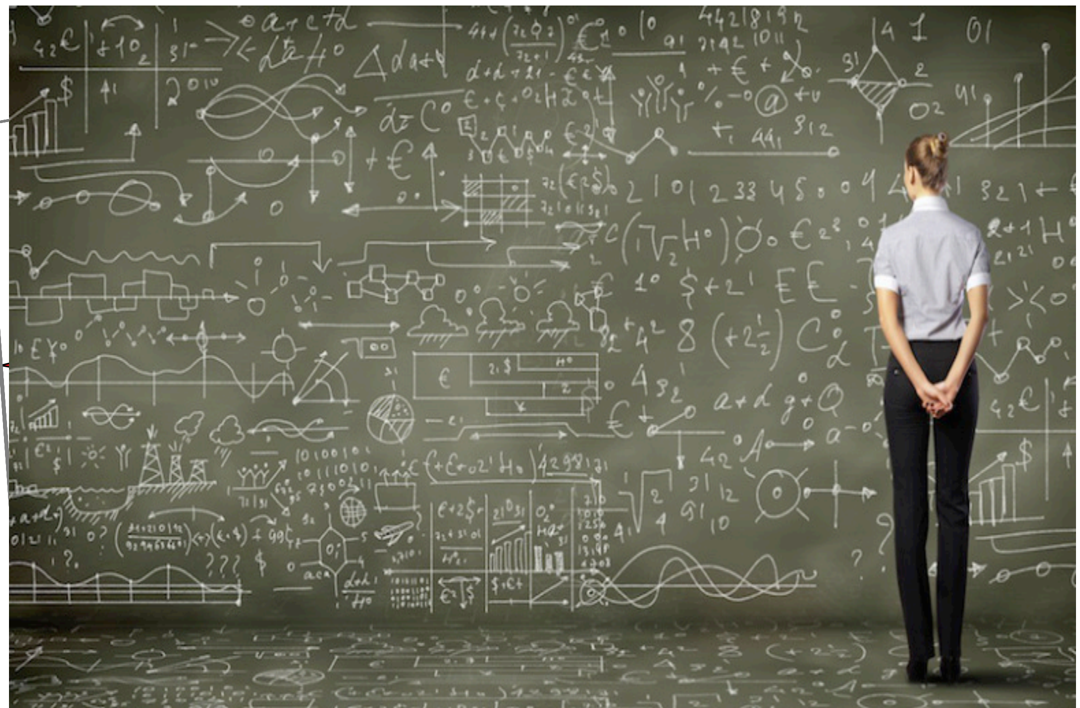
Recent Advances in Machine Learning
And Their Application to Networking



David Meyer
dmm@{brocade.com,uoregon.edu,1-4-5.net,..}

http://www.1-4-5.net/~dmm/talks/2015/hoti.pptx

23rd Annual Symposium on High-Performance Interconnects
26-27 Aug 2015
http://www.hoti.org/

Goals for this T

To take a look at t
of the art in M
give us a
wh to the
ur limited
t erstand how we
mi a network/automation
setti

While Machine Learning is really all about math, this talk attempts to go easy on all of that (just a little math…) ☺

Agenda

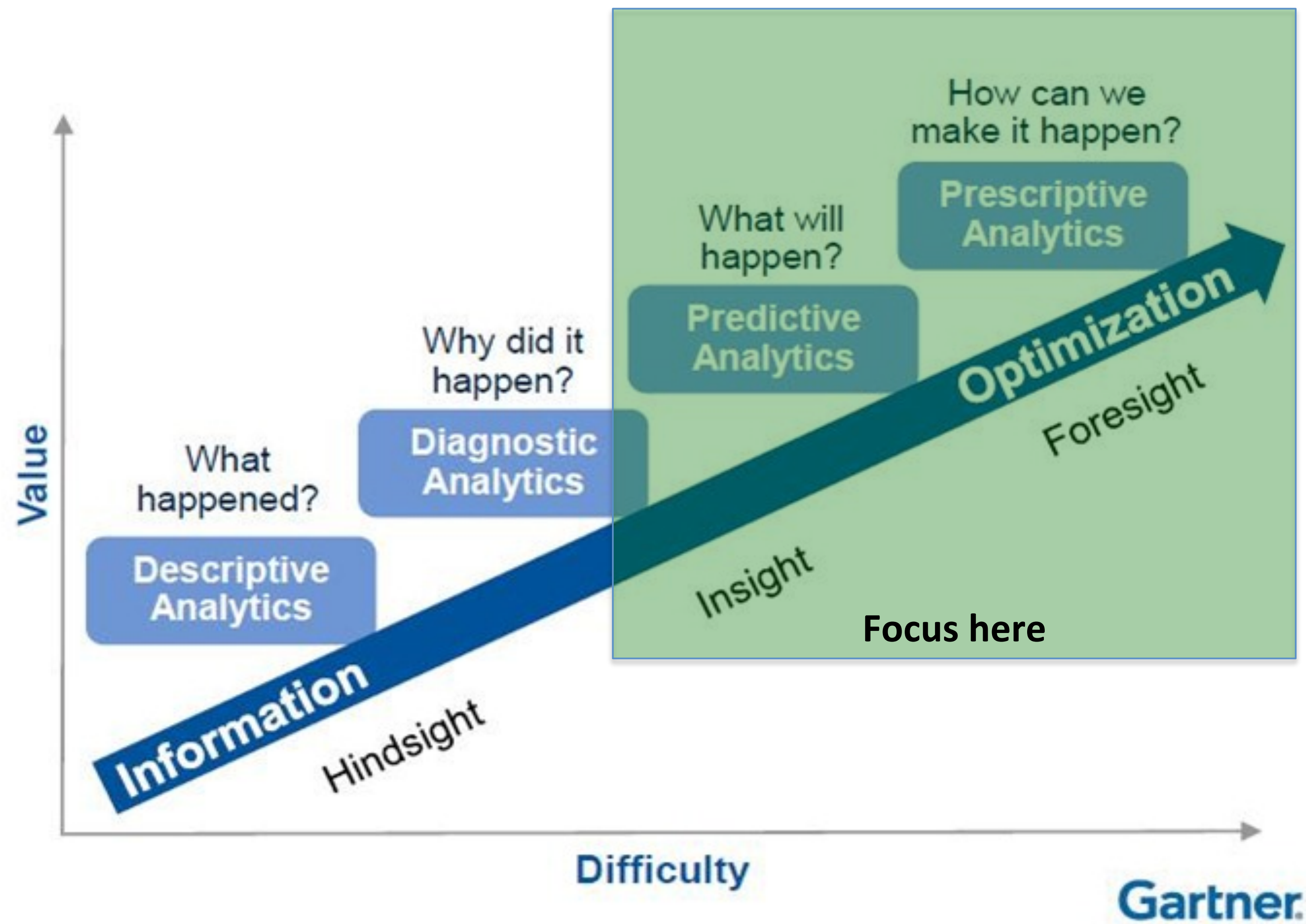
- What is all the (ML) excitement about?
- Very Briefly: What is ML and why do we care?
- ML Tools for DevOPs
- What the Future Holds
- Q&A

What is all the Excitement About?
Context and Framing

Lots of excitement around "analytics" and machine learning

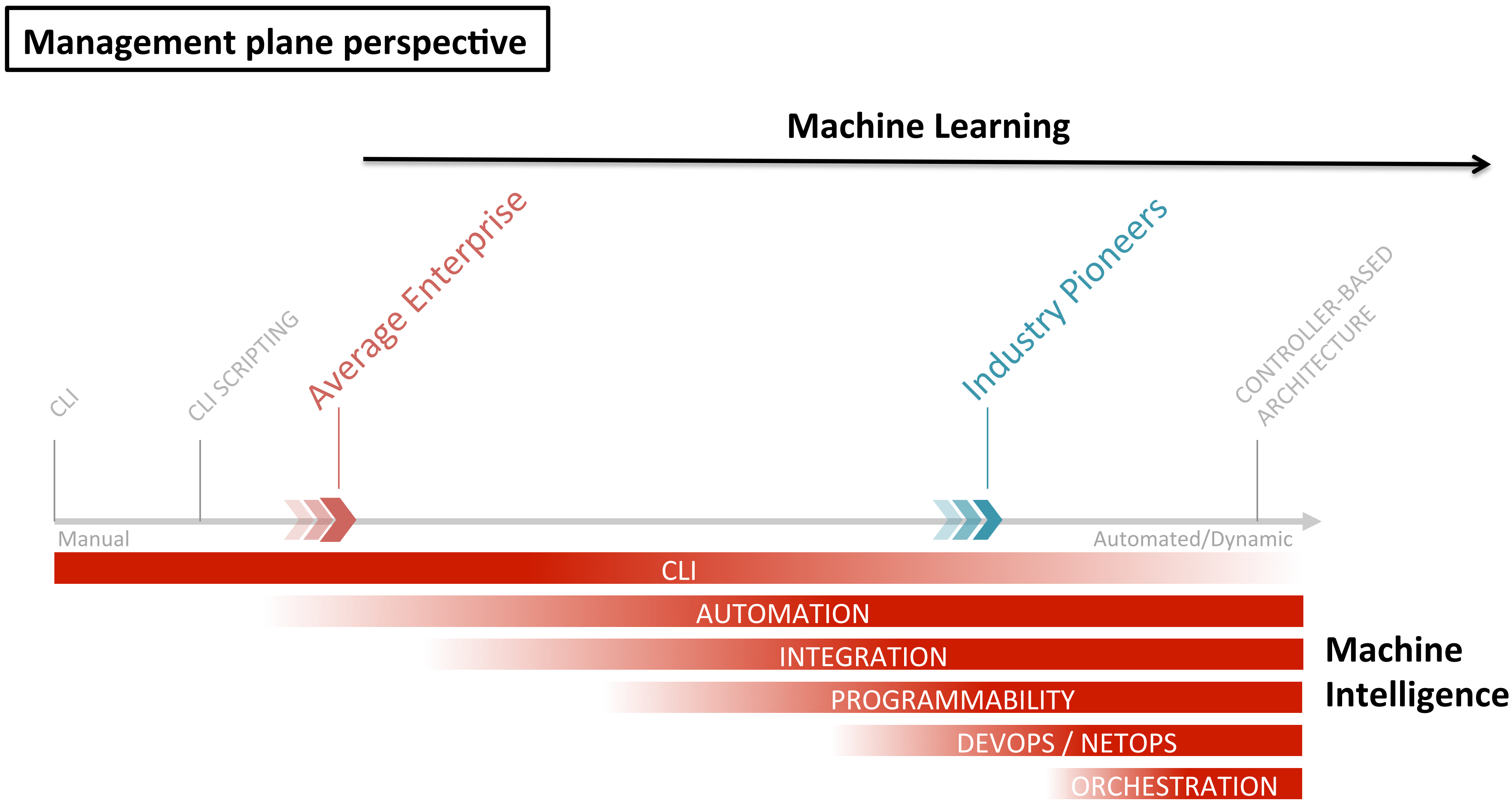
But what are "analytics"?

Conventional View of the Analytics Space



Another Way To Think About This

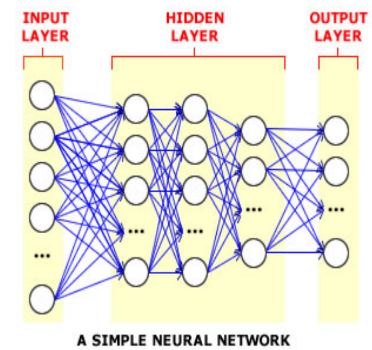
The Automation Continuum

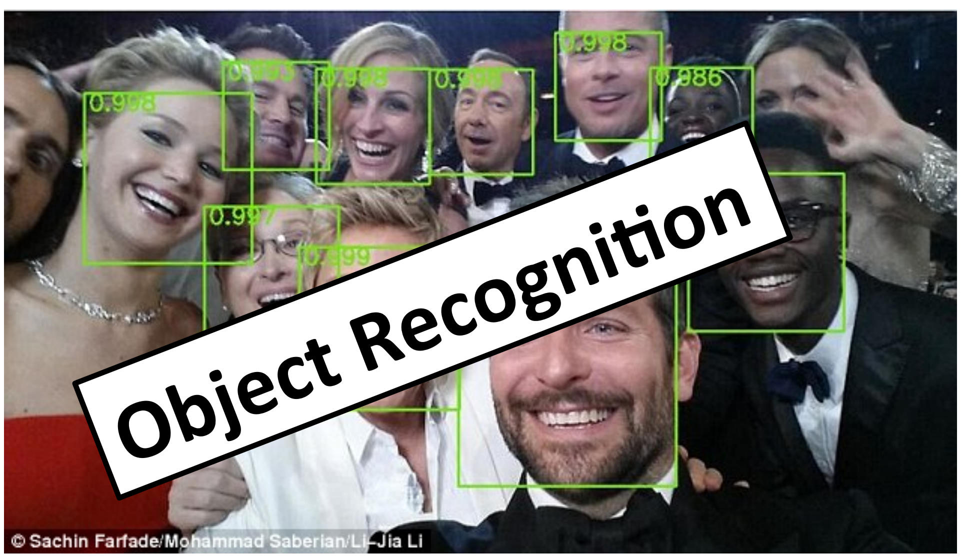


Original slide courtesy Mike Bushong and Joshua Soto

Ok, What is All the ML Excitement About?

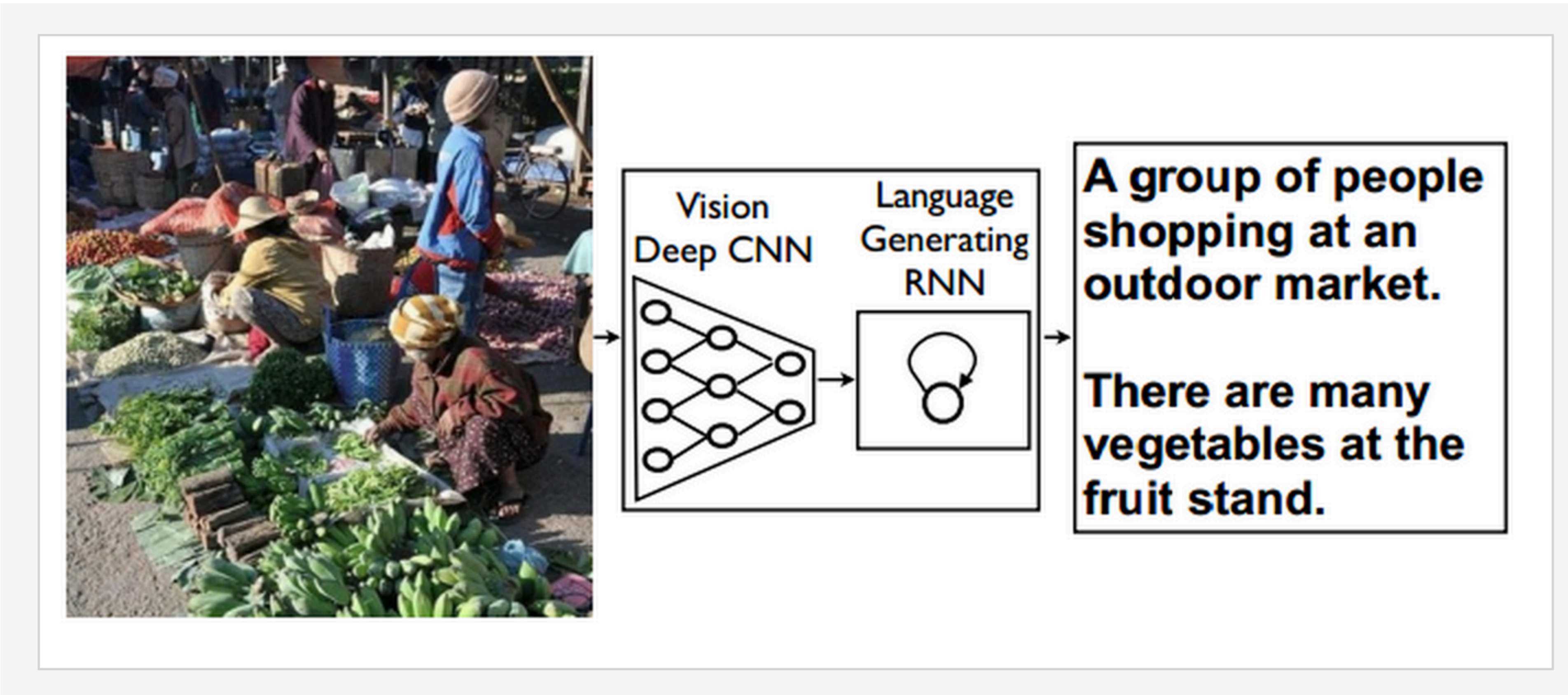
- *Deep learning* is enjoying great success in an ever expanding number of use cases
  - Multi-hidden layer neural networks
  - "Perceptual" tasks reaching super-human performance
  - Networking/non-cognitive domains still lagging
    - http://caia.swin.edu.au/urp/diffuse/papers.html (a bit older research)
    - Networking is a relatively new (but recently active) domain for ML





Why this is relevant: Network use cases will (eventually) use similar technologies

Auto-Captioning Cartoon
How it Works

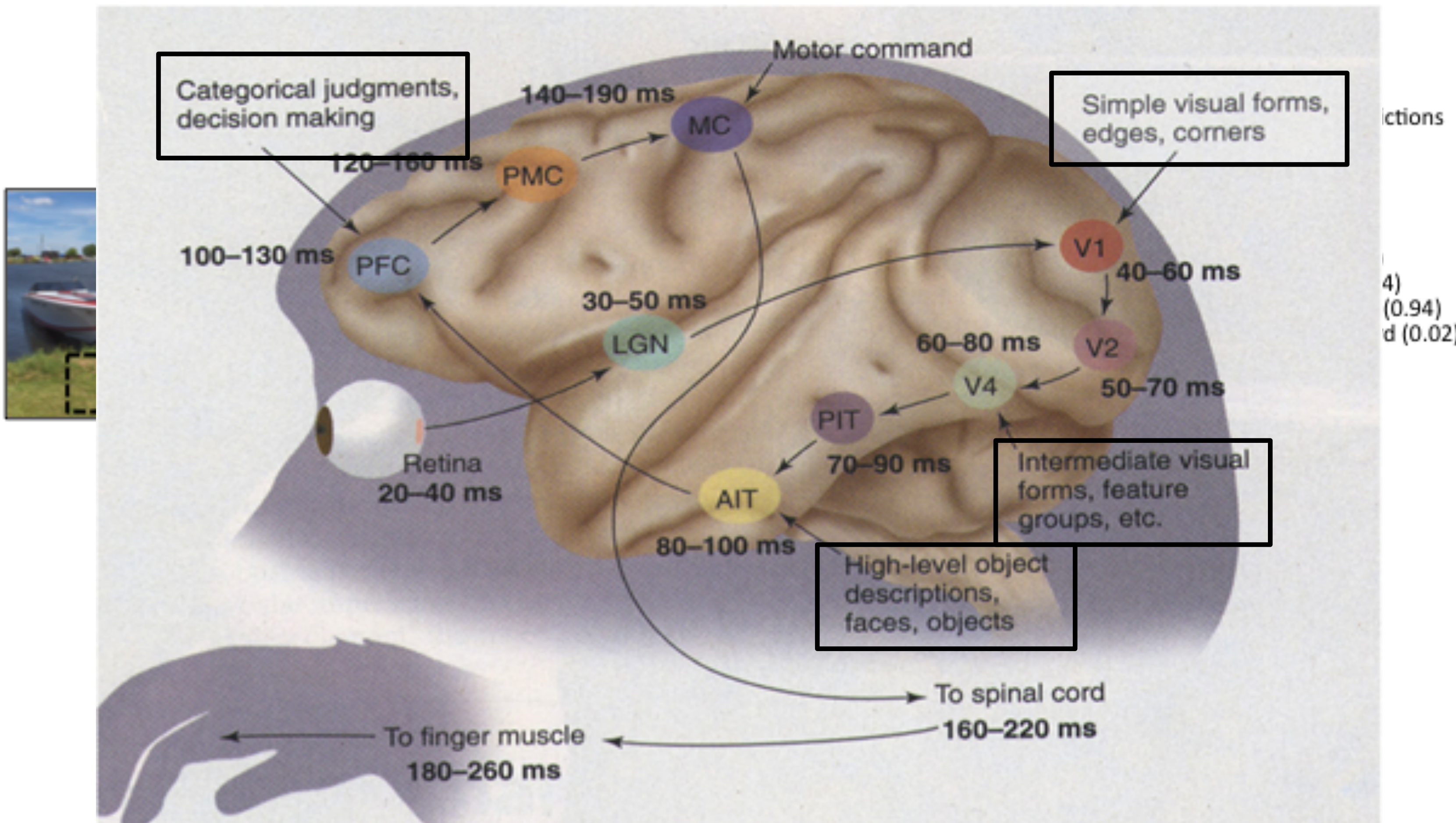


Self-Driving Cars

How Does Your Car Actually See?

How your car (camera...) sees

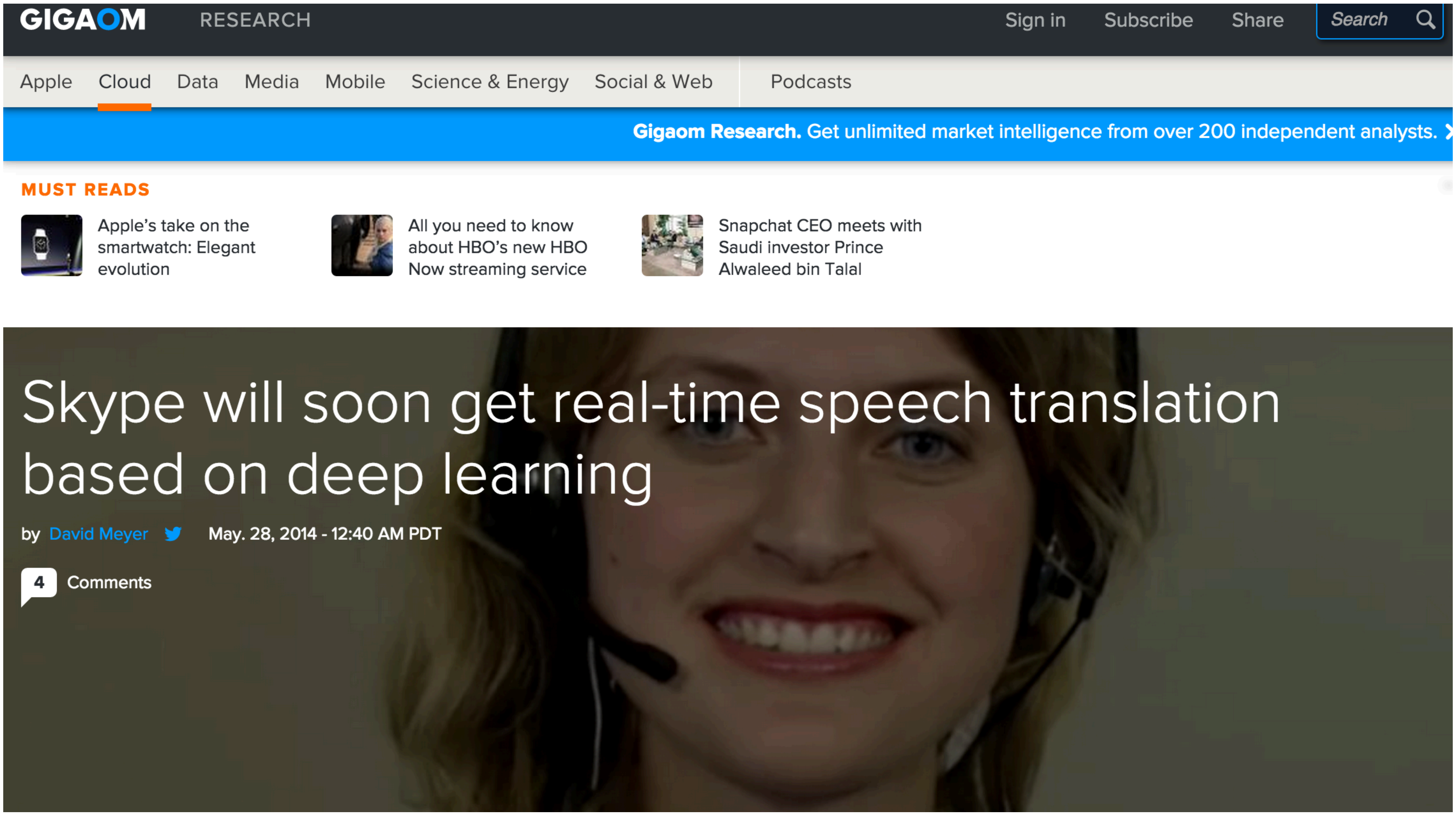
How your brain sees



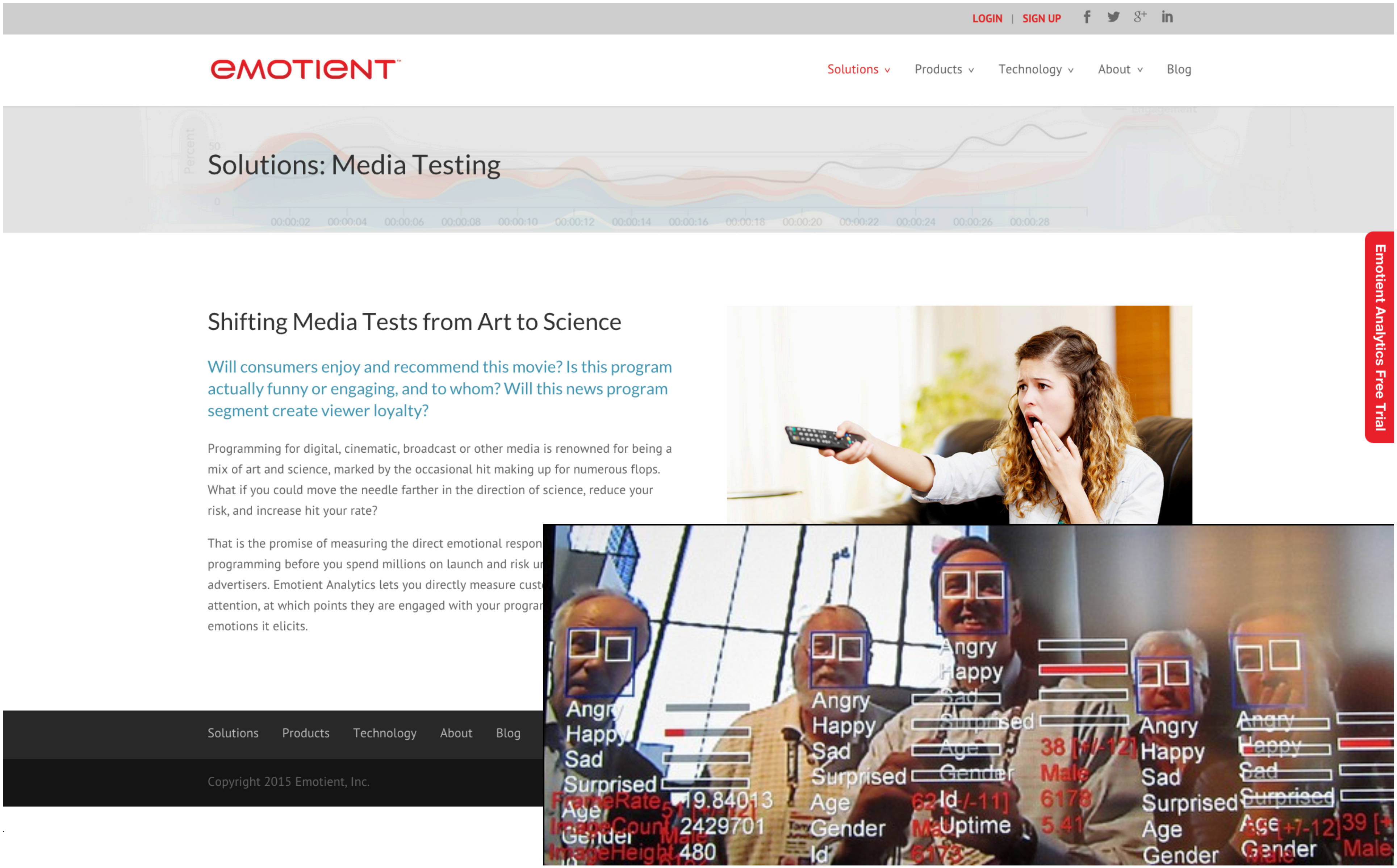
http://www.cns.nyu.edu/heegerlab/content/publications/Tolhurst-VisNeurosci1997a.pdf Slide courtesy Simon Thorpe

See http://www.wired.com/2015/05/wolframs-image-rec-site-reflects-enormous-shift-ai/

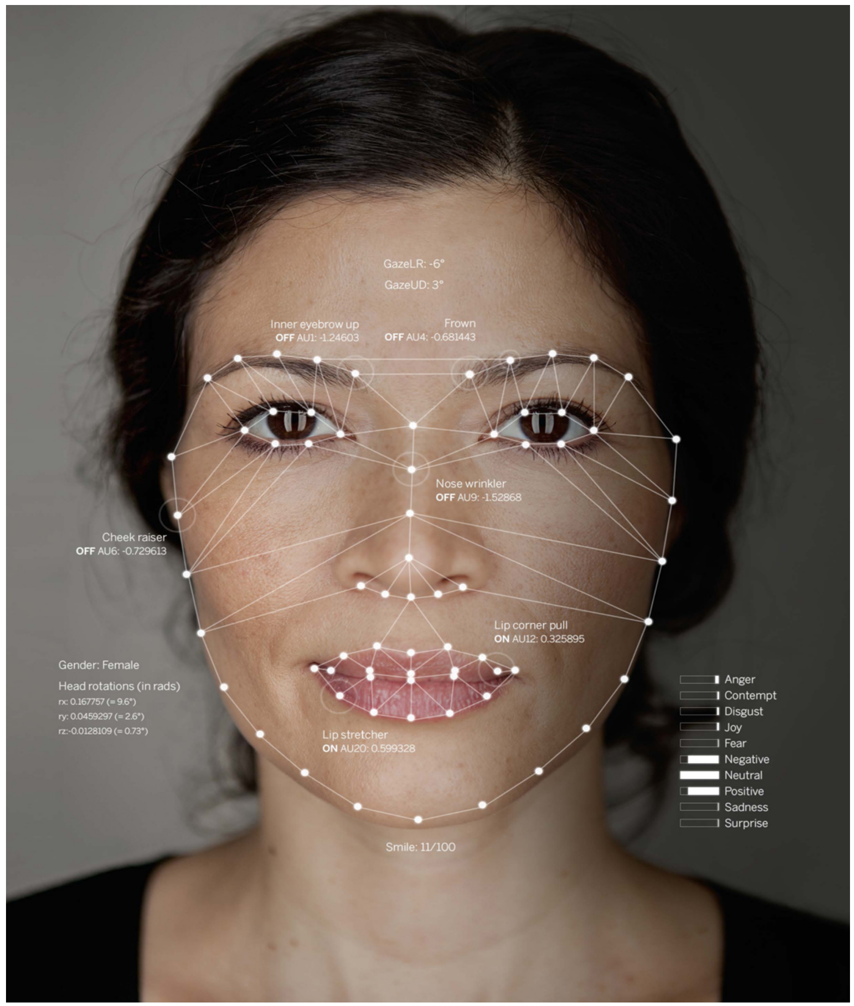
But There’s More



Think Speech/Object Recognition is Impressive?



So How Does This Work?
Facial Keypoints



Published by AAAS

Jelena Stajic et al. Science 2015;349:248-249

Everyone is getting into the game

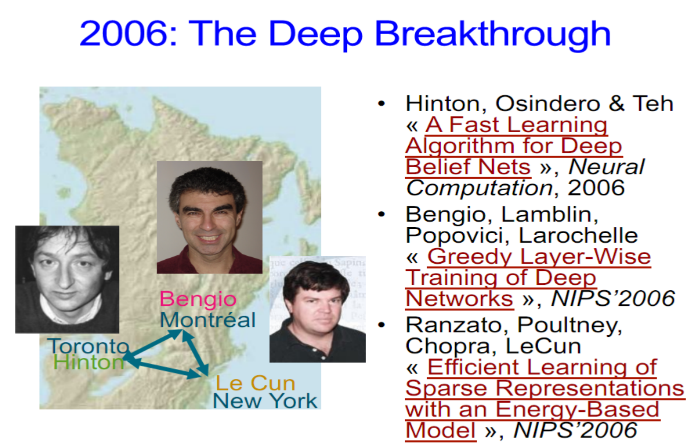
(M&A Gone Wild)

Why is this all happening now?

- **Before 2006 people thought deep neural networks couldn't be trained**
  - So why now?

- **Theoretical breakthroughs in 2006**
  - Learned how to train deep neural networks
    - Technically: Solved the vanishing/exploding gradient problem(s) ("butterfly effects")
  - More recently: http://www.cs.toronto.edu/~fritz/absps/momentum.pdf
  - Nice overview of LBH DL journey: http://chronicle.com/article/The-Believers/190147/

- **Compute**
  - CPUs were 2^20s of times too slow
  - Parallel processing/algorithms
  - GPUs + OpenCL/CUDA

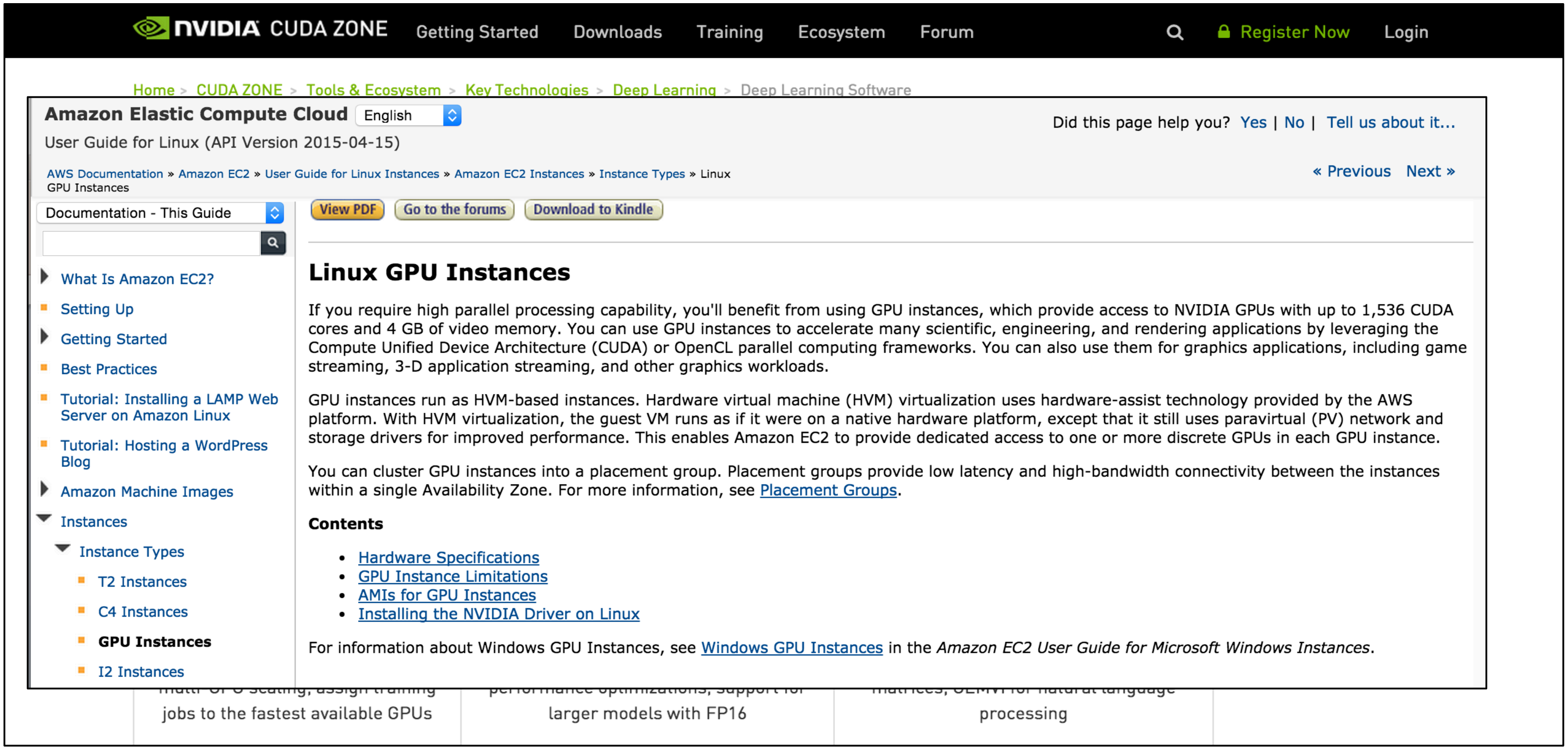


- **Datasets**
  - Massive data sets: Google, FB, Baidu, …
  - And the convergence of theory/practice in ML

- **Alternate view of history?**
  - LBH Nature DL review: http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html
  - Jürgen Schmidhuber's critique : http://people.idsia.ch/~juergen/deep-learning-conspiracy.html
  - LBH rebuttal: http://recode.net/2015/07/15/ai-conspiracy-the-scientists-behind-deep-learning/

Image courtesy Yoshua Bengio

Aside: GPUs



- CUDA/OpenCL support built into most open source ML frameworks
  - http://scikit-learn.org
  - http://torch.ch/
  - http://caffe.berkeleyvision.org/
  - http://apollo.deepmatter.io/
  - …

BTW, the ML community has a strong and long standing open{source,data,model} tradition/culture #openscience

Ok, But What About Networking?

(from NANOG 64)

document - wednesday_general_szarecki_telemetry.pptx 6/3/15, 3:46 PM

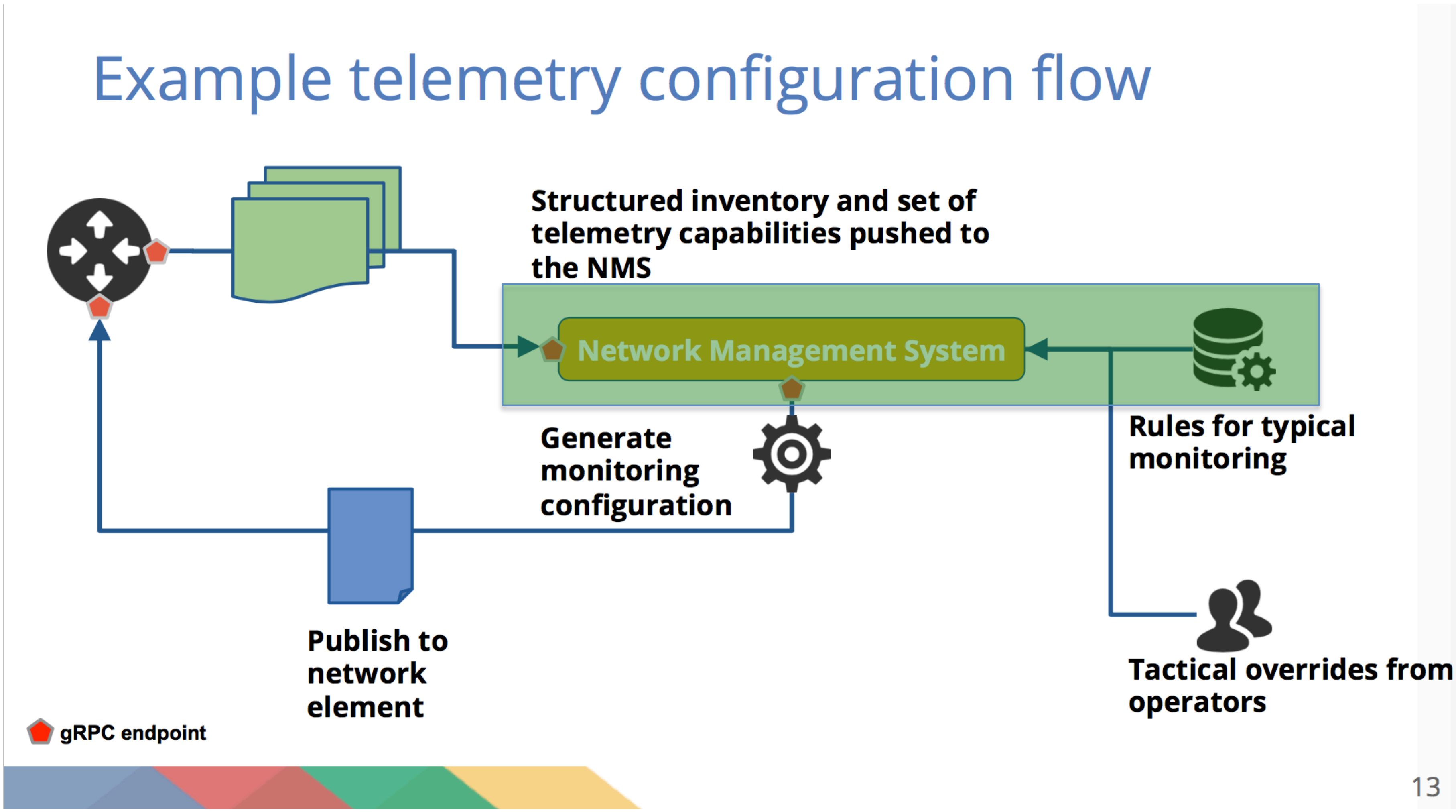
Next Generation Telemetry

- Better suited for high fidelity event reporting.
  - Traffic Micro-bursts
  - Short-living states e.g. Signaling/routing messages peaks
  - In run-time
- Feeds Machine Learning [ML] network analytics
  - We do not know about network what we do not know.
  - Enables AI/ML/BigData analytics to locate patterns and relationships the operators haven't yet discovered. This requires a lot of data at high fidelity.
- Enables optimization:
  - usage-based provisioning and configuration.
  - Trend-based behavior prediction and proactive actions.

High-Fidelity, Run-Time statistics and events collection

See https://www.nanog.org/sites/default/files//meetings/NANOG64/1023/20150603_Szarecki_Architecture_For_Fine-Grain___v2.pdf

More from NANOG 64



See https://www.nanog.org/sites/default/files//meetings/NANOG64/1011/20150604_George_Sdn_In_The_v1.pdf

IRTF Current Events

http://trac.tools.ietf.org/group/irtf/trac/wiki/nml

Networks Machine Learning (NML) RG (Proposed)

Charter

Machine learning technologies can learn from historical data, and make predictions or decisions, rather than following strictly static program instructions. They can dynamically adapt to a changing situation and enhance their own intelligence with by learning from new data. This approach has been successful in image analysis, pattern recognition, language recognition, conversation simulation, and many other applications. It can learn and complete complicated tasks. It also has potential in the network technology area. It can be used to intelligently learn the various environments of networks and react to dynamic situations better than a fixed algorithm. When it becomes mature, it would be greatly accelerate the development of autonomic networking.

The Network Machine Learning Research Group (NMLRG) provides a forum for researchers to explore the potential of machine learning technologies for networks. In particular, the NMLRG will work on potential approaches that apply machine learning technologies in network control, network management, and supplying network data for upper-layer applications.

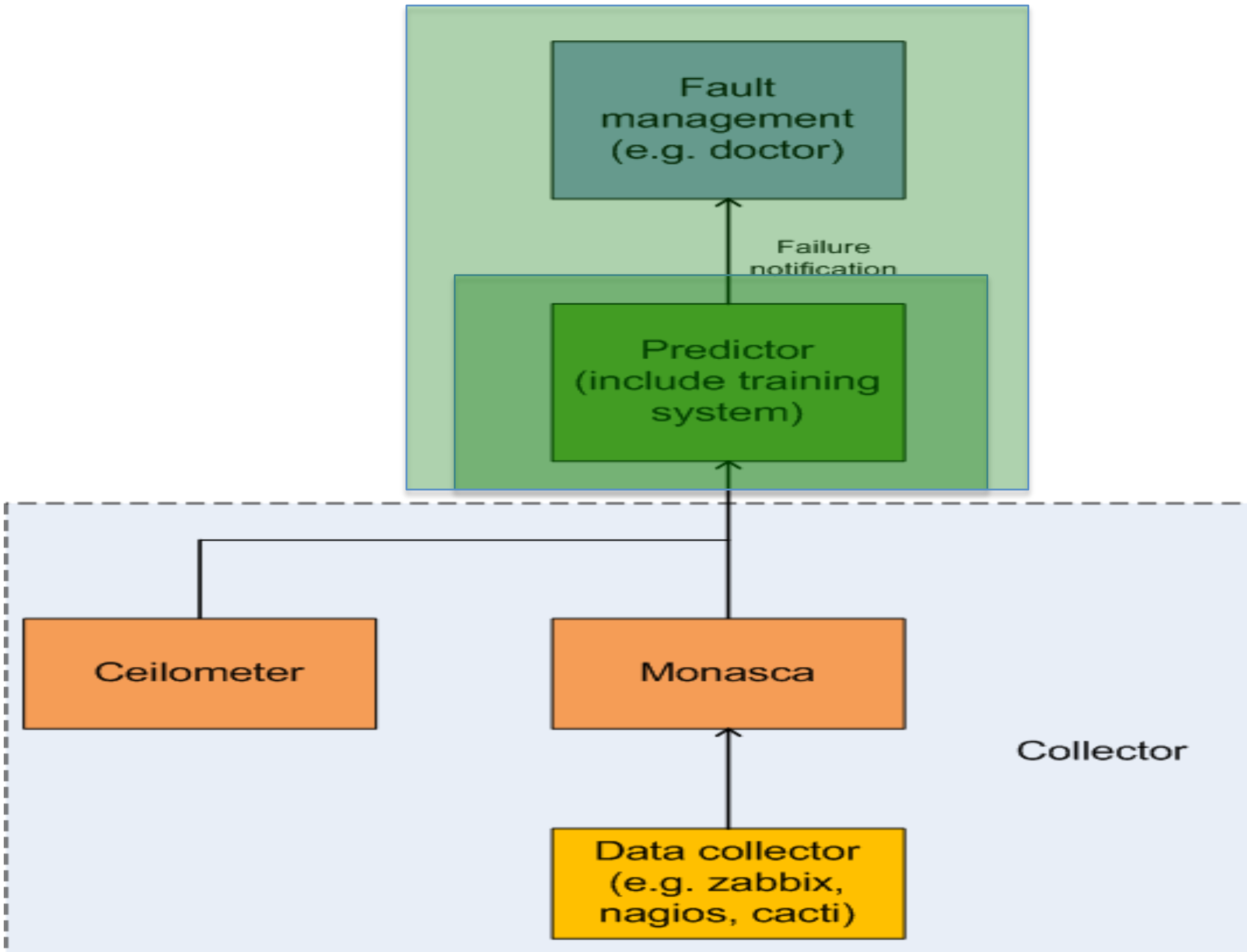
The initial focus of the NMLRG will be on higher-layer concepts where the machine learning mechanism could be applied in order to enhance the network establishing, controlling, managing, network applications and customer services. This includes mechanisms to acquire knowledge from the existing networks so that new networks can be established with minimum efforts; the potential to use machine learning mechanisms for routing control and optimization; using machine learning mechanisms in network management to predict future network status; using machine learning mechanisms to autonomic and dynamically manage the network; using machine learning mechanisms to analyze network faults and support recovery; learning network attacks and their behavior, so that protection mechanisms could be self-developed; unifying the data structure and the communication interface between network/network devices and customers, so that the upper-layer applications could easily obtain relevant network information, etc.

The NMLRG is expected to identify and document requirements, to survey possible approaches, to provide specifications for proposed solutions, and to prove concepts with prototype implementations that can be tested in real-world environments.

The group will report its progress through a publicly accessible web site and presentations at IETF meetings. Specifications developed by the NMLRG will be submitted for publication as Experimental or Informational RFCs.

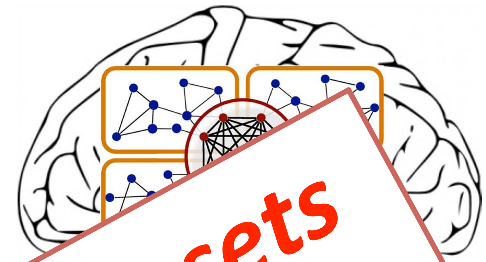
This topic is rapidly moving from academic research into practical application. Therefore we hope to attract participants from both university environments and industrial research and development organizations, in order to create synergy and convert theory into practice. People actively implementing relevant software will be especially welcome.

OPNFV



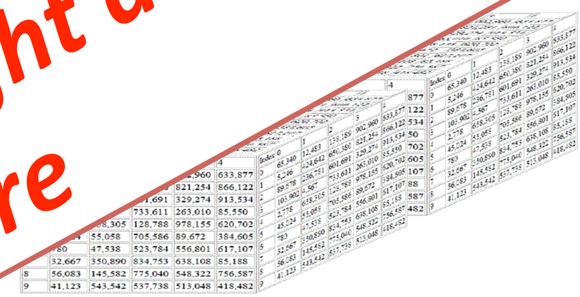
See https://wiki.opnfv.org/requirements_projects/data_collection_of_failure_prediction

OK, Now Imagine This…



First, envision the network as a huge sensor netw…

- Everything is a sensor (each counter, etc)
- Each sensor is a dimension
- This forms a *high-dimensional real-valued ve…*
- Note: Curse of dimensionality (more la…



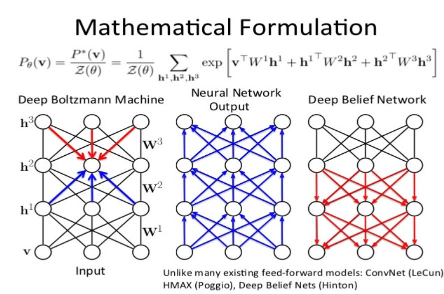
Guess what: This data…
learning with de…

Contrast M… *…stimation*[1]

No… capability:

Inte… …w-foo1 just flapped. This is going to cause cpu u… …gw-foo10 to spike and cause you to blackhole traffic … …C.D/16 with probability .85. The probability distribution is visualized at http://….

Well, guess what: With the right datasets we can do this and much more

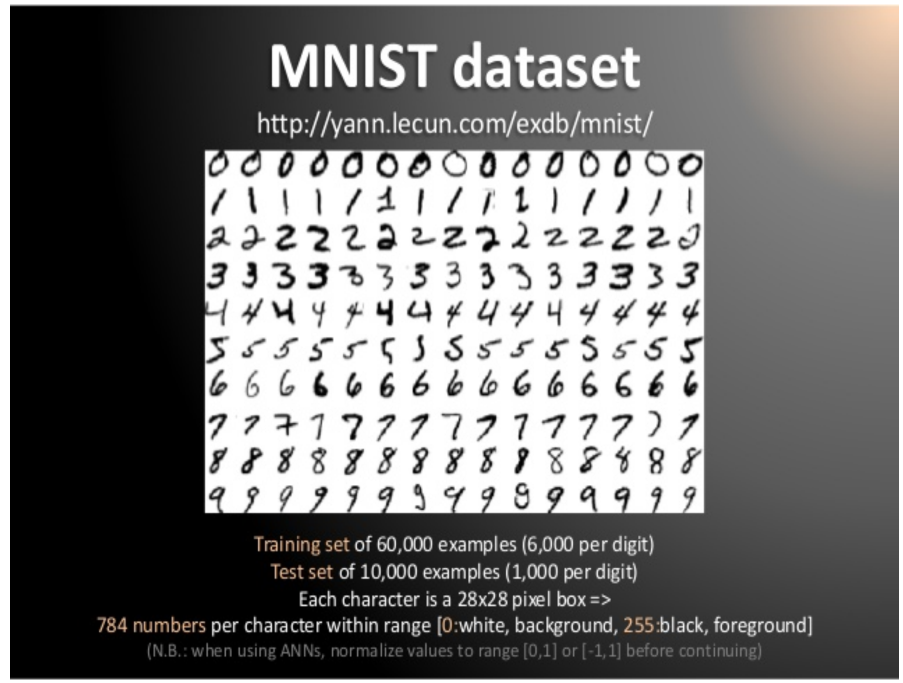


And consider the implications for the security space

1 http://www.iro.umontreal.ca/~lisa/pointeurs/TR1312.pdf

Aside: Dimensionality

- Machine Learning is good at understanding the structure of high dimensional spaces
- Humans aren't ☹
- What is a dimension?
  - Informally…
  - A direction in the input vector
  - "Feature"



- Example: MNIST dataset
  - Mixed NIST dataset
  - Large database of handwritten digits, 0-9
  - 28x28 images
  - 784 dimensional input data (in pixel space)

- Consider 4K TV → 4096x2160 = 8,847,360 dimensional pixel space

- But why care?

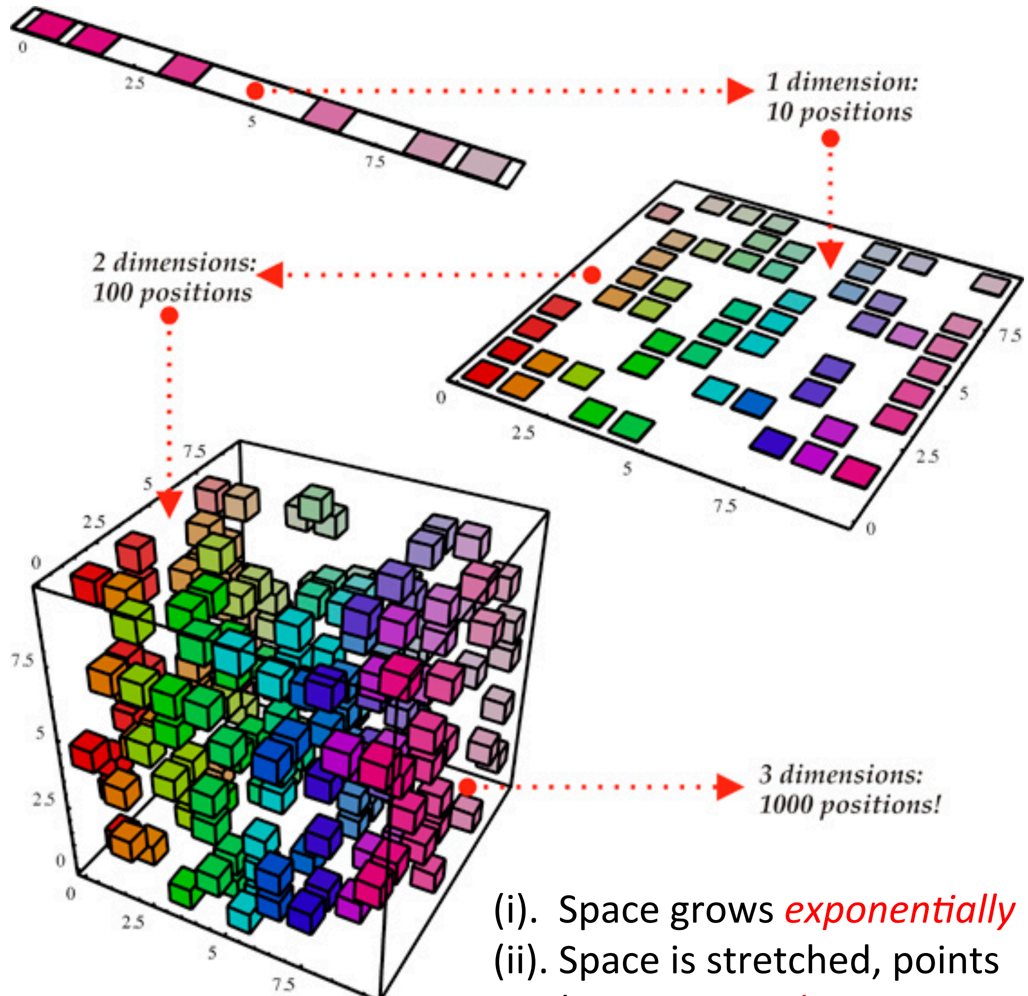
Because interesting and unseen relationships frequently live in high-dimensional spaces

But There's a Hitch

The Curse Of Dimensionality

- To generalize locally, you need representative examples from all relevant variations
    - But there are an *exponential* number of variations
    - So local representations might not (don't) scale

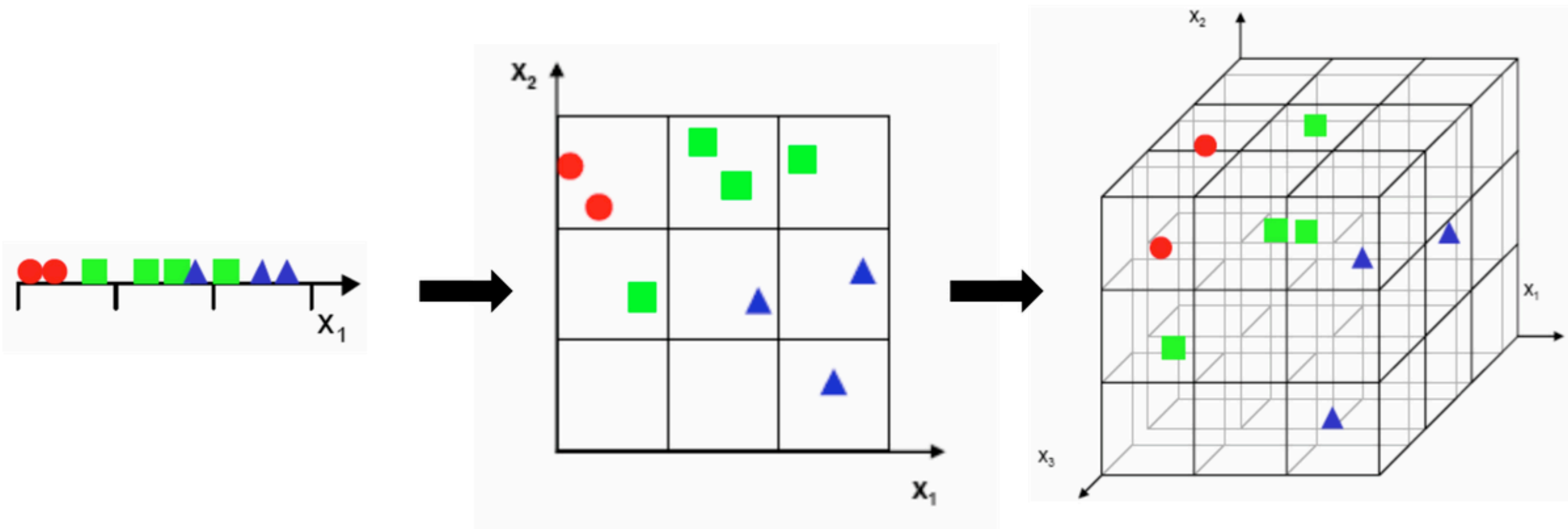
- Classical Solution: Hope for a smooth enough target function, or make it smooth by handcrafting good features or kernels. But this is sub-optimal. Alternatives?
    - Mechanical Turk (get more examples)
    - Deep learning
    - Distributed Representations
    - Unsupervised Learning
    - …



(i). Space grows *exponentially*
(ii). Space is stretched, points become *equidistant*

See also "Error, Dimensionality, and Predictability", Taleb, N. & Flaneur, https://dl.dropboxusercontent.com/u/50282823/Propagation.pdf for a different perspective

Seen Another Way



Sparsity becomes exponentially worse as dimension increases

Agenda

- ~~What is all the (ML) excitement about?~~
- Review: What is ML (and why do we care)?
- ML Tools for DevOPs
- What the Future Holds
- Q&A

All Cool, But What is Machine Learning?

The complexity in traditional computer programming is in the code (programs that people write). In machine learning, learning algorithms are in principle simple and the complexity (structure) is in the data. Is there a way that we can automatically learn that structure? That is what is at the heart of machine learning.

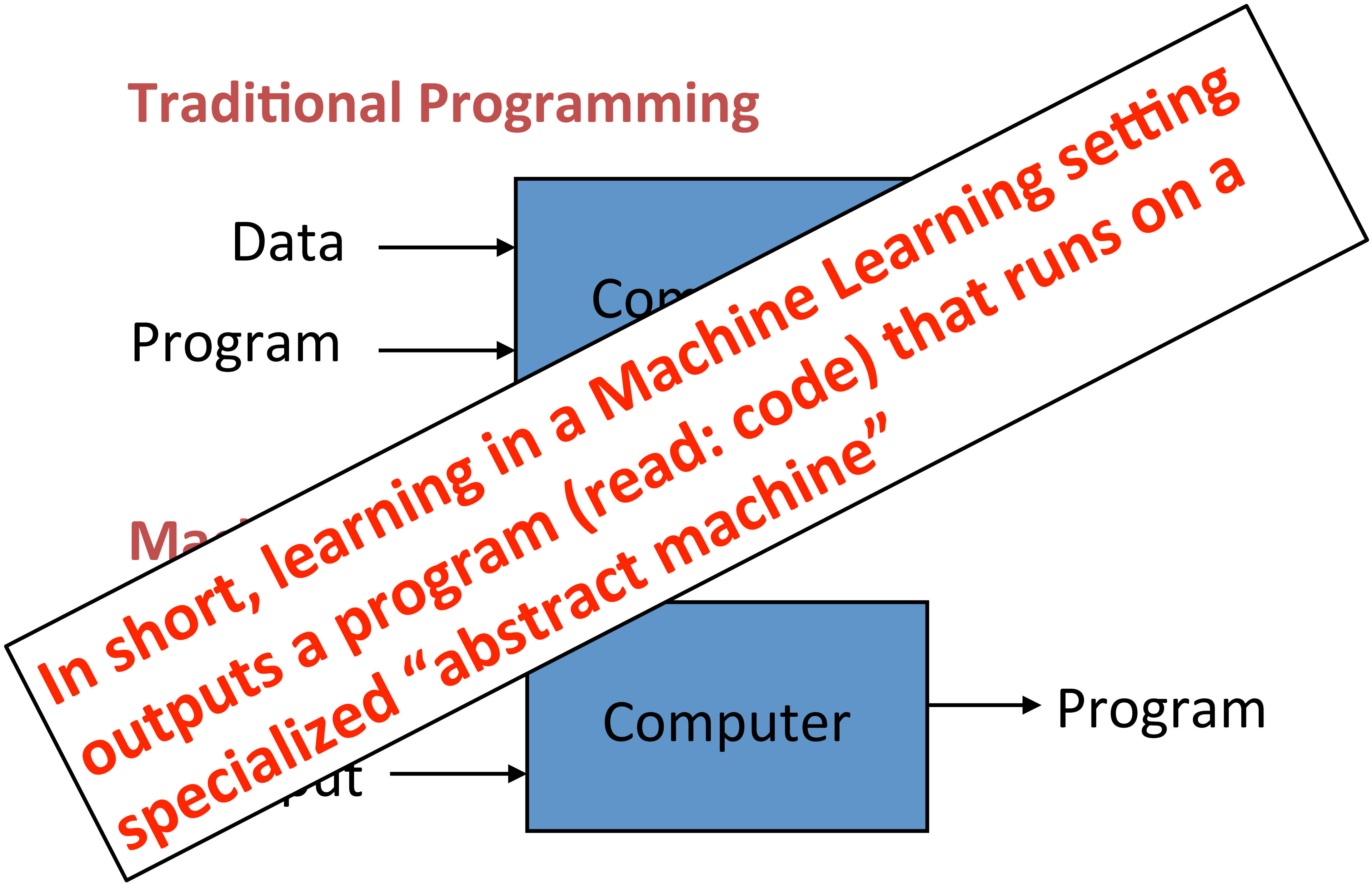
-- Andrew Ng

- Said another way, we want to discover the *Data Generating Distribution* (DGD) that underlies the data we observe. This is the function that we want to learn.

- Moreover, we care about primarily about the generalization accuracy of our model (function)
  - *Accuracy on examples we have not yet seen*
  - as opposed the accuracy on the training set (note: overfitting)

The Same Thing Said in Cartoon Form

Traditional Programming



In short, learning in a Machine Learning setting outputs a program (read: code) that runs on a specialized "abstract machine"

A Little More Detail

- Machine Learning is a procedure that consists of estimating model parameters so that the learned model can perform a specific task (sometimes called *Narrow* or *Weak* AI; contrast AGI)
  - Approach: Estimate model parameters (usually denoted $\boldsymbol{\theta}$) such that ***prediction error is minimized***
  - ***Empirical Risk Minimization*** casts learning as an *optimization problem*

- 3 Main Classes of Machine Learning Algorithms
  - Supervised
  - Unsupervised
  - Reinforcement learning
  - Semi-supervised learning

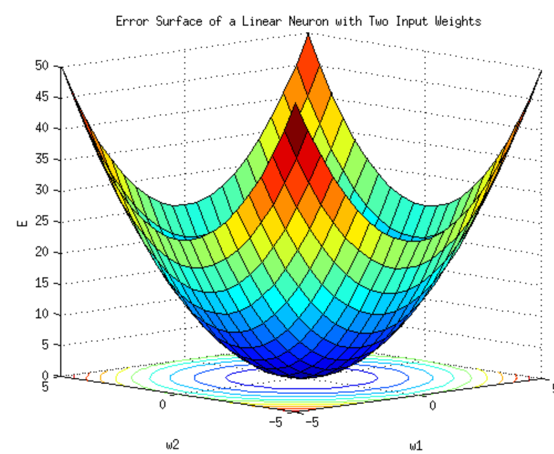
$$\arg\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_t l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)}) + \lambda \Omega(\boldsymbol{\theta})$$

- $l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)})$ is a loss function
- $\Omega(\boldsymbol{\theta})$ is a regularizer (penalizes certain values of $\boldsymbol{\theta}$)

- Supervised learning
  - Here we show the learning algorithm a set of examples ($\mathbf{x}_i$) and their corresponding outputs ($\mathbf{y}_i$)
    - You are given a training set $\{(\mathbf{x}_i,\mathbf{y}_i)\}$ where $\mathbf{y}_i = f(\mathbf{x}_i)$. We want to learn f
  - Essentially have a "teacher" that tells you what each training example is
  - See how closely the actual outputs match the desired ones
    - Note generalization error (bias, variance) vs. accuracy on the training set
  - *Most* of the big breakthroughs have come in supervised deep learning

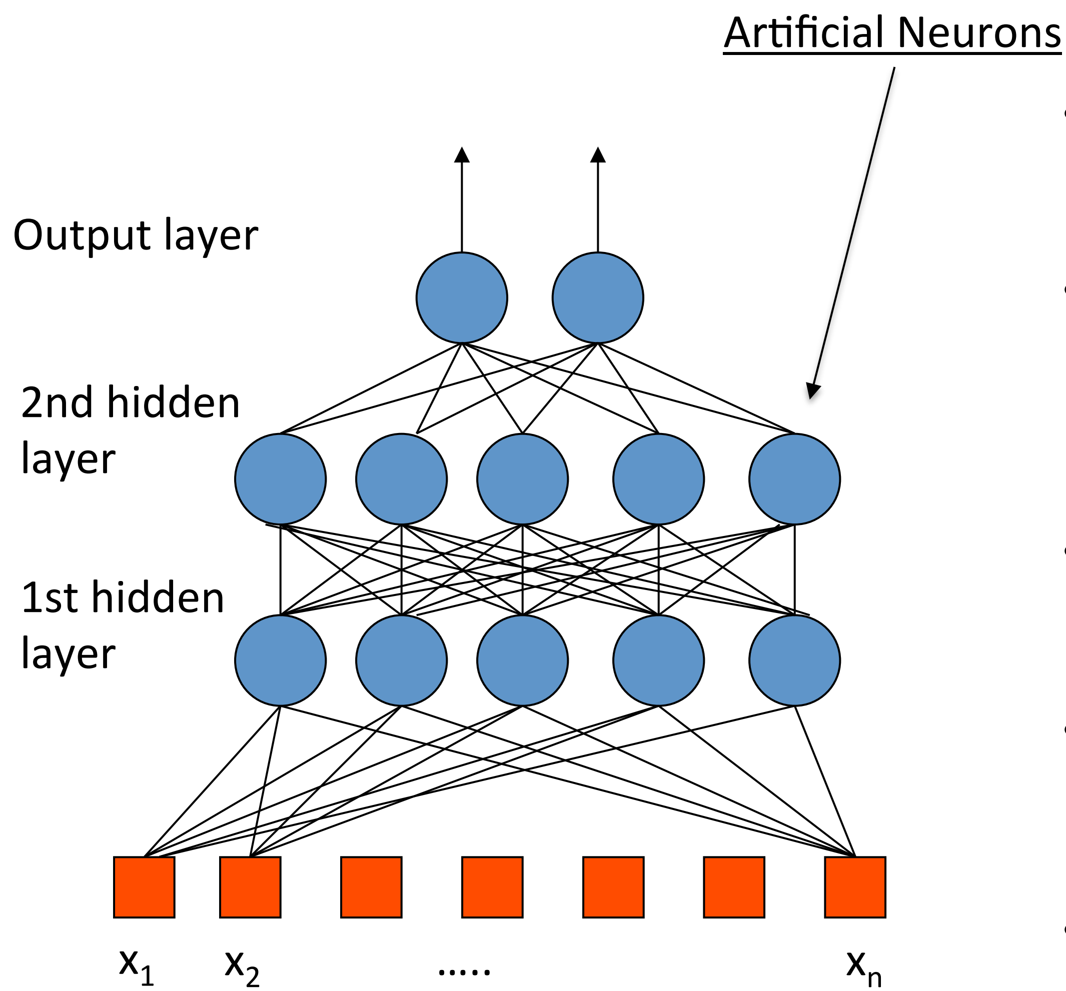
- Unsupervised Learning
  - Algorithm learns internal representations and important features
  - Unlabeled data sets

- Reinforcement Learning
  - Learning agent maximizes future reward
  - Dynamic system with feedback control
  - Robots



Images courtesy Hugo Larochelle and Andrew Ng

Aside: Feed Forward Neural Networks

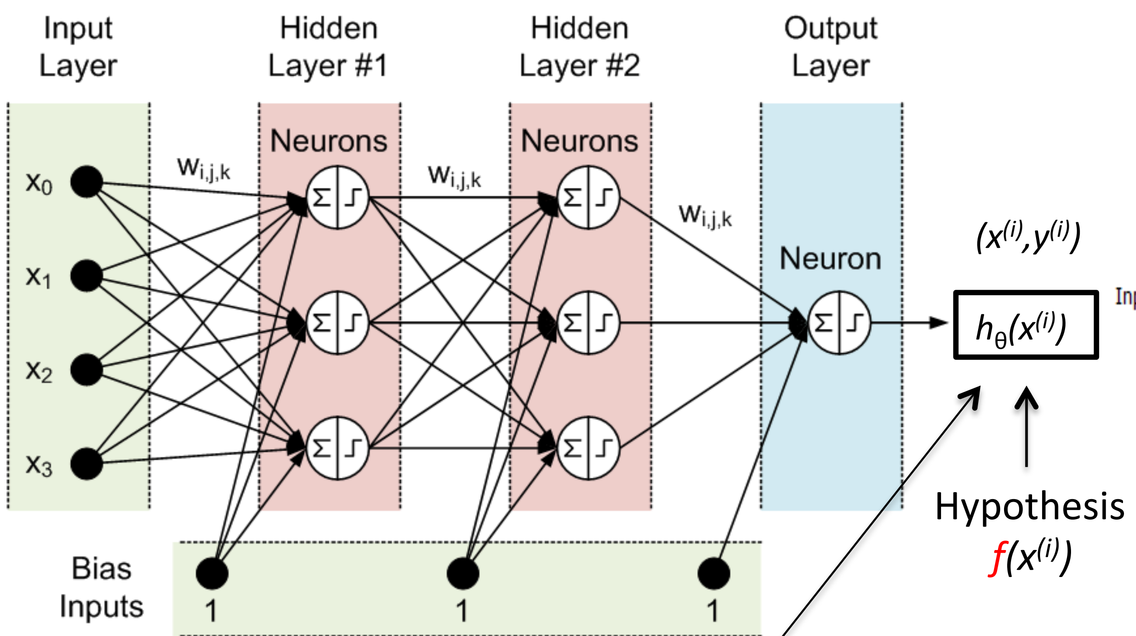


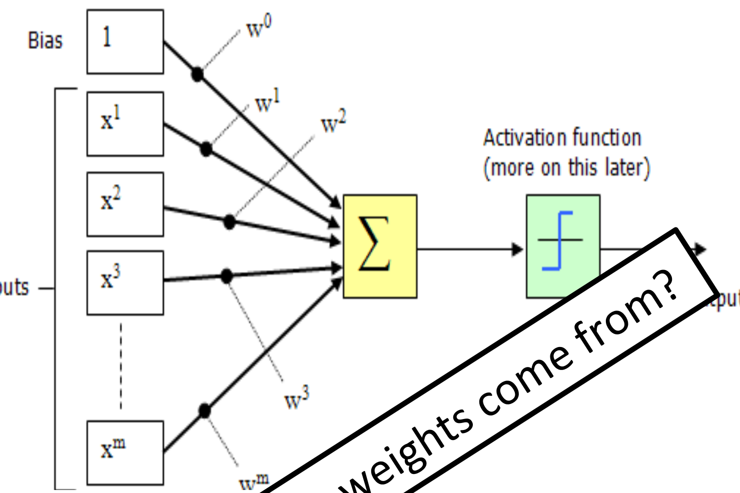
- The information is propagated from the inputs to the outputs
  - Directed Acyclic Graph (DAG)
- Computes one or more non-linear functions
  - Computation is carried out by composition of some number of algebraic functions implemented by the connections, weights and biases of the hidden and output layers
- Hidden layers compute intermediate representations
  - Dimension reduction
- Time has no role -- no cycles between outputs and inputs
  - However, in some models each hidden layer models one time step
- Maps vectors to vectors

We say that the input data, or features, are *n* dimensional

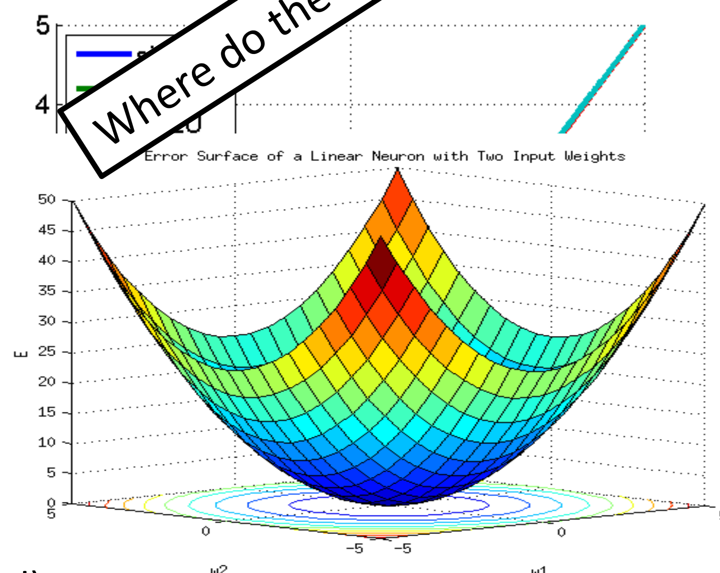
Deep Feed Forward Neural Nets

(most of the math I'm going to give you is on this slide ☺)





Where do the weights come from?



Forward Propagation →

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

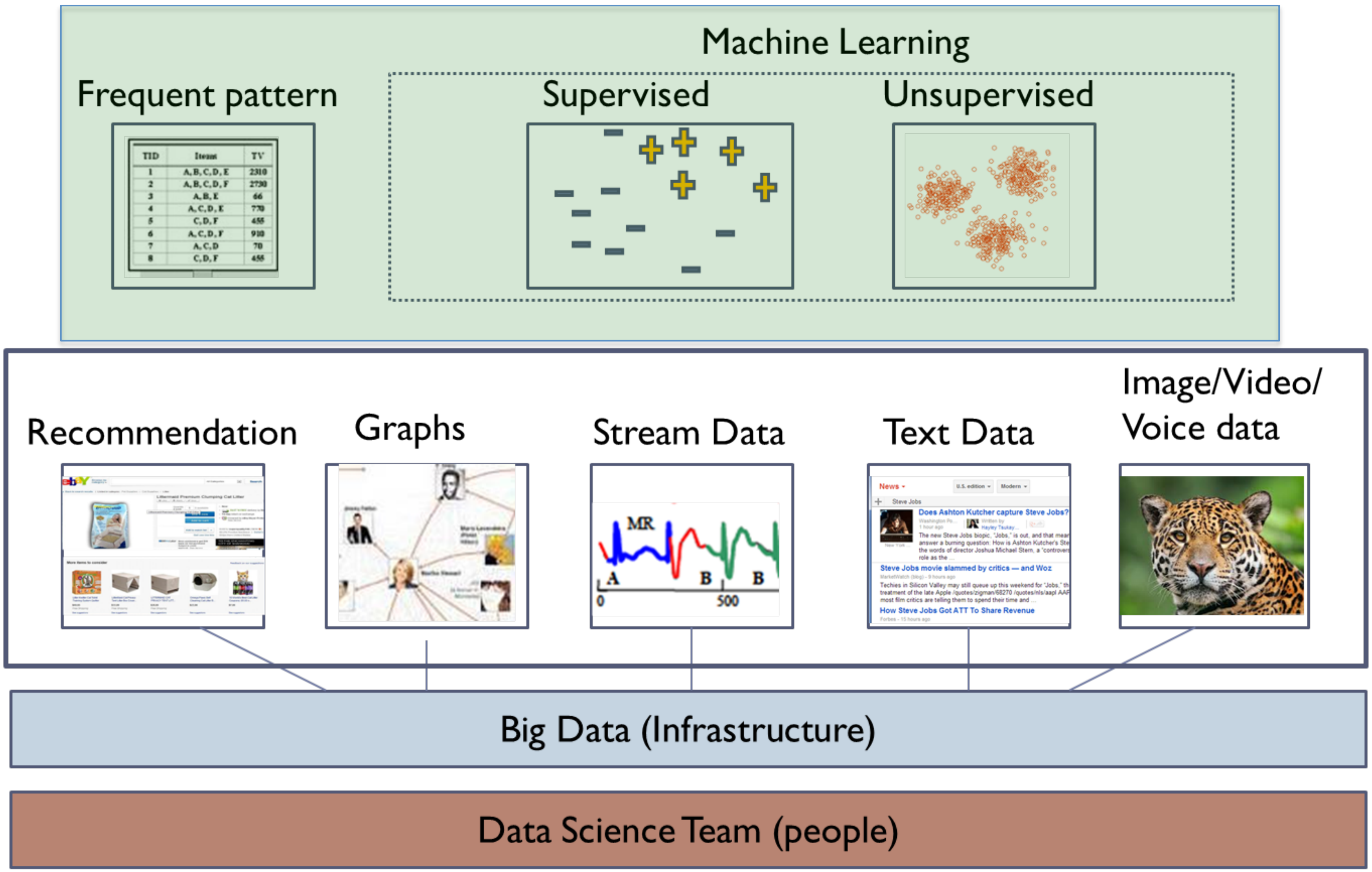
Learning is the adjusting of the weights $w_{i,j}$ such that the *cost function* $J(\boldsymbol{\theta})$ *is minimized*

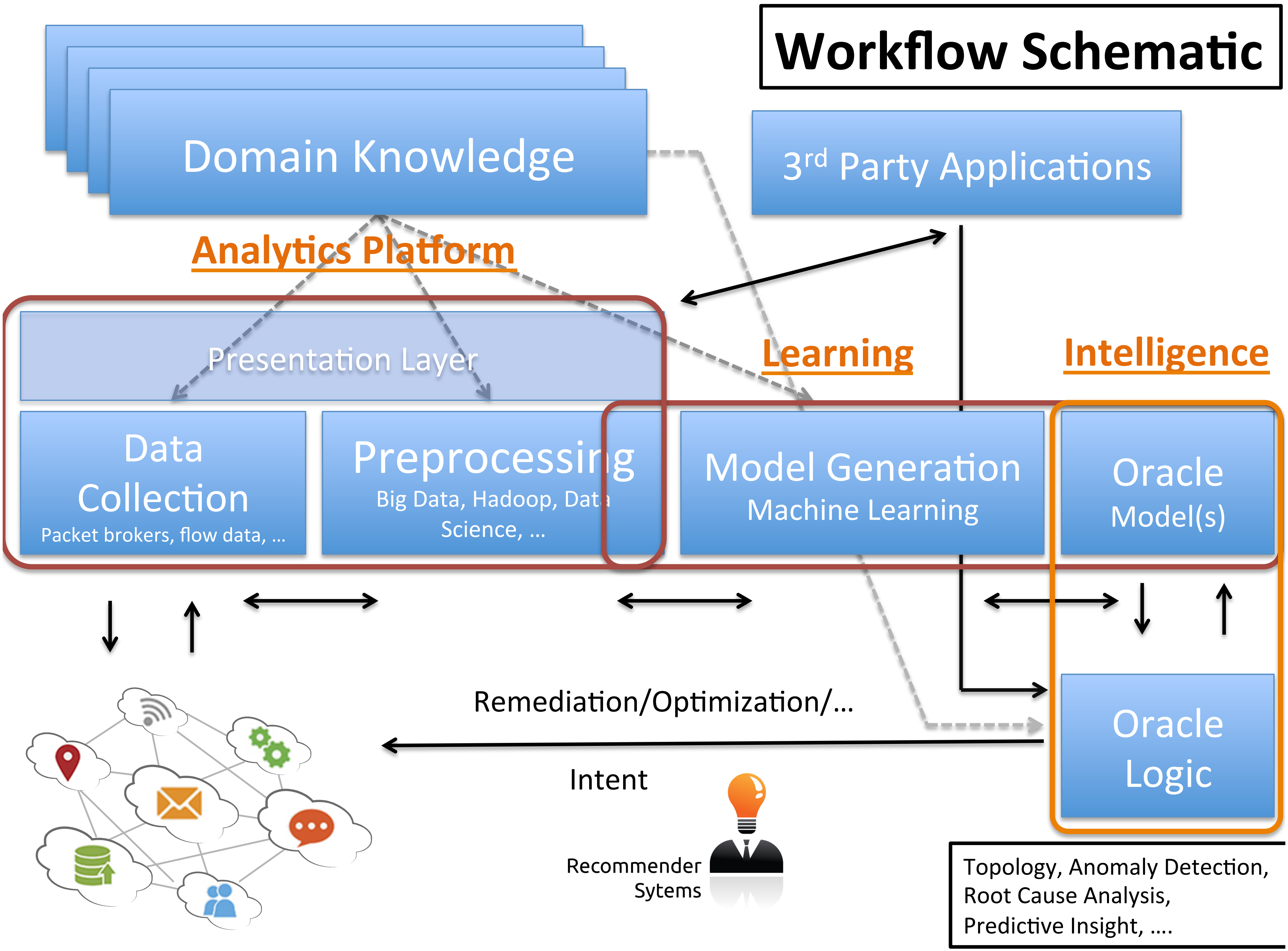
Simple learning procedure: *Back Propagation* (of the error signal)

Agenda

- ~~What is all the (ML) excitement about?~~
- ~~Review: What is ML (and why do we care)?~~
- ML Tools for DevOPs
- What the Future Holds
- Q&A

Prototypical ML Stack





Workflow Schematic
Domain Knowledge
3rd Party Applications
Analytics Platform
Learning
Intelligence
Presentation Layer
Data Collection
Packet brokers, flow data, …
Preprocessing
Big Data, Hadoop, Data Science, …
Model Generation
Machine Learning
Oracle Model(s)
Oracle Logic
Remediation/Optimization/…
Intent
Recommender Sytems
Topology, Anomaly Detection, Root Cause Analysis, Predictive Insight, ….

Simple Example: Application Profiling

- Goal: Build tools for the DevOps environment
  - Provide deeper automation and new capabilities/insight
  - Obvious ML application
  - Primarily management plane

- One approach: *Frequent Pattern Mining* and *K-Means* to learn/predict application behavior
  - FP Mining really more of a simple statistical method
  - K-Means is an unsupervised *local estimator* →
    - Partitions the input space into regions
    - Each region requires different parameters/degrees of freedom
    - Regions are needed to account for the shape of the target function

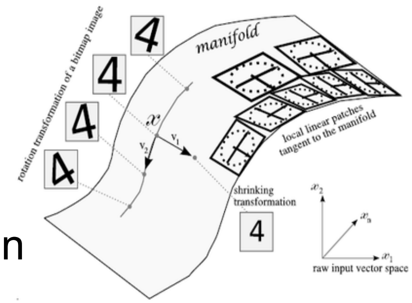


Image courtesy Yoshua Bengio

Applying Occam's Razor

- One of the first applications for DevOps we've done is Application Profiling
  - Continues to the right on the Automation Continuum
- Use *Frequent Pattern (FP) Mining* and *K-Means* to gain a deeper understanding of what's happening in our CSNSE
  - CSNSE = Compute, Storage, Network, Security, and Energy
  - Why is this an application of Occam's Razor?
- What do we mean by "Application Profiling"?
  - First, let's briefly look at the FP Mining and K-Means algorithms

Frequent Pattern Mining
(FP Growth Algorithm)

- FP builds a inclusion graph
  - reports frequent patterns (itemsets)
  - works on “categorical” data
    - data with no meaningful ordering

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

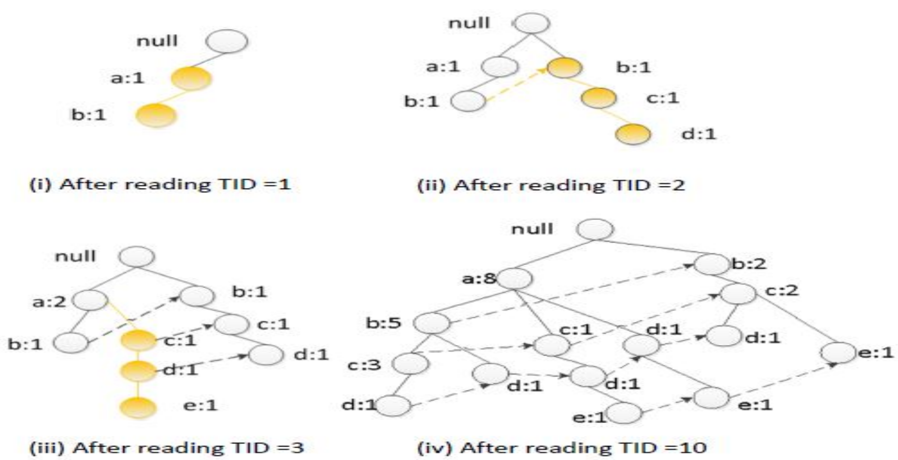
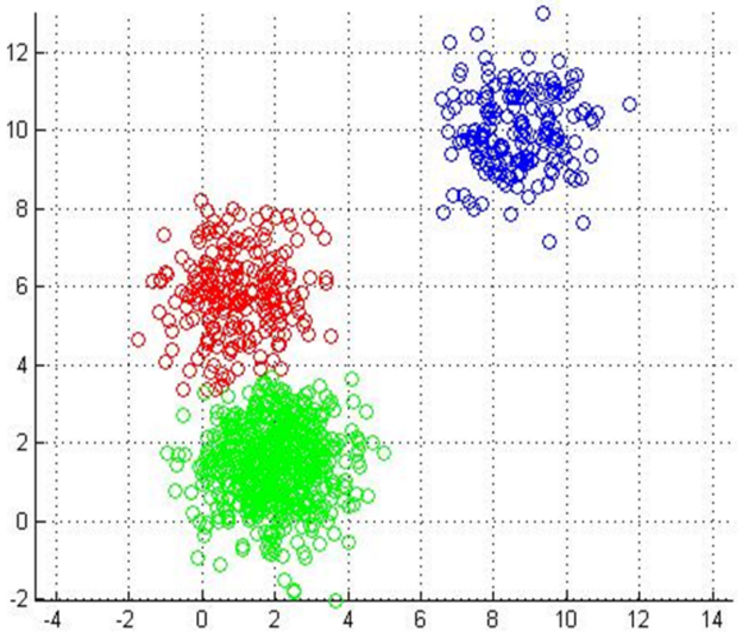
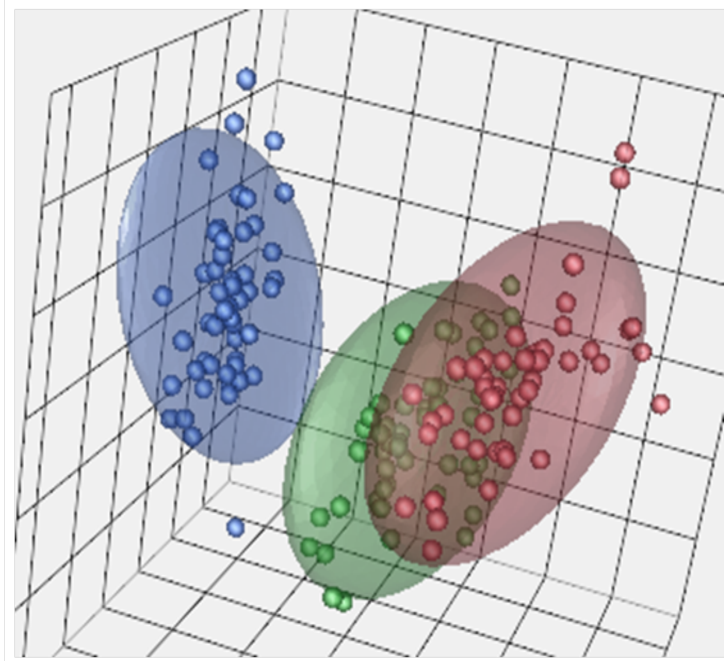


Fig. 7. Construction of an FP-tree - Based on [28]

K-Means
(K-Means Clustering)

PCA





number of clusters

number of cases

case i

centroid for cluster j

objective function $\leftarrow$ $$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \underbrace{\left\| x_i^{(j)} - c_j \right\|^2}_{\text{Distance function}}$$

A Little More on K-Means

K-Means Algorithm

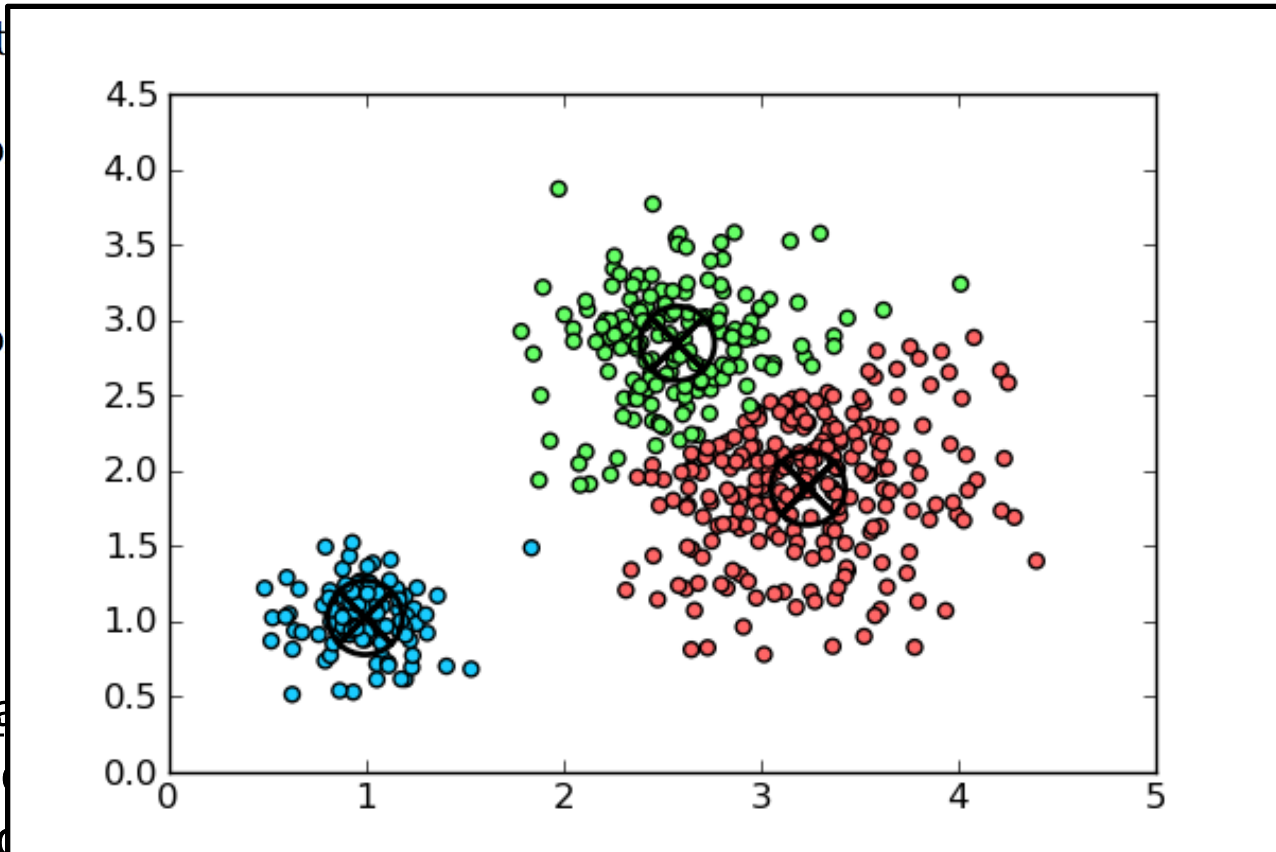
1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat

Fo

Fo

}



In words Ca on

- Rand
- Until c
  - Assign each roid
  - Update each igned to it

$$J(c, \mu) = \sum_{i=1} ||x^{(i)} - \mu_{c(i)}||^2$$

Putting It All Together

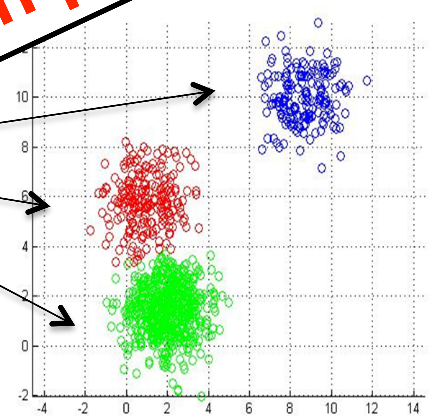
- *FP Mining* finds patterns in
  - Returns “itemsets”
    - Sets of Transaction I
    - **[src,dest,sr**
- *K* *uous* data

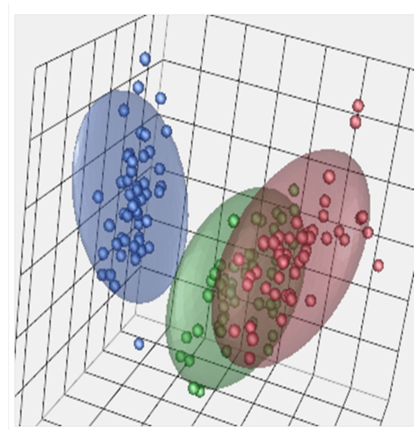
congestion, …

Putting these algorithms together allows us to make the following (very) simple inference:

$TIDset_{FP} \wedge TIDset_{K\text{-}Means} \rightarrow$ *patterns that cluster together*

“These application patterns occur with congestion”





BTW, how hard is it to code up FP in Spark/MLlib/Scala (or Python, Java, R)?

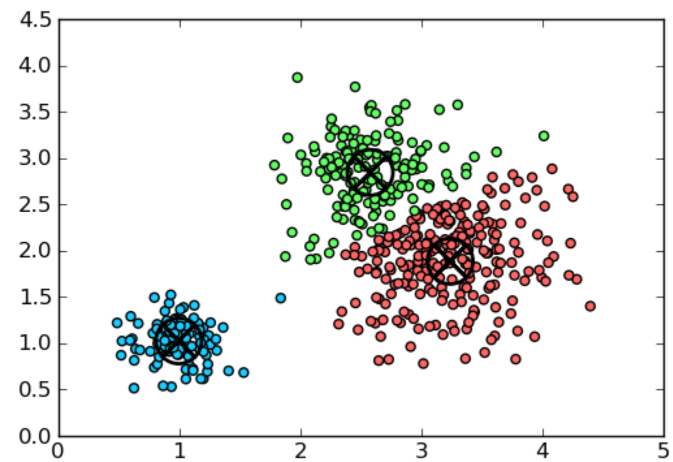
```
// ... ETL the dataset(s)
val transactions = rawData.map {line =>
        val buffer = ArrayBuffer[String]()
        buffer.appendAll(line.split(","))          // kddcup99 csv and terminology
                 Array(buffer(Proto),              // buffer(1)  protocol_type: symbolic
                       buffer(Service),            // buffer(2)  service: symbolic
                       buffer(Flag),               // buffer(3)  flag: symbolic
                       buffer(buffer.length-1))    // buffer(length-1) label
}.cache()
```

```
val fpg = new FPGrowth()
              .setMinSupport(MinSupport)       // model hyper-parameters
              .setNumPartitions(Partitions)    // hyper-parameters
val model = fpg.run(transactions)

model.freqItemsets.collect().foreach {itemset =>
  println(itemset.items.mkString("[", ",", "]") + ", " + itemset.freq)
}
```

Spark: https://spark.apache.org/downloads.html
Code: https://github.com/davidmeyer/ml
Dataset: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99

What About K-Means?

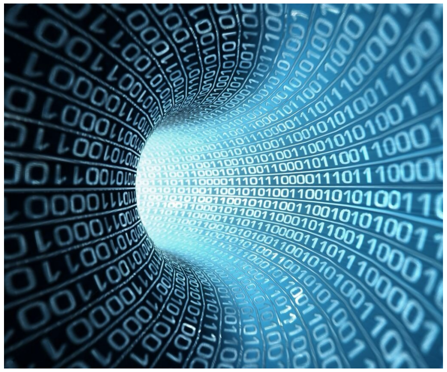



```
// ... ETL the dataset(s) → normalizedData
val kmeans = new KMeans()
                   .setK(K)
                   .setRuns(Runs)
                   .setEpsilon(Epsilon)
val model = kmeans.run(normalizedData)
val clusterAndLabel = rdataAndLabel.map {
   case (normalizedData,label) => (model.predict(normalizedData), label)}
val clusterLabelCount = clusterAndLabel.countByValue
clusterLabelCount.toList.sorted.foreach {
   case ((cluster,label), count) => println(f"$cluster%1s$label%18s$count%8s")}
```

Code: https://github.com/davidmeyer/ml

Application Profiling, cont

- First, we need data (obvious, but ingestion, … not trivial)
  - Lots of engines (spark, storm, tigon/cask.io,…)
  - Data we have collected (among other things)
    - Network and endpoint information
    - Environmental sensor data
    - Chef/Puppet, Openstack Heat, server/cluster state,…
    - …



- The FP-KMeans pipeline can be used build *application profiles*
  - Which endpoints an application talks to (and associated templates)
  - Which ports and protocols it uses
    - and associated meta-data, geo-ip, …
  - Flow characteristics including as TOD, volume and duration
  - Other *CSNSE* configuration associated with the application
    - ACL/QoS, routing policies,…
  - …

- We are really limited only by our imagination and (of course) our datasets

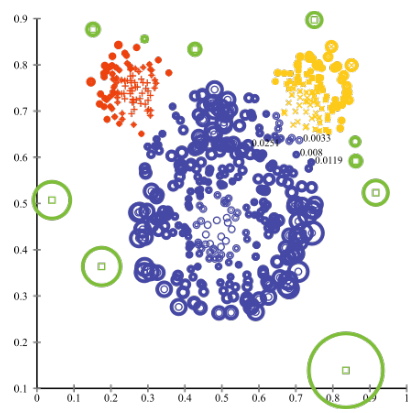
- Primarily descriptive/diagnostic analyzes

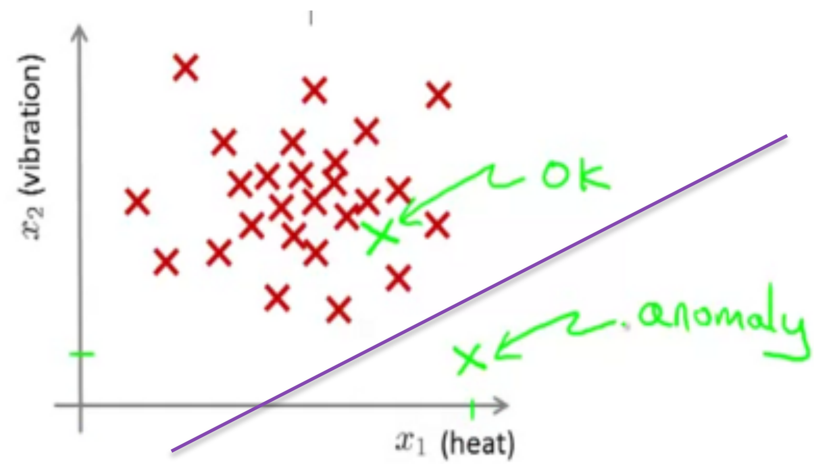
So what is more interesting…

- We can use the same FP-KMeans pipeline in a *predictive* way
  - For example, we can analyze changes to predict possible behavior
    - This ACL/Routing/QoS change will cause event <X> with *probability P*
    - If you configure app <X> with params <Y> there is *prob P* of congestion
    - …
  - We can correlate real-time application profiles with events/state
    - Application <X> is green (intelligent dashboard)
    - Queue <X> is dropping <Y>% of it's packets; app <Z> is talking to this endpoint
    - …
  - We wan also use application profiles to train other ML instances
    - Recognize application behavior in real time
    - Detect ***anomalies***
      - Points that are far from any cluster (K-Means), and/or
      - $p(X; \mu, \mathbf{\Sigma}) < \varepsilon$ (say in a multivariate Gaussian $\mathcal{N}(\mu,\mathbf{\Sigma})$ anomaly detection setting)
      - Security use cases
    - …
  - …
- This can all be made "streaming"/real-time

BTW, What (technically) are Anomalies?

- An anomaly is a pattern that does not conform to the expected behaviour
  - How to define expected behaviour?
  - How to find the "outliers"?



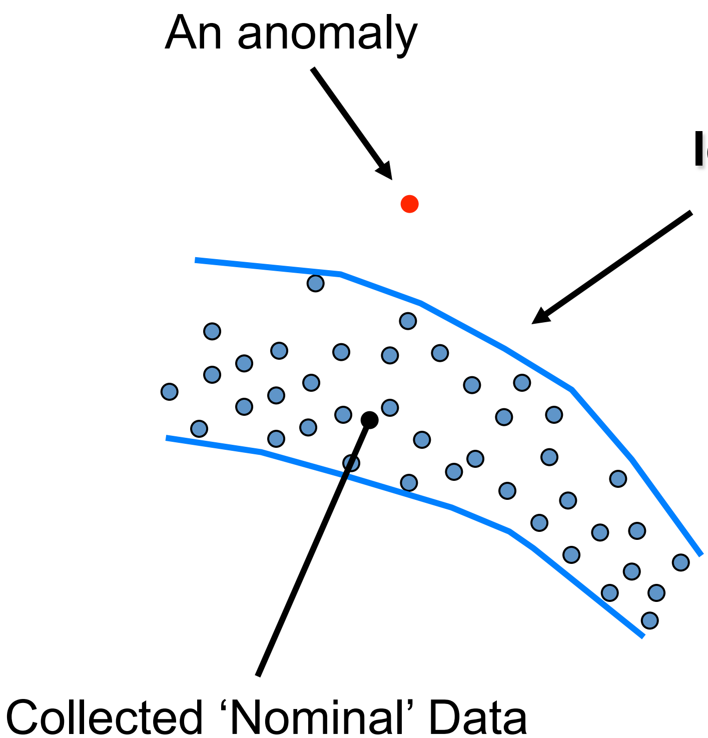


Linear Decision Boundary

- Anomalies translate to significant real life events
  - Cyber intrusions
  - Cyber crime
  - Manufacturing/product defects
  - …

Graphic courtesy Andrew Ng, others

Basic Idea Behind Anomaly Detection



Idea: Assume that a boundary exists and that

- Nominal data is inside the boundary
- Anomalous data is outside the boundary

Problem: How to estimate/approximate the boundary?

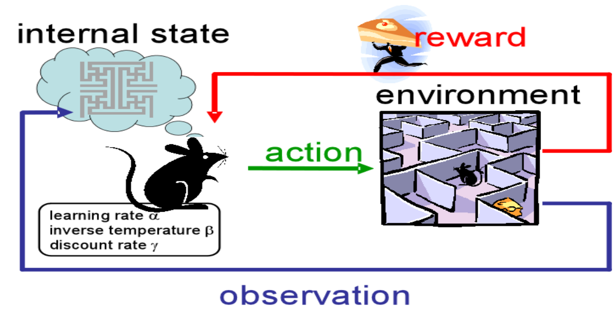
Problem: What measurement(s) caused the anomaly?

Problem: How far off-nominal is the anomaly/feature?

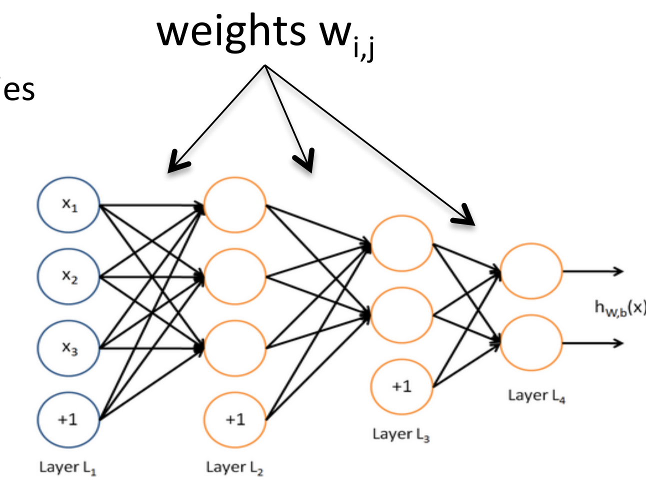
Agenda

- ~~What is all the (ML) excitement about?~~
- ~~Review: What is ML (and why do we care)?~~
- ~~ML Tools for DevOPs~~
- What the Future Holds
- Q&A

What The Future Holds



- These technologies, which have been so successful in perceptual tasks, are coming to networking…now
  - Extremely powerful deep neural nets (DNNs)
    - Remember, conventional statistical models learn simple patterns or clusters
    - OTOH, large DNNs learn a computation (function)
  - More emphasis on control
    - e.g., RNN/Memory Nets, Reinforcement learning
    - Can analyze sophisticated time-series/long range dependencies



- ML will be doing unexpected network (CSNSE) tasks
  - Who thought we'd be this close to self-driving cars?
  - DNNs already *write code*
    - The weight matrix **W** (this is what is learned)
  - DNNs solve the *selectivity-invariance dilemma*

- We will see progressively more ML in networking
  - Predictive and reactive roles in management, control and data planes
  - This will change the nature of how we design, build and operate networks
  - Recently proposed: IRTF RG on ML for Networking
    - http://trac.tools.ietf.org/group/irtf/trac/wiki/nml

- We are at the very beginning of a "network intelligence" revolution

Q&A

Thanks!

http://www.1-4-5.net/~dmm/talks/2015/hoti.pptx