GENERAL ASSEMBLY

DATA SCIENCE
11 WEEK PART TIME COURSE

Week 10 – Artificial Neural Networks
Wednesday 25th May 2016

1. Tasks from Wednesday
2. Artificial Neural Networks
3. Lab
4. Review of last week

DATA SCIENCE - Week 10 Day 2

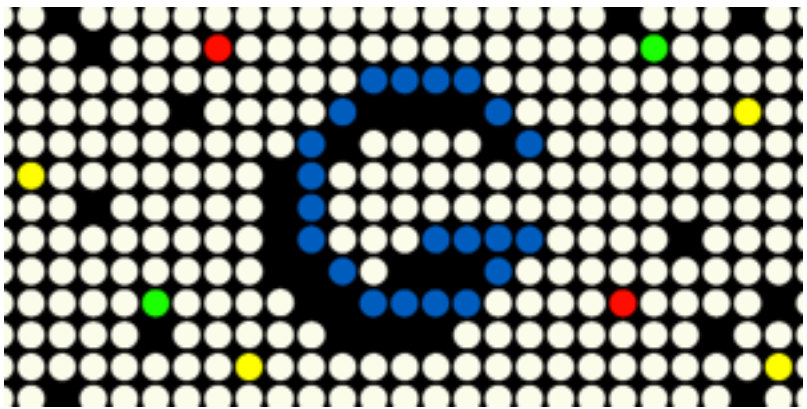
Task List

- [ ] Check EC2 is Working
- [ ] Read the following articles:
  - [ ] http://www.wired.com/2016/01/googles-go-victory-is-just-a-glimpse-of-how-powerful-ai-will-be/
  - [ ] http://www.wired.com/2014/01/geoffrey-hinton-deep-learning
  - [ ] https://sites.google.com/site/deepernn/home/blog/briefsummaryofthepaneldiscussionatdlworkshopicml2015
- [ ] Install TensorFlow

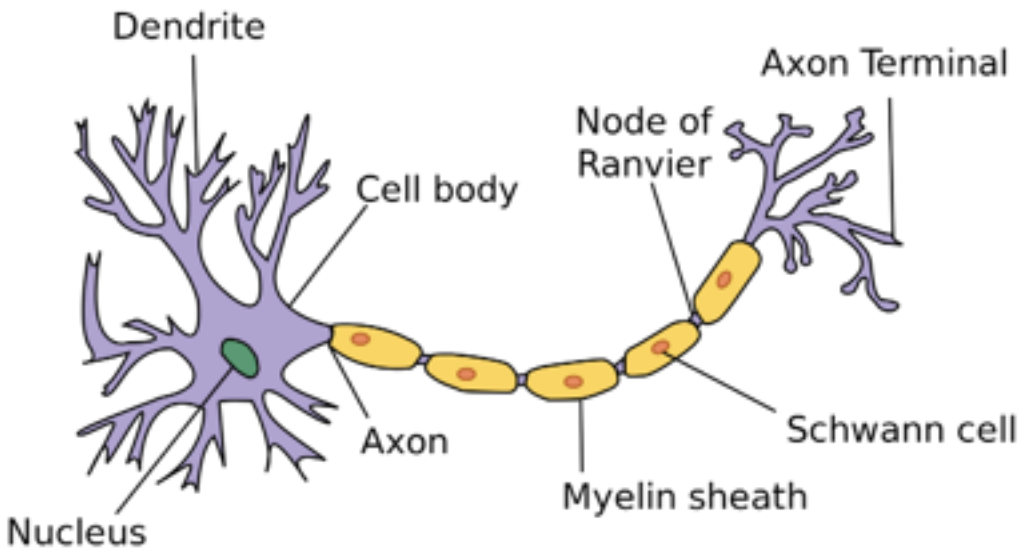
MEET THE MAN GOOGLE HIRED TO MAKE AI A REALITY

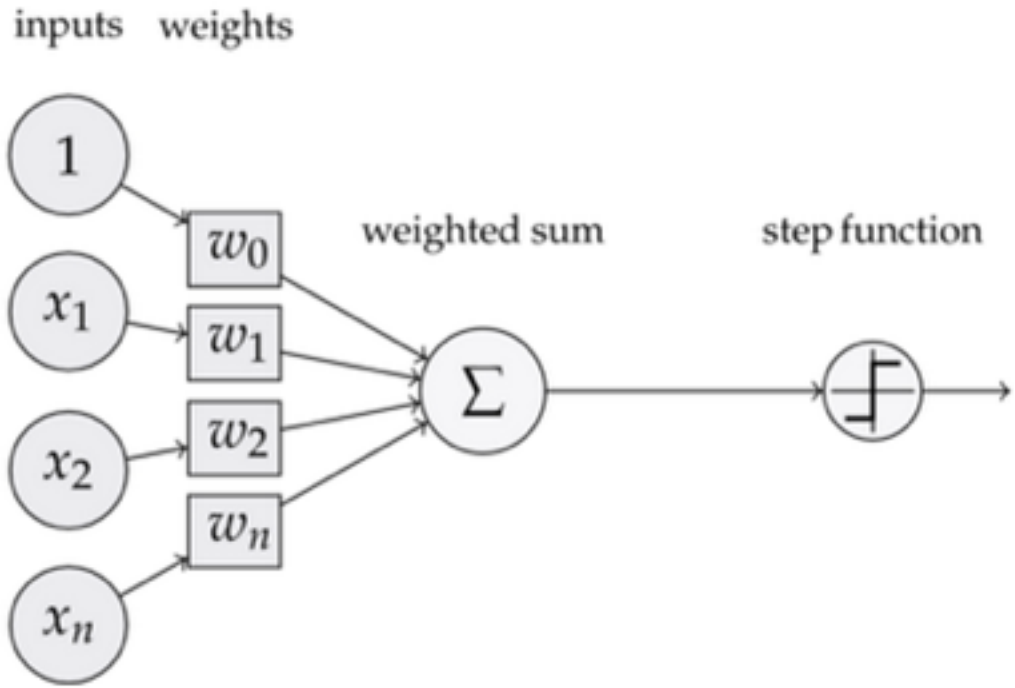
Brief Summary of the Panel Discussion at DL Workshop @ICML 2015

posted Jul 13, 2015, 5:27 AM by KyungHyun Cho [**updated Jul 14, 2015, 11:04 PM**]

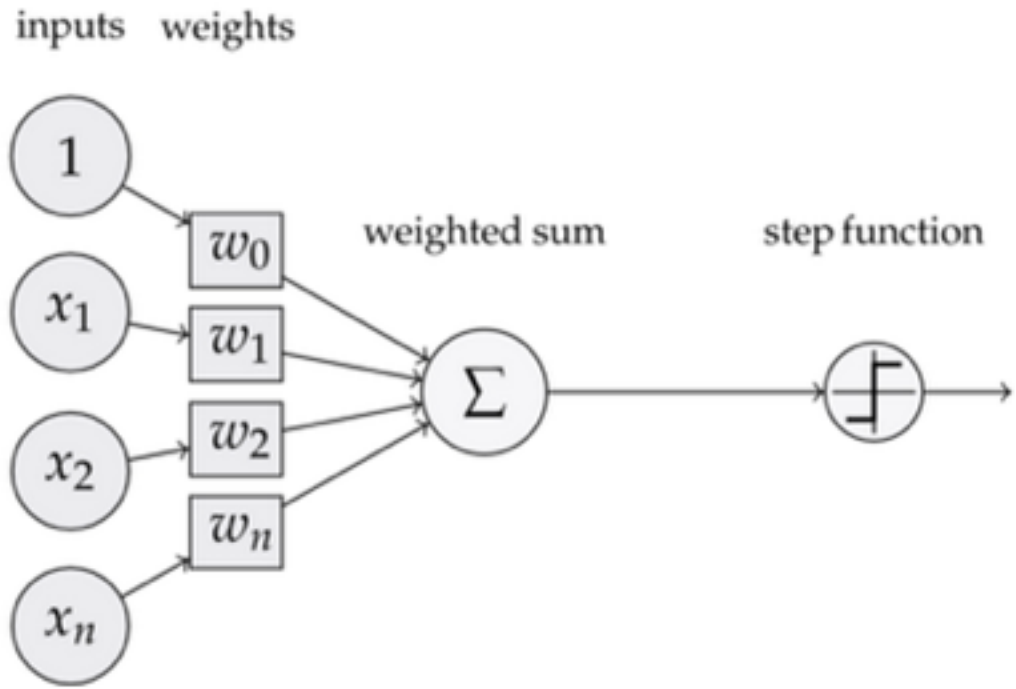


WHAT IS A NEURAL NETWORK?

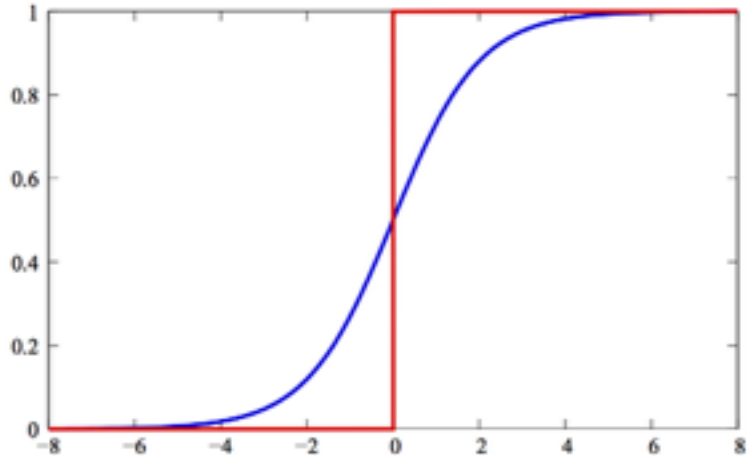

Dendrite
Axon Terminal
Node of
Ranvier
Cell body
Schwann cell
Axon
Myelin sheath
Nucleus


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

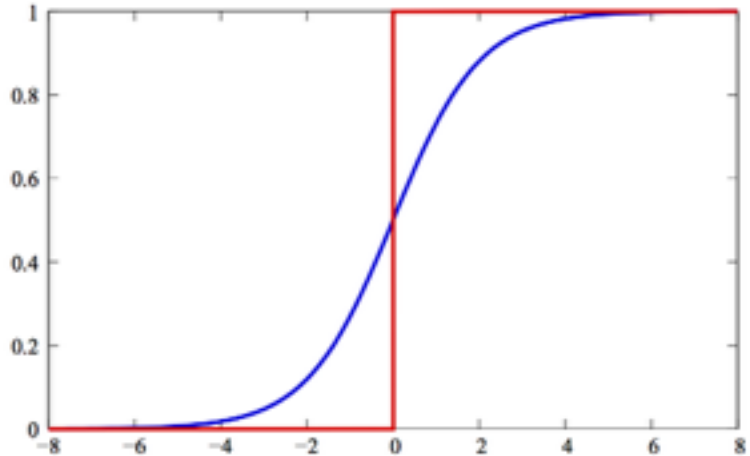


$$f_{log}(z) = \frac{1}{1 + e^{-z}}$$

f_{log} is called logistic function

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

What is z?



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

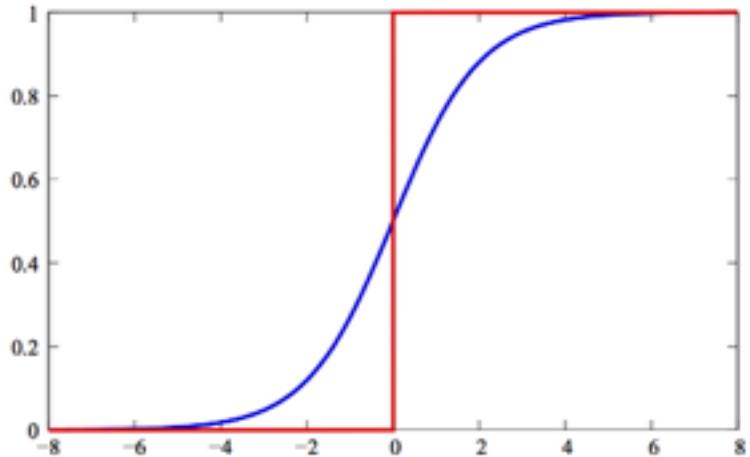
f_{log} is called logistic function

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

z is a weighted sum on the inputs.

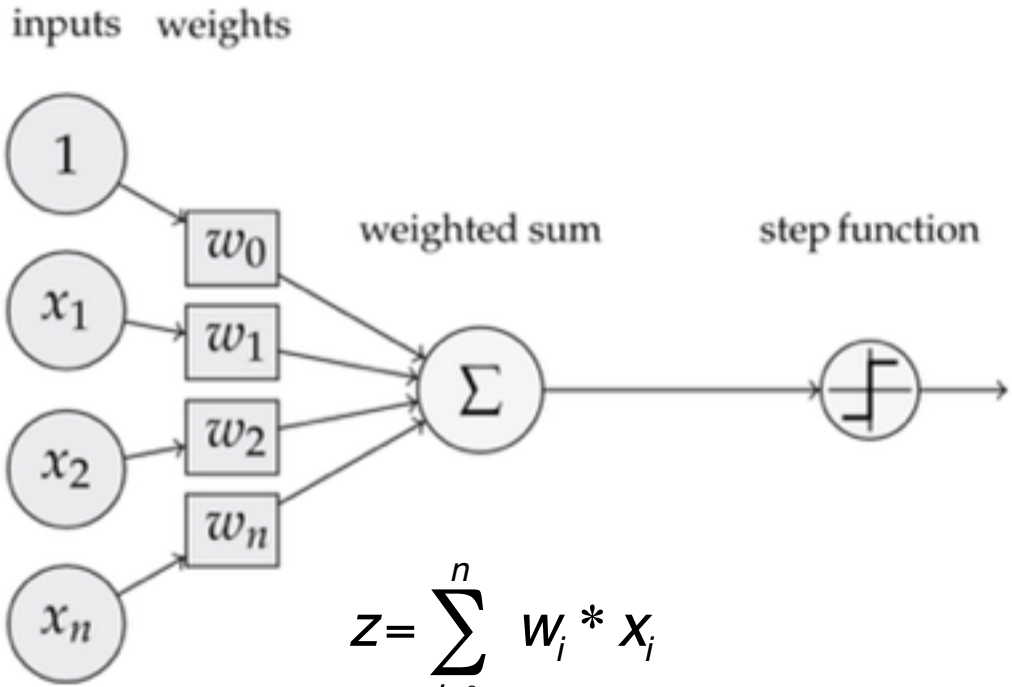
$$z = \sum_{i=0}^{n} w_i * x_i$$

Where w_i is the weight on input x_i



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

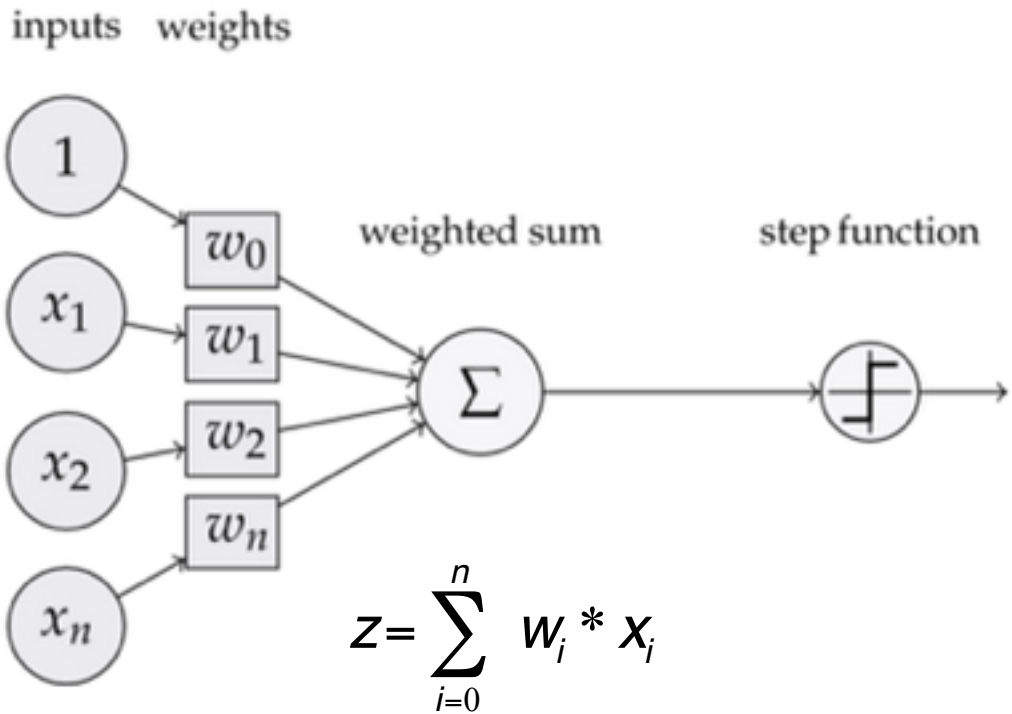
f_{log} is called logistic function



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

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

f_{log} is called logistic function



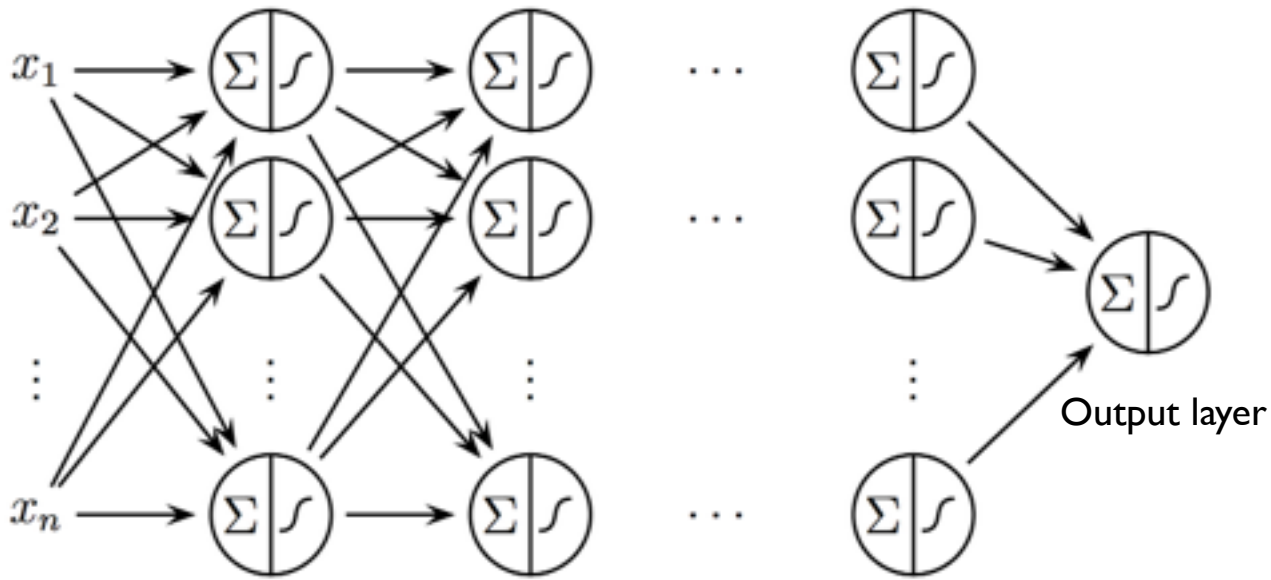
$$z = \sum_{i=0}^{n} w_i * x_i$$

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

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.



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

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.

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?

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

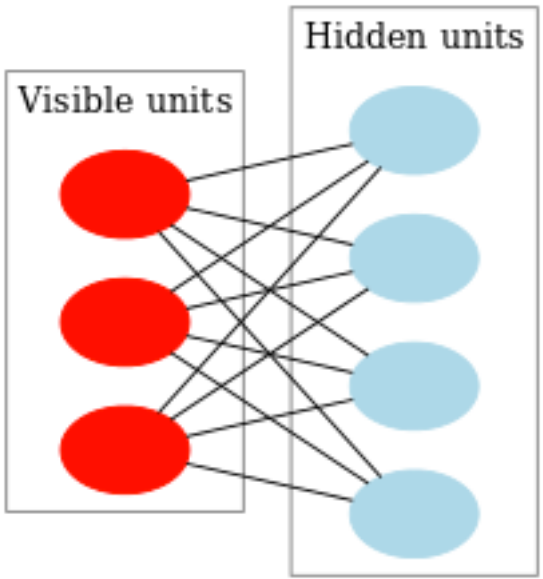
Three papers were published in 2006 that were breakthroughs for Deep Learning. They shared the following principles:

- 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)

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

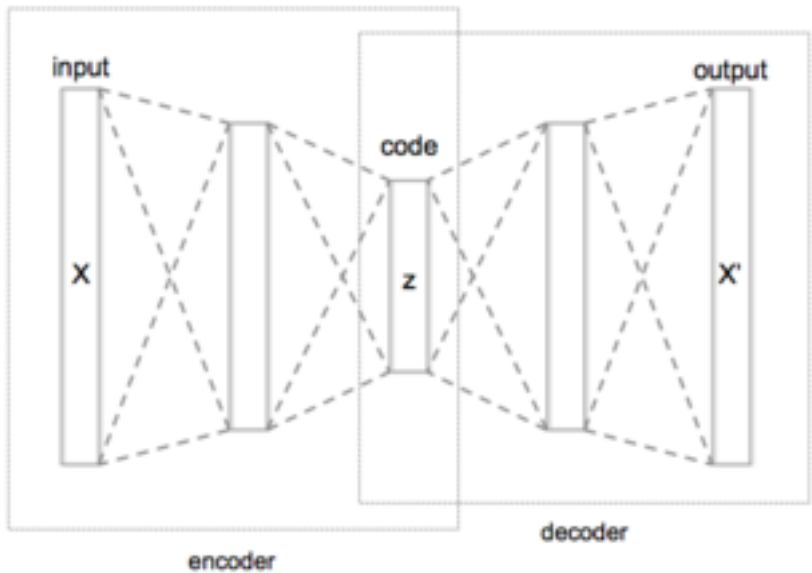
This allows for training to be efficient and takes less time to train.

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
output
code
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.

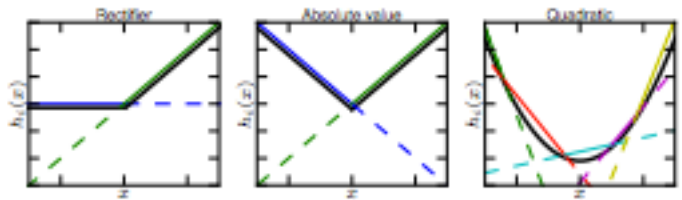
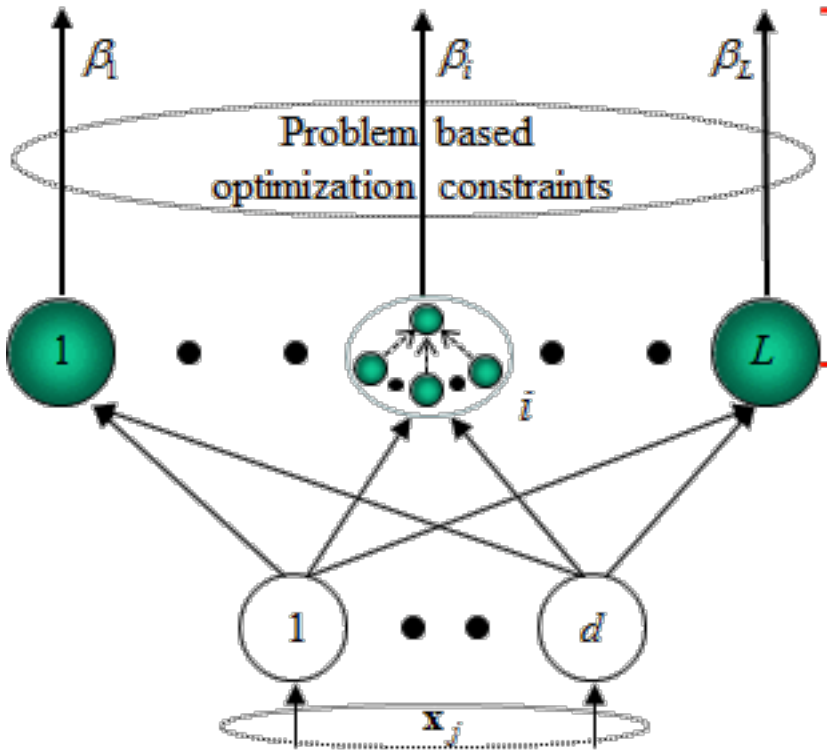


Figure 1. Graphical depiction of how the maxout activation function can implement the rectified linear, absolute value rectifier, and approximate the quadratic activation function. This diagram is 2D and only shows how maxout behaves with a 1D input, but in multiple dimensions a maxout unit can approximate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

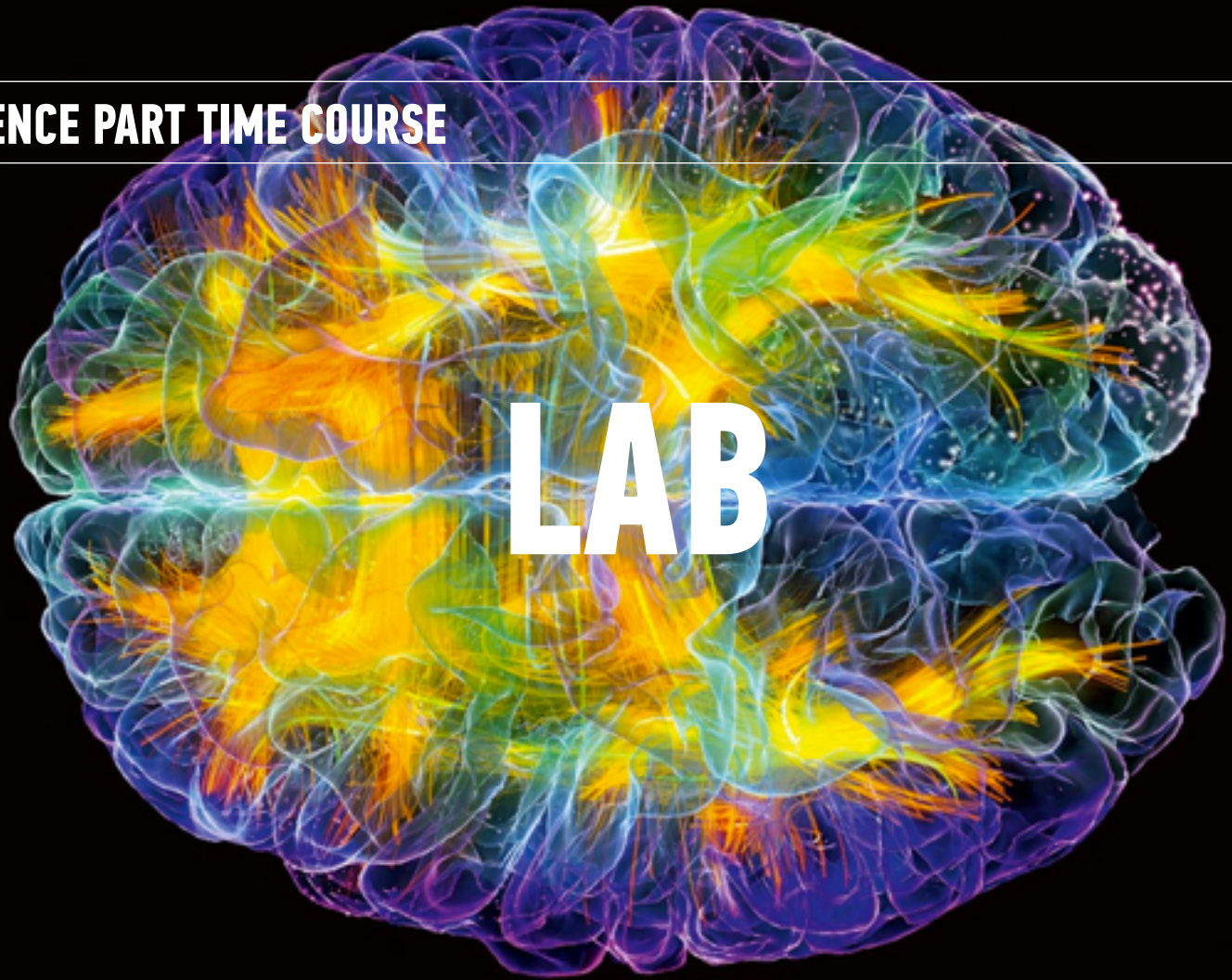


Feature learning
Clustering
Regression
Classification

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

http://deepdreamgenerator.com/

DATA SCIENCE PART TIME COURSE



DISCUSSION TIME

- Questions
- Review
- Presentations

NATURAL LANGUAGE PROCESSING

COMMUNICATION SKILLS

DATA SCIENCE – Week 11 Day 1

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

- Saturday 10 am – 3 pm @ General Assembly
- Available to go over presentations
- Send me the presentation on Friday evening if you want to come in on the Saturday
- Give me a guide of what time you'll be in
- I'll still be available over Slack if you can't come in