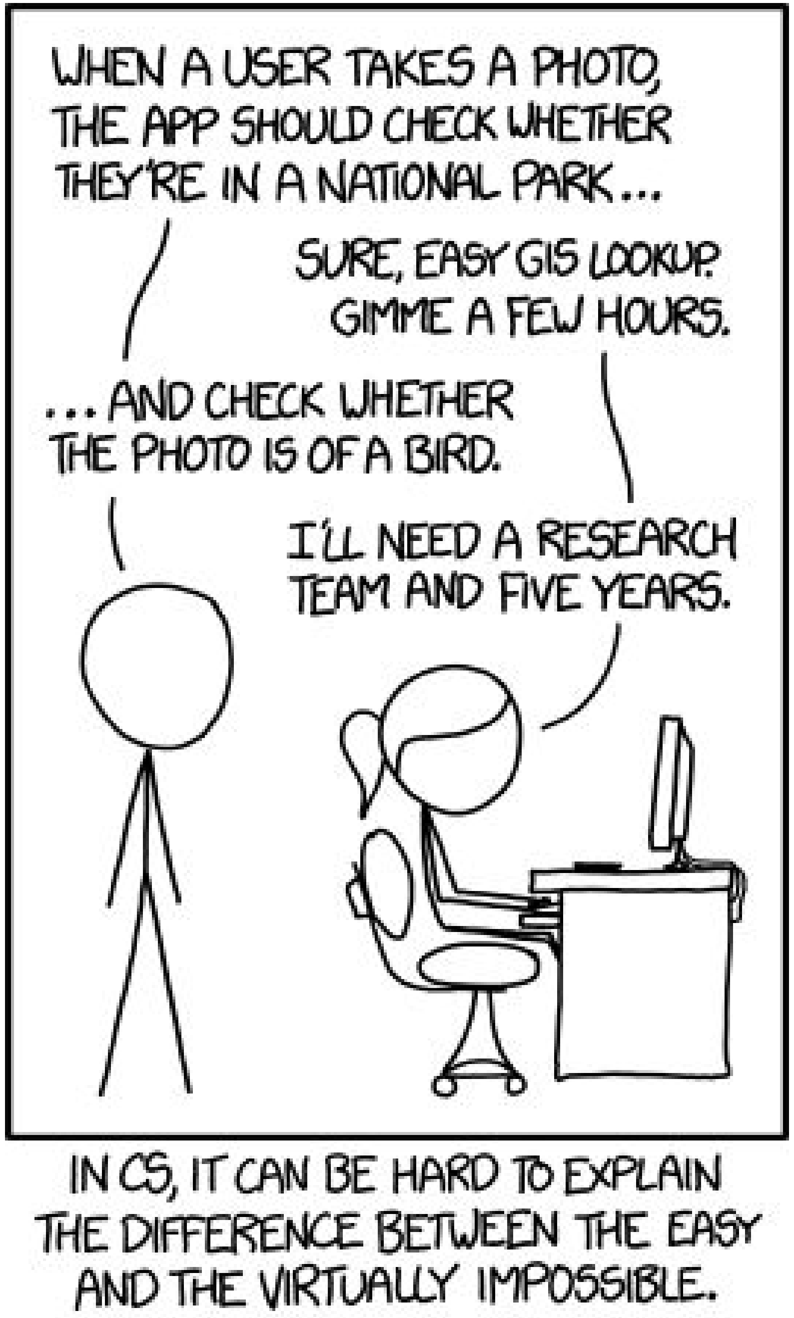
A Neural Networks Presentation

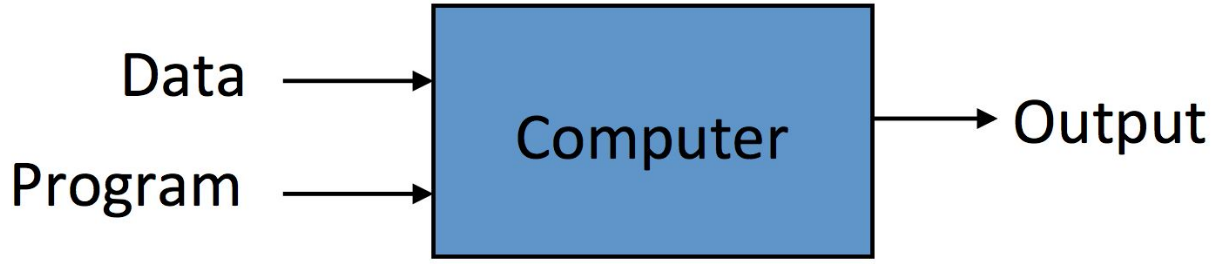
Lawrence Chan

A out-of-date joke

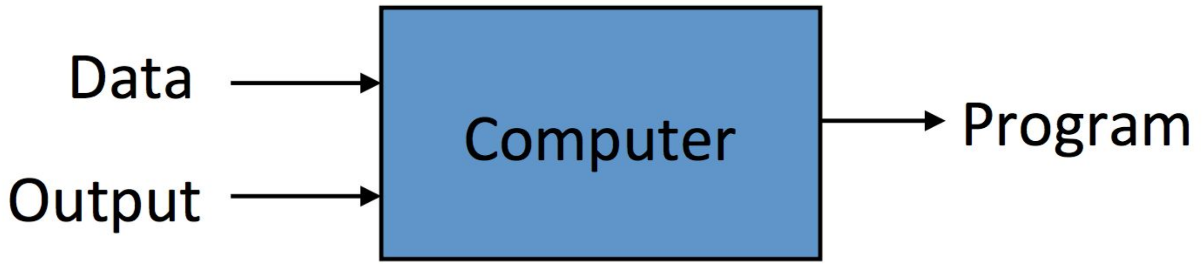


What is Machine Learning?

Traditional Programming



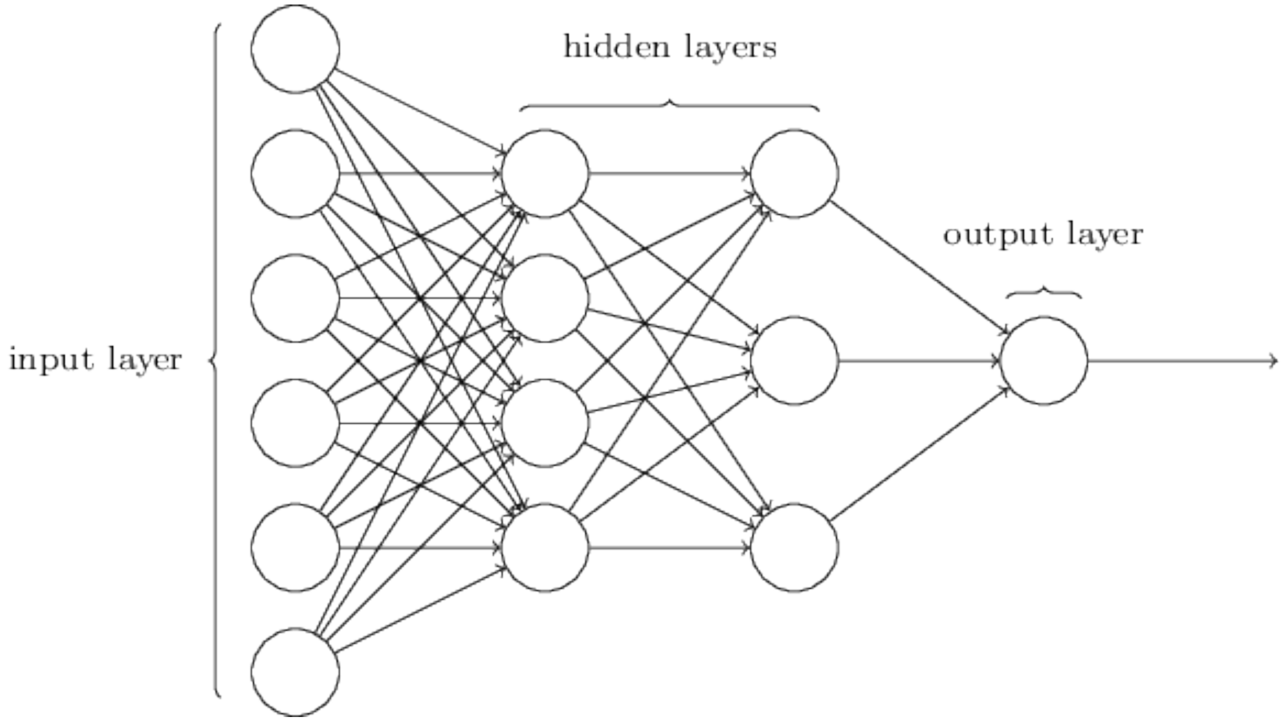
Machine Learning



Slide Credit: Pedro Domingos

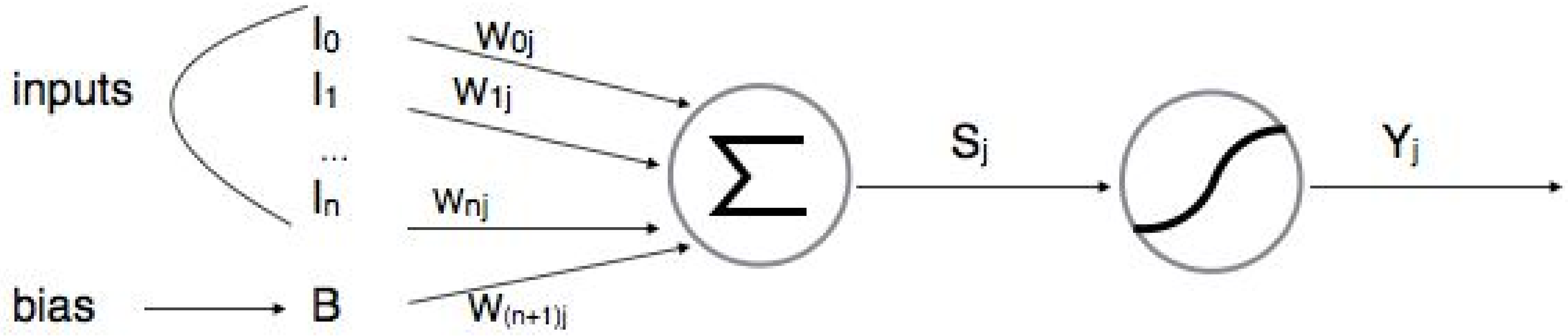
Deep Neural Networks

- Complicated function made of simple parts
  - "Universal approximator property"
- A lot of recent success in ML/AI has been due to better architectures for and increased ability to train neural nets
- Very messy; not a lot of good theory behind them.



Picture Credit: Michael Nielson

A “Neuron”

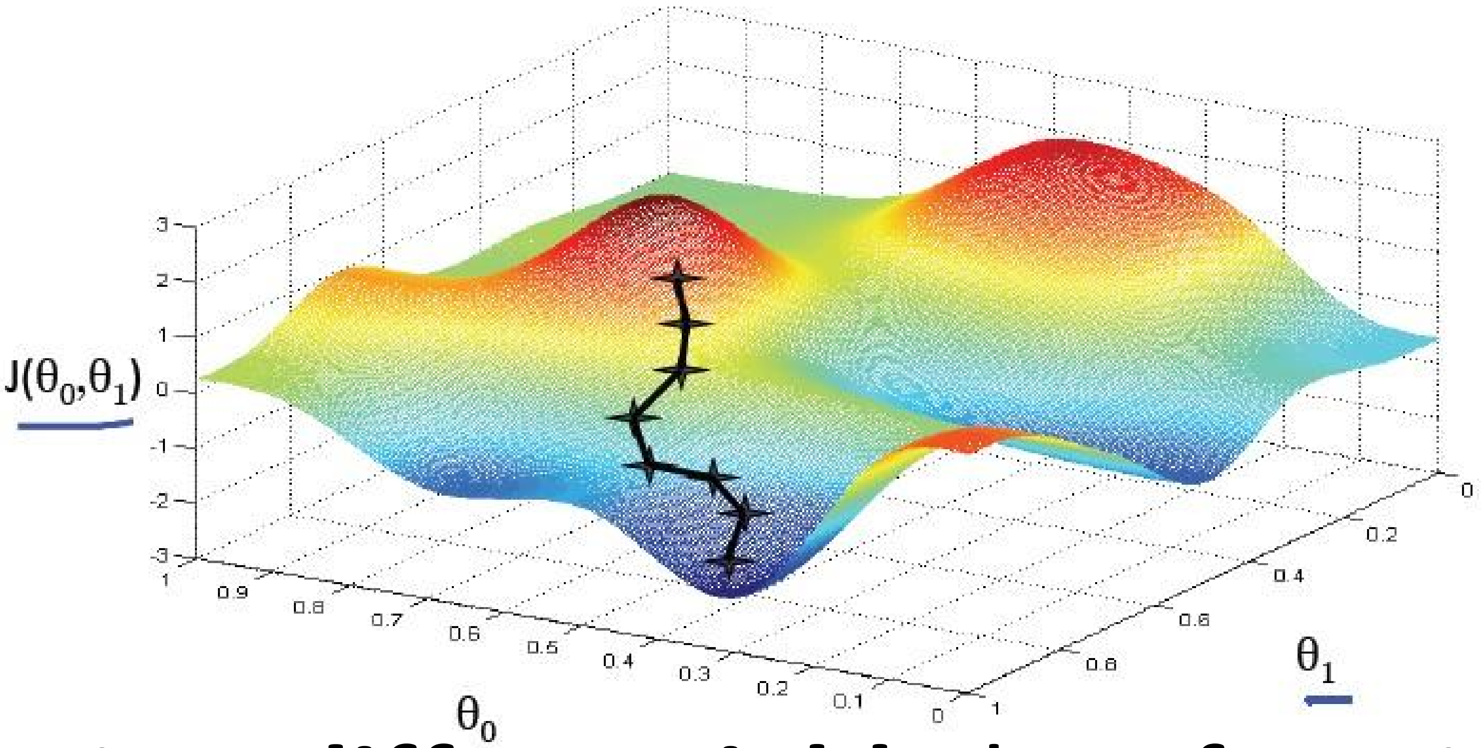


Train a Network by Minimizing a Loss Function

- Loss functions measure how well the classifier is doing
- For example:
  - 0-1 Loss
  - Word-Edit Distance
  - Mean-squared error
  - Cross-Entropy Loss
- Warning: loss functions not guaranteed to capture what you want!

Some math on the board.

Modern Deep Learning - “Programming by gradient descent”



Requires **differentiable** loss function!

Picture Credit: Andrew Ng

Backpropagation - Efficiently computing gradients

- Chain rule!

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$$

More math on board

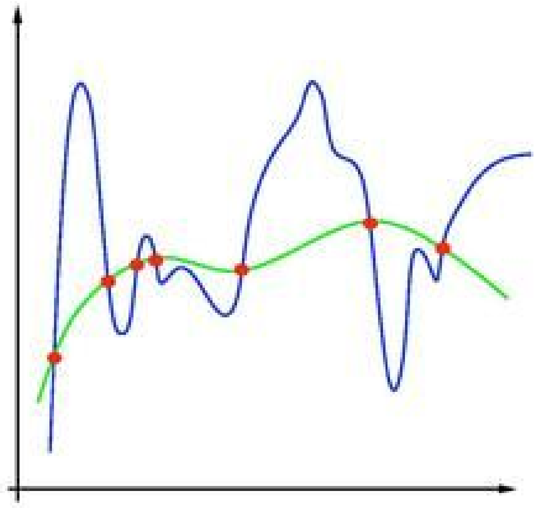
Thankfully, you don't actually need any of that to do (simple) deep learning!

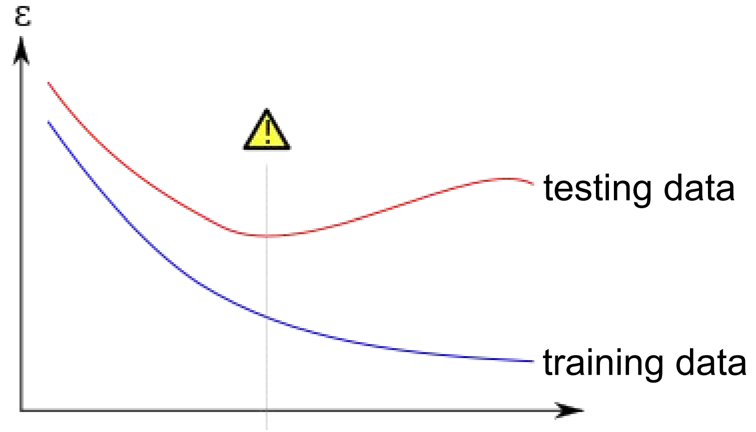
- High level interfaces (handle almost everything):
  - Sklearn
  - Keras
- Lower-level interfaces (still handle backprop and gradient updates)
  - Tensorflow
  - PyTorch

We’re just going to use Scikit-Learn.

One last warning: overfitting!

- Model may be fitting to noise instead of capturing “true” trend.
- Solution:
  - Use unseen data to measure how good your classifier is “really” performing.
  - Regularization
  - Visualizing Decision Boundary (aka, manual inspection)





A Pause for Questions