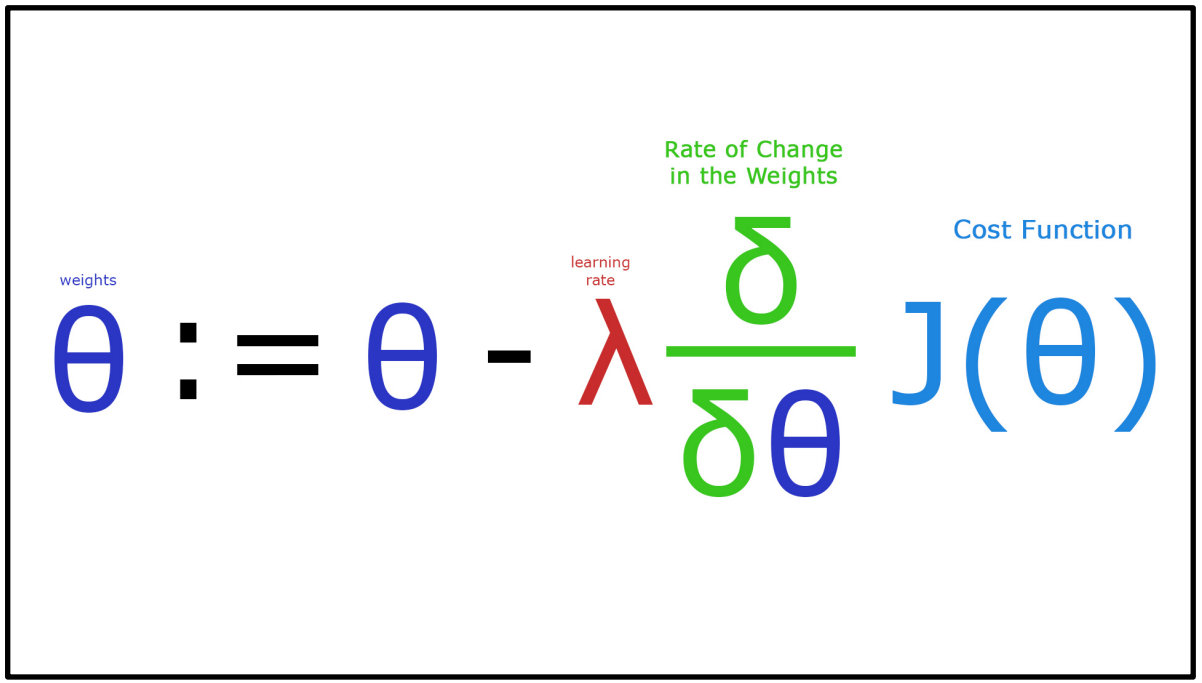
Machine Intelligence and Networking

Challenges, Opportunities and Realities



David Meyer
Brocade Chief Scientist, VP and Fellow
Senior Research Scientist, Computer Science, University of Oregon
dmm@{brocade.com,uoregon.edu,1-4-5.net,..}
http://www.1-4-5.net/~dmm/ml/talks/2016/cor_ml4networking.{pptx,pdf}
Network Machine Learning Research Group
IETF 97
13 – 18 Nov 2016
Seoul, Republic of Korea

Goals For This Talk

The goals for this talk are to briefly review the state of the art in Machine Learning as applied to data networking, with particular focus on

- What is practical today
- What are the challenges[1], and
- What are the opportunities

We'll go easy on the math here, but if you're interested in that part of all of this see http://www.1-4-5.net/~dmm/ml for some introductory material.

1 Noting the these challenges are both technological and socio-economic.

You might be surprised but what is going to drive innovation in the enterprise and in the public cloud is machine learning.

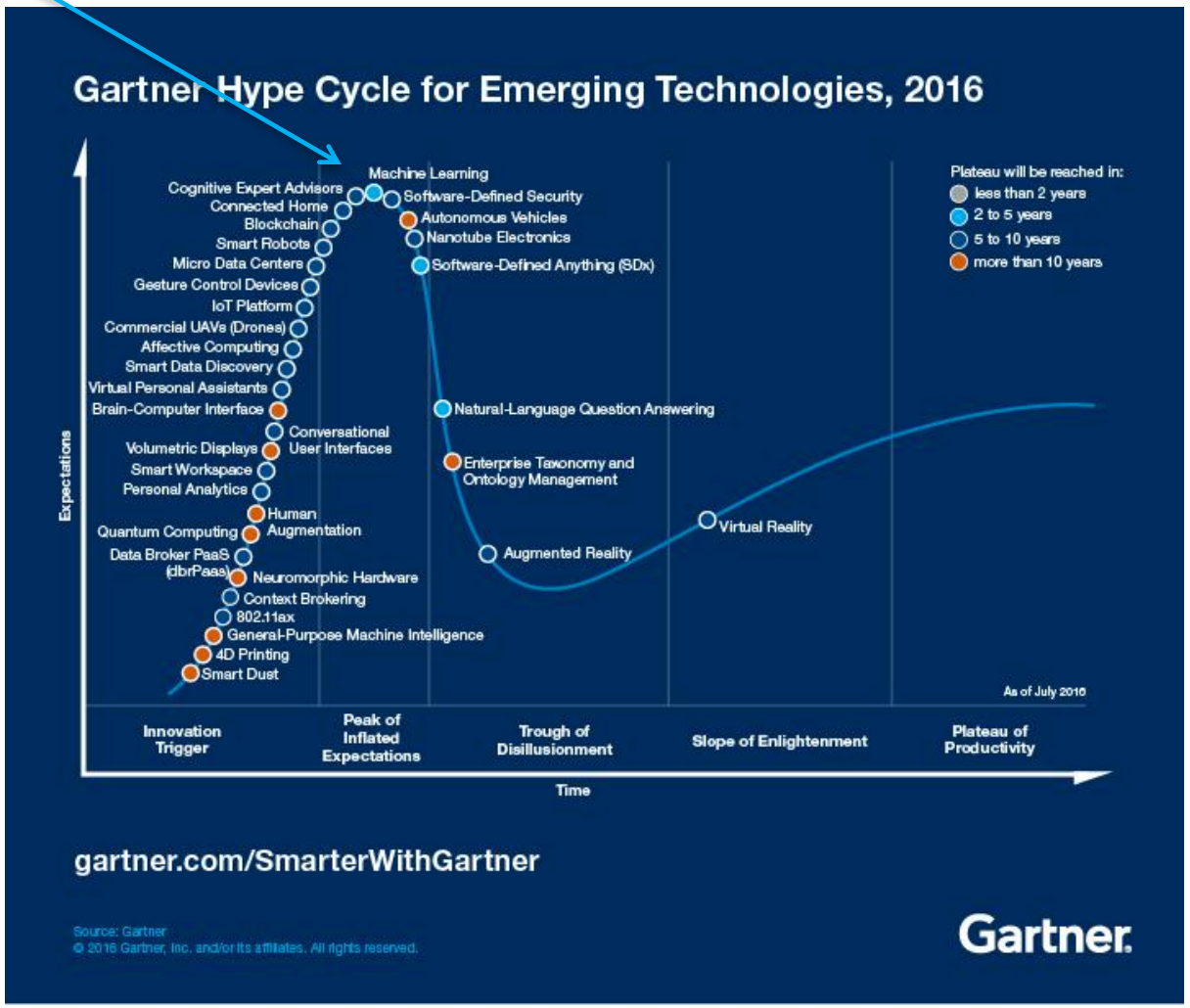
Bill Coughran, Sequoia Capital #ONUGSpring16

Machine Learning is the way we are going to automate your automation.

Chris Wright, RedHat CTO #RHSummit

But Be Careful…

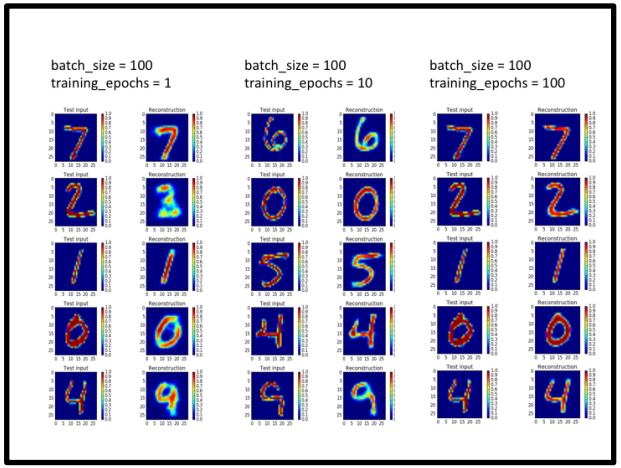
You are here (again)



https://en.wikipedia.org/wiki/AI_winter

Agenda

- What Is Machine Learning?
- What Can We Do Today, And What Are The Challenges?
- What's On the Horizon?
- Technical explanations/code
  - https://github.com/davidmeyer/ml
  - http://www.1-4-5.net/~dmm/ml



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.

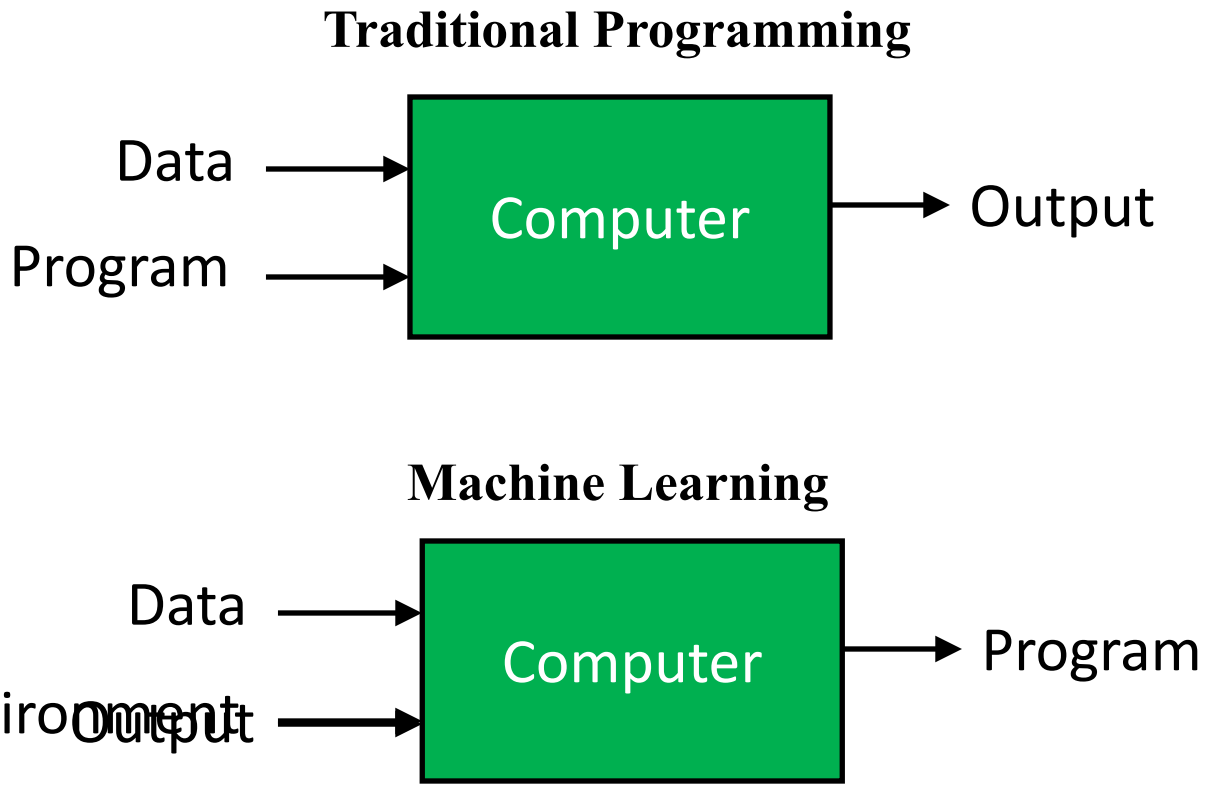
-- Andrew Ng

- Said another way, we want to discover the *Data Generating Distribution and the associated process* (or processes) that underlies the data that 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 (BTW, how is this possible?)*
  - as opposed the accuracy on the training set (note: overfitting)

Same Thing Said In Cartoon Form

(Three main types of Machine Learning)



So When Would We Use Machine Learning?

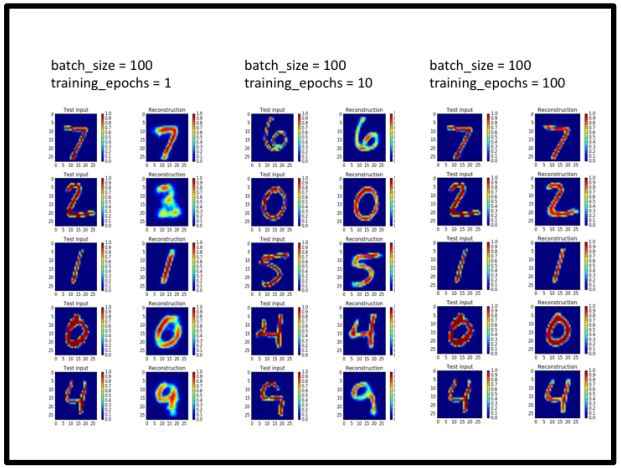
- When patterns exists in our data
  - Even if we don't know what they are
    - Or perhaps especially when we don't know what they are
    - Or if they are just noise
- We can not pin down the functional relationships mathematically
  - Else we would just code up the algorithm
  - Neural networks as function approximators
    - Need this for scale
- 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
    - All kinds of open source frame works, e.g., https://www.tensorflow.org/ or http://torch.ch/ or …
    - Well-worn libraries http://scikit-learn.org/stable/ (many others)

Examples of Machine Learning Problems

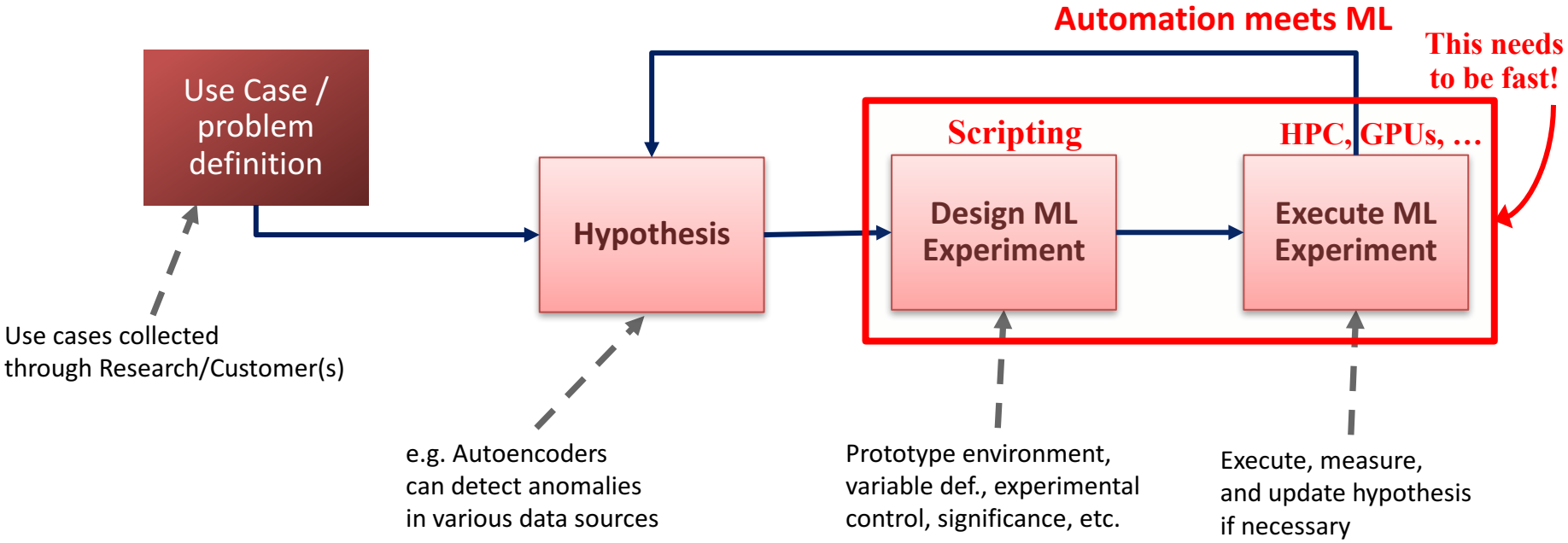
- Pattern Recognition
  - Facial identities or facial expressions
  - Handwritten or spoken words (e.g., Siri)
  - Medical images
  - Sensor Data/IoT
  - Recommender Systems
- Optimization
  - Many parameters have “hidden” relationships that can be the basis of optimization
- Pattern Generation
  - Generating 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
  - Network events
  - …

Agenda

- What Is Machine Learning?
- What Can We Do Today, And What Are The Challenges?
- What's On the Horizon?
- Technical explanations/code
  - https://github.com/davidmeyer/ml
  - http://www.1-4-5.net/~dmm/ml



First, What Does The Scientific Method Look Like When Applied To Machine Learning?



"If you want to increase your success rate, double your failure rate"

Thomas Watson Sr. (founder of IBM)

Original slide courtesy Armin Wasicek

Where Have All The Successes Been?

Perceptual Tasks

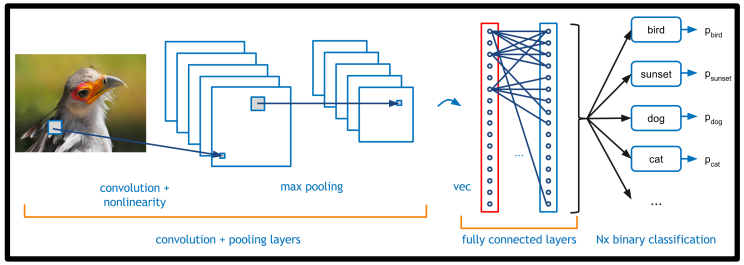
- Vision
- NLP
- Robotics
- …

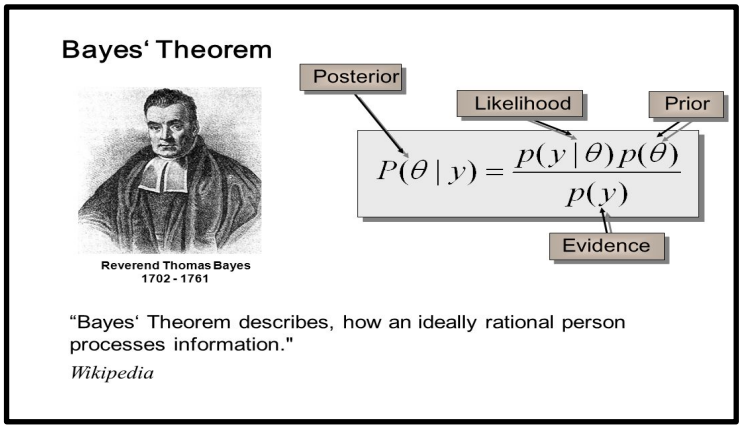
Statistical Approaches

Some Anomaly Detection

What has driven all of this progress?

- Theoretical breakthroughs in 2006
- Compute
- Data
- Talent pipelines

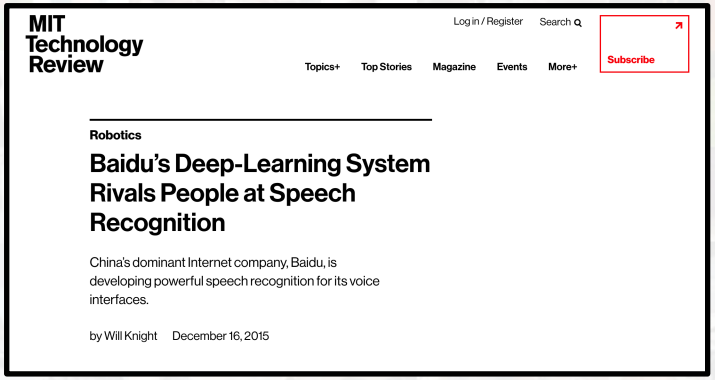




We have seen relatively little progress in the network space. Why?

Data

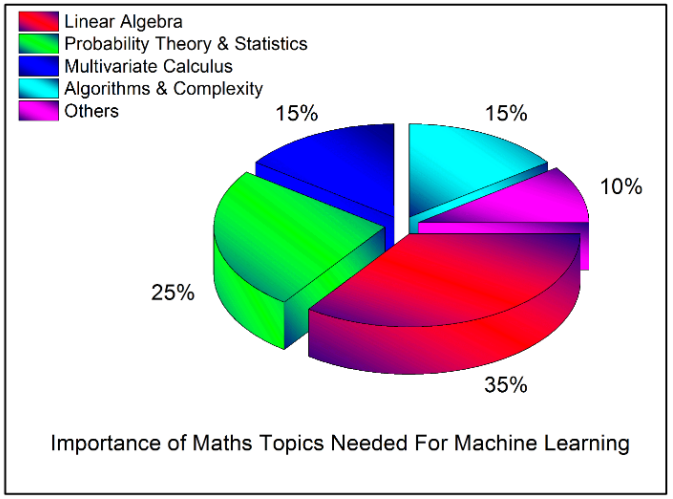
- *Standardized* data sets have been a crucial factor in the success of ML
  - http://yann.lecun.com/exdb/mnist/
  - http://image-net.org/
  - http://allenai.org/data.html
  - …
  - *Data is the rocket fuel of Machine Learning* – Andrew Ng
- Allows for direct comparison of learning and inference algorithms
- The result has been the steady ratcheting down of error rates on perceptual tasks to super-human levels
  - Object/Scene recognition
  - NLP/Voice recognition and generation
  - AlphaGo
  - …



- We have nothing like this for networking
  - Many/most data sets are toy, noisy, incomplete, unnormalized, …
  - Many data sets are proprietary
  - Much data non-iid
  - Most of our data sources (e.g., netflow) are representations of network data that were **not** built for ML
    - → Effective ML systems that rely on network data will require new learning algorithms and abstractions
  - There is no standard way to integrate network data with other data sources like CPU usage, memory, ...

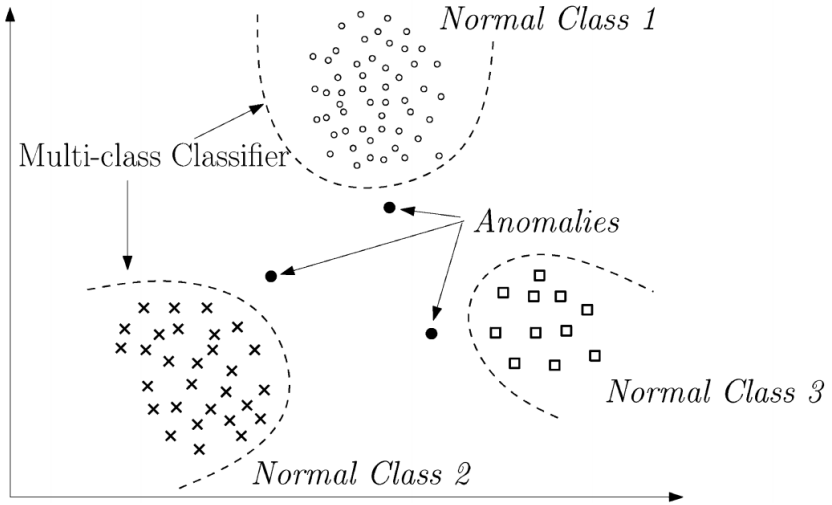
Talent Pipelines

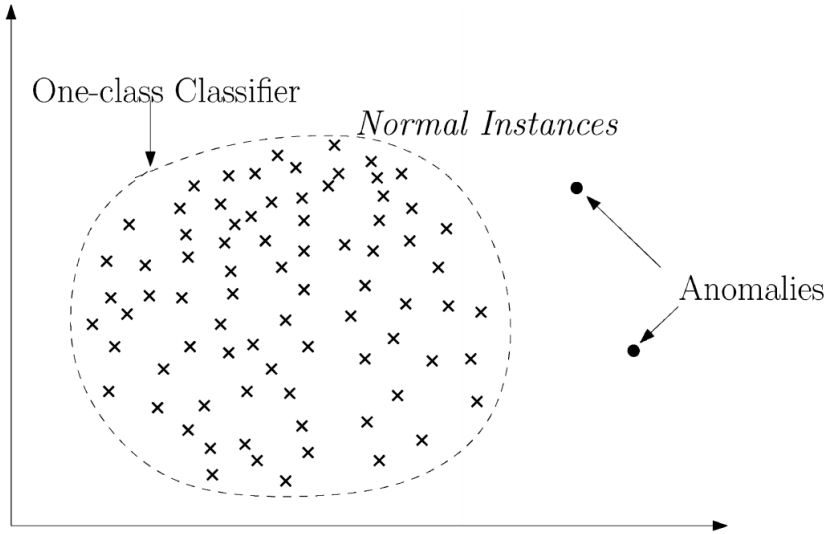
- ML is a form of Applied Mathematics
  - Moreover, it is a multi-disciplinary empirical science
- Network engineers (us!) typically don't have backgrounds that include the kinds of mathematical training and experience that are essential in the ML space
- Effective ML requires additional skills
  - Ability to rapidly learn new concepts
  - Data modeling and evaluation
  - Communications
  - Software engineering and system design
  - ML algorithms and libraries
  - …



- The result is that there is a *serious skills gap*
  - Open source frameworks are easing this problem
  - However, deploying ML at scale still requires intimate understanding of the mathematics
  - Further, most ML {under}graduates train on perceptual tasks

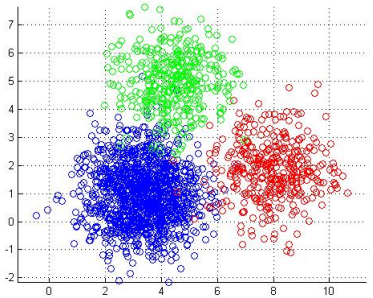
Anomaly Detection as a Classification Problem





What is going on here?

- Data don't fit an explanation model
  - Impossible, assuming the model is correct
- Data do not conform to some *normal* behavior



Clustering
- K-means
- K-NN
- …

PCA

Autoencoders

….

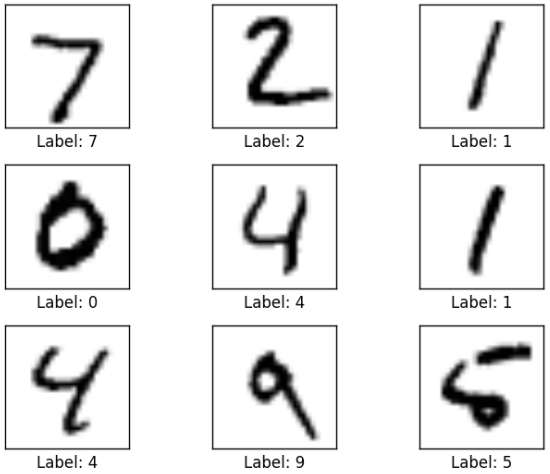
Dimension Reduction

We assume that the anomalous data are generated by a different process than our *baseline* → *stationary distribution*

Original slide courtesy Armin Wasicek

Autoencoder Example

Detecting Anomalies in Handwritten Digits



MNIST: 28x28 Grayscale Handwritten Digit Dataset

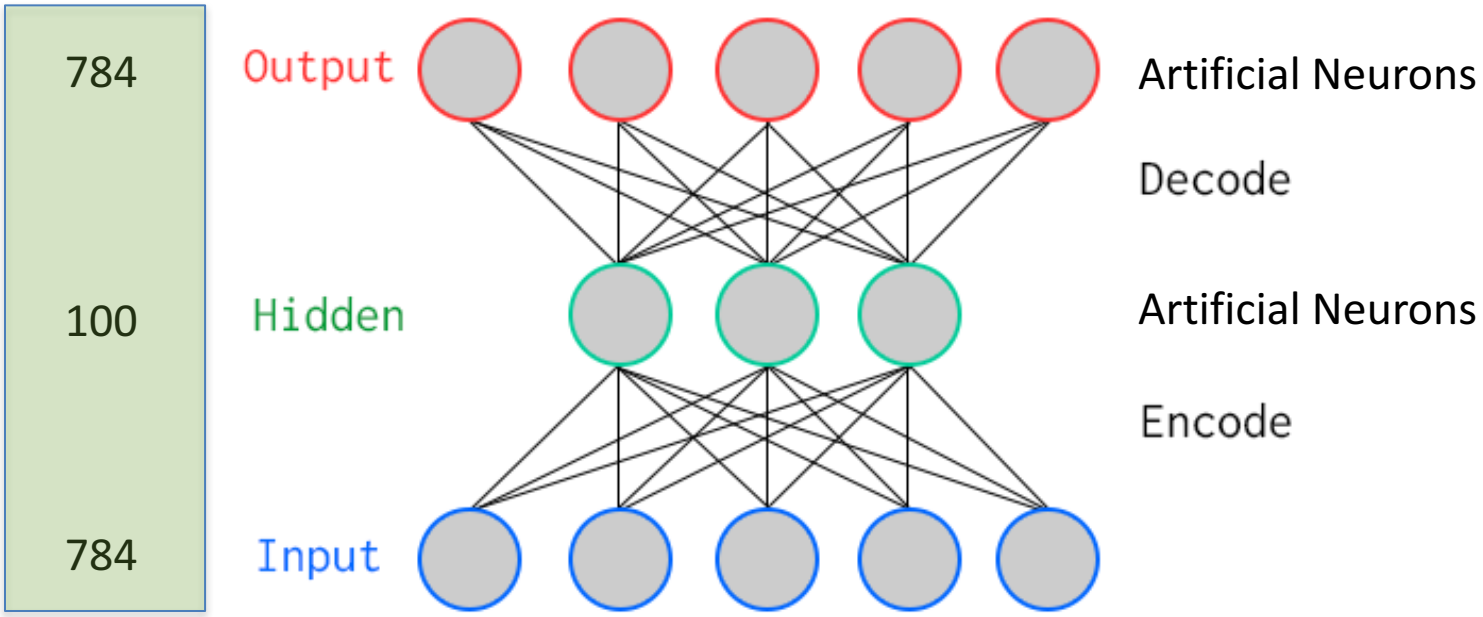
- Training set: 55000 images
- Test set: 10000 images
- Validation set: 5000 images

Features here are pixels (28x28 = 784)

I'm using MNIST here because you easily visualize what is going on. In our case, instead of the input being vectors of {0,1} we will have vectors of counters of various features s/a bandwidth, and other network and host based metrics/KPIs.

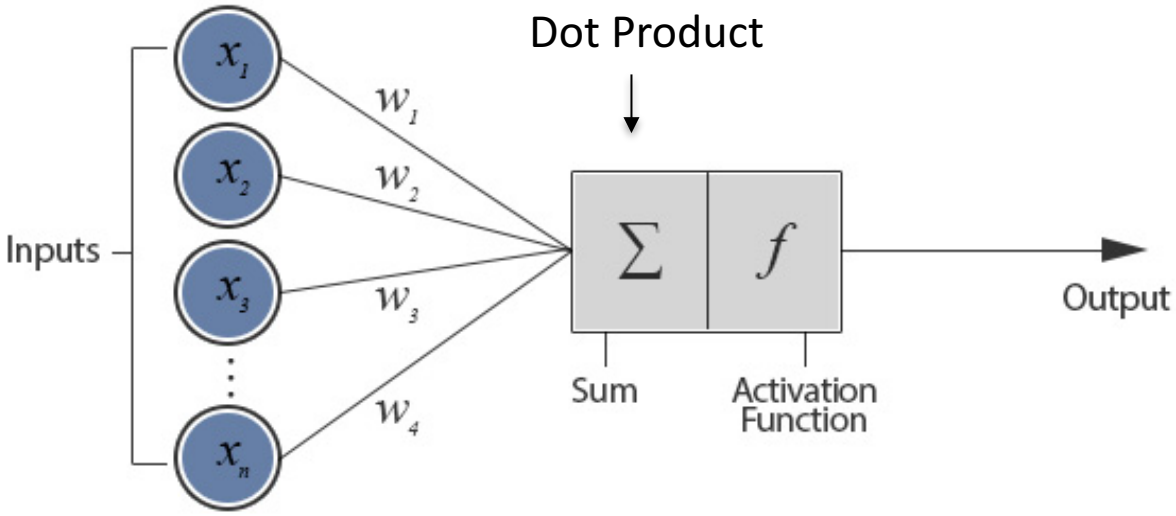
One Way To Learn To Recognize MNIST: Use An Autoencoder

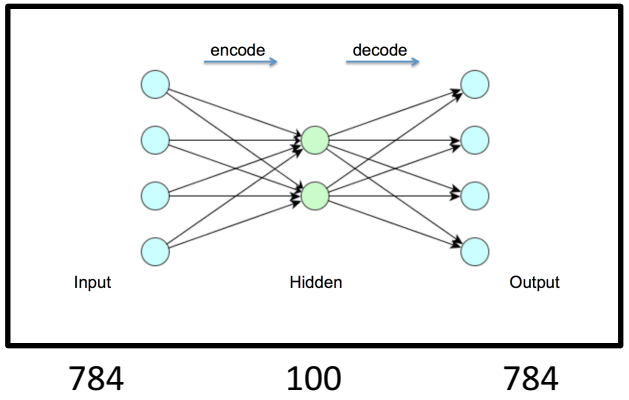
Special Kind of Neural Network



- Key Characteristic: Hidden layer has fewer units than input/output → Compression
- Goal: Minimize reconstruction (decode) error
  - How to define error (loss, cost)?
  - Binary classification: Threshold reconstruction error → normal/abnormal
- Unsupervised learning

But First: What Does A Single Neuron Do?



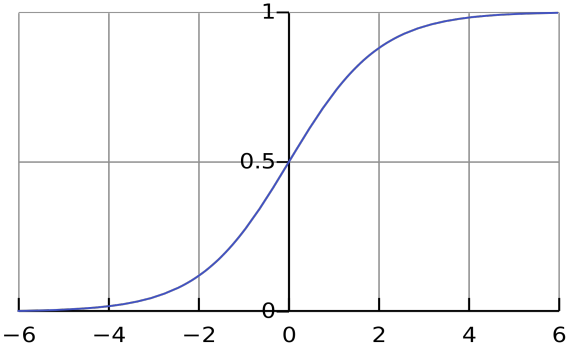


BTW, how many parameters?

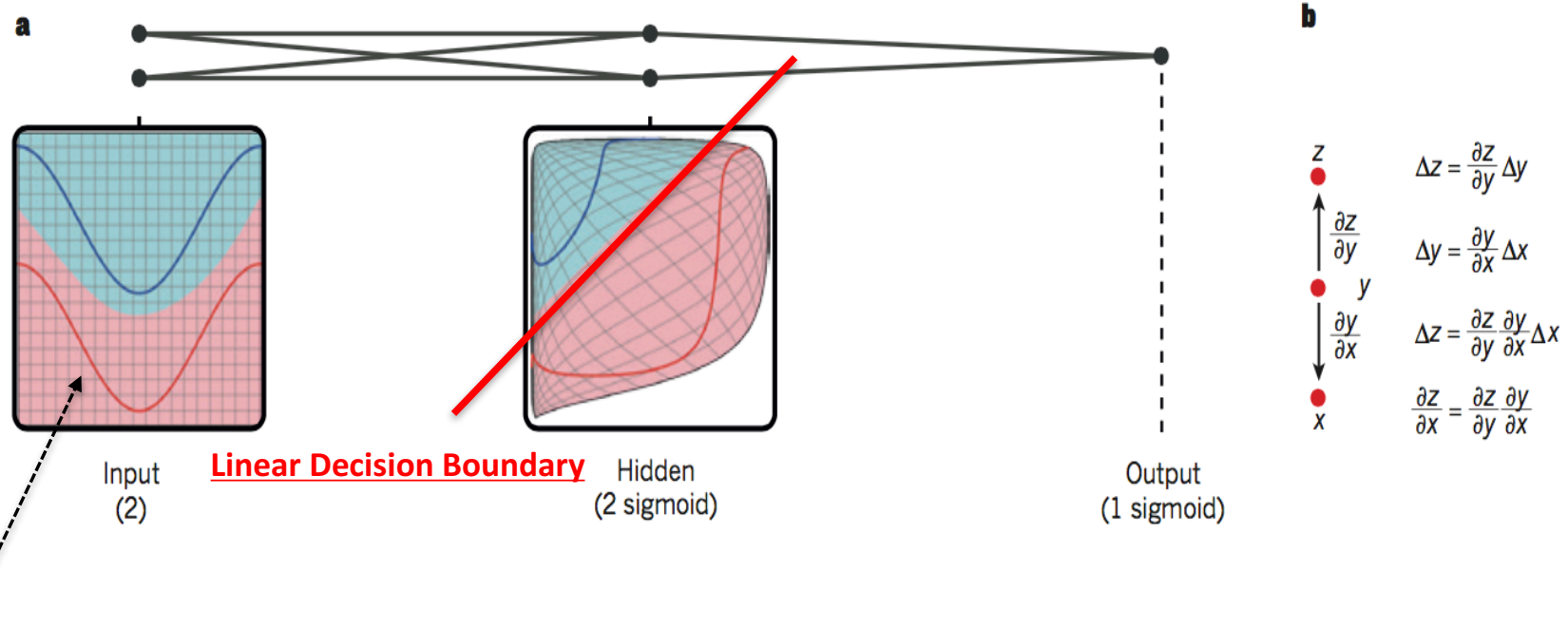
784x100 + 100x784 = 156800

$$\boldsymbol{W}^T\boldsymbol{x} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} w_i x_i = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

$$f(\boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}) = \sigma(\boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}) = \frac{1}{1 + e^{-(\boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b})}}$$



So What is Really Happening Here?



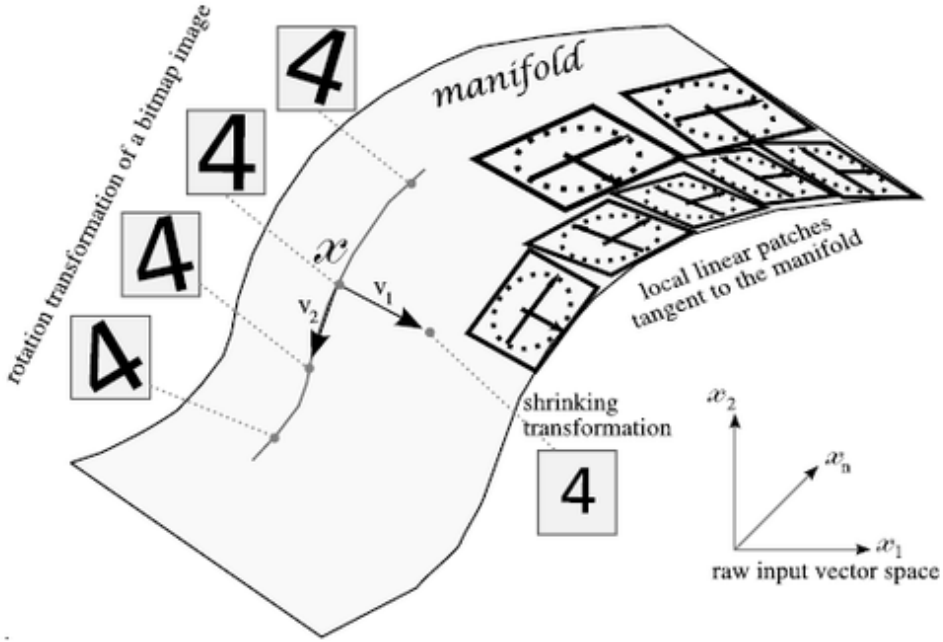
Target function represented by input data is some twisted up manifold

Deep Nets ***disentangle*** the underlying explanatory factors in the data so as to make them *linearly separable*

Graphic courtesy Christopher Olah

Manifolds?

- X is a high dimensional vector ($28^2 = 784$)
- Data concentrated around a lower dimensional manifold



- Hope to find representation *R* of that manifold

Graphic courtesy http://www.deeplearningbook.org

So OK, All Cool, But How Does The Autoencoder Actually Work?

- Encoder $h^{(t)} = f_\theta(x^{(t)}), \{x^{(1)}, \ldots, x^{(T)}\}$

Where h is feature vector or representation or code computed from x

- Decoder

maps from feature space back into input space, producing a reconstruction

$$r = g_\theta(h)$$

attempting to incur the lowest possible reconstruction error $L(x, r)$.

Good generalization means low reconstruction error at test examples, while having high reconstruction error for most other x configurations

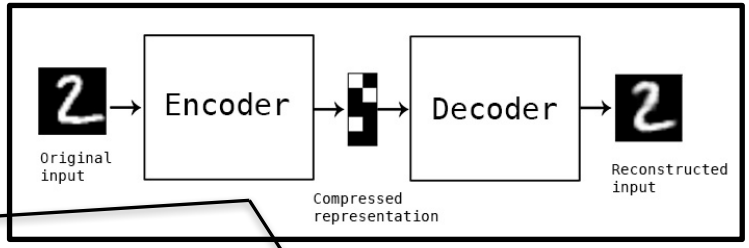
$$h^{(i)} = f_{\theta_e}(x^{(i)}) = \sigma(\theta_e^{\mathrm{T}} x^{(i)} + b_e^{(i)})$$
$$r^{(i)} = g_{\theta_r}(h^{(i)}) = \sigma(\theta_r^{\mathrm{T}} h^{(i)} + b_r^{(i)})$$

$$L(x, r) = \frac{1}{T} \sum_{i=1}^{T} (x^{(i)} - r^{(i)})^2$$

How Hard To Code This Up In Tensorflow?

```
#
#       encoder/decoder
#
#       try tf.nn.sigmoid, tf.nn.relu, etc for nonlinearity
#       nonlinearity=False means transfer function (aka activation funtion)
#       g(x) = x
#
def encoder(x, nonlinearity=False):
    code = tf.add(tf.matmul(x, weights['encoder']), biases['encoder'])
    if nonlinearity:
        code = nonlinearity(code)
    return code

def decoder(code, nonlinearity=False):
    reconstruction = tf.add(tf.matmul(code, weights['decoder']), biases['decoder'])
    if nonlinearity:
        reconstruction = nonlinearity(reconstruction)
    return reconstruction
#
#       get the encoding and decoding operations
#
#       relu seems less efficient here
#
#
#       first encode
#
encoder_op = encoder(X,tf.nn.sigmoid)
#
#       then decode
#
decoder_op = decoder(encoder_op,tf.nn.sigmoid)
#
#       decoder_op is our predicted value (y_pred)
#
y_pred = decoder_op
#
#       y_true is the input X
#
y_true = X
#
reg_losses   = tf.get_collection(tf.GraphKeys.REGULARIZATION_LOSSES)
reg_constant = 0.01
#
if (USE_REGULARIZER):
        error = tf.add(tf.reduce_mean(tf.square(tf.sub(y_true,y_pred))),
                       tf.mul(reg_constant,tf.reduce_sum(reg_losses)))
else:
        error = tf.reduce_mean(tf.square(tf.sub(y_true,y_pred)))
```



$$h^{(i)} = f_{\theta_e}(x^{(i)}) = \sigma(\theta_e^{\mathrm{T}} x^{(i)} + b_e^{(i)})$$

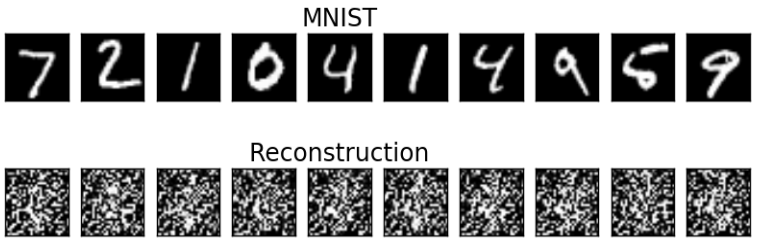
$$r^{(i)} = g_{\theta_r}(h^{(i)}) = \sigma(\theta_r^{\mathrm{T}} h^{(i)} + b_r^{(i)})$$

$$L(x, r) = \frac{1}{T} \sum_{i=1}^{T} (x^{(i)} - r^{(i)})^2$$

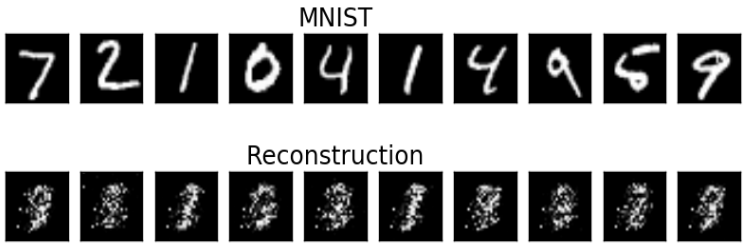
https://github.com/davidmeyer/ml/blob/master/tensorflow/autoencoder.{py,ipynb}

Autoencoder Output

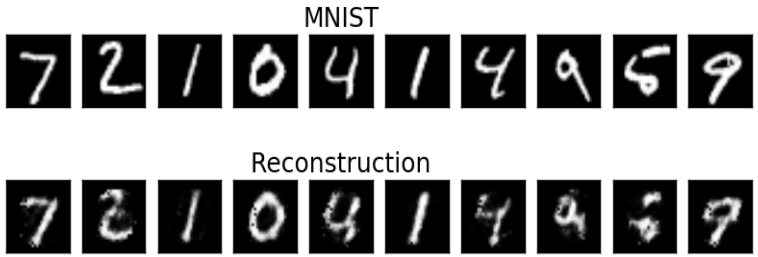
1 example



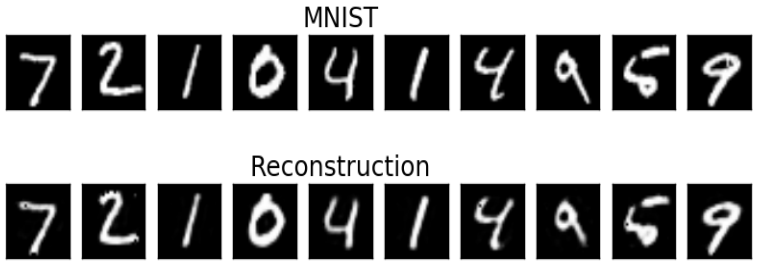
10 examples



100 examples



1000 examples

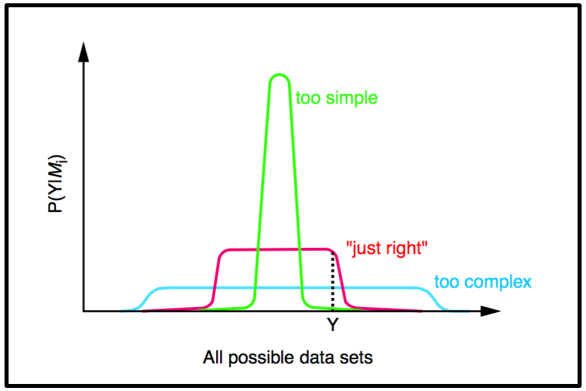


After training, the AE gets low *reconstruction error* on digits from MNIST and high reconstruction error on everything else: It has learned to recognize MNIST

$$L(x, r) = \frac{1}{T} \sum_{i=1}^{T} (x^{(i)} - r^{(i)})^2$$

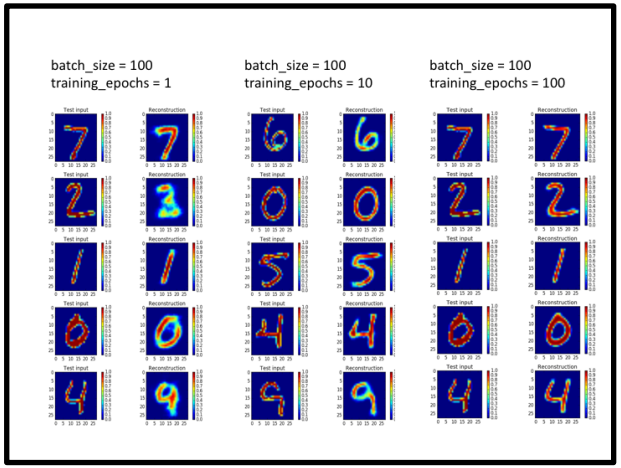
How Much Of This Has Been Applied To Networking?

- Not too much. Many reasons:
- Still early days
- Diverse types of network data
  - Flows, logs, various KPIs, … with no obvious/consistent way to combine
  - Incomplete data sets, non-iid data
  - Network data not designed for ML
- Different models for different data types
  - Still active area of investigation
  - Occam's Razor
- Is there a useful "Theory of Network"?
  - Consider the problem of object recognition/conv nets
  - Transfer learning
  - Markov Chains?
- *Community* challenges: Skill sets, proprietary data sets and use-cases, …
  - Concern about the probabilistic nature of ML

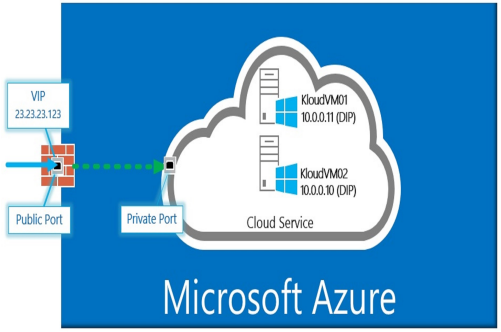


Agenda

- What Is Machine Learning?
- What Can We Do Today, And What Are The Challenges?
- What's On the Horizon?
- Technical explanations/code
  - https://github.com/davidmeyer/ml
  - http://www.1-4-5.net/~dmm/ml



Well, We've Seen All of This



But Now…

techemergence

Search

TECHNOLOGY INTERVIEWS BUSINESS RESEARCH SUBSCRIBE ABOUT

AT&T Predicts Future, Saves Service with Machine Learning – A Conversation with Dr. Mazin Gilbert

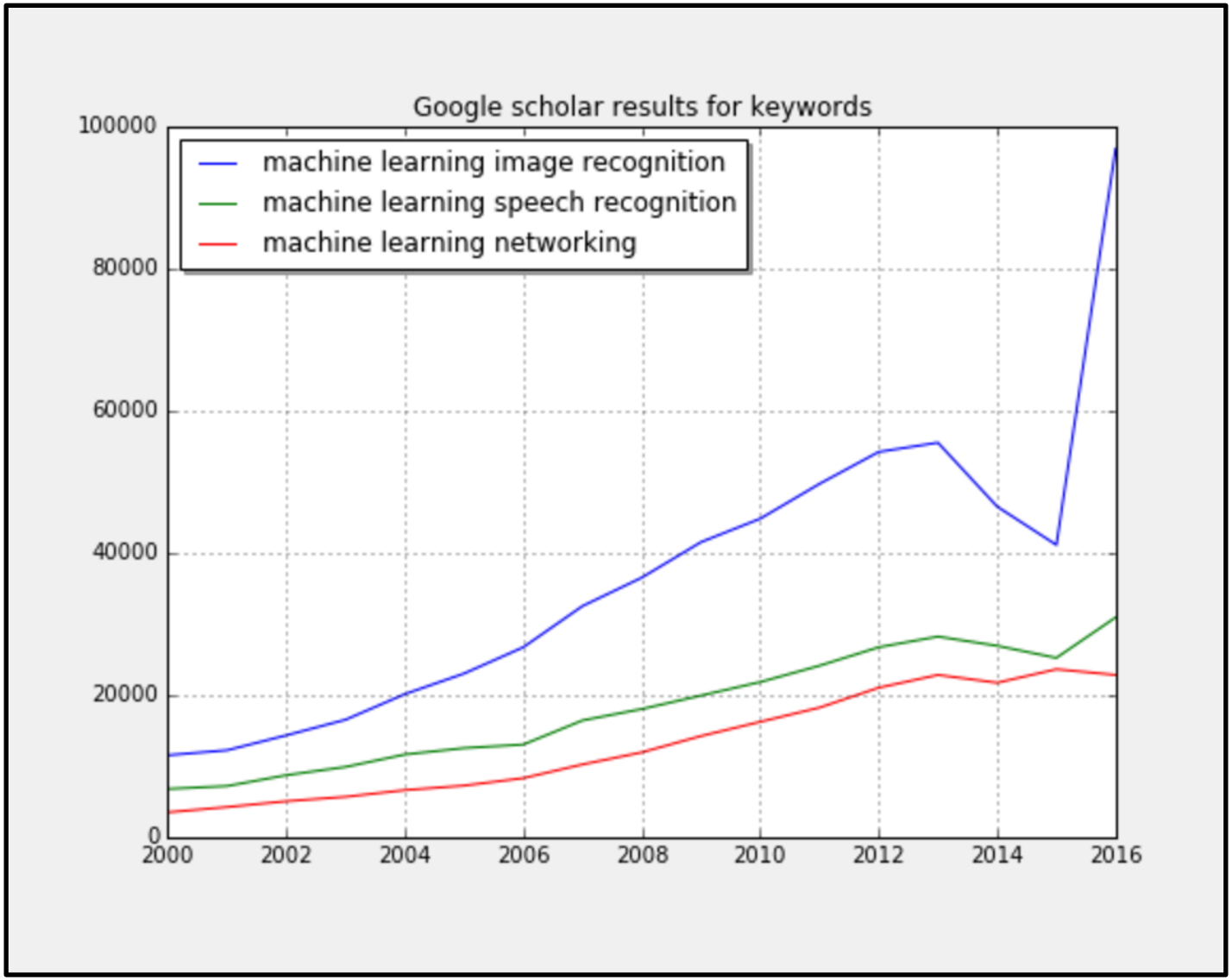
DANIEL FAGGELLA × FEBRUARY 28, 2016

Machine learning is revolutionizing the world as we know it, both in and out of the digital realms, and is predicted to expand to a $2 trillion market by 2025.

5G Networks

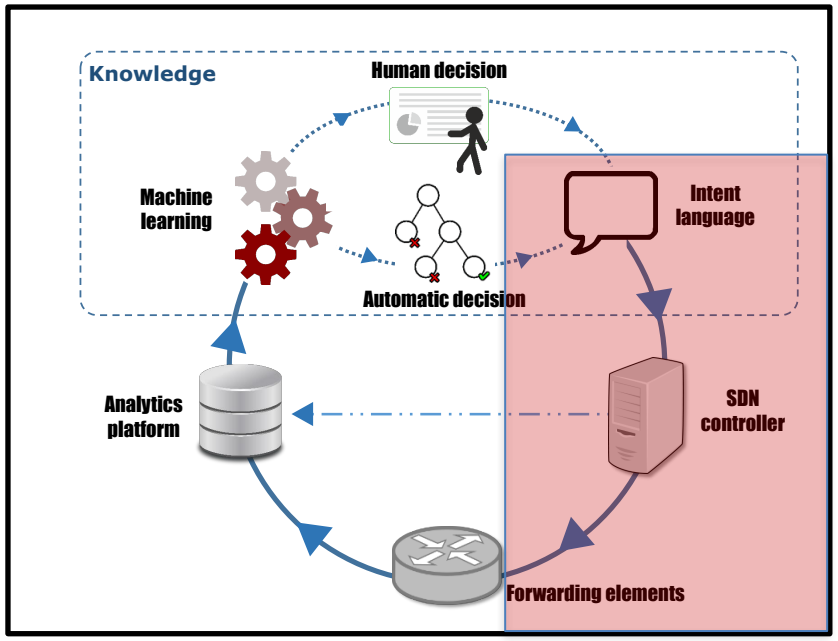

CogNet
Home
Project
CogNet in 5GPPP
Dissemination
Events
News
Contact
Building an Intelligent System
of Insights and Action
for 5G Network Management
COGNET

However….

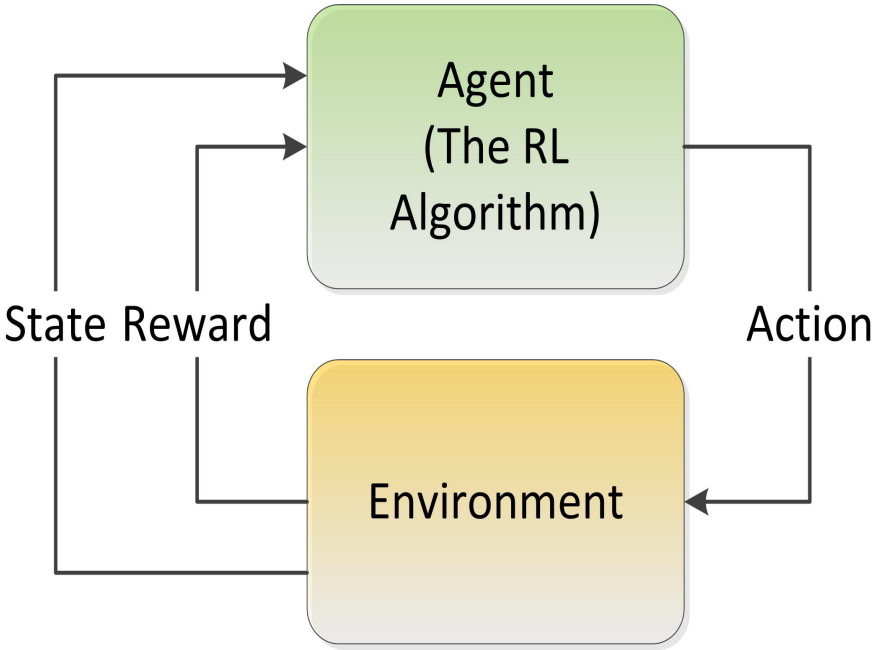

Google scholar results for keywords
machine learning image recognition
machine learning speech recognition
machine learning networking
100000
80000
60000
40000
20000
0
2000
2002
2004
2006
2008
2010
2012
2014
2016

We Want To Learn Control

Reinforcement Learning Meets Monte Carlo Tree Search and Deep Learning



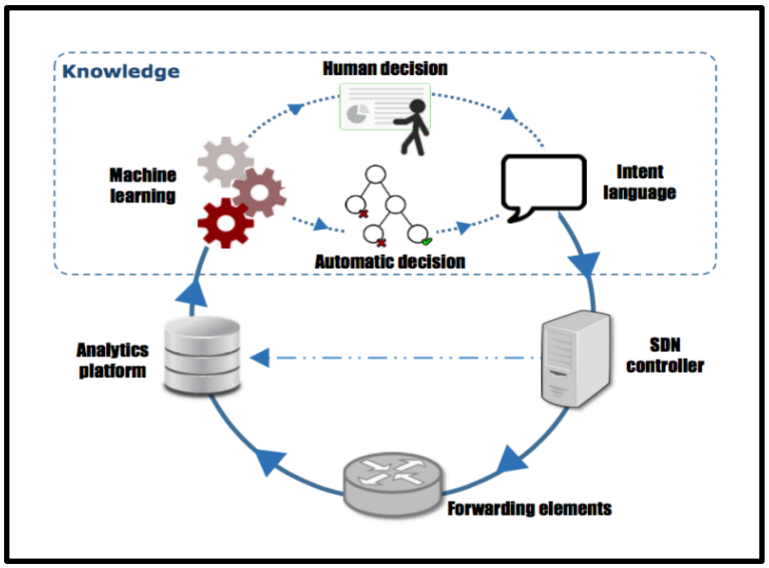
Hard coded control logic (not learned)



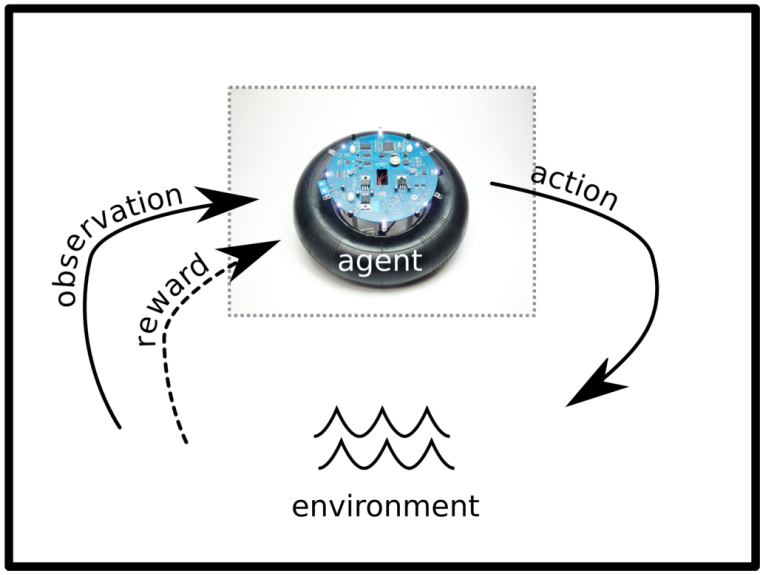
- Security: Agent can learn dynamic/evolving behavior of adversary
- DevOPs: Agent can learn workflow automation (e.g. Openstack Mistral)
- Orchestration: Agent can learn dynamic behavior of VNFs/system
- Deep policy net, $\pi(s) = a, s \in S, a \in A(s)$, can capture human intuition

http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

Static vs. Reinforcement Learning Architectures



- Today's Static ML Architectures
  - ML doesn't have "agency"
  - Hard-coded or open loop control
- Doesn't learn control
- Assumes stationary DGD
- Largely off-line



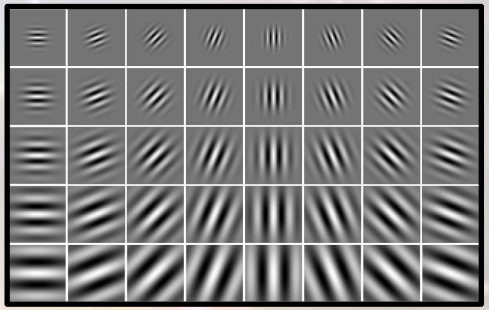
- Reinforcement Learning Architecture
  - Agent architecture
  - Agent learns control/action selection
- Adapts to evolving environment
- Non-stationary distributions
- *Network gamification*

Summary

- Lots of Network Data
  - Standardized and labeled datasets still scarce
  - However, most network data sources (e.g., netflow) not designed for ML
- Lots of open source ML frameworks, Cloud APIs, Examples, …
  - Tensorflow (tensorflow.org)
  - Torch (torch.ch)
  - Scikit-learn (scikit-learn.org)
  - Many others
- Skills gap persists and it still requires skill/experience to
  - Build DNN architectures/models
  - Find ”good” settings for hyper-parameters
  - Prevent overfitting
  - …
- All of this said, you *will* be seeing ML in all facets of networking
  - And everything else for that matter

Finally…Many Beautiful Mysteries Remain

- Back-propagation
  - Gradient-based optimization with gradients computed by back-prop
  - Why is such a simple algorithm so powerful?
  - Optimization an active area of research
  - Many new techniques , e.g., Layer Normalization
  - https://arxiv.org/pdf/1607.06450v1.pdf

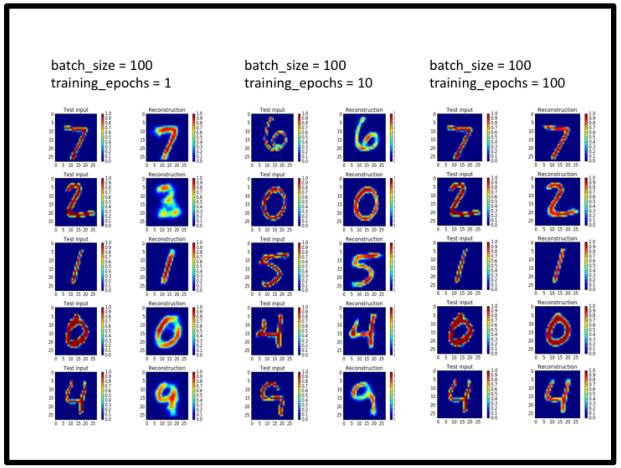


- Neural Nets
  - What are the units (artificial neurons) actually doing?
  - https://arxiv.org/pdf/1509.06321.pdf
- Adversarial Images/Generative Adversarial Nets (GANs)
  - https://arxiv.org/abs/1412.6572
- Why does deep and cheap learning work so well?
  - Physics perspective
  - http://arxiv.org/pdf/1608.08225v1.pdf
- And many more

- So how many 28x28 grayscale images are there?
- Say there are 8 bits of grayscale → 256^{784} possible images
- Our autoencoder had 156800 parameters and $156800/256^{784} \approx 0$
- So how can a simple autoencoder find the MNIST subspace?

Agenda

- What Is Machine Learning?
- What Can We Do Today, And What Are The Challenges?
- What's On the Horizon?
- Technical explanations/code
  - https://github.com/davidmeyer/ml
  - http://www.1-4-5.net/~dmm/ml



Q&A

Thank you

(have more questions/comments? dmm@1-4-5.net)