Artificial Intelligence

Intro to Machine Learning (ML)

1

ML - Introduction

- ML?
  - Machine Learning
  - Predictive Analytics
  - Statistical Learning
- Subjects
  - Computer Science
  - Statistics
- ML = extracting knowledge from data
- Examples
  - Recommender systems
    - Which music to listen to (Spotify)
    - Which videos to watch (Netflix)
  - Data-driven research
    - Astronomy (finding planets)
    - DNA analysis

ML - contrast

- Early A.I. systems
  - "Expert" systems
    - Hardcoded rules
    - Feasible
      - When humans have a good understanding of the process
    - Very specific
      - The logic required is specific to a single domain and task
  - Examples where it might work:
    - Fighting email spam with blacklists
    - ?
  - Examples where it will fail
    - Detecting human faces in photos
    - In general, processes where it is very hard to identify a set of rules and/or where the whole is more than the sum of its parts
- ML is "learning by example"
  - With enough examples, a solution for the task arises

Supervised ML

- Assist on decision-making by generalization from examples
- Supervised learning
  - Algorithms that learn from supervised (input, desired output) pairs
  - Researcher provides pairs (input, desired output)
    - For example, pictures with animals and the label "dog", for some
    - For example, a list of emails and the label "spam" for some
  - Algorithm automatically finds how to reach the output, from the input
    - In particular, for never seen inputs
    - The algorithm does "predictions" with a level of probabilistic certainty
  - The "supervision" (creating the dataset) is hard work
  - Well understood
  - Measurable performance

Supervised ML - more examples

- Handwritten digits recognition
  - Inputs: scans of handwritten digits
  - Outputs: their correct values
  - To create a dataset for building a ML model: collect, collect...
- Medical diagnosis from images
  - Is a tumor benign or not?
    - Inputs: images of tumors
    - Outputs: the correct classification of the tumors
    - To create a dataset for building a ML model: database of medical images + assessment by experts
- Detecting fraud in financial transactions
  - Inputs: records of financial transactions
  - Outputs: their classification as "fraud", or not
  - To create a DB for building a ML model: logs + the users' reports

arturmarques.com

Supervised ML - classification concepts

- Samples are examples of something
- Samples have "features", which are measurements of properties/attributes in them
- A collection of samples is a "dataset"
- If, for every sample, there is a label available
  - Then, the dataset is a "labelled dataset" that can be used for "supervised learning"
- The idea is to create a function F that, receiving a [new] sample, can assign it a classification
  - F(sample) = classification
  - F(X) = y

Supervised ML - train and test sets

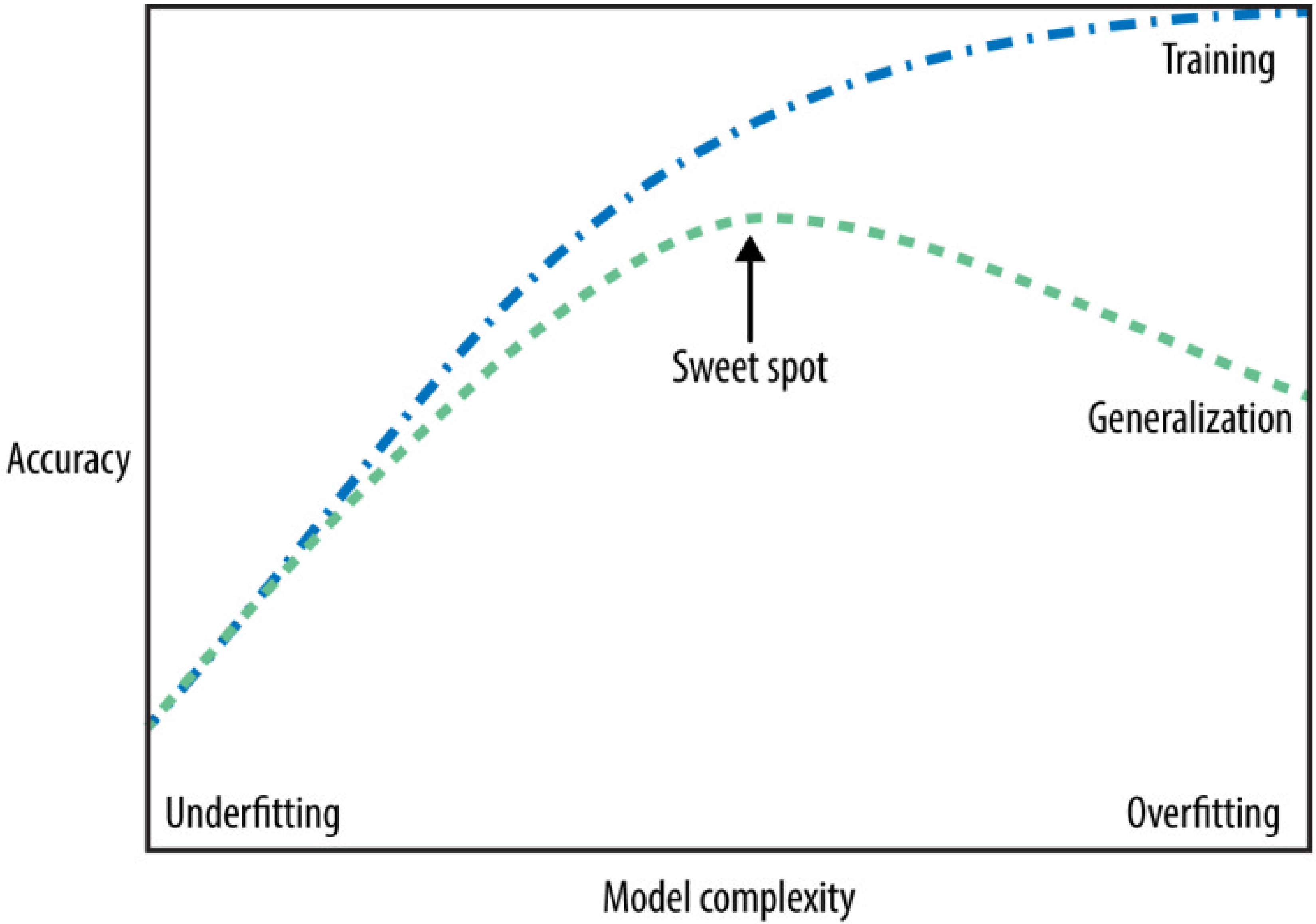
- There are known examples, AKA "samples"
  - They capture what is known, observed
- Do NOT use them all for training a model
  - Select some samples, say 75%, of all the available data
    - This is the "training set"
- Reserve some samples, say 25%, of all the data
  - To test the accuracy of the model
  - The accuracy is a measure of how correct the model is expected to be with new samples
  - The test samples, because they are not used in training, are like "new samples", although their correct classification is known

Supervised ML - some concepts

- Overfitting?
  - Model too complex, for the amount of available data
  - Fits great the available data
    - Predicts great, on the training set
    - But if it does so with too many rules, too much complexity...
      - It will, probably, not generalize well, to new data
- Underfitting?
  - Model too vague, for the amount of available data
- Ideal?
  - A balance between fitting the training set, without "gluing" to less represented samples, so it has opportunity to generalize to new samples

Supervised ML - balancing C/A/G

- Balancing complexity / accuracy, for generalization



arturmarques.com

Supervised ML using Keras

- In Python, Keras provides easy access to many datasets
  - import tensorflow.keras.datasets.mnist as mnist
  - import tensorflow.keras.datasets.fashion_mnist as fashion_mnist
  - import tensorflow.keras.datasets.cifar10 as cifar10
  - import tensorflow.keras.datasets.cifar100 as cifar100
- Take a peek at a them using the code snippet:

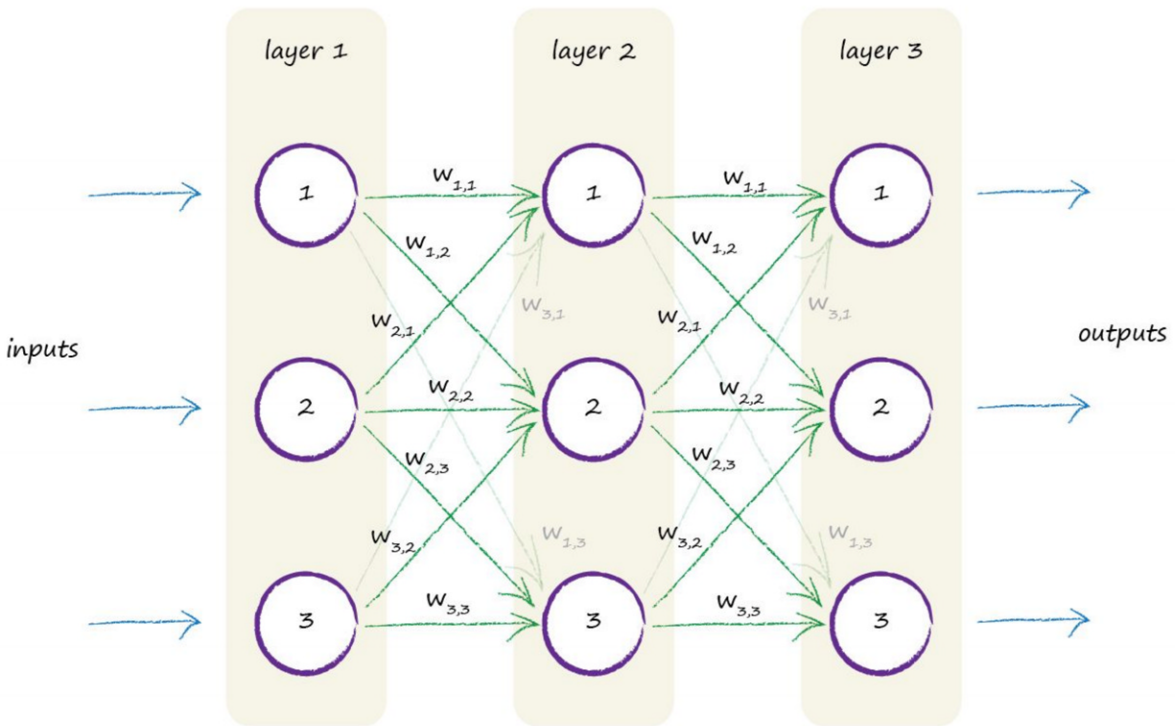
```
tupleForTraining, tupleForTesting = mnist.load_data()
displayRandomNImagesFrom(
    tupleForTraining[0],
    tupleForTraining[1]
)
```

One Hot Encoding

- In coding for classification problems, it is usual to use the "one hot encoding" technique to represent the possible classes
- For example, in the hand written digits classification problem, there will be 10 classes (0 to 9)...
- ...that can be encoded as vectors with a single 1 for the right switch
- [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.] #class 0
- ...
- [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.] #class 10
- The above representation is when using numpy.ndarray objects, which are arrays of floats

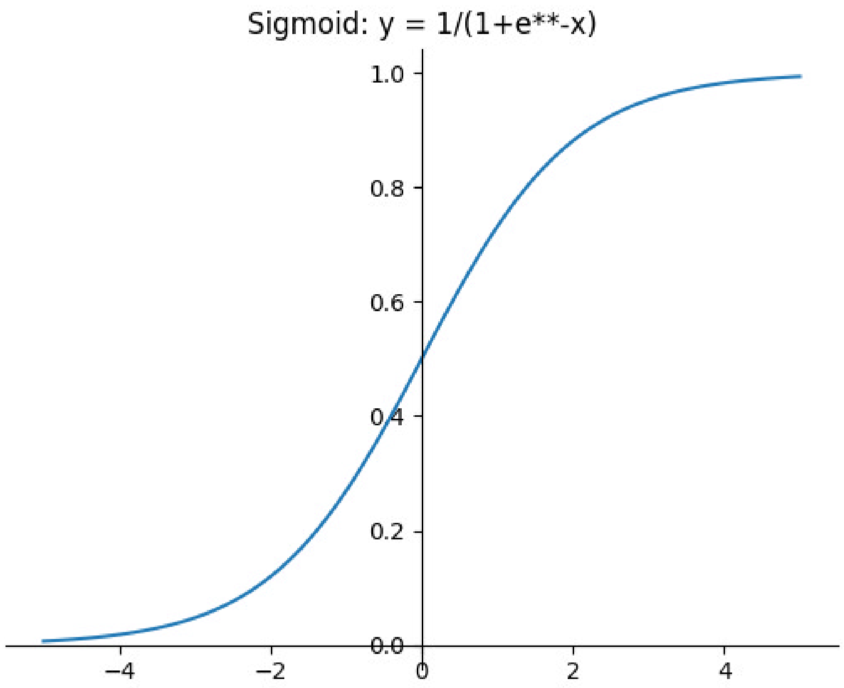
Neural Networks can have billions of nodes

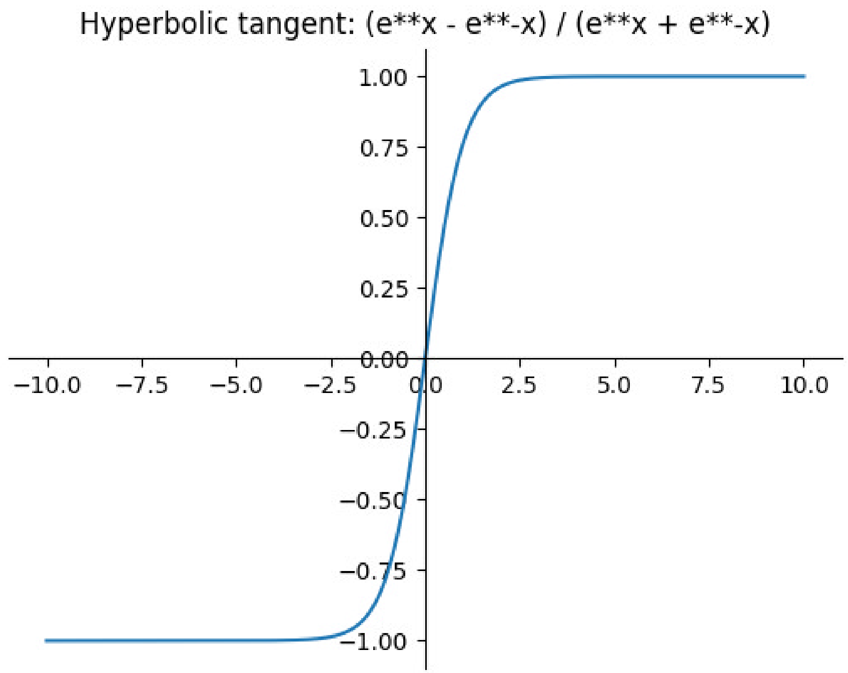
- A neuron "fires" a signal to the "next" neuron
  - Different signals (inputs) are subject different to "intensities"
- Think of "layers" of nodes
  - Each signal that is input to a node is "weighted"
  - All the weighted inputs are summed and subjected to an activation function, which returns an output, input to all the next



Brain neurons as an approach

- A bio-neuron does not behave as a linear function
  - output-via-axon-terminals != constant * input-via-dendrites + bias
  - Output is suppressed, until it reaches a level when it then triggers an output
    - Threshold
  - A function that takes inputs and returns outputs taking into account some threshold is an "activation function"; e.g.: sigmoid aka logistic, h-tangent





Neural Networks for Classification in ML

- A computational solution to find the optimal W and b
- Plain algebra could be used
- But there is noise and unknown data
- The theoretical function is actually UNKNOWN
  - Probably not linear at all
- The challenge is:
  - Find the values for W and b
  - That best FIT the data that is actually available
- Find an approximate function that maps inputs to outputs, as closely as possible
  - But that does NOT just memorize the example mappings
  - This is an optimization problem

Matrices multiplication

- Matrix multiplication is
  - a binary operation (means 2-operands) that takes a pair of matrices and returns another matrix
  - NOT commutative (order of operands is relevant)
- The standard way to multiply matrices is called "row by column" multiplication
  - The element at the i-th row and j-th column of the resulting matrix is calculated by multiplying each element of the i-th row of the first matrix by the corresponding element of the j-th column of the second matrix and adding the results.
  - Done for all combinations of rows from the first matrix and columns from the second matrix.
  - The number of columns in the first matrix must be equal to the number of rows in the second matrix. If the first matrix is of dimension m * n, and the second matrix is of dimension n * p, then the resulting matrix will be of dimension m * p.

Matrices multiplication - examples

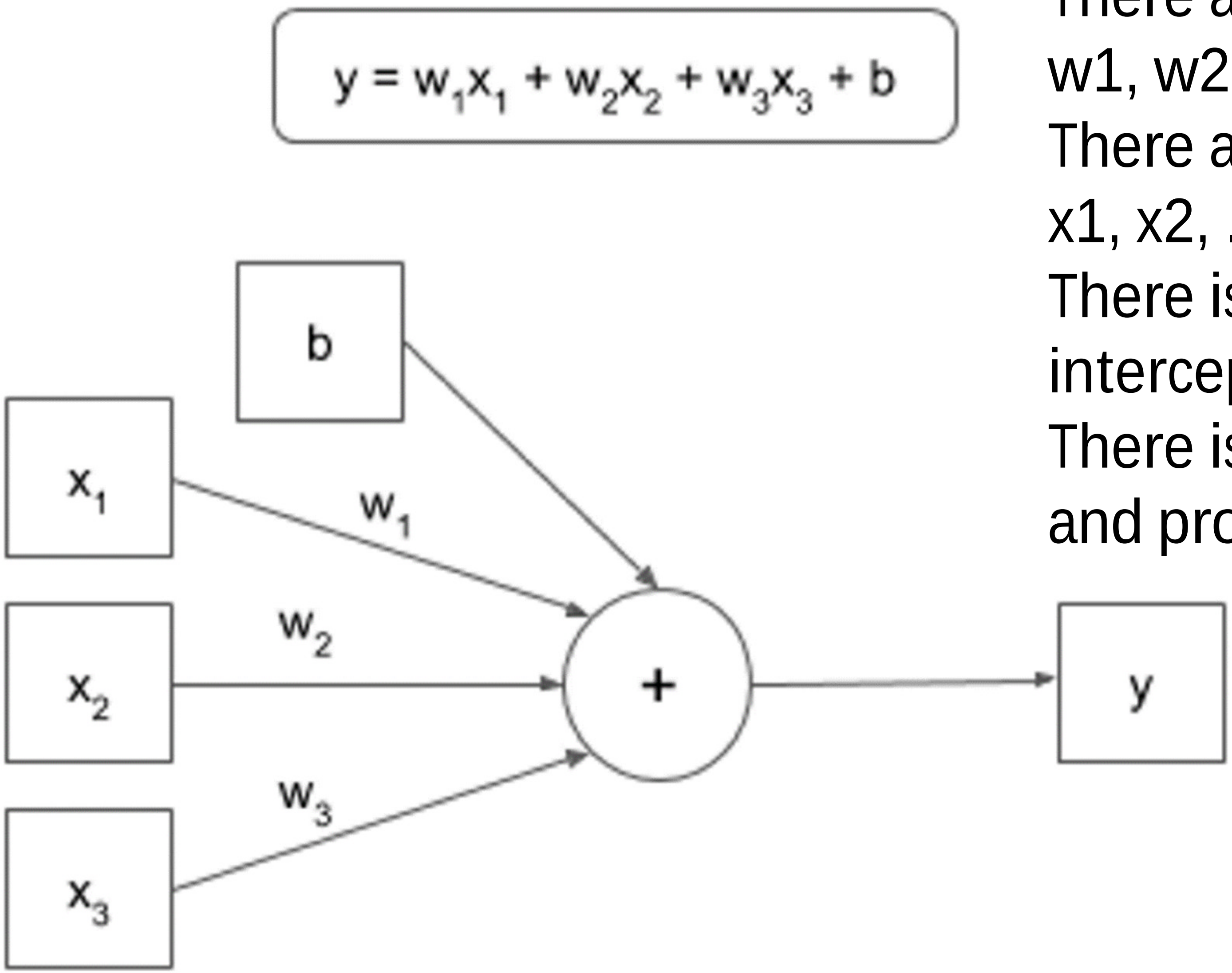
- A(2,3) =
  a b c
  d e f
- B(2,3) =
  g h i
  j k l
- AB can NOT be computed

Matrices multiplication - examples

- M1(3,3)
- a@(0,0) b@(0,1) c@(0,2)
- d@(1,0) e@(1,1) f@(1,2)
- g@(2,0) h@(2,1) i@(2,2)
- M2(3,3)
- j@(0,0) k@(0,1) l@(0,2)
- m@(1,0) n@(1,1) o@(1,2)
- p@(2,0) q@(2,1) r@(2,2)

- M1*M2
- a*j + b*m + c*p@(0,0)   a*k + b*n + c*q@(0,1)   a*l + b*o + c*r@(0,2)
- d*j + e*m + f*p@(1,0)   d*k + e*n + f*q@(1,1)   d*l + e*o + f*r@(1,2)
- g*j + h*m + i*p@(2,0)   g*k + h*n + i*q@(2,1)   g*l + h*o + i*r@(2,2)

Linear Equations Representations



This is a graph representation of a linear equation
There are "weights" w1, w2 ... , wn
There are terms or "samples" x1, x2, ... , xn
There is a crossing point / intercept / "bias" b
There is a "node" doing a sum and producing an output y

Linear Equations Representations

Vectorized Implementation

$$Y = W \,.\, X + b$$

Where $W = [\, w_1 \;\; w_2 \;\; w_3 \,]$

and $X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$

This is a vectored representation of a linear equation

There is the W matrix (1D array)

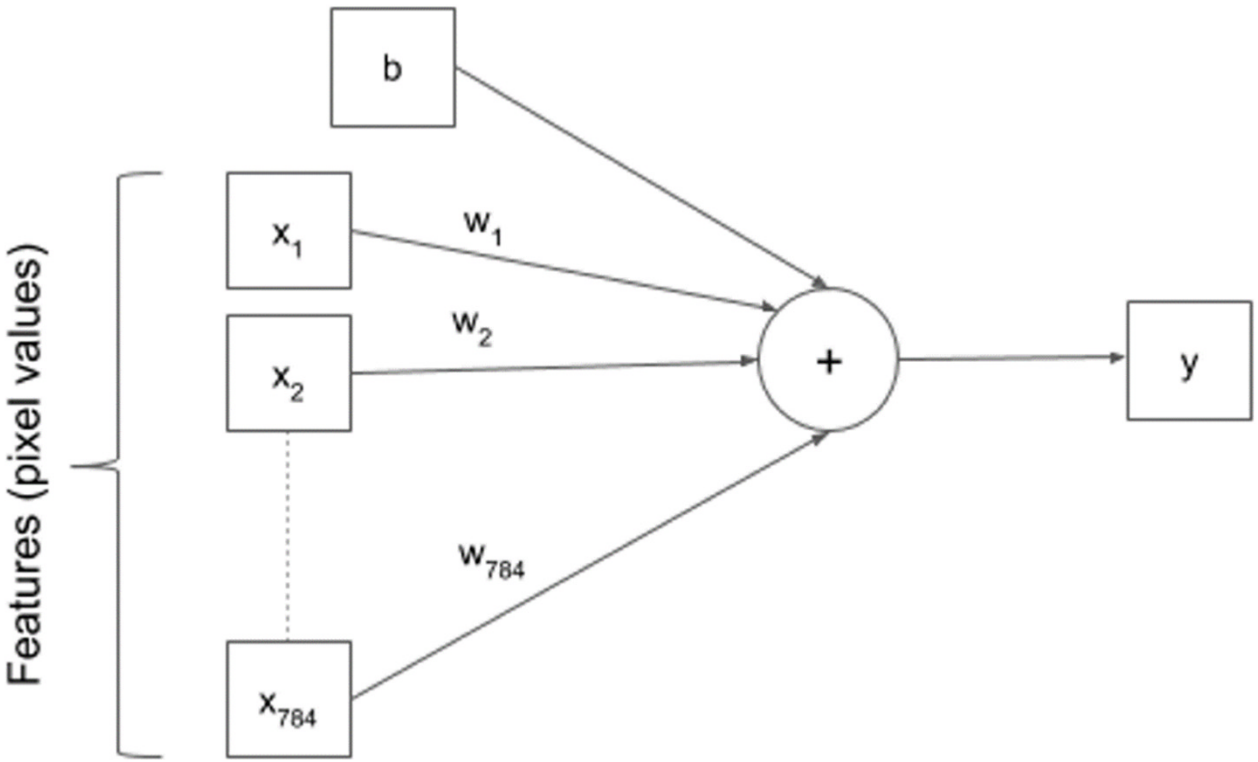
W = [w1, ... , wn]

There is a single column samples matrix

X = [x1, ... , xn]

There is a crossing point / intercept / "bias" b

If a problem can be modeled as a linear equation...



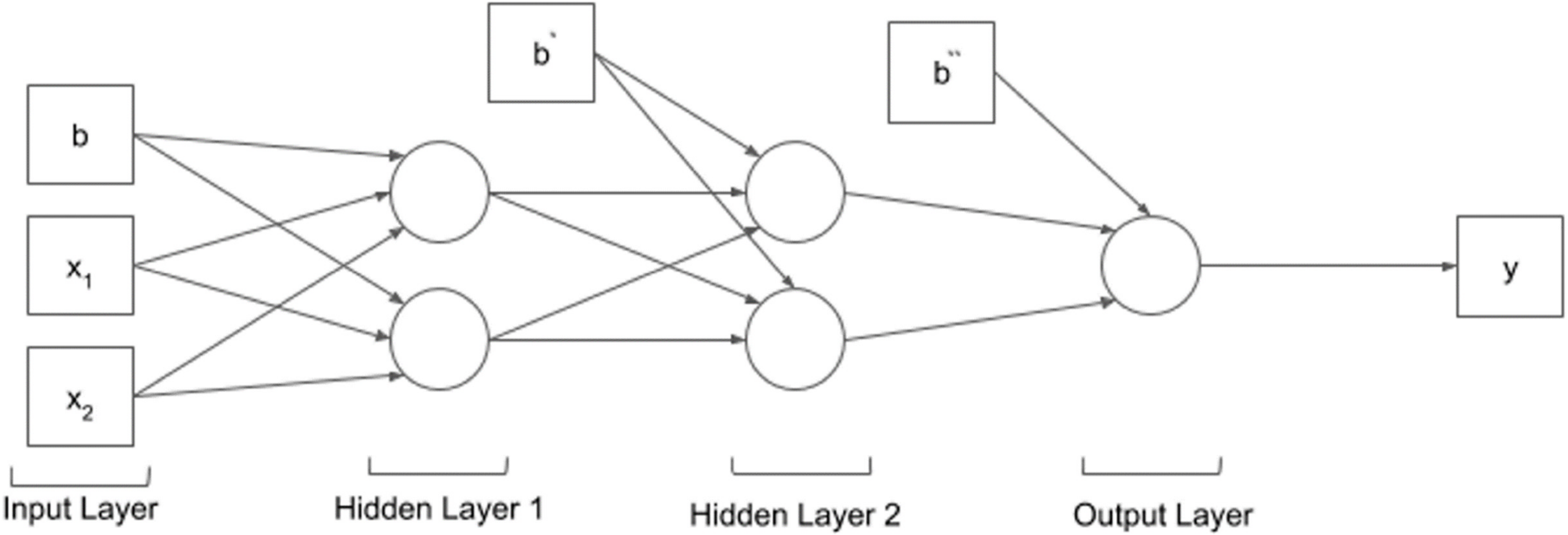
Each sample is an image
28 x 28 pixels = 784 pixels in total
Each pixel is a "feature", an attribute, of the image
Each feature is to be multiplied by a weight

The "Shape" of each example is 28x28

In the MNIST example project there are 60K samples for training + 10K samples for testing the training results

Each feature, each x, is a single pixel
Quite a long linear equation!

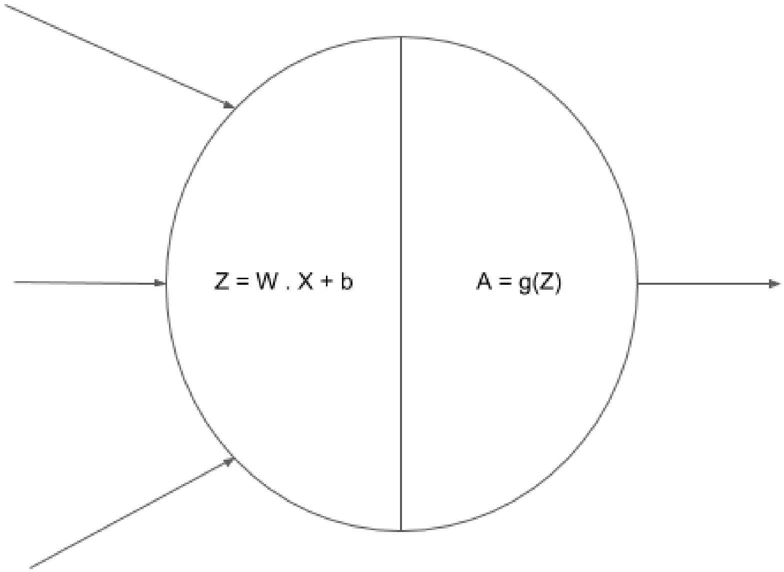
Neural Network - Model Example



Graph: 784 inputs ; 10 possible outputs (Y); in-between sums and "activation functions" in HL
By cascading (and injecting non-linearity) to the linear sums, a much more complex function is possible
The difference (to a linear eq) is that not only the sum happens, but also an activation function
Activation function?
- bio (neuron) inspiration : takes the output signal from the previous cell and converts it to a form that can be taken as input to the next cell
- restricts values to controlled intervals
- adds non-linearity to the neural network
- contribute to change in W (the weights) to reduce loss, epoch (iteration) after epoch

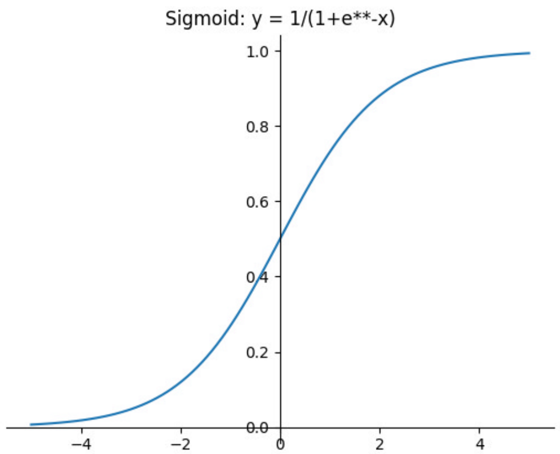
arturmarques.com

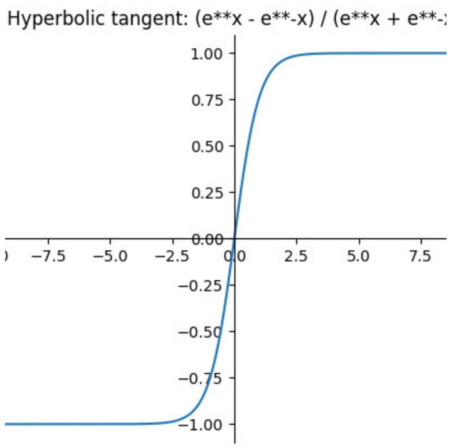
So, at each node of each HL

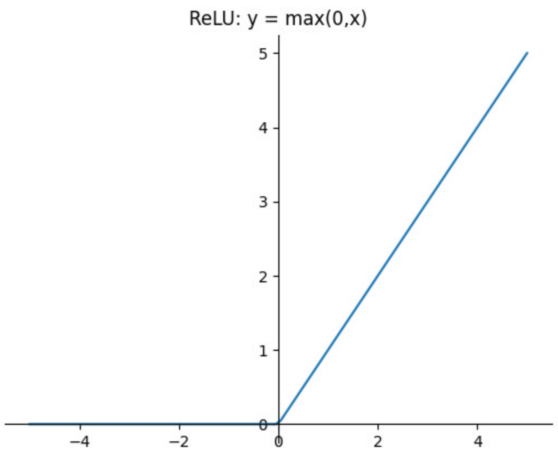


The linear equation (the sum W.X + b)
The activation function (A)
There is an entire math field dedicated to the study of functions that can inject non-linearity and do so in a computationally efficient way.

Same A names:
Sigmoid - Historical relevance, less used now. Computationally expensive, non-zero centered output, derivative always in 0..0.25
HTangent - a zero-centered (f(0)=0) sigmoid
ReLU - Rectified Linear Unit : f(x) = max(0,x)







References

- Free book on the math of ML:
  - http://statweb.stanford.edu/~tibs/ElemStatLearn/
- scikit-learn ML library
  - https://scikit-learn.org/stable/