Demystifying Deep Learning

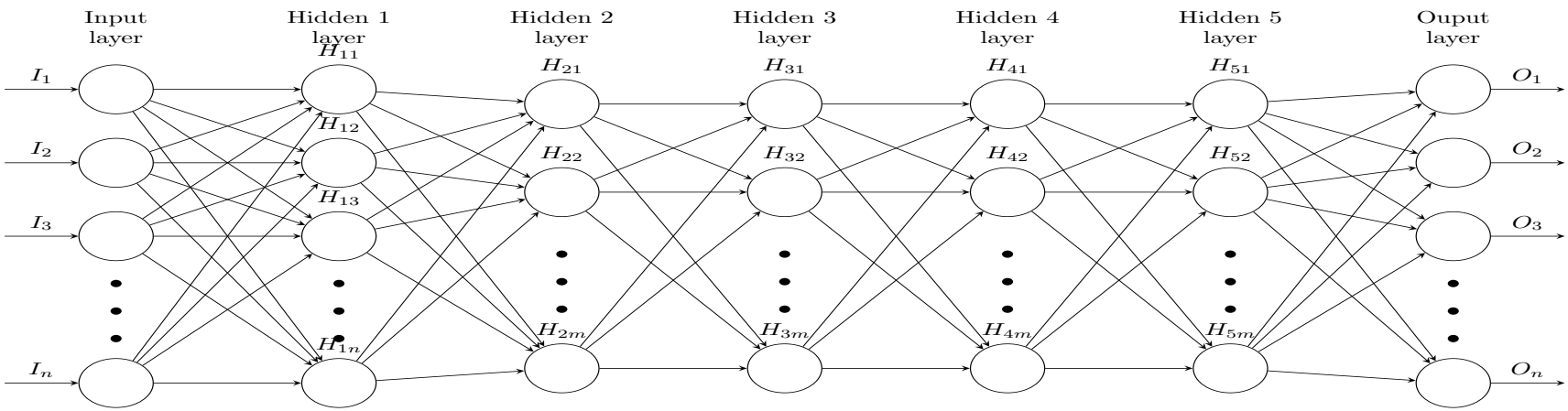
Guest Lecturer: Vinh Nguyen
(DMD 7.14, vinh.nguyen@unimelb.edu.au)

What is Deep Learning?

A state-of-the-art machine learning technique that has consistently broken record in a range of difficult AI problems

- Computer vision: image recognition/understanding
- Natural language processing: speech recognition, machine translation

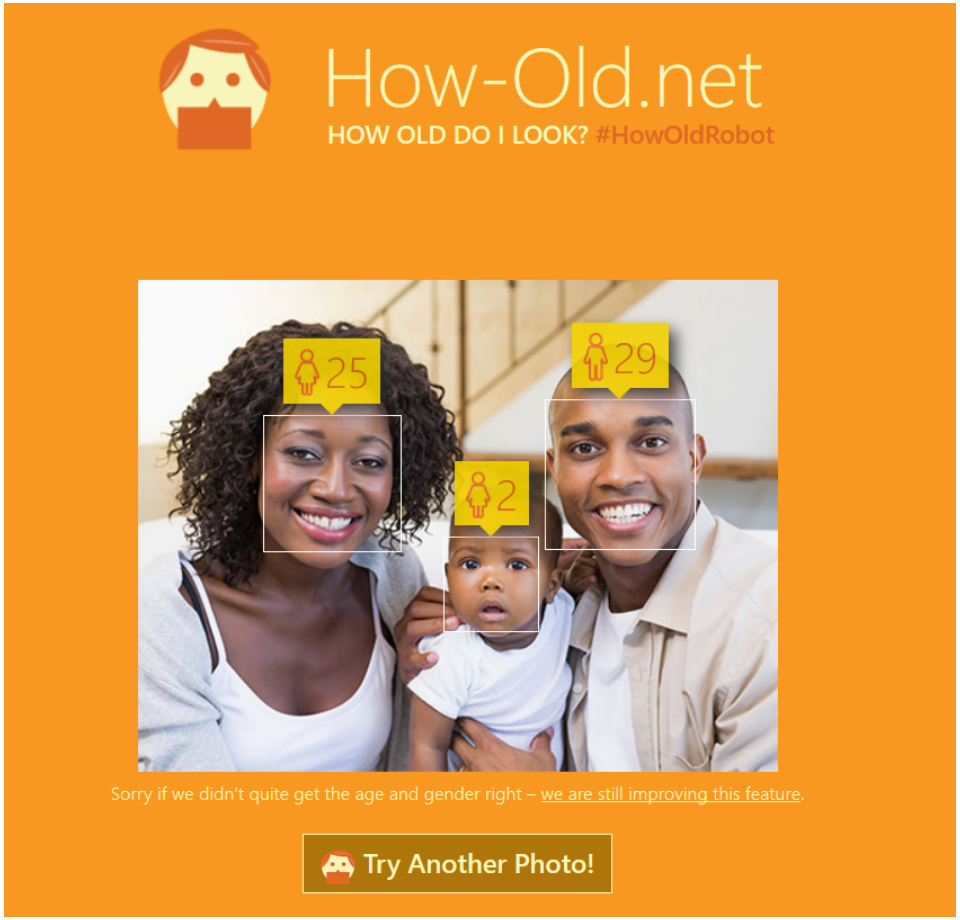
Often involves training neural networks comprising many hidden layers



What Deep Learning Can Do? Computer Vision

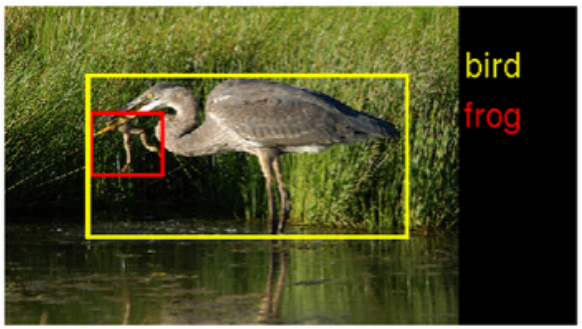
http://deeplearning.cs.toronto.edu/i2t

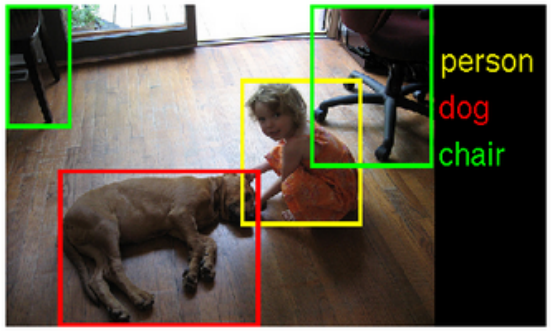
Predicting age and gender

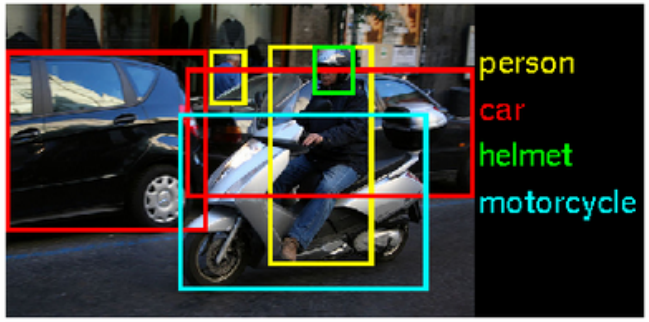


http://how-old.net/

Benchmark in Computer Vision
ImageNet contest

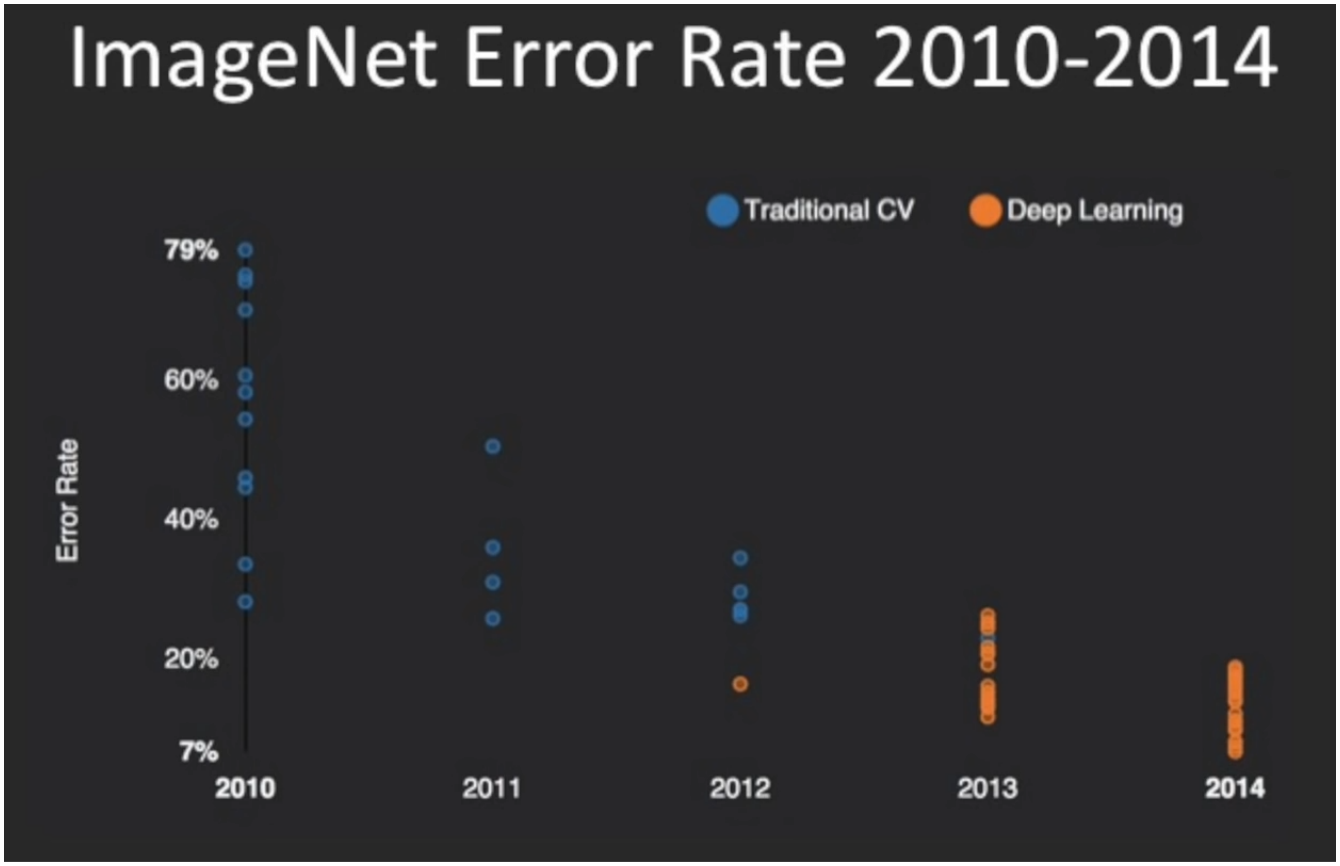






Source: http://image-net.org/challenges/LSVRC/2014/index

Benchmark in Computer Vision



Source: Clarifai

Applications of Deep Learning-Powered Computer Vision

Vision is an essential component of an AI system

- Robotics: self driving car, robots
- Image search & tagging (Facebook DeepFace)
- Virtual reality

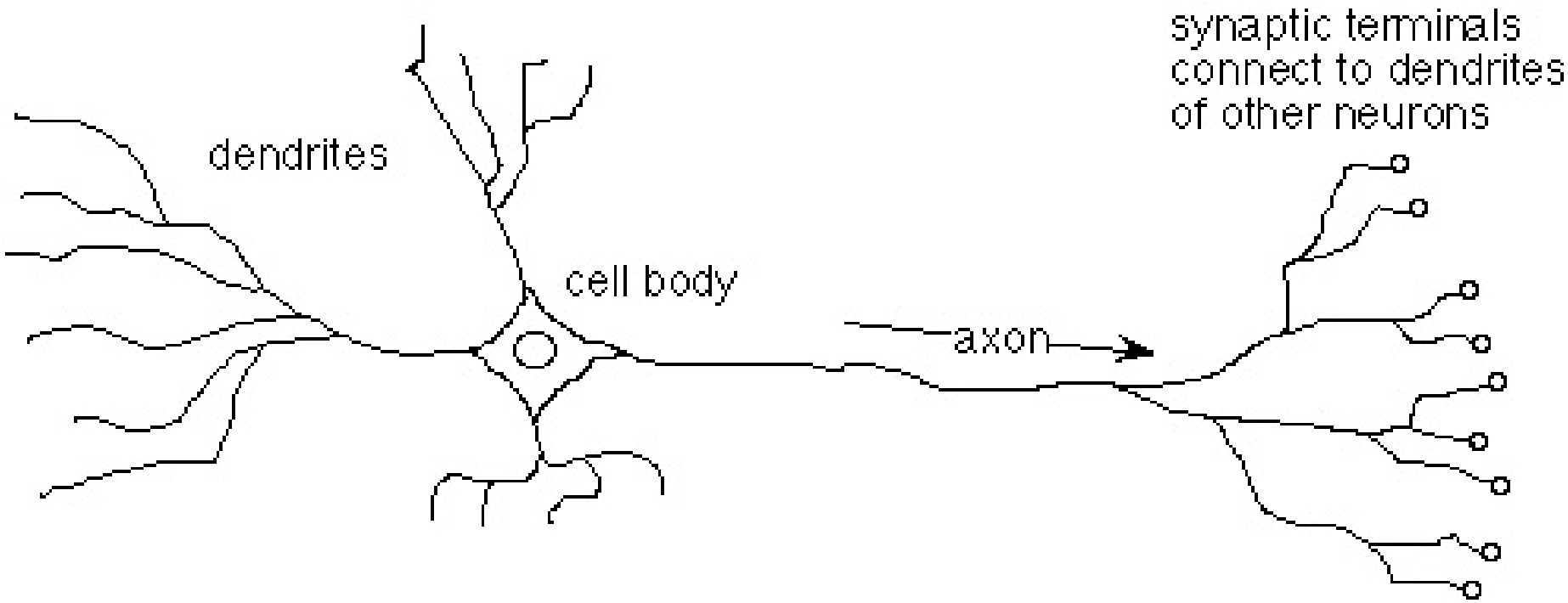
Other Domains

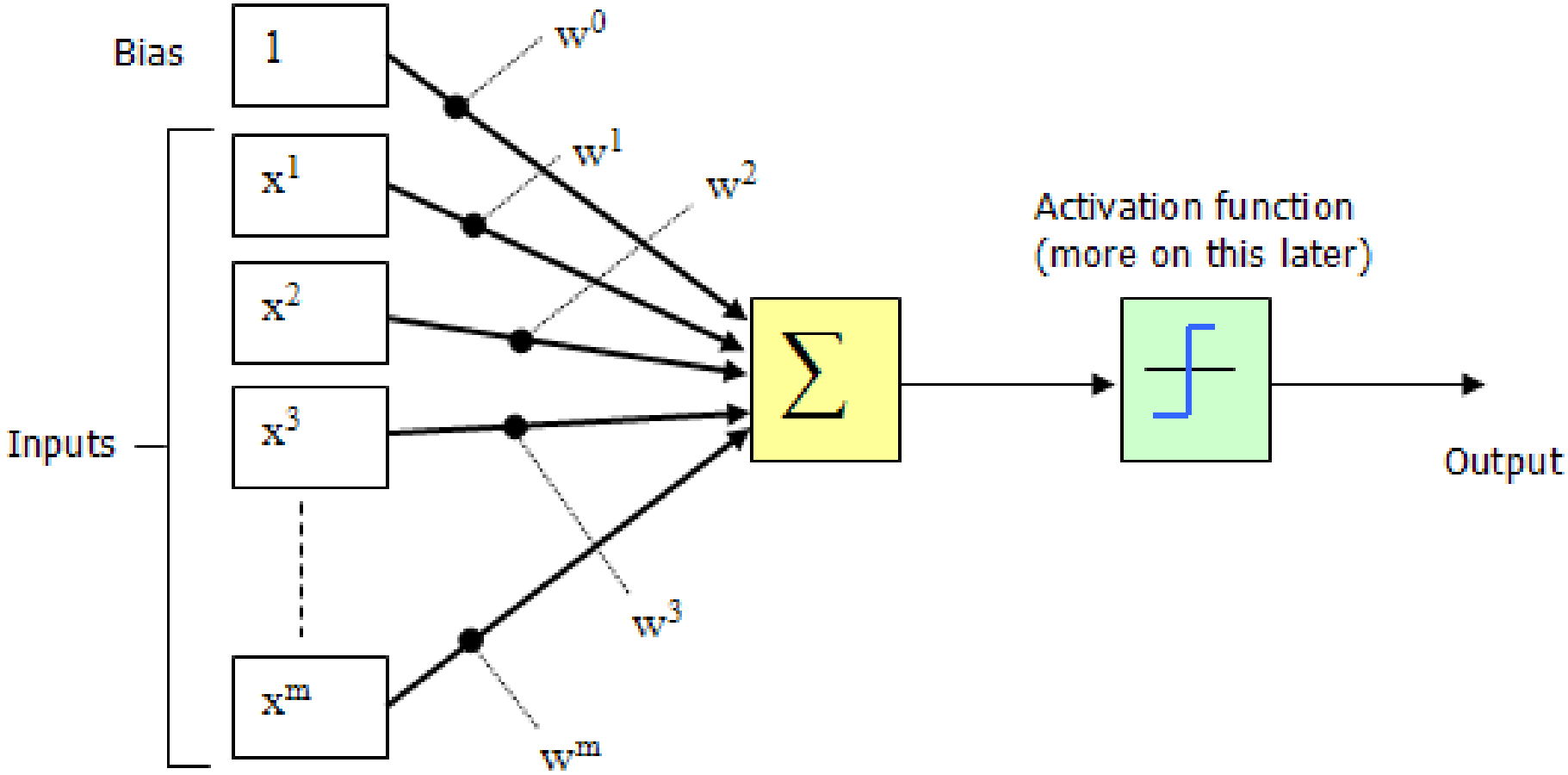
Deep learning is making breakthrough in

- Natural language processing: speech recognition, machine translation
  - ⋆ Dictation, Siri
  - ⋆ Live translation
    https://www.youtube.com/watch?v=cJIILew6l28

A Historical Perspective

- The first artificial neural networks were invented in 1950's
- Inspired by the biological brain





http://www.codeproject.com/Articles/16419/AI-Neural-Network-for-beginners-Part-of

What does an artificial neuron do?

- Receive signals from input neurons: $x_1, x_2, \ldots, x_m$
- Weight signals according to the link strength between neurons: $w_1x_1, w_2x_2, \ldots, w_mx_m$
- Add the input signals: $w_1x_1 + w_2x_2 + \ldots + w_mx_m = \sum_{i=1}^{m} w_ix_i$
- Emit an output signal: activation function $f_A(.)$

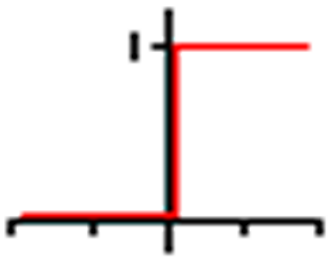
$$f_A\left(\sum_{i=1}^{m} w_ix_i\right)$$

Activation functions

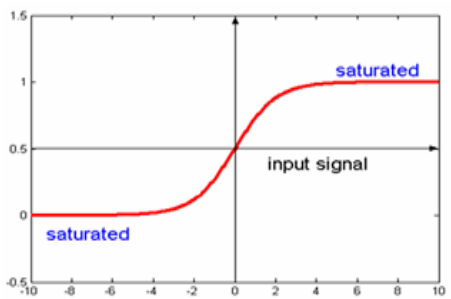
- Step function

$$f_{step}(x) = \begin{cases} 1 & \text{if } x \geq b; \\ 0 & \text{if } x < b. \end{cases}$$

- Sigmoid function: $f_{sigm}(x) = \frac{1}{1+e^{-\beta x}}$



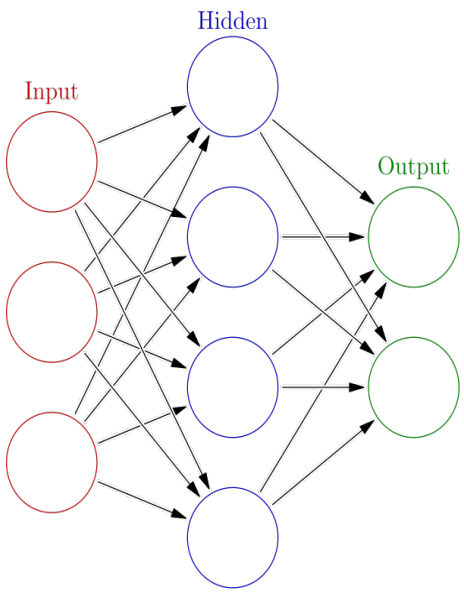
(a) Step



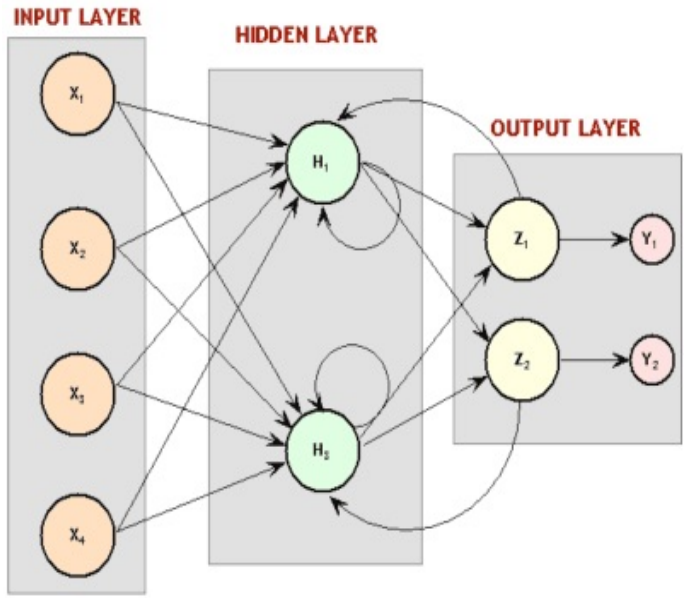
(b) Sigmoid

Network Architecture

- Feed-forward NN
- Recurrent NN



(c) Feed-forward



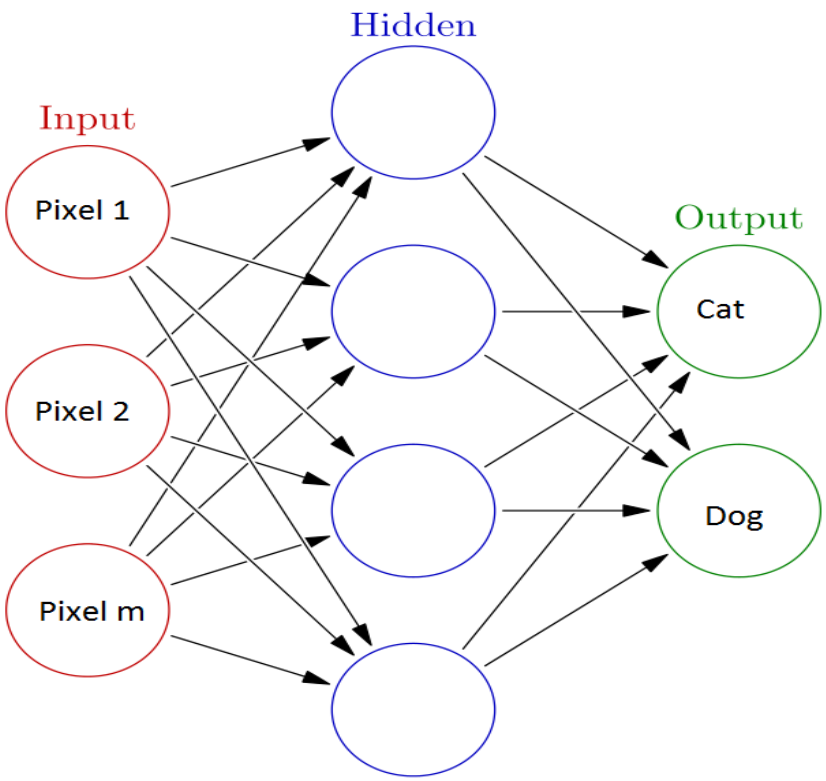
(d) Recurrent

Network Training

- Find a set of weights so that the the network exhibits the desired behaviour
- Example: cat vs. dog

https://www.kaggle.com/c/dogs-vs-cats/data

Training Data: input & label



Input: pixels, Output: {1-1} for cat & dog, {1-0} for cat only, {0-1} for dog only, {0-0} if no cat & dog

Error Function

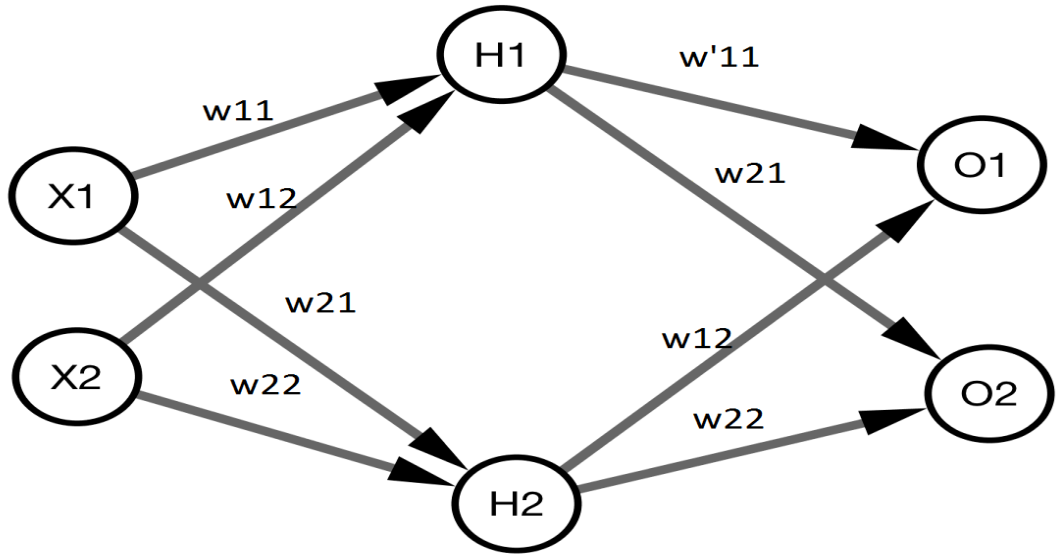
- Measure the difference between actual output and expected output
- One popular measure: sum of squared error

$$Error = \sum (output - label)^2$$

$$output = f_{NN}(input, weights)$$

- Note: Neural network is a composite/nested function that map the input to the output

Functional Form of Neural Networks



$$H_1 = f_A(X_1 w_{11} + X_2 w_{12}) \quad (1)$$

$$H_2 = f_A(X_1 w_{21} + X_2 w_{22}) \quad (2)$$

$$O_1 = H_1 w_{11} + H_2 w_{12} = f_A(X_1 w_{11} + X_2 w_{12}) w_{11} + f_A(X_1 w_{21} + X_2 w_{22}) w_{12} \quad (3)$$

$$O_2 = H_1 w_{21} + H_2 w_{22} = f_A(X_1 w_{11} + X_2 w_{12}) w_{21} + f_A(X_1 w_{21} + X_2 w_{22}) w_{22} \quad (4)$$

$$output = f_{NN}(input, weights) \quad (5)$$

Minimizing the Error Function

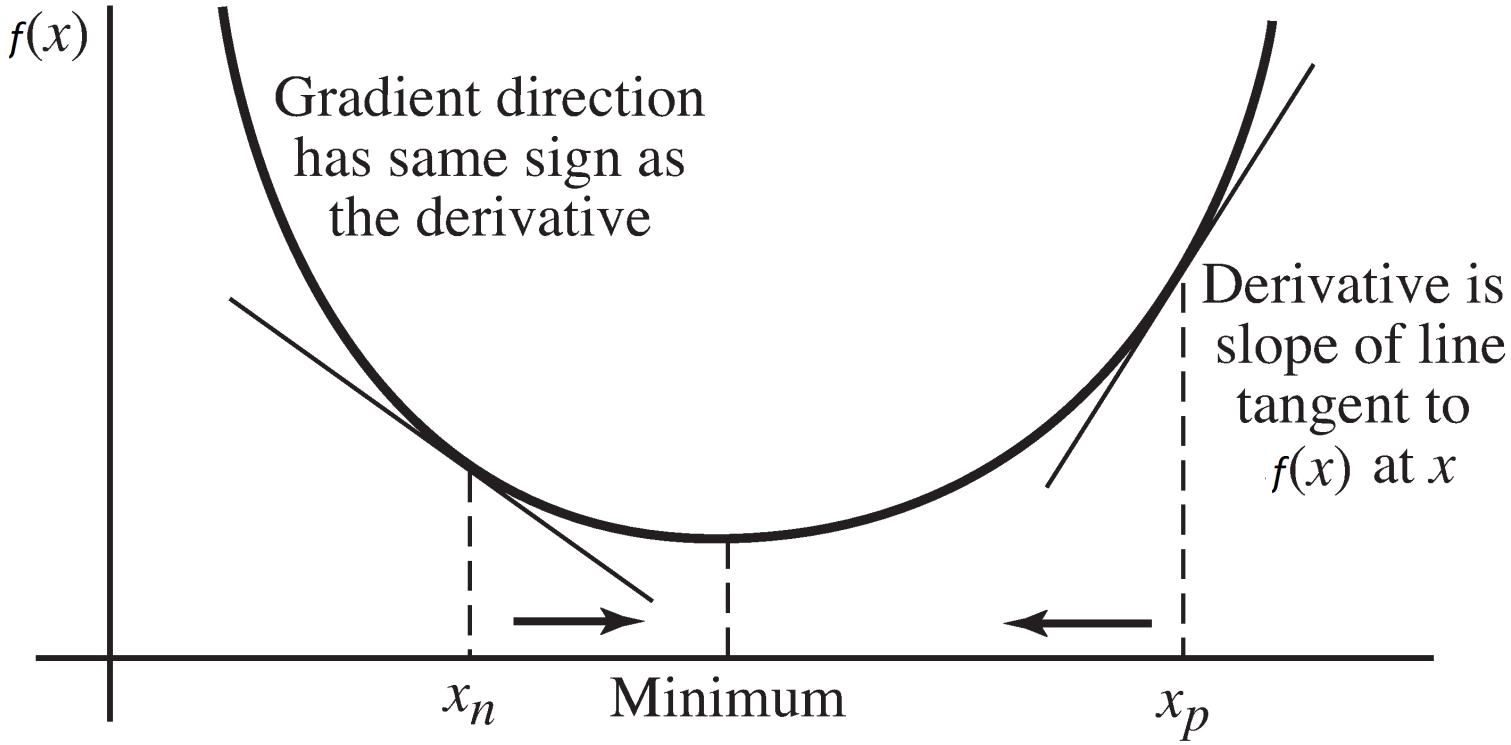
- Training Neural network is to minimize the error function with respects to the edge weights: $\min f(\mathbf{w})$
- Gradient descent: iterative algorithm for finding a local minimum of a function
  - ⋆ Compute the gradient

$$\frac{\partial f}{\partial \mathbf{w}} = \left(\frac{\partial f}{\partial w_{11}}, \frac{\partial f}{\partial w_{12}}, \ldots \right)$$

  - ⋆ Weights update: take small step in the direction of negative gradient

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha \frac{\partial f}{\partial \mathbf{w}}$$

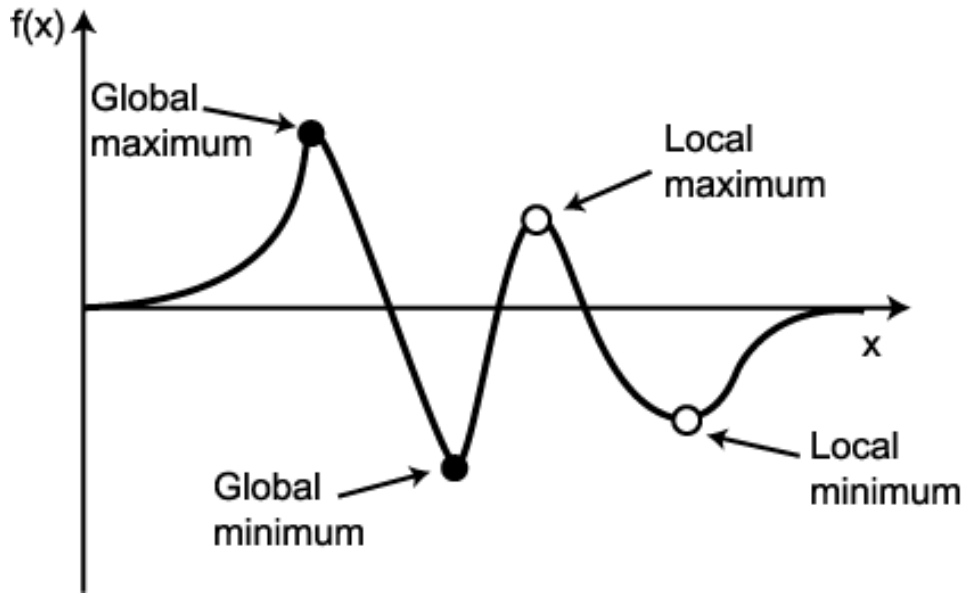
Intuition behind Gradient Descent



Arrows point in minus gradient direction towards the minimum

Intuition behind Gradient Descent

The iterative algorithm might converge to one of the many local minima

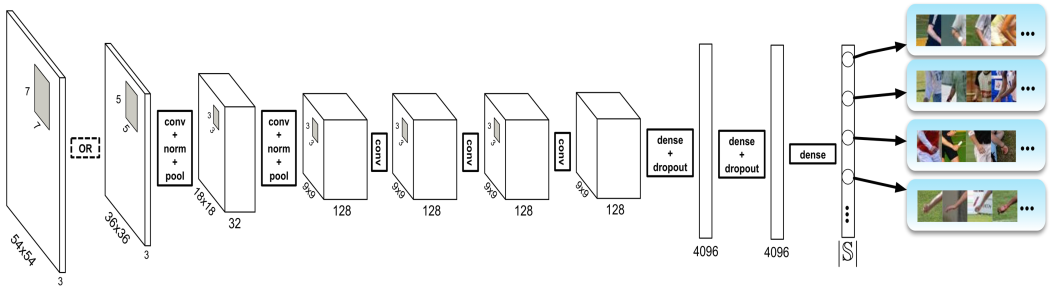


In neural networks, gradient is computed through a procedure called back propagation

Why Deep Learning Works Now? (but not 30 years ago)

- Lots more computing power
  - ⋆ GPU: mini-super computer: 3000 cores, 12GB RAM, 100x speed up over CPU
  - ⋆ Much deeper and larger networks, trained for a few days-a week

(e) GTX Titan X



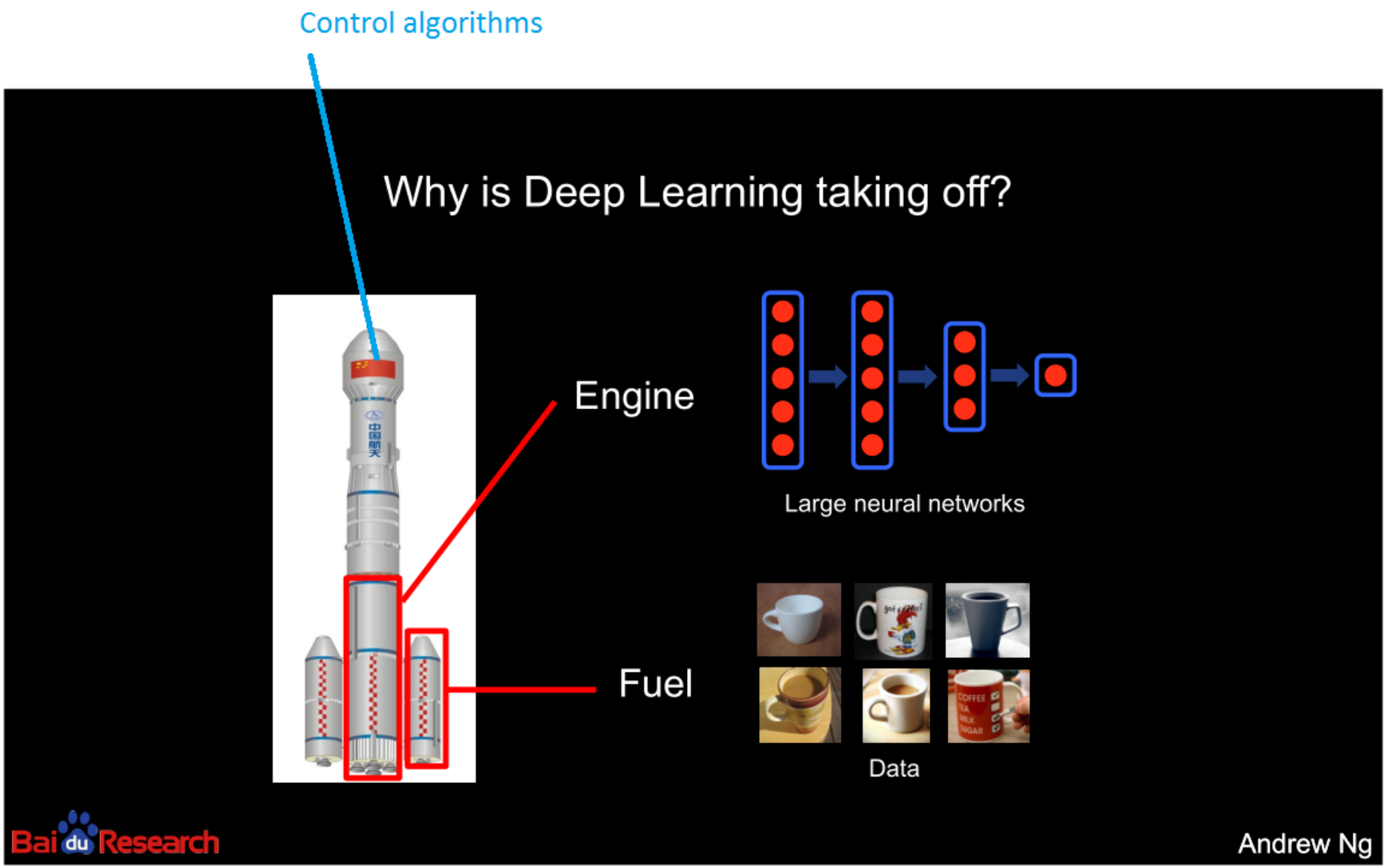
(f) A deep network

Why Deep Learning Works Now? (but not 30 years ago)

- Lots more data
  - ⋆ phone, camera, sensors
  - ⋆ Internet, Croud-Sourcing (Amazon Mechanical Turk) → huge labeled data sets of millions of training samples

Why Deep Learning Works Now? (but not 30 years ago)

- More sophisticated algorithms
  - ⋆ Dropout: strong regularization that prevent deep nets from overfitting
  - ⋆ Clever initialization: initial weights are close to a good minimum
  - ⋆ Stochastic gradient descent: fast approximate gradient descent algorithms
  - ⋆ Rectified Linear Unit: ReLU$(x) = \max(0, x)$


Control algorithms
Why is Deep Learning taking off?
Engine
Fuel
Large neural networks
Data
Baidu Research
Andrew Ng

Further Resources

http://www.wired.com/2014/01/geoffrey-hinton-deep-learning
http://chronicle.com/article/The-Believers/190147/

Courses

- COMP90051 Machine Learning
- COMP30018-COMP90049 Knowledge Technologies
- https://class.coursera.org/neuralnets-2012-001
- https://www.coursera.org/course/ml