z)$ is:



Signal Flow in Shallow Networks



The flow of signals through network with input $x = 1$.

- Top row: inputs multiplied by weights $\theta_{h_i 1}$ and added to biases $\theta_{h_i 0}$
- Middle row: summed inputs passed through ReLU
- Third row: ReLU outputs scaled by weights ϕ
- Final row: outputs of hidden units summed and added to biases ϕ

Look at the plots and guess the parameters. Then go to https://udlbook.github.io/udlfigures/, select "3.3a - 1D shallow network (ReLU)", check your guess and play around with the parameter values.

KENNESAW STATE UNIVERSITY

General Single Input/Single Output Shallow Network Definition

Given a shallow network with one input, D hidden units, and one output, each hidden unit h_i computes:

$$h_i = a(\theta_{i0} + \theta_{i1}x)$$

- What assumption does the equation above make about the activation function(s)?

and output y is computed by:

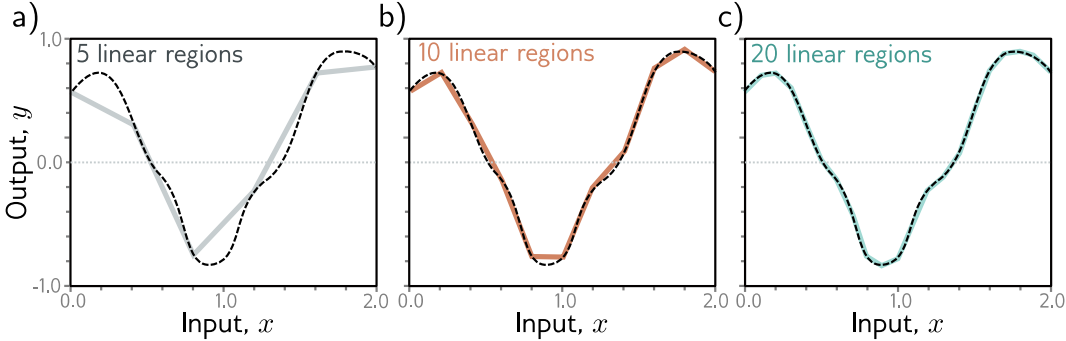
$$y = \phi_0 + \sum_{i=1}^{D} \phi_i h_i$$

- Number of hidden units determines the network's *capacity*
- With ReLU units, a network with D hidden units has D joints and thus $D + 1$ linear regions

KENNESAW STATE UNIVERSITY

Universal Approximation Theorem

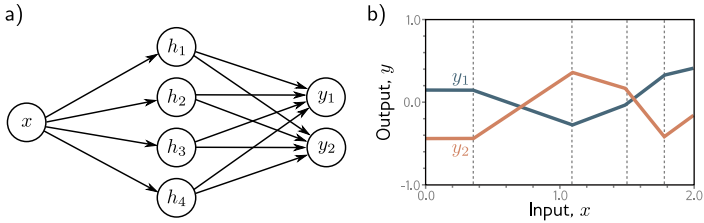
For any function, there exits a shallow network with enough hidden units to approximate the function to any precision.



For each of the function plots above, how many hidden units are in the network that produced the function?

Multivariate Outputs

Most practical neural networks have multivariate inputs and outputs. With multiple outputs:



First two layers are same as in single input/output network, but output layer computed by:

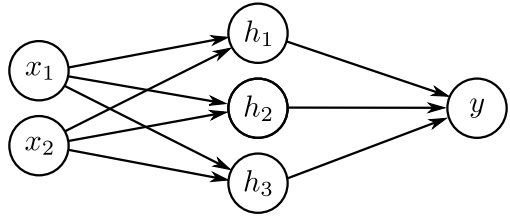
$$y_1 = \phi_{10} + \phi_{11}h_1 + \phi_{12}h_2 + \phi_{13}h_3 + \phi_{14}h_4$$
$$y_2 = \phi_{20} + \phi_{21}h_1 + \phi_{22}h_2 + \phi_{23}h_3 + \phi_{24}h_4$$

Each output unit has distinct weights on inputs from hidden units, so each can produce a different function, as shown in the plot above.

KENNESAW STATE UNIVERSITY

Multivariate Inputs

With two inputs and one output, we have:

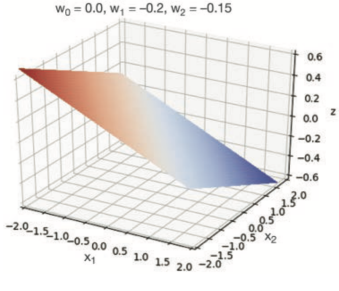


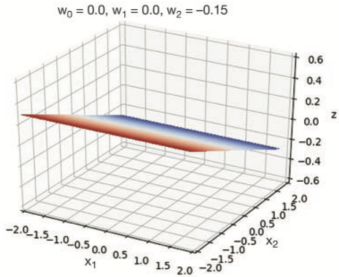
Output layer same as in single input/output network. Hidden unit outputs now computed by:

$$h_1 = a(\theta_{10} + \theta_{11}x_1 + \theta_{12}x_2)$$
$$h_2 = a(\theta_{20} + \theta_{21}x_1 + \theta_{22}x_2)$$
$$h_3 = a(\theta_{30} + \theta_{31}x_1 + \theta_{32}x_2)$$

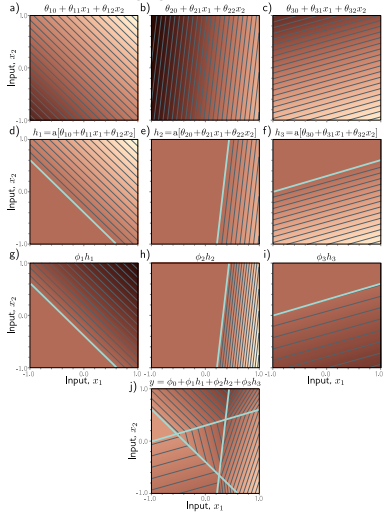
Signal Flow in Multi-Input Network

With two inputs to hidden layers we have two slopes and thus the inputs to the activation function are planes.





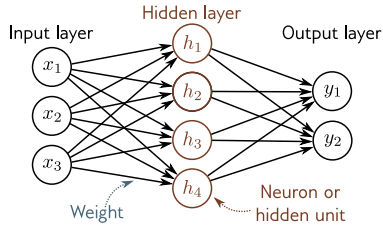
In 2nd row, the ReLUs clip planes where $z < 0$:



Output layer combines clipped planes into polygonal surface.

See https://udlbook.github.io/udlfigures/: 3.8b - 2D shallow network (ReLU)

General Shallow Networks



A shallow neural network is a funciton $\mathbf{y} = f(\mathbf{x}, \phi)$ with $\mathbf{x} \in \mathbb{R}^{D_i}$, $\mathbf{y} \in \mathbb{R}^{D_o}$ and $\mathbf{h} \in \mathbb{R}^{D}$ from D hidden units. Each $\{h_d\}_{d=1}^{D}$ computes

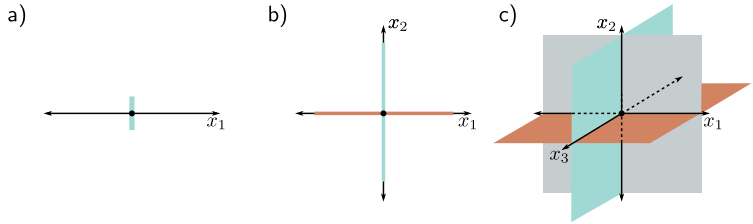
$$h_d = a(\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di} x_i)$$

and the output is computed by

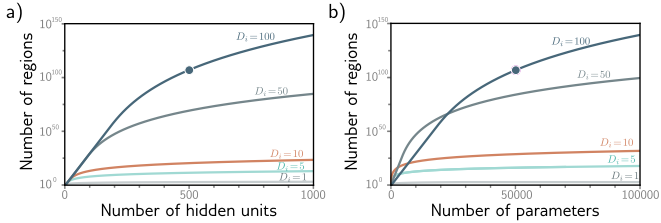
$$y_j = \phi_{j0} + \sum_{d=1}^{D} \phi_{jd} h_d$$

Capacity of Shallow Networks

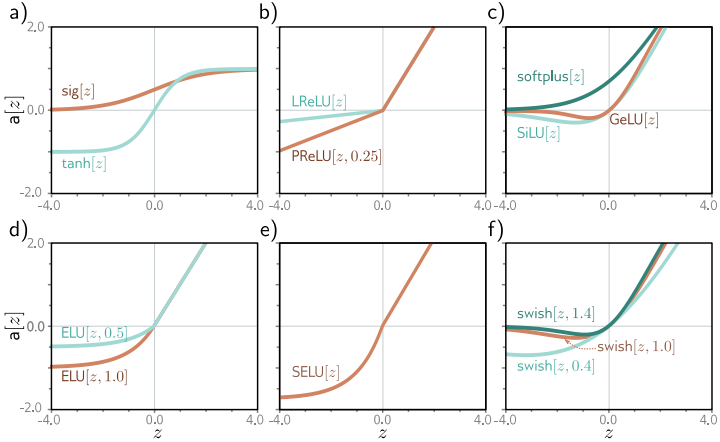
For i-dimentional input, D hidden units can divide input with D_i hyperplanes into 2_{D_i} linear regions.



More generally:



Other Kinds of Activation Functions



Play around with the interactive figures on https://udlbook.github.io/udlfigures/ to get a feel for the different funciton surfaces produced by different activation functions.

Closing Thoughts

- Shallow networks are an important technical and conceptual building block to deep networks
- So far we've only talked about *inference* – producing an output from an input
- Next we'll learn about the structure and inference in deep networks
- Then we'll spend a lot of time learning how to *train* these networks – the thing that makes them useful!