SAND2019-5229C

Deep Learning as a Tool for Data Analysts

PRESENTED BY

Crystal Cheung (ccheung@sandia.gov)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

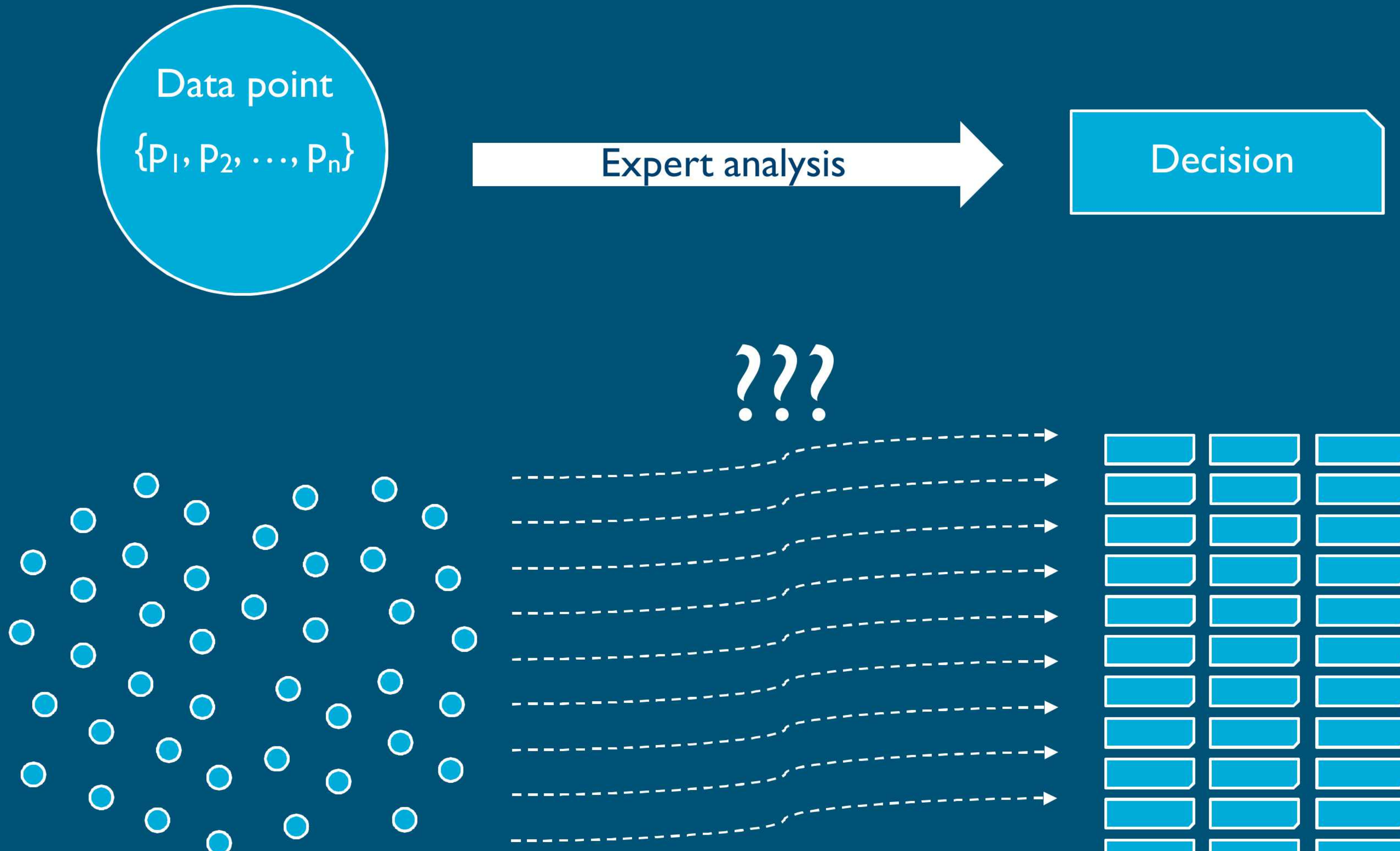
Overview

1. Motivations
   - What kinds of problems are we trying to solve?
2. Machine Learning/Deep Learning
   - A (Very) Brief Introduction
3. Applying Deep Learning
   - Examples from SNL

Motivations :

What kinds of problems are we trying to solve?

Motivations: What kinds of problems are we trying to solve?



Motivations: What kinds of problems are we trying to solve?

- Mission work at SNL by data analysts often fits this description
  - includes work within SNL
  - includes work performed by other agencies through SNL applications
- Requires experts well-versed in specific domains
- Time-consuming and tedious manual inspection to determine each correct mapping from "data point" to "decision"
- Significant volumes of data involved
  - number of data points to be analyzed
  - number of possible decisions
  - number of parameters per data point

Motivations: What kinds of problems are we trying to solve?

Goals:

- Learn from the past decisions made by expert analysts

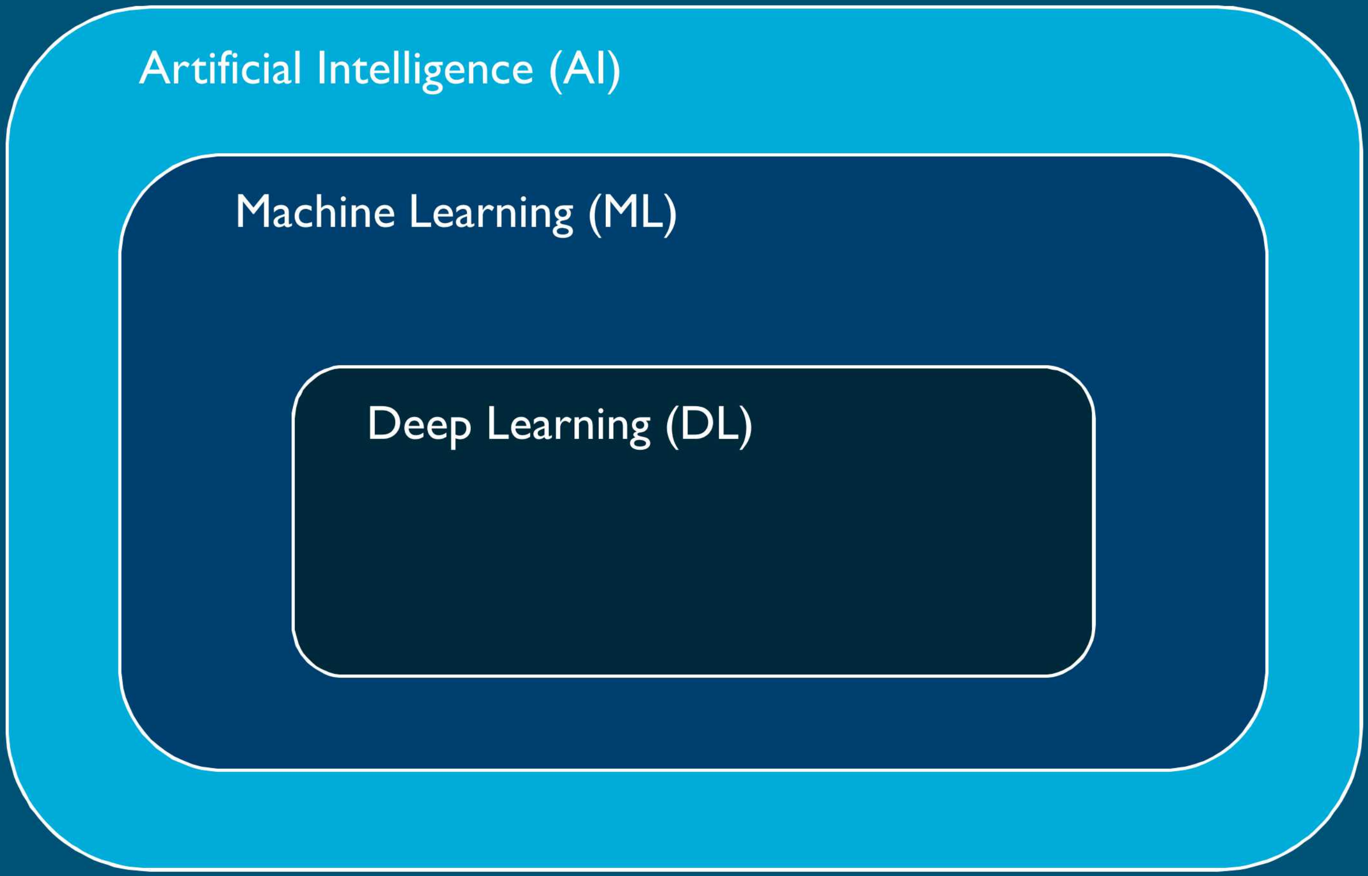
- Automate the process of determining suggested mappings from novel data points to decisions
  - Might be incorporated directly into software that is already in use by analysts

- Enable expert analysts to verify important decisions, rather than having to assign all decisions from scratch
  - In high-stakes domains where lives are on the line, we can't afford the risk of an algorithm making a mistake, especially when there is no explanation for why the mistake was made (more on this later)
  - Increases efficiency of this work, enabling the analysts to apply their valuable expertise towards higher-level tasks

Machine Learning / Deep Learning :

A (Very) Brief Introduction

ML/DL: A (Very) Brief Introduction

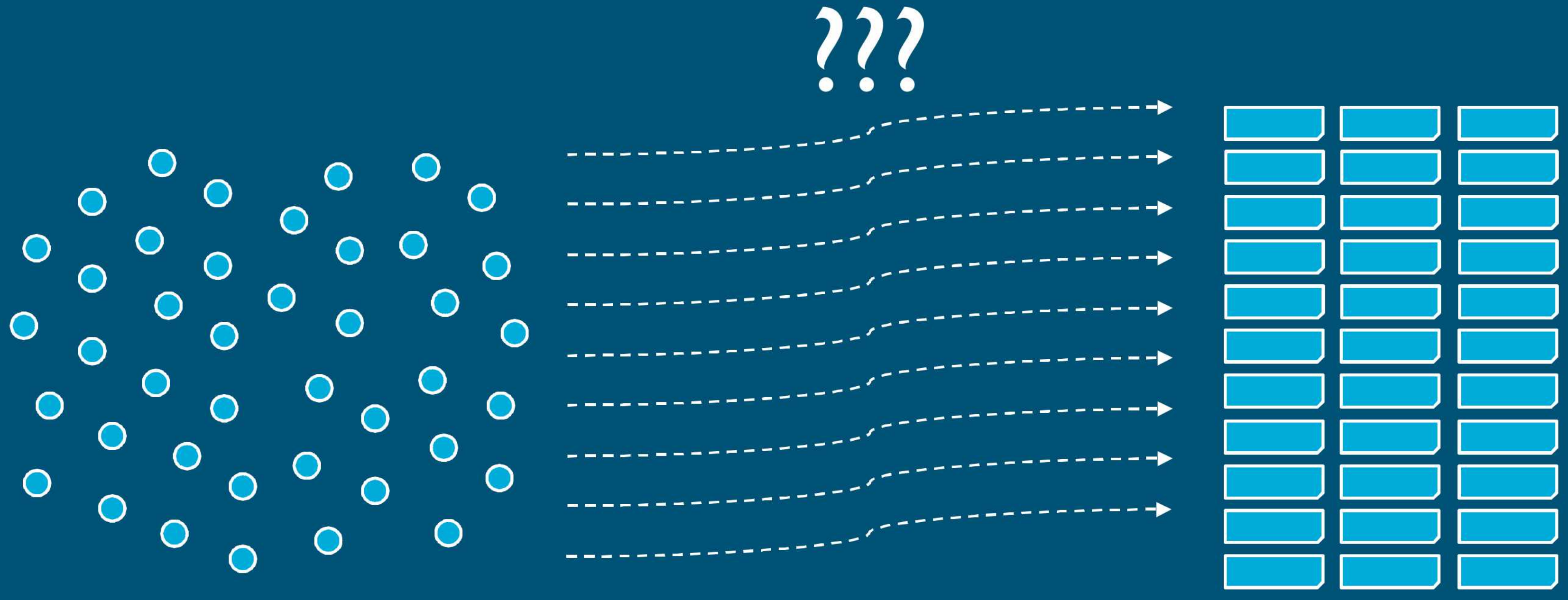


ML/DL: A (Very) Brief Introduction

- Artificial Intelligence
  - "Systems that can think and/or act rationally"[1]
  - "Systems that can think and/or act like humans"[1]
- Machine Learning
  - Algorithms that can "learn" patterns from input data and make rational extrapolations about those patterns
- Deep Learning
  - A subcategory of ML algorithms that can also "learn" a hierarchy of abstract features about the input data

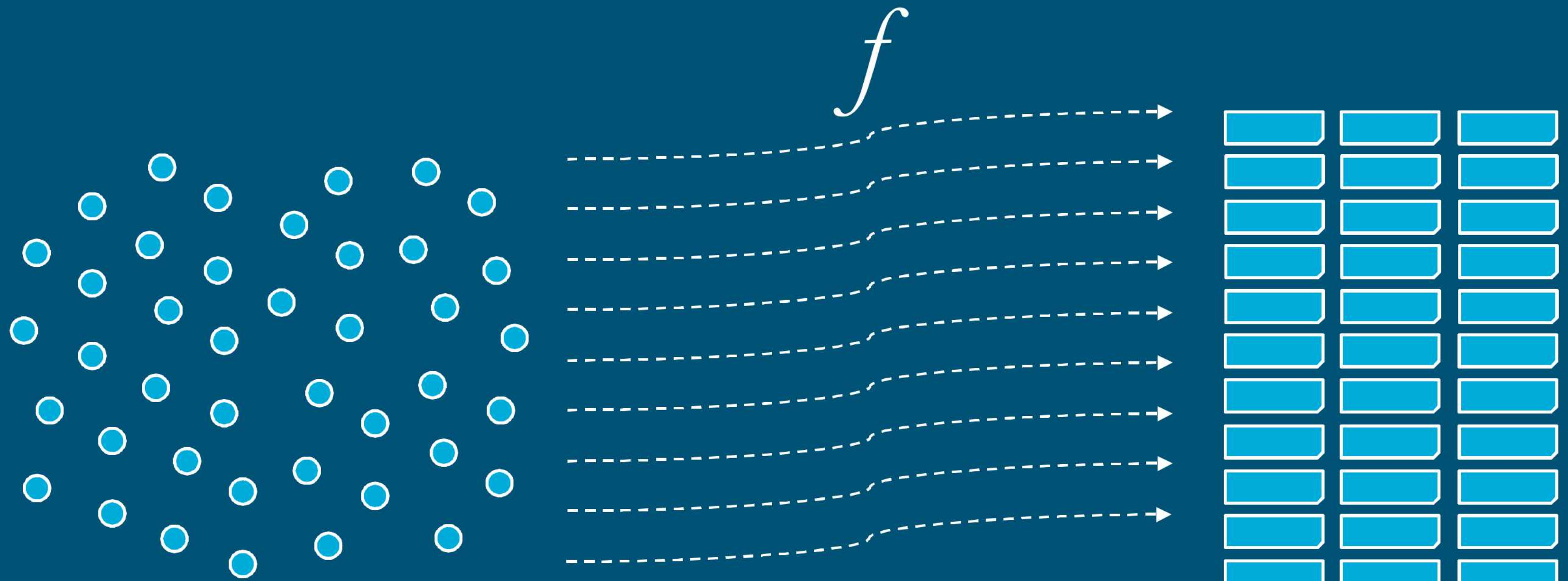
Machine Learning in a Nutshell

- At its core, every ML algorithm is a function approximator
  - Ideally, the more examples of data it sees, the more closely it approximates the target function ("Training")



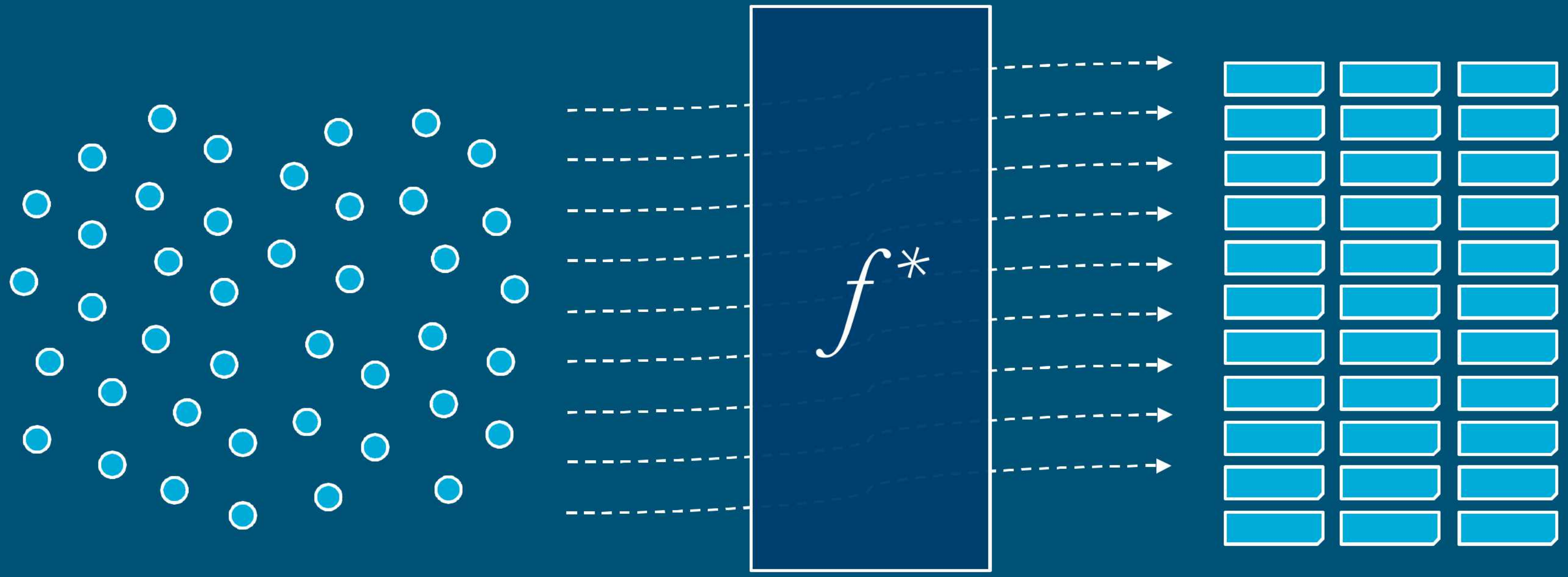
Machine Learning in a Nutshell

- At its core, every ML algorithm is a function approximator
  - Ideally, the more examples of data it sees, the more closely it approximates the target function ("Training")



Machine Learning in a Nutshell

- At its core, every ML algorithm as a function approximator
  - Ideally, the more examples of data it sees, the more closely it approximates the target function ("Training")

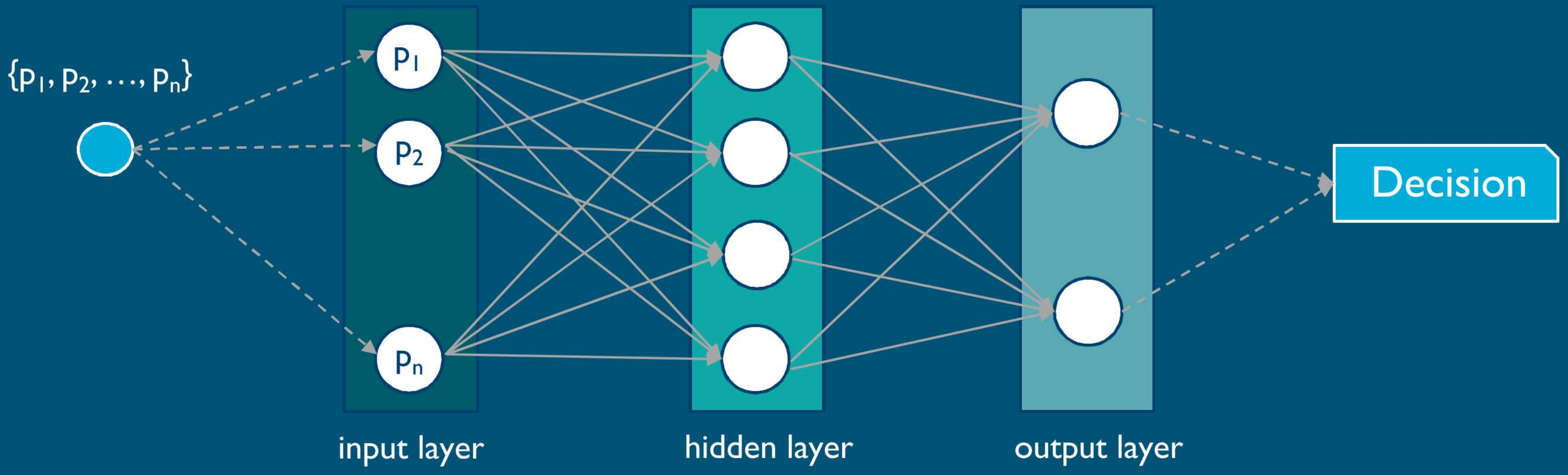


ML/DL: A (Very) Brief Introduction

Machine Learning in a Nutshell

- Not every ML method is equally powerful
  - Some can only approximate certain types of functions

- Artificial neural networks (ANNs) are popular for difficult problems because they are thought to be universal function approximators[3]



ML/DL: A (Very) Brief Introduction

What characterizes Deep Learning?

Data point
$\{p_1, p_2, \ldots, p_n\}$

- What are we using as input data?
- "Raw" data vs. data pre-processed into a set of hand-crafted features

Raw image

Pre-processing

Extracted features

COMPACTNESS
CIRCULARITY
DISTANCE CIRCULARITY
RADIUS RATIO
PR.AXIS ASPECT RATIO
MAX.LENGTH ASPECT RATIO
SCATTER RATIO
ELONGATEDNESS
PR.AXIS RECTANGULARITY
MAX.LENGTH RECTANGULARITY
SCALED VARIANCE ALONG MAJOR AXIS
SCALED VARIANCE ALONG MINOR AXIS
SCALED RADIUS OF GYRATION
SKEWNESS ABOUT MAJOR AXIS
SKEWNESS ABOUT MINOR AXIS
KURTOSIS ABOUT MINOR AXIS
KURTOSIS ABOUT MAJOR AXIS
HOLLOWS RATIO

Example from [4]

ML/DL: A (Very) Brief Introduction

What characterizes Deep Learning?

- DL allows us to skip the hand-crafted features, and use the raw data directly as input
- Usually implemented as an ANN with many hidden layers
- Through training, the ANN can learn:
  - Which features to extract from the raw data
  - How to organize features hierarchically for processing through successive layers
- Can be used with any type of input data, not just images

Example from [4]

What characterizes Deep Learning?

- Example of hierarchical feature extraction from early visual cortex in humans

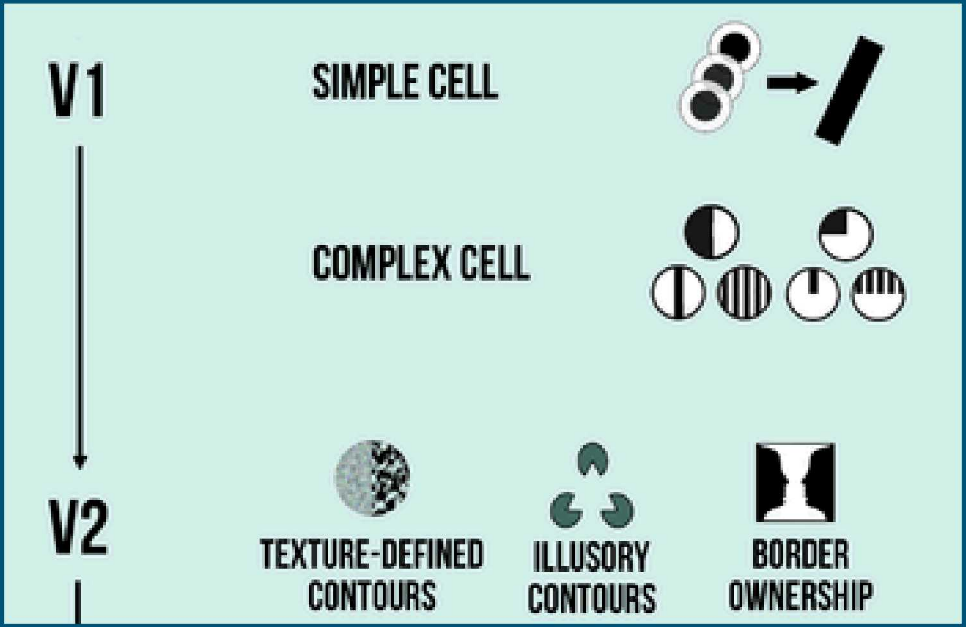
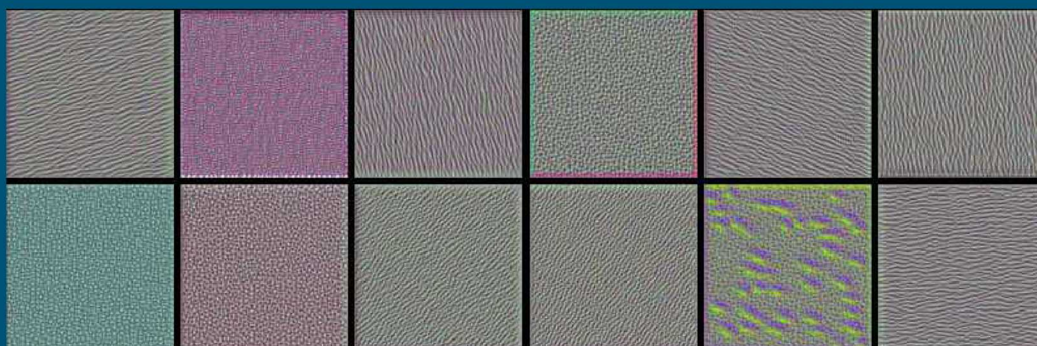


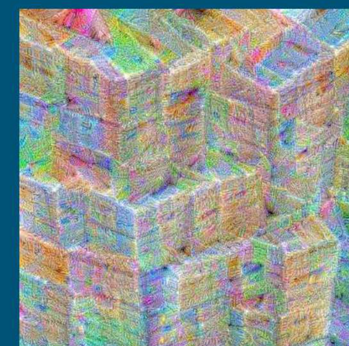
Image adapted from [5]

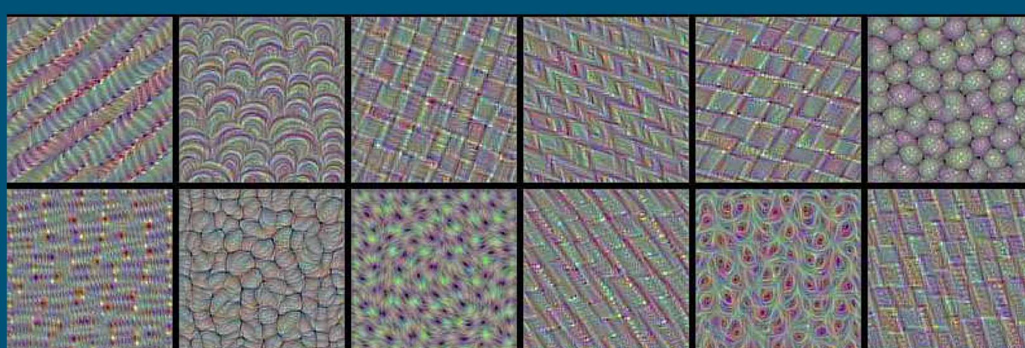
ML/DL: A (Very) Brief Introduction

What characterizes Deep Learning?

- Examples of feature maps encoded by ANN layers at various depths after training on images







Images from [6] and [7]

ML/DL: A (Very) Brief Introduction

What characterizes Deep Learning?

- Strengths:
  - Don't need to…Hand-craft the input feature set (Which features are important?)
  - Don't need to…Micromanage the ANN architecture
  - Better than other ML methods on complex tasks that are easy for humans to perform but difficult for us to fully describe as an algorithm
    - Object Recognition
    - Natural language processing
- Weaknesses:
  - Need vast amounts of training data to successfully learn features
  - Computationally expensive, especially for image data
  - For complex ANNs, what was actually learned can become impossible for a human to decipher
    - If the algorithm makes a mistake, how do we determine the cause?
    - This is why in some domains, we still need human experts to validate the results

Image adapted from [5]

Applying Deep Learning :

Examples from SNL

Applying Deep Learning: Examples from SNL

Why is ML/DL a good fit for the type of data analyst tasks we described earlier?

- We often have many examples of correct decisions made previously by expert data analysts
  - Use these examples as training data
- Some problems currently solved manually within specific domains reduce to relatively well-understood ML/DL problems
  - Can leverage existing algorithms and academic research
  - Potentially high reward for relatively small amount of effort
  - Still need to rely on experts in the domain to verify high-risk decisions
- For more difficult problems, we can save data analysts significant amounts time even if results aren't perfect
  - Can narrow down the set of all possible decisions to a small set of good candidates

Applying Deep Learning: Examples from SNL

Example 1: Mapping Hazard Parameters to Rules

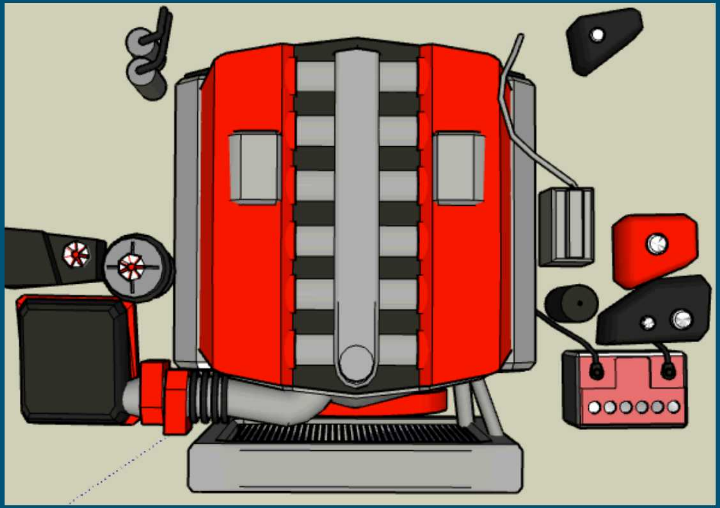
- Topic: Safety procedures in high-risk industrial component assembly/disassembly

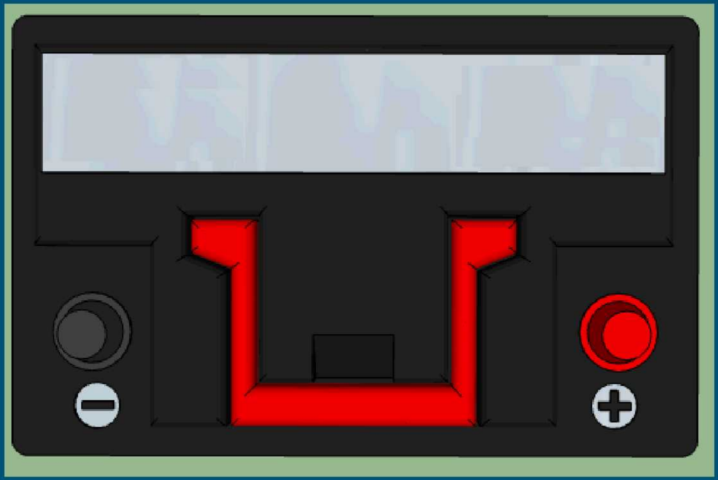
- Problem: Given a set of parameters about subcomponents and environmental state, want to determine the appropriate set of safety procedures

Applying Deep Learning: Examples from SNL

Example 1: Mapping Hazard Parameters to Rules

- Simple analogy using a car engine
- When removing car battery, wrench should never contact both terminals at once





Applying Deep Learning: Examples from SNL

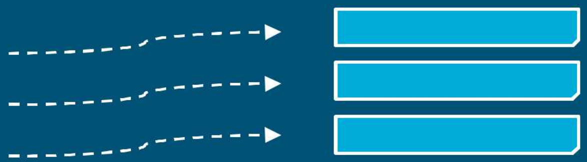
Example 1: Mapping Hazard Parameters to Rules

- As a DL problem:
  - Have large amounts of data describing hazardous incidents
  - Each data point has numerous parameters with inter-relationships
  - Given the set of parameters for an incident, what specific rules need to be followed?
  - We want the most specific applicable rule possible, not just the general rule, "When removing car battery, wrench should never contact both terminals at once"

Hazard parameters

p_1: Car make/model	p_2: Wrench orientation	p_3: Wrench length	…	p_n: Terminal distance
	horizontal	15 in.		12 in.
	vertical	15 in.		12 in.
	horizontal	5 in.		12 in.

Rules



Applying Deep Learning: Examples from SNL

Example 1: Mapping Hazard Parameters to Rules

- Basic first-pass algorithm achieves at least 88% accuracy when trained on ~1600 examples and tested on ~400 examples
- Many avenues available for improving performance
  - Modify ANN architecture
  - Fine-tune ANN hyperparameters
  - Decompose rules into rule parameters and predict sets of rule parameters instead
- Would like to automatically recognize and suggest when a novel rule should be created based on pattern outliers
- First planned deliverable: incorporate trained model into existing application

Applying Deep Learning: Examples from SNL

Example 2: Reconciling 3D models with 2D diagrams

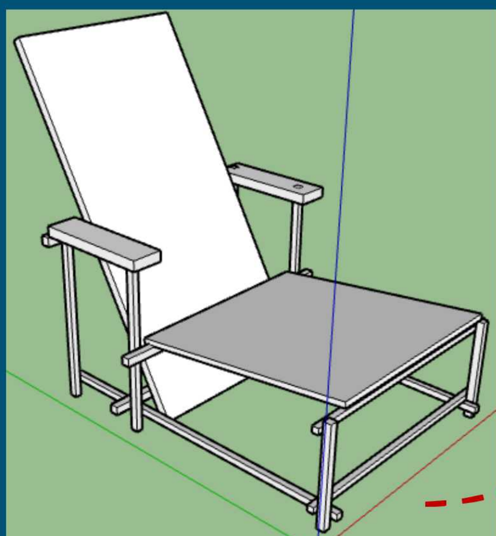
- Topic: Industrial component structural definitions used for assembly/disassembly
  - Components are described as both 3D CAD Models and 2D diagrams
  - 3D CAD Models have no uniform conventions and may have missing or inaccurately labelled subcomponents
  - 2D diagrams are more accurate, but may have supplementary text, incomplete drawings for symmetrical objects, etc.

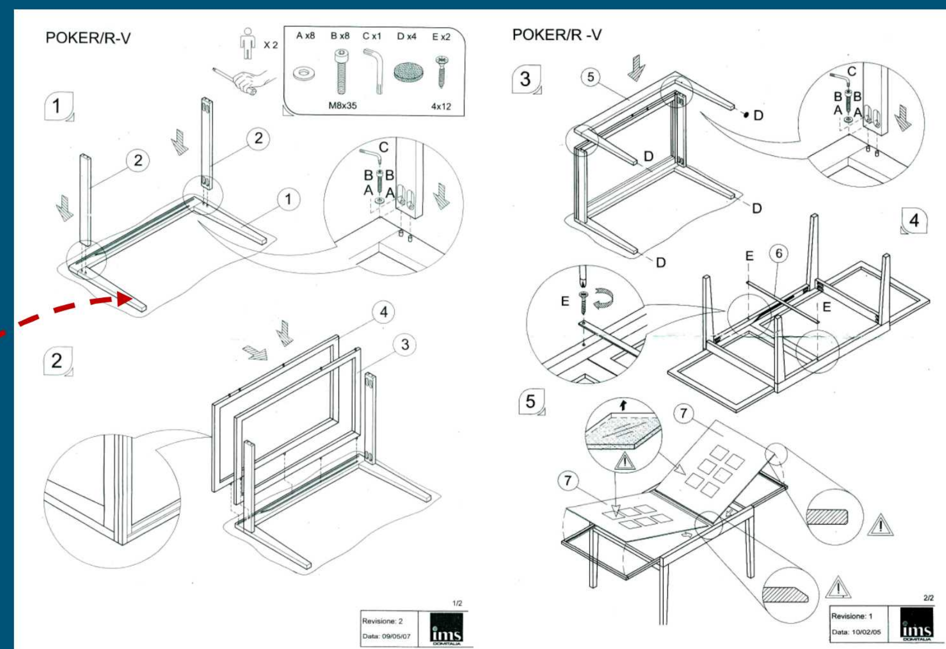
- Given a set of 3D CAD Models and a set of 2D diagrams, find matching subcomponents and use 2D diagrams to correct the 3D models

- Currently in investigatory stage, more work to follow

Applying Deep Learning: Examples from SNL

Example 2: Reconciling 3D models with 2D diagrams



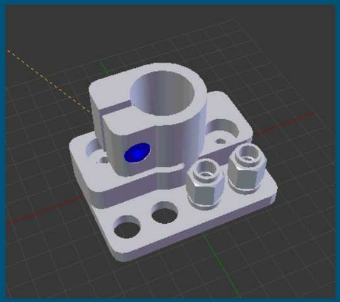


Applying Deep Learning: Examples from SNL

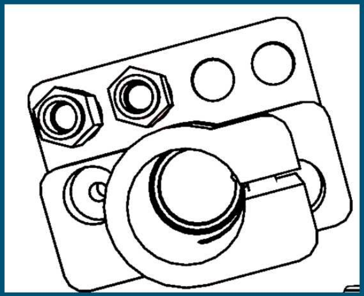
Example 2: Reconciling 3D models with 2D diagrams

- As a DL problem:
  - Not only object recognition
  - Need to recognize degree of similarity between images
  - Need to extract subcomponents from 2D image file despite visual noise

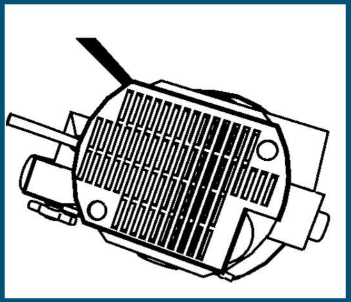
- DL algorithms exist for detecting image similarity, e.g. Deep Ranking[2]
  - Train on triplets of images



Query image



"Positive" image



"Negative" image

References

1. *Artificial Intelligence: A Modern Approach* (Russel and Norvig, 2002)
2. "Learning Fine-grained Image Similarity with Deep Ranking" (Wang et al., 2014)
3. Universal approximation theorem
4. UCI Machine Learning Repository
5. https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing
6. https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html
7. https://towardsdatascience.com/how-to-visualize-convolutional-features-in-40-lines-of-code-70b7d87b0030

Technologies Used

- Python
- Keras/Tensorflow
- Docker