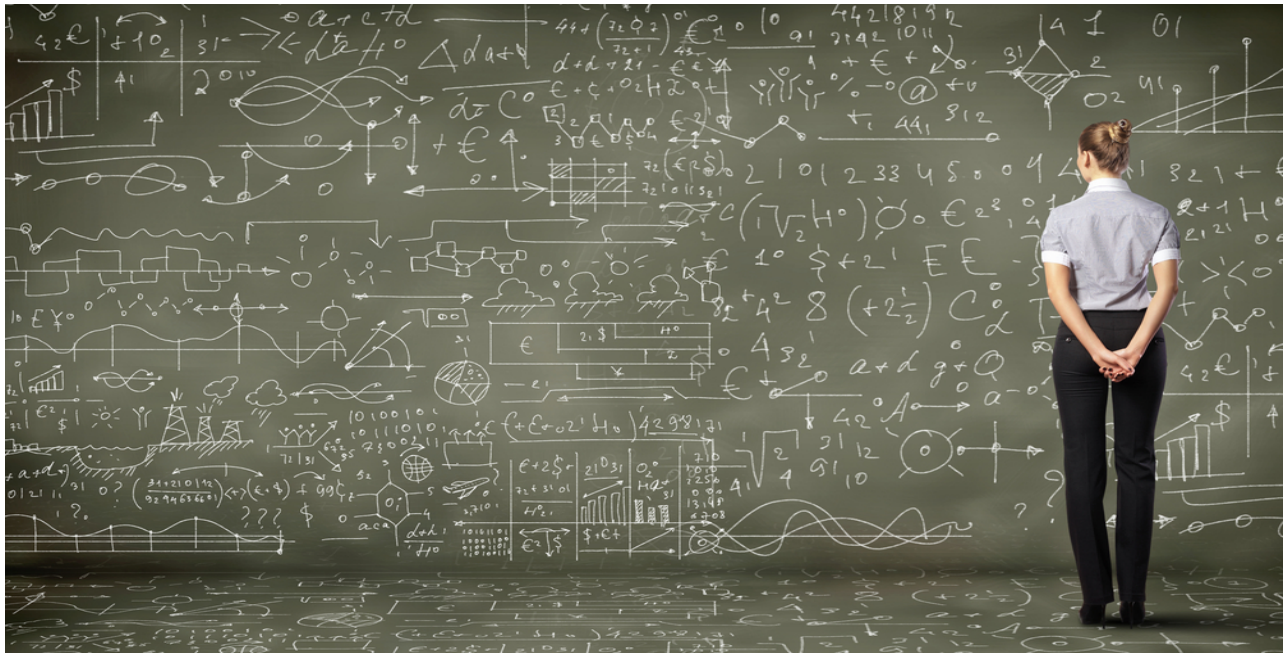
A (Very) Brief Introduction to Machine Learning

and its Application to Mobile Networks



David Meyer
SP CTO and Chief Scientist
Brocade
dmm@brocade.com

Agenda

- Goals for this Talk
- Automation Continuum
- Software Defined Intelligence
  - Architecture and Pipeline
- What is Machine Learning?
- Mobile Use Case(s)
- Appendix: How can machine learning possibly work?

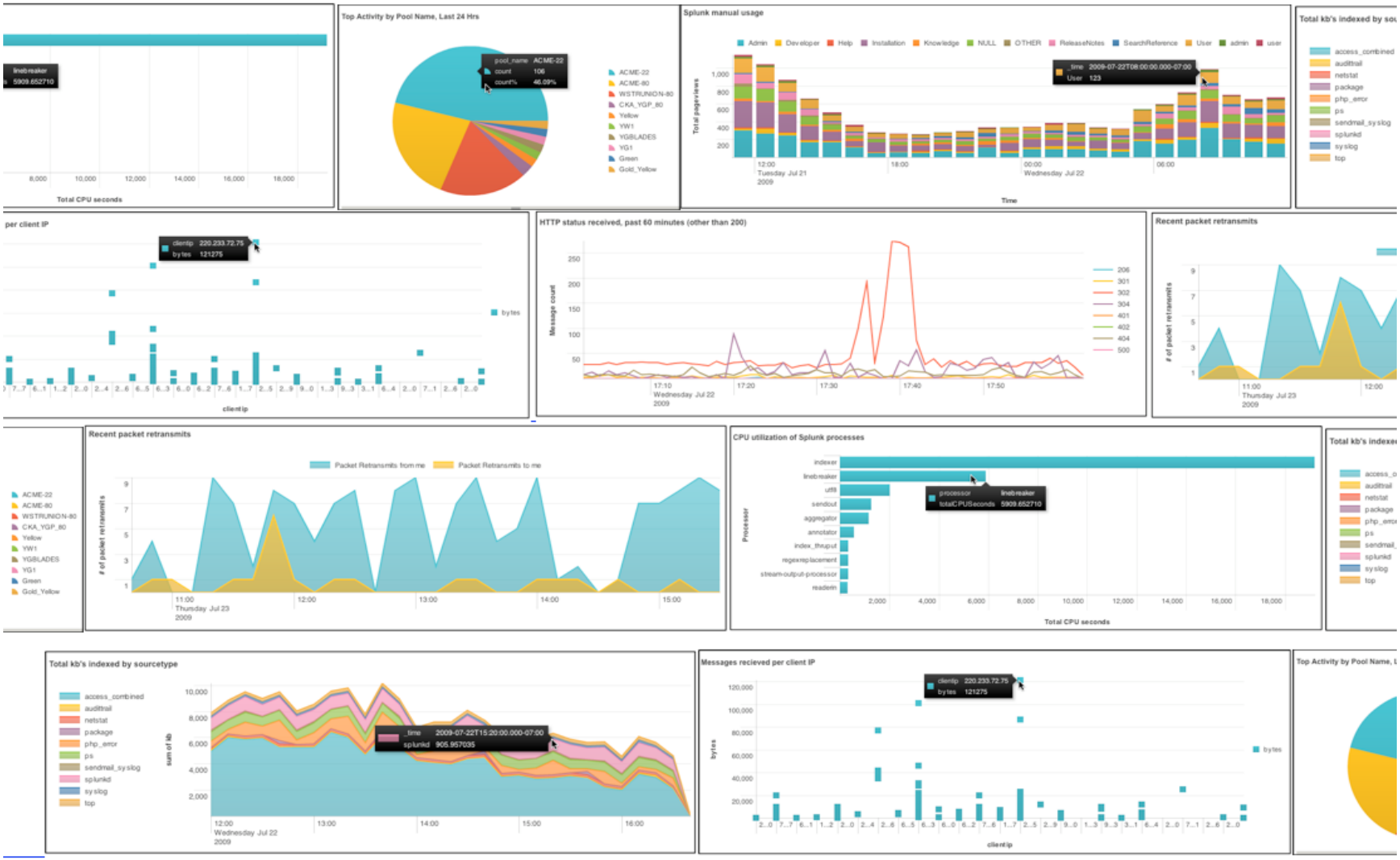
Goals for this Talks

To give us a basic common understanding of Machine Learning and Software Defined Intelligence so that we can discuss their application to Carrier Mobile use cases.

Agenda

- ~~Goals for this Talk~~
- Automation Continuum
- Software Defined Intelligence
  - Architecture and Pipeline
- What is Machine Learning?
- Mobile Use Case(s)
- Appendix: How can machine learning possibly work?

Brief Overview of Analytics Use Cases



SDM and Analytics Use-Cases

SDM FUNCTIONS SUPPORTED	MOBILE NETWORKS	FIXED NETWORKS	ANALYTICS ENHANCED
Increasing database efficiency (includes replacing legacy databases such as HLRs with next-gen databases)	✓		
Increasing database efficiency (includes deployment of an LTE-capable HSS)	✓		
Advanced reporting and profiling support (analysis on quality of service, end-user behavior)	✓	✓	✓
Plan offer enrichment (includes personalized and potentially location-based advertisements)	✓	✓	✓
Personalized services access to social media and OTT services (e.g., Facebook)	✓	✓	✓
Operator sponsored/partnered content applications	✓	✓	✓
Parental controls	✓	✓	✓
Mobile messaging services	✓		✓
Bill shock notification (notice of threshold reached)	✓		✓
Customer loyalty programs	✓	✓	✓
Managing M2M devices	✓		✓
Online access to services profile	✓	✓	✓
Support of multitenant applications (e.g., hosting applications for a number of carriers or MVNOs)	✓		✓
Support of cloud-based services	✓	✓	✓
Managing privacy data	✓	✓	✓

Source: Heavy Reading

Segmentation and Hierarchy of Analytics

Analytics can be looked at in multiple segments

- ***Historical Analytics***: Build data warehouses / run batch queries to predict future events / generate trend reports
- ***Near Real-Time Analytics***: Analyze indexed data to provide visibility into current environment / provide usage reports
- ***Real-Time Analytics***: Analyze data as it is created to provide **instantaneous**, **actionable** business intelligence to affect immediate change
- ***Predictive Analytics:*** Build statistical models that can classify/predict the near future

Each segment of analytics serves specific purposes

- Historical Analytics: Campaign & service plan creation, network planning, subscriber profiling, customer care
- Near Real-time Analytics: Network optimization, new monetization use-cases, targeted services (ex. Location-based)
- Real-time Analytics: Dynamic policy, self-optimizing networks, traffic shaping, topology change, live customer care

Data is richer when associated to context – location, time of day, etc.

For each type of data, there is a window / meaningful time period of which the data is relevant

"Right time" or "timeliness" is a consideration to the query itself, not the data set

In mobile, the "window of relevance" of contextual data is consistently shrinking

The Network and Big Data

- So…the network is evolving into a source of data for analytics
  - Utilization, performance, security data, …
  - Traffic – what is crossing the network?
    - Browsing
    - Applications
- The network is evolving as a transport of data for analytics
  - M2M
  - Not location bound, distributed over many points of network attachment
- In many cases data will be write once, read many to support a variety of analytics processes
  - Access to data via distributed compute/analytics
  - Access to data via distribution of the data

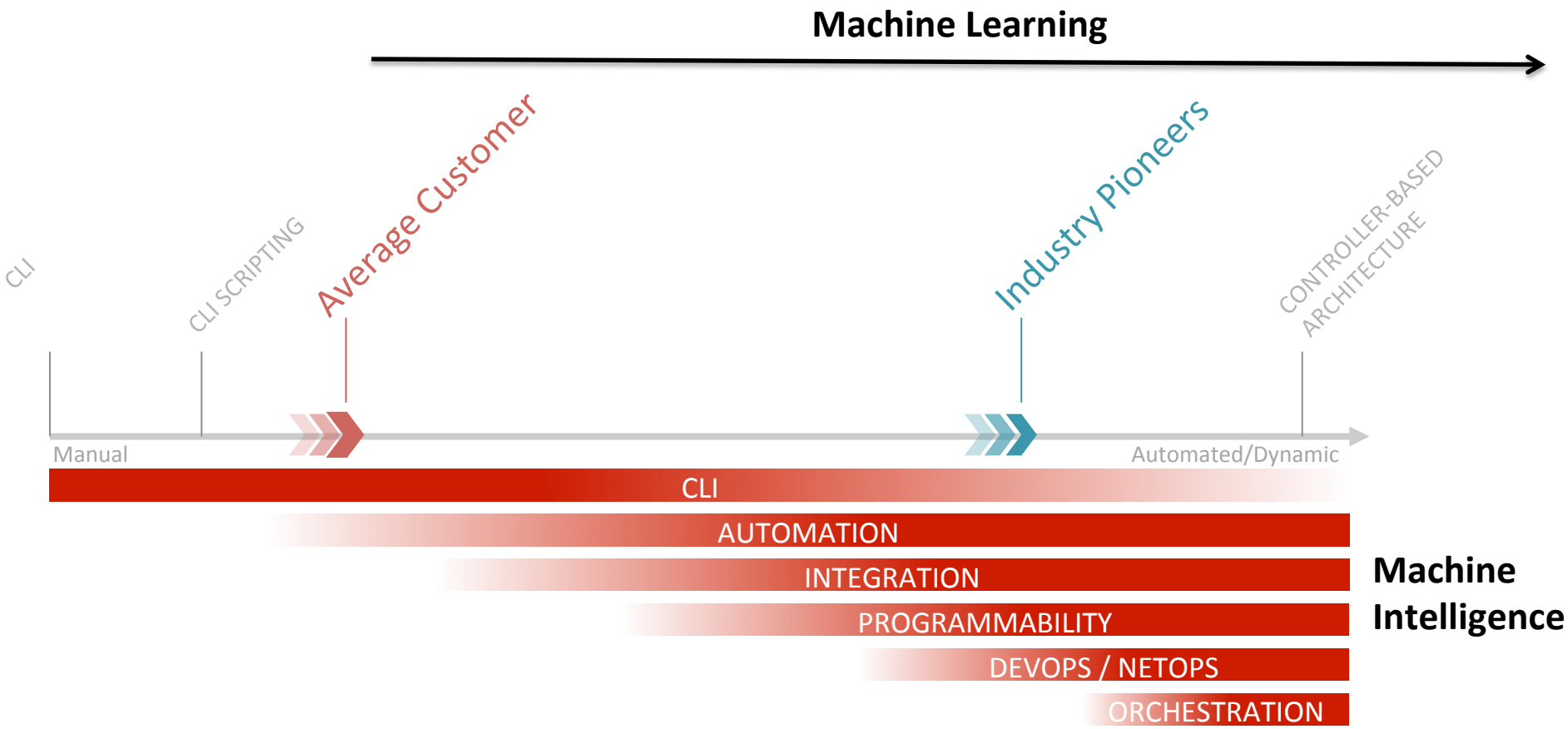
What types of “big data” are out there?

	Profile	Profiling
User Analytics	Identity (Persistent) Demographics Explicit profile (interests, etc.) Device(s) and capabilities Billing / Subscription plan	Device sensor data Persistent Location / Presence Behavioral / Search / Social Purchasing / Payments Mobility patterns Usage data (from device)
Content Analytics	Catalog / Title Topic / Keywords CA / Rights management Encryption / DRM Format(s) / Aspect ratio(s) Resolution(s) / Frame rate(s)	Consumption data Content reach Asset popularity / revenue Distribution/Retention/Archival Search / Discover / Recommend Usage Data (from content source)
Network Analytics	Bandwidth and latency Access types IP pools Routes / topology / Path QoS / Policy Rulesets Network Service Capabilities	Active subscriber demographics Crowdsourced data Geographic segmentation Network Performance / Quality Network sensor data (IoT/M2M) Usage (from DPI)

Slide courtesy Kevin Shatzkamer

Automation Continuum

While customers today have a broad range of network management approaches, nearly all struggle to understand how to reach their goal of a fully automated and dynamic architecture.



Original slide courtesy Mike Bushong and Joshua Soto

Machine Intelligence LANDSCAPE

CORE TECHNOLOGIES

ARTIFICIAL INTELLIGENCE
IBM WATSON, MetaMind, Numenta, ai-one, Cycorp, Microsoft Research, nara, Reactor, SI SCALED INFERENCE

DEEP LEARNING
vicarious, Vision Factory, facebook, Baidu IDL, Google, ersatz, SKYMIND, SignalSense

MACHINE LEARNING
rapidminer, context relevant, 0xdata H2O, DATARPM, Liftigniter, SPARKBEYOND, Azure ML, ayhat, Wise.io, Sense, GraphLab, Alpine, Metonian

NLP PLATFORMS
cortical.io, idibon, LUMINOSO, wit.ai, MaluubA

PREDICTIVE APIS
AlchemyAPI, MINDOPS, Google, bigml, indico, ALGORITHMIA, Expect Labs, PredictionIO

IMAGE RECOGNITION
clarifai, MADBITS, DNNresearch, DEXTRO, VISENZE, lookflow

SPEECH RECOGNITION
GRIDSPACE, popUP archive, NUANCE

RETHINKING ENTERPRISE

SALES
Preact, AVISO, RelateIQ, NGDATA, CLARABRIDGE, FRAMED, infer, ATTENTIVITY, causata

SECURITY / AUTHENTICATION
CROSSMATCH, conjur, EYEVERIFY, BITSIGHT, CYLANCE, AREA 1 SECURITY, bionym

FRAUD DETECTION
sift science, SOCURE, ThreatMetrix, feedzai, Brighterion, VERAFIN

HR / RECRUITING
TalentBin, entelo, predikt, Connectifier, gild, hiQ, CONCEPT NODE

MARKETING
brightfunnel, bloomreach, CommandIQ, AIRPR, RADIUS, TellApart, people pattern, Freshplum

PERSONAL ASSISTANT
Siri, X, Google now, Cortana, cleversense, tempo, Robinlabs, KASISTO, fuse|machines, VIV, CLARA LABS

INTELLIGENCE TOOLS
ADATAO, Palantir, Quid, Digital Reasoning, FirstRain

RETHINKING INDUSTRIES

ADTECH
METAMARKETS, dstillery, rocketfuel, YieldMo, ADBRAIN

AGRICULTURE
BLUE RIVER, TerrAvion, ceresimaging, HONEYCOMB, THE CLIMATE CORPORATION, tule, mavrx

EDUCATION
declara, coursera, KNEWTON, kidaptive

FINANCE
Bloomberg, FinGenius, alphasense, KENSHO, Dataminr, minettabrook, BINATIX

LEGAL
Lex Machina, brightleaf, COUNSELYTICS, RAVEL, JUDICATA, Brevia, DiligenceEngine

MANUFACTURING
SIGHT MACHINE, MICROSCAN, BOULDER IMAGING, IVISYS

MEDICAL
Parzival, transcriptic, ZEPHYR HEALTH, Genescient, grand round table, bina, TUTE GENOMICS

OIL AND GAS
kaggle, AYASDI, TACHYUS, biota, Flutura

MEDIA / CONTENT
Outbrain, newsle, ARRIA, SAILTHRU, wavii, Owlin, NarrativeScience, vsenp, Summly, Prismatic, ai

CONSUMER FINANCE
Affirm, inVenture, zest finance, BILLGUARD, LendUp, LendingClub, Kabbage

PHILANTHROPIES
DataKind, thorn, DATA GUILD

AUTOMOTIVE
Google, Continental, TESLA, MobileEye, CRUISE

DIAGNOSTICS
enlitic, 3SCAN, lumiata, CENTOR

RETAIL
BAY SENSORS, PRISM SKYLABS, celect, euclid

RETHINKING HUMANS / HCI

AUGMENTED REALITY
wearable intelligence, MAGNETIC, APX, blippar, META, layar

GESTURAL COMPUTING
THALMIC LABS, omek, FLUTTER, LEAP MOTION, eyeSight, 3Gear SYSTEMS, GestureTek, nod

ROBOTICS
intel, LIQUID ROBOTICS, SoftBank, iRobot, Boston Dynamics, jibo, anki, evolution robotics

EMOTIONAL RECOGNITION
affectiva, BEYOND VERBAL, EMOTIENT, BRIGHAM, cogito

SUPPORTING TECHNOLOGIES

HARDWARE
NVIDIA, XILINX, QUALCOMM, NERVANA SYSTEMS, TERADEEP, Artificial Learning, rigetti

DATA PREP
TRIFACTA, Paxata, tamr, Alation

DATA COLLECTION
diffbot, kimono, CrowdFlower, Connotate, WorkFusion, import.io

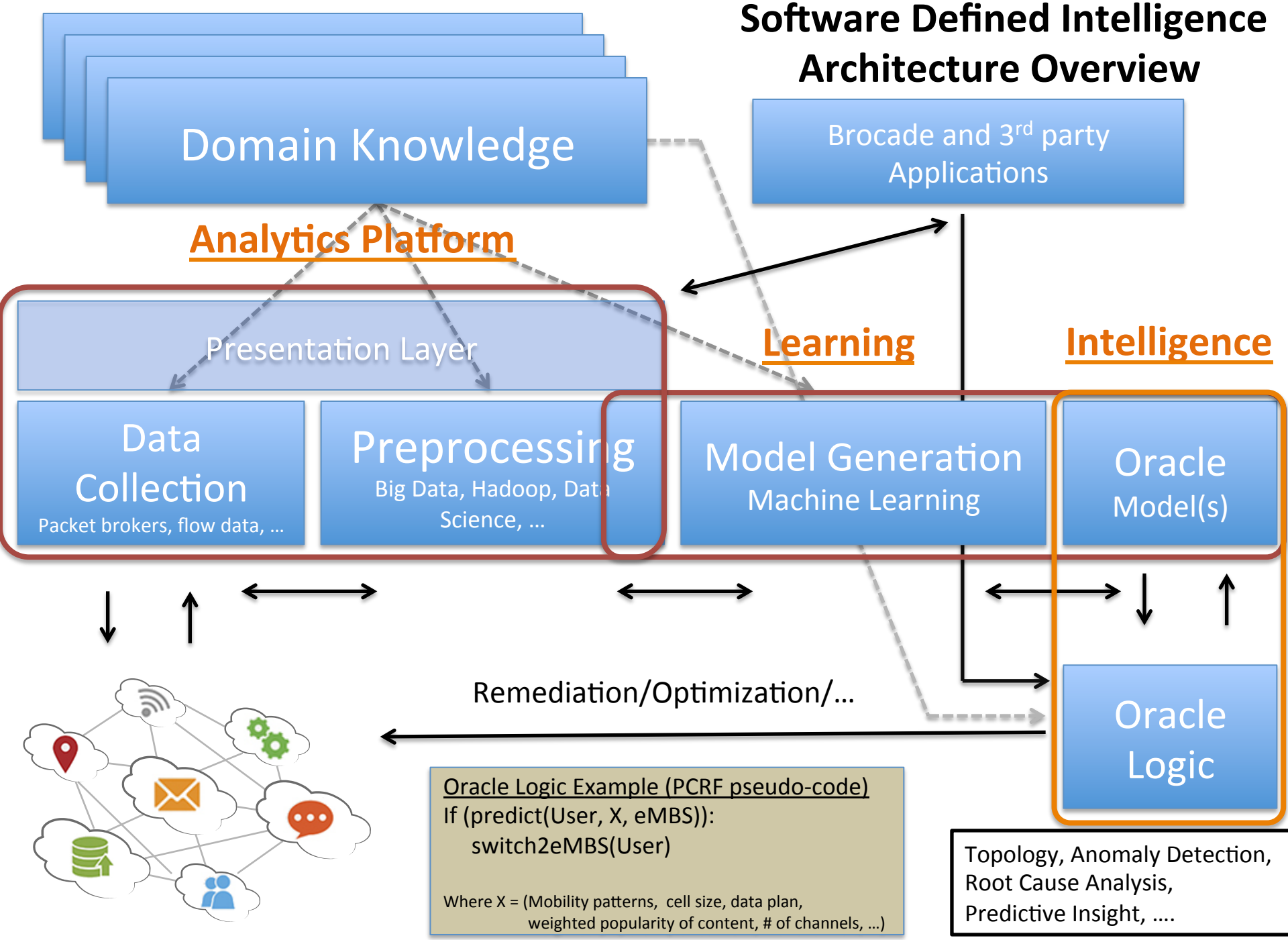
www.shivonzilis.com/machineintelligence

Bloomberg BETA

Agenda

- ~~Goals for this Talk~~
- ~~Automation Continuum~~
- Software Defined Intelligence
  - Architecture and Pipeline
- What is Machine Learning?
- Mobile Use Case(s)
- Appendix: How can machine learning possibly work?

Software Defined Intelligence Architecture Overview



What Might a Mobile Analytics Platform Look Like?

Think "Platform", not Applications, Algorithm, Visualization

Brocade / 3rd Party Applications

- SON
- PCRF
- SDN Controller
- NFV-O

Index / Schema
(Metadata Mgmt)

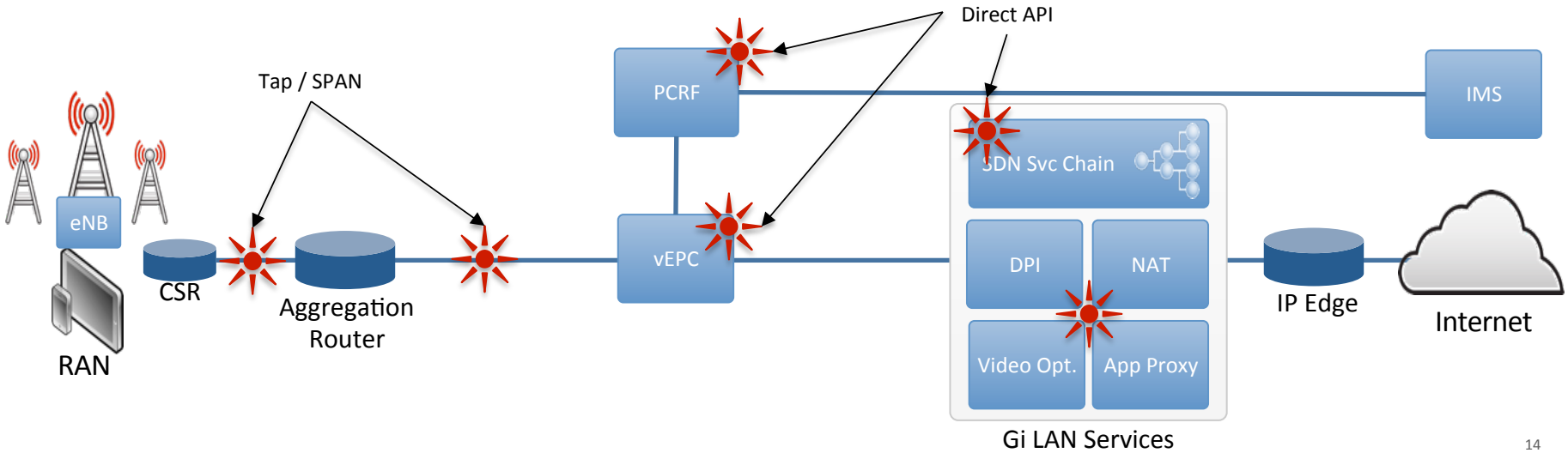
Service Provider Use-Cases

- Operations
- Marketing
- Cust. Care
- NW Planning
- Security

Distributed Data Management
(Pre-filtering, aggregation, normalization (time / location), distribution)

Big Data Management
(Correlation, trend analysis, pattern recognition)

Data Collection (Push) / Extraction (Pull)
(RAN, IPBH, LTE EPC, Gi LAN, IMS, Network Services, OSS)



Slide courtesy Kevin Shatzkamer

© 2014 BROCADE COMMUNICATIONS SYSTEMS, INC. CONFIDENTIAL—FOR INTERNAL USE ONLY

Agenda

- ~~Goals for this Talk~~
- ~~Automation Continuum~~
- ~~Software Defined Intelligence~~
  - ~~Architecture and Pipeline~~
- What is Machine Learning?
- Mobile Use Case(s)
- Appendix: How can machine learning possibly work?

Before We Start

What is the SOTA in Machine Learning?

- "Building High-level Features Using Large S
  Andrew Ng, et. al, 2012
  - http://arxiv.org/pdf/1112.6209
  - Training a *deep neural net*
  - Showed that it is
    entirely *unl*
  - In par

out-of-plane

you are interested in auto-encoders)

machines, 16K cores

train

racy categorizing 22K object classes

70% improvement over current results

- Random guess achieves less than 0.005% accuracy for this dataset

Andrew Ng and his crew at Baidu have recently beat this record with their (GPU based) Deep Speech work. See **http://arxiv.org/abs/1412.5567**

1 http://www.image-net.org/

What is Machine Learning?

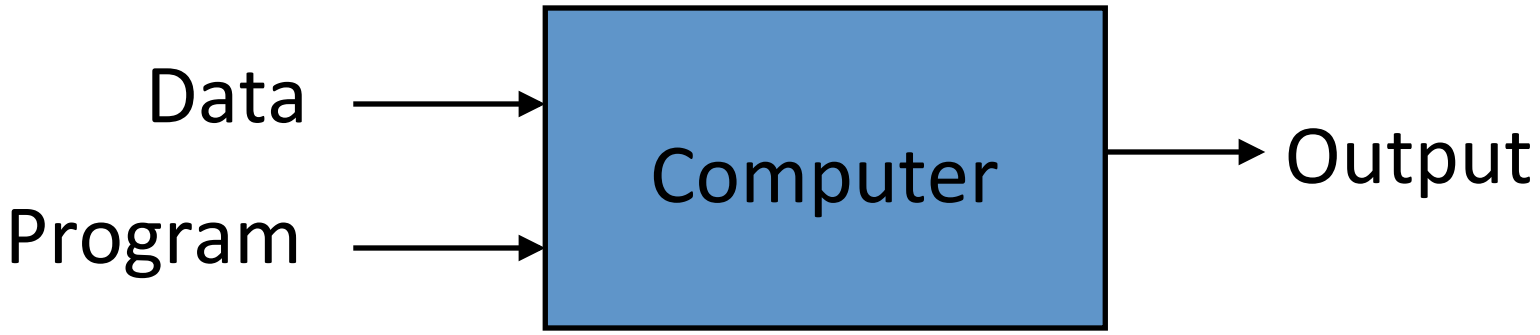
The complexity in traditional computer programming is in the code (programs that people write). In machine learning, algorithms (programs) are in principle simple and the complexity (structure) is in the data. Is there a way that we can automatically learn that structure? That is what is at the heart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

-- Andrew Ng

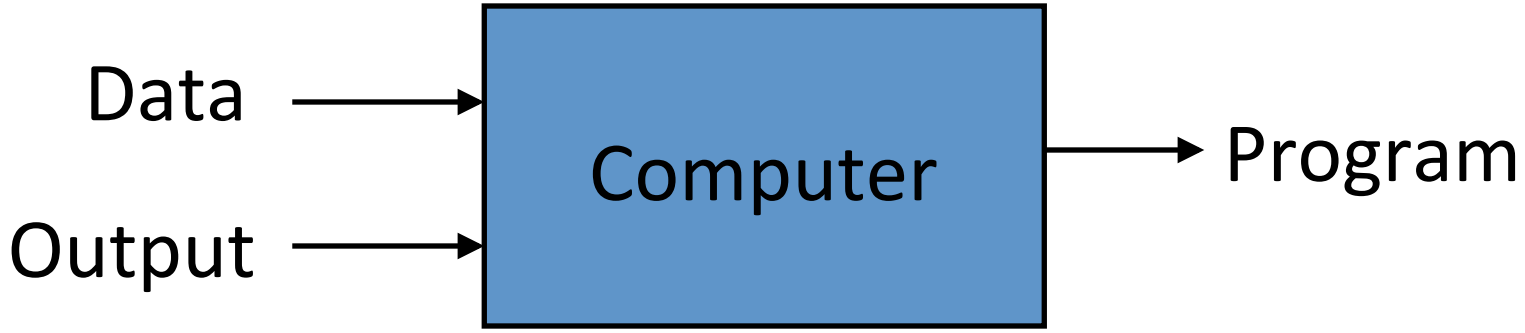
That is, machine learning is the about the construction and study of systems that can learn from data. This is very different than traditional computer programming.

The Same Thing Said in Cartoon Form

Traditional Programming



Machine Learning



When Would We Use Machine Learning?

- When patterns exists in our data
  - Even if we don't know what they are
    - Or perhaps especially when we don't know what they are
- We can not pin down the functional relationships mathematically
  - Else we would just code up the algorithm
- When we have lots of (unlabeled) data
  - Labeled training sets harder to come by
  - Data is of high-dimension
    - High dimension "features"
    - For example, sensor data
  - Want to "discover" lower-dimension representations
    - Dimension reduction
- Aside: Machine Learning is heavily focused on implementability
  - Frequently using well know numerical optimization techniques
  - Lots of open source code available
    - See e.g., libsvm (Support Vector Machines): http://www.csie.ntu.edu.tw/~cjlin/libsvm/
    - Most of my code in python: http://scikit-learn.org/stable/ (many others)
    - Languages (e.g., octave: https://www.gnu.org/software/octave/)

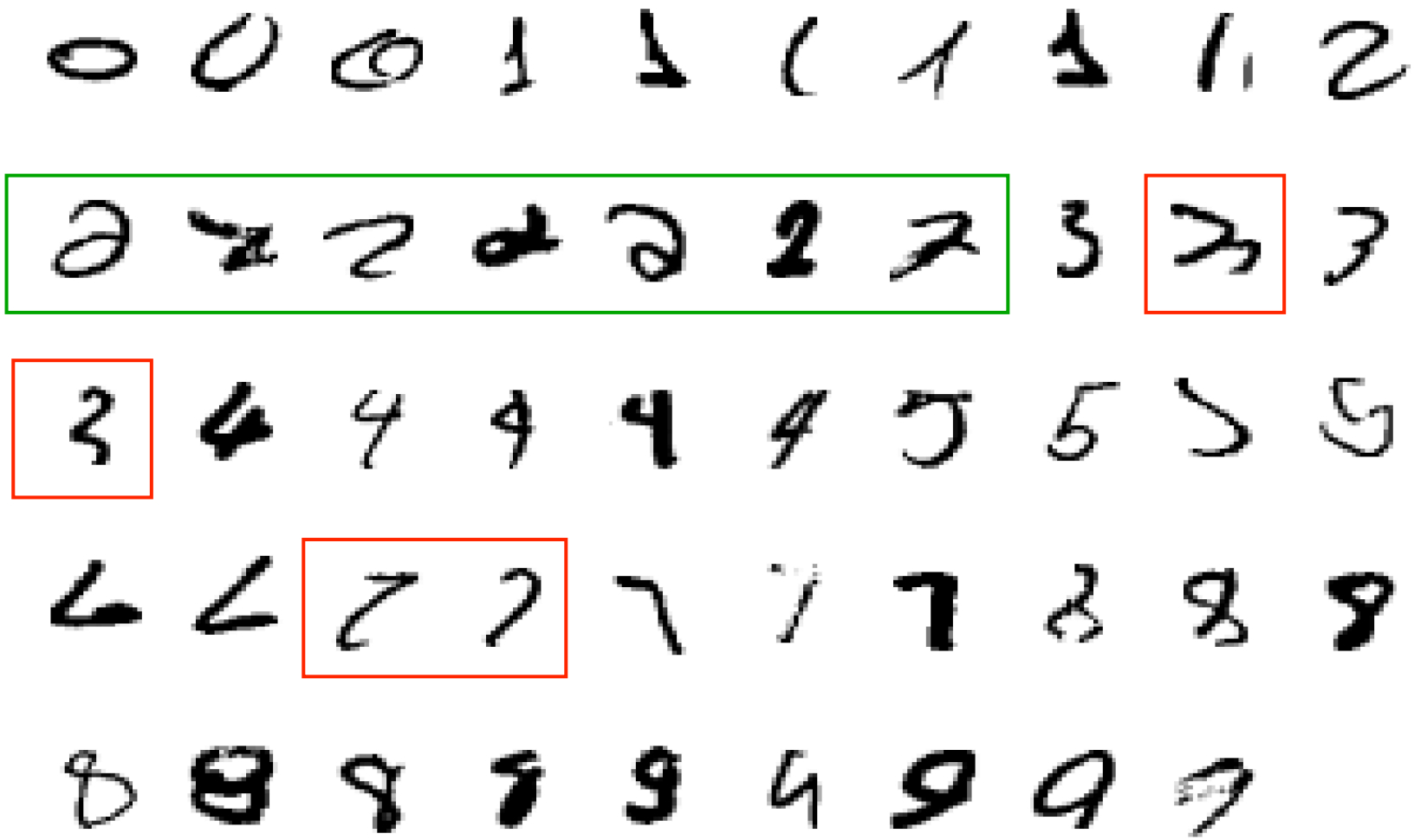
Why Machine Learning is Hard

You See

Your ML Algorithm Sees

Why Machine Learning Is Hard, Redux

What is a “2”?



Examples of Machine Learning Problems

- Pattern Recognition
  - Facial identities or facial expressions
  - Handwritten or spoken words (e.g., Siri)
  - Medical images
  - Sensor Data/IoT
- Optimization
  - Many parameters have “hidden” relationships that can be the basis of optimization
- Pattern Generation
  - Generating images or motion sequences
- Anomaly Detection
  - Unusual patterns in the telemetry from physical and/or virtual plants (e.g., data centers)
  - Unusual sequences of credit card transactions
  - Unusual patterns of sensor data from a nuclear power plant
    - or unusual sound in your car engine or …
- Prediction
  - Future stock prices or currency exchange rates

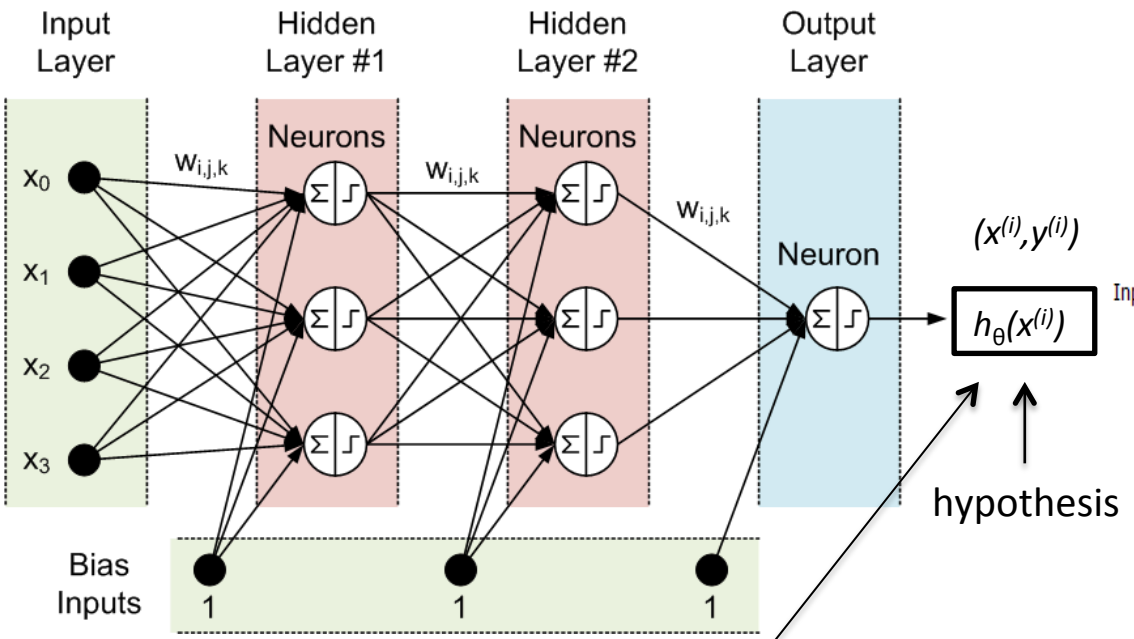
Machine Learning is a form of *Induction*

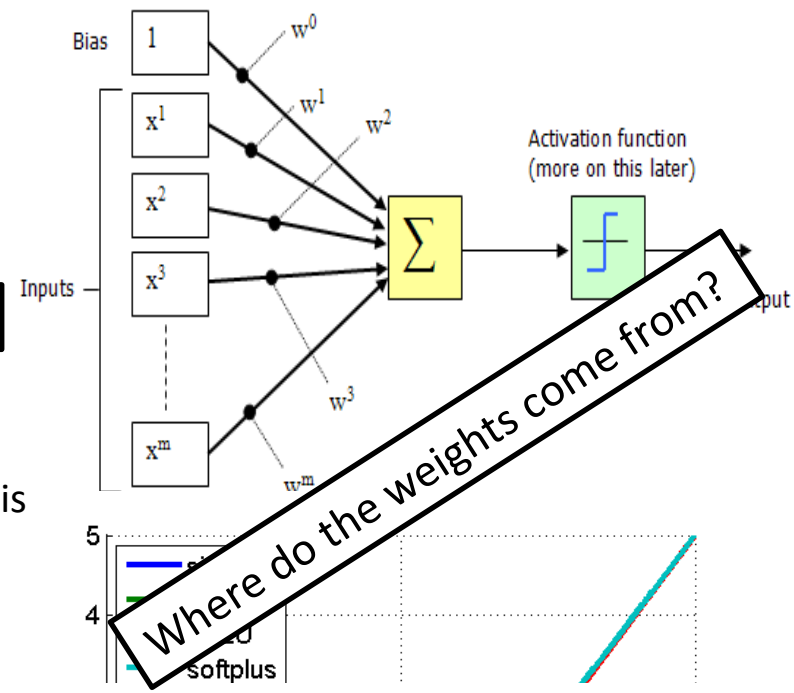
- Given examples of a function *(x, f(x))*
  - Supervised learning (because we're given *f(x)*)
  - Don't explicitly know *f*
    - Rather, trying to learn *f* from the data
  - Labeled data set (i.e., the *f(x)*'s)
  - Training set may be noisy, e.g., $(x, (f(x) + \varepsilon))$
  - Notation: $(x_i, f(x_i))$ denoted $(x^{(i)}, y^{(i)})$
  - $y^{(i)}$ sometimes called t_i (t for "target")

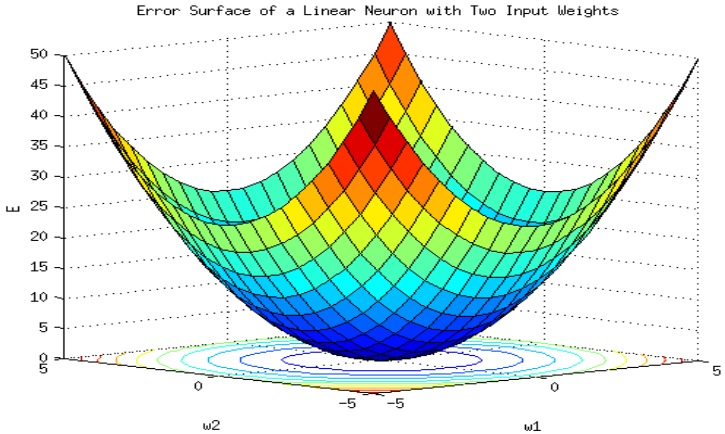
- Predict function ***f(x)*** for new examples *x*
  - Discrimination/Prediction (Regression): *f(x)* continuous
  - Classification: *f(x)* discrete
  - Estimation: $f(x) = P(Y = c|x)$ for some class c

Deep Feed Forward Neural Nets

(in 1 Slide (☺))





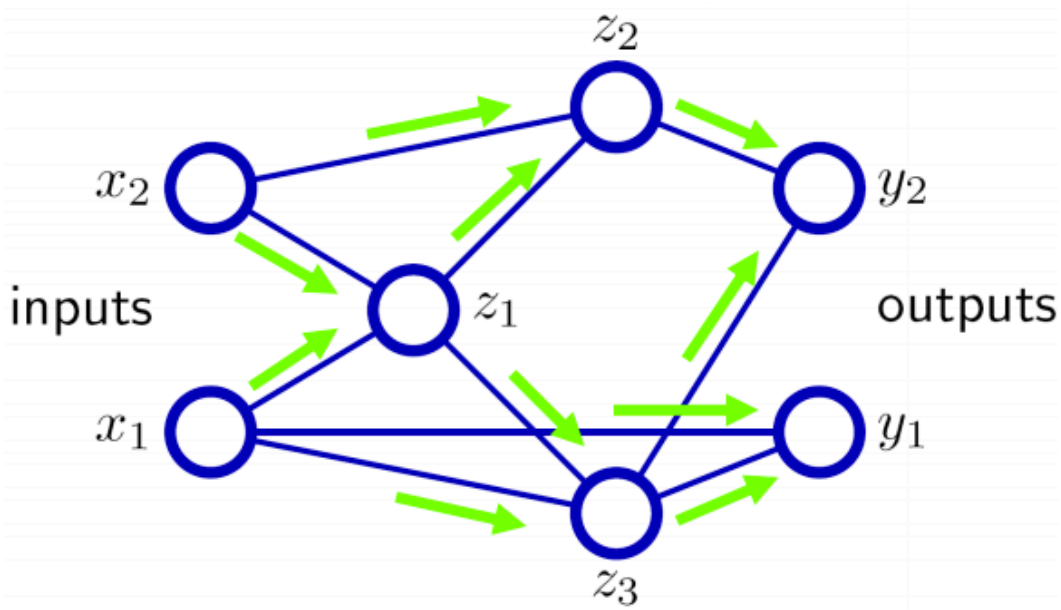


$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Learning is adjusting the $w_{i,j}$'s such that the cost function $J(\boldsymbol{\theta})$ is minimized (a form of Hebbian learning)

Forward Propagation Cartoon

- Forward Propagation :
  - Sum inputs, produce activation, feed-forward



Backpropagation Cartoon

http://chronicle.com/article/The-Believers/190147

$$l(\theta) = \sum_{i=1}^{n} y^{(i)} log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))$$

More Formally
Empirical Risk Minimization

- Empirical risk minimization
  - framework to design learning algorithms

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

  - $l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)})$ is a loss function (loss function also called "cost function" denoted $J(\theta)$)
  - $\Omega(\boldsymbol{\theta})$ is a regularizer (penalizes certain values of $\boldsymbol{\theta}$)
- Learning is cast as optimization
  - ideally, we'd optimize classification error, but it's not smooth
  - loss function is a surrogate for what we truly should optimize (e.g. upper bound)

Any interesting cost function is complicated and non-convex

Solving the Risk (Cost) Minimization Problem
Gradient Descent – Basic Idea

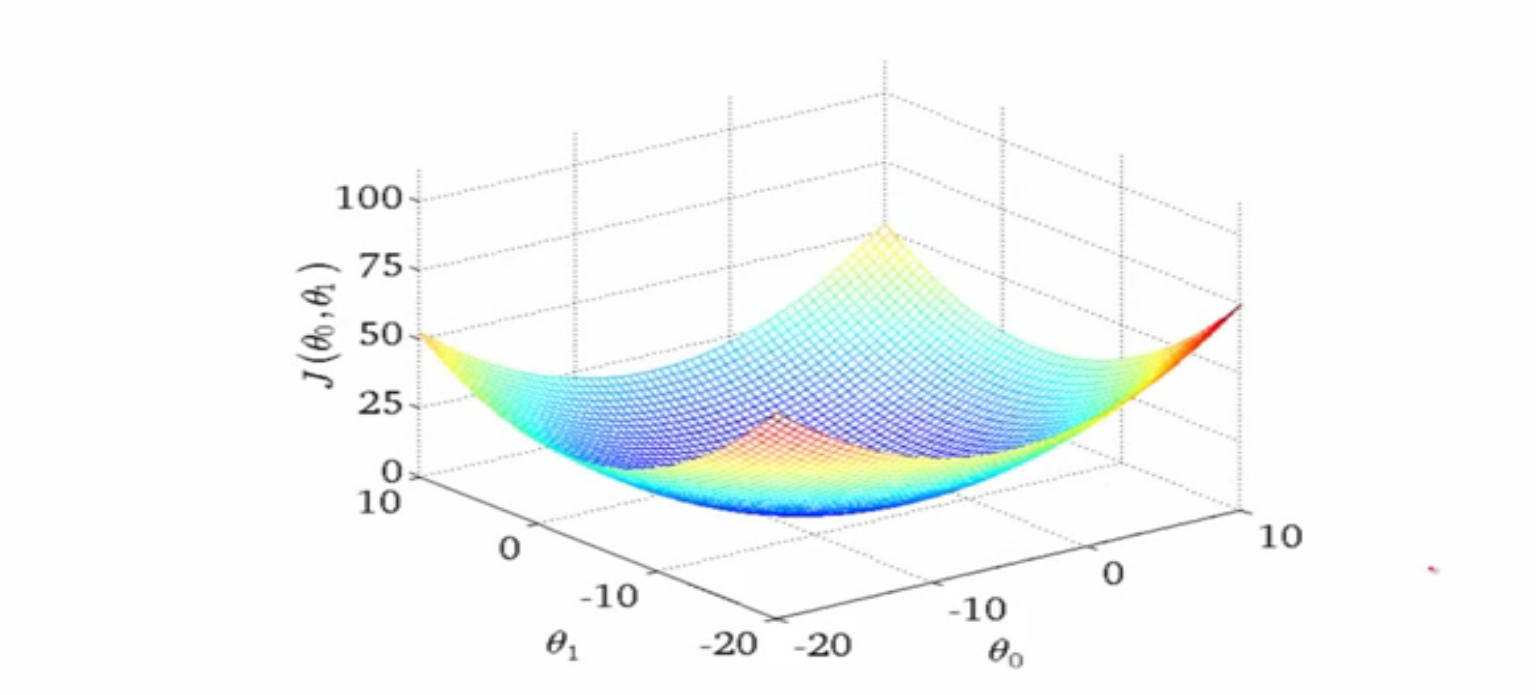
Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

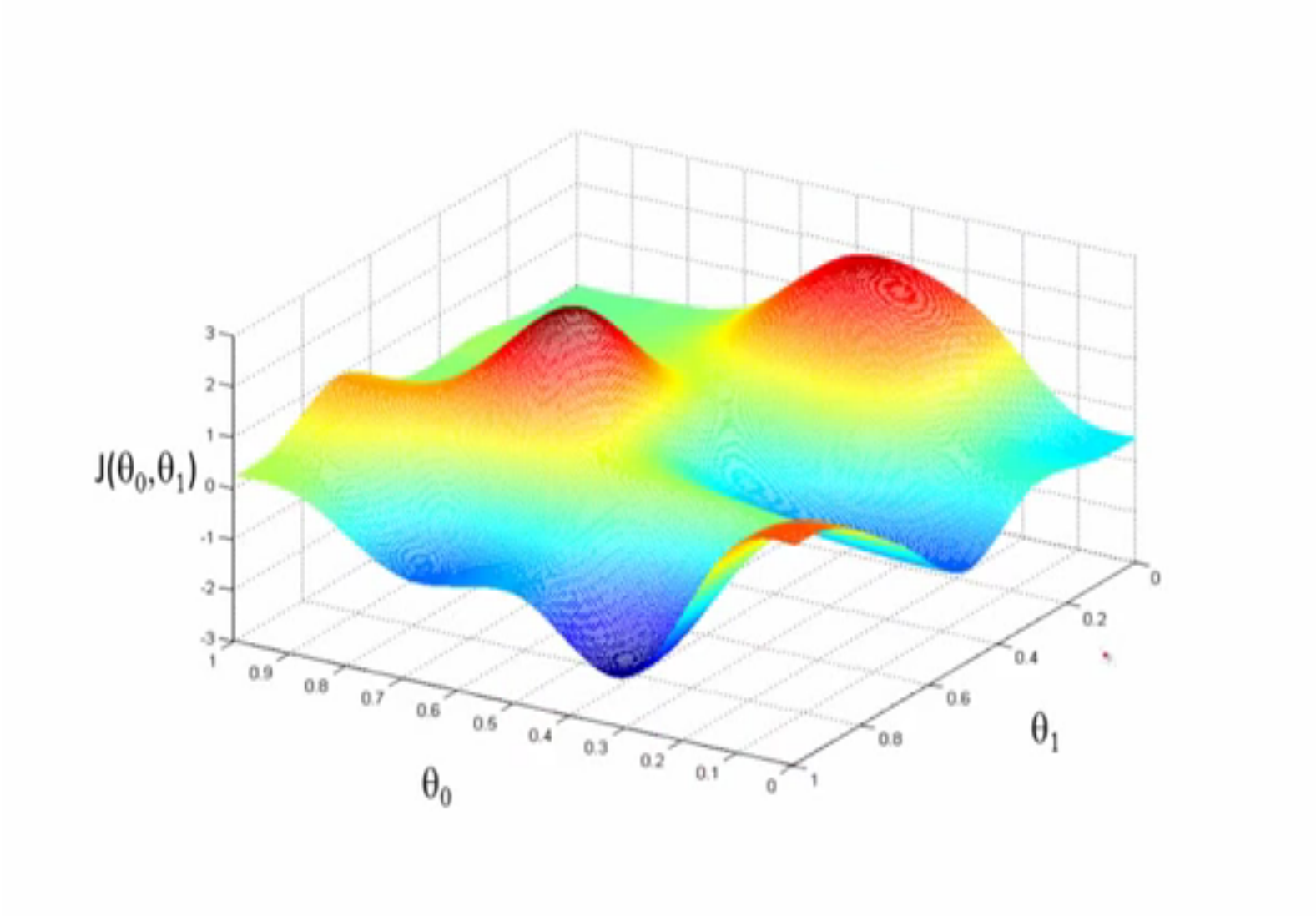
- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

Gradient Descent Intuition 1
Convex Cost Function



One of the many nice properties of convexity is that any local minimum is also a global minimum

Gradient Decent Intuition 2



Unfortunately, any interesting cost function is likely non-convex

Solving the Optimization Problem
Gradient Descent for Linear Regression

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(for $j = 1$ and $j = 0$)

}

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

The big breakthrough in the 1980s from the Hinton lab was the backpropagation algorithm, which is a way of computing the gradient of the loss function with respect to the model parameters θ

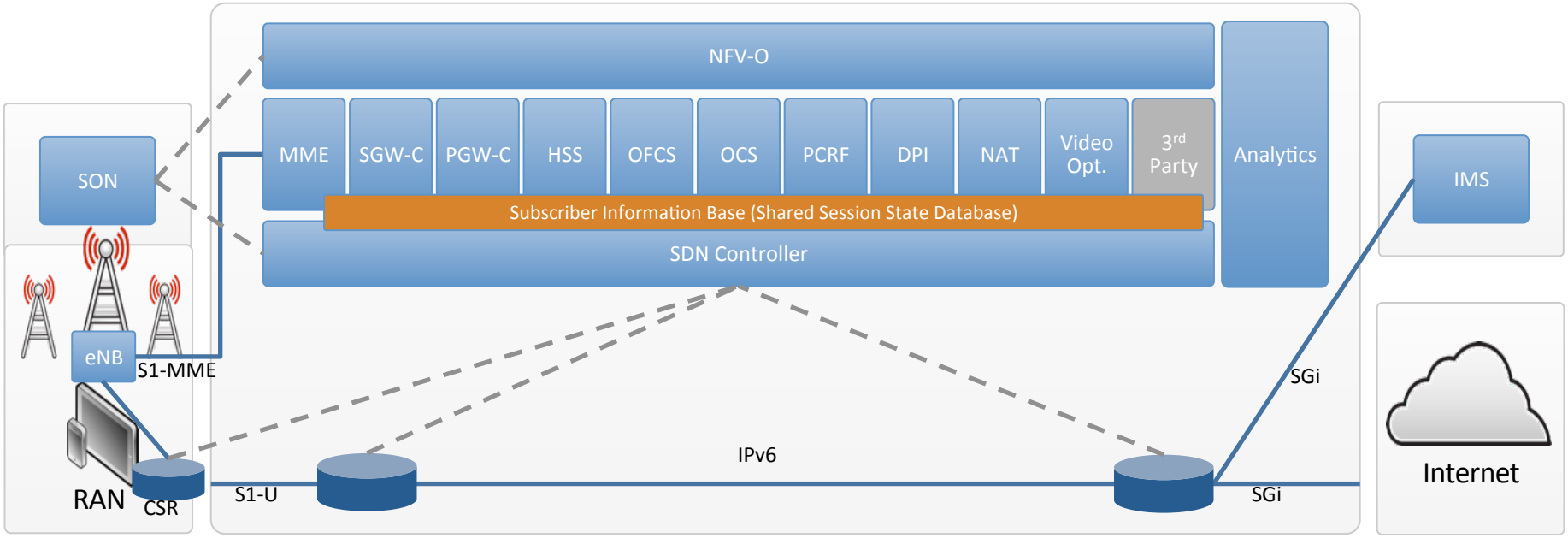
Agenda

- ~~Goals for this Talk~~
- ~~Automation Continuum~~
- ~~Software Defined Intelligence~~
  - ~~Architecture and Pipeline~~
- ~~What is Machine Learning?~~
- Mobile Use Case(s)
- Appendix: How can machine learning possibly work?

Now, How About Mobile Use Cases?

- Mobile ideally suited to SDN and Machine Learning
- Can we infer properties of paths/equipment/users we can't directly see?
  - Likely living in high-dimensional space(es)
  - i.e., those in other domains
- Other inference tasks?
  - Aggregate bandwidth consumption
  - Most loaded links/congestion
  - Cumulative cost of path set
  - Uncover unseen correlations that allow for new optimizations
- How to get there from here
  - Applying Machine Learning to the Mobile spacerequires understanding the problem you want to solve and what data sets you have

A Future State Mobile Architecture



- Control Functions Integrated into NFV, Bearer Functions Integrated into SDN
- Enhanced NB and SB APIs in SDN Controller
- SGW-C and PGW-C maintain 3GPP-compliant external interfaces (S1-U, S5, S11, SGi, S7/Gx, Gy, Gz)
- Integrated Security (Firewall, NAT), removal of physical boundary constraints
- Session State Convergence: Subscriber Management delivered via shared columnar/hybrid database
- Integrated SON + SDN + NFV-O for Radio + Network + Datacenter policy convergence
- Open APIs (Database, Controller, Orchestrator) for 3rd Party Applications

Slide courtesy Kevin Shatzkamer

A Few Principles of Future Mobile Architectures

- Elastic (*for the variance*)
  - Access: Baseband Processing (Cloud RAN), RAN Controllers (Cloud Controllers)
    Core: Evolved Packet Core, Video Optimization, Deep Packet Inspection, NAT, Firewall, VPN
    Services: VoLTE/IMS, Video, CDN, Policy, Identity
    SDP: APIs, M2M
    Hardware-independence + Virtualization + VM Mobility
- Scalable (*for the aggregate*)
  - Highly distributed bearer plane
    Independent control plane (inline or centralized)
    Policy + Orchestration = Subscriber + Resource Optimization
- ***Dynamic (Evolving to Self-Organizing)***
  - Big data analytics models unpredictability in Aggregates and Variances
    Dynamic decisions (manual or automatic intervention) based on analytics
    Adaptable routing/forwarding decisions that follow mobility events (subscribers, content, identity, services, applications, virtual machines)
- Cost-Effective (*OPEX and CAPEX*)

© 2014 BROCADE COMMUNICATIONS SYSTEMS, INC. CONFIDENTIAL—FOR INTERNAL USE ONLY

Mobile Data Sets

- Assume we have labeled data set
  - $\{(X^{(1)},Y^{(1)}),\ldots,(X^{(n)},Y^{(n)})\}$
    - Where $X^{(i)}$ is an m-dimensional vector, and
    - $Y^{(i)}$ is usually a k dimensional vector, $k < m$

$$\mathbf{X} \in \mathbb{R}^m$$

$$\mathbf{Y} \in \mathbb{Z}^k$$

- Strawman X (the network has this information, and much much more)

- $X^{(i)}$ = (Path end points,
  Desired path constraints,
  Signal impairment,
  Computed path,
  Aggregate path constraints (e.g. path cost),
  Minimum cost path,
  Minimum load path,
  Maximum residual bandwidth path,
  Aggregate bandwidth consumption,
  Load of the most loaded link,
  Cumulative cost of a set of paths,
  (some measure of buffer occupancy),
  …,
  Other (possibly exogenous) data)

- If we have $Y^{(i)}$'s are a set of classes we want to predict, e.g., congestion, latency, …

What Might the Labels Look Like? (sparseness)

$$\mathbf{Y} = \begin{bmatrix} Congestion \\ Latency \\ Class2 \\ Class3 \\ Class4 \\ \dots \end{bmatrix} \quad \xrightarrow[\text{(instance)}]{} \quad Y^{(i)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \dots \end{bmatrix}$$

Making this Real
(what do we have to do?)

- Choose the labels of interest
  - What are the classes of interest, what might we want to predict?
- Get the data sets (this is always the "trick")
  - Labeling?
  - Split into training, test, cross-validation
    - Avoid generalization error (bias, variance)
  - Avoid data leakage
- Choose a model
  - I would try supervised DNN
    - We want to find "non-obvious" features, which likely live in high-dimensional space
- Write code
  - Then write more code
- Test on (previously) unseen examples
- Iterate

Issues/Challenges

- Is there a unique model that Mobile Oracles would use?
  - Unlikely → online learning
  - Ensemble learning, among others

- Mobile is a *non-perceptual* tasks (we think)
  - Does the Manifold Hypothesis hold for non-perceptual data sets?
  - Seems to (Google PUE, etc)

- Unlabeled vs. Labeled Data
  - Most commercial successes in ML have come with deep supervised learning → labeled data
  - We don't have ready access to large labeled data sets (always a problem)

- Time Series Data
  - With the exception of Recurrent Neural Networks, most ANNs do not explicitly model time (e.g., Deep Neural Networks)
  - Flow data/sampling

- Training vs. {prediction,classification} Complexity
  - Stochastic (online) vs. Batch vs. Mini-batch
  - Where are the computational bottlenecks, and how do those interact with (quasi) real time requirements?

Q & A

(or we can take a look at how ML could possibly work)

Thanks!

How Can Machine Learning Possibly Work?

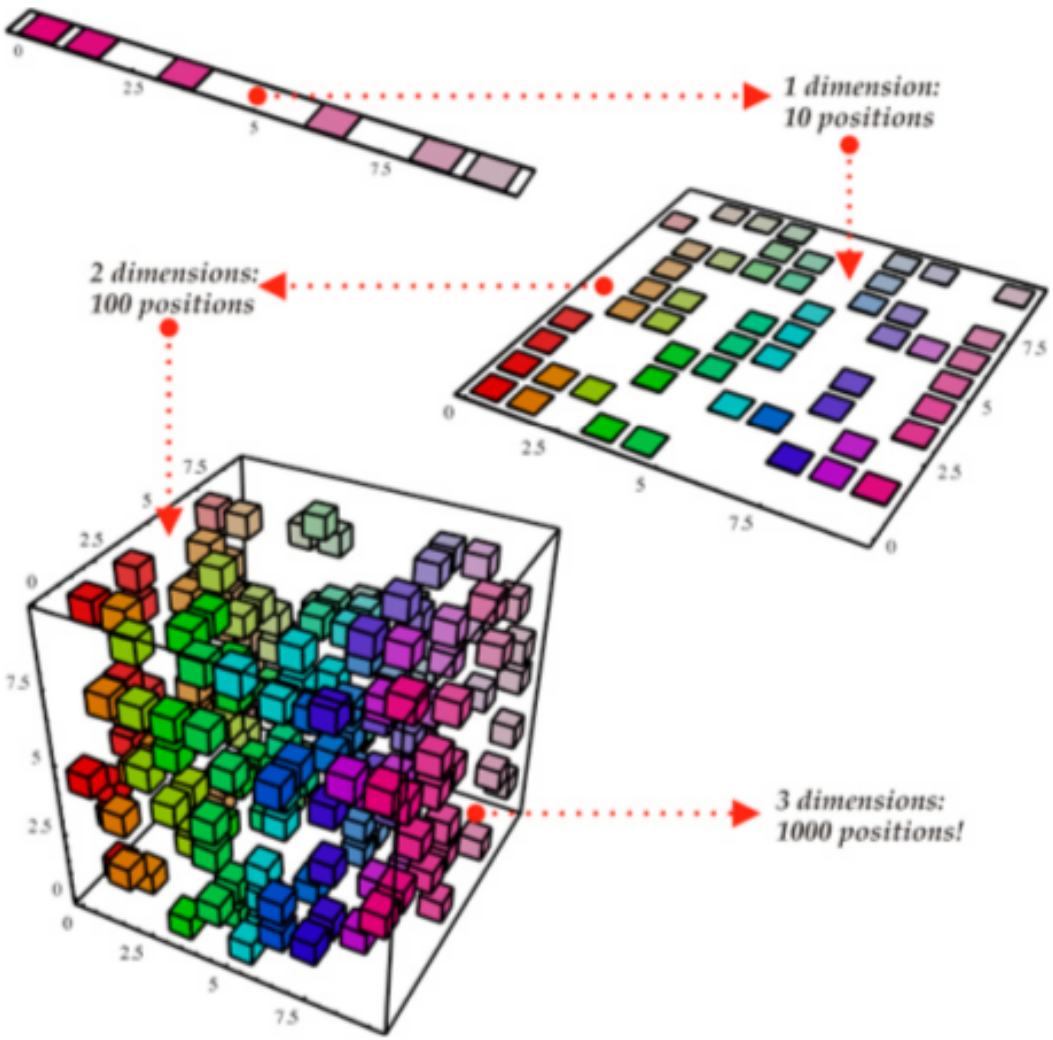
- We want to build statistical models that **generalize to unseen cases**
- What assumptions do we need to do this (essentially predict the future)?
- 4 main "prior" assumptions are (at least) required
  - Smoothness
  - Manifold Hypothesis
  - Distributed Representation/Compositionality
    - Compositionality is useful to describe the world around us efficiently → distributed representations (features) are meaningful by themselves.
    - Non-distributed → # of distinguishable regions linear in # of parameters
    - Distributed → # of distinguishable regions grows almost exponentially in # of parameters
      - Each parameter influences many regions, not just local neighbors
    - Want to generalize non-locally to never-seen regions
  - Shared Underlying Explanatory Factors
    - The assumption here is that there are shared underlying explanatory factors, in particular between p(x) (prior distribution) and p(Y|x) (posterior distribution). Disentangling these factors is in part what machine learning is about.
- Before this, however: What is the problem in the first place?

Why This Is Hard

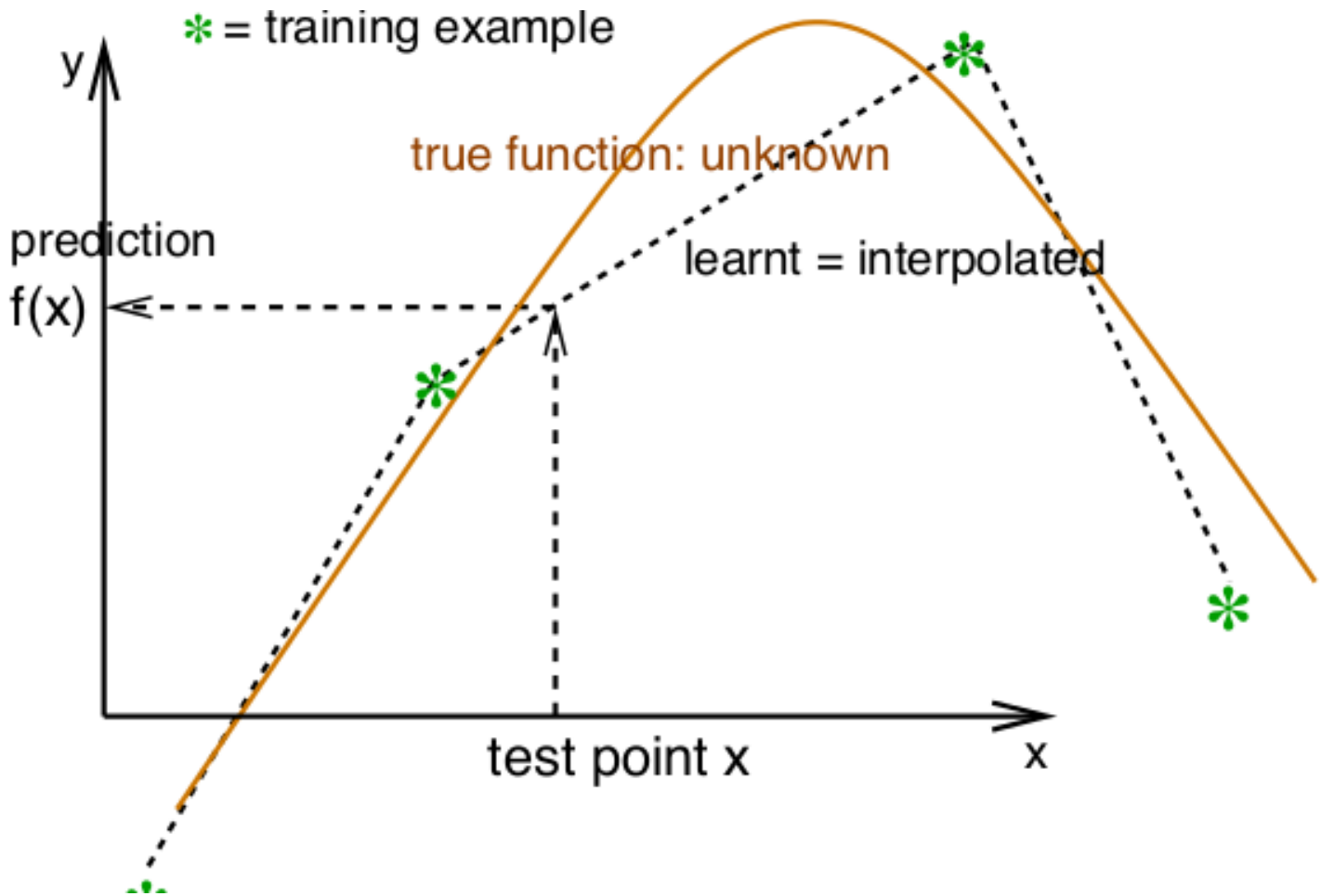
The Curse Of Dimensionality

To generalize locally, need representative examples for all relevant variations!

Classical solution: hope for a smooth enough target function, or make it smooth by handcrafting good features / kernel



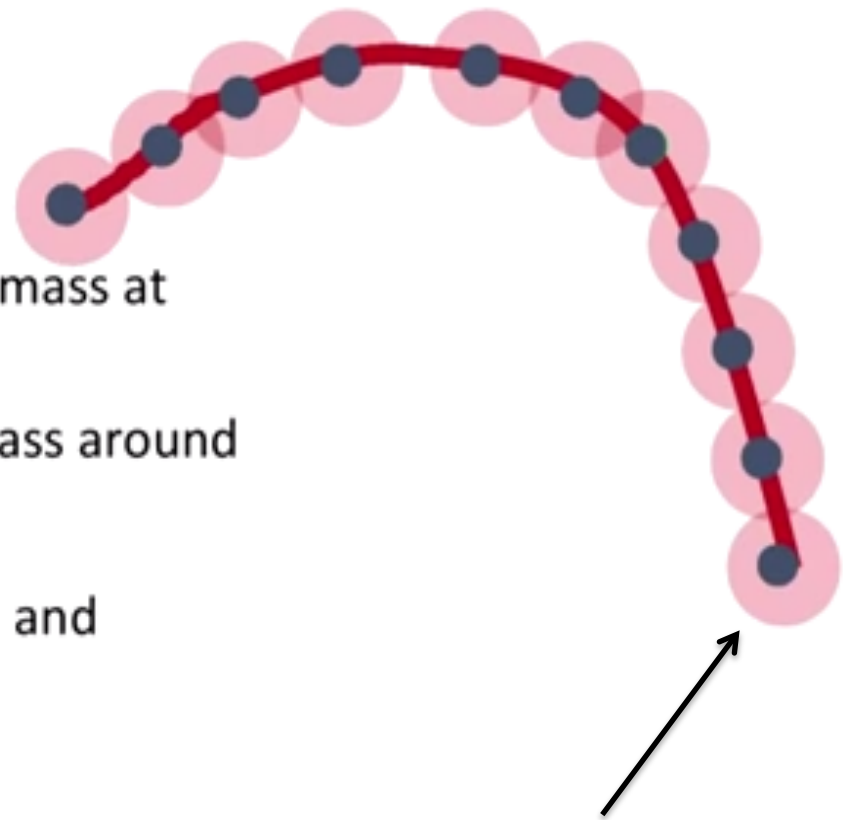
So What Is Smoothness?



Smoothness assumption: If **x** is geometrically close to **x'** then **f(x) ≈ f(x')**

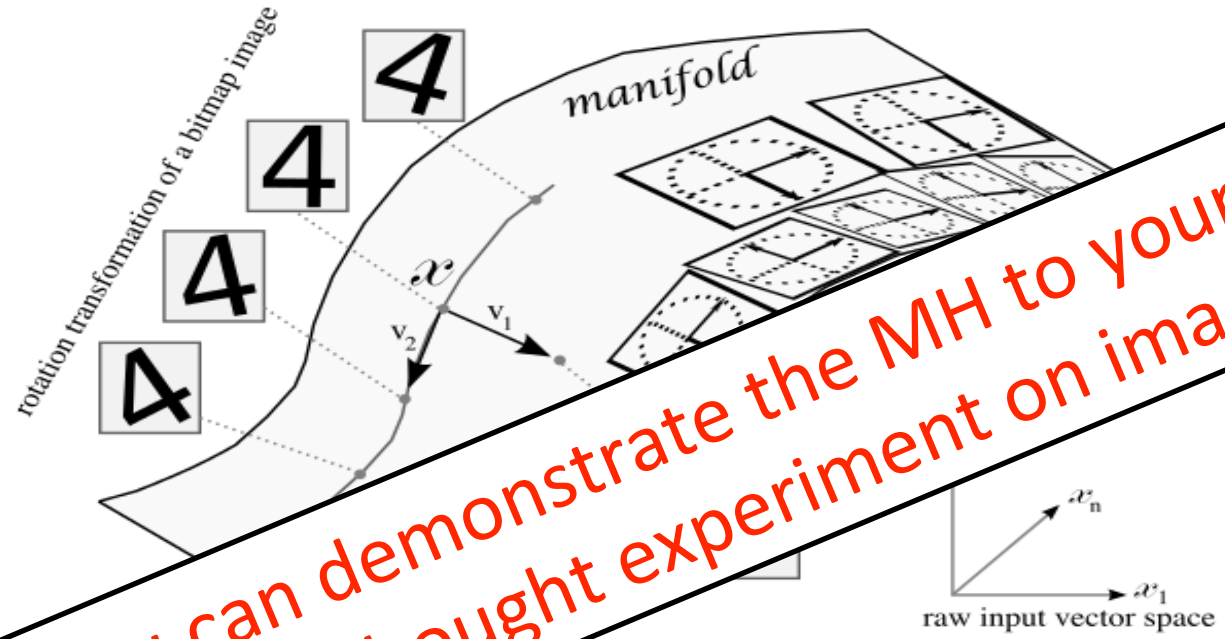
Smoothness, basically…

- Empirical distribution: mass at training examples
- Smoothness: spread mass around
- Insufficient
- Guess some 'structure' and generalize accordingly



Probability mass **P(Y=c|X;θ*)***

Manifold Hypothesis

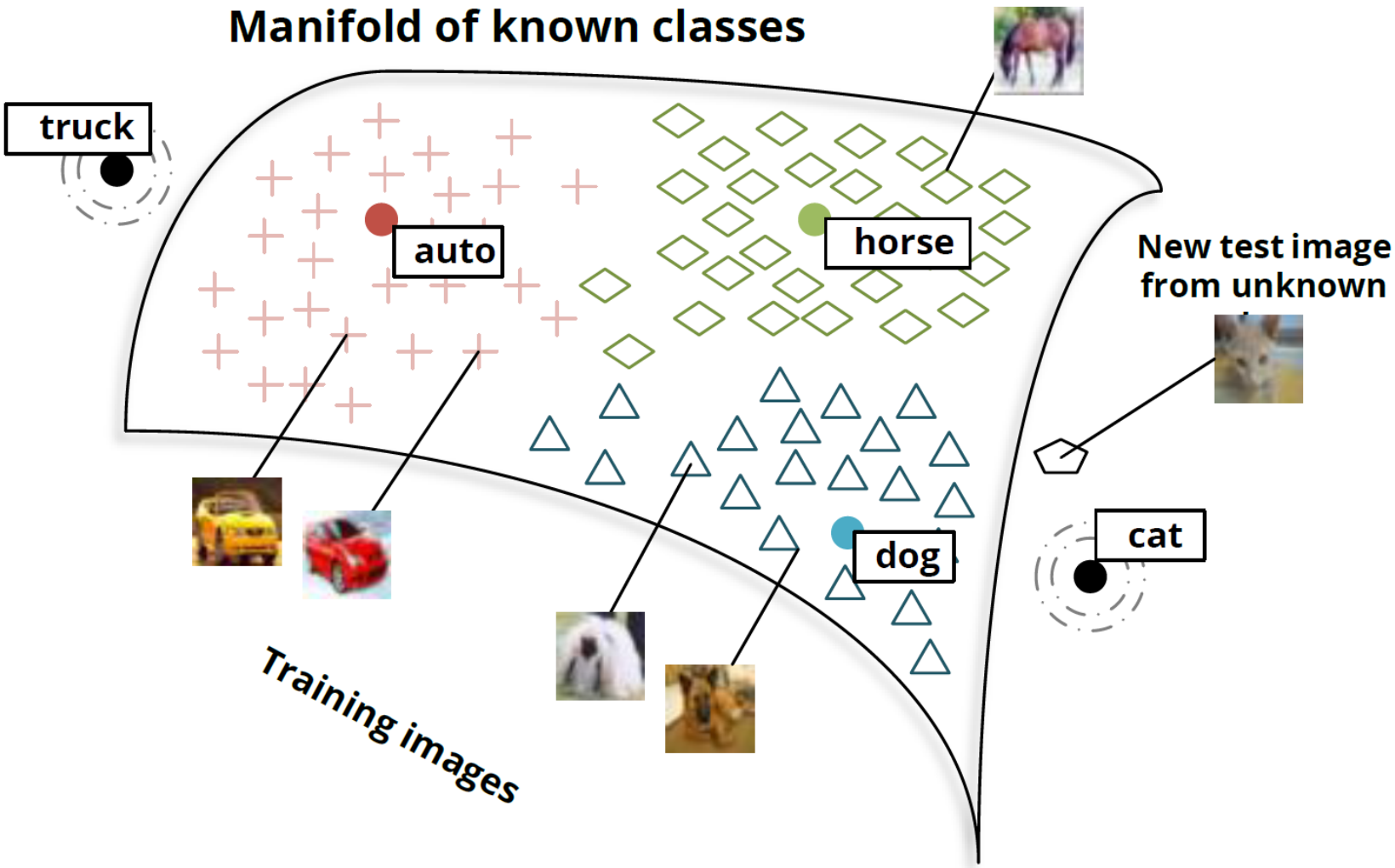
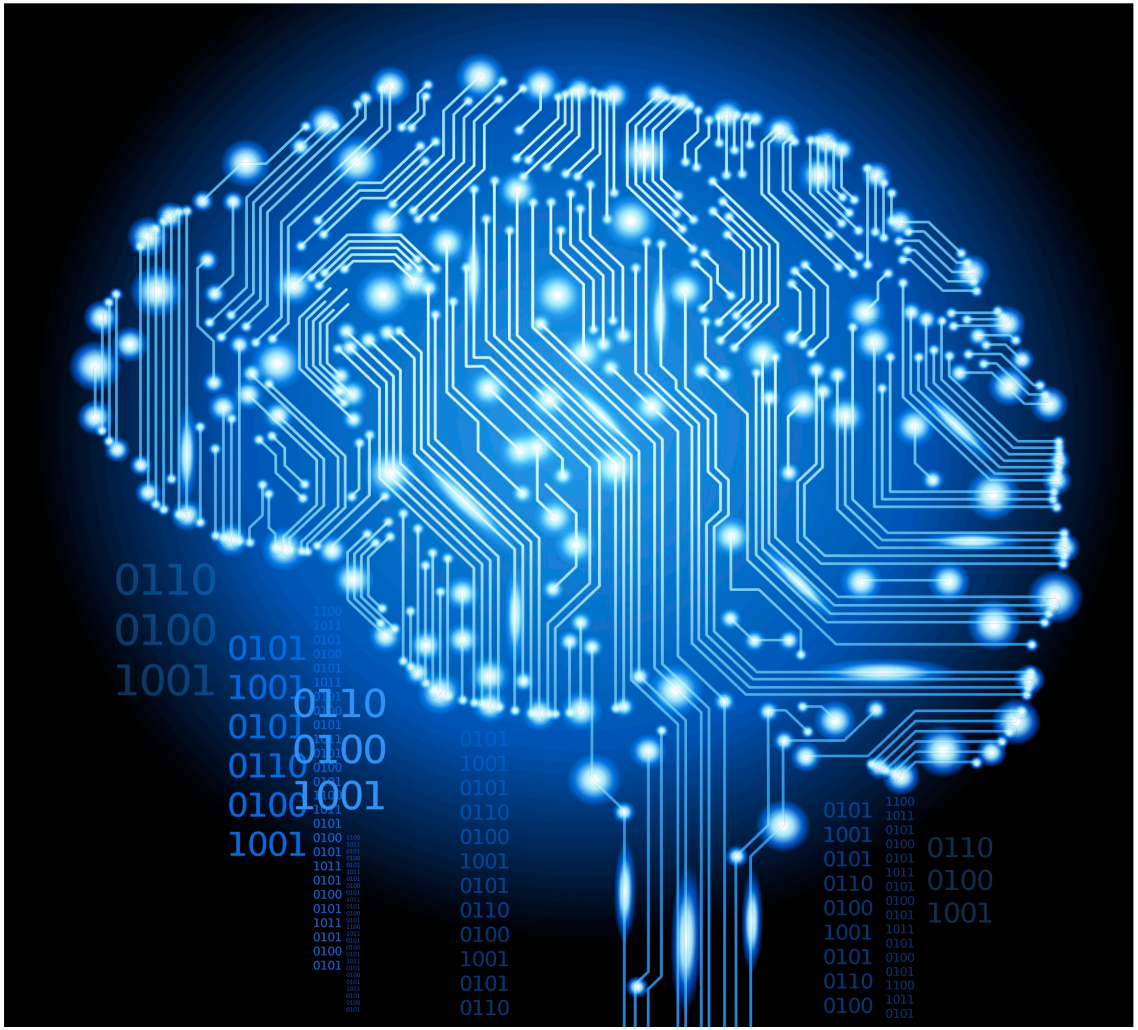


BTW, you can demonstrate the MH to yourself with a simple thought experiment on image data...

The Manifold Hypothesis states that ***natural data*** forms lower dimensional manifolds embedding space. Why should this be? Well, it seems that there are both theoretical and experimental reasons to suspect that the Manifold Hypothesis is true.

So if you believe that the MH is true, then the task of a machine learning classification algorithm is fundamentally to separate a bunch of tangled up manifolds.

Manifolds and Classes



Backup