IDC Corp.

Hands-on Cybersecurity Artificial Intelligence

Gordon W. Romney, Ph.D., CEH

University of San Diego
– Shiley-Marcos School of Engineering

James Guymon Director of Data Science NA

Edge by Ascential
– An industry leader in eCommerce analytics

An Algorithm for Magic Tricks

I. **The Pledge**

Where the magician sets expectations

II. **The Turn**

The twist in the plot that brings drama and fascination

III. **The Prestige**

The unexpected and illuminating fulfillment of the Pledge

People Can Apply Models Too

I. **The Pledge**

Where we promise to entice you to explore data science

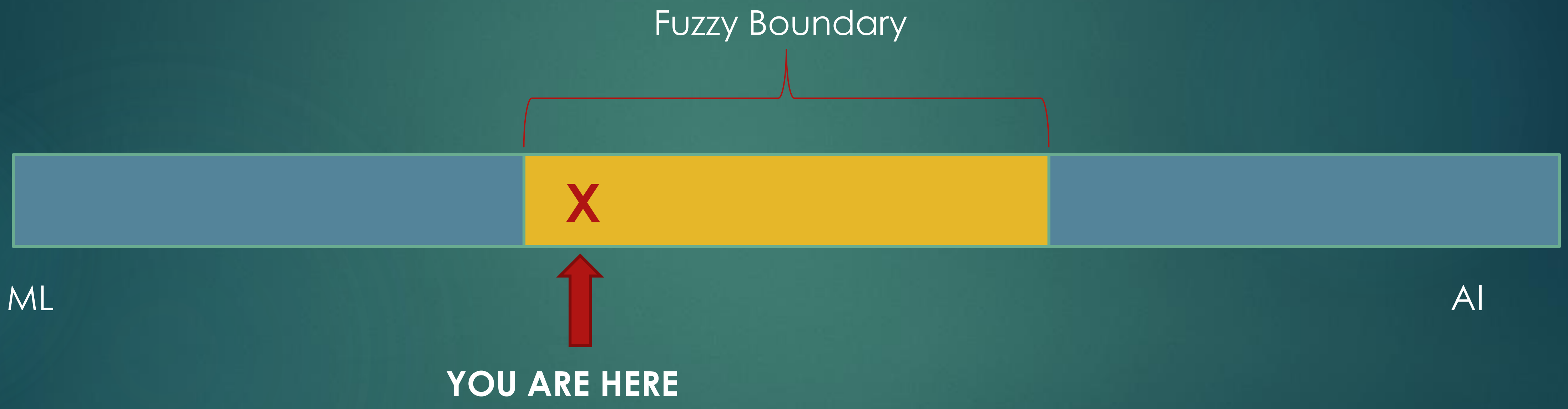
II. **The Turn**

Where we temporarily scare you away from data science

III. **The Prestige**

Where we provide solutions and welcome you to the data science family

Secondary Objective: Define AI vs ML



Agenda

- ✓ Ice Breaker: Magic
- ❑ Enticement to Data Science
- ❑ Healthy Terror
- ❑ The Trellis
- ❑ Appendix

Enticement

“Interest and awareness of AI is at a fever pitch”

[IDC 2019 May]

The AI Opportunity and Need

- From predictions, recommendations, and advice to automated customer service agents and intelligent process automation, AI is changing the face of how we interact with computer systems.“

- International Data Corporation (IDC) Worldwide Semiannual Cognitive Artificial Intelligence Systems Spending Guide forecasts cognitive and AI spending will grow to $52.2 billion in 2021 and achieve a compound annual growth rate (CAGR) of 46.2% over the 2016-2021 forecast period.
- The strongest spending growth over the five-year forecast will be in Japan (73.5% CAGR) and Asia/Pacific (excluding Japan and China) (72.9% CAGR). China will also experience strong spending growth throughout the forecast (68.2% CAGR).

- Other words used in marketing for AI: cognitive, omni-present, smart, intelligent, predictive, deep learning, artificial neural networks (ANNs)

NVIDIA

"Powering Change with AI and Deep Learning.

AI doesn't stand still. It's a living changing entity that powers change throughout every industry across the globe. As it evolves, so do we all. From the visionaries, healers, and navigators to the creators, protectors and teachers. It's what drives us today. And what comes next."

[Nvidia 2019]

Industry Adopters of AI

- 2018 Retail AI spending $3.8B in automated customer service agents and expert shopping and product recommendations
- 2018 Banking AI spending $3.3B in automated threat intelligence and prevention systems, fraud analysis and investigation
- 2018 Discrete manufacturing $2.0B in preventative maintenance & QA
- 2018 Healthcare spending $1.7B in diagnosis and treatment systems

Forecasts are understated as they include very little AI in Cybersecurity

Banking, Connected Car and Healthcare Telemedicine initiatives are the exception.

Cisco, Splunk, NVIDIA, IBM, Intel, Google, AWS, Elephant Scale and DataRobot are early adopters.

AIOps is a potential, future critical area for Cybersecurity education.

Artificial Intelligence for IT Operations (AIOps) is a Gartner-defined platform that combines big data and artificial intelligence (AI) functionality to replace a broad range of IT Operations processes and tasks including availability and performance monitoring, event correlation and analysis, IT service management, and automation. [Splunk 2019]

Terror

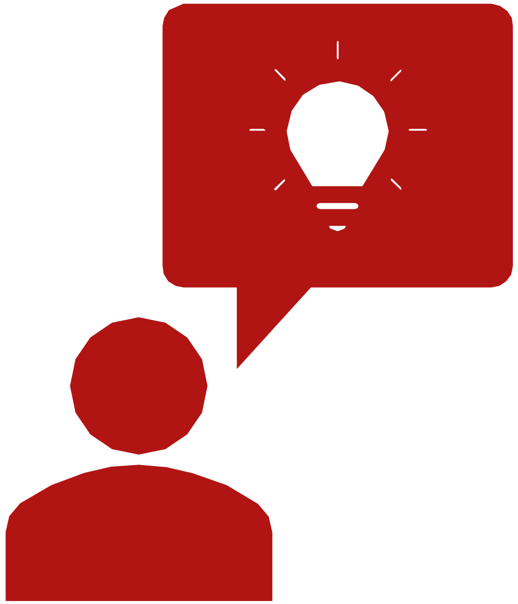
Math

There is no escaping the need for a strong mathematical foundation

How that manifests itself daily may surprise you, however.

Probability and Statistics occupy the role most people assume math will.

Many people who assume they could not be data science professionals can.



Couture Math Curriculum?

Cyber security curricula often does not include mathematics as required at the data science level

Simply porting courses from math departments may not be best

TWIST! The Terror? It's Ethics

ML and AI often determine:

- which job applicants get seen by a human
- which people or businesses get loans & at what rates
- sentencing in court proceedings
- wealth at retirement through portfolio management
- who is authenticated as you, granting access to your identity
- operation of planes & cars
- who justifies surveillance from the government

And can result in institutional discrimination, loss of wealth, liberty, and life

Will YOUR model be one that denies people their civil liberties due to non-malicious error? Or wipes away their retirement?

DON'T PANIC!

We come bearing gifts.

Download the Trellis code and grow yourself around it until you mature a bit:
https://github.com/SonicAlch3mist/CISSE_AI

The code, tools, specific algorithms are not the point- it's about the thought process, guiding principles, and order of tasks.

Every decision you make will have data to justify it, and your prediction accuracy will also be accompanied by a robust measurement of uncertainty.

Start With Sense of Inadequacy

… then realize that this is not about you.

Our confidence must anchor to ***the process, not ourselves***.

Let proven methodology bear the burden. Then hold to it religiously.

Pay more attention to measuring uncertainty than accuracy.

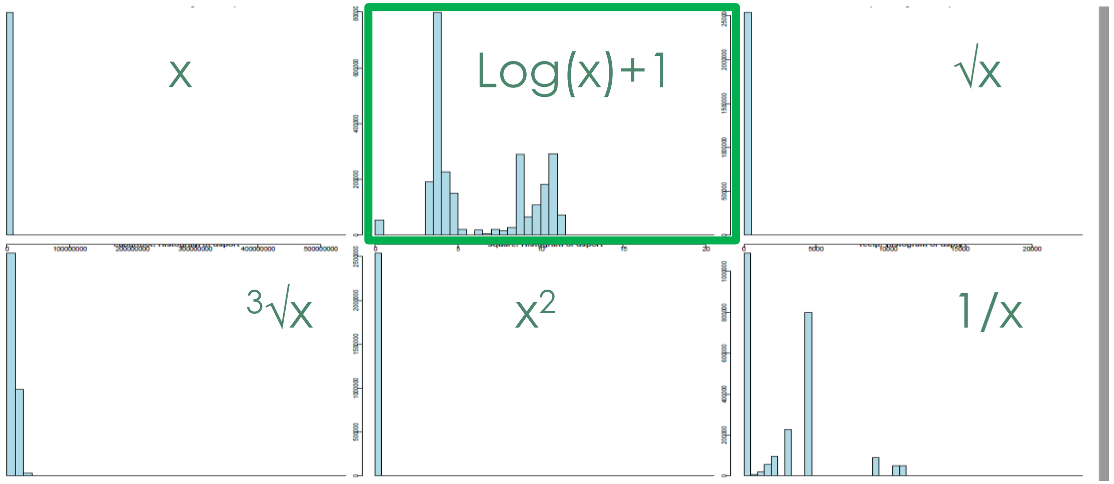
Attack Category vs. Features

Category	Feature Numbers
Normal	11,34,19,20,21,37,6,10,11,36,47
DoS	6,11,15 16,36,37,39,40,42,44,45
Fuzzers	6,11,14,15,16,36,37,39,40,41,42
Backdoors	6,10,11,14,15,16,37,41,42,44,45
Exploits	10,41,42,6,37,46,11,19,36,5,45
Analysis	6,10,11,12,13,14,15,16,34,35,37
Generic	6,9,10,11,12,13,15,16,17,18,20
Reconnaissance	10,14,37,41,42,43,44,9,16,17,28
Shellcode	6,9,10,12,13,14,15,16,17,18,23
Worms	41,37,9,11,10,46,23,17,14,5,13

Features (Attributes) of Data Set (6-18)

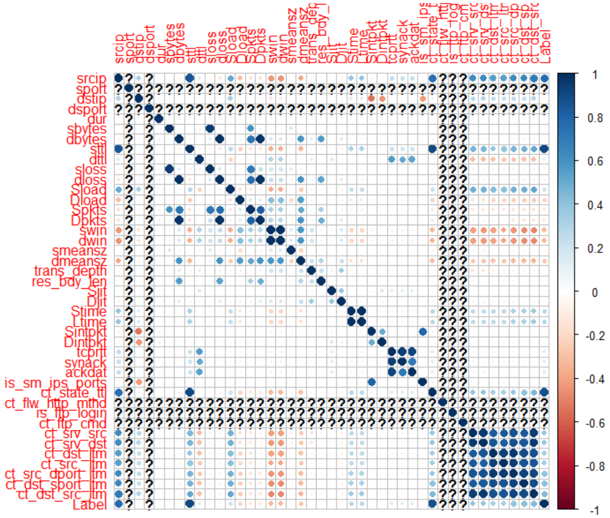
2. Basic Features		
6	state	The states and its dependent protocol e.g., CON.
7	dur	Row total duration.
8	sbytes	Source to destination bytes.
9	dbytes	Destination to source bytes.
10	sttl	Source to destination time to live.
11	dttl	Destination to source time to live.
12	sloss	Source packets retransmitted or dropped.
13	dloss	Destination packets retransmitted or dropped.
14	service	Such as http, ftp, smtp, ssh, dns and ftp-data.
15	sload	Source bits per second.
16	dload	Destination bits per second.
17	spkts	Source to destination packet count.
18	dpkts	Destination to source packet count.

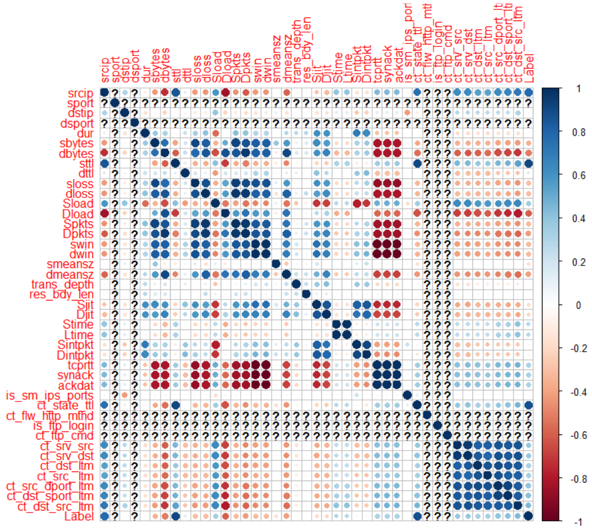
Visualizations lead to ideas



That are tested

Correlation Matrix: Before/After Transformations

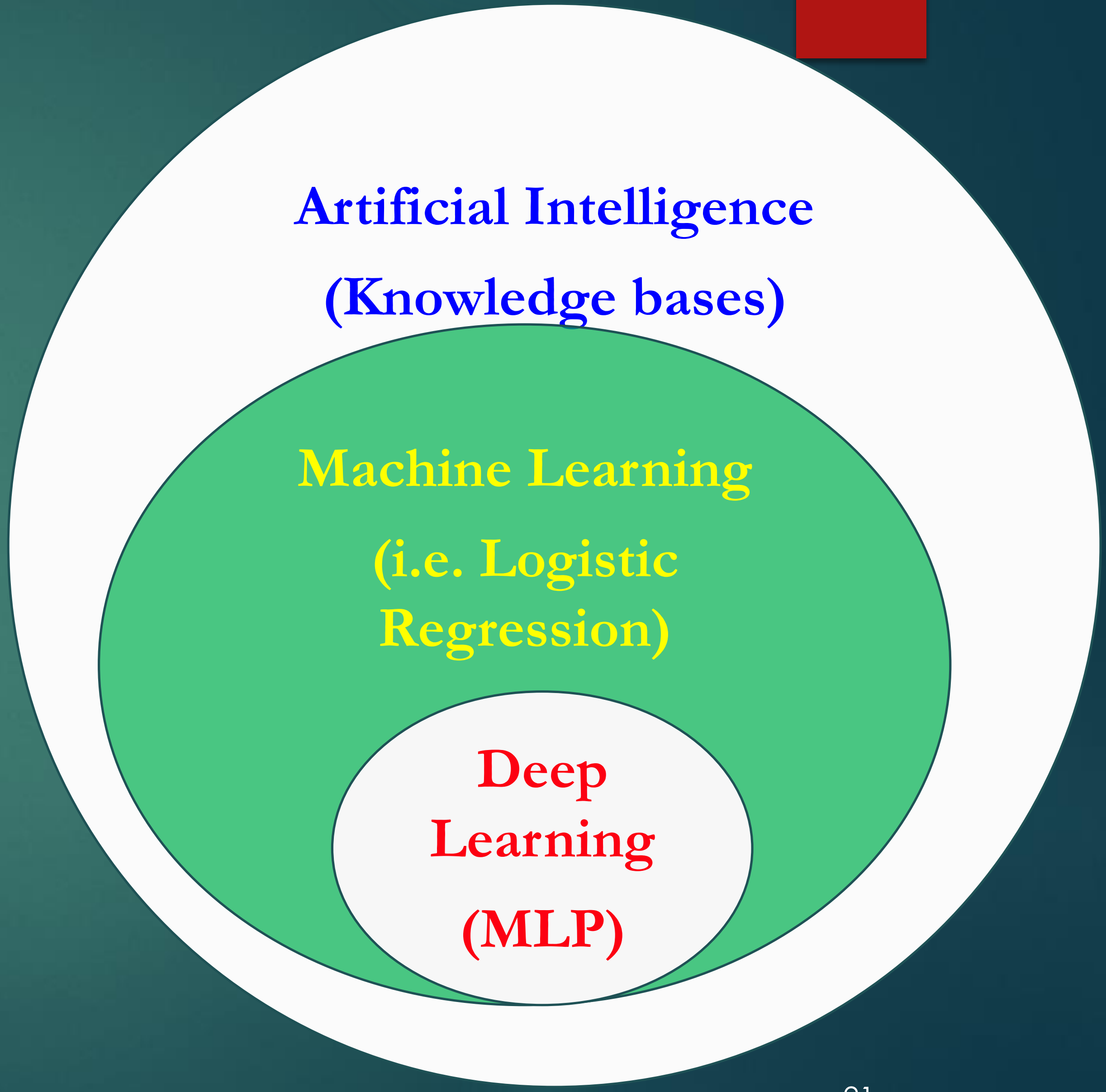




Data Drives Decisions

Back To Definitions …

- **Artificial Intelligence (AI)**
  - Combined learning technologies
- **Machine Learning**
  - Math and stats
- **Deep Learning**
  - Neural networks
  - Representation learning



Neural Networks

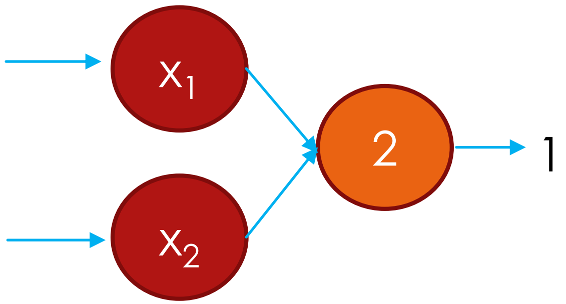
- Modeled after human brain
- Recognized patterns
  - Numerical
  - Contained in vectors
  - Translated from real-world data
    Images, Sound, Text, Time series
- Invented in the 1960's
- "Re-invented" in 2012

Image by Elephant Scale

Neural Network Basics: And, Or, Xor

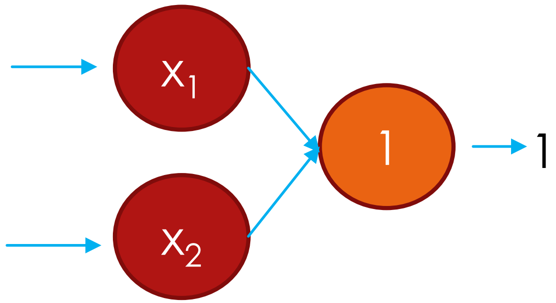
∅: FORCES THE OUTPUT TO 0

AND



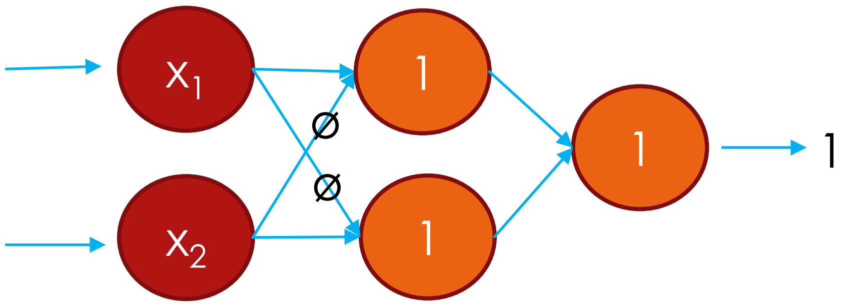
x_1	Truth Value	x_2
1	1	1
1	0	0
0	0	1
0	0	0

OR



x_1	Truth Value	x_2
1	1	1
1	1	0
0	1	1
0	0	0

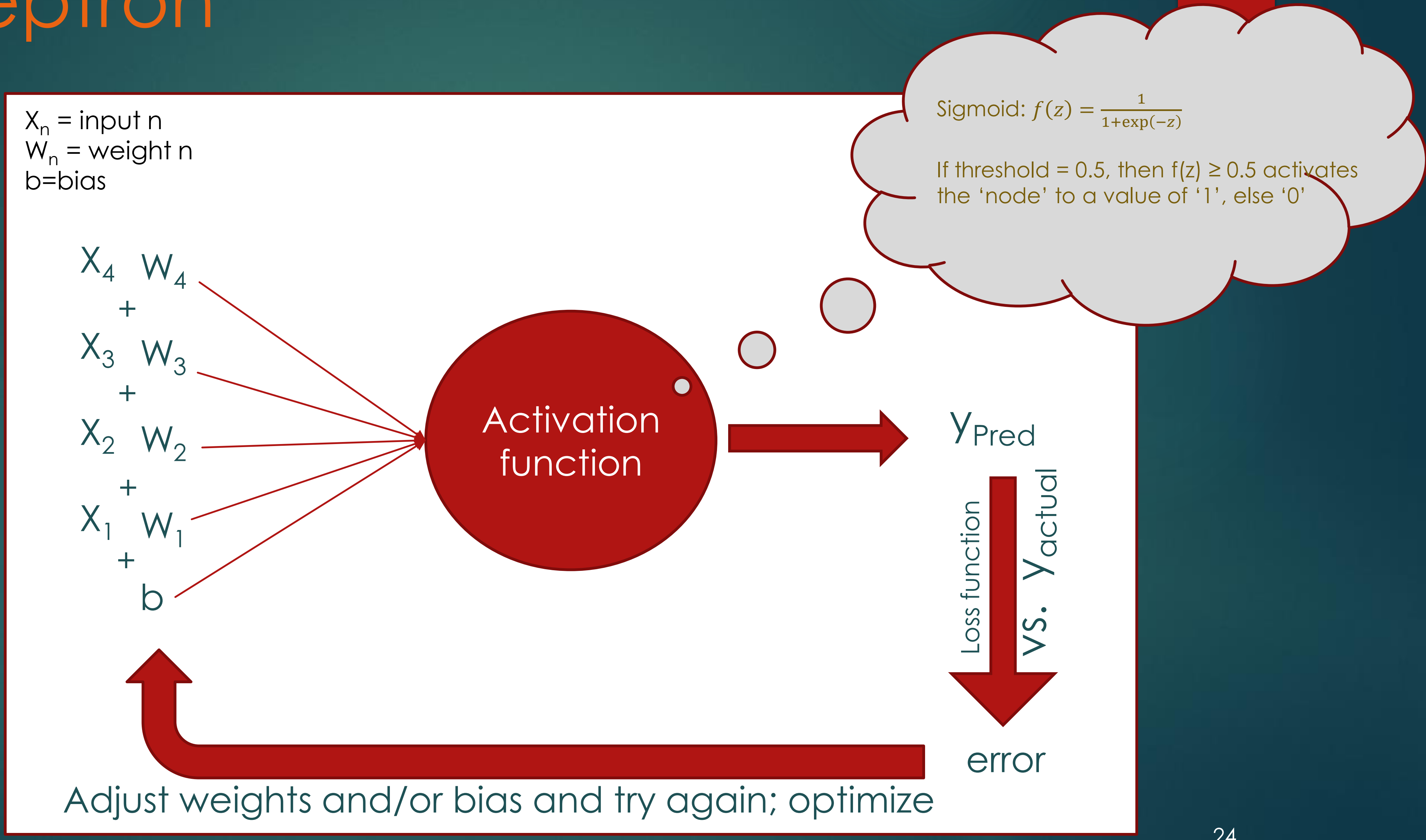
XOR



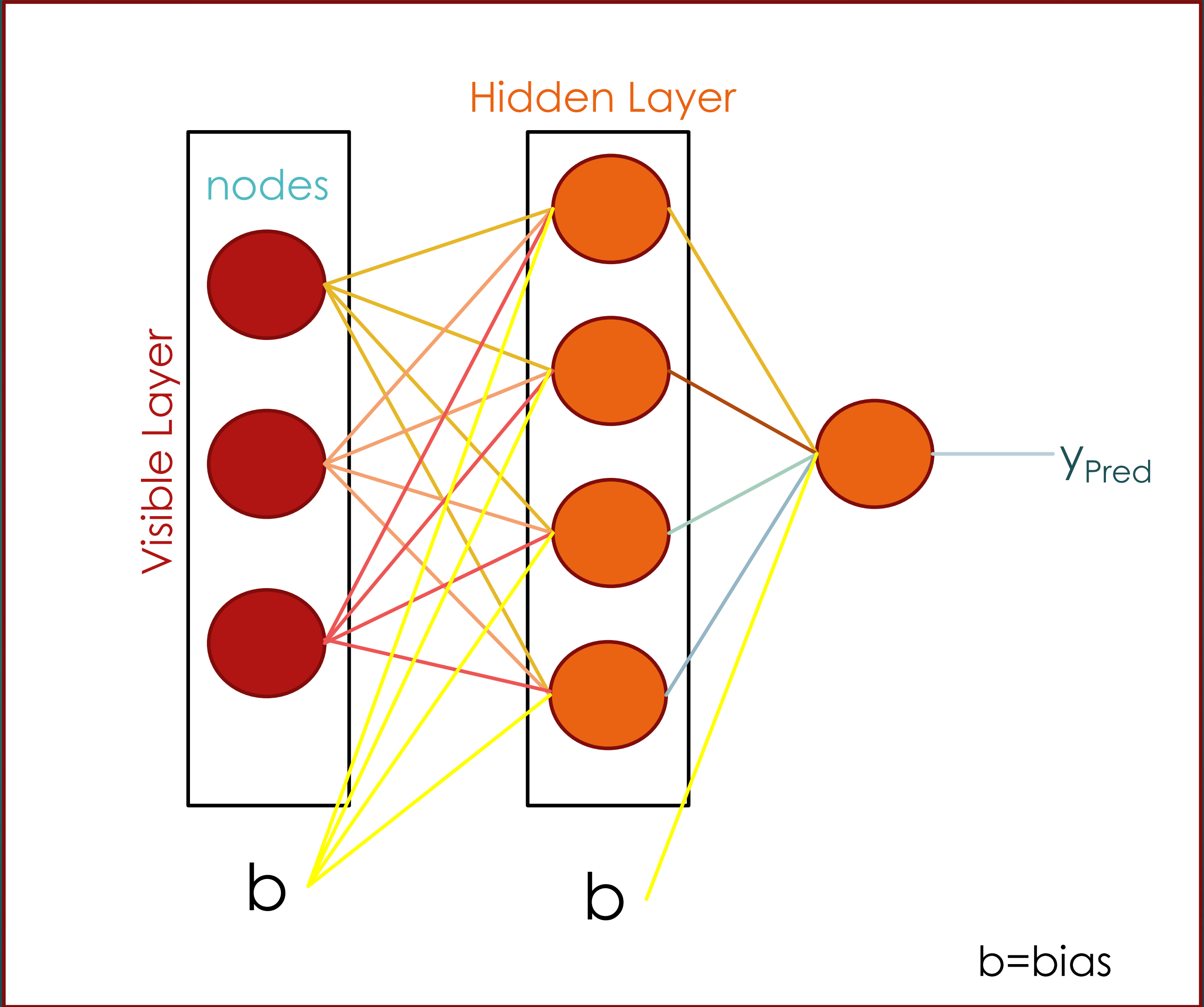
x_1	Truth Value	x_2
1	0	1
1	1	0
0	1	1
0	0	0

ML AI

Perceptron

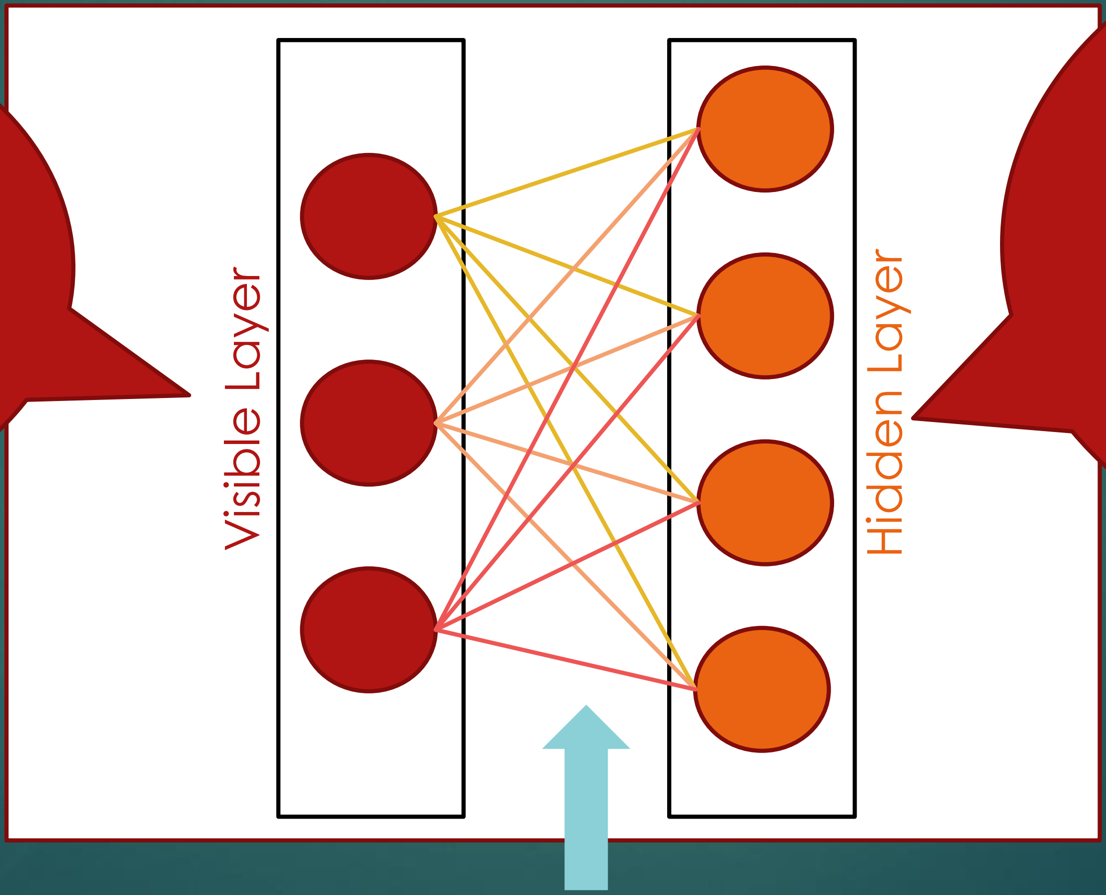


Neural Network Basics: Layers



Restricted Boltzmann Machine

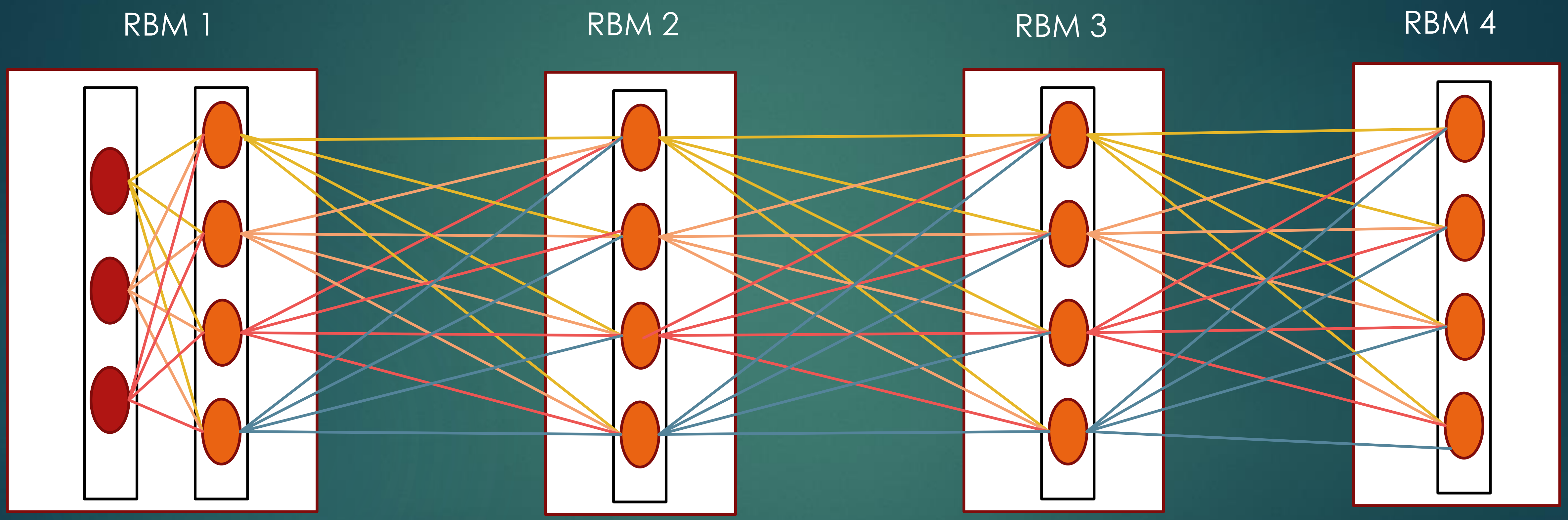
Each node in the visible layer is ‘restricted’ from communicating with the others. This makes them *independent*



Learns the probability distributions of what it is fed, ‘thinking’ in terms of *independent* binary events [with its probability accumulation termed ‘energy’]

Latent relationship hunting

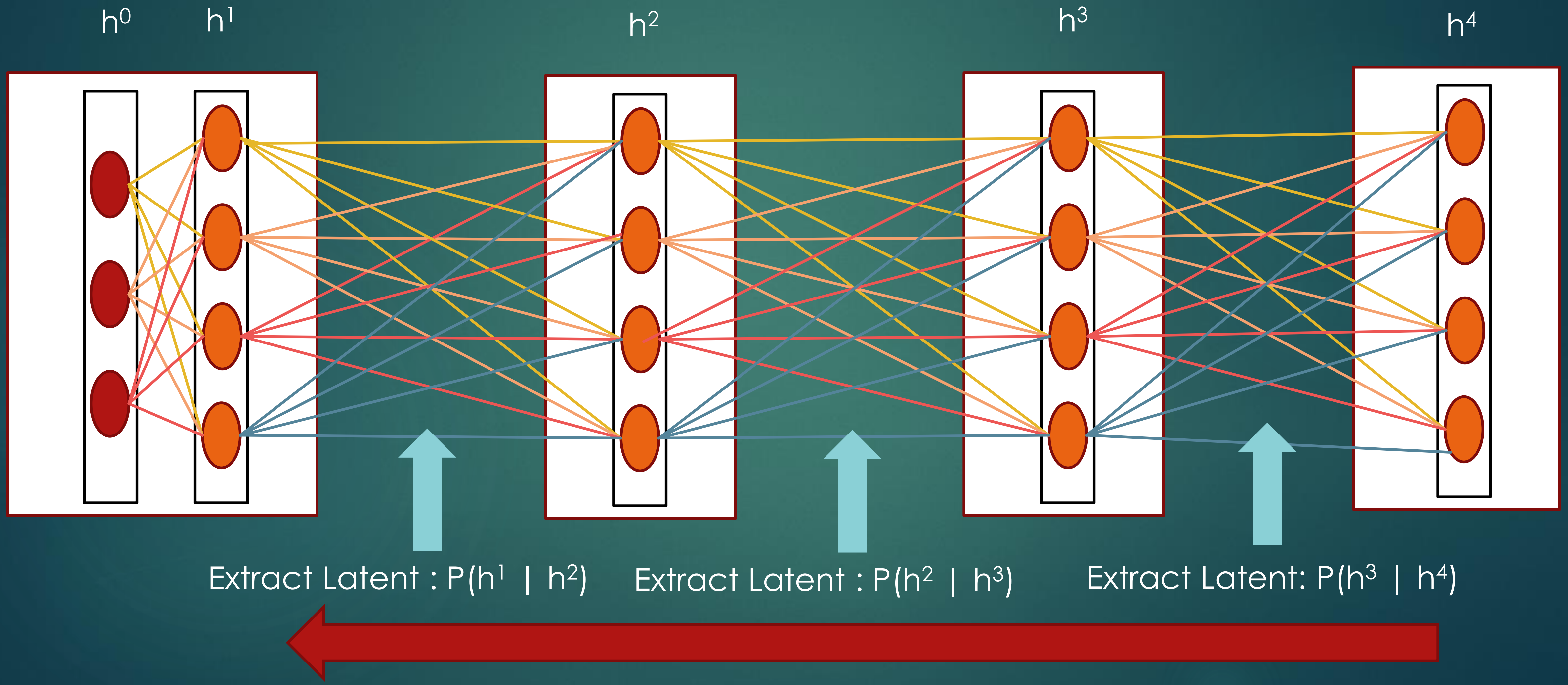
Deep Belief Network



Each RBM #2:4 is the visible layer to the RBM after it, and the hidden layer to the RBM before it

Deep Belief Network: Step 1

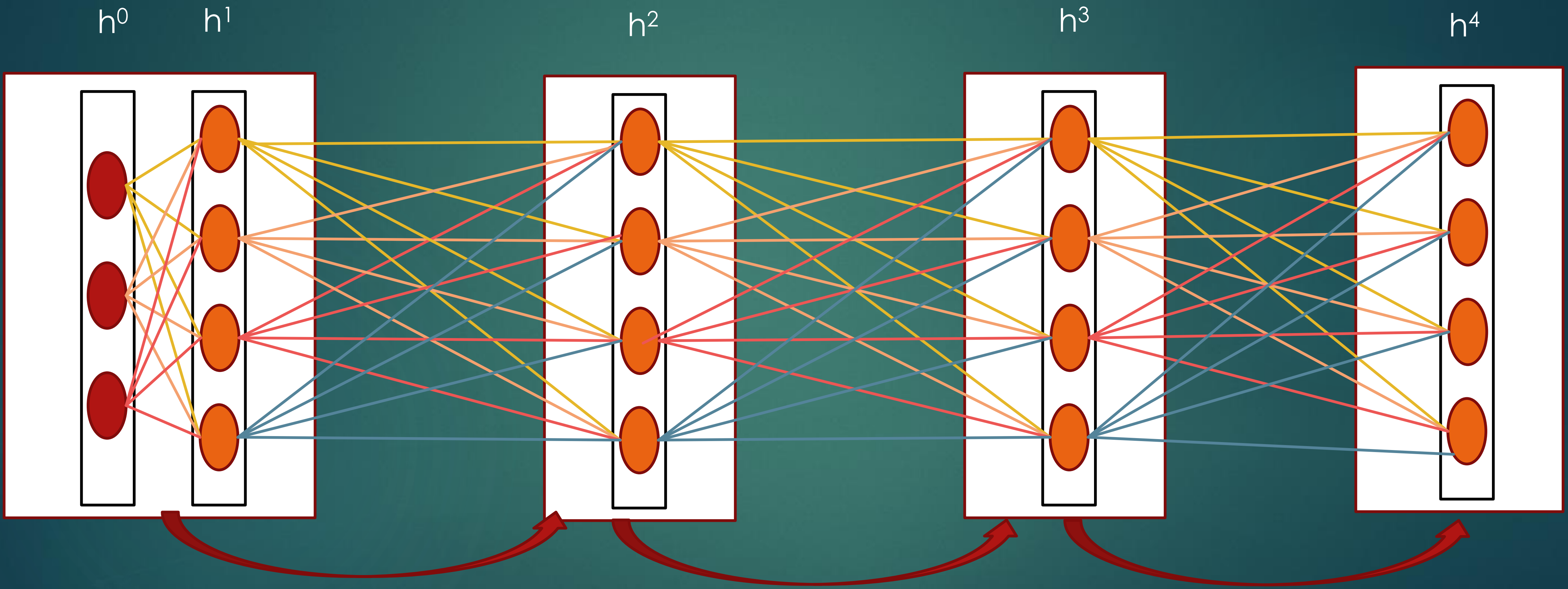
Construct Probability Network with 3 features and 4 hidden layers



ML AI

Deep Belief Network: Step 2

Classify using probabilities gleaned from Step 1

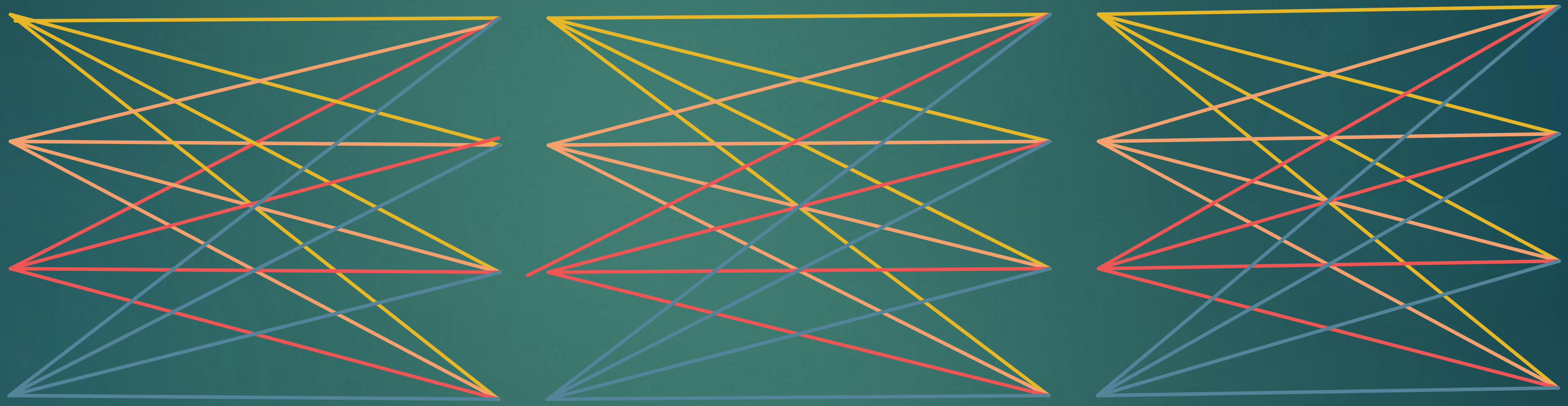


Given the visible-hidden joint distribution: $P(h^1 \mid h^2)$ & $P(h^2 \mid h^3)$ & $P(h^3 \mid h^4)$ when $h^0 = x^0$

Solve for x^{input}

‘AI’ Emerges In The Networks

The math is the same as with ML



The Trellis

The Trellis https://github.com/SonicAlch3mist/CISSE_AI

- Philosophical Grounding in Scientific Method
  - Constant experimentation to validate decisions
- Data Load
- Pre-Visualizations
  - Missingness, Network Graphs, Histograms, Correlation Matrices
- Data Cleaning
  - Outliers, Nulls, Large-Level Factors
- Feature Engineering
  - Transformation, Balancing, One-Hot Encoding, Feature Selection
- Model Building
  - Train, Test, Validation Sets
  - Bootstrapping, Cross-fold Validation
  - XGBoost, DBN, Ensembles
    - Parameter Tuning
  - Confusion Matrices
  - Business Outcome Optimization

Research Suggestions, Hints, & Side Quests Sprinkled Throughout!

Field Trip To Trellis

Q & A

Contributor Biographies

Gordon W. Romney, Ph.D., CEH, is Professor of Computer Science and Cybersecurity in the Shiley-Marcos School of Engineering of the University of San Diego (USD). He is the Director of the Center for Cyber Security Engineering and Technology, and oversees the MS in Cyber Security Engineering program at USD. Current research includes developing an Artificial Neural Network for HIPAA-compliant eVisit telemedicine medical diagnosis and hardening of IoT Electronic Control Units in Cisco's Connected Vehicle initiative.

University of San Diego
SHILEY-MARCOS
SCHOOL OF ENGINEERING

CENTER FOR
CYBER SECURITY
ENGINEERING AND
TECHNOLOGY

Contributor Biographies

James Guymon is the Director of Data Science, North America for Edge by Ascential – an industry leader in eCommerce analytics. Prior to Edge by Ascential, James worked as a data scientist at Progrexion Marketing, where he helped build the industry's first passive authentication system for account opening in cooperation with Experian.

James.Guymon@ascentialedge.com

Contributor Biographies

Mark Kerzner, President
Elephant Scale
https://elephantscale.com/
Its mission is to offer high quality services and training in Big Data eco systems.

Mark contributed a number of slides attributed to Elephant Scale as well as the Python notebook of a Domain Generation Algorithm attack.

mark@elephantscale.com

References

ADFA AND UNSW DATA SETS (2019, MAY). INTRUSION DETECTION DATA SET OF TWO MILLION AND 540,044 RECORDS FOR NINE TYPES OF ATTACKS. RETRIEVED MAY 2019 FROM HTTPS://WWW.UNSW.ADFA.EDU.AU/UNSW-CANBERRA-CYBER/CYBERSECURITY/ADFA-NB15-DATASETS/

CISCO (2011). CISCO GLOBAL CLOUD INDEX: FORECAST AND METHODOLOGY, 2011–2016 [WHITE PAPER]. RETRIEVED OCTOBER 2016 FROM HTTP://WWW.CISCO.COM/EN/US/SOLUTIONS/COLLATERAL/NS341/NS525/NS537/NS705/NS1175/CLOUD_INDEX_WHITE_PAPER.HTML

CISCO (2013). DAWN OF THE ZETTABYTE ERA [WEB LOG POST]. RETRIEVED OCTOBER 2016 FROM HTTP://BLOGS.CISCO.COM/NEWS/THE-DAWN-OFTHE-ZETTABYTE-ERA-INFOGRAPHIC/

DGA. (2019). DOMAIN GENERATION ALGORITHM. RETRIEVED MAY 2019 FROM HTTPS://EN.WIKIPEDIA.ORG/WIKI/DOMAIN_GENERATION_ALGORITHM

IDC (2011, DECEMBER). IDC PREDICTS 2012 WILL BE THE YEAR OF MOBILE AND CLOUD PLATFORM WARS AS IT VENDORS VIE FOR LEADERSHIP WHILE THE INDUSTRY REDEFINES ITSELF [PRESS RELEASE]. RETRIEVED FROM HTTP://WWW.BUSINESSWIRE .COM/NEWS/HOME/20111201005201/EN/IDC-PREDICTS-2012-YEAR-MOBILE-CLOUD-PLATFORM#.UUMNFSRTMJA

IDC (2019 MARY). RETRIEVED MAY 2019 FROM HTTPS://WWW.BUSINESSWIRE.COM/NEWS/HOME/20180322005847/EN/WORLDWIDE-SPENDING-COGNITIVE-ARTIFICIAL-INTELLIGENCE-SYSTEMS-GROW

KATZ, R. N. (ED.). (2011). THE TOWER AND THE CLOUD: HIGHER EDUCATION IN THE AGE OF CLOUD COMPUTING. WASHINGTON, DC: EDUCAUSE. RETRIEVED FROM HTTP://NET.EDUCAUSE.EDU/IR/LIBRARY/PDF/PUB7202.PDFF

NOUR M., AND SLAY J. (2015). UNSW-NB15: A COMPREHENSIVE DATA SET FOR NETWORK INTRUSION DETECTION SYSTEMS (UNSW-NB15 NETWORK DATA SET), MILITARY COMMUNICATIONS AND INFORMATION SYSTEMS CONFERENCE (MILCIS), IEEE 2015.

NOUR M., AND SLAY J. (2016). THE EVALUATION OF NETWORK ANOMALY DETECTION SYSTEMS: STATISTICAL ANALYSIS OF THE UNSW-NB15 DATASET AND THE COMPARISON WITH THE KDD99 DATASET, INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE 2016: 1-14.

NOUR M., AND ET. AL. (2017). NOVEL GEOMETRIC AREA ANALYSIS TECHNIQUE FOR ANOMALY DETECTION USING TRAPEZOIDAL AREA ESTIMATION ON LARGE-SCALE NETWORKS." IEEE TRANSACTIONS ON BIG DATA *2017*.

NVIDIA (2019). DEEP LEARNING AI. RETRIEVED JUNE, 2019, FROM HTTPS://WWW.NVIDIA.COM/EN-US/DEEP-LEARNING-AI/

ROMNEY, G.W. & BRUESEKE, B.W. (2014, MARCH). MERGING THE TOWER AND THE CLOUD THROUGH VIRTUAL INSTRUCTION: THE NEW ACADEMY OF DISTANCE EDUCATION, NATIONAL UNIVERSITY JOURNAL OF RESEARCH IN INNOVATIVE TEACHING, LA JOLLA, CA VOLUME 7, ISSUE 1, MARCH 2014, P. 93- 118, HTTP://WWW.JRIT-NU.ORG/

ROMNEY, G. W., & BRUESEKE, B. W. (2013). APPARATUS, SYSTEM AND METHOD FOR A VIRTUAL INSTRUCTION CLOUD, U.S. PATENT NO. 9,947,236. WASHINGTON, DC: U.S. PATENT AND TRADEMARK OFFICE.

SPLUNK. (2019). AI FOR IT: PREVENTING OUTAGES WITH PREDICTIVE ANALYTICS. RETRIEVED FROM HTTPS://WWW.SPLUNK.COM/EN_US/FORM/AI-FOR-IT-PREVENTING-OUTAGES-WITH-PREDICTIVE-ANALYTICS.HTML

UTERMOHLEN, K. (2018, APRIL). 15 ARTIFICIAL INTELLIGENCE (AI) STATS YOU NEED TO KNOW IN 2018. RETRIEVED FROM HTTPS://TOWARDSDATASCIENCE.COM/15-ARTIFICIAL-INTELLIGENCE-AI-STATS-YOU-NEED-TO-KNOW-IN-2018-B6C5EAC958E5

ZETTABYTES. (N.D.). IN WIKIPEDIA. RETRIEVED MAY 2019 FROM HTTP://EN.WIKIPEDIA.ORG/WIKI/ZETTABYTE

TERABYTE. ONE THOUSAND GIGABYTES, OR 10 TO THE 12TH POWER BYTES OF DATA.

PETABYTE. ONE THOUSAND TERABYTES, OR 10 TO THE 15TH POWER BYTES OF DATA.

ZETTABYTE. ONE MILLION PETABYTES, OR 10 TO THE 21ST POWER BYTES OF DATA.

Thank You!

Center for Cyber Security Engineering and Technology

JUNE 10, 2019

COLLOQUIUM FOR INFORMATION SYSTEMS SECURITY EDUCATION (CISSE)

UNIVERSITY OF SAN DIEGO SHILEY-MARCOS SCHOOL OF ENGINEERING

Appendix to *Hands-on AI in Cybersecurity*

- UNSW-NB15 data set Infrastructure, data set Generating Architecture and 49 Features
- Machine Language and AI Technologies
- DGA domain data set example using Python notebook
- Confusion Matrices

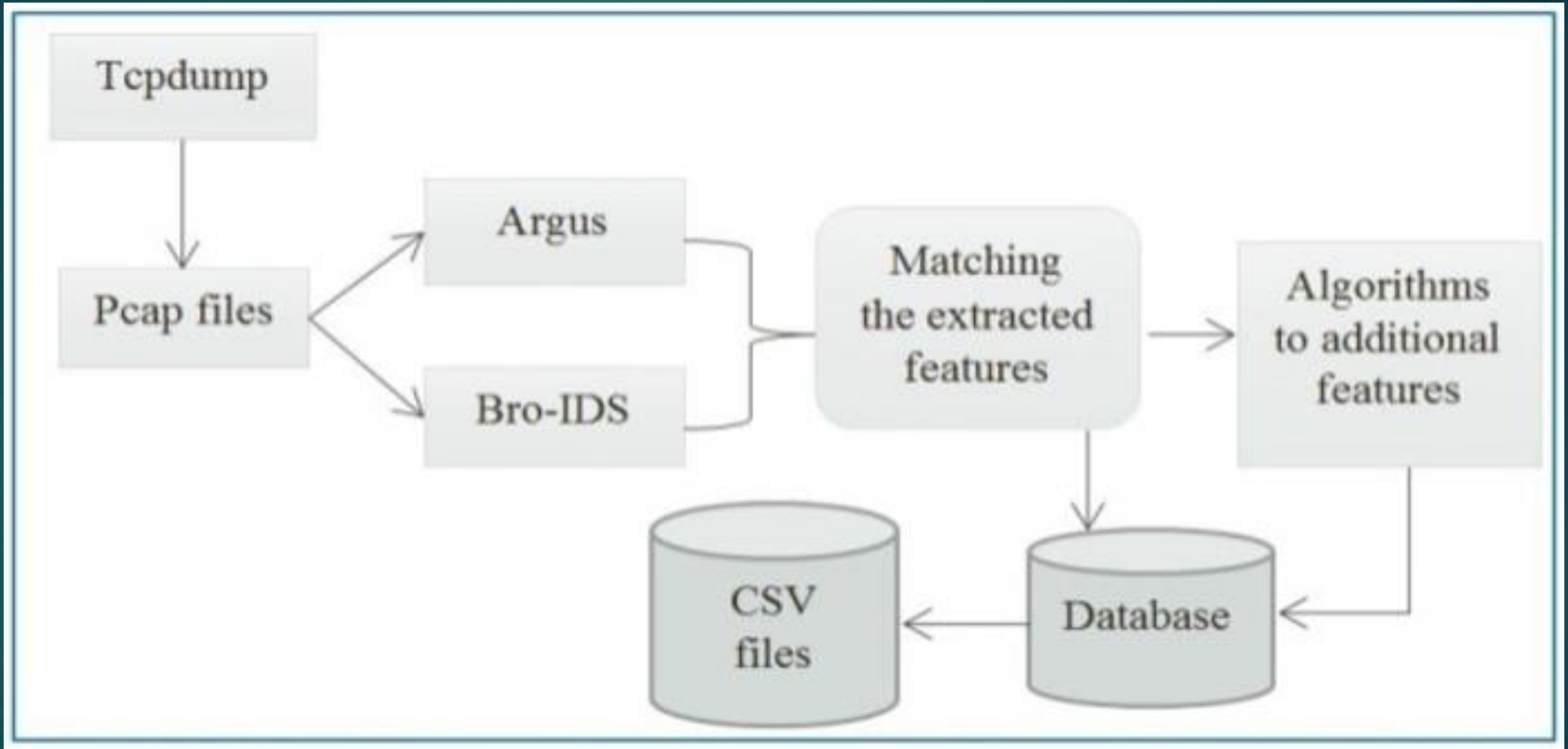
UNSW-NB15 data set Infrastructure, data set Generating Architecture and 49 Features

Cyber Traffic Specifics for UNSW-NB15 Data Set

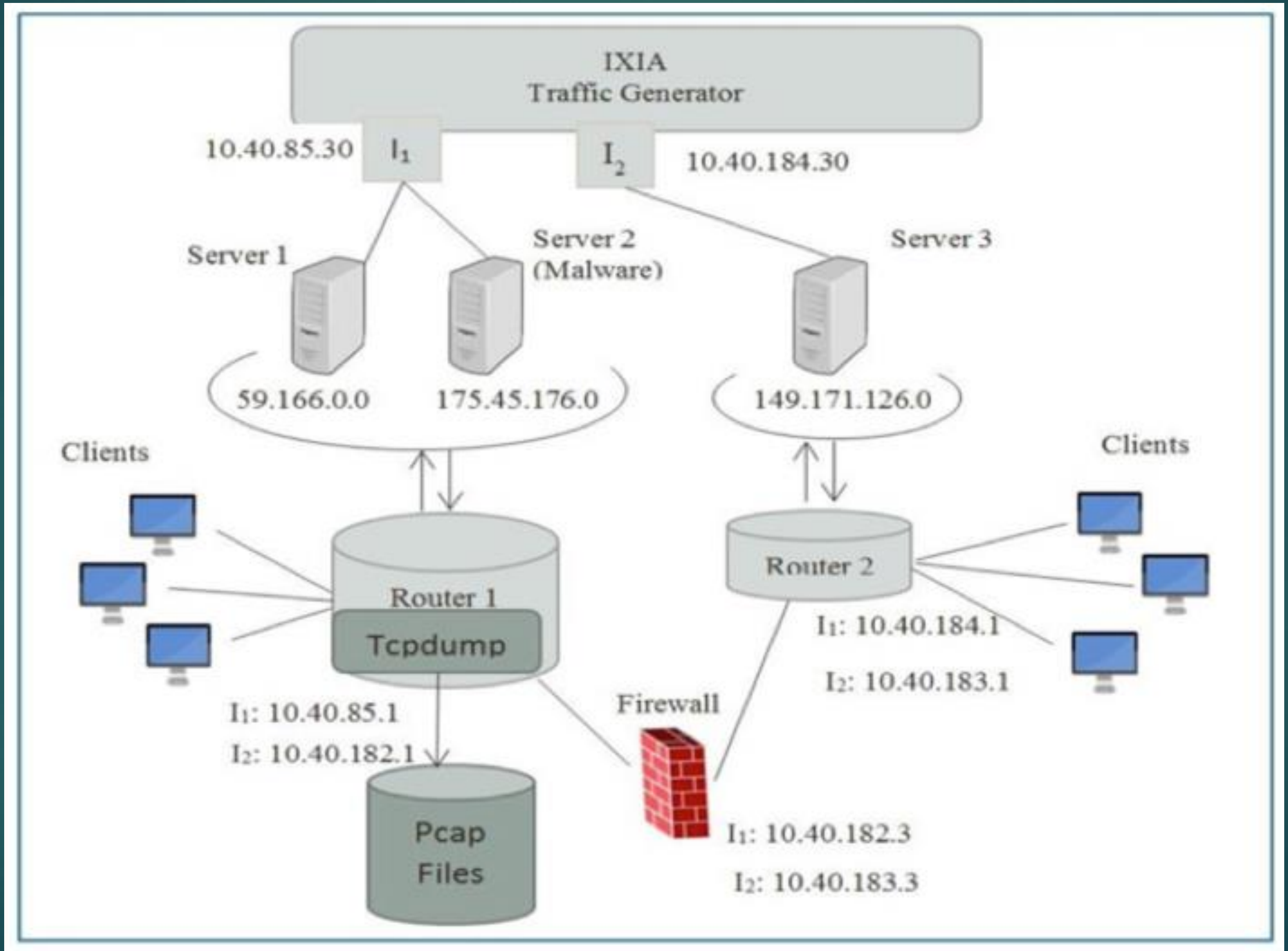
- Created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors.
- Tcpdump was utilized to capture 100 GB of the raw traffic (e.g., Pcap files).
- Nine types of attacks were captured: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.
- Twelve algorithms were developed to generate 49 features with the class label.
- The total number of records is 2,540,044 which are stored in four CSV files.

The details of the data set are summarized in the appendix to this slide deck and specified in [Nour 2015], [Nour 2016] and [Nour 2017].

Framework for Generating Data Set



Infrastructure for Intrusion Detection Data Set UNSW-NB15 by Dr. Nour Moustafa



Attack Category vs. Features

Category	Feature Numbers
Normal	11,34,19,20,21,37,6,10,11,36,47
DoS	6,11,15 16,36,37,39,40,42,44,45
Fuzzers	6,11,14,15,16,36,37,39,40,41,42
Backdoors	6,10,11,14,15,16,37,41,42,44,45
Exploits	10,41,42,6,37,46,11,19,36,5,45
Analysis	6,10,11,12,13,14,15,16,34,35,37
Generic	6,9,10,11,12,13,15,16,17,18,20
Reconnaissance	10,14,37,41,42,43,44,9,16,17,28
Shellcode	6,9,10,12,13,14,15,16,17,18,23
Worms	41,37,9,11,10,46,23,17,14,5,13

Features (Attributes) of Data Set (1-18)

#	Name	Description
1. Flow Features		
1	srcip	Source IP address.
2	sport	Source port number.
3	dstip	Destinations IP address.
4	dsport	Destination port number.
5	proto	Protocol type, such as TCP, UDP.

Features (Attributes) of Data Set (6-18)

2. Basic Features		
6	state	The states and its dependent protocol e.g., CON.
7	dur	Row total duration.
8	sbytes	Source to destination bytes.
9	dbytes	Destination to source bytes.
10	sttl	Source to destination time to live.
11	dttl	Destination to source time to live.
12	sloss	Source packets retransmitted or dropped.
13	dloss	Destination packets retransmitted or dropped.
14	service	Such as http, ftp, smtp, ssh, dns and ftp-data.
15	sload	Source bits per second.
16	dload	Destination bits per second.
17	spkts	Source to destination packet count.
18	dpkts	Destination to source packet count.

Features (Attributes) of Data Set (19-26)

3. Content Features		
19	swin	Source TCP window advertisement value.
20	dwin	Destination TCP window advertisement value.
21	Stcpb	Source TCP base sequence number.
22	dtcpb	Destination TCP base sequence number.
23	smeansz	Mean of the packet size transmitted by the srcip.
24	dmeansz	Mean of the packet size transmitted by the dstip.
25	trans_depth	The connection of http request/response transaction.
26	res_bdy_len	The content size of the data transferred from http.

Features (Attributes) of Data Set (27-36)

4. Time Features		
27	sjit	Source jitter.
28	djit	Destination jitter.
29	stime	Row start time.
30	ltime	Row last time.
31	sintpkt	Source inter-packet arrival time.
32	dintpkt	Destination inter-packet arrival time.
33	tcprtt	Setup round-trip time, the sum of 'synack' and 'ackdat'.
34	synack	The time between the SYN and the SYN_ACK packets.
35	ackdat	The time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	If srcip (1) = dstip (3) and sport (2) = dsport (4), assign 1 else 0.

Features (Attributes) of Data Set (37-49)

5. Additional Generated Features		
37	ct_state_ttl	No. of each state (6) according to values of sttl (10) and dttl (11).
38	ct_flw_http_mthd	No. of methods such as Get and Post in http service.
39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	No of flows that has a command in ftp session.
41	ct_srv_src	No. of rows of the same service (14) and srcip (1) in 100 rows.
42	ct_srv_dst	No. of rows of the same service (14) and dstip (3) in 100 rows.
43	ct_dst_ltm	No. of rows of the same dstip (3) in 100 rows.
44	ct_src_ ltm	No. of rows of the srcip (1) in 100 rows.
45	ct_src_dport_ltm	No of rows of the same srcip (1) and the dsport (4) in 100 rows.
46	ct_dst_sport_ltm	No of rows of the same dstip (3) and the sport (2) in 100 rows.
47	ct_dst_src_ltm	No of rows of the same srcip (1) and the dstip (3) in 100 records.
6. Labelled Features		
48	Attack_cat	The name of each attack category.
49	Label	0 for normal and 1 for attack records

Machine Language and AI Technologies

Technology Stack Comparison

Technology	Pros	Cons
R	- Rich environment - Thousands of libraries	- Rough on data cleanup - Not a general purpose language - **Data must fit on one machine**
Python	- General purpose programming language - Excellent libraries (Pandas / scikit-learn) - Gaining popularity in recent years	- **Data must fit on one machine**

AI Software Eco System

	Machine Learning	Deep Learning
Java	- Weka - Mahout	- DeepLearning4J
Python	- SciKit - (Numpy, Pandas)	- Tensorflow - Theano - Caffe
R	- Many libraries	- Deepnet - Darch
Distributed	- H20 - Spark	
Cloud	- Google: GCP - Microsoft: ML on Azure - Amazon: SageMaker	

Tools for Scalable Machine Learning

- **Spark ML**
  - Runs on top of popular Spark framework
  - Massively scalable
  - Can use memory (caching) effectively for iterative algorithms
  - Language support: Scala, Java, Python, R

- Amazon Machine Learning (SageMaker)
  - Ready to go algorithms
  - Wizards to guide
  - Scalable on Amazon Cloud
  - Integrated with AWS

Tools for Scalable Machine Learning

- [Azure ML Studio]
  - Built on Azure cloud (Microsoft)
  - Language support: Python, R
- [H2O]
  - Easy to use API
  - WebUI
  - Supports reading from multiple datasources (Excel/SQL/HDFS)
  - In memory compute
  - Works on top of Spark ("Sparkling Water")
  - Vendor: 0xData
  - http://www.h2o.ai/

Tools for Scalable Deep Learning

- TensorFlow
  - Based on “data flow graphs”
  - “Tensor” = batches of data
  - Language support: Python, C++
  - Run time: CPU, GPU

- Intel BigDL
  - Deep learning library
  - Built on Apache Spark
  - Language support: Python, Scala

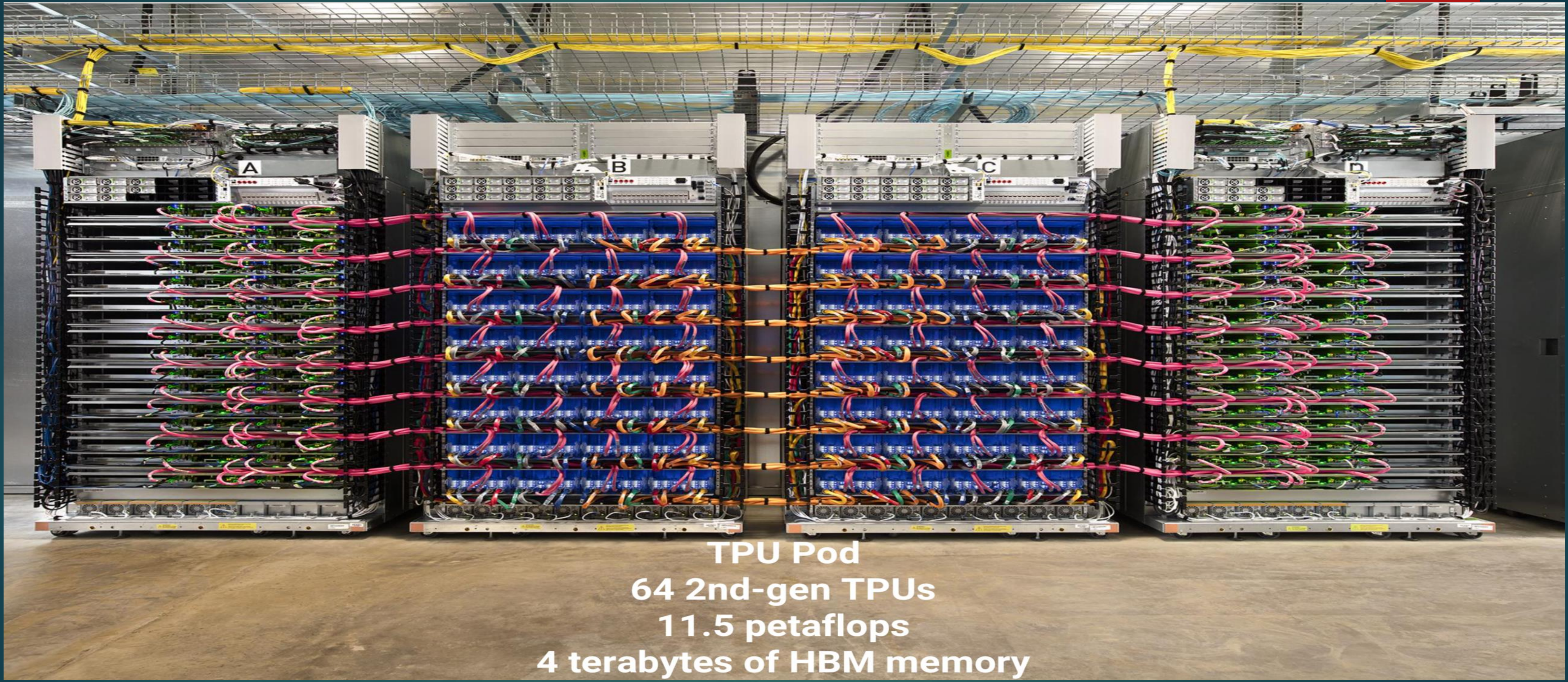
Hardware Progression

- CPU
  - Moore’s law
  - Number of transistors x2 in 2 years
  - Till 2012
- GPU
  - Performance x1000
  - Scala, Go
- ASIC
  - Application-specific integrated circuit
- Computation-specific hardware

Hardware – TPU (Tensor Processing Unit)

- A Tensor processing unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning
- More capable than CPUs or GPUs in certain tasks
- Designed for Tensorflow
- Designed for high volume computes
  - A TPU can process 100 million photos a day
- Available in Google Cloud platform

Google TPU Computer



AI Chips

- Azure
  - A New Era in Computer Architecture - Doug Burger
  - https://www.youtube.com/watch?v=iJo_sSzioxM
- Intel+Facebook “Nervana”
  - NNP – Neural Network Processor
  - Pre-trained learning
- Nvidia is the current market leader
- Amazon (”Inferentia”)
- Alibaba
- Startups

DGA data set example using Python notebook included as next five slides

Domain Generation Algorithm Attack

DGA. (2019). Domain Generation Algorithm. Retrieved May 2019 from https://en.wikipedia.org/wiki/Domain_generation_algorithm

"According to network security firm Damballa, the top-5 most prevalent DGA-based crimeware families are Conficker, Murofet, BankPatch, Bonnana and Bobax as of 2011.

"Domain generation algorithms (DGA) are algorithms seen in various families of malware that are used to periodically generate a large number of domain names that can be used as rendezvous points with their command and control servers. The large number of potential rendezvous points makes it difficult for law enforcement to effectively shut down botnets, since infected computers will attempt to contact some of these domain names every day to receive updates or commands. The use of public-key cryptography in malware code makes it unfeasible for law enforcement and other actors to mimic commands from the malware controllers as some worms will automatically reject any updates not signed by the malware controllers."

The specified Python notebook (next 4 slides) is a Machine Learning approach to

```
In [ ]: import pandas as pd
        import numpy as np
        import gensim
        from sklearn.pipeline import Pipeline
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.feature_extraction.text import TfidfTransformer
        from sklearn.model_selection import train_test_split
        from sklearn.ensemble import RandomForestClassifier
```

```
In [ ]: dga = pd.read_csv('dga-dataset-words.csv')
        dga.words = dga.words.fillna('')
        dga
```

```
In [ ]: # source is not a number, so transform it into an number
        dga['source_fact'] = pd.factorize(dga['source'])[0]

        # toplevel is not a number, so transform it into a number
        dga['toplevel_fact'] = pd.factorize(dga['toplevel'])[0]

        dga['label_fact'] = pd.factorize(dga['label'])[0]

        # get length of site as a new engineered featrues
        dga['url_length'] = dga['site'].apply(lambda x : len(x))

        # get num of words as a new engineered featrues
        dga['word_num'] = dga['words'].apply(lambda x : len(x.split()))
```

```
In [ ]: dga
```

```
In [ ]: dga.describe()
```

TF/IDF Pipeline

Let's try a basic tf/idf pipeline without using any of our other features

In [ ]:

More Ca

```
from sklearn.linear_model import SGDClassifier

X_train, X_test, y_train, y_test = train_test_split(dga['words'], dga['label'], test_size=0.33, random_state=42)
text_clf = Pipeline([('vect', CountVectorizer()),
                     ('tfidf', TfidfTransformer()),
                     ('clf', SGDClassifier(loss='hinge', penalty='l2', alpha=1e-3, random_state=42)),
])
text_clf.fit(X_train, y_train)
predicted = text_clf.predict(X_test)
np.mean(predicted == y_test)
```

In [ ]:

```
from sklearn.metrics import confusion_matrix

confusion_matrix(y_test, predicted)
```

Results

77% accuracy, not bad. But not great. Looks like we were much better at identifying one class than the other.

Extract features

In [ ]:
```
text_clf = Pipeline([('vect', CountVectorizer()),
                     ('tfidf', TfidfTransformer())])

text_clf = text_clf.fit(dga['words'])
dga['tfidf'] = text_clf.transform(dga['words'])
tfidf = text_clf.transform(dga['words'])
dga
```

Train/Test Split

Let's do a basic train/test split 80% training / 10% test

In [ ]:
```



msk = np.random.rand(len(dga)) < 0.8
train = dga[msk]
test = dga[~msk]

train_tfidf = tfidf[msk]
test_tfidf = tfidf[~msk]
```

In [ ]:
```
train
```

In [ ]:
```
from scipy import sparse

text_features = train_tfidf
other_features = train[['source_fact', 'toplevel_fact', 'url_length', 'word_num']]
all_features = sparse.hstack((text_features, other_features)).tocsr()
```

```
In [ ]: print(dga.shape)
        print(text_features.shape)
        print(other_features.shape)
        print(tfidf.shape)
```

```
In [ ]: mixed_classifier = SGDClassifier(loss='hinge', penalty='l2', alpha=1e-3, random_state=42).fit(all_features, train['lab
        el'])
```

More Capture

```
In [ ]: text_features_test = test_tfidf
        other_features_test = test[['source_fact', 'toplevel_fact', 'url_length', 'word_num']]
        all_features_test = sparse.hstack((text_features_test, other_features_test)).tocsr()

        predicted =  mixed_classifier.predict(all_features_test)
        np.mean(predicted == test['label'])
```

```
In [ ]: ## Cool 86% -- that's better.
        confusion_matrix(test['label'], predicted)
```

Results

86% Results are much more balanced too. The engineered features must have helped.

TODO:

We should try some other methods, like random forest classifier or a DNN classifier.

```
In [ ]: from sklearn.ensemble import RandomForestClassifier

        rf = RandomForestClassifier(n_estimators=100, oob_score=True, random_state=123456)
        rf.fit(all_features, train['label'])
        predicted_rf = rf.predict(all_features_test)
        np.mean(predicted_rf == test['label'])
```

Confusion Matrices

Anatomy of a Confusion Matrix

```
Confusion Matrix and Statistics

                  Reference
Prediction             0        1
True Negatives 0  285626    11469  False Negatives (Type II Errors)
False Positives 1 158409    52505  True Positives
(Type I Errors)

               Accuracy : 0.6656   (TP+TN) / (TP+TN+FP+FN)
                 95% CI : (0.6643, 0.6669)
    No Information Rate : 0.8741  Accuracy if predicted the most oft-occurring result all the time)
    P-Value [Acc > NIR] : 1

                  Kappa : 0.234

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.8207
            Specificity : 0.6433
         Pos Pred Value : 0.2489
         Neg Pred Value : 0.9614
             Prevalence : 0.1259
         Detection Rate : 0.1034
   Detection Prevalence : 0.4152
      Balanced Accuracy : 0.7320  Accuracy score that accounts for imbalances)

       'Positive' Class : 1
```

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

Example round of bootstrapping:

xgboost:

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 437333      5
         1   6373  64298

               Accuracy : 0.9874
                 95% CI : (0.9871, 0.9877)
    No Information Rate : 0.8734
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.9455

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9999
            Specificity : 0.9856
         Pos Pred Value : 0.9098
         Neg Pred Value : 1.0000
             Prevalence : 0.1266
         Detection Rate : 0.1266
   Detection Prevalence : 0.1391
      Balanced Accuracy : 0.9928

       'Positive' Class : 1
```

dbn:

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 437403      8
         1   6303  64295

               Accuracy : 0.9876
                 95% CI : (0.9873, 0.9879)
    No Information Rate : 0.8734
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.9461

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9999
            Specificity : 0.9858
         Pos Pred Value : 0.9107
         Neg Pred Value : 1.0000
             Prevalence : 0.1266
         Detection Rate : 0.1266
   Detection Prevalence : 0.1390
      Balanced Accuracy : 0.9928

       'Positive' Class : 1
```

ensemble:

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 437414      6
         1   6292  64297

               Accuracy : 0.9876
                 95% CI : (0.9873, 0.9879)
    No Information Rate : 0.8734
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.9462

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9999
            Specificity : 0.9858
         Pos Pred Value : 0.9109
         Neg Pred Value : 1.0000
             Prevalence : 0.1266
         Detection Rate : 0.1266
   Detection Prevalence : 0.1390
      Balanced Accuracy : 0.9929

       'Positive' Class : 1
```

Skipping straight to Deep Learning Will Be Frustrating – it is harder

xgboost mediocre technique

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 437782     22
         1   6156  64049

               Accuracy : 0.9878
                 95% CI : (0.9875, 0.9881)
    No Information Rate : 0.8739
    P-Value [Acc > NIR] : < 0.0000000000000022

                  Kappa : 0.947

 Mcnemar's Test P-Value : < 0.0000000000000022

            Sensitivity : 0.9997
            Specificity : 0.9861
         Pos Pred Value : 0.9123
         Neg Pred Value : 0.9999
             Prevalence : 0.1261
         Detection Rate : 0.1261
   Detection Prevalence : 0.1382
      Balanced Accuracy : 0.9929

       'Positive' Class : 1
```

DBN with same mediocre data prep technique

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 443938  64071
         1      0      0

               Accuracy : 0.8739
                 95% CI : (0.873, 0.8748)
    No Information Rate : 0.8739
    P-Value [Acc > NIR] : 0.5011

                  Kappa : 0

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.8739
             Prevalence : 0.1261
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 1
```

Predicts non-intrusion for everything

DEVIL in the Details:

DBN: Subtle Data Prep Oversights*

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 285626  11469
         1 158409  52505

               Accuracy : 0.6656
                 95% CI : (0.6643, 0.6669)
    No Information Rate : 0.8741
    P-Value [Acc > NIR] : 1

                  Kappa : 0.234

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.8207
            Specificity : 0.6433
         Pos Pred Value : 0.2489
         Neg Pred Value : 0.9614
             Prevalence : 0.1259
         Detection Rate : 0.1034
   Detection Prevalence : 0.4152
      Balanced Accuracy : 0.7320

       'Positive' Class : 1
```

DBN: All Trellis steps followed

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 437403      8
         1   6303  64295

               Accuracy : 0.9876
                 95% CI : (0.9873, 0.9879)
    No Information Rate : 0.8734
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.9461

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9999
            Specificity : 0.9858
         Pos Pred Value : 0.9107
         Neg Pred Value : 1.0000
             Prevalence : 0.1266
         Detection Rate : 0.1266
   Detection Prevalence : 0.1390
      Balanced Accuracy : 0.9928

       'Positive' Class : 1
```

DID: transformations, proper scaling of initial numeric variables, balancing, one-hot encoding, Boruta feature selection.
MISSED: 1) did not simplify categorical levels before one-hot encoding, 2) did not scale the 3 PCA dimensions to 0,1 range.