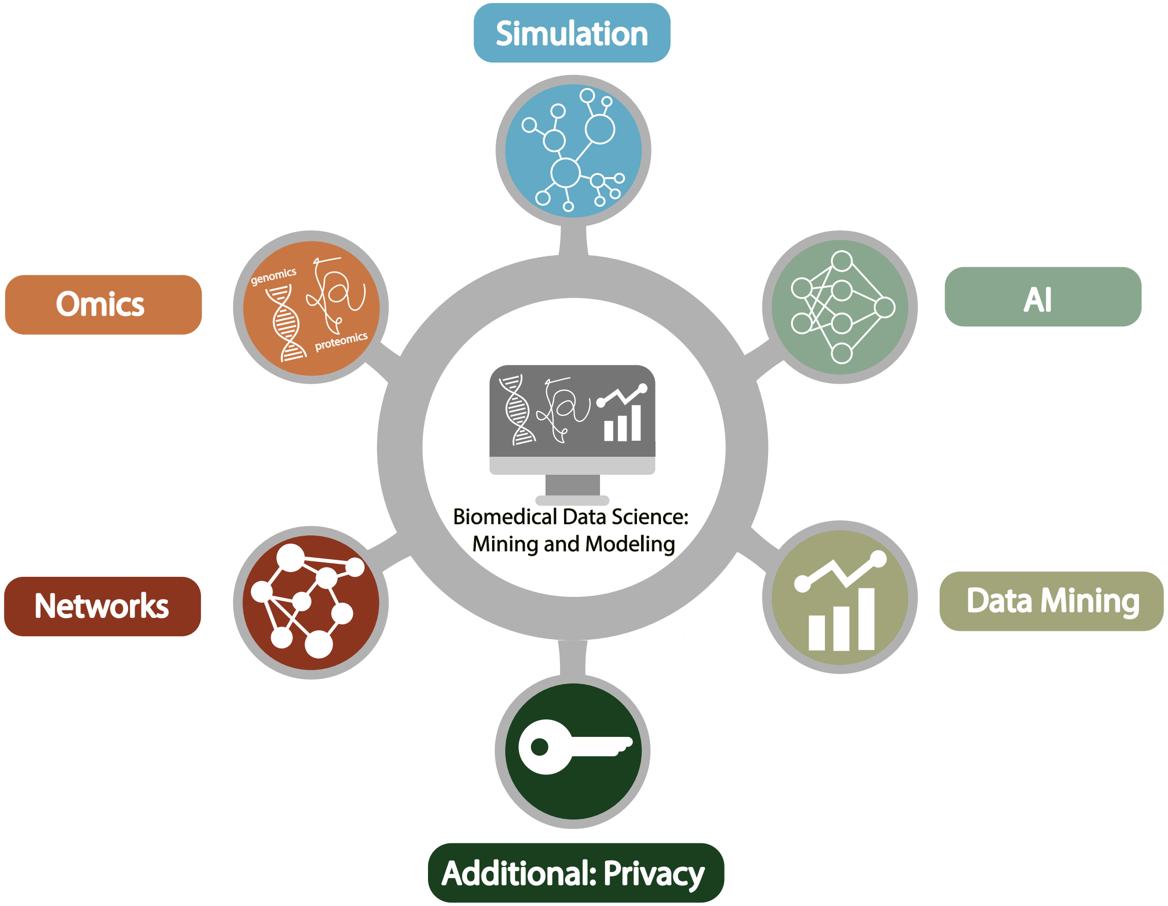
Biomedical Data Science (GersteinLab.org/courses/452)

Transition from Mining to Modeling (23i3)



Last edit in spring '23.
Very similar to 2019's pack #14.
New video in '23.

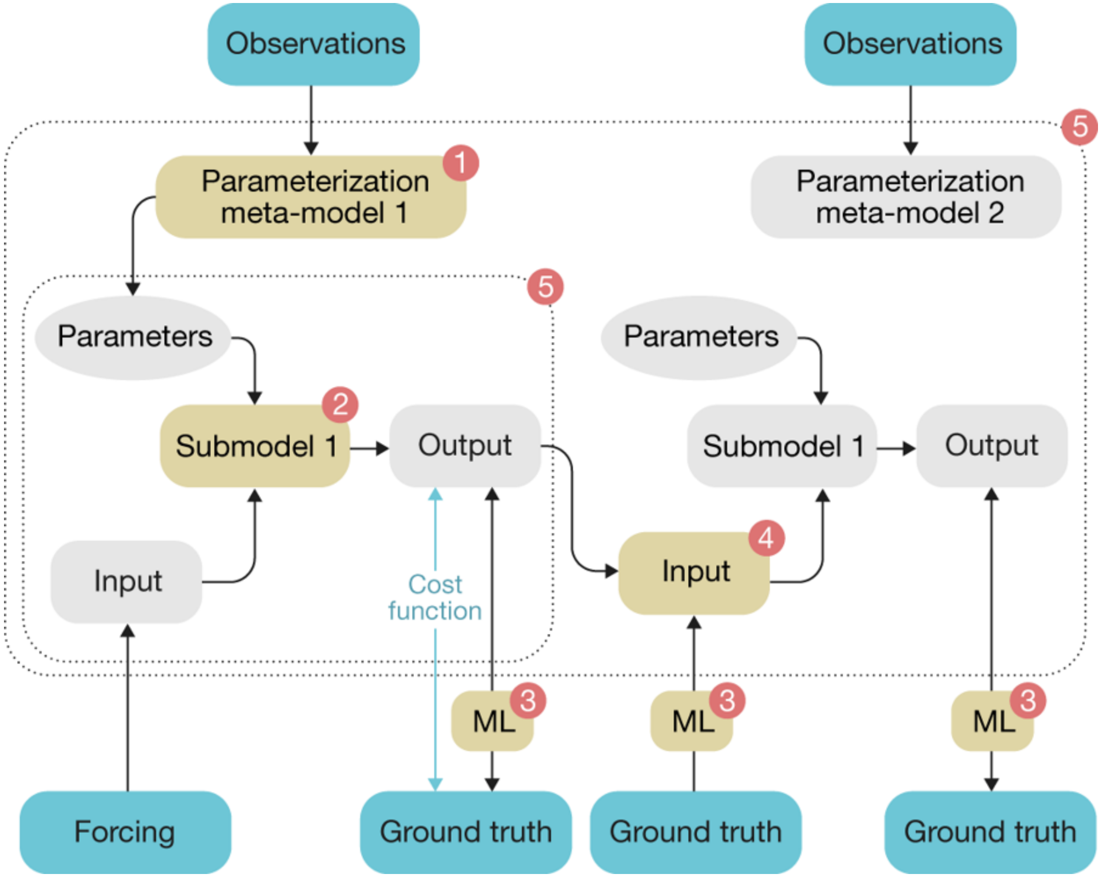
Mark Gerstein
Yale U.

Combining Mining & Modeling

- Complementarity of physical & ML approaches
  - “Physical approaches in principle being directly interpretable and offering the potential of extrapolation beyond observed conditions, whereas data-driven approaches are highly flexible in adapting to data”
- Hybrid #1: ML into physical
  - e.g., Emulation of specific parts of a physical model for computational efficiency
  - More..
- Hybrid #2:
  Physical knowledge can be integrated into ML framework
  - Network architecture
  - Physical constraints in the cost function
  - Expansion of the training dataset for under sampled domains (i.e., physically based data augmentation) [More….]

https://www.nature.com/articles/s41586-019-0912-1
2019 Nature volume 566, pages195–204 (2019)

Hybrid #1: ML into physical models



(1) Improving parameterizations

(2) Replacing a ‘physical’ sub-model with a machine learning model

(3) Analysis of model–observation mismatch

(4) Constraining submodels

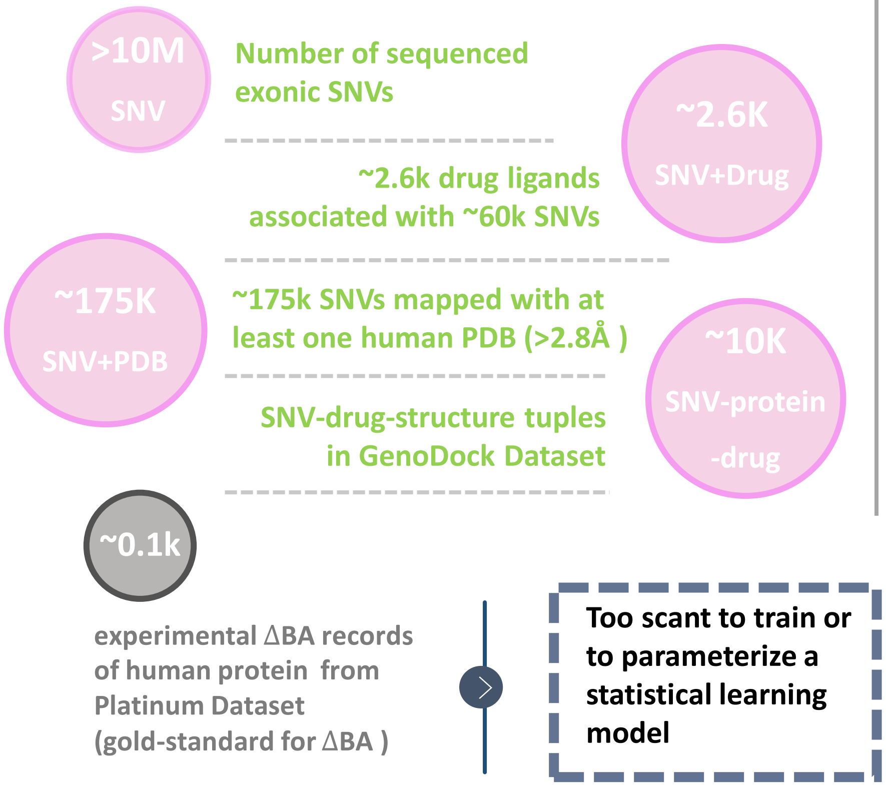
(5) Surrogate modelling or emulation

https://www.nature.com/articles/s41586-019-0912-1
2019 Nature volume 566, pages195–204 (2019)

Example of Hybrid #2: Integrating Physical Knowledge into Machine Learning

Physical Data Augmentation for Hybrid Physical-Statistical Model Construction: Example for Building a Model to Predict Sensitivity of Drug Binding to a SNV in a Protein

The Major Hurdle: Highly Scant Ligand Binding Assay Data for ΔBA



>10M SNV

Number of sequenced exonic SNVs

~2.6K SNV+Drug

~2.6k drug ligands associated with ~60k SNVs

~175K SNV+PDB

~175k SNVs mapped with at least one human PDB (>2.8Å)

~10K SNV-protein -drug

SNV-drug-structure tuples in GenoDock Dataset

~0.1k

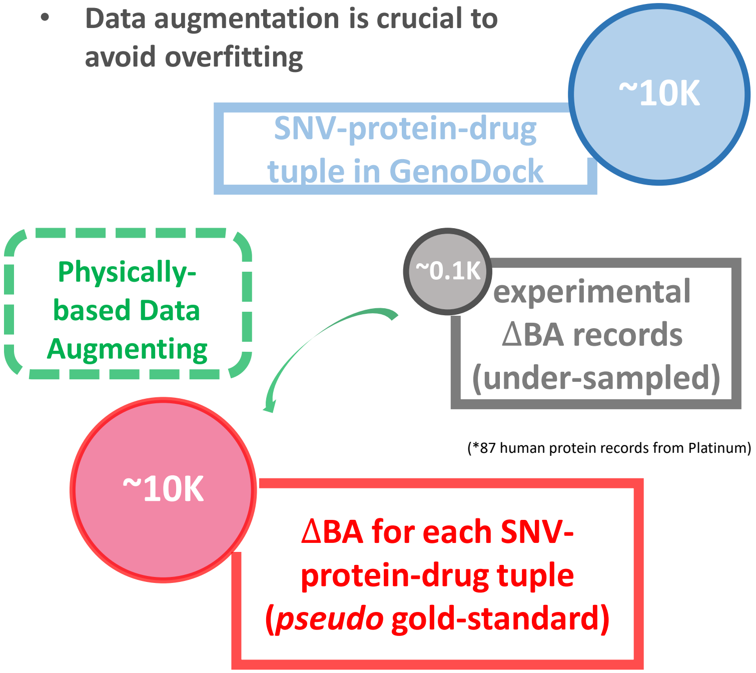
experimental ΔBA records of human protein from Platinum Dataset (gold-standard for ΔBA)

Too scant to train or to parameterize a statistical learning model

The Physically-based Data Augmentation Approach: Leveraging Physical Calculations of ΔBA to Fill the Gap

(Reichstein *et al.*, **Nature**, 2019 & Xie et al., preprint, 2018)

- Expansion of the training dataset for under sampled domains
- Data augmentation is crucial to avoid overfitting



~10K

SNV-protein-drug tuple in GenoDock

Physically-based Data Augmenting

~0.1K

experimental ΔBA records (under-sampled)

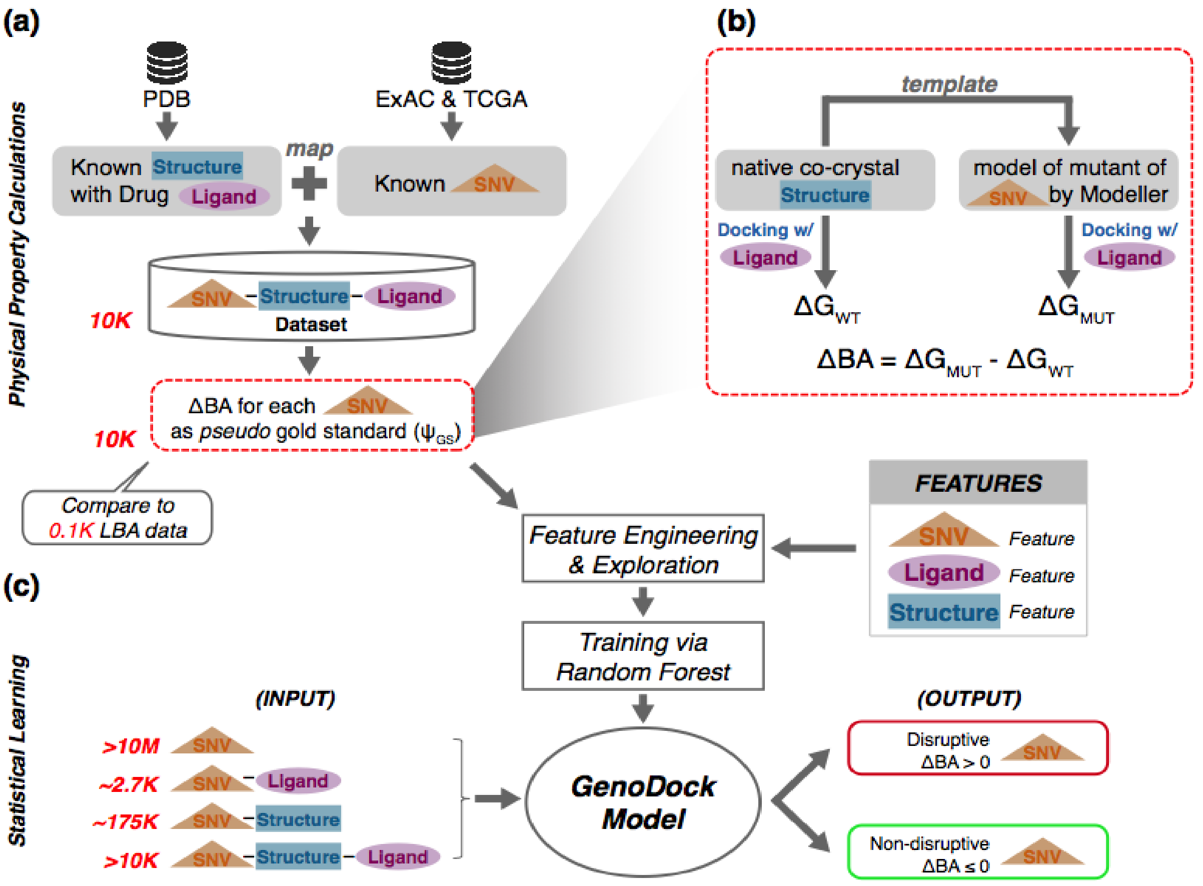
(*87 human protein records from Platinum)

~10K

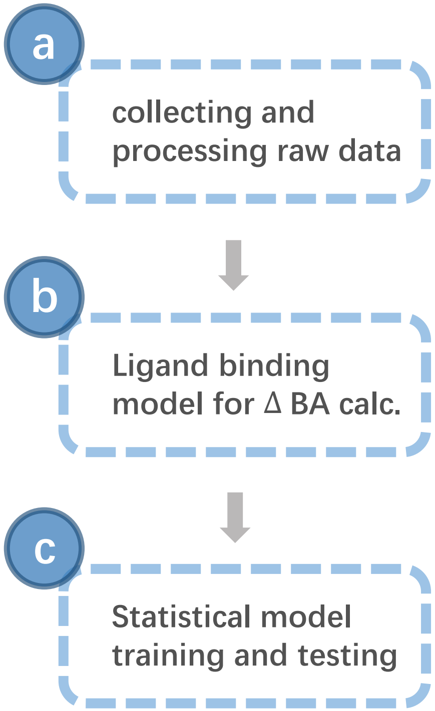
ΔBA for each SNV-protein-drug tuple (*pseudo* gold-standard)

[Wang et al. Structure ('19)]

Framework of the GenoDock Project - from Dataset Preparation to Model Construction


(a)
Physical Property Calculations
PDB
ExAC & TCGA
Known Structure with Drug Ligand
map
Known SNV
SNV - Structure - Ligand Dataset
10K
ΔBA for each SNV as *pseudo* gold standard (ψGS)
10K
Compare to 0.1K LBA data
(b)
template
native co-crystal Structure
model of mutant of SNV by Modeller
Docking w/ Ligand
Docking w/ Ligand
ΔGWT
ΔGMUT
ΔBA = ΔGMUT - ΔGWT
(c)
Statistical Learning
FEATURES
SNV Feature
Ligand Feature
Structure Feature
Feature Engineering & Exploration
Training via Random Forest
(INPUT)
>10M SNV
~2.7K SNV - Ligand
~175K SNV - Structure
>10K SNV - Structure - Ligand
GenoDock Model
(OUTPUT)
Disruptive ΔBA > 0 SNV
Non-disruptive ΔBA ≤ 0 SNV

Wang *et al.*, accepted by *Structure*, 2019



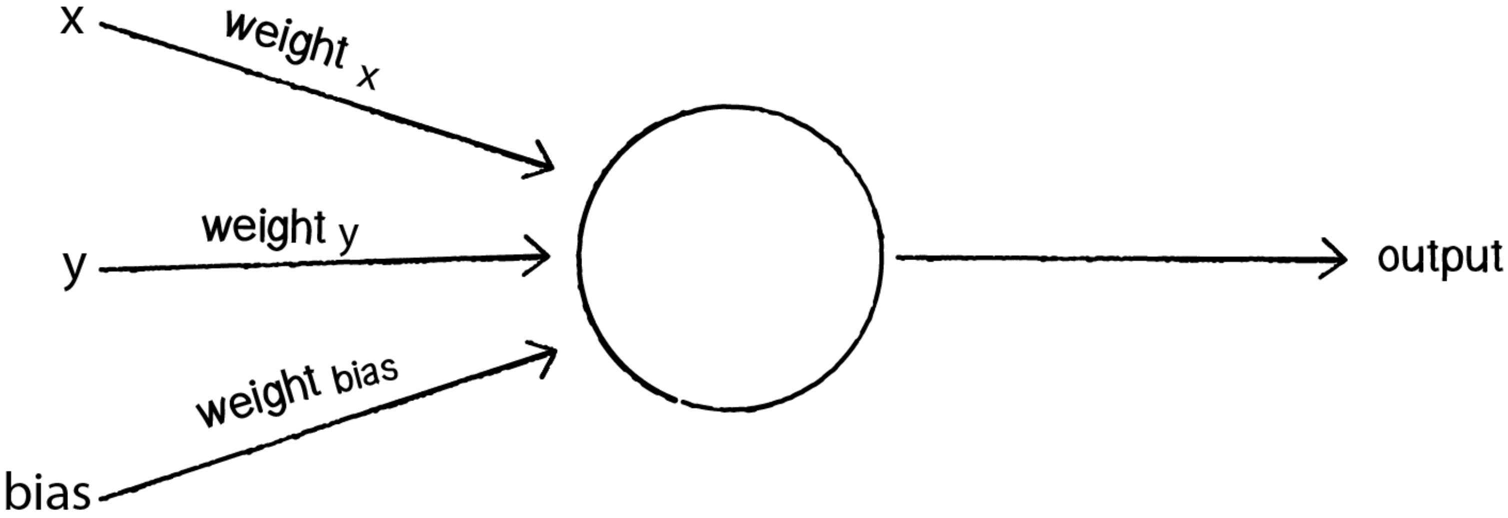
a collecting and processing raw data

b Ligand binding model for Δ BA calc.

c Statistical model training and testing

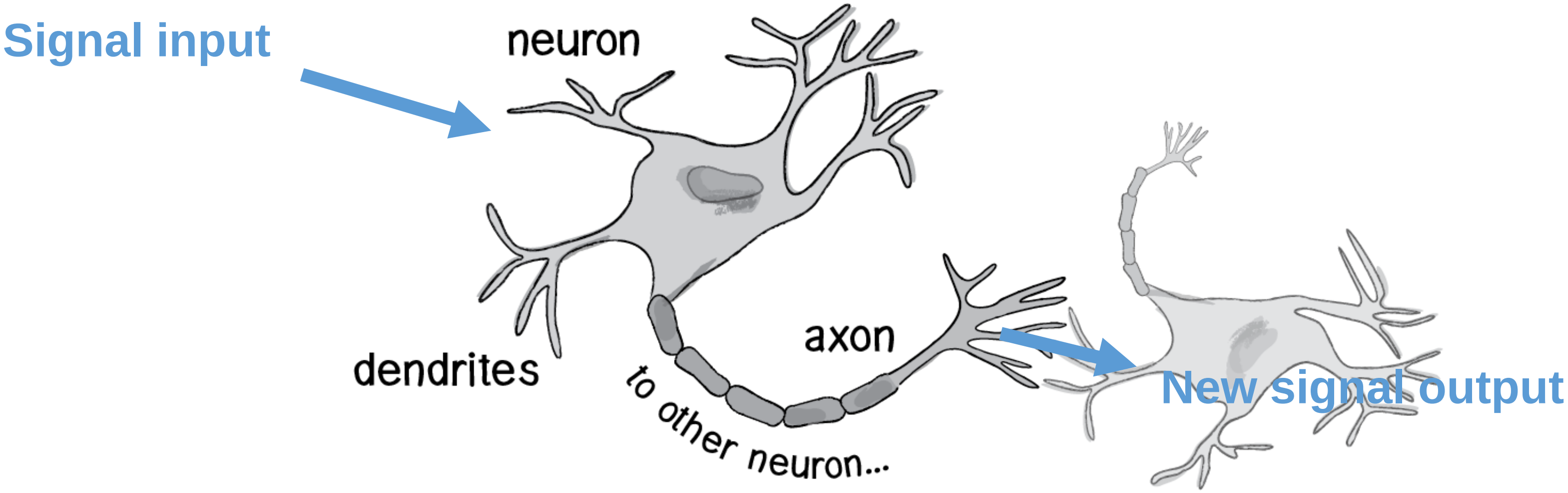
[Wang et al. Structure ('19)]

Perceptron



- A perceptron is the **simplest neural network** possible: a computational model of a single neuron
- Works as a linear classifier with n-1 dimension hyperplane
- It computes the elementary logical functions we usually think of as underlying computation, functions such as AND, OR, and NAND

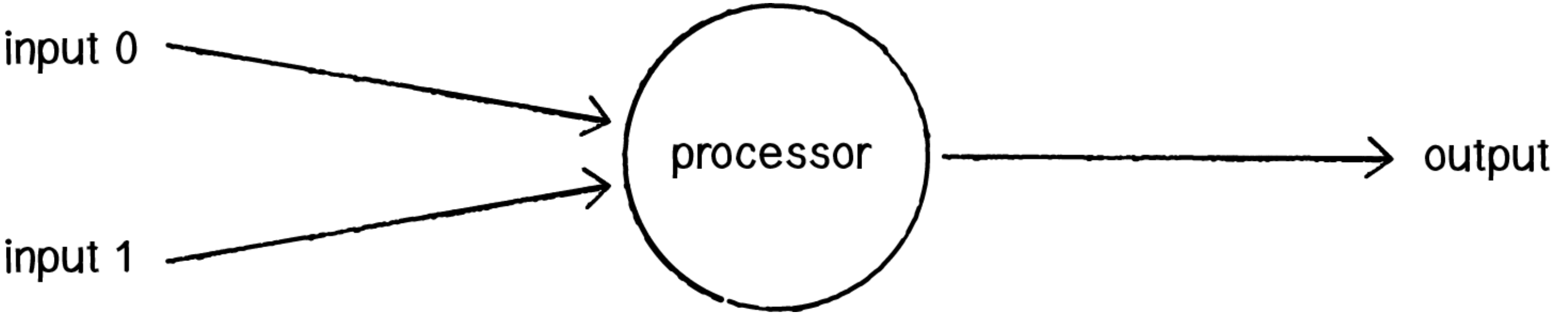
Neural Networks



- Artificial Neural Network is loosely connected to biological neural networks
- A simple artificial node known as “neuron” (or “unit”) has
  1. Inputs
  2. Outputs
  3. Adaptive weights
  4. Activation

Daniel Shiffman - The Nature of Code

Example



1. **Multiply each input by its weight**.

2. **Sum** all of the weighted inputs.

3. Compute the output of the perceptron based on that sum passed through an activation function (e.g., the **sign of the sum**).