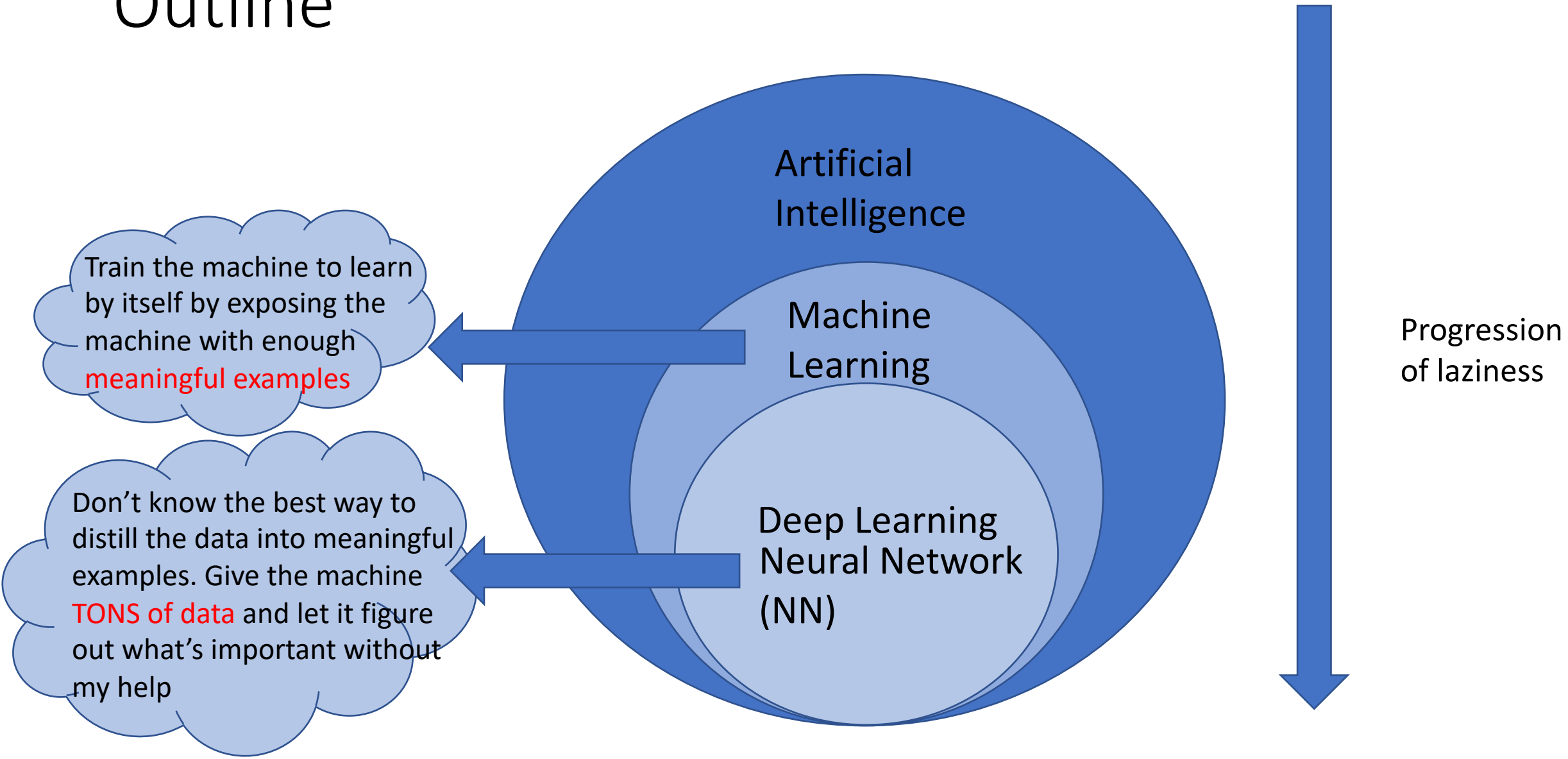
Machine Learning

Yitong

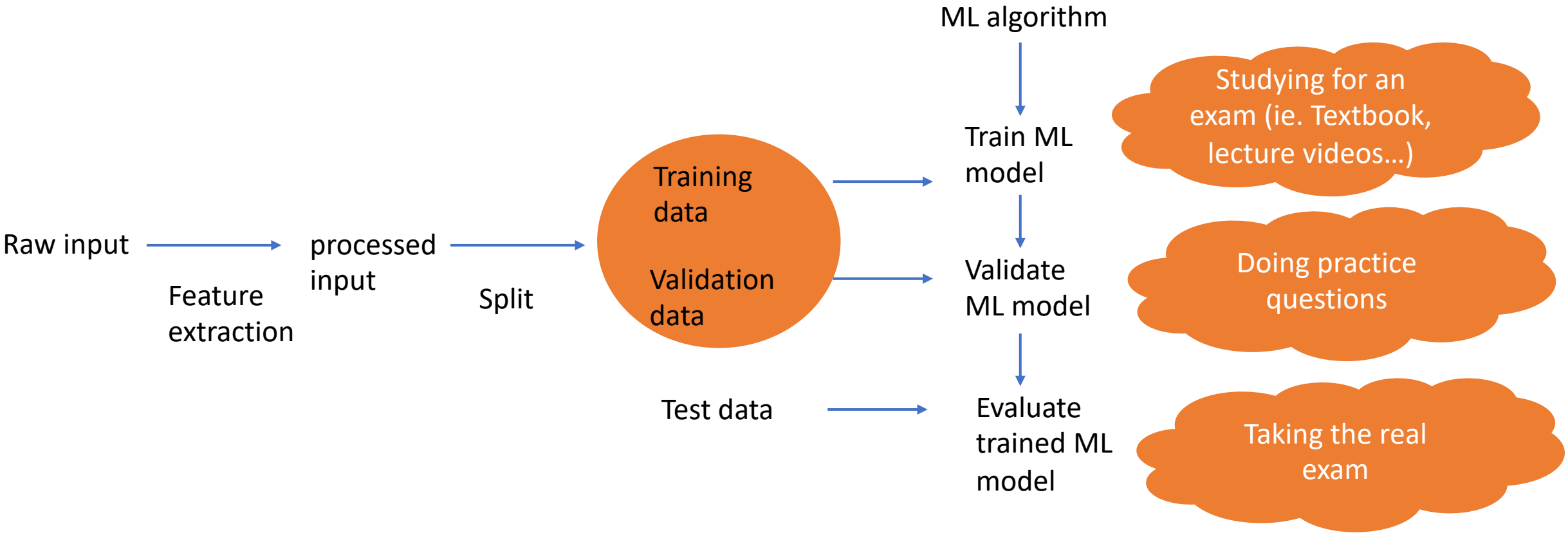
4.29 update

- Tested the pulse classification plots & algs on PREM webpage
  - Most plots are showing properly
  - The plots with [SS] are not showing up –> special characters [] → change [] to __
  - Re-group the pulse plots with their S1, S2, SPE, MPE, SE sub-plots → change titles
  - Fraction algs are returning value 0 on PREM webpage
- The Hitchhiker's Guide to Machine Learning
  - NERSC videos on basic ML, Deep Learning, Neural networks
  - LZ ML Jupiter notebook tutorials
  - Search for the possibility of ML + data quality checks

Outline

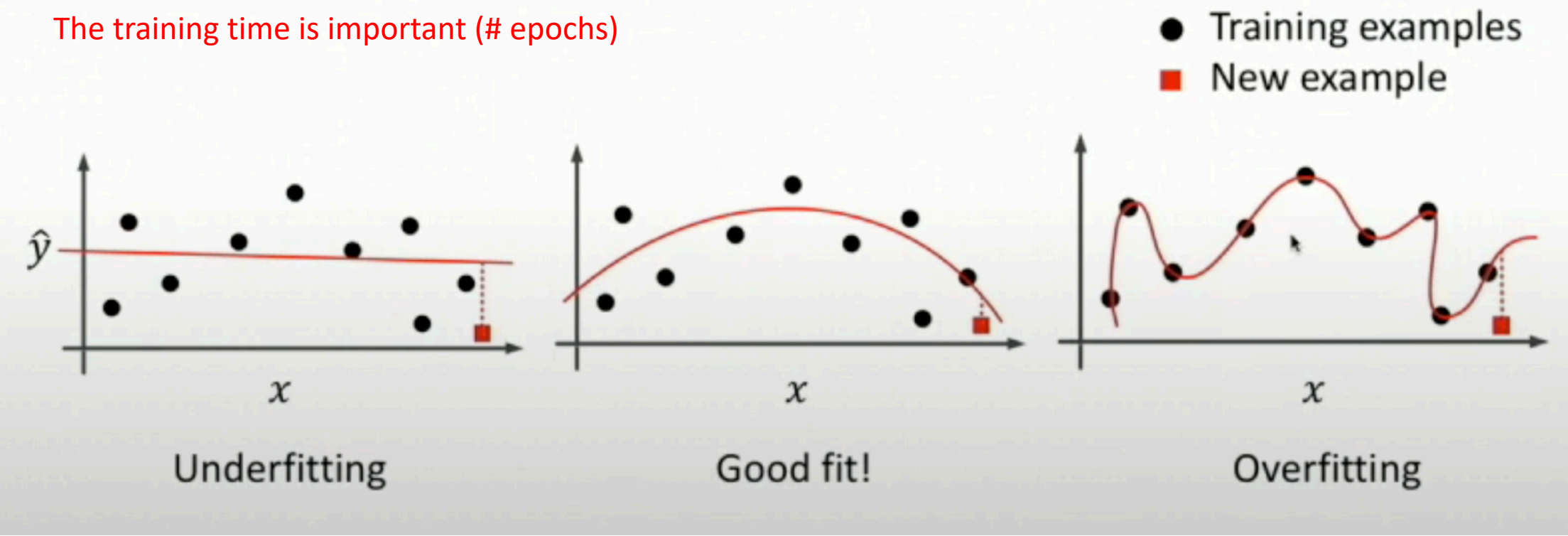

Train the machine to learn by itself by exposing the machine with enough meaningful examples
Don’t know the best way to distill the data into meaningful examples. Give the machine TONS of data and let it figure out what’s important without my help
Artificial Intelligence
Machine Learning
Deep Learning Neural Network (NN)
Progression of laziness

ML Workflow


ML algorithm
Raw input
Feature extraction
processed input
Split
Training data
Validation data
Test data
Train ML model
Validate ML model
Evaluate trained ML model
Studying for an exam (ie. Textbook, lecture videos…)
Doing practice questions
Taking the real exam

Training

The training time is important (# epochs)



Too generalized

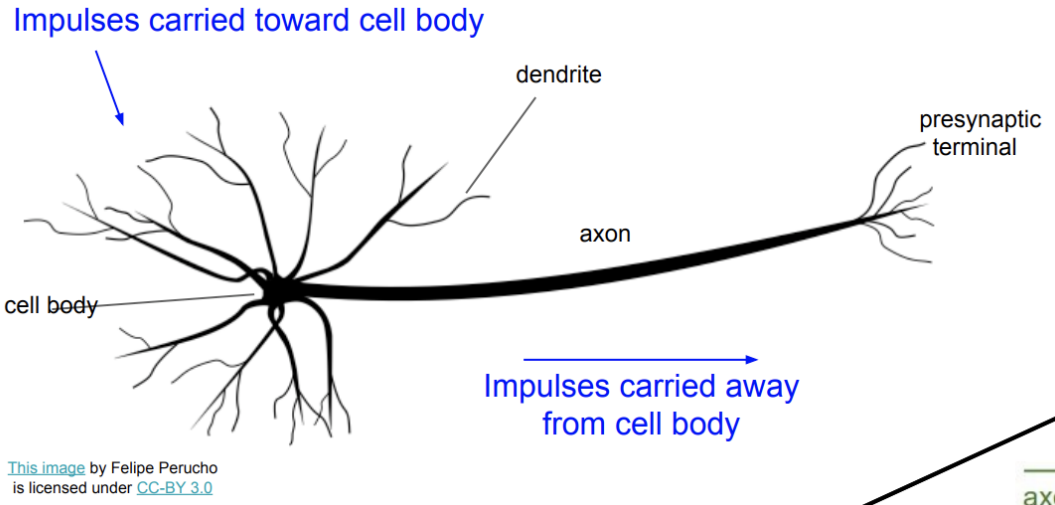
Too accurate to a specific training data set

Optimal Training

Loss Function
Y[predict] vs. Y[real]

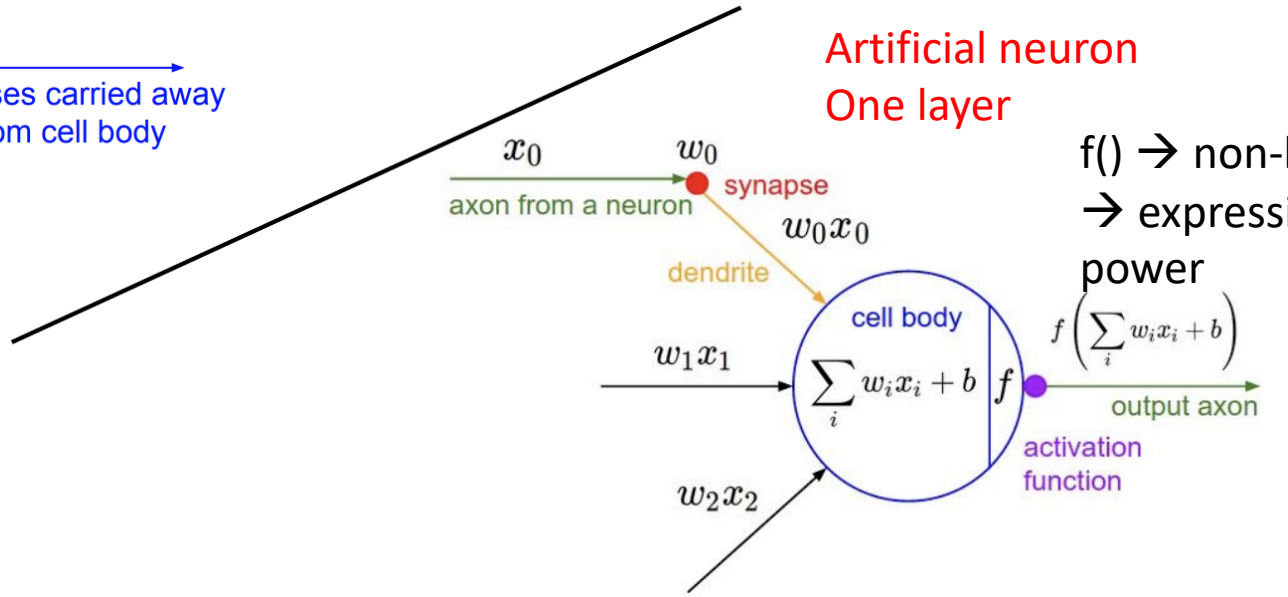
Deep Learning

Biological neuron
One neuron



Artificial neuron
One layer

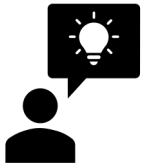
f() → non-linear
→ expressive power



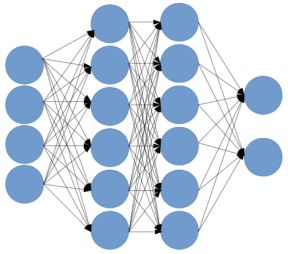
Deep Learning

Machine Learning

Input



Get features

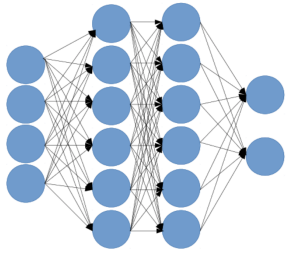


Classification

Output

Deep Learning

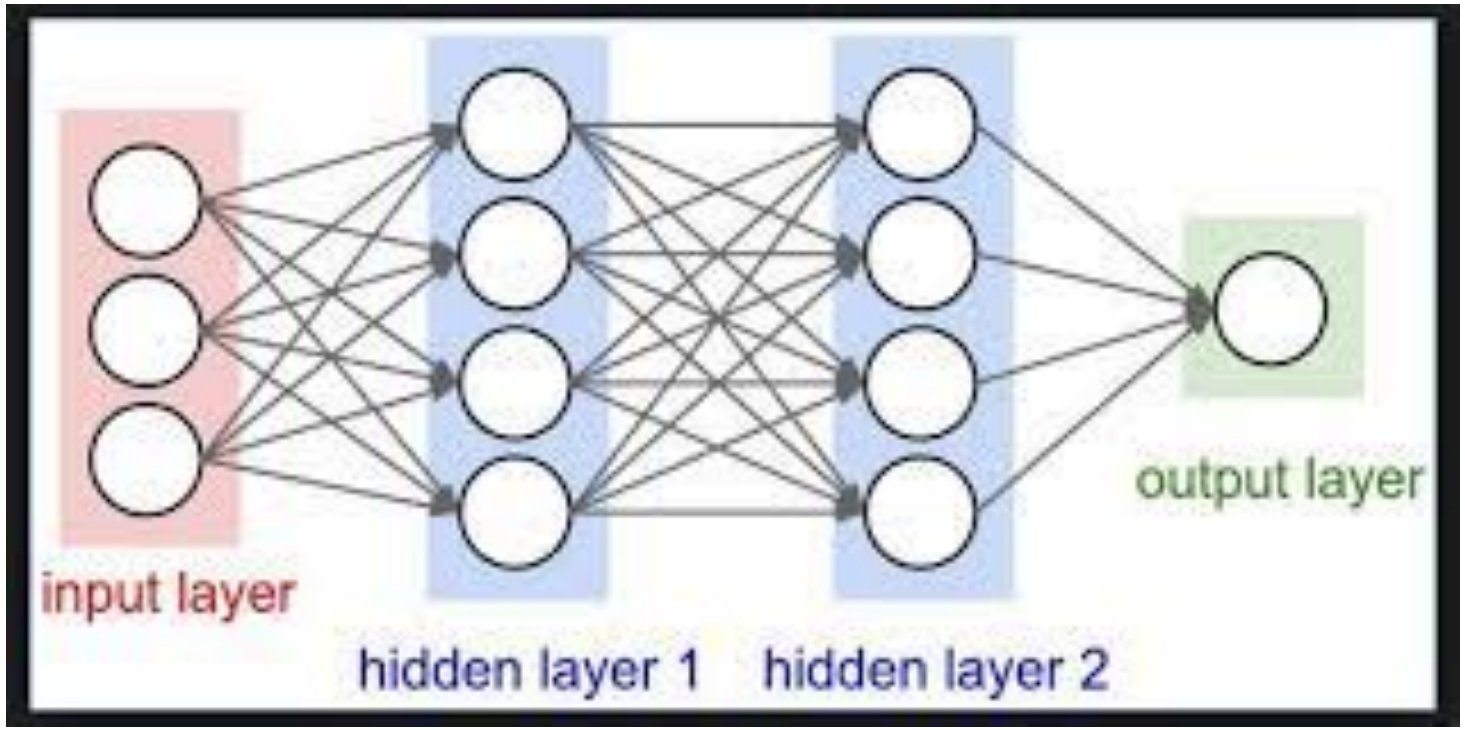
Input



Get features + Classification

Output

Layer



$$M(x,\theta) = W^{③T}\left(f\left(W^{②T}\left(f\left(W^{①T}x + b^{①}\right)\right) + b^{②}\right)\right) + b^{③}$$

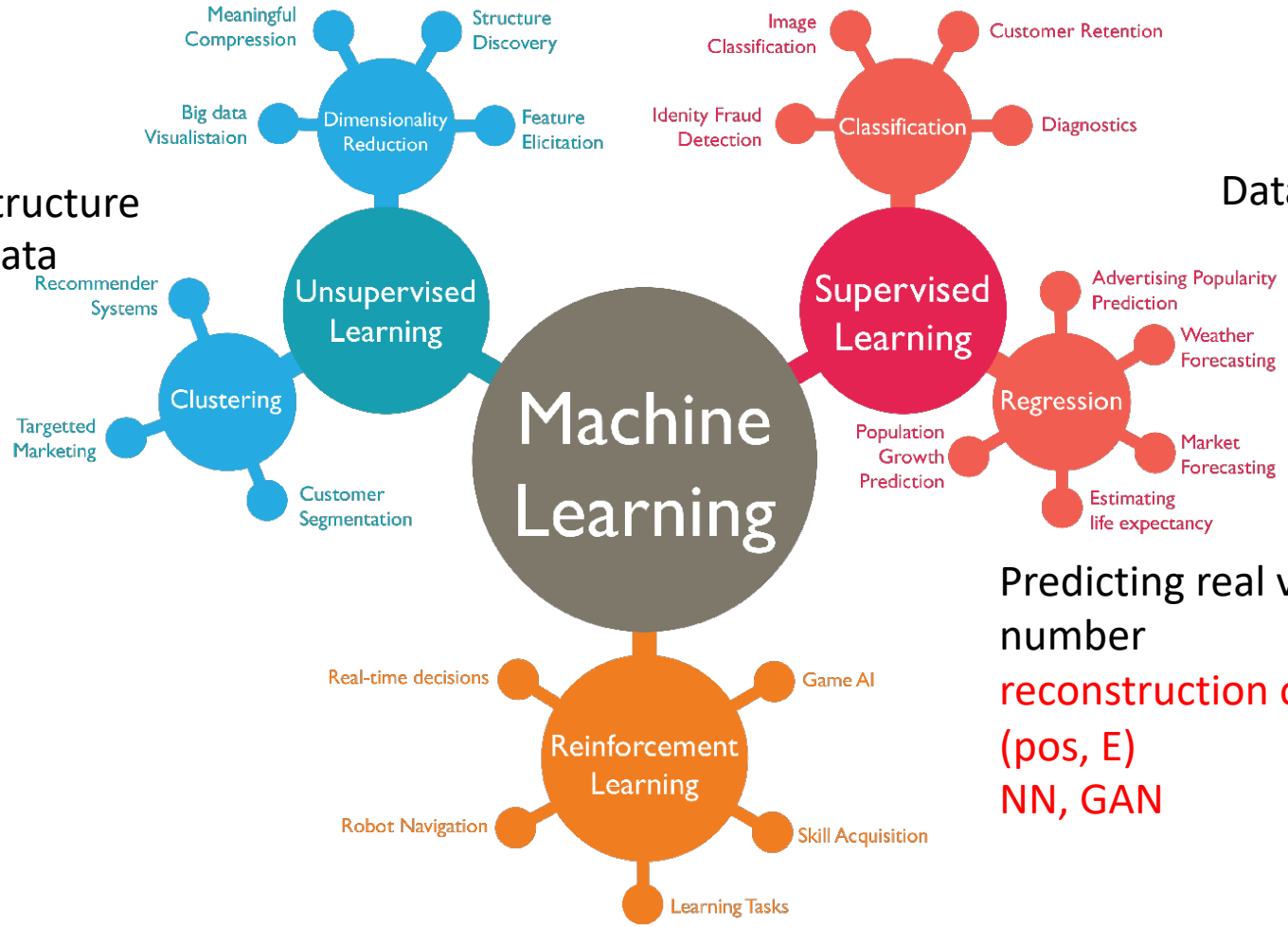
More hidden layers → more chaining of non-linearity → increase the complexity of the model

Branches

Predicting categorical things
signal/bck
BDT, SVM, NN

Discover hidden structure in the unlabeled data

Data is labeled

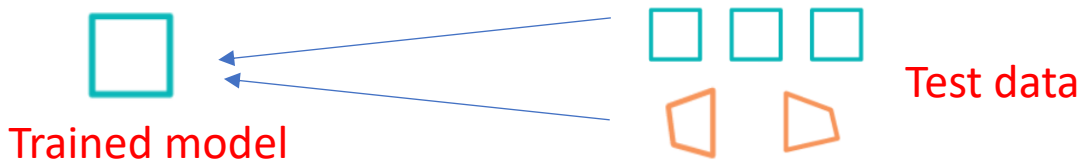


Predicting real value number
reconstruction of (pos, E)
NN, GAN

Learning actions from the environment

Initial Questions (data quality)

- **What are we trying to model?**
  - MDC3 simulation data (normal distribution data with label), 5 main populations? Each plot?
  - Possibly real data later (normal + anomalous data without labels)
- **What are we trying to classify?**
  - Normality vs. anomalies:
    - 1. Shift of the current population
    - 2. Disappearance of the current population
    - 3. Appearance of the new population
- **How are we going to classify anomalies?**
  - 1. Compare the test data against the trained normal set



Fixed learning

  - 2. Compare the data with with time propagation

Online learning

Later Questions (data quality)

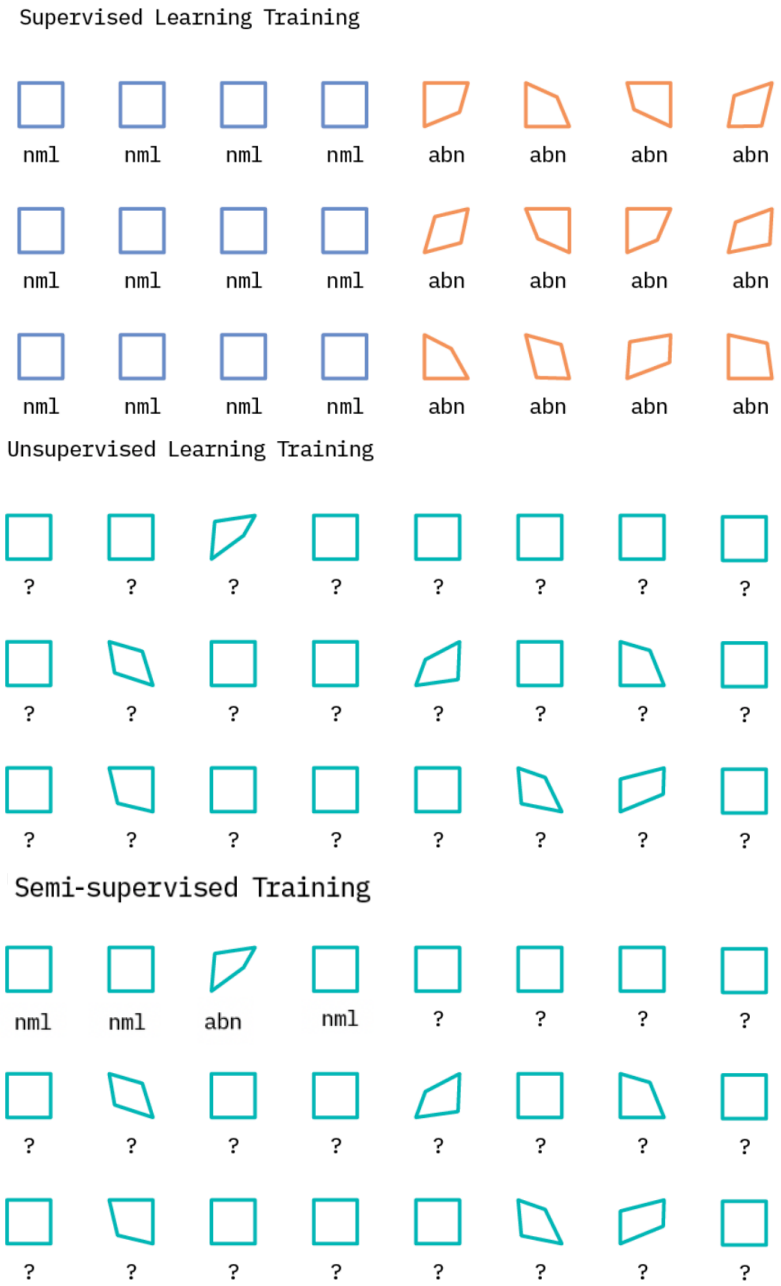
- **How to evaluate that the model is doing a good job?**
  - Choose the training epoch → optimal fit
  - Computation time: quick check
  - A certain tolerance on some anomalies (i.e. noise…)
  - The determination of both abnormality of current model (shift & disappearance) & novel model (appearance)
- **How do we know if this model will be productive on the data that we've never seen before?**
  - Tests on real data and see if the output matches our expectations

Anomaly Detection Deep Learning

- To model normal behavior first, and then exploit this knowledge to identify anomalies.
  - An anomaly score is assigned to each data point → measurement of the deviation from the normal behavior
  - Determine the threshold: scores above a given threshold are tagged as anomalies; below this threshold are tagged as normal
- Three branches (supervised, unsupervised, semi-supervised)
- Five models (AutoEncoder; Variational AutoEncoder; GAN; Sequence-to-Sequence Model; One Class SVM)
- Data Quality & Physics Analysis (signal/background)

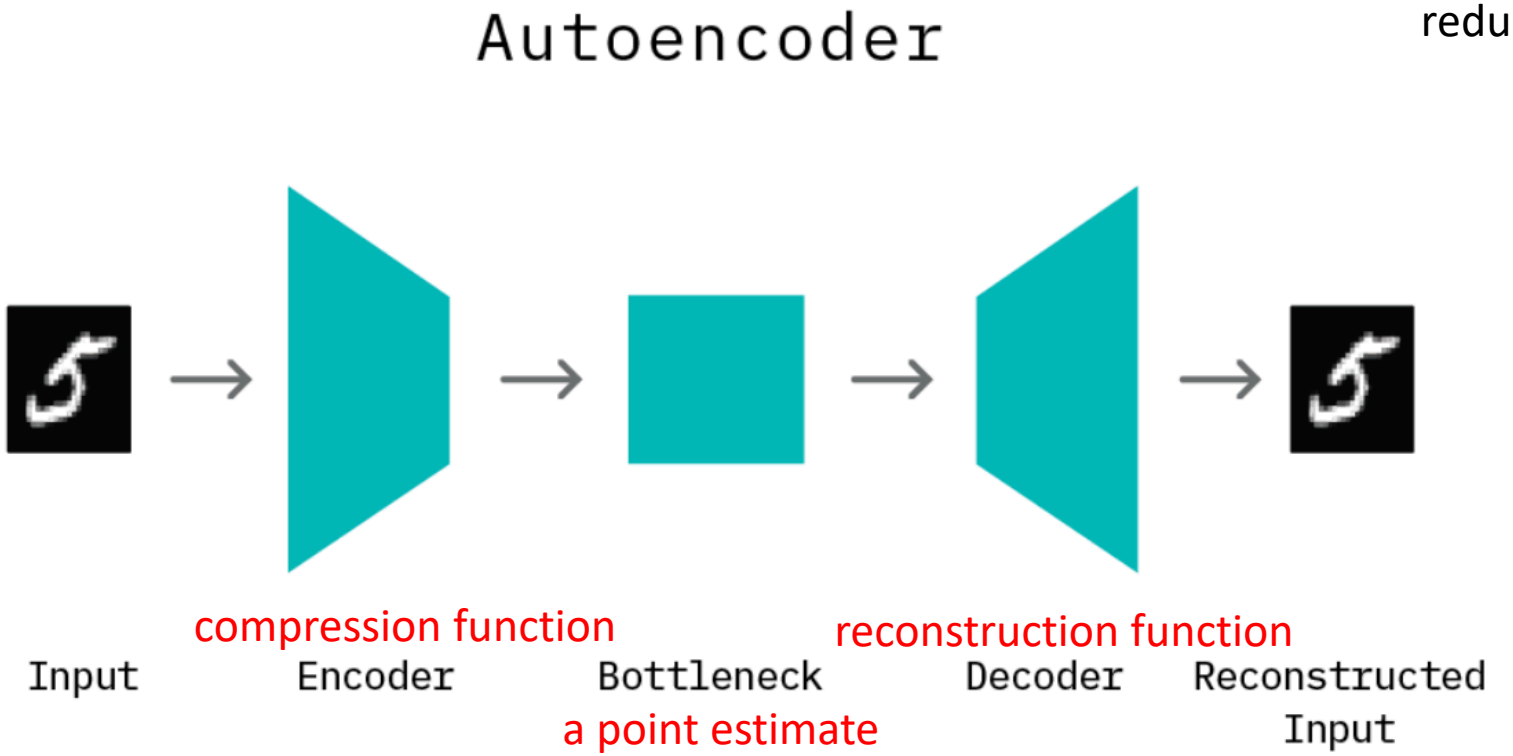
Branches

- Supervised: labels are available for both normal & anomalous sets
  - Both normal and anomalous datasets are labeled
  - Don’t have labeled anomalous data
- Unsupervised: no labels for the training set
  - Both normal and anomalous datasets are not labeled
  - Learn by finding structure within the input features
- Semi-supervised: labels are only available for the normal set, but not the anomalous set
  - With the assumption: most data points within an unlabeled dataset are normal
  - Large amounts of unlabeled data + small amounts of labeled data
  - Detection of both known and previously unseen anomalies

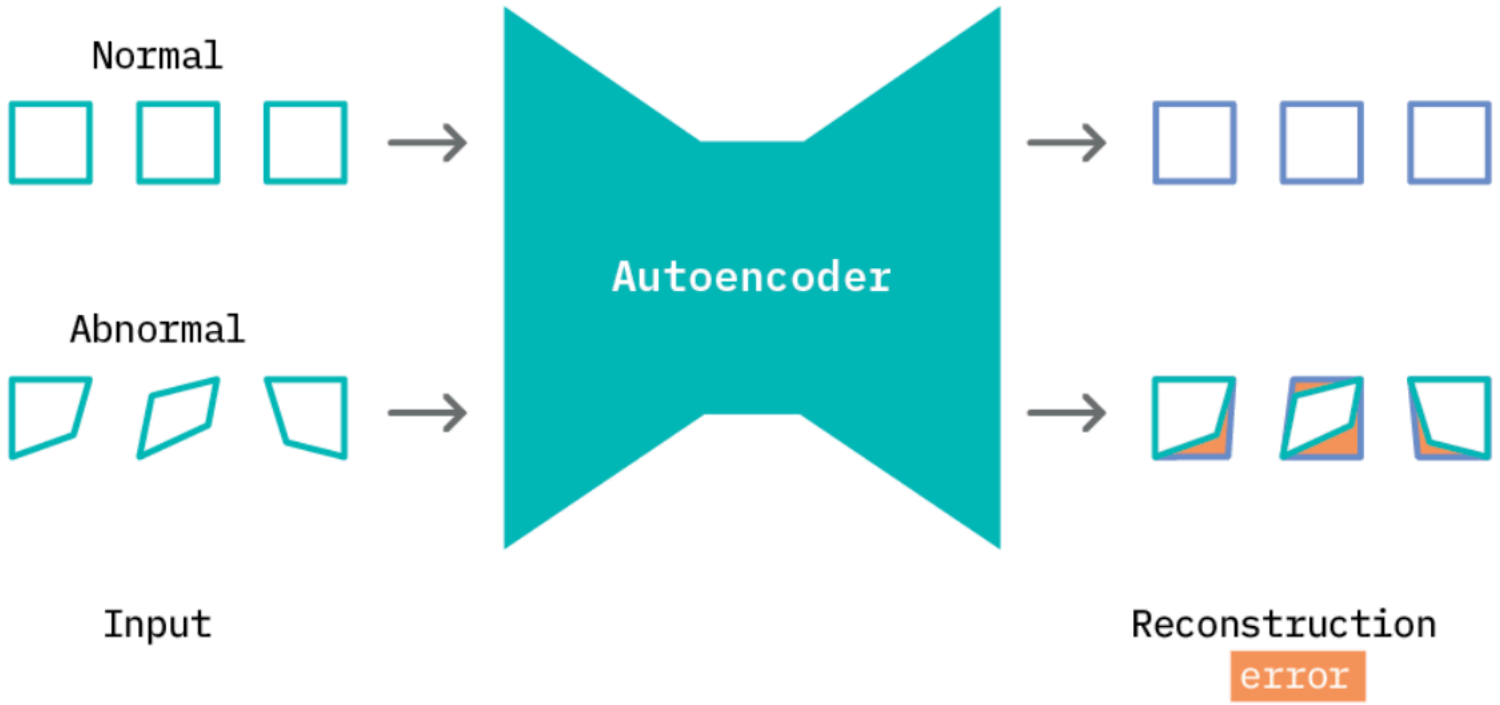


Autoencoders

- The model is trained to minimize the reconstruction error
- Dimensionality reduction technique



Autoencoder



To model normal behavior

- Semi-supervised approach
- Train the autoencoder on normal data samples (sim data)
- Model learns a mapping function to reconstruct normal data samples successfully

To identify anomalies

- Reconstruction error score → Anomaly score
- Flag error if above a certain threshold score