Big Data and You

Wesley Emeneker
wesley@emeneker.com
wesley.d.emeneker.ctr@mhpcc.hpc.mil

June 23, 2022

Warning

Since this is a keynote, I don't have to impart anything technically correct.

Warning

Since this is a keynote, I don't have to impart anything technically correct.
Nor useful.

When does **data** become **Big Data**?

Simple answer: When doing simple things with it becomes hard due to

- its size
- your team size
- the data's cleanliness
- the data's organization and indexing
- your cleanliness
- your ability to ask questions of the data

When does **data** become **Big Data**? An Example

When do simple things become hard?

When does **data** become **Big Data**? An Example

When do simple things become hard?
Count the number of people in the room right now.

When does **data** become **Big Data**? An Example

When do simple things become hard?
Count the number of people in the room right now.
When would "Count the number of people in the XXX" become hard?

Describing **Big Data**

The 4 "V"s are often : velocity, volume, veracity, variety.

Cyber has problems with all of these things in spades.

Big Data

The new oil™!
What do we do with it?

Big Data

The new oil™!
What do we do with it?
Make lots of money with targeted advertising!

Big Data

The new oil™!
What do we do with it?
Make lots of money with targeted advertising!
Once we do the necessary refinement of the data.

Some things we have tried

- Problem: We have too much data for any human to tease through and understand

Some things we have tried

- Problem: We have too much data for any human to tease through and understand
- Solution: We use computers to find patterns and make predictions

Some things we have tried

- Problem: We have too much data for any human to tease through and understand
- Solution: We use computers to find patterns and make predictions
- Question: Can the computers explain why they give an answer?

Some things we have tried

- Problem: We have too much data for any human to tease through and understand
- Solution: We use computers to find patterns and make predictions
- Question: Can the computers explain why they give an answer?
- Answer: Yes. Kind of.

Some things we have tried

- Problem: We have too much data for any human to tease through and understand
- Solution: We use computers to find patterns and make predictions
- Question: Can the computers explain why they give an answer?
- Answer: Yes. Kind of.
- Problem: We can't understand why the computer gives us the answer even after it explains why.

Some things we have tried

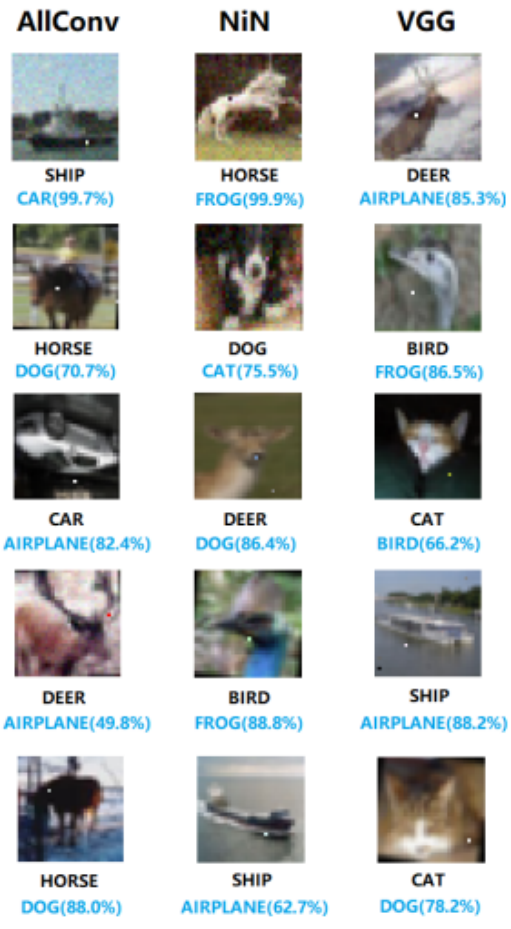
- Problem: We have too much data for any human to tease through and understand
- Solution: We use computers to find patterns and make predictions
- Question: Can the computers explain why they give an answer?
- Answer: Yes. Kind of.
- Problem: We can't understand why the computer gives us the answer even after it explains why.
- Solution: ¯_(ツ)_/¯

The Magic of Machine Learning

Imagined by a GAN (generative adversarial network). StyleGAN2 (Dec 2019) - Karras et al. and Nvidia

Some images courtesy of https://thispersondoesnotexist.com/

The Dangers of Machine Learning



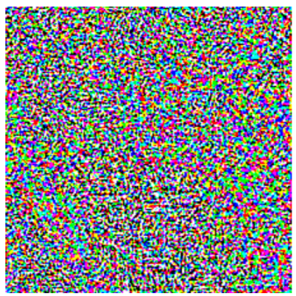
One Pixel Attack for Fooling Deep Neural Networks. Su, Vargas, Sakurai, IEEE Transactions on Evolutionary Computation 2019

The Dangers of Machine Learning

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Explaining and Harnessing Adversarial Examples. Goodfellow, Shlens, Szegedy, ICLR 2015

Big Data Analysis steps

What is the hard part? Getting the data so you can work on it.
What do you do after the hard part?

Big Data Analysis steps

What is the hard part? Getting the data so you can work on it.
What do you do after the hard part?

You do the actual hard part. Correctly analyzing and learning from the data.

Big Data Analysis steps

What is the hard part? Getting the data so you can work on it.
What do you do after the hard part?

You do the actual hard part. Correctly analyzing and learning from the data.
Don't fool yourself! You are the easiest person to fool!
What do you do after the actual hard part?

Big Data Analysis steps

What is the hard part? Getting the data so you can work on it.
What do you do after the hard part?

You do the actual hard part. Correctly analyzing and learning from the data.
Don't fool yourself! You are the easiest person to fool!
What do you do after the actual hard part?

You do the actual, no-kidding hard part.
Getting other people in your organization to believe the analysis, understand it, and act on it. What do you do after the actual, no-kidding hard part?

Big Data Analysis steps

What is the hard part? Getting the data so you can work on it.
What do you do after the hard part?

You do the actual hard part. Correctly analyzing and learning from the data.
Don't fool yourself! You are the easiest person to fool!
What do you do after the actual hard part?

You do the actual, no-kidding hard part.
Getting other people in your organization to believe the analysis, understand it, and act on it. What do you do after the actual, no-kidding hard part?

You do the actual, no-kidding, final, pinky-swear hard part.

Big Data Analysis steps

What is the hard part? Getting the data so you can work on it.
What do you do after the hard part?

You do the actual hard part. Correctly analyzing and learning from the data.
Don't fool yourself! You are the easiest person to fool!
What do you do after the actual hard part?

You do the actual, no-kidding hard part.
Getting other people in your organization to believe the analysis, understand it, and act on it. What do you do after the actual, no-kidding hard part?

You do the actual, no-kidding, final, pinky-swear hard part.
Do all of the above from the start, at the same time. The above steps cannot correctly, reliably be done in sequence.

Guidelines and Best Practices for Scaling Big Data

Start as simply and generically as possible.[1]

[1] https://adamdrake.com/command-line-tools-can-be-235x-faster-than-your-hadoop-cluster.html

Guidelines and Best Practices for Scaling Big Data

Start as simply and generically as possible.[1]
If you expect to have $<$ 1 petabyte of data in the next 2 years, get a generic network-mounted storage server. Put SSDs in the storage server. NOT HDDs!

[1] https://adamdrake.com/command-line-tools-can-be-235x-faster-than-your-hadoop-cluster.html

What to do if you don't have a full-time staff

- Use SQLite for indexing and cataloging your data
- Use a network storage server as reliable and low-latency as possible
- Do **NOT** use any network services (web servers, indexers, databases, dashboarding, etc.)
- Spend a lot of time initially thinking about what answers you want from the data, and how to formulate those questions

What to do if you have a full-time staff for Big Data handling?

Below are the vaguest guidelines based on my 20+ years. YMMV.[2]

[2] It depends

What to do if you have a full-time staff for Big Data handling?

Below are the vaguest guidelines based on my 20+ years. YMMV.[2]

- If you have 1 FTE, spend them on the serving the data reliably. This can make up to 2-ish petabytes of usable, useful data.[2]
- If you have 2 FTEs, spend #2 on data engineering (HW+SW+cleaning+analysis). This can make up to 5-ish petabytes of usable, useful analysis.[2]
- If you have 3 FTEs, spend #3 on indexing, cataloging, searching, cleaning, and curating (see SQLite).[2]
- If you have 4 or more FTEs, use the expertise of 1-3 to determine the current and future needs based on hard-won domain expertise.[2]

[2] It depends

When should you use the Hadoop stack?

- When you have 4+ dedicated staff for operations and maintenance.
- When you have 30+ petabytes of data.
- When you can afford to wait for answers (e.g. no interactivity).
- When you can spend months becoming MR/Spark programming experts and can spend a lot of time optimizing.
- When the data doesn't decay rapidly; e.g. it has a long lifespan of utility.

When should you use the Hadoop stack?

- When you have 4+ dedicated staff for operations and maintenance.
- When you have 30+ petabytes of data.
- When you can afford to wait for answers (e.g. no interactivity).
- When you can spend months becoming MR/Spark programming experts and can spend a lot of time optimizing.
- When the data doesn't decay rapidly; e.g. it has a long lifespan of utility.

The Hadoop stack is **great** in some circumstances.

Q&A

- What software, hardware, or solutions do you recommend for **Big Data** in Cyber?[1]
- What do you recommend for CyberSecurity? SIEM tools?[1]
- What do you recommend?[1]

[1] It depends