Hacker's Guide to Neural Networks

a. k. a.

Data Driven Code - 101

Anoop Thomas Mathew

@atmb4u

FullContact

https://fullcontact.com

WHAT WE WILL NOT COVER

- Recent developments in
  - United States of America! ;)
  - Deep Learning, CNN, RNN, DCGAN etc.
  - Frameworks like TensorFlow, Theano, Keras etc.
- Advanced concepts of neural networks

FullContact

WHAT WE WILL COVER

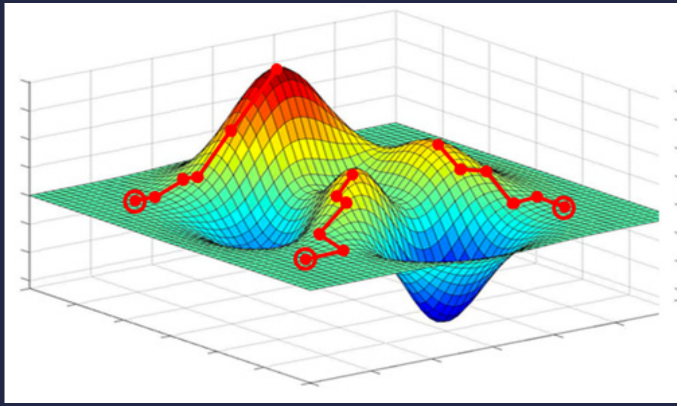
- Basic Concepts of
  - Parameter Optimization
  - Entropy (Sparse Coding)

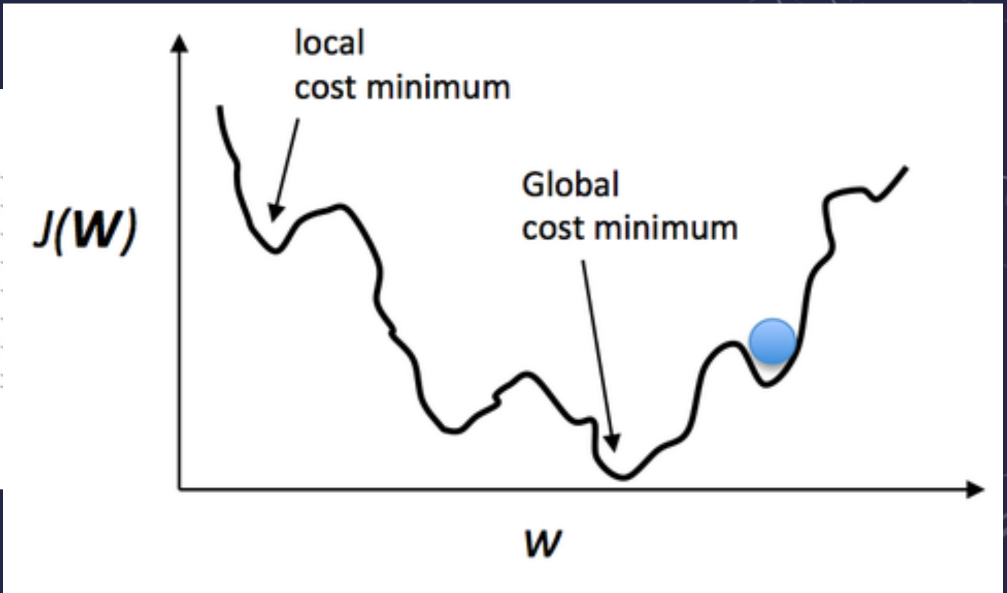
- Little bit of Mathematics
  - Linear Algebra – Matrix Multiplication
  - Differential Calculus – Sigmoid Function (2 equations only)

- (Try to) Build a 2 layer neural network
- Way Forward

FullContact

PARAMETER OPTIMIZATION

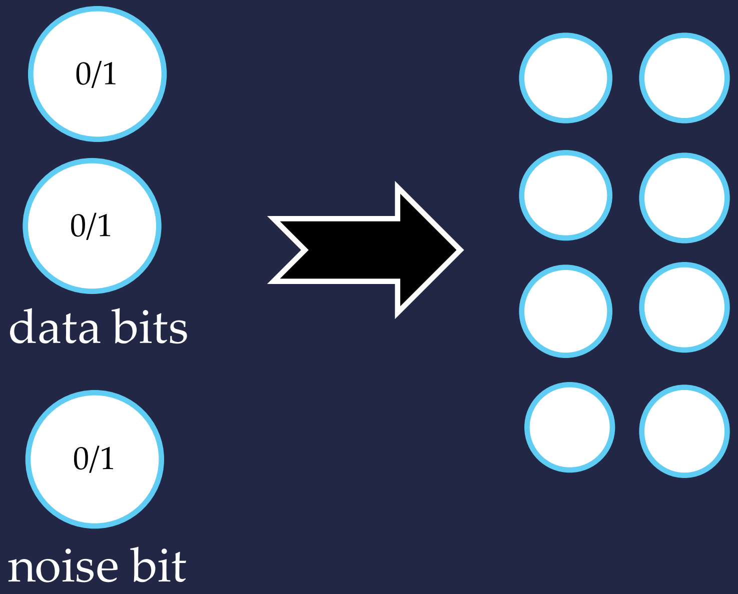




- Infinite Monkey Theorem
- Any problem is fundamentally a parameter optimization problem

ENTROPY

- Sparse Coding



0/1

0/1

data bits

0/1

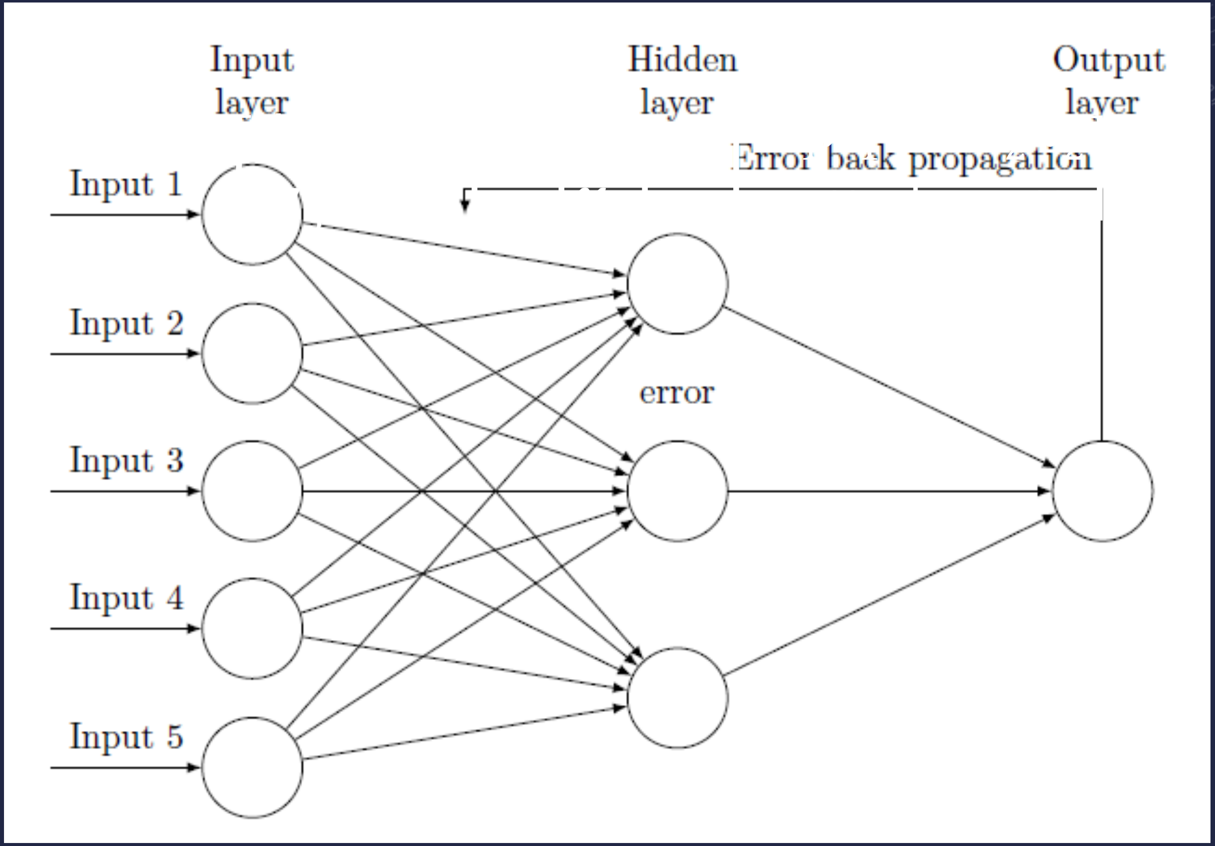
noise bit

$2^8 = 128$

Imagine:

- divided into 4 spaces
- each bit – a feature
- generalize data bits
- attenuates noise bit

LINEAR ALGEBRA PART



LINEAR ALGEBRA PART (matrix multiplication)

$$\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 \\ w_3x_1 + w_4x_2 \end{bmatrix}$$

FullContact

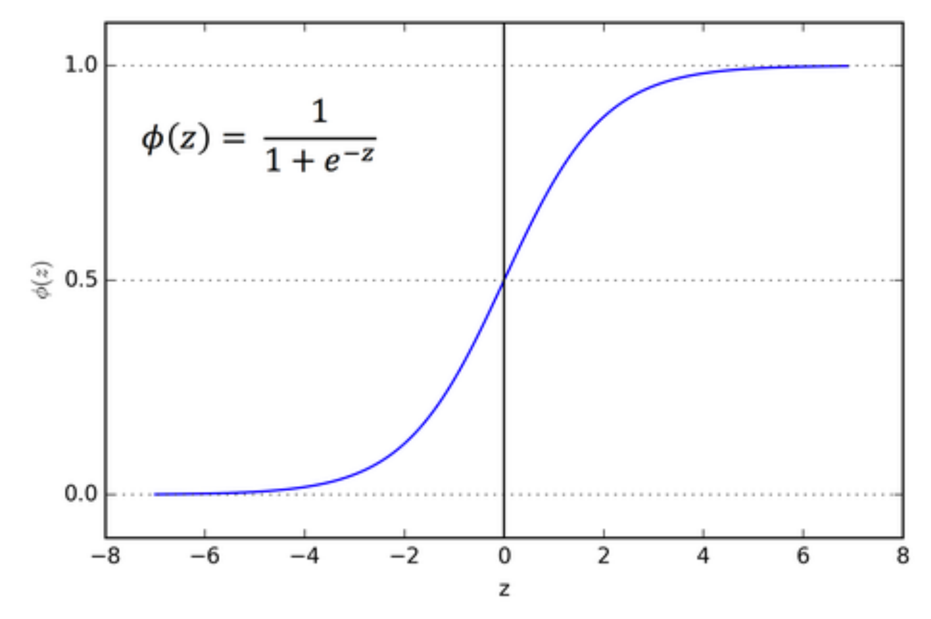
ACTIVATION FUNCTION (differential calculus part)

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

(Sigmoid Function)

$$\frac{d\,\partial\sigma(x)}{dx} = \sigma(x) \times (1-\sigma(x))$$

(Derivative of Sigmoid Function)



Stochastic Gradient Decent

QUICK RECAP

- Layers
- Dataset
- Training
- Network
- Features
- Spare Encoding
- Back Propagation
- Activation Function
- Parameter Optimization

FullContact

LET'S MAKE ONE NOW !

CODE + DATA = MAGIC

(Teach machine **XOR** truth table)

FullContact

XOR GATE

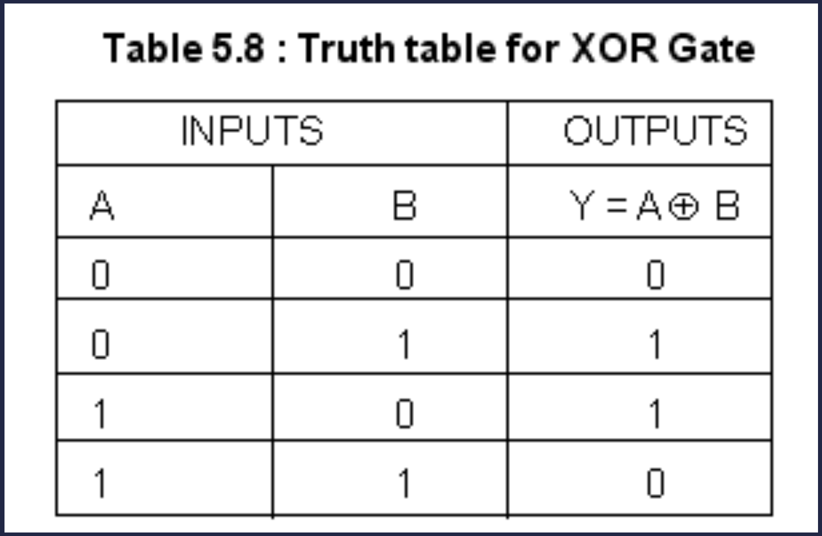
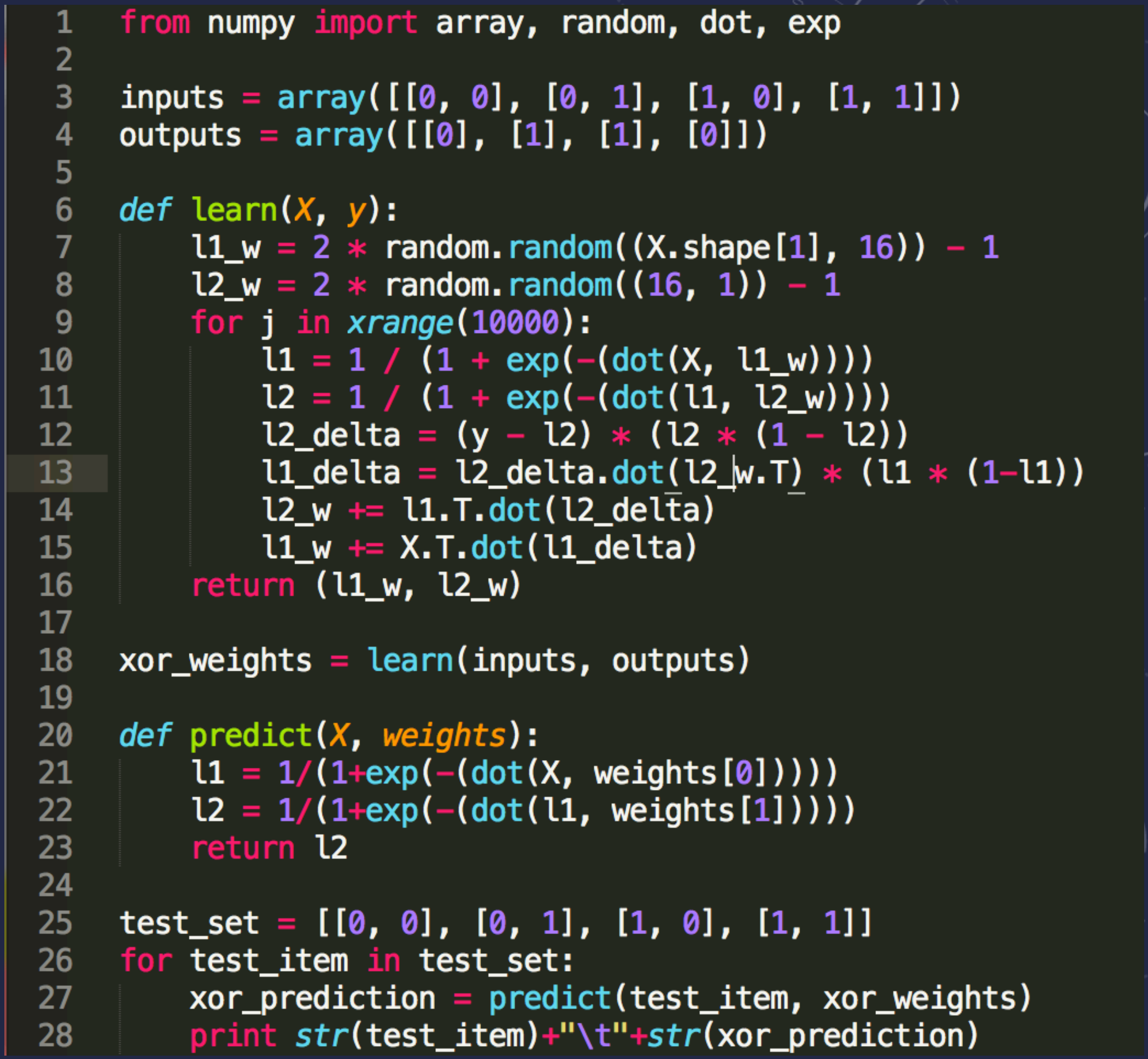


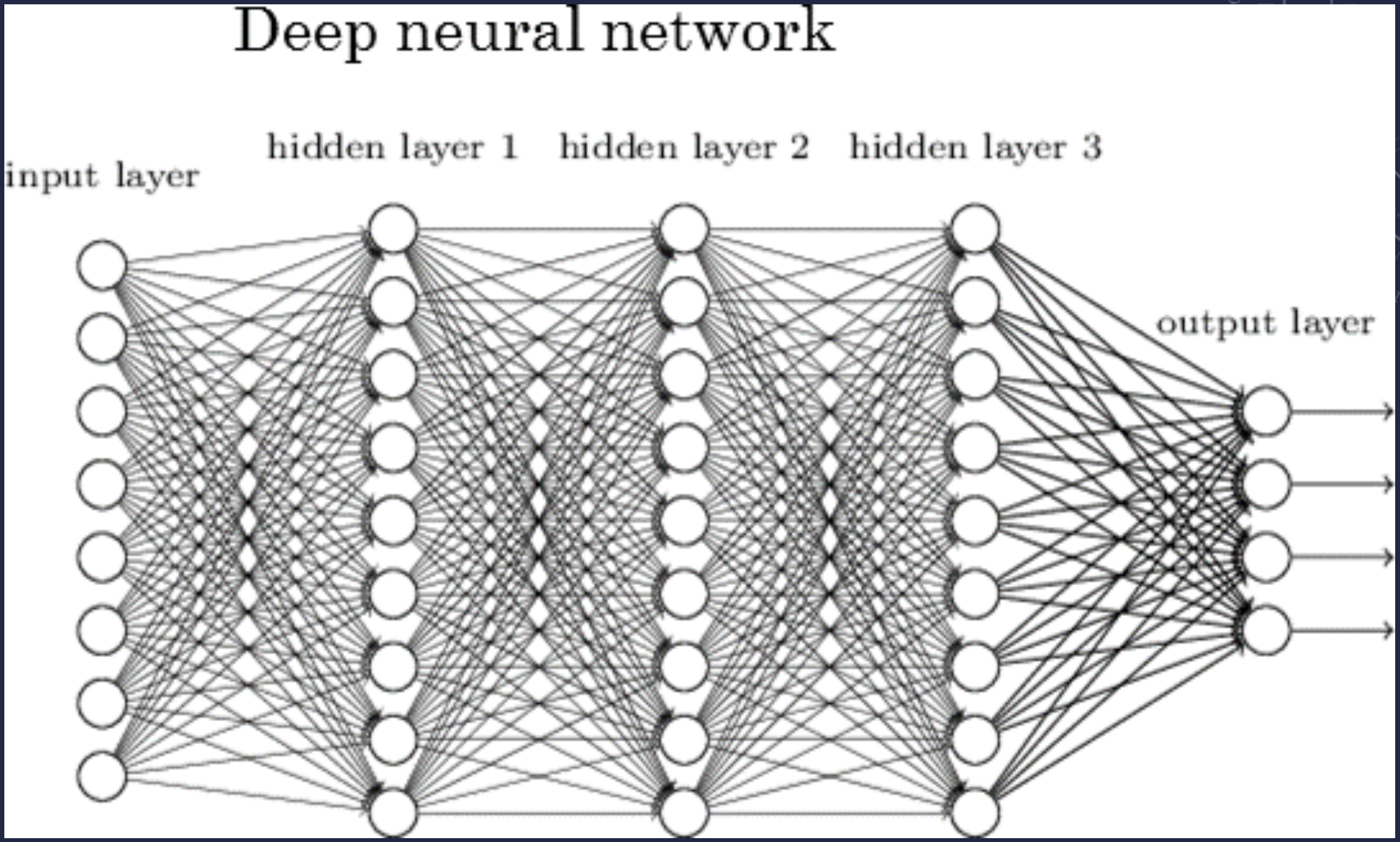
Table 5.8 : Truth table for XOR Gate

INPUTS		OUTPUTS
A	B	Y = A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

THE CODE

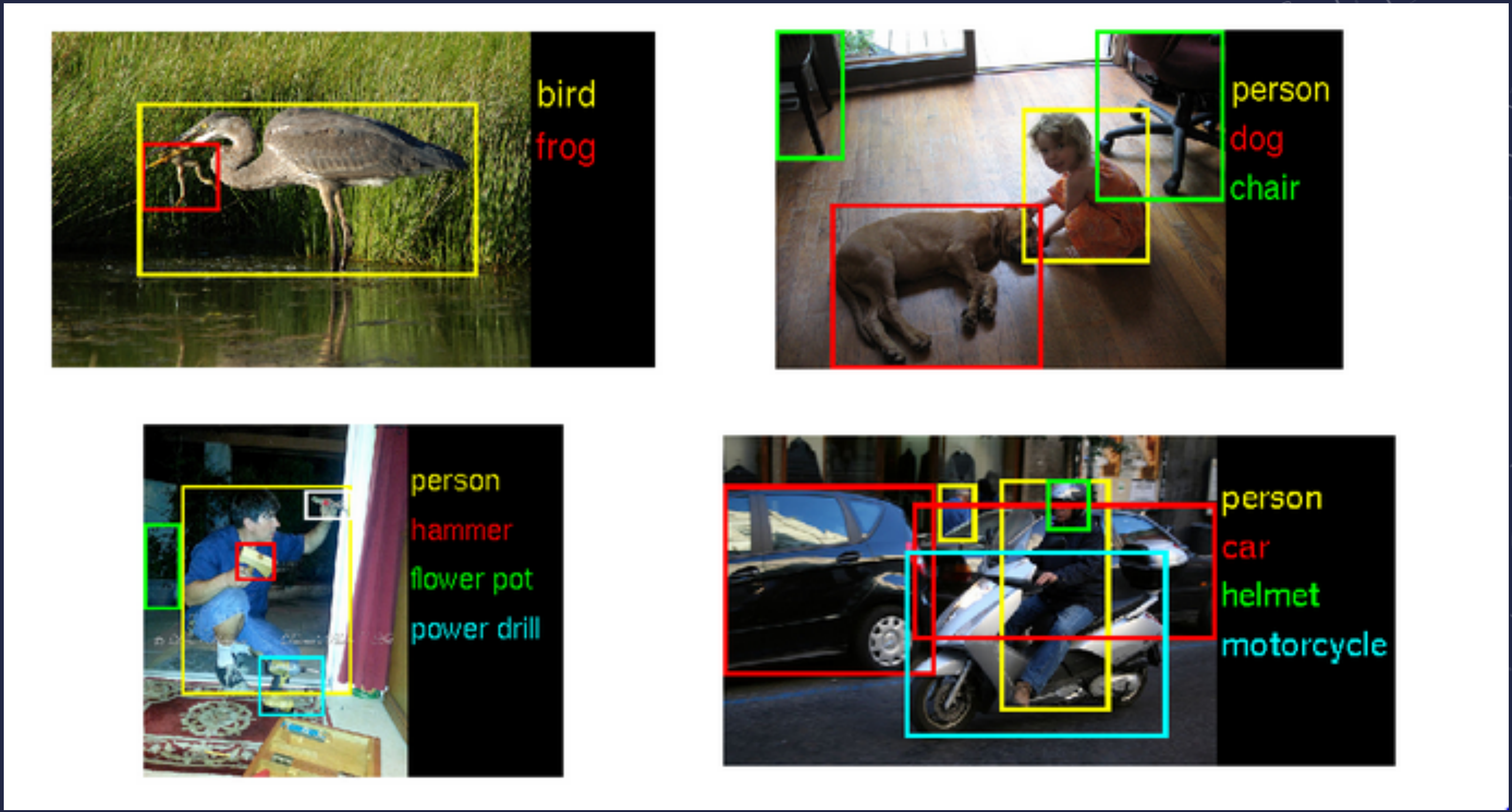


```python
from numpy import array, random, dot, exp

inputs = array([[0, 0], [0, 1], [1, 0], [1, 1]])
outputs = array([[0], [1], [1], [0]])

def learn(X, y):
    l1_w = 2 * random.random((X.shape[1], 16)) - 1
    l2_w = 2 * random.random((16, 1)) - 1
    for j in xrange(10000):
        l1 = 1 / (1 + exp(-(dot(X, l1_w))))
        l2 = 1 / (1 + exp(-(dot(l1, l2_w))))
        l2_delta = (y - l2) * (l2 * (1 - l2))
        l1_delta = l2_delta.dot(l2_w.T) * (l1 * (1-l1))
        l2_w += l1.T.dot(l2_delta)
        l1_w += X.T.dot(l1_delta)
    return (l1_w, l2_w)

xor_weights = learn(inputs, outputs)

def predict(X, weights):
    l1 = 1/(1+exp(-(dot(X, weights[0]))))
    l2 = 1/(1+exp(-(dot(l1, weights[1]))))
    return l2

test_set = [[0, 0], [0, 1], [1, 0], [1, 1]]
for test_item in test_set:
    xor_prediction = predict(test_item, xor_weights)
    print str(test_item)+"\t"+str(xor_prediction)
```



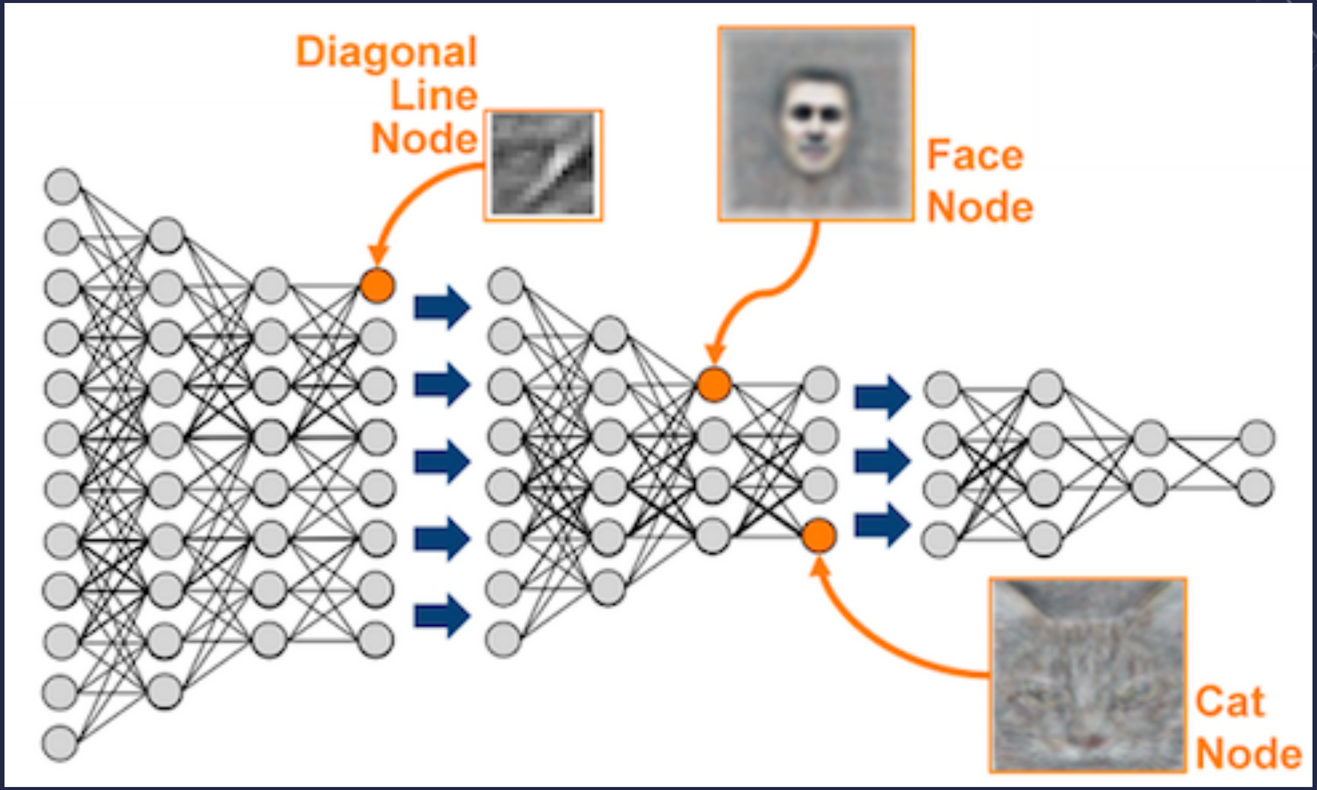
FullContact

DEEP LEARNING



FullContact

DEEP LEARNING



FullContact

WAY FORWARD

Play with Neural networks in the browser

http://playground.tensorflow.org/

Comprehensive list of resources available online on Deep Learning

https://github.com/ChristosChristofidis/awesome-deep-learning

FullContact

REFERENCE LINKS / IMAGE COURTESY

- https://medium.com/technology-invention-and-more/how-to-build-a-simple-neural-network-in-9-lines-of-python-code-cc8f23647ca1
- https://medium.com/technology-invention-and-more/how-to-build-a-multi-layered-neural-network-in-python-53ec3d1d326a#.dis92by0e
- http://iamtrask.github.io/2015/07/12/basic-python-network/
- http://iamtrask.github.io/2015/07/27/python-network-part2/
- http://sebastianraschka.com/faq/docs/logisticregr-neuralnet.html

FullContact

THANK YOU

// atm@infiniteloop.in

- Code - https://github.com/atmb4u/data-driven-code
- Slides – https://slideshare.net/atmb4u/data-driven-code

FullContact