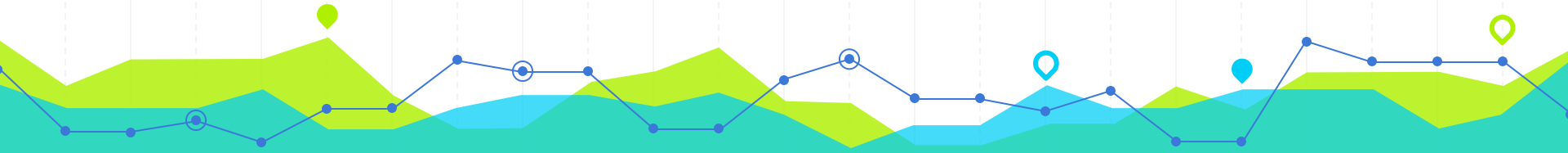


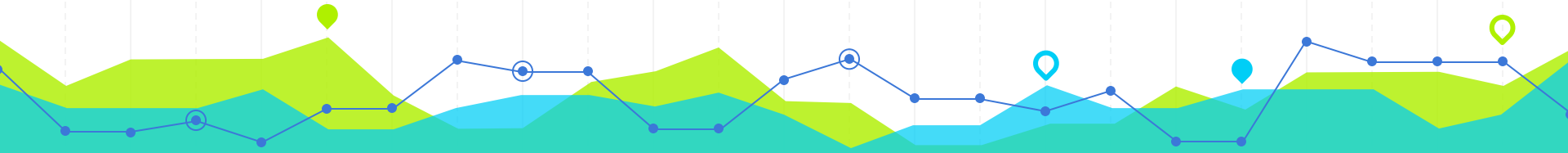
Machine Learning Crash Course

Week 3

“Some people call this artificial intelligence, but the reality is this technology will enhance us. So instead of artificial intelligence, I think we’ll augment our intelligence.”
~Ginni Rometty

Topics

- Introduction
  - Non-linear Hypothesis
  - Neurons and the brain
- Neural Networks
  - Representations
- Applications
  - Examples
- Code



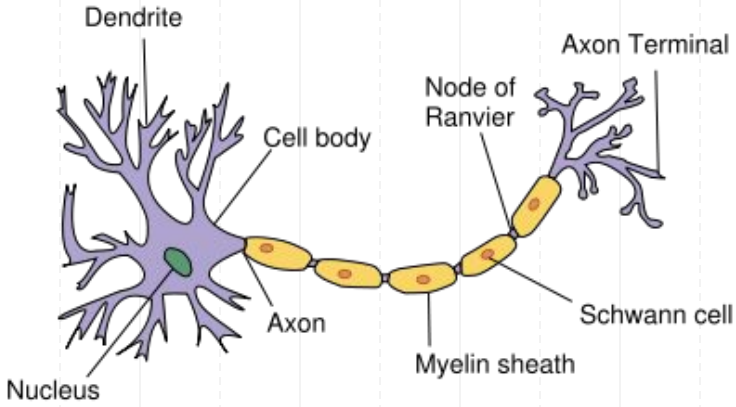
Introduction 1

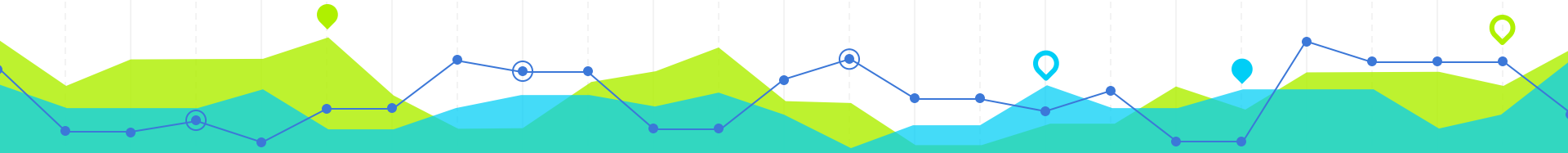
Non-linear Hypothesis

- More features at the cost of time
  - Ex: a 50 x 50 B&W image equates to approximately 3 million features
  - **DEMO**

Neurons and the Brain

- Create algorithms that mimic the brain
- Originally modelled in the 1940s
- Field finally gained traction in the 80s





Neural Networks 2

Representation 1

- Neuron model (Logistic Unit)
  - Input, Body, and Output (DEMO)
- Neural Network
  - Consider 3 layers (input / hidden / output) (DEMO)
  - Calculate the hypothesis of each node in the hidden layer (DEMO)

Representation 2

- Neural Network (Vectorized Implementation)
  - Consider 3 layers (input / hidden / output)
  - g is a sigmoid function that is applied element-wise to z
  - Calculating H(x) is called "*forward propagation*"
    - For each layer (begin:end):

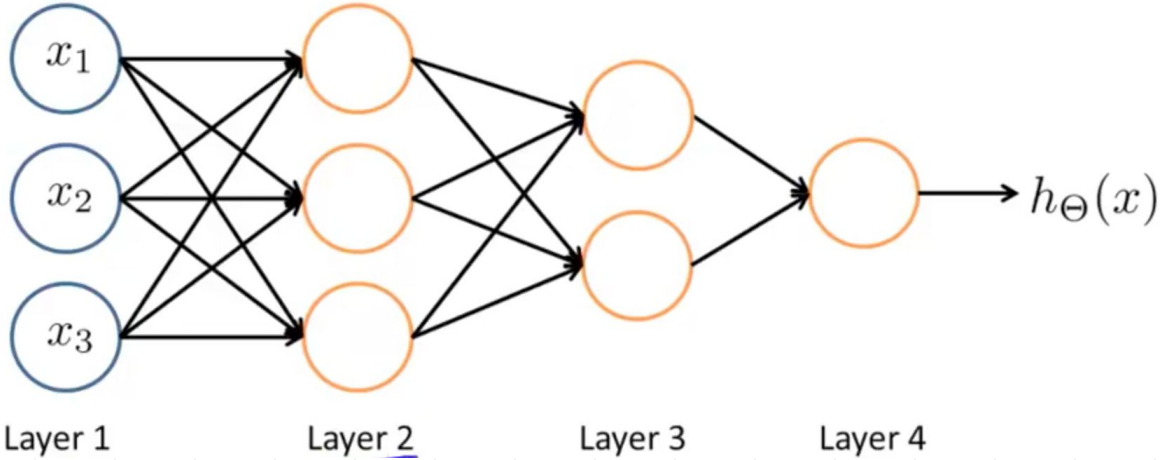
      Calculate the activation of the layer
  - DEMO

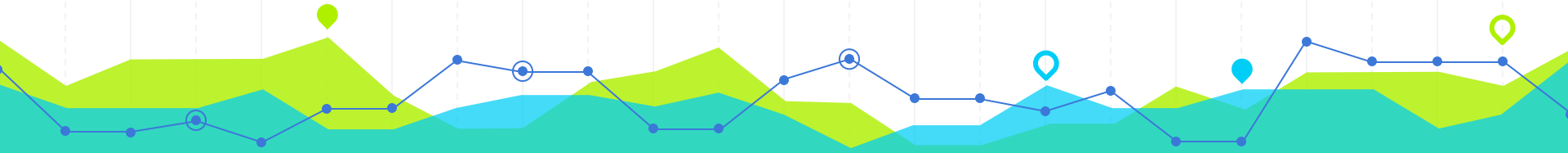
Interesting Relation

Recall Logistic Regression

Alternate Ideas

- Adding multiple hidden layers?!





Applications 2

Logical Operations

- AND function (DEMO)
- OR function (DEMO)
- NOT function (DEMO)
- XOR function (DEMO)

Multi-class Classification

We want to get a 10 x 1 vector for h_theta(X):

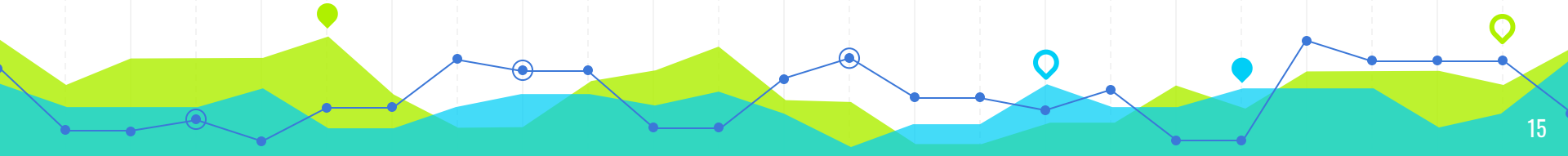
- Create 10 logistic classifiers in the output layer
- CODE EXAMPLE & DEMO (dependencies instruction and code)
  - https://medium.com/@kzeller_133/a-convolutional-neural-network-implementation-with-tensorflow-ad23b8cc0691
  - https://github.com/TensorFlow-ML-Architectures/TensorFlow-Multilayer-Perceptron

THANKS!

Any questions?

You can find me at

http://cs.oswego.edu/~kzeller



Sources

- https://www.ritchieng.com/neural-networks-representation/
- https://simple.wikipedia.org/wiki/Neuron