MACHINE LEARNING
(FOR PHYSICISTS)

Luc Le Pottier
University of Michigan
1/30/2020

(Partially based on materials from Eric Eaton)

What is ML?

"the study of algorithms that improve their performance *P* at task *T* with experience *E.*"

- A well-defined machine learning task is one for which *P, T,* and *E* are defined.
- Examples?
  - best method of navigating undersea

Why?

ML is useful when:

- We have large datasets (particle physics)
- We want to create things that can do things we can’t (robots)
- We want to customize models (recommendation systems)

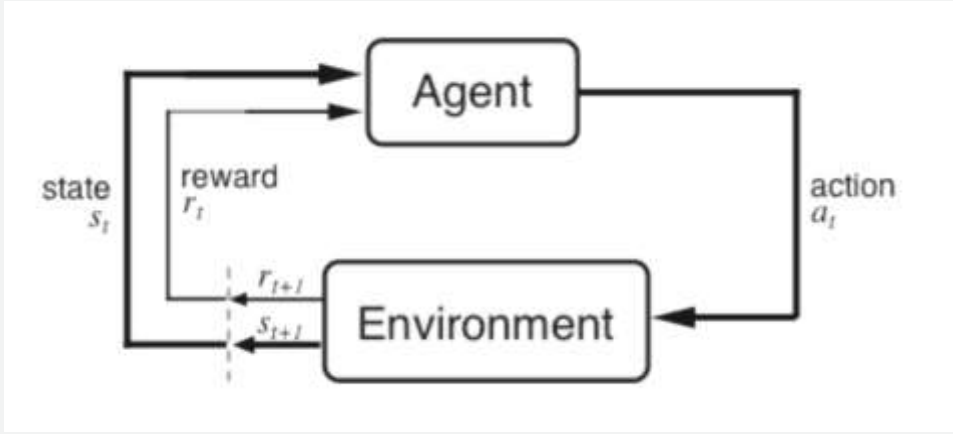
It is *not* always useful (but this is frequently disregarded)

Types of Learning

- Supervised Learning
  - Given data samples *X* and their labels *Y*
  - **Regression:** learn a function *f(x)* to predict *y* given *x*
  - **Classification:** regression, but *y* is categorical
- Unsupervised Learning
  - Given data samples *X* only
  - Want to determine hidden structure in some way

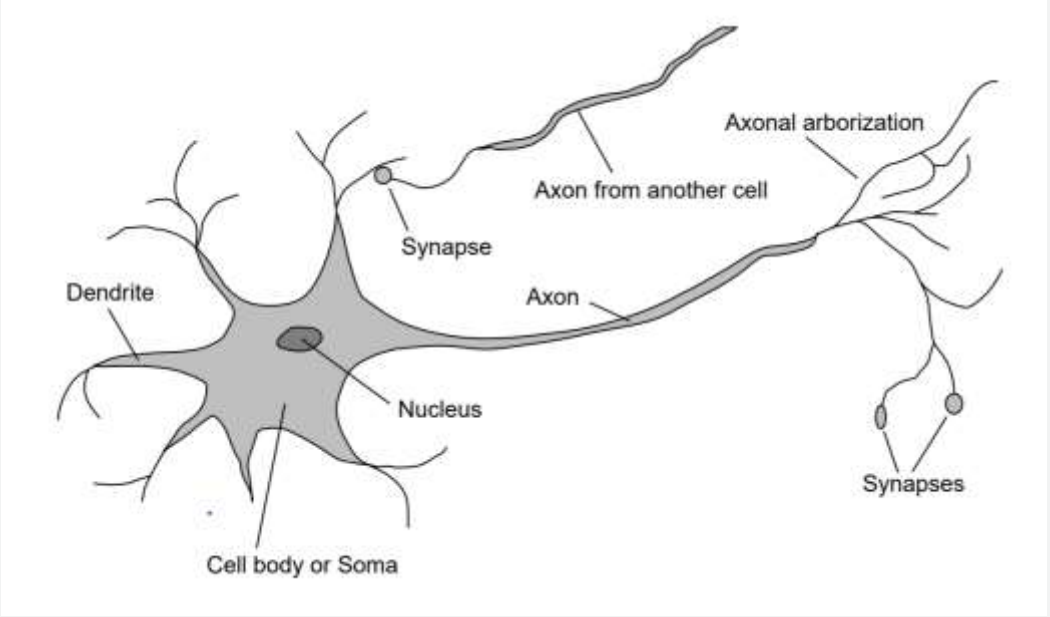
Types of Learning

- Reinforcement Learning
  - Given a sequence of states + actions with delayed rewards
  - Determine a mapping from **states** to **actions** which describes the correct action given a state
  - Playing video games, etc.



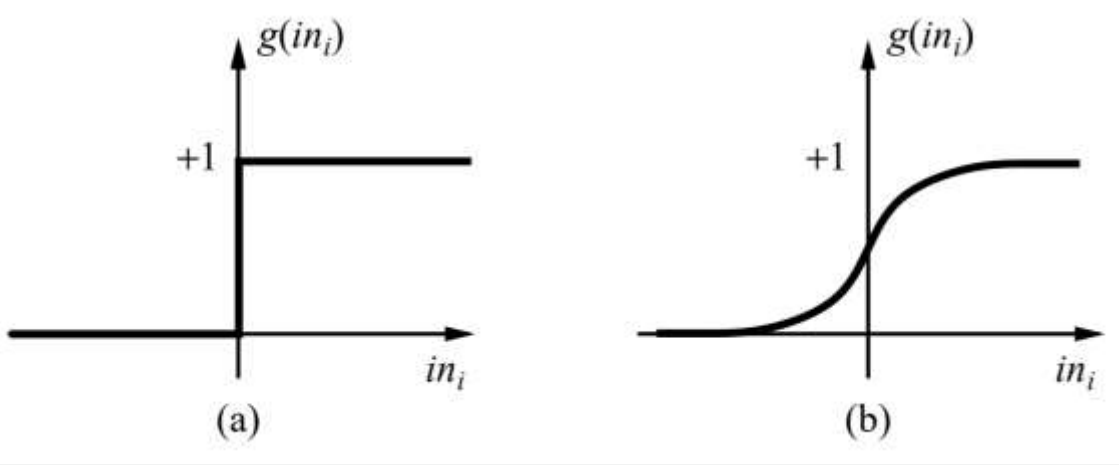
Neural Networks

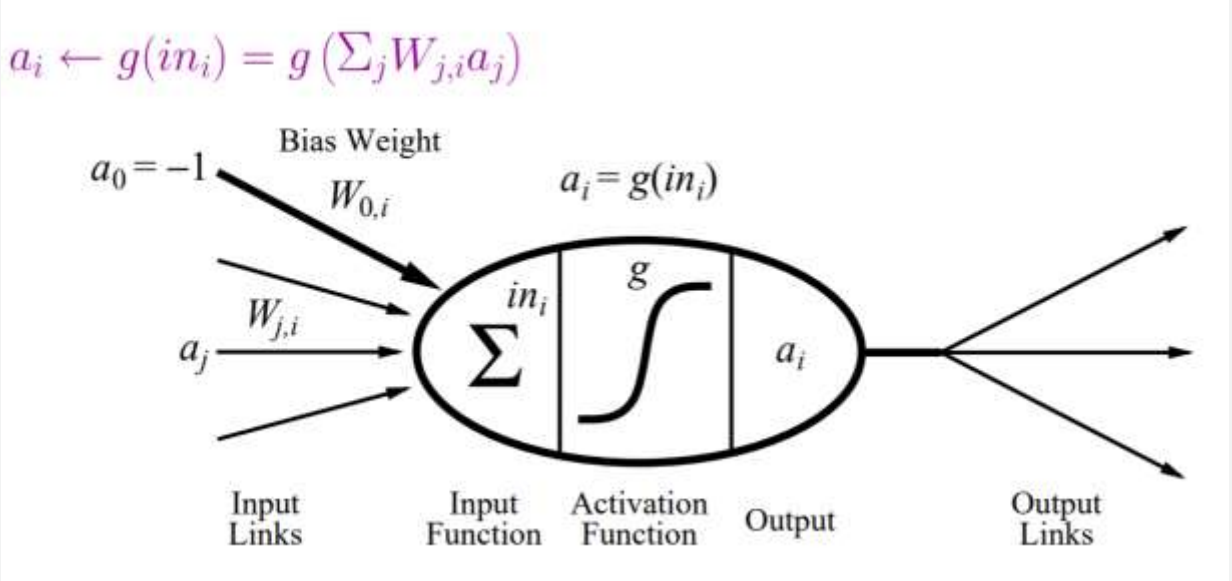
- Will focus mostly on algorithms which use combinations of neural networks
- Basic component of many important algorithms



Neural Networks: Neurons

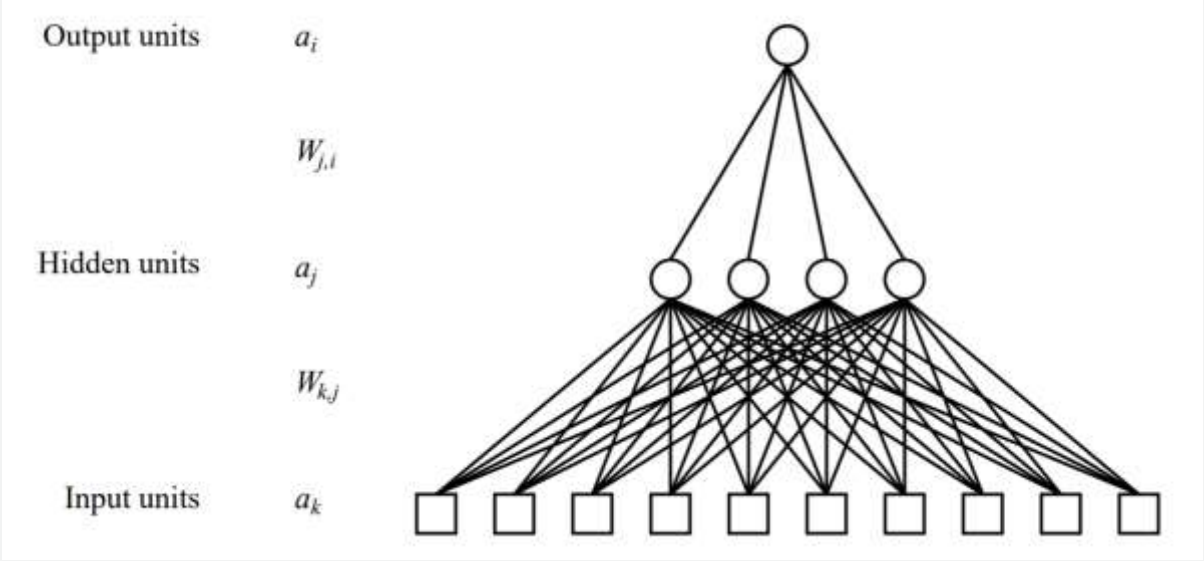
- Output a squashed linear function of the inputs
- Activations can be step functions, inverse tangent functions, sigmoid functions, etc.





Neural Networks: Architecture

- Layers are fully connected, with # of hidden units and their layers chosen by hand
- Can have any number of input/output units

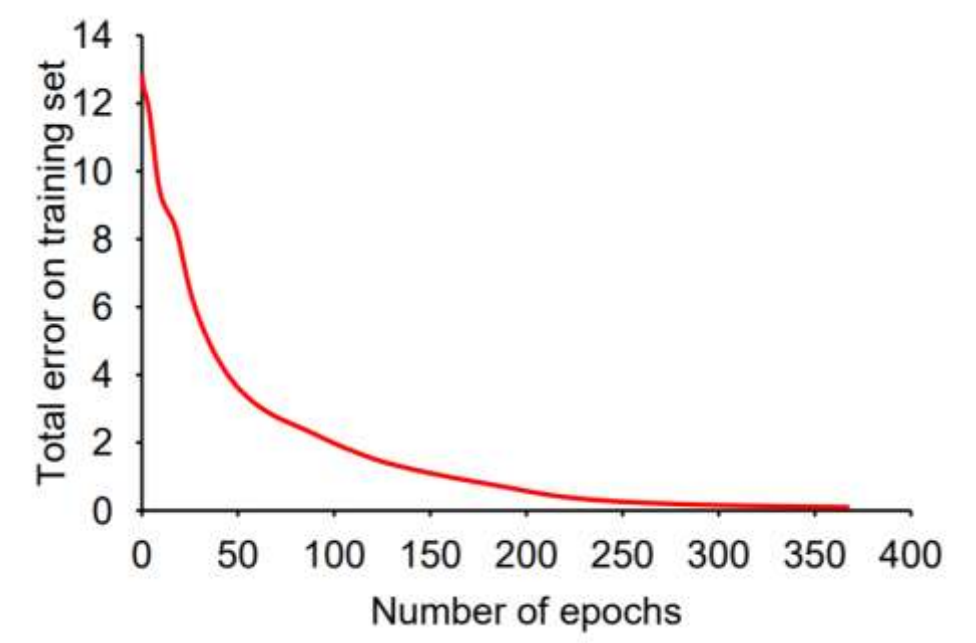


Neural Networks: Training

- From a theoretical perspective
  - NNs are usually trained by *backpropagating* (working back through the network and updating weight matricies *W*)
  - Gradients are automatically estimated by tensor libraries (tensorflow, pytorch, etc.)
  - At each *epoch* (instance of training on a batch of samples), the summed gradient updates for all examples

Neural Networks: Training

- Networks usually *converge* (stop learning) after many epochs (can vary a lot based on the model)
- Getting good results can be very very difficult
  - Often depends on the data!
  - Algorithms are fickle



Neural Networks: Training

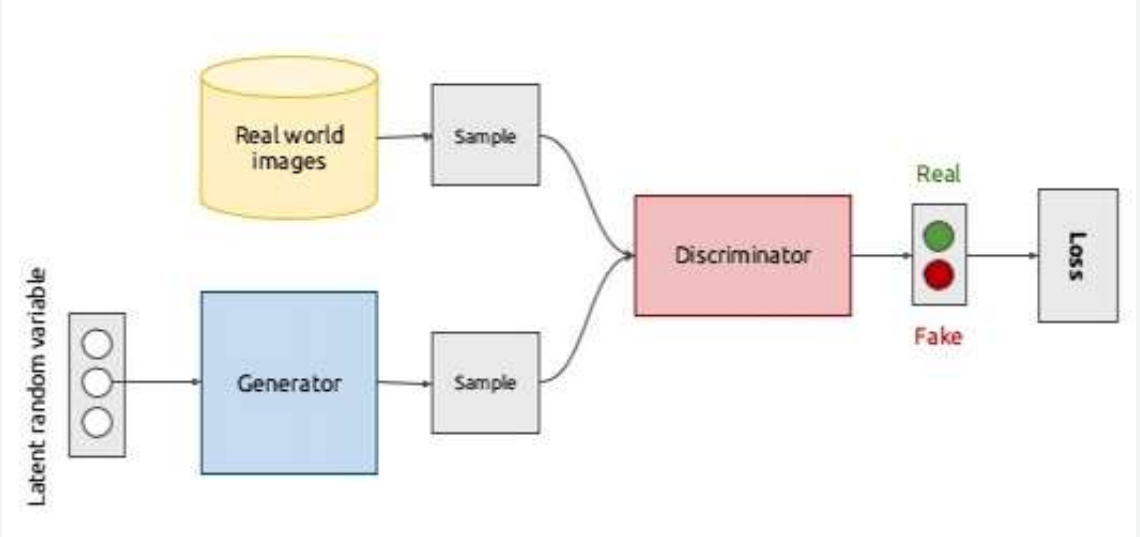
- Good resource for getting things to work (from Tesla AI director)
- Recipe (from karpathy) :
  1. Spend LOTS of time visualizing the data
  2. Set up end-to-end training/eval skeleton
  3. Get a large enough model to overfit (be simple!)
  4. Regularize the model (prevent overfitting)
  5. Tune the model (hyperparameters!!)
  6. “Squeeze out the juice”

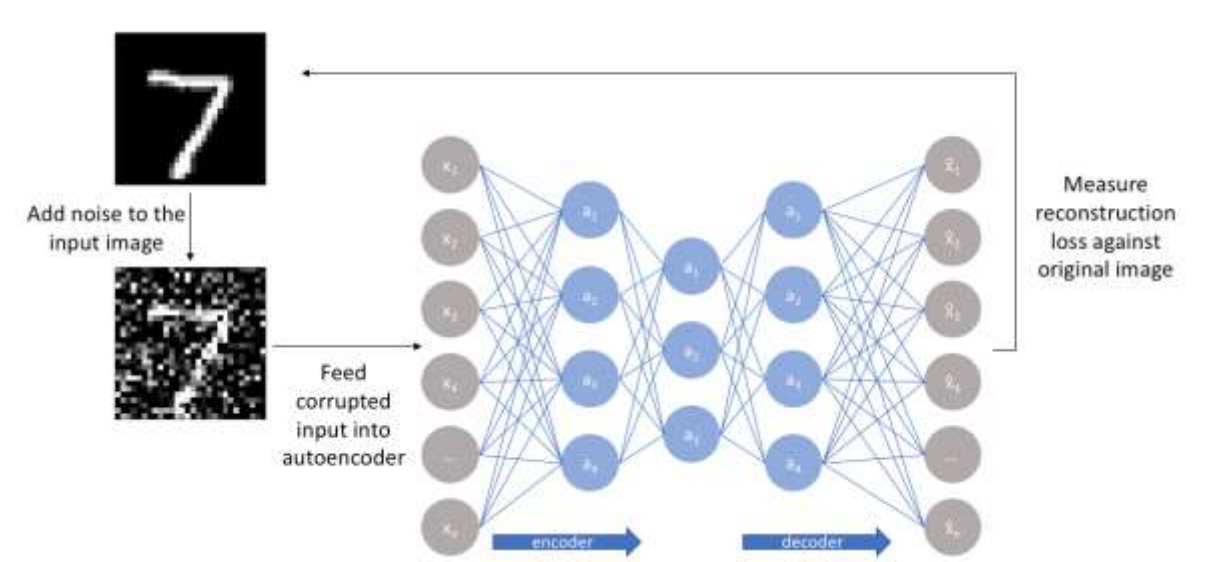
Hyperparameters

- Variable network parameters which change how well everything works
- NEED to optimize on these in order to be successful
- Examples:
  - Optimizer (thing which does SGD) and learning rate of optimizer
  - Regularization
  - # of nodes in each layer, # of layers, etc
  - Early stopping
  - Get more data!

Types of Models

- Deep networks
  - Very common, just means ‘lots of layers’
- Generative Adversarial Networks (GAN)
  - Network that generates fake data competes with a network that discriminates fake vs. real
  - Gans work for both people and cats
- Autoencoder
  - Inputs match outputs
  - Learn some meaningful information about your data in an unsupervised fashion





Types of Models

- MANY more networks than we could possibly go through
- Let's build one!