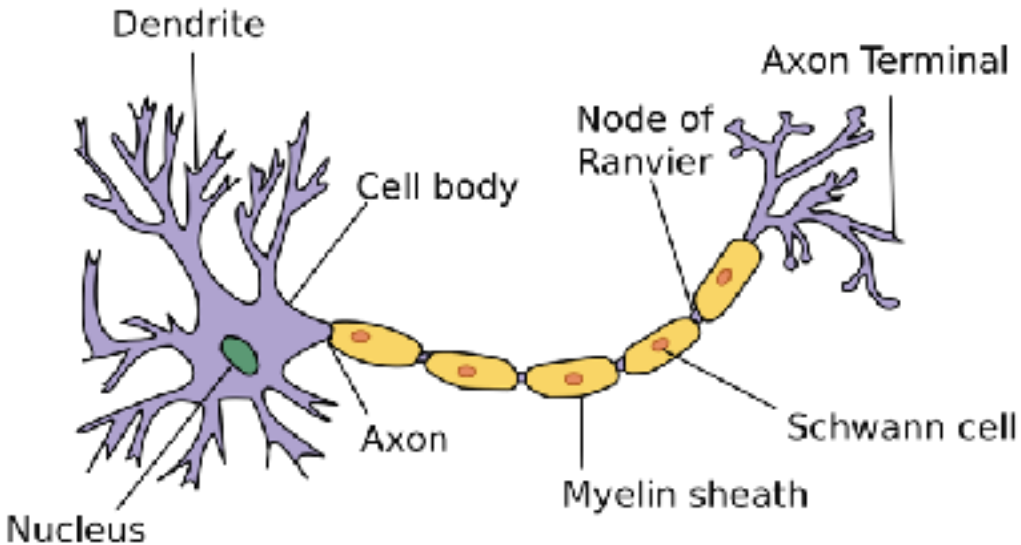
GENERAL ASSEMBLY

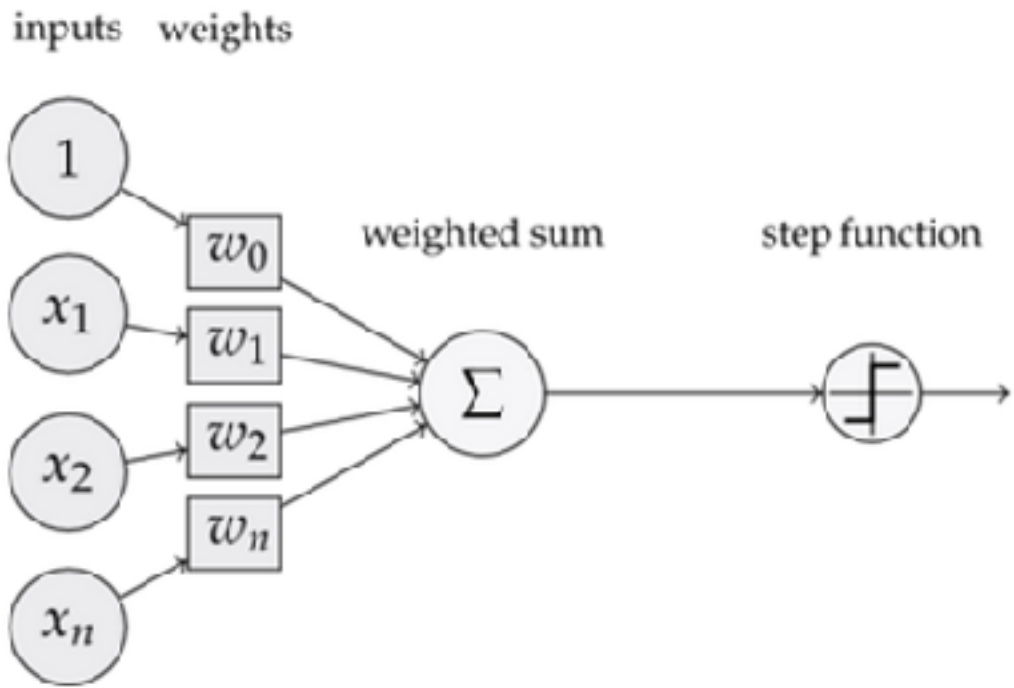
DATA SCIENCE
10 WEEK PART TIME COURSE

Week 9 – Artificial Neural Networks
Wednesday 7th December 2016

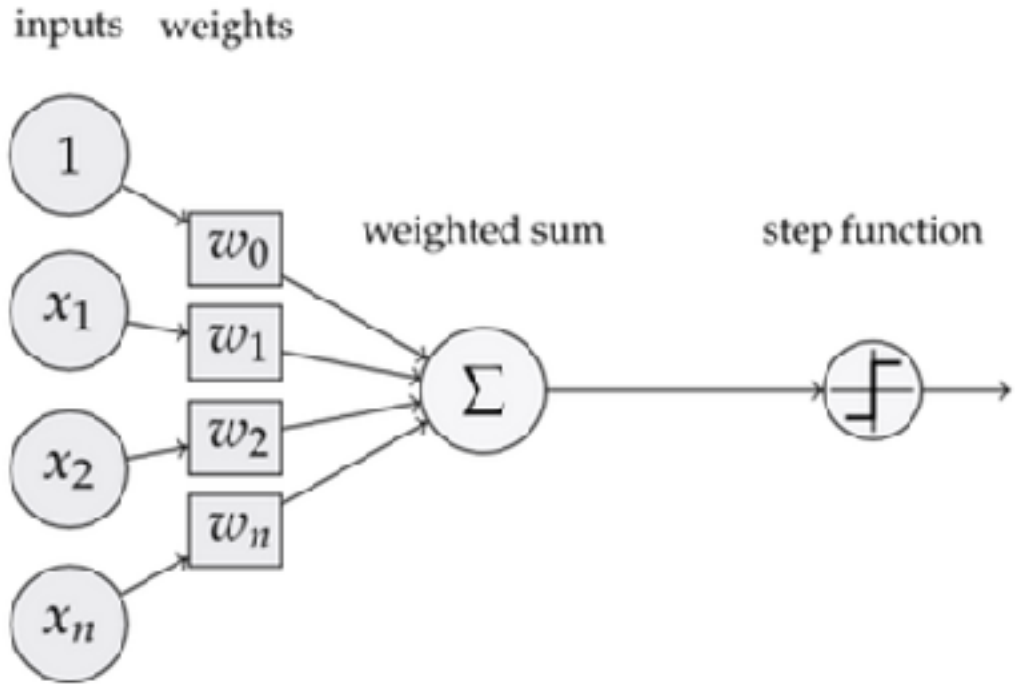
1. Review of Communication module and Causality
2. Artificial Neural Networks
3. Lab
    1. Use a neural network algorithm from SciKit Learn
    2. Set up a GPU Deep Learning instance on AWS
    3. Go through a Tensorflow tutorial
4. Plan for project presentations next week

WHAT IS A NEURAL NETWORK?

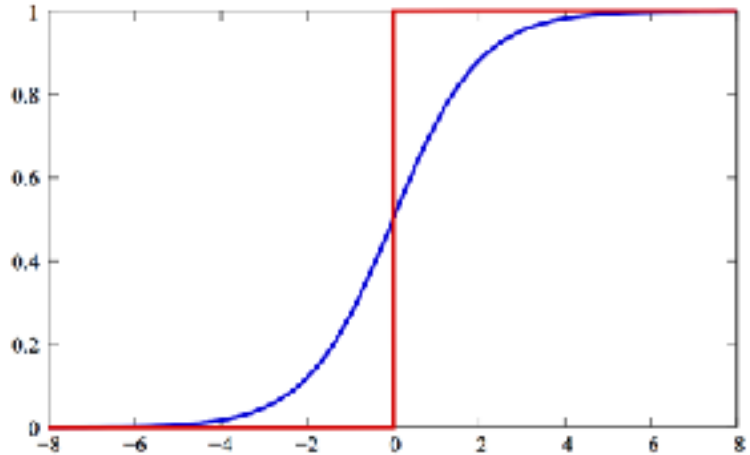

Dendrite
Axon Terminal
Node of
Ranvier
Cell body
Schwann cell
Axon
Myelin sheath
Nucleus


inputs
weights
1
x_1
x_2
x_n
w_0
w_1
w_2
w_n
weighted sum
Σ
step function

A computational system comprised of layers and each layer is built of interconnected perceptrons



Takes in input and uses an activation function in order to output

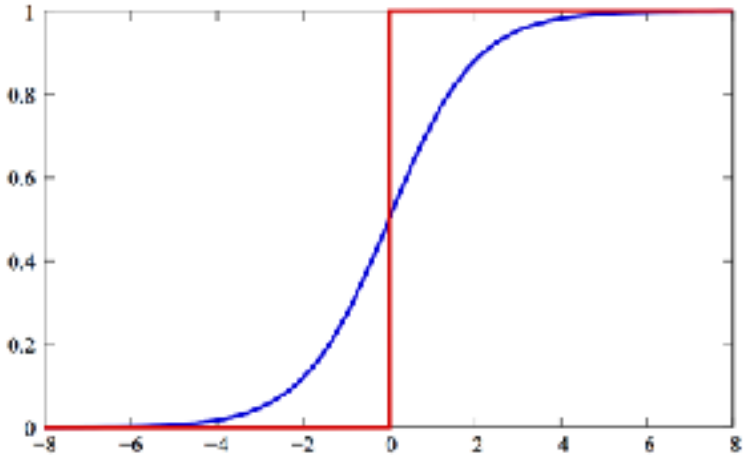


$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$

f_{log} is called logistic function

Takes in input and uses an activation function in order to output

What is z?



$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$

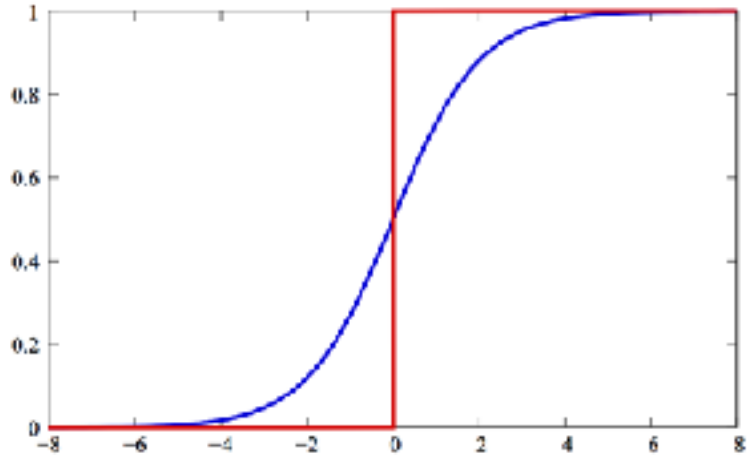
f_{log} is called logistic function

Takes in input and uses an activation function in order to output.

z is a weighted sum on the inputs.

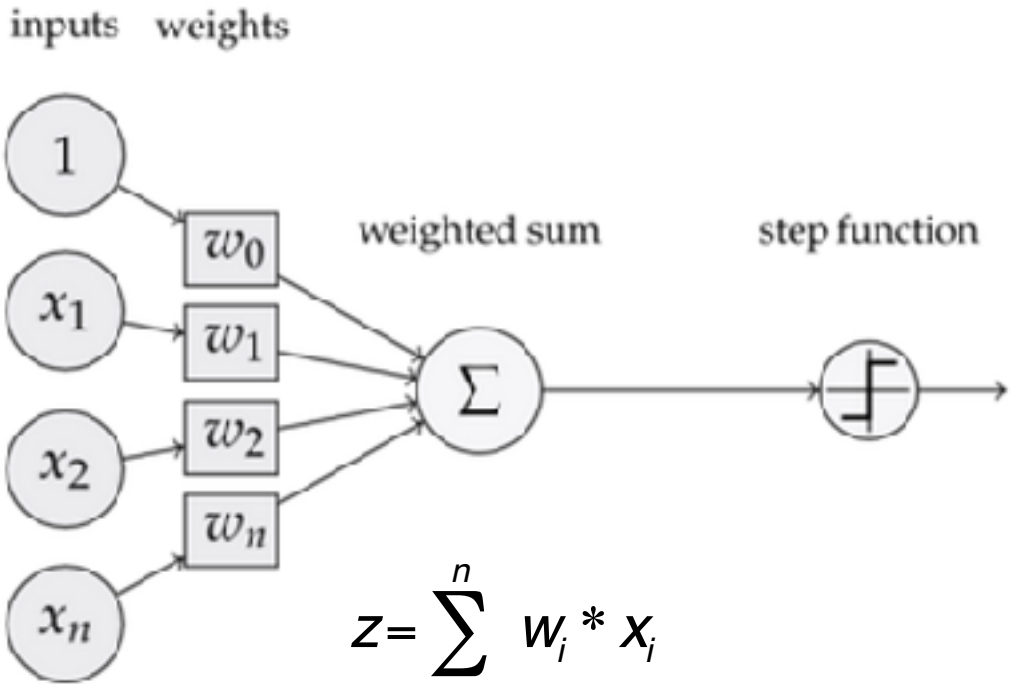
$$z = \sum_{i=0}^{n} w_i * x_i$$

Where w_i is the weight on input x_i



$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$

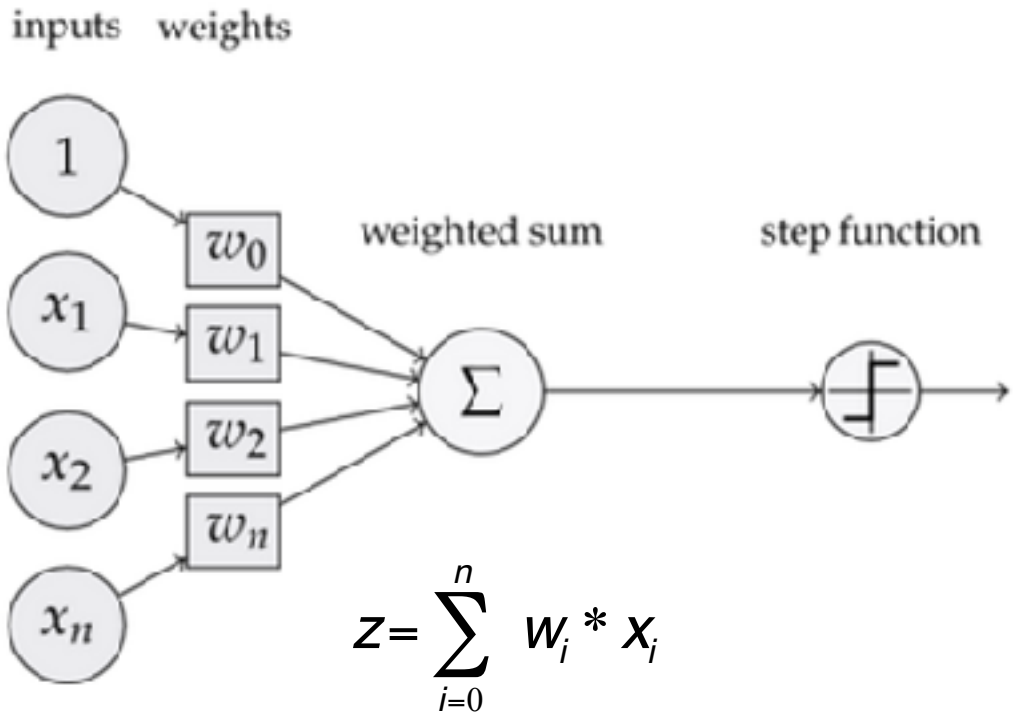
f_{log} is called logistic function



$$z = \sum_{i=0}^{n} w_i * x_i$$

$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$

f_{log} is called logistic function



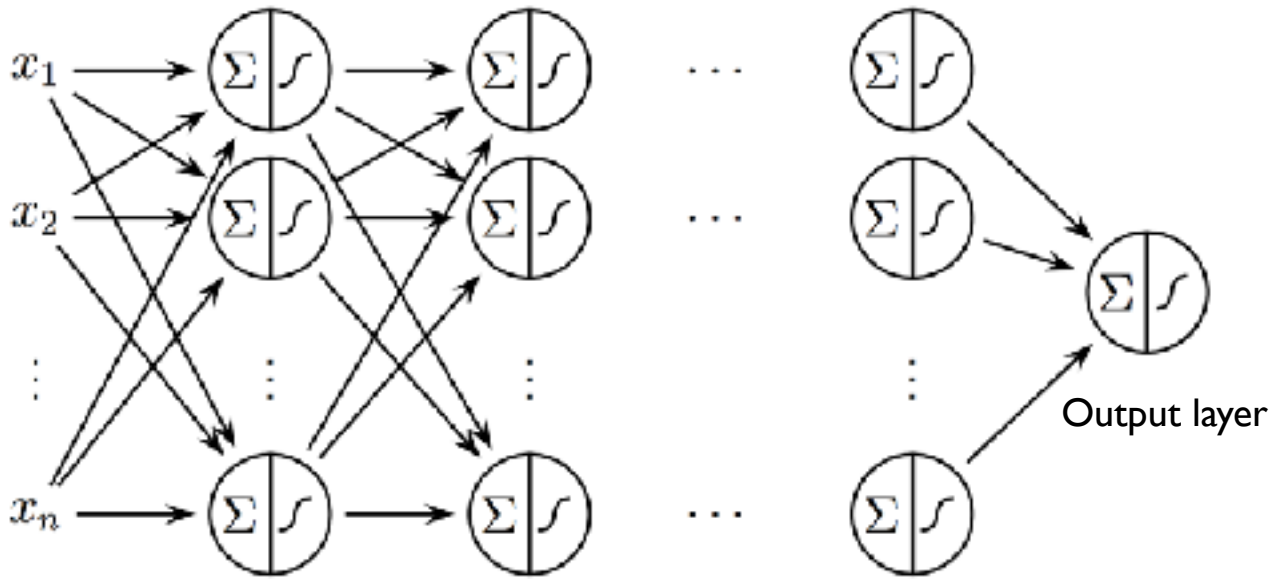
$$z = \sum_{i=0}^{n} w_i * x_i$$

$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$

f_{log} is called logistic function

If f(z) if above a threshold, generally called theta, then the neuron “fires”

A multi layer perceptrons (MLP) is a finite acyclic graph. The nodes are neurons with logistic activation.



DATA SCIENCE PART TIME COURSE

NETWORK STRUCTURE

Input Layer - the original features of our dataset (our X)

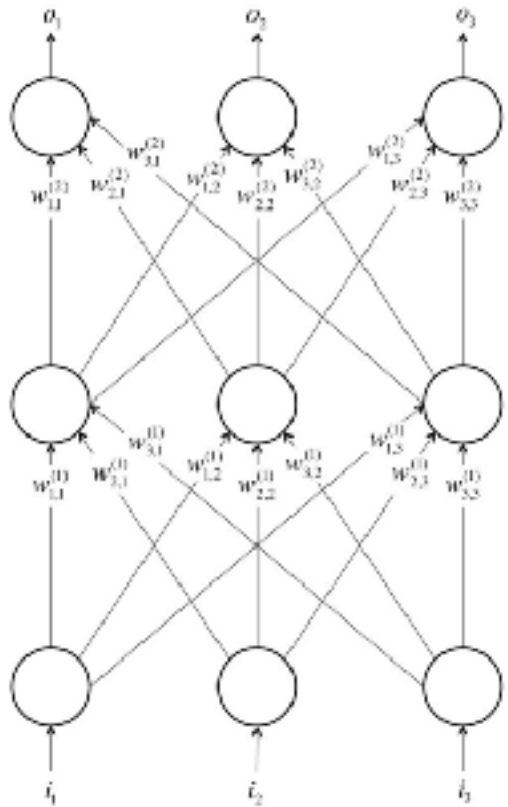
Hidden Layer - these are the derived features of the network. They are called hidden because they are not directly observed.

Output Layer - the final transformation of the inputs into a result

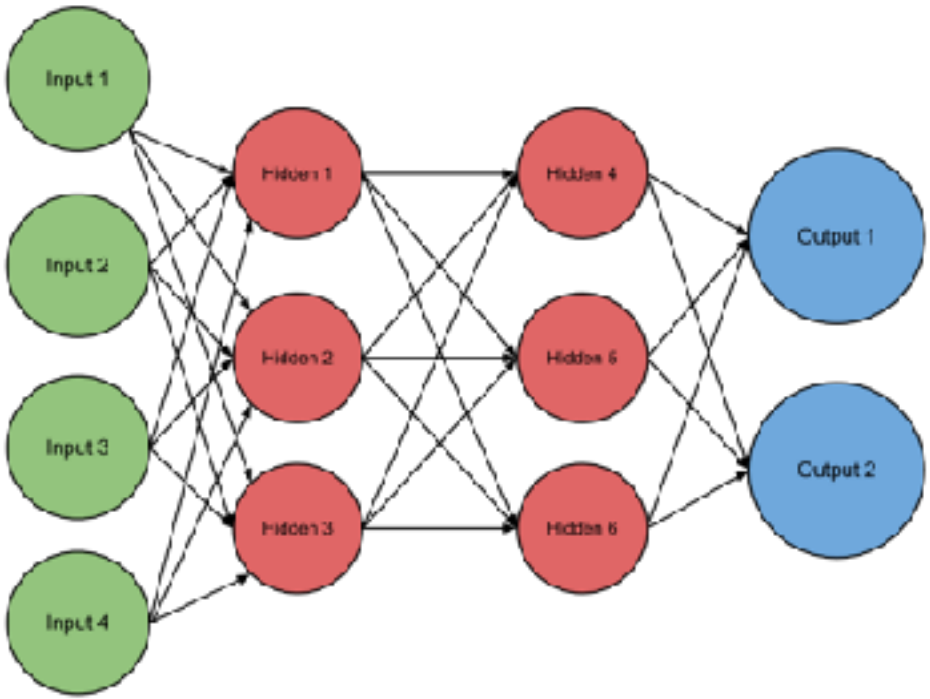
Output Layer ->

Hidden Layers ->

Input Layer ->



Hidden layers often have fewer neurons than the input layer to force the network to learn compressed representations of the original input.



The structure of the network can be thought of as just more hyper-parameters of the model. We search for the optimal structure just as we search for other hyper-parameters such as the learning rate.

It is not required that every neuron has its output connected to the inputs of all neurons in the next layer.

HOW DO WE FIND THE WEIGHTS?

DATA SCIENCE PART TIME COURSE

BACK-PROPAGATION

As we train the model we update the sigmoid function weights in order to get the best predictions possible

If an observation goes through the model and is outputted as False when it should have been True the logistic functions in the single perceptrons are changed slightly.

Back-Propogation is a two-pass algorithm.

The Forward pass fixes the current weights and the predicted values are calculated.

The Backward pass calculates the errors on the output layer and are then back-propagated to give the errors at the hidden layer units.

The good ol' chain rule!

Given the errors calculated between the test observations and the output, calculate the derivative of the error function using the chain rule. From this derivative we can compute the gradient and use it in an optimisation function like gradient descent to improve the weights at that neuron. Repeat down each layer of the network after each error calculation.

Pros

- Online model (updates as you go)
- Very fast predictions
- Can approximate almost any type of function
- Can be used in a supervised and unsupervised manner
- Very topical area of machine learning (lots of investment)
- Getting easier to run

Cons

- Requires many training samples to be considered good
- Hard to describe what is happening
- Requires a lot of hardware / computation power
- (Can be) Slow to train

WHY ARE NEURAL NETWORKS IN THE NEWS?

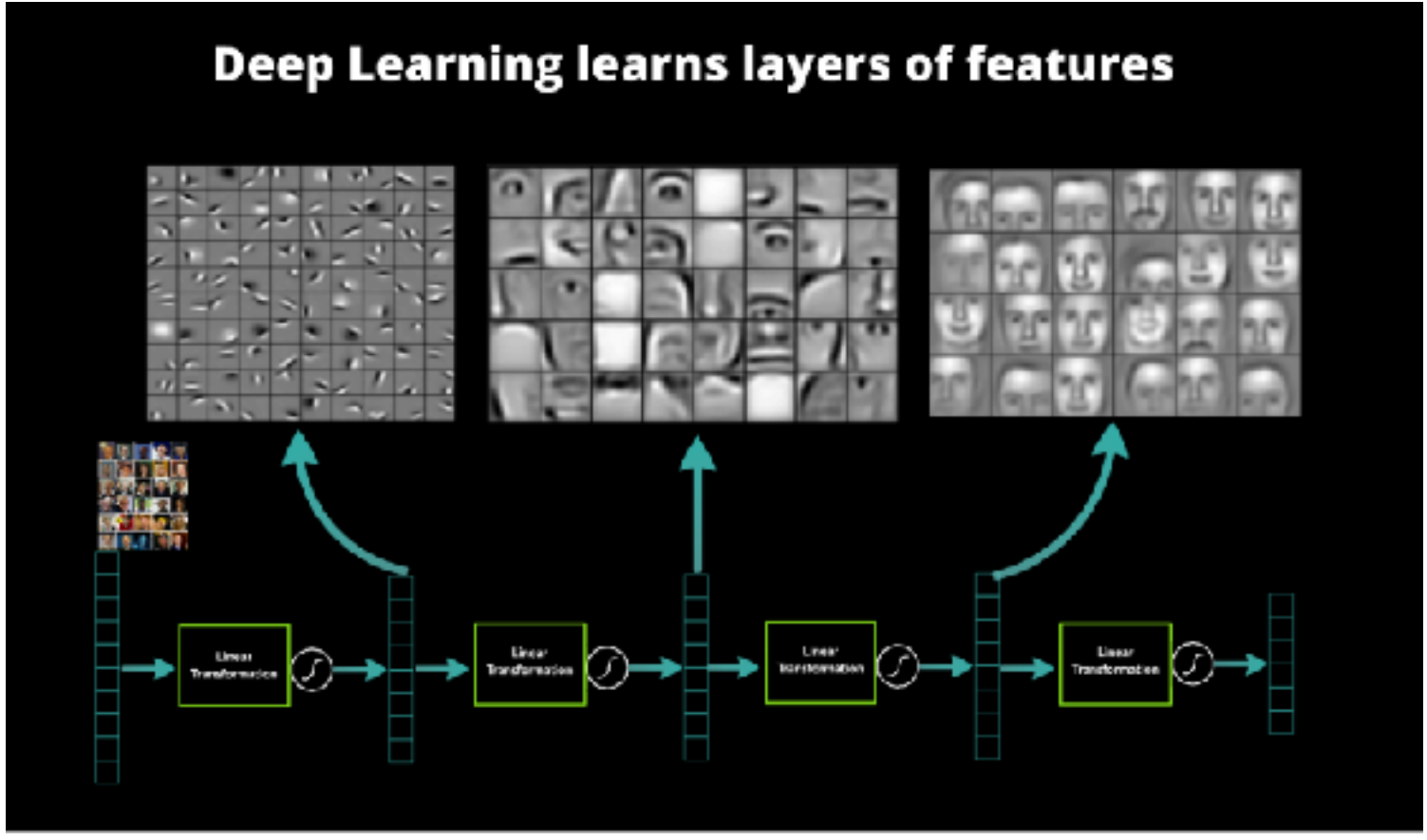
- http://www.wired.com/2016/01/googles-go-victory-is-just-a-glimpse-of-how-powerful-ai-will-be/

DATA SCIENCE PART TIME COURSE

DEEP LEARNING

Traditional feedforward neural networks can be considered to have depth equal to the number of layers (i.e. the number of hidden layers plus 1, for the output layer)

Depth 2 is enough in many cases to represent any function with a given target accuracy. But this may come with a price: that the required number of nodes in the graph may grow very large


Deep Learning learns layers of features
Linear Transformation
Linear Transformation
Linear Transformation
Linear Transformation

When is deep learning the right choice of algorithm?

When the input features are dense:

- Images
- Videos
- Audio
- Text

When is deep learning the right choice of algorithm?

When the input features are dense:

- Images
- Videos
- Audio
- Text

Some recent interesting applications:

- Colorisation of Black and White Images.
- Adding Sounds To Silent Movies.
- Automatic Machine Translation.
- Object Classification in Photographs.
- Automatic Handwriting Generation.
- Character Text Generation.
- Image Caption Generation.
- Automatic Game Playing.

Three papers were published in 2006 that were breakthroughs for Deep Learning. They shared the following principles:

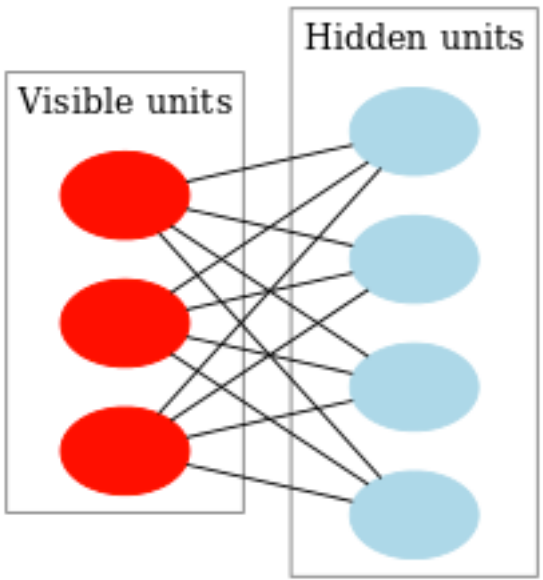
- Unsupervised learning of representations is used to (pre-)train each layer
- Unsupervised training of one layer at a time, on top of the previously trained ones. The representation learned at each level is the input for the next layer
- Use supervised training to fine-tune all the layers (in addition to one or more additional layers that are dedicated to producing predictions)

https://www.quora.com/How-is-deep-learning-different-from-multilayer-perceptron

The visible and hidden units may have a symmetric connection between them, and there are no connections between nodes within a group.

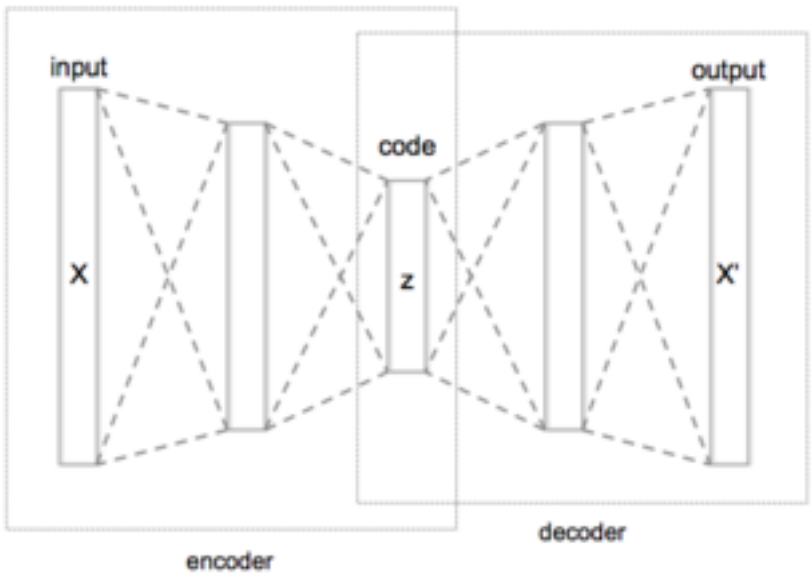
This allows for training to be efficient and takes less time to train.

Deep Belief Networks can be formed by stacking RBMs and then tuning the resulting network

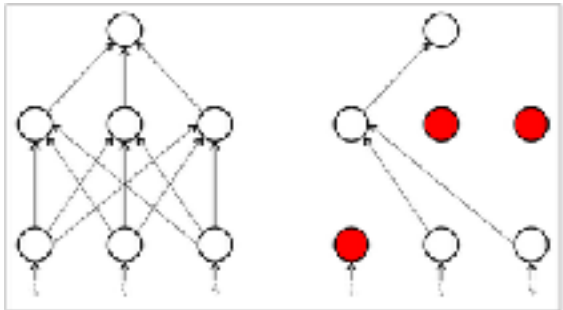


The simplest form of an autoencoder is a feedforward neural net which is very similar to the multilayer perceptron (MLP), with an input layer, an output layer and one or more hidden layers connecting them.

The differences between autoencoders and MLPs, though, are that in an autoencoder, the output layer has the same number of nodes as the input layer. And instead of being trained to predict the target value Y given inputs X, autoencoders are trained to reconstruct their own inputs X. Therefore, autoencoders are unsupervised learning models.


input
code
output
X
z
X'
encoder
decoder

Dropout reduces overfitting by randomly omitting half of the feature detectors on each training case.



A single maxout unit can be interpreted as making a piecewise linear approximation to an arbitrary convex function. Maxout networks learn not just the relationship between hidden units, but also the activation function of each hidden unit.

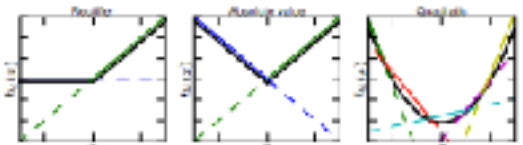
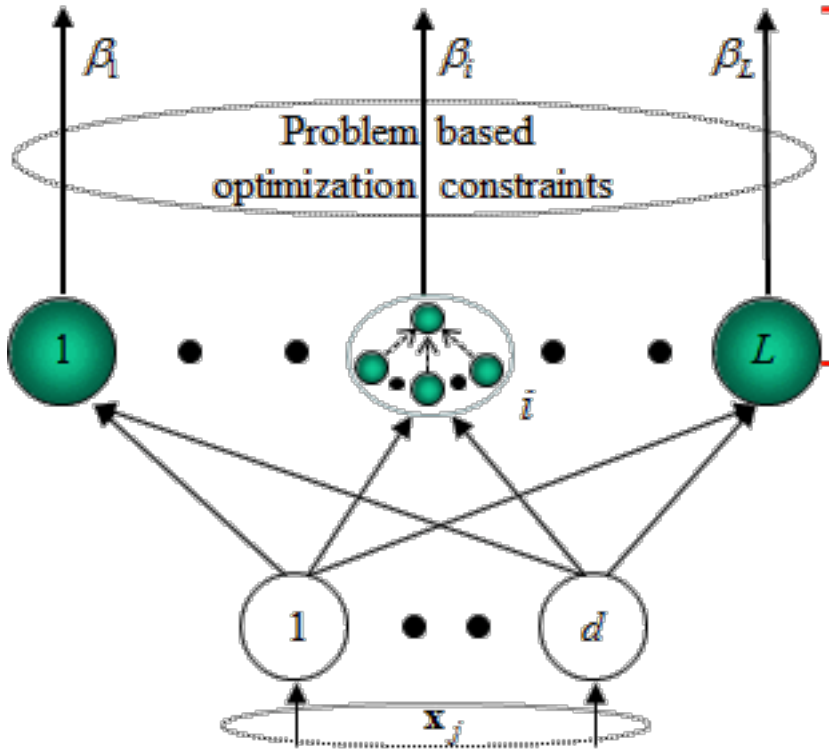


Figure 1. Graphical depiction of how the maxout activation function can implement the rectified linear, absolute value rectifier, and approximate the quadratic activation function. This diagram is 2D and only shows how maxout behaves with a 1D input, but in multiple dimensions a maxout unit can approximate arbitrary convex functions.

Data Parallelism - parallelizes the training process by splitting the data set across processors (GPUs/CPUs). We use the same weights but different mini-batches in each processor. The gradients need to be synchronised after each pass.

Model Parallelism - this involves distributing the neurone across different processors. The output after each layer needs to be synchronised.

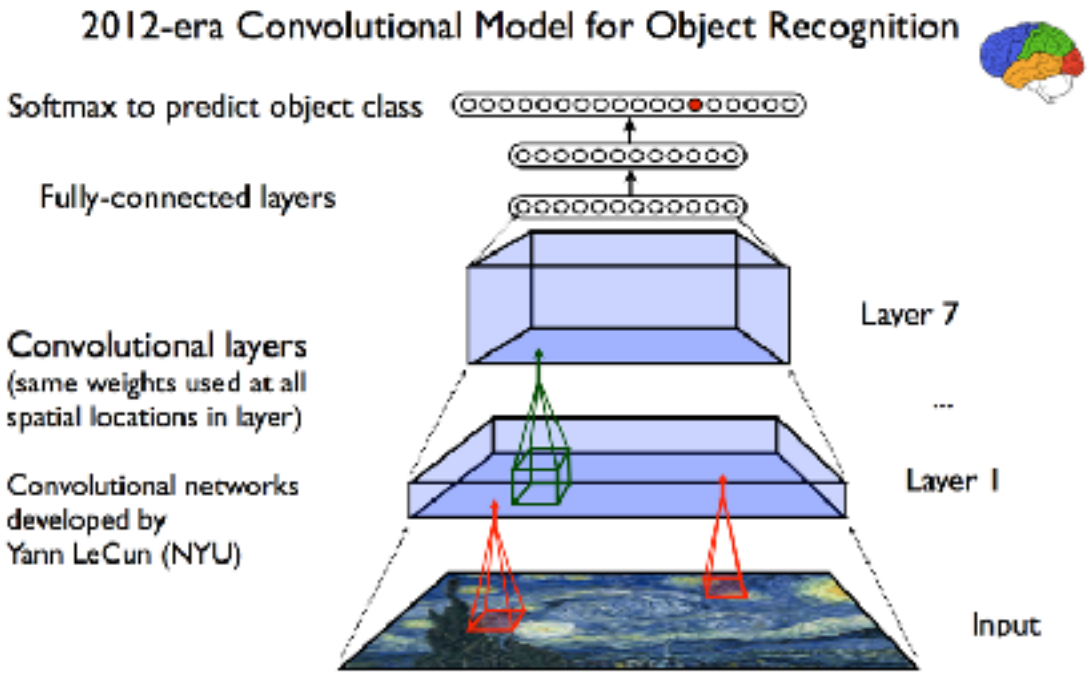


Feature learning
Clustering
Regression
Classification

Hidden nodes need not be tuned. A hidden node of ELM can be a subnetwork of several nodes.

Video on what neural networks see. Helpful for realising how the networks learns to distinguish through transformations.

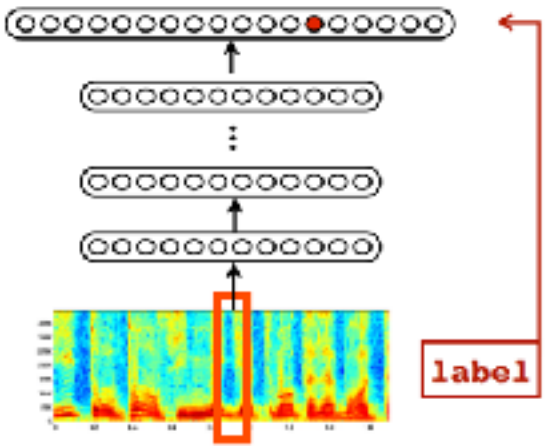
https://aiexperiments.withgoogle.com/what-neural-nets-see



Basic architecture developed by Krizhevsky, Sutskever & Hinton (all now at Google).

Won 2012 ImageNet challenge with 16.4% top-5 error rate

Acoustic Modeling for Speech Recognition



Close collaboration with Google Speech team

Trained in <5 days on cluster of 800 machines

30% reduction in Word Error Rate for English
("biggest single improvement in 20 years of speech research")

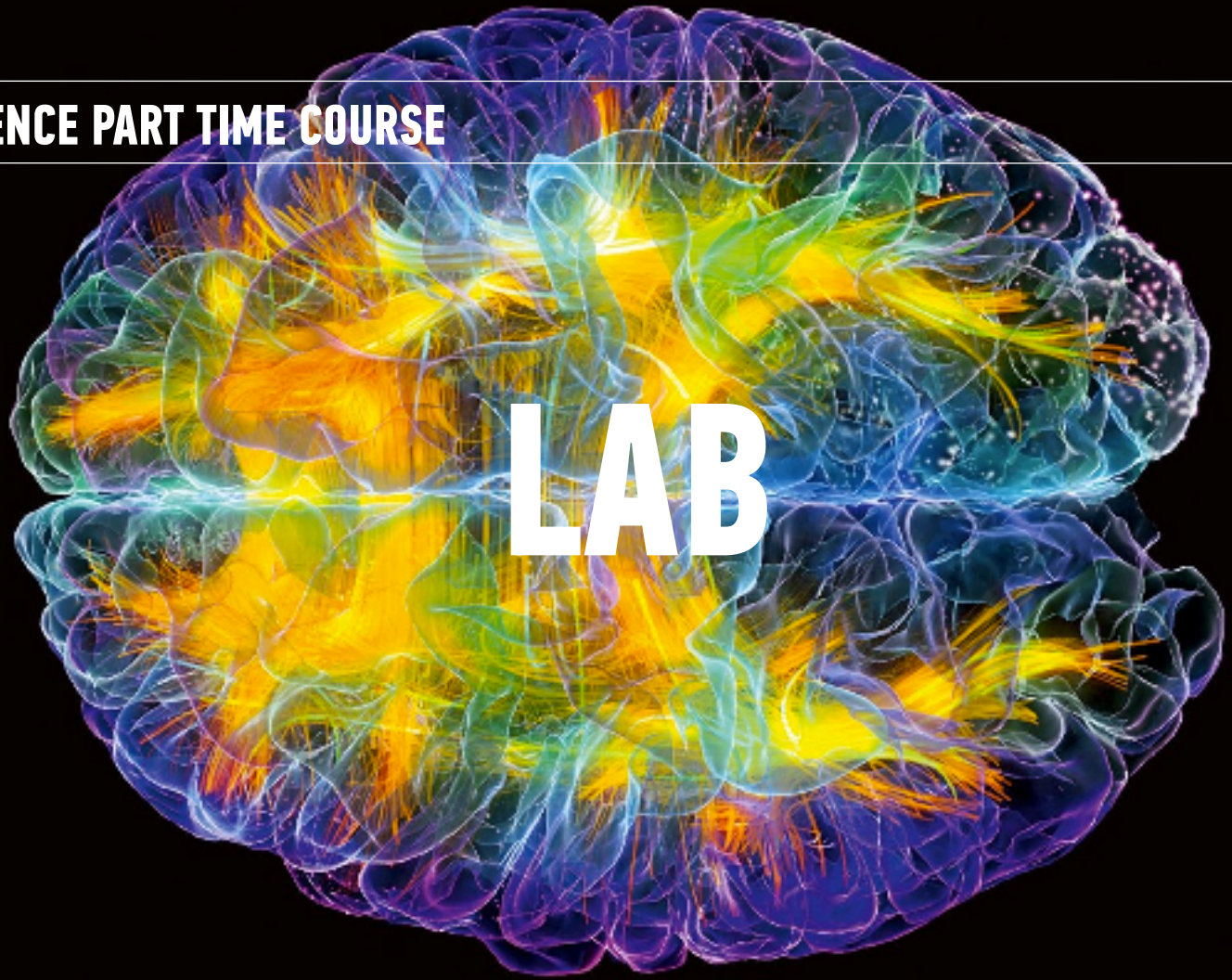
Launched in 2012 at time of Jellybean release of Android

http://deepdreamgenerator.com/

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.

GPUs for processing

DATA SCIENCE PART TIME COURSE



SYNCHING YOUR FORK WITH THE COURSE REPO

1. re-name your labs with lab_name.<yourname>.ipynb (to prevent a conflict)
2. cd <path to the root of your SYD_DAT_6 local repo>
3. commit your changes ahead of sync
   - git status
   - git add .
   - git commit -m "descriptive label for the commit"
   - git status
4. download new material from official course repo (upstream) and merge it
   - git checkout master (ensures you are in the master branch)
   - git fetch upstream
   - git merge upstream/master

A demo of deep learning being applied in the browser:

http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html

HOMEWORK

Homework

- Prepare you project presentations

Reading

- Evaluation of deep learning frameworks: https://github.com/zer0n/deepframeworks/blob/master/README.md
- Beginner: Fundamentals of Deep Learning – Nikhil Budhema
- Advanced: Deep Learning – Ian Goodfellow, Yoshua Bengio, Aaron Courville

PRESENTATIONS

- 10 mins presentation with 5 mins for questions
  - What did you do?
  - What were the results?
  - What did you achieve?
  - What did you learn?
  - What else will you try in the future?
  - Appendix with any interesting findings
- On your own laptop or mine, your choice

PRESENTATIONS

Student	Presentation Date	Presentation Slot
Sriram Rajagopalan	Monday 30/11/16	1
Elena Irsetskaya	Monday 12/12/16	1
Susan do	Monday 12/12/16	2
Quan Dai	Monday 12/12/16	3
Muhsin Karim	Monday 12/12/16	4
Wendy Wong	Monday 12/12/16	5
Roberto	Monday 12/12/16	6
Sai Krishna	Monday 12/12/16	7
Alister Palmer	Monday 12/12/16	8
Harry Peppit	Wednesday 14/12/16	1
Raj Srikanth	Wednesday 14/12/16	2
Tim Walker	Wednesday 14/12/16	3
James Katz	Wednesday 14/12/16	4
Jiamin Lim	Wednesday 14/12/16	5
Simon Wong	Wednesday 14/12/16	6
Jon Kaethner	Wednesday 14/12/16	7
Martin Cvizek	Wednesday 14/12/16	8