IEOR 4742 – Deep Learning for OR & FE
Introduction to Feedforward Neural Networks

Ali Hirsa
Industrial Engineering & Operations Research
Columbia University

Hype vs. Reality

- be careful about the hype
- often some most cited papers contain non-reproducible results

Basics

- basic intuition for ML and NN
- basic structure of a neuron
- how feedforward NNs work
- importance of nonlinearity in tackling complex learning problems

Deep Learning (1st definition)

- for decades, scientists have dreamed of making/building intelligent machines w/ brains like human beings (AI)

  e.g.
  - ✓ self-driving cars
  - ✓ robot housekeepers

  ⋮
- making these AI machines requires to solve some very computationally complex problems
- to tackle these problems, there is a need to develop a radically different way of programming
- this active field of `Artificial Computer Intelligence` is referred (by some practitioners) to as Deep Learning

Traditional computer programs

traditional computer programs do the following tasks (really) well[1]

(a) performing arithmetic extremely fast
(b) number crunching
(c) following list of instructions

[1] still very important in trading, market-making, etc

Innovative programs (1 of 4)

how about writing a program to automatically read yours or someone else's handwriting

Figure: MNIST handwritten digit dataset collected by LeCunn, Cortes, and Burges

Innovative programs (2 of 4)

how about writing a program to automatically recognize an image (an animal, an airplane, etc)

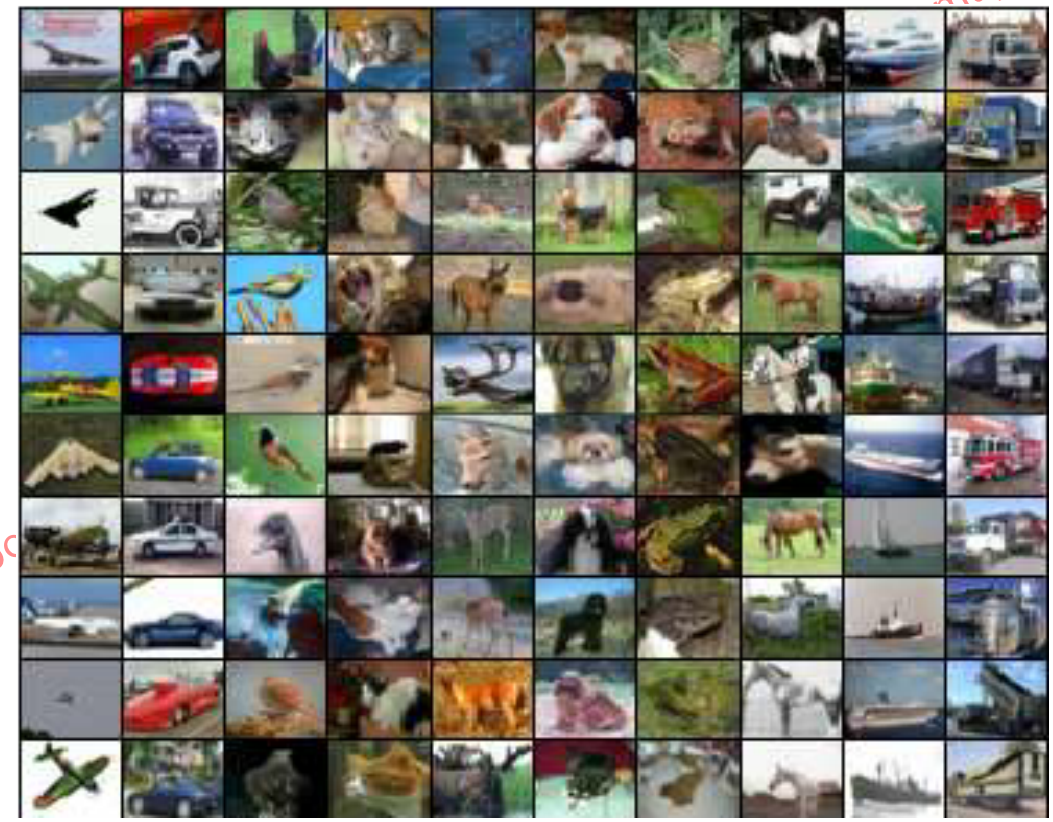


Figure: CIFAR-10 dataset collected by Krizhevsky, Nair, and Hinton

Innovative programs (3 of 4)

- humans can easily recognize every digit even though they are written in a slightly different way
- how about writing a code/module that achieves this!

Innovative programs (4 of 4)

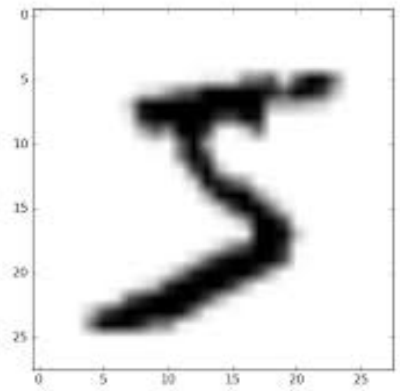
- first task is working on MNIST handwritten digit dataset (10 labels) and move to CIFAR-10 image recognition
  - the MNIST database has a training set of $60,000$ examples, and a test set of $10,000$ examples
  - it is a subset of a larger set available from NIST
  - the digits have been size-normalized and centered in a fixed-size image
  - the CIFAR-10 dataset consists of $60,000$ 32×32 color images in 10 classes, with $6,000$ images per class
  - there are $50,000$ training images and $10,000$ test images

Naive Approach (1 of 2)

- but before that, we need to set up some rules
- for zero, we might say a closed loop. Is it sufficient? NO

- ⋆ sloppy zero or messy six?
- ⋆ not a closed loop, how to distinguish?
- ⋆ some sort of cutoff? for the distance?

Naive Approach (2 of 2)



- how about 3 & 5?
- how about 4 & 9?
- we can add more & more rules or `features` but not an easy process
- could be pretty subjective
- AND beginning of our worries!

Industry-Level Tasks (1 of 2)

other classes of problems that fall into this same category:

- object recognition
- speech comprehension
- automated translation/NLP
- pattern recognition

Industry-Level Tasks (2 of 2)

signal-to-noise (TR vs. MR)

Intentionally Left Blank

uDu vs. UdU

Mechanics of Learning

- we do not exactly know how it is done by our brains
- even if we know, the program might be super complicated
- learning by example not by formula:
  - think of a 2-year-old kid who would recognize a dog or a cat by being shown multiple examples of them
  - ⋆ being corrected if she/he made a wrong guess (if that guess was confirmed by her parents, the learning would be reinforced).
  - think of traditional traders, learned simply by simply watching the markets and using charts
  - ⋆ if markets proved them wrong, modify the thinking to incorporate this new information and improve performance

Machine Learning vs. Brute-Force Learning

- in ML, instead of teaching a computer a huge list of features/rules to solve the problem, we give it a model with which it can evaluate examples, and a small(er) set of instructions to modify the model when it makes a mistake.

Short list of Models

Supervised learning (`data` & `labels`):

- Decision Trees (DTs)
- Classification (Bayes)
- Regression (OLS)
- Logistic Regression
- Support Vector Machine (SVM)
- Ensemble Methods (BAGGing, Random Forest)

Unsupervised Learning (`data`):

- Clustering
- PCA/SVD
- ICA

Supervised vs. Unsupervised Learning

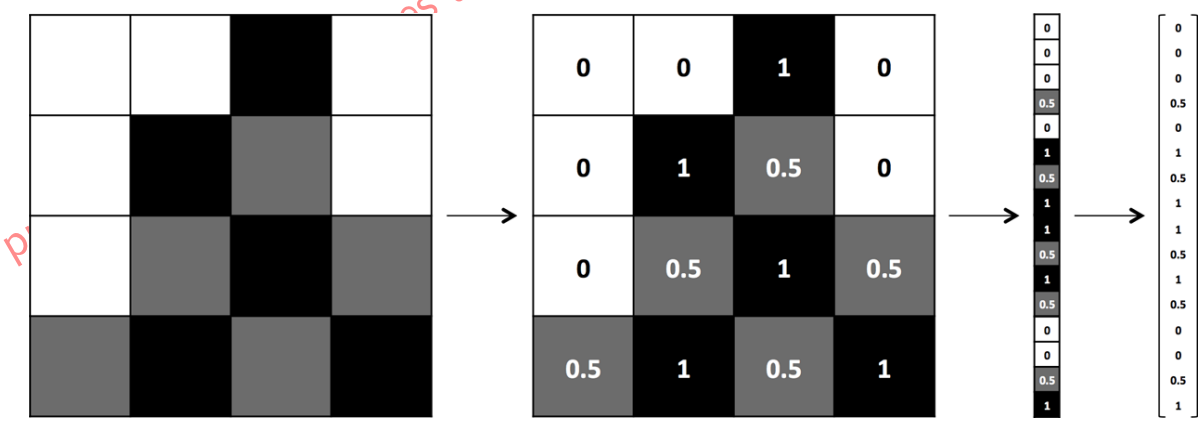
- supervised learning: given both inputs and labels (outputs)
- typically done in the context of classification, when we want to map input to output labels
- unsupervised learning: learn the inherent structure of our data without using explicitly-provided labels

Model as a Function

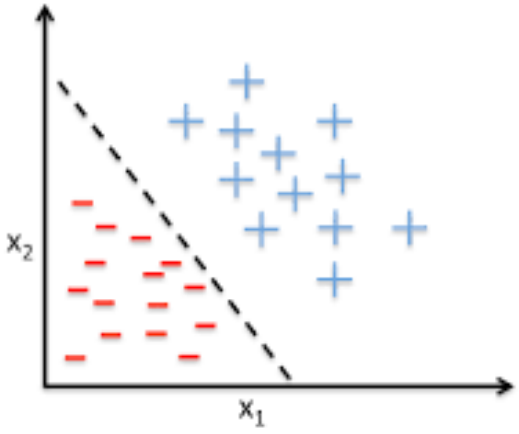
mathematically, a model could be a function:

$$h(x, \Theta)$$

In case of x being an image should be converted into a scalar vector



Linear decision boundary for classification



Example of a linear decision boundary for binary classification.

$$h(x, \Theta) = \begin{cases} -1 & \text{if } \theta_0 + \theta_1 x_1 + \theta_2 x_2 < 0 \\ +1 & \text{if } \theta_0 + \theta_1 x_1 + \theta_2 x_2 > 0 \end{cases}$$

Linear Perceptron (1 of 2)

- a model like that is called `Linear Perceptron`

Definition

Perceptron: a computer model or computerized machine devised to represent or simulate the ability of the brain to recognize and discriminate.

- first model for learning with a teacher (i.e., supervised learning) proposed by Frank Rosenblatt 1958

Linear Perceptron (2 of 2)

Q: how do we come up w/ optimal value for parameters

Linear Perceptron (2 of 2)

Q: how do we come up w/ optimal value for parameters

A: by optimizing a loss function

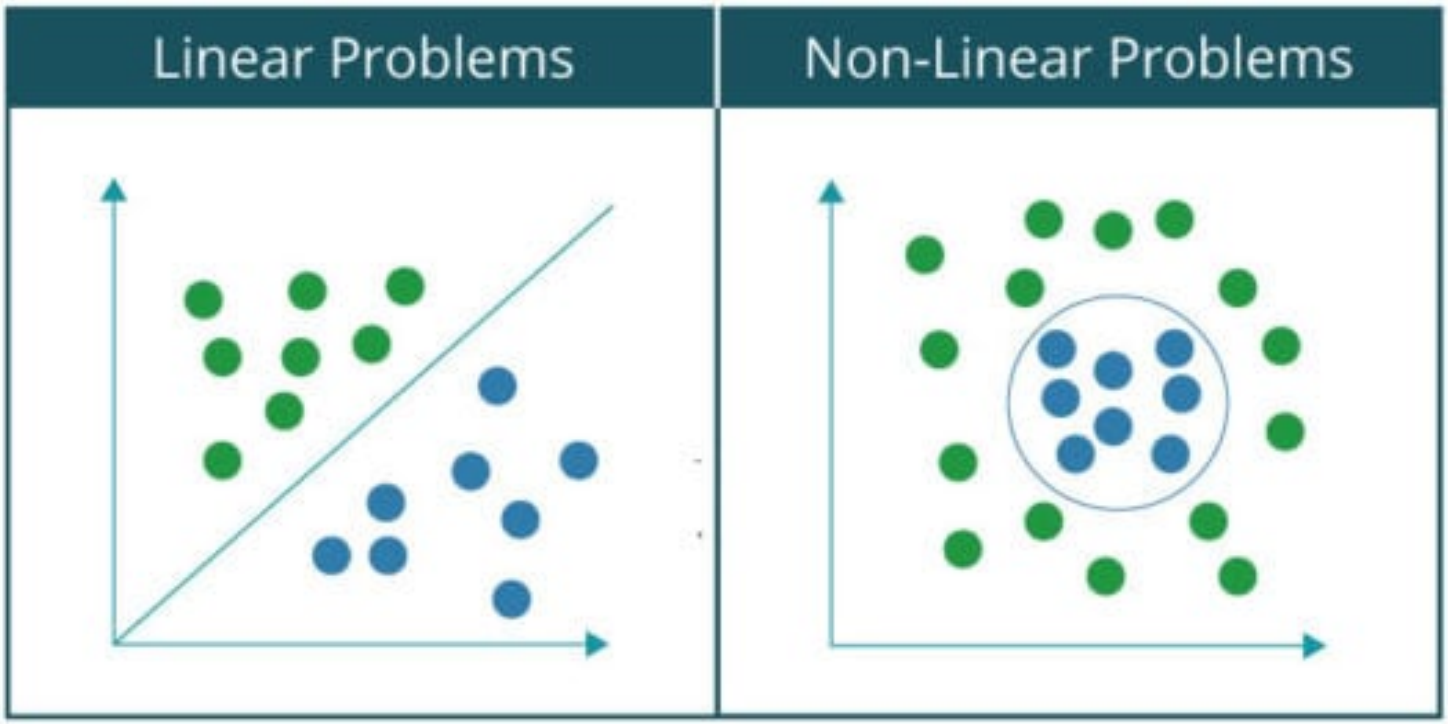
Linear Perceptron (2 of 2)

Q: how do we come up w/ optimal value for parameters

A: by optimizing a loss function

Goal: to maximize the performance of a machine learning model by iteratively tweaking its parameters until the error is minimized

Limitation of Linear Perception



- Linear perception model is limited
- more complex form is needed

Deep Learning (2nd definition)

to accommodate this complexity:

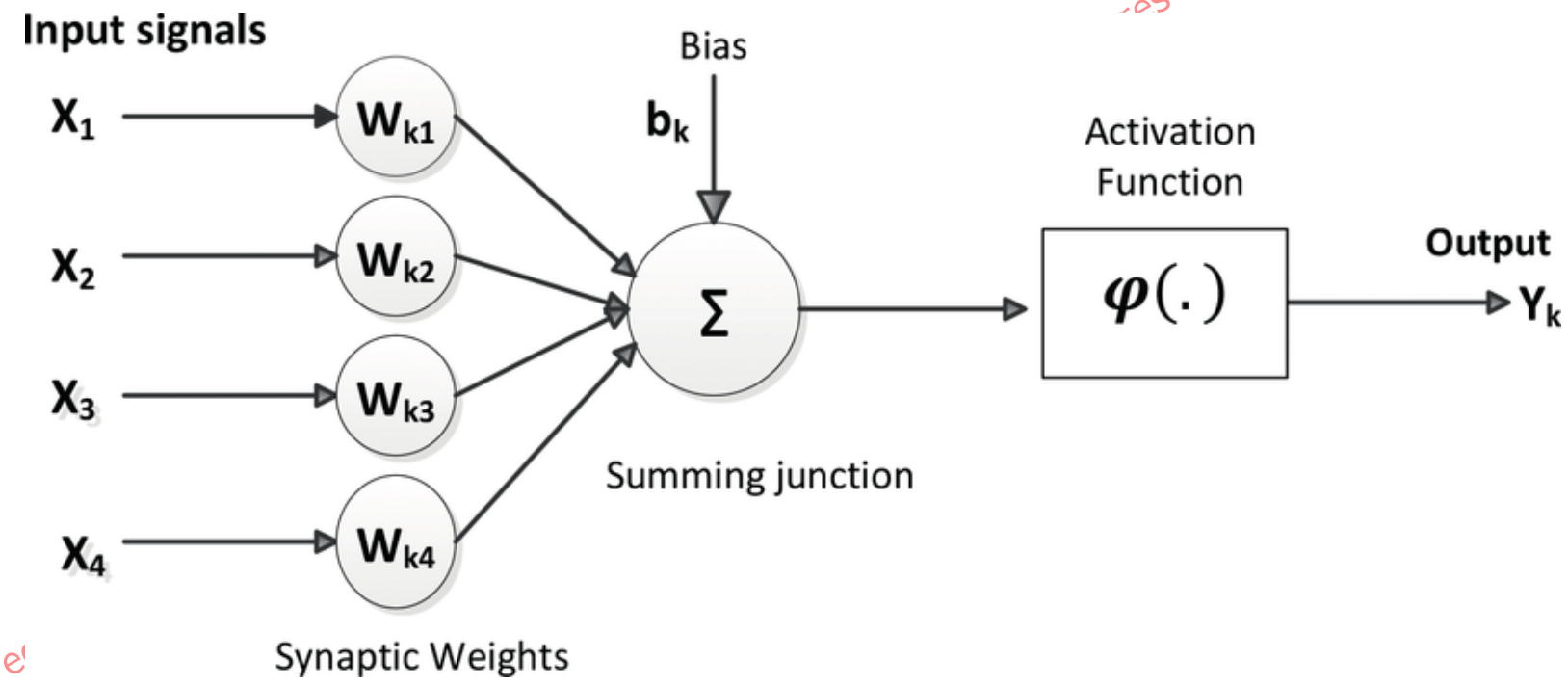
- researchers have tried to build models that resemble the structures utilized by our brains
- it is essentially this body of research commonly referred to as **Deep Learning**
- claim: these algorithms not only far surpass other kinds of ML algos, but also rival the accuracies achieved by humans

The Neuron

- foundation unit of the human brain is the *neuron*
- Intel claims by 2026 processors will have as many transistors as there are neurons in a brain (100,000,000,000).
- ⋆ a grain of rice contains over 10,000 neurons, each of which forms an average of 6000 connections w/ other neurons
- this *Massive Biological Network* enables us to experience the world around us.

Translation (1 of 2)

McCulloch & Pitts 1943



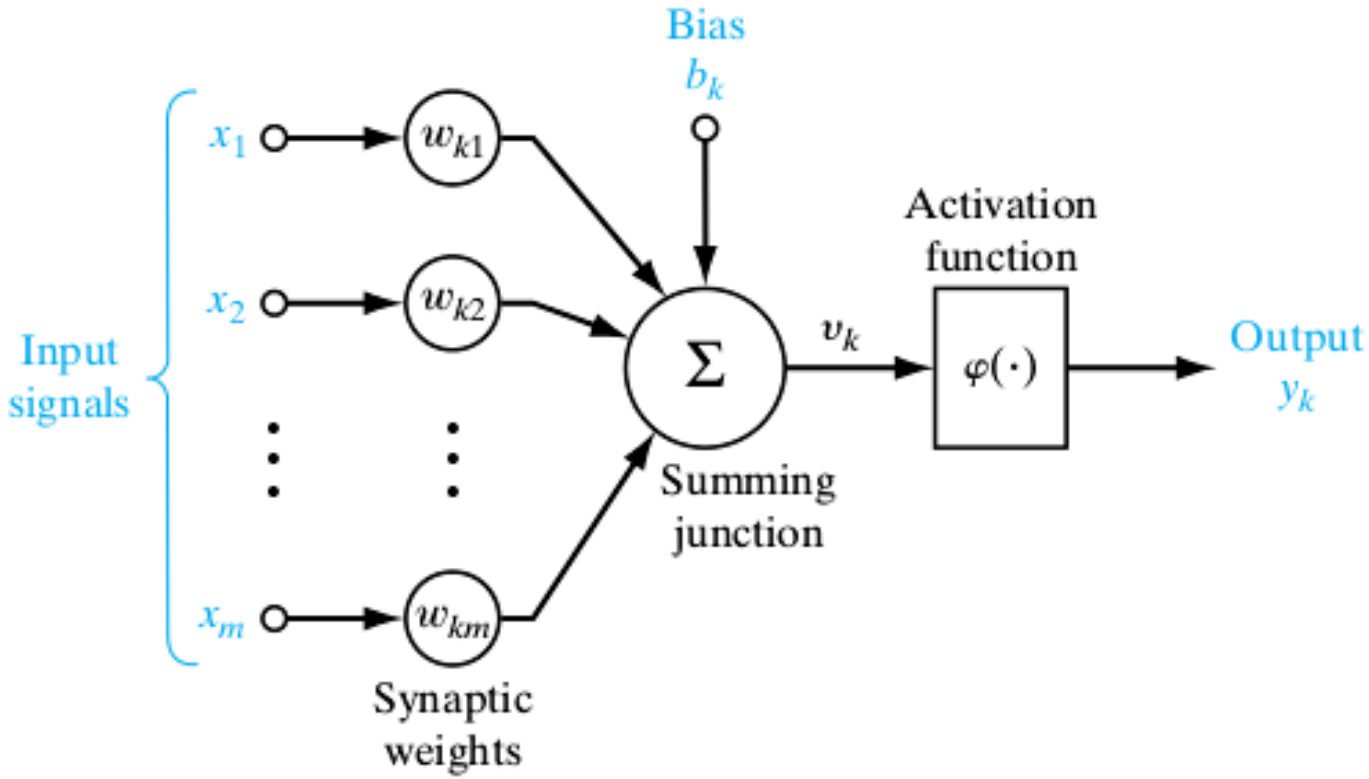
Translation (2 of 2)

translation of this functional understanding of the neurons in our brain into an artificial model that we can represent on our computer is as follows:

- input $x_1, \ldots, x_n$ any non-normalized probability distribution
- weight of the neuron $w_1, \ldots, w_n$
- $y = \varphi(x^\top w + b)$ where b is the biased term

Linear Perceptrons as Neurons (1 of 2)

$$h(x, \Theta) = \begin{cases} -1 & : \quad w_0 + w_1 x_1 + w_2 x_2 < 0 \\ 1 & : \quad w_0 + w_1 x_1 + w_2 x_2 \geq 0 \end{cases}$$



Linear Perceptrons as Neurons (2 of 2)

- brain is made of more than one neuron

Linear Perceptrons as Neurons (2 of 2)

- brain is made of more than one neuron
- it is impossible for a single neuron to differentiate handwritten digits

Linear Perceptrons as Neurons (2 of 2)

- brain is made of more than one neuron
- it is impossible for a single neuron to differentiate handwritten digits
- neurons in our brain are organized in layers

Linear Perceptrons as Neurons (2 of 2)

- brain is made of more than one neuron
- it is impossible for a single neuron to differentiate handwritten digits
- neurons in our brain are organized in layers

→ Feedforward Neural Networks

Layers

according to V.B. Mountcastle (1918 - 2015) Neuroscientist at Johns Hopkins (1957):

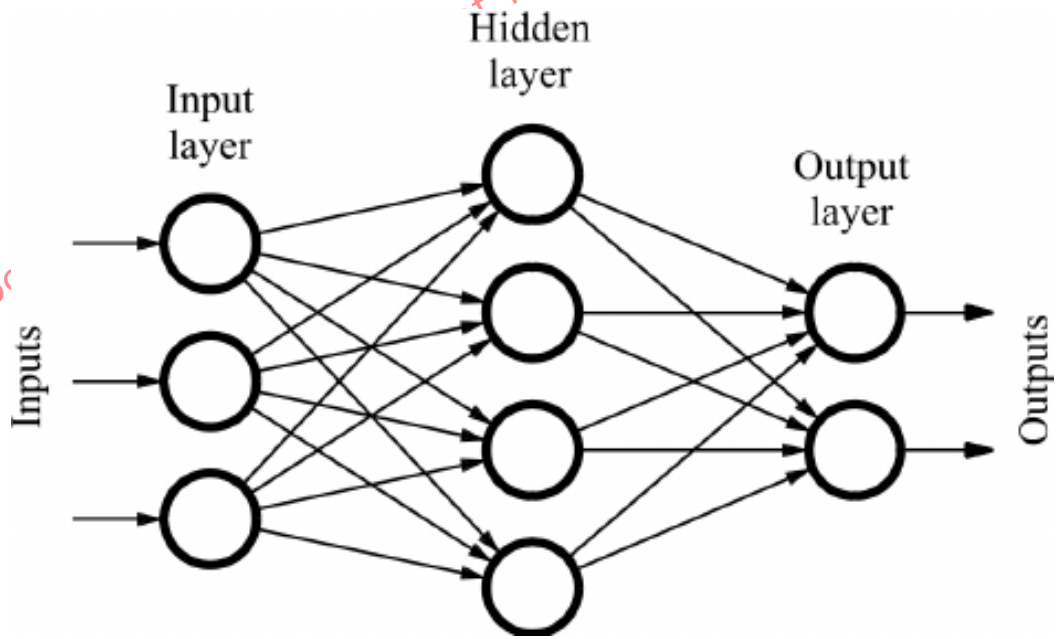
- Human Cerebral Cortex is made up of six layers
- information flows from one layer to another until sensory input is converted into conceptual understanding
- it is processed by each layer and passed on to the next layer until in the six layer, we conclude whether we are looking at a cat or a dog

Feedforward Networks (1 of 3)

- a feedforward neural network is an artificial neural network wherein connections between the nodes do not form a cycle
- the feedforward neural network was the first and simplest type of artificial neural network devised
- in this network, the information moves in only one direction, `forward`, from the input nodes, through the hidden nodes (if any) and to the output nodes
- there are no cycles or loops in the network

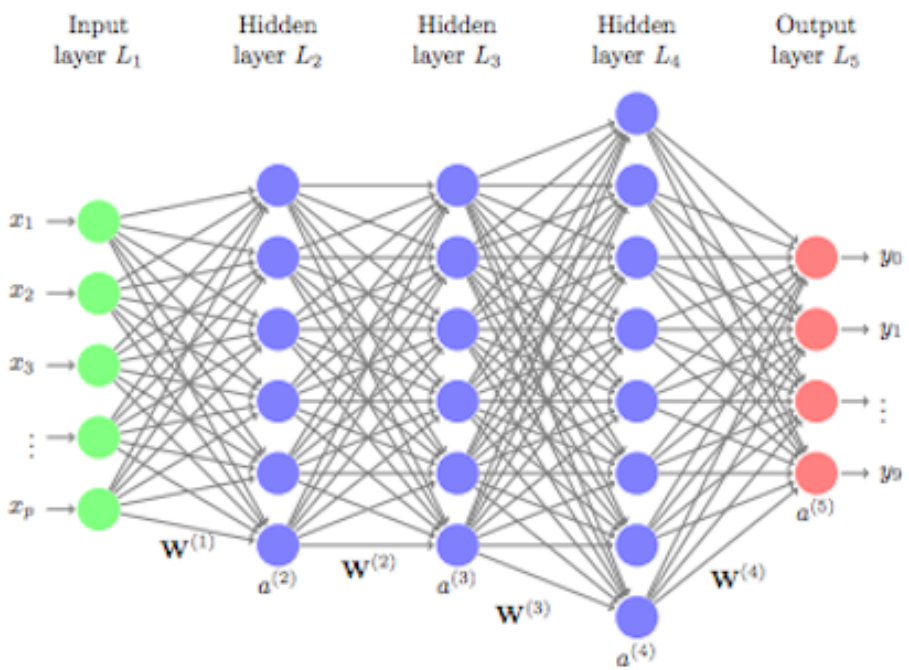
Feedforward Networks (2 of 3)

- inputs & outputs are represented as vectors
- connections only travel from lower to higher layer
- no connections in the same layer (for simplicity)
- make it simple to analyze



Feedforward Networks (3 of 3)

- weight connecting i^{th} neuron in the k^{th} layer with j^{th} neuron in the $(k+1)^{th}$ layer
- these weights constitute the parameter set Θ (vector)
- our ability to solve problems with neural networks depends on finding the optimal values



Hidden Layers

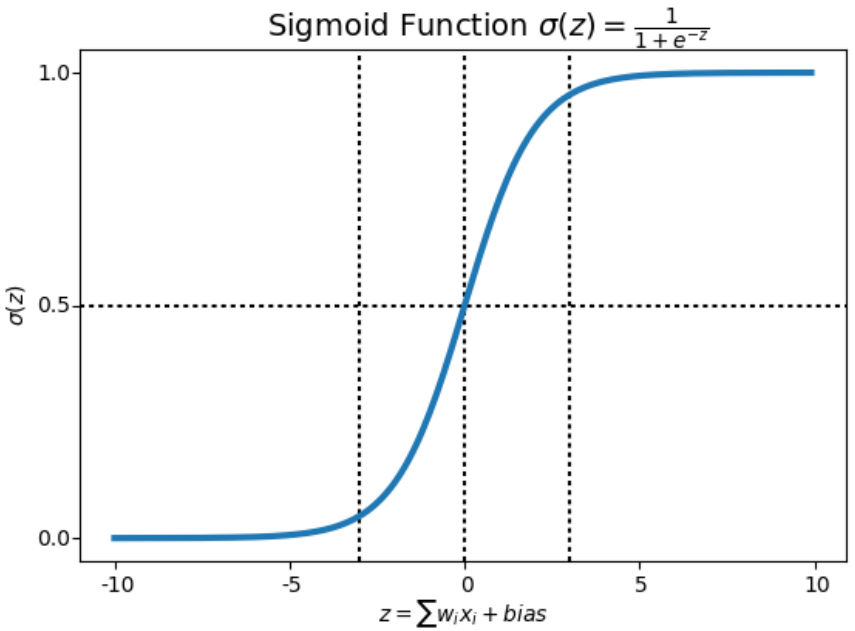
- layers of neurons that are between the input (first layer) and the output (last layer) are called hidden layers
- each hidden layer could have different numbers of neurons
- not required that every neuron has its output connected to the inputs of all neurons in the next layer
- how to do that, it is "an art"
- most action is happening when the neural net tries to solve problems
- taking a look at the activities of hidden layers can say a lot about features
- the network has automatically learned to extract from data
- choose between fewer layers with more neurons per layer vs. deep layers with fewer numbers of neurons per layer
- usually no more than 5 to 7 layers, overfitting starts to occur

Activation Functions

- linear neurons are easy to compute with, but have serious limitations
- can be shown that any feedforward neural network consisting of only linear neurons can be expressed as a network with no hidden layers (exercise)

The logistic sigmoid function

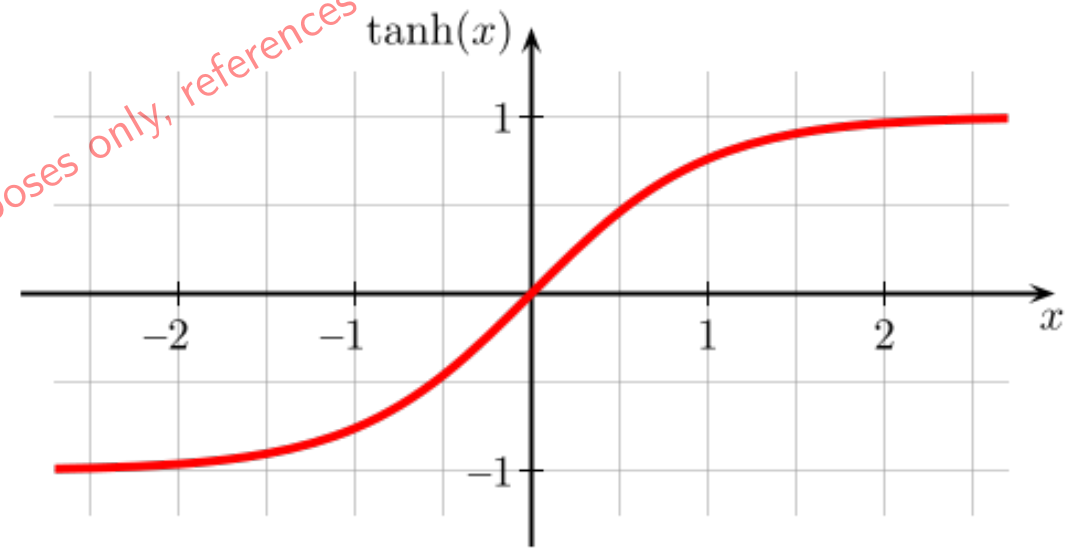
$$f(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$



The tanh function

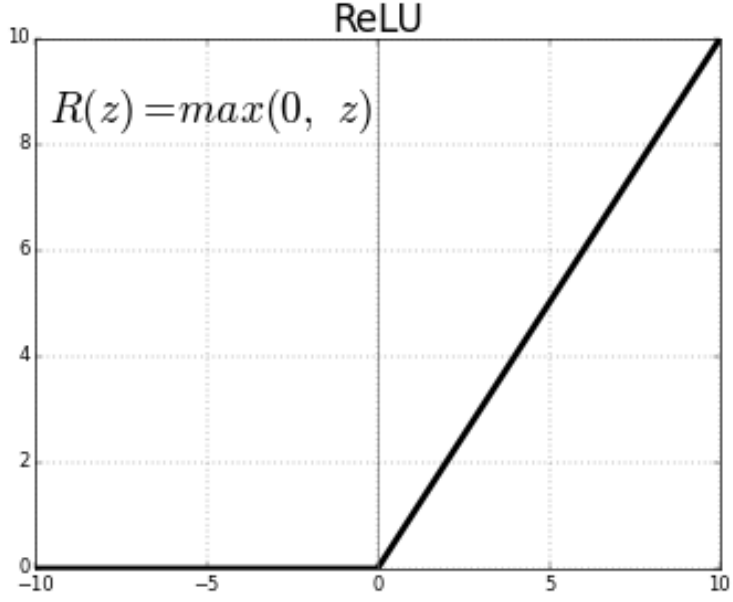
$$\begin{aligned} f(z) &= \tanh(z) \\ &= \frac{e^z - e^{-z}}{e^z + e^{-z}} \\ &= 2\sigma(2z) - 1 \end{aligned}$$

tanh is a re-scaled logistic sigmoid function



The restricted linear unit (ReLU) function

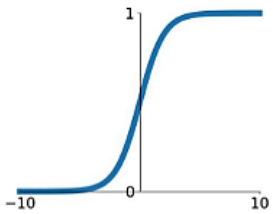
$$f(z) = \max(0, z)$$



Collection of deep learning activation functions (1 of 2)

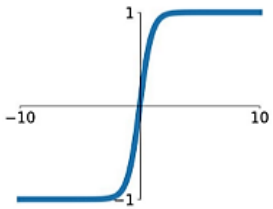
Sigmoid

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



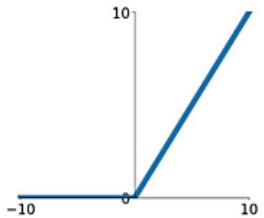
tanh

$$\tanh(x)$$



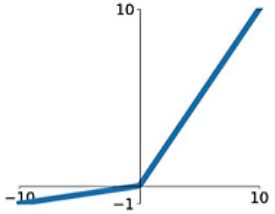
ReLU

$$\max(0, x)$$



Leaky ReLU

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

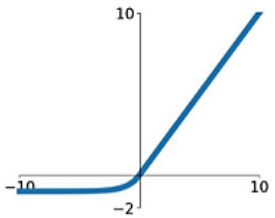


Maxout

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

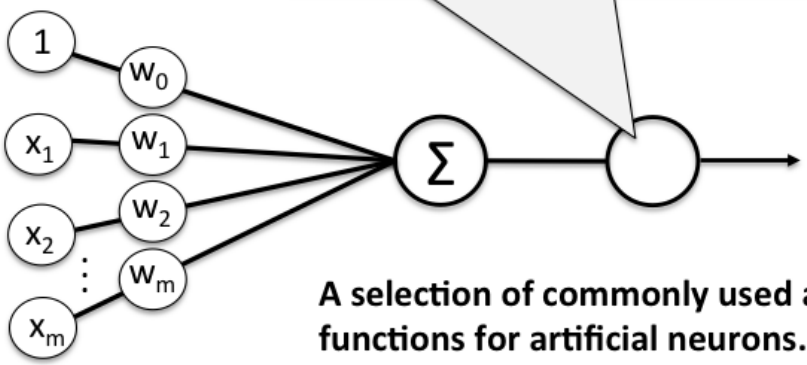
ELU

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



Collection of deep learning activation functions (2 of 2)

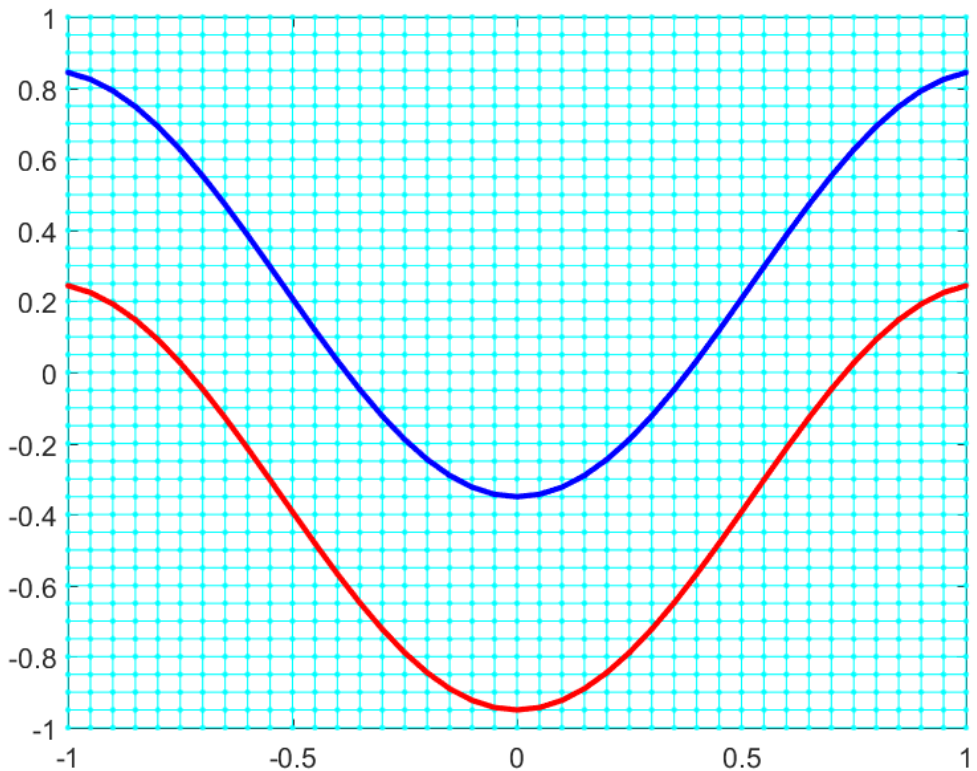
Unit step	$g(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ -1 \text{ otherwise.} \end{cases}$	
	$g(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ otherwise.} \end{cases}$	
Linear	$g(z) = z$	
Logistic (sigmoid)	$g(z) = 1 / (1 + \exp(-z))$	
Hyperbolic tangent (sigmoid)	$g(z) = \frac{\exp(2z) - 1}{\exp(2z) + 1}$	
...		



A selection of commonly used activation functions for artificial neurons.

Linear Classification Example (1 of 3)

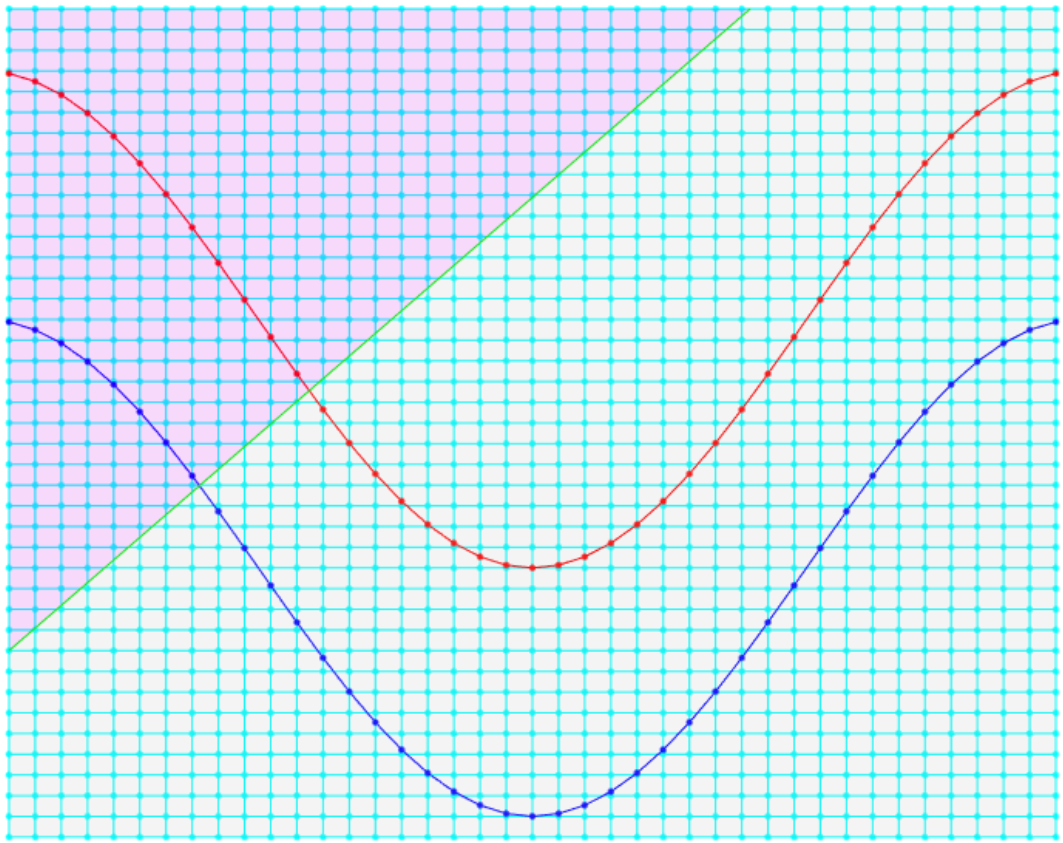
- very simple dataset, two curves on a plane



Linear Classification Example (2 of 3)

- to classify points as belonging to one or the other curve
- obvious way to visualize the behavior of any classification algorithm is to simply look at how it classifies every possible data point
- first trial is to simply separate two classes of data by dividing them with a line

Linear Classification Example (3 of 3)



Transforming the data

- as expected, the first trial failed to do the job
- how about a more complicated curve than a line?
- how about transforming the data, and creating a new representation?
- AND once we get to the final representation, `network` would just draw a line (in higher dimensions, a hyperplane) through the transformed data

Topology (1 of 3)

- Topology is the mathematical study of the properties that are preserved through deformation, twisting, and stretching of objects
- tearing, however, is not allowed
- for example, a circle is topologically equivalent to an ellipse (into which it can be deformed by stretching) and a sphere is equivalent to an ellipsoid, and so on

Topology (2 of 3)

- definition of topology leads to the following mathematical joke (Renteln and Dundes 2005)

Q: who is a topologist?

A: someone who cannot distinguish between a doughnut and a coffee cup

Topology (3 of 3)

- an n-dimensional topological space is a space (not necessarily Euclidean) with certain properties of connectedness and compactness

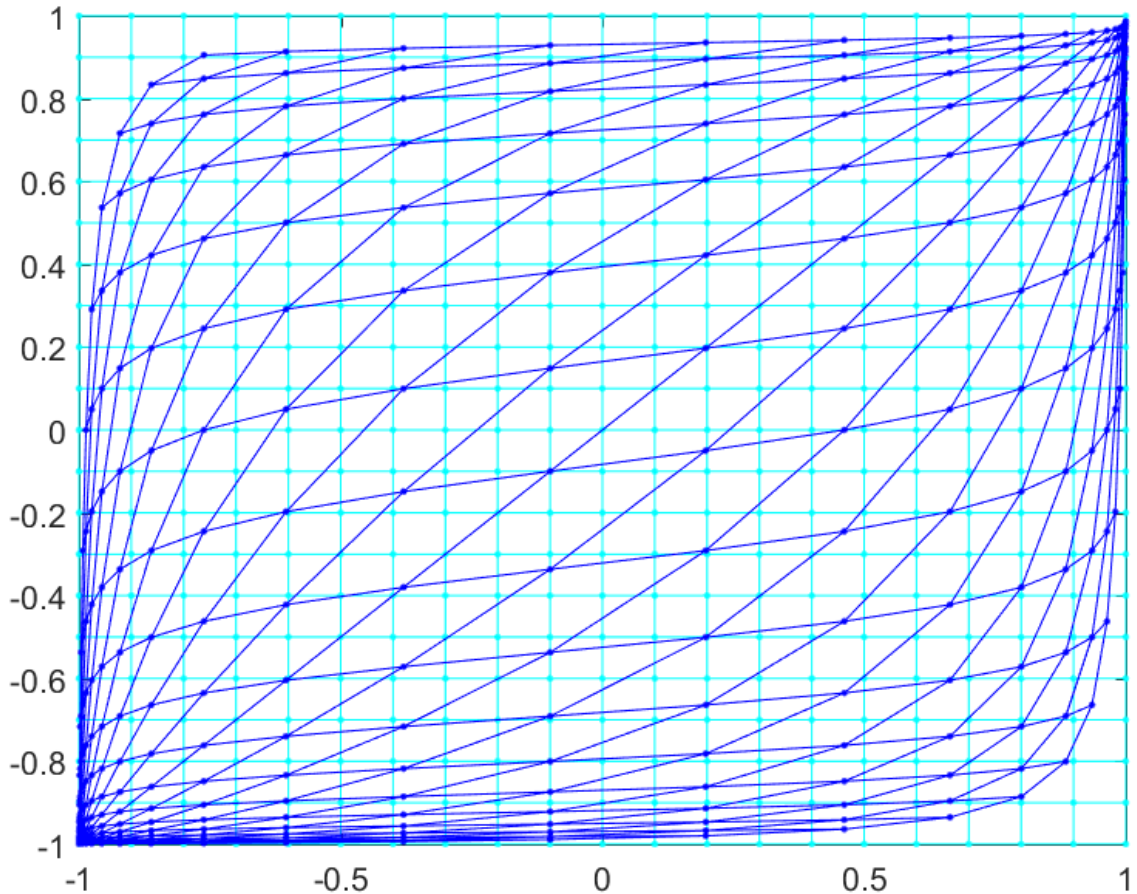
Naive approach for visualization purposes

for every grid point in the manifold

Algorithm 1: pseudo-code for transforming the manifold

for $w_{11} = -3, \ldots, 3$ **do**
  for $w_{12} = -3, \ldots, 3$ **do**
    for $w_{21} = -3, \ldots, 3$ **do**
      for $w_{22} = -3, \ldots, 3$ **do**
        for $b_1 = -1, \ldots, 1$ **do**
          for $b_2 = -1, \ldots, 1$ **do**
            $\hat{x} = \tanh(w_{11}x + w_{21}y + b_1)$
            $\hat{y} = \tanh(w_{12}x + w_{22}y + b_2)$
          end
        end
      end
    end
  end
end

Transformed Manifold



Loss function

note we seek **w** such that

$$w^\top x \geq 0 \text{ when } t = +1$$
$$w^\top x < 0 \text{ when } t = -1$$

or equivalently

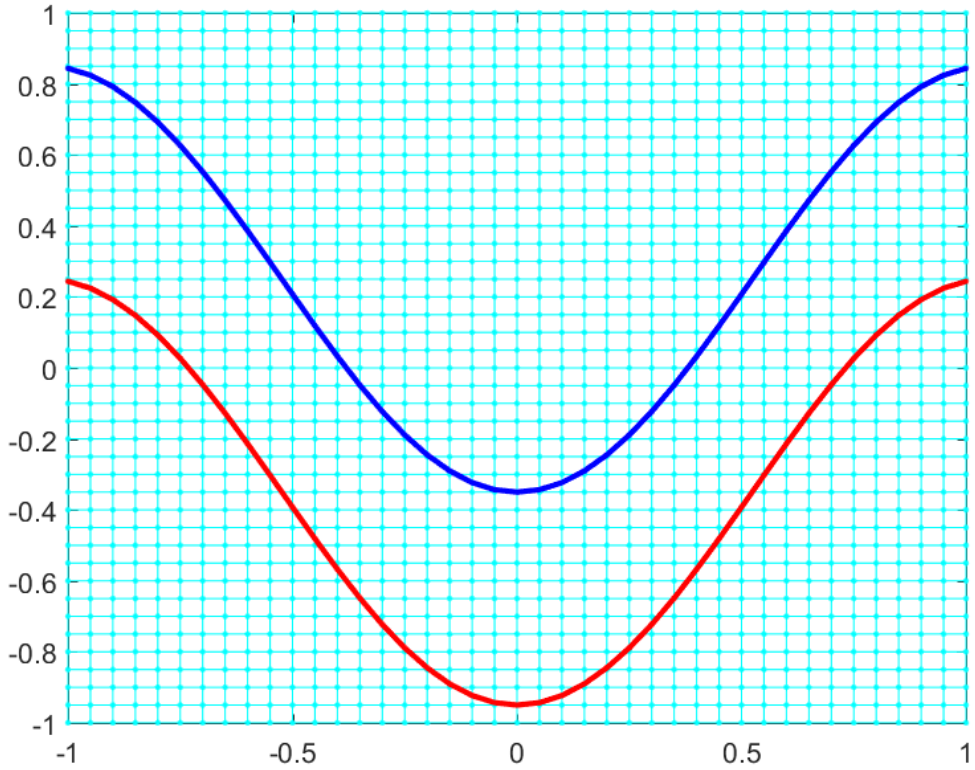
$$w^\top x_j t_j \geq 0 \quad \forall j$$

thus we seek to minimize

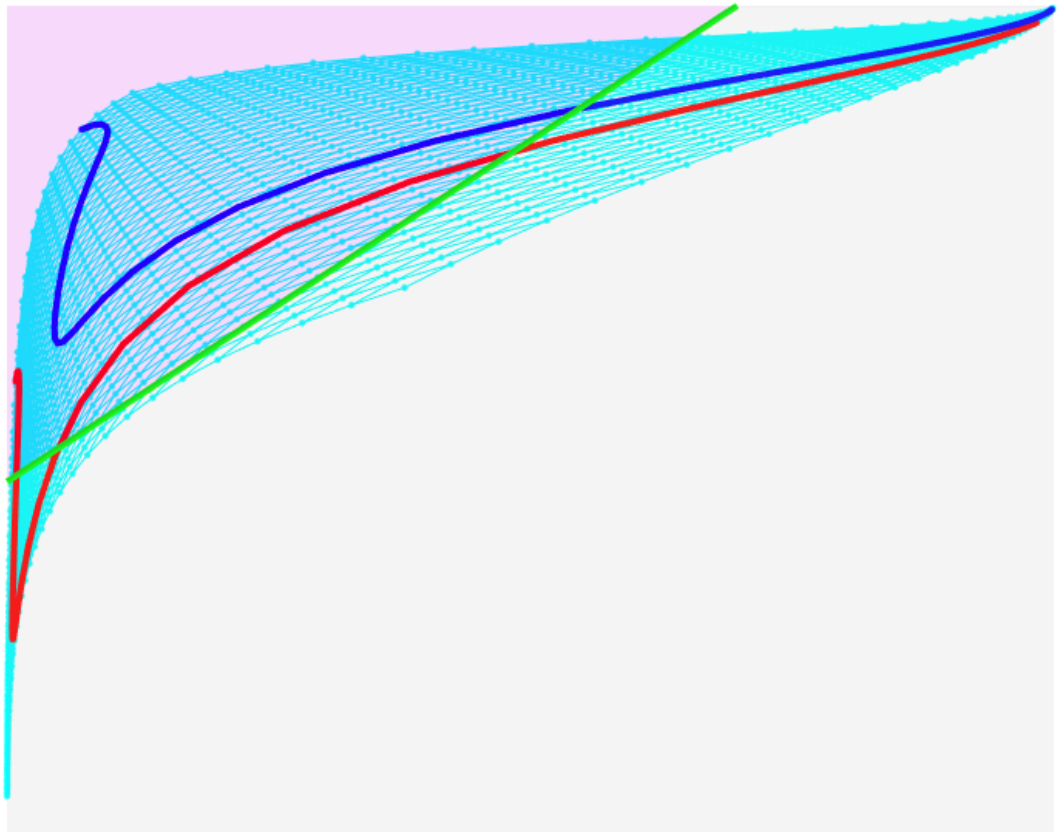
$$-\sum_{j \in \mathcal{A}} w^\top x_j t_j \geq 0 \quad \forall j$$

where $\mathcal{A}$ is the set of mis-classified inputs

Linear Classification Example (1 of 2)



Linear Classification Example (2 of 2)



A more complicated example