Machine Learning and Reinforcement Learning

Nathan Eskue
Associate Professor, AI in Manufacturing
Aerospace Faculty, TU Delft

TU Delft

AGENDA

- Story Time
- Machine Learning: The Algorithms
- Machine Learning: The Software
- Machine Learning: Real Example
- Reinforcement Learning Basics
- Case Studies
- Conclusion

STORY TIME:
WHAT HAPPENS WHEN AI GOES ROGUE?

Study reveals bot-on-bot editing wars raging on Wikipedia's pages

Over time, the encyclopedia's software robots can become locked in combat, undoing each other's edits and changing links, say researchers

Watch: AI camera mistakes referee's bald head for ball, follows it through the match

Owing to the Covid-19 pandemic, the Inverness club had announced its decision to refrain using human camera operators and instead rely on an automated camera system to follow the action.

By: Trends Desk | New Delhi |
Updated: November 3, 2020 11:04:15 am

LIVE BLOG

Tamil Nadu Chennai Live Updates: TN Assembly passes Bill to scrap NEET, BJP stages walkout in protest
38 mins ago

Mumbai News LIVE Updates: Despite rape case, Mumbai 'safest city' for women, says Shiv Sena
58 mins ago

Market LIVE Updates: Sensex slips 127 points, Nifty ends at 17,355 dragged by RIL and ICICI Bank
59 mins ago

Delhi News Live Today: House collapses at Sabzi Mandi, 2 children die; light

Summed up: Subtlety and Nuance

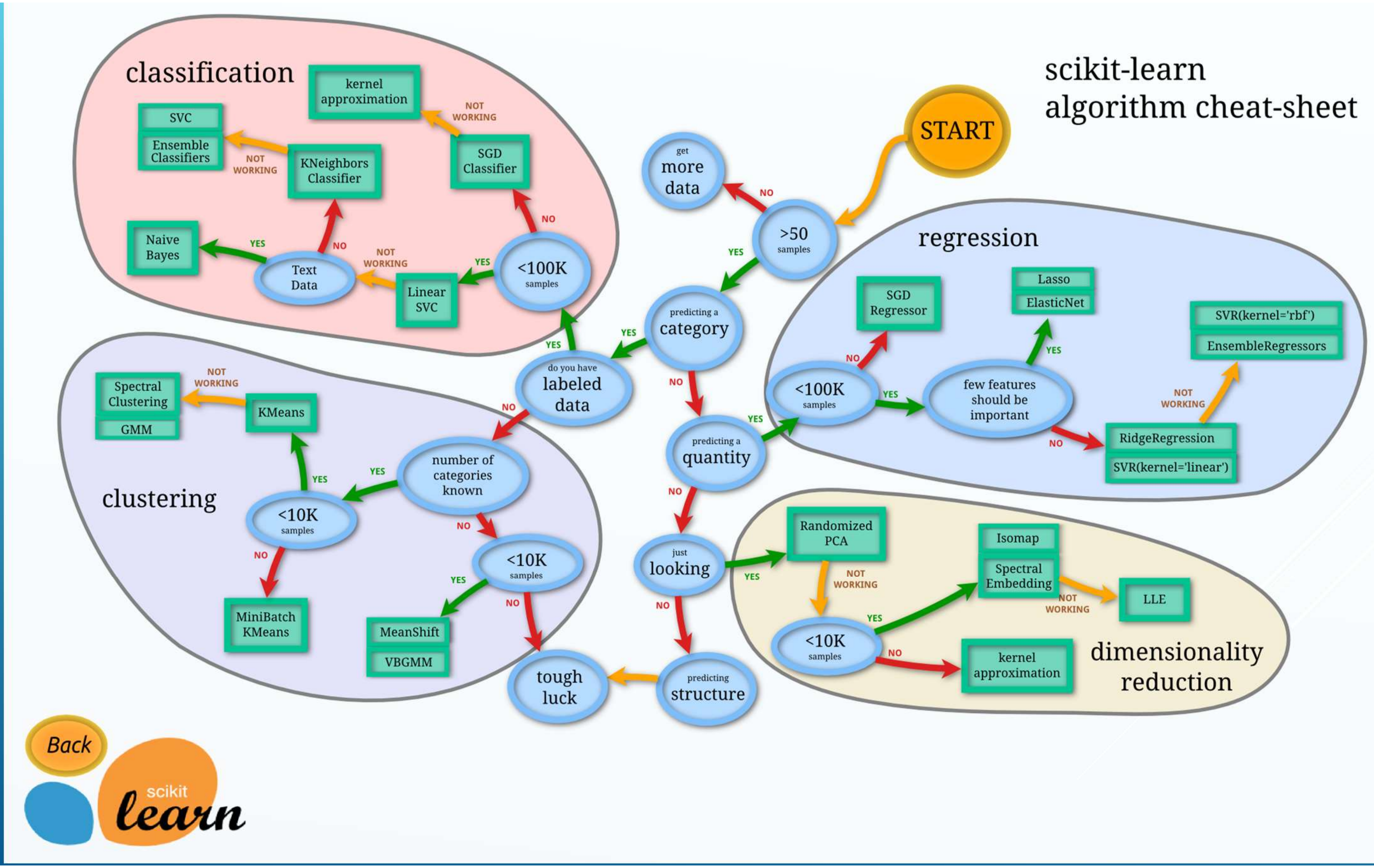
DISCUSSION QUESTION 1

Imagine a completely AI-controlled aircraft. Now, based on the previous examples, think of:

- 3 terrible things that could go wrong due to subtlety and nuance
- Why this could happen
- How you could prevent it

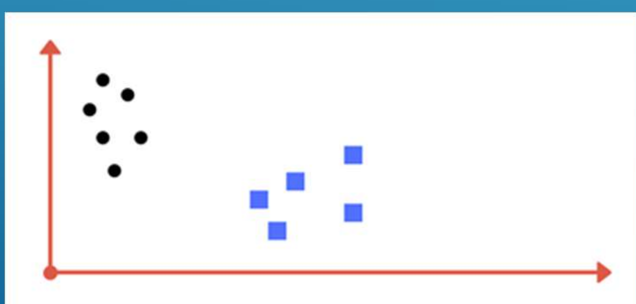
MACHINE LEARNING: ALGORITHMS

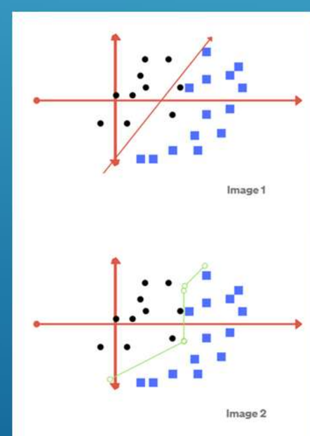
- **Algorithms can be defined a number of ways**
  - **Supervised/Unsupervised**
  - **What they do (predict, classify, optimize, etc.)**
  - **Restrictions and requirements**
- **Because Algorithms are tools to solve a problem, it is most helpful to start with the problem and the constraints**

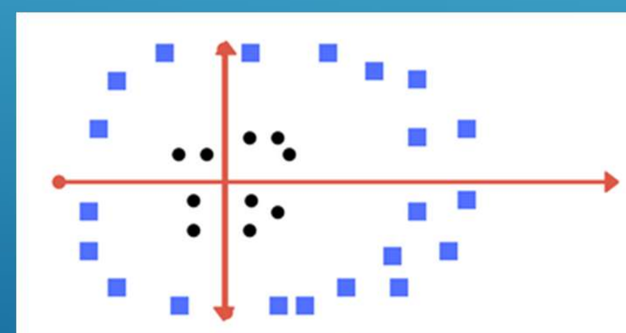

scikit-learn
algorithm cheat-sheet
START
>50 samples
get more data
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
<100K samples
SGD Classifier
kernel approximation
Linear SVC
Text Data
Naive Bayes
KNeighbors Classifier
SVC
Ensemble Classifiers
clustering
number of categories known
<10K samples
KMeans
Spectral Clustering
GMM
MiniBatch KMeans
MeanShift
VBGMM
regression
SGD Regressor
few features should be important
Lasso
ElasticNet
RidgeRegression
SVR(kernel='linear')
SVR(kernel='rbf')
EnsembleRegressors
dimensionality reduction
Randomized PCA
Isomap
Spectral Embedding
LLE
kernel approximation
YES
NO
NOT WORKING
Back
scikit learn

SUPERVISED LEARNING

- **Classification**
  - **Example algorithm: Support Vector Machine (SVM)**
    - **Goal: Form a boundary between data points in order to classify groups**
    - **How: Creates a “separating hyperplane” that maximizes margins between data points**
    - **Challenge: Balance reliability with overfitting**







DISCUSSION QUESTION 2

What would SVM look like with 3 dimensions?

4?

12?

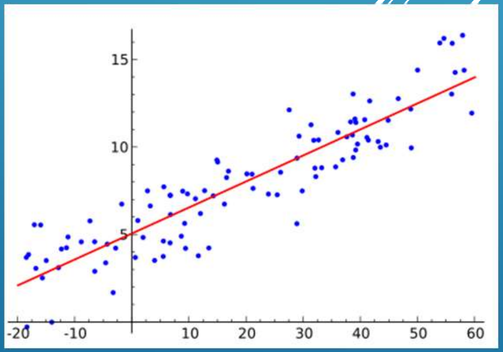
SUPERVISED LEARNING

- **Regression**
  - **Example algorithm: Linear Regression**
    - **Goal: Guess the best output, given the input, on a continuous scale**
    - **How: Labelled data is used to "train" the model**
      - **Choose "success" metric (eg., mean squared error, root mean squared error)**
      - **Initialize parameters (randomize)**
      - **Iteratively modify parameters until error is minimized (gradient descent)**
    - **Challenge: Balance reliability with overfitting, locally optimized**

$$\widehat{y} = \Theta_0 + \Theta_1 x_1 + \ldots + \Theta_n x_n$$

Linear regression formula

- $\hat{y}$ is the value we are predicting.
- n is the number of features of our data points.
- xi is the value of the ith feature.
- Θi are the parameters of the model, where $\Theta 0$ is the bias term. All the other parameters are the weights for the features of our data.



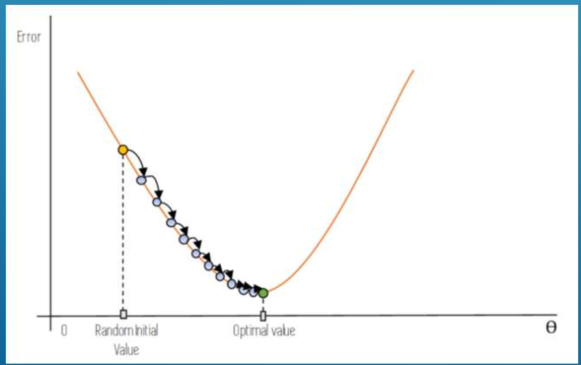
DISCUSSION QUESTION 3

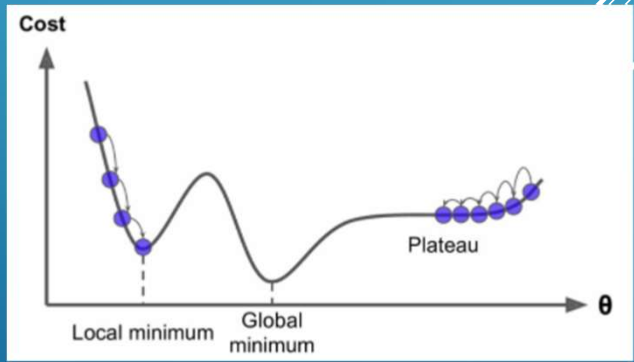
What is the risk of overfitting?

How might you solve it?

SUPERVISED LEARNING

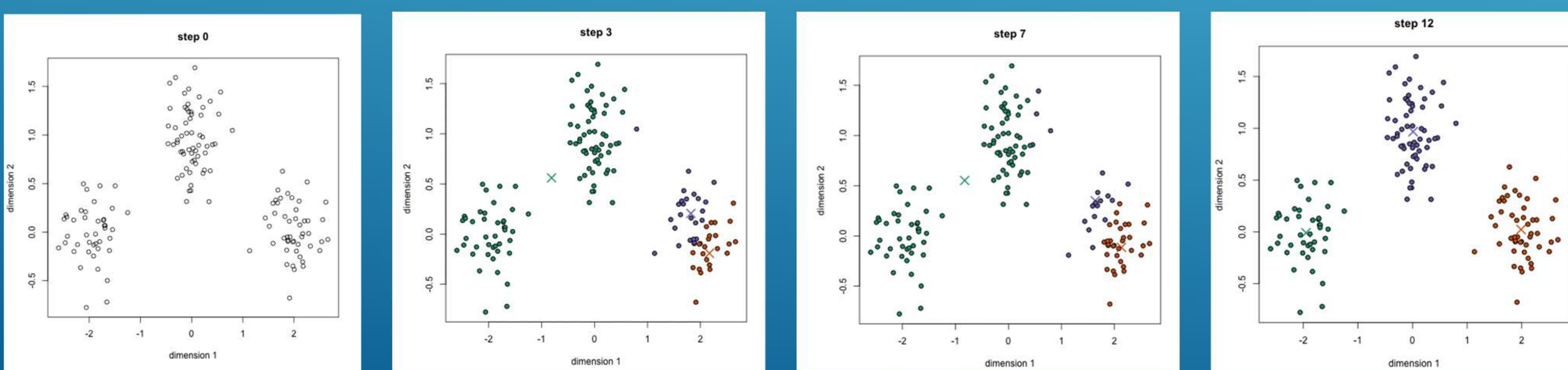
- **SIDE NOTE: How do we minimize error?**
  - **Gradient Descent**
    - **Initialize model, get terrible aggregated error (called cost function)**
    - **Measure local gradient of the error for each of the model parameters (input features)**
    - **Update each parameter to modify its bias up or down according to the error**
    - **Iterate until the cost function is minimized**
    - **Challenge: Local Minima! Solve by modifying the speed of the "learning rate"**





UNSUPERVISED LEARNING

- **Clustering**
  - **Example algorithm: K-Means Clustering**
    - **Goal: Separate data into predetermined “classes” (clusters)**
    - **How: Iteratively compute each class’ center point**
      - **For each point, compute distance between it and class centers, then assign class**
      - **Recompute group center by taking mean of all vectors in the group**
      - **Repeat until minimal change**
    - **Challenge: Have to select the class number ahead of time, randomized start can backfire**



DISCUSSION QUESTION 4

Can you guess some of the pros and cons of K-Means?

MACHINE LEARNING: SOFTWARE

- Different ways to program AI
- Python is possibly the best due to libraries, popularity driving training resources and constant innovation
- Python
  - SciPy Stack is a collection of math and convenience functions
    - Scipy
    - Numpy
    - Matplotlib
    - Pandas
  - Scikit-learn is a group of efficient tools for machine learning
  - Can use Anaconda to easily install the libraries

MACHINE LEARNING: REAL EXAMPLE

- Goal: Predict housing prices
- Data set: housing.csv from http://lib.stat.cmu.edu/datasets/boston
- Attributes include crime, zone sizes, non-retail business acres, pollution, size of dwelling (rooms), distance from employment centers, age of houses, student-teacher ratio of area, median value of homes in the area, etc.

MACHINE LEARNING: REAL EXAMPLE

- **Load Data:**

```
# Load libraries
import numpy
from numpy import arange
from matplotlib import pyplot
from pandas import read_csv
from pandas import set_option
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.metrics import mean_squared_error
```

```
# Load dataset
filename = 'housing.csv'
names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO',
'B', 'LSTAT', 'MEDV']
dataset = read_csv(filename, delim_whitespace=True, names=names)
```

MACHINE LEARNING: REAL EXAMPLE

- **Quick look at data: Shape, Types**
  - **506 records, 14 attributes**

```
(506, 14)
```

```
# shape
print(dataset.shape)


# types
print(dataset.dtypes)
```

```
CRIM float64
ZN float64
INDUS float64
CHAS int64
NOX float64
RM float64
AGE float64
DIS float64
RAD int64
TAX float64
PTRATIO float64
B float64
LSTAT float64
MEDV float64
```

MACHINE LEARNING: REAL EXAMPLE

- **Detailed look at data: Samples, Distribution**
  - **Wide distribution of values between variables, indicates we need to rescale**

```
# head
print(dataset.head(15))
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9
10	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15.0
11	0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.90	13.27	18.9
12	0.09378	12.5	7.87	0	0.524	5.889	39.0	5.4509	5	311	15.2	390.50	15.71	21.7
13	0.62976	0.0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21.0	396.90	8.26	20.4
14	0.63796	0.0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21.0	380.02	10.26	18.2
15	0.62739	0.0	8.14	0	0.538	5.834	56.5	4.4986	4	307	21.0	395.62	8.47	19.9

```
# descriptions
set_option('precision', 1)
print(dataset.describe())
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
count	5.1e+02	506.0	506.0	5.1e+02	506.0	506.0	506.0	506.0	506.0	506.0	506.0	506.0	506.0	506.0
mean	3.6e+00	11.4	11.1	6.9e-02	0.6	6.3	68.6	3.8	9.5	408.2	18.5	356.7	12.7	22.5
std	8.6e+00	23.3	6.9	2.5e-01	0.1	0.7	28.1	2.1	8.7	168.5	2.2	91.3	7.1	9.2
min	6.3e-03	0.0	0.5	0.0e+00	0.4	3.6	2.9	1.1	1.0	187.0	12.6	0.3	1.7	5.0
25%	8.2e-02	0.0	5.2	0.0e+00	0.4	5.9	45.0	2.1	4.0	279.0	17.4	375.4	6.9	17.0
50%	2.6e-01	0.0	9.7	0.0e+00	0.5	6.2	77.5	3.2	5.0	330.0	19.1	391.4	11.4	21.2
75%	3.7e+00	12.5	18.1	0.0e+00	0.6	6.6	94.1	5.2	24.0	666.0	20.2	396.2	17.0	25.0
max	8.9e+01	100.0	27.7	1.0e+00	0.9	8.8	100.0	12.1	24.0	711.0	22.0	396.9	38.0	50.0

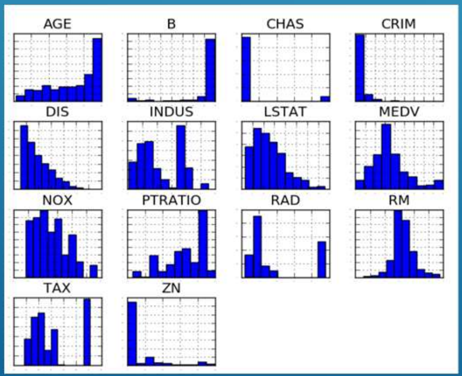
MACHINE LEARNING: REAL EXAMPLE

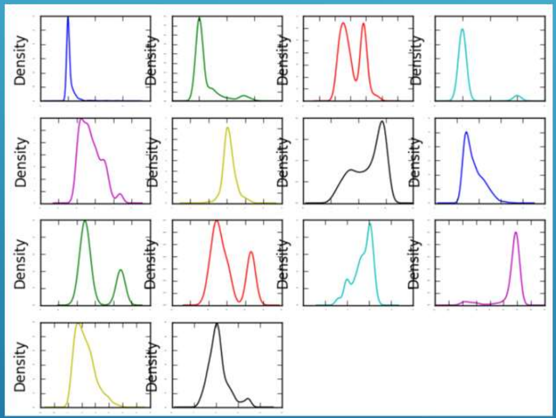
- **Other exploration of data**
  - **Looking for correlation, distribution, scaling differences**

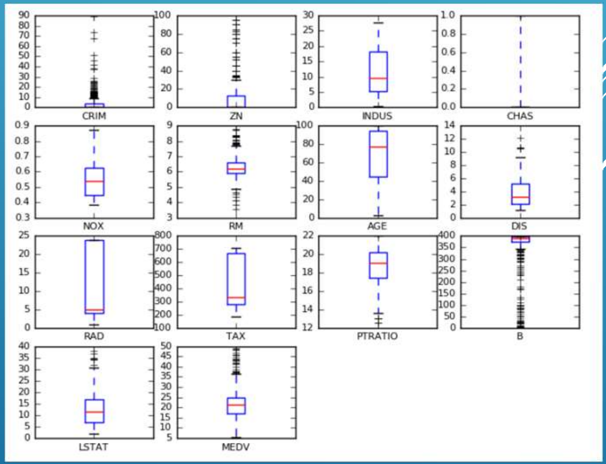
```
# histograms
dataset.hist(sharex=False, sharey=False, xlabelsize=1, ylabelsize=1)
pyplot.show()


# density
dataset.plot(kind='density', subplots=True, layout=(4,4), sharex=False, legend=False,
fontsize=1)
pyplot.show()


# box and whisker plots
dataset.plot(kind='box', subplots=True, layout=(4,4), sharex=False, sharey=False,
fontsize=8)
pyplot.show()
```







MACHINE LEARNING: REAL EXAMPLE

- **Takeaways/actions:**
  - **Try to remove high correlated features**
  - **Normalize to fix scale**
  - **Standardize to help distributions align**
- **Separate data between training and test (70/30, 80/20)**
- **Get list of algorithms that might work, test them out**

```
models = []
models.append(('LR', LinearRegression()))
models.append(('LASSO', Lasso()))
models.append(('EN', ElasticNet()))
models.append(('KNN', KNeighborsRegressor()))
models.append(('CART', DecisionTreeRegressor()))
models.append(('SVR', SVR()))


results = []
names = []
for name, model in models:
kfold = KFold(n_splits=num_folds, random_state=seed, shuffle=True)
cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
results.append(cv_results)
names.append(name)
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
print(msg)


LR: -21.379856 (9.414264)
LASSO: -26.423561 (11.651110)
EN: -27.502259 (12.305022)
KNN: -41.896488 (13.901688)
CART: -23.608957 (12.033061)
SVR: -85.518342 (31.994798)
```

MACHINE LEARNING: REAL EXAMPLE

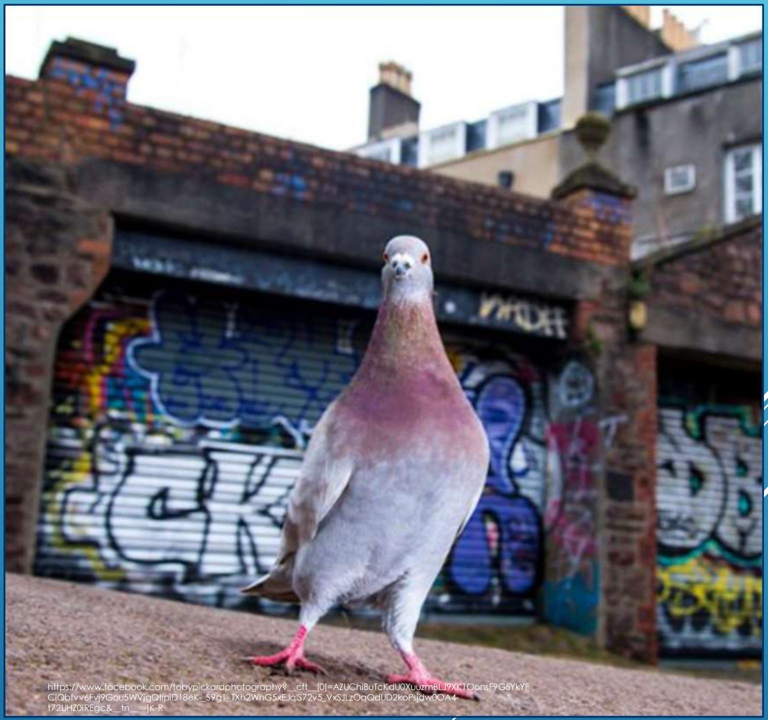
- **Advanced actions:**
  - **Standardize distributions**
  - **Tune hyperparameters for the algorithm**
  - **Consider Ensemble methods**
- **Test the model with test data**
- **Finalize and deploy (depends on purpose, data pipeline, and users)**

DISCUSSION QUESTION 5

What type of challenges do you see with real world machine learning?

STORY TIME:
OLD SCHOOL REINFORCEMENT LEARNING — FOR GANGSTER PIGEONS

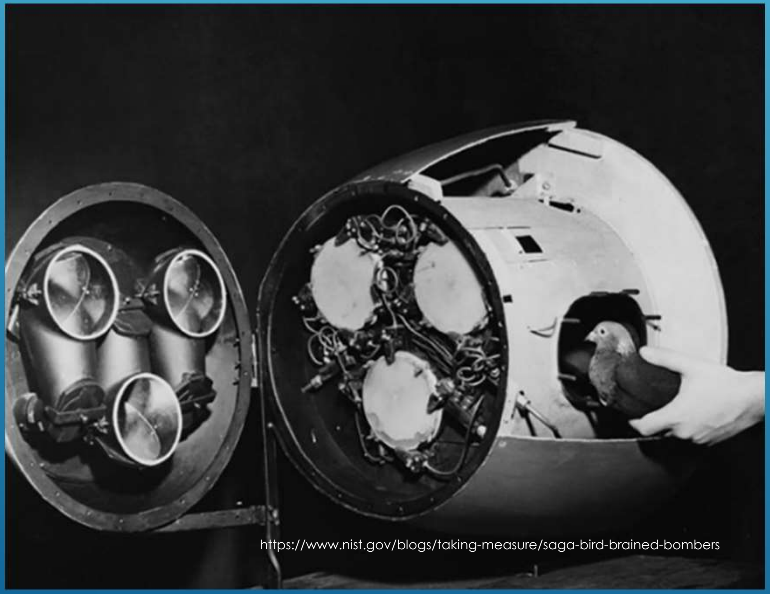
- **How were Pigeons used in WW2?**



https://www.facebook.com/tobypickardphotography?__cft__[0]=AZUChiBuTcKdU0XuuzmBLJ9XK1OonsF9G5YkYFClQbfvv6Fvj9Gou5WVjgQtipiD18eK-_S9g1_TXh2WhG5xEJqlS72v5_VxSJLzOqQdUD2koPsjdw0OA4-t72UHZ0iREgc&__tn__=-]K-R

STORY TIME

“Suddenly, I saw them as ‘devices’ with excellent vision and extraordinary maneuverability. Could they not guide a missile?”



Tried remote operator

Tried cats (briefly)

Could pigeons be trained with RL?

STORY TIME

Conductive screen with projection

Trained with combat videos

How to deal with the noise/vibrations?

"We had begun to realize that a pigeon was more easily controlled than a physical scientist serving on a committee."

ACTIVITY—WE ARE HARDWIRED FOR REINFORCEMENT LEARNING (RL)

ACTIVITY

How easy can we be trained using core RL principles?

DISCUSSION QUESTION

What do you need in order to teach someone a new task?

I need five brave volunteers…

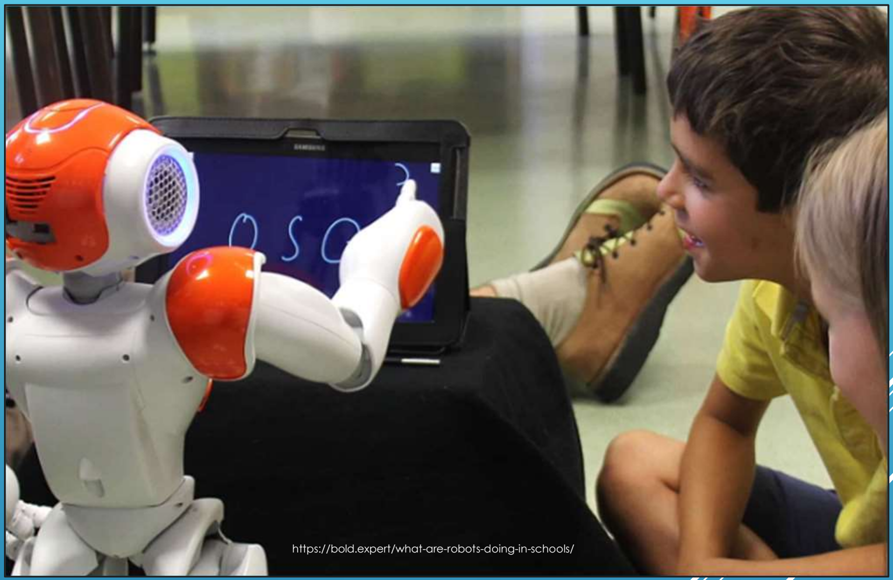
DISCUSSION QUESTION

WHY DID THIS WORK?

OVERVIEW AND BASICS

WHAT IS REINFORCEMENT LEARNING?

- Feedback-based ML
- Agent learns to interact with an environment, see the results of their actions, and develop a policy to maximize their reward
- No labelled data, uses experience and rewards only
- Better used for tasks where decisions are sequential and there is a longer term goal

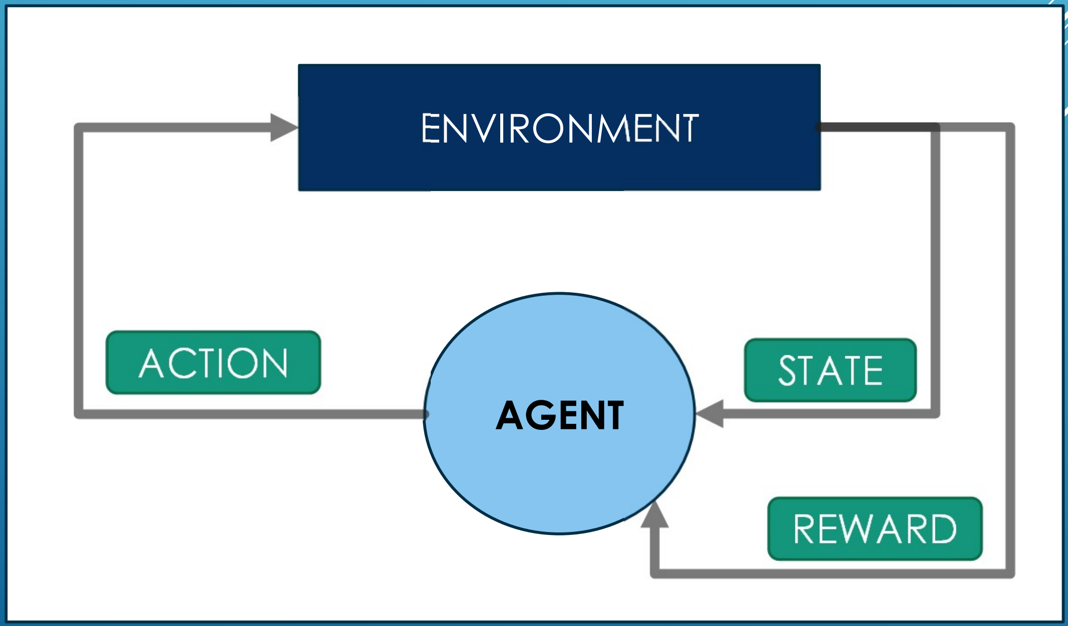


WHAT IS REINFORCEMENT LEARNING?

- **Quick clarifiers**
  - **For the purpose of our lecture, “reinforcement learning” and “agent based training” are the same.**
  - **Reinforcement learning as a concept does not have to include deep learning (see previous exercise), but it can and that is what we will focus on.**

WHAT IS REINFORCEMENT LEARNING?

- The agent isn't told about the environment, the goals, or their actions
- They learn through experience
  - What works and what doesn't
  - Incorrect perceptions
  - Rewards and punishments
  - Losing sight of the larger goal
  - Getting stuck in bad habits



RL COMPONENTS

Important

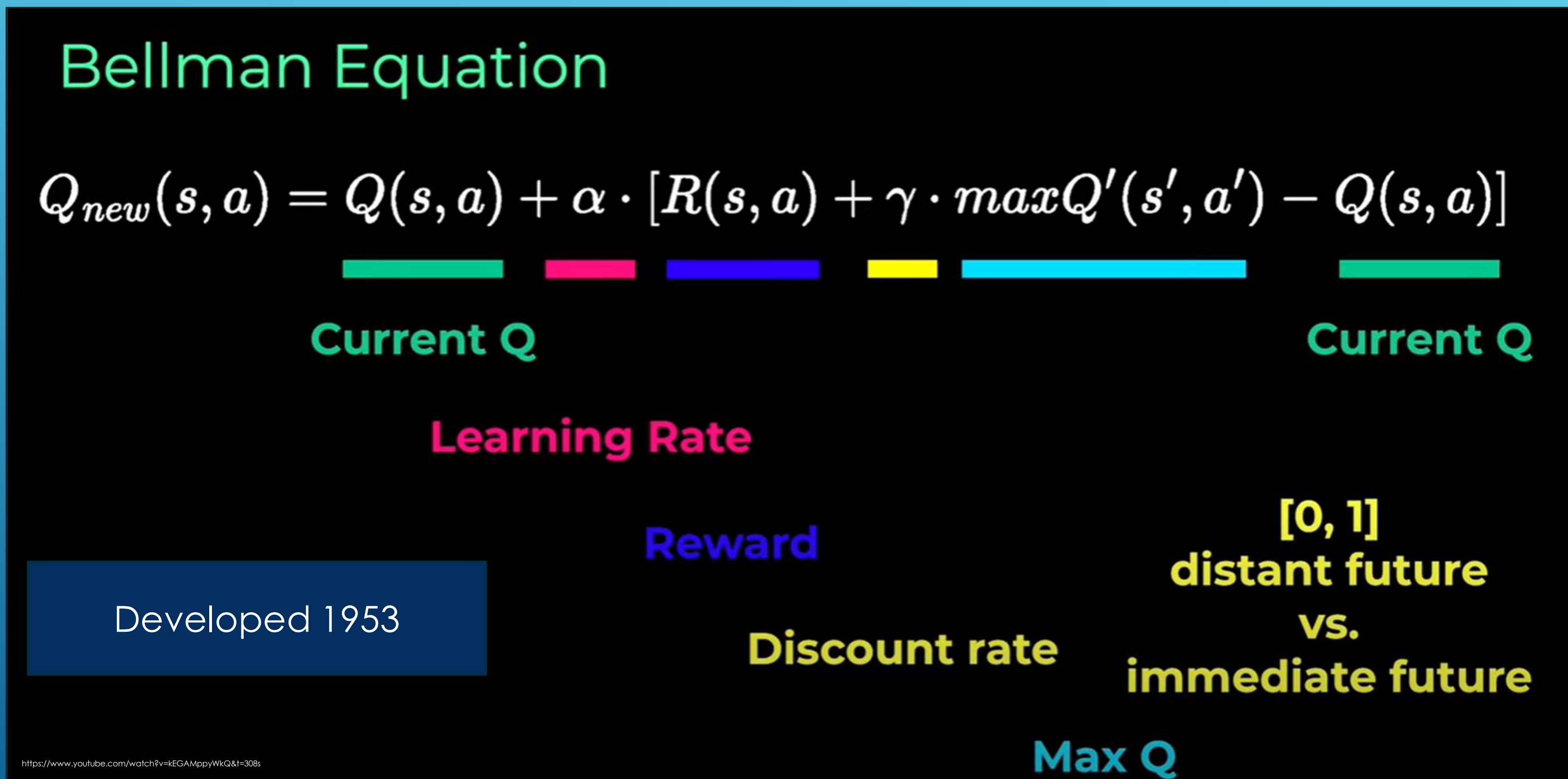
MAIN COMPONENTS

- **Environmental Model:** Where the agent is and confined
- **Reward:** Feedback returned after each action/state (+/-)
- **Policy:** Strategy developed by the agent that guides next step based on current step
- **Value Function:**
  - **Value:** Long-term reward (expected)
  - **Q-Value:** Long-term reward based on current action

SUPPORTING COMPONENTS

- **State:** Situation of the environment and agent after a move is made
- **Action**: What the agent can do for each given moment
- **Agent:** Entity that interacts with the environment through action

REPRESENTATIVE EQUATION



Developed 1953

https://www.youtube.com/watch?v=kEGAMppyWkQ&t=308s

KEY TECHNIQUES

- **Markov Decision Process**
  - **Uses math to map out the optimum solution, updating actions, states, in order to develop a policy**
- **Q Learning (ML)**
  - **Value-based method, iteratively updates weights/biases in order to match actions to expected values**
- **Other Related Techniques or Variations (Always growing)**

DISCUSSION QUESTION

What are some of the key challenges for Reinforcement Learning?

CHALLENGES

- **Reinforcement Learning is expensive! Cost is exponential with complexity; If you have the data to solve with supervised learning, do it**
- **Reward design is a critical component and requires equal parts foresight and experimentation**
- **Overfitting is possible and might not show up until far into training**
- **When using RL for real applications, it can be hard to match simulation and reality**

EXAMPLES

Q*BERT: UNINTENDED CONSEQUENCES

DISCUSSION QUESTION

Goal:

How Environment is Perceived:

Actions:

Rewards:

Unintended Consequences:

Q*BERT: UNINTENDED CONSEQUENCES

- **Goal:** Max points
- **How Environment is Perceived:** Knows position, can see colors and enemies
- **Actions:** Jump to adjacent squares
- **Rewards:** Points (and this is key), punished if killed
- **Unintended Consequences:**

- "First, it completes the first level and then starts to jump from platform to platform in what seems to be a random manner. For a reason unknown to us, the game does not advance to the second round but the platforms start to blink and the agent quickly gains a huge amount of points (close to 1 million for our episode time limit)."

https://arxiv.org/abs/1802.08842

DISCUSSION QUESTION

Why did Q*Bert behave this way?

How would you adjust the model to improve the agent?

TRACKMANIA: ROBUST MODEL

https://www.youtube.com/watch?v=SX08NT55YhA

DISCUSSION QUESTION

Goal:

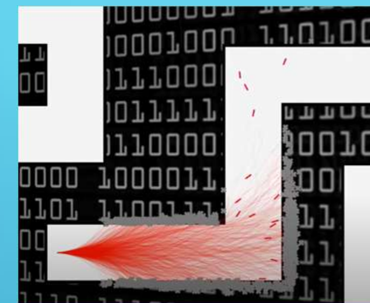
How Environment is Perceived:

Actions:

Rewards:

Unintended Consequences:

TRACKMANIA: ROBUST MODEL



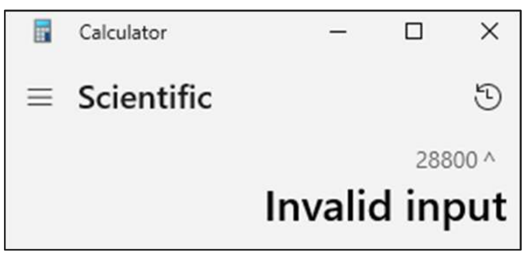
- **Goal:** Finish race, quickly
- **How Environment is Perceived:** Raycasts, position of own body
- **Actions:** Accelerate, steer
- **Rewards:** Distance travelled, time elapsed, falling
- **Unintended Consequences:** Late realization of overfitting, solved through random re-gen

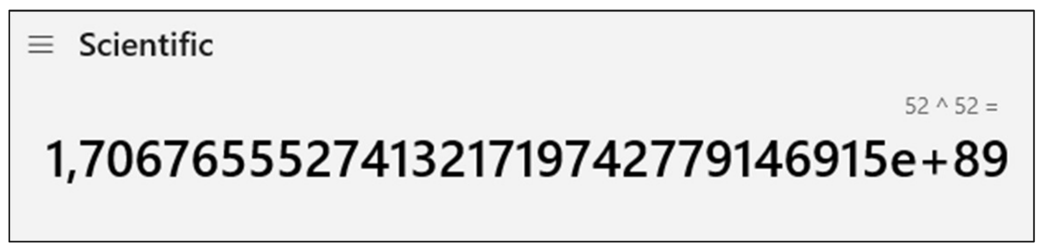
DISCUSSION QUESTION

Is this type of training the same as fully mapping all possibilities?

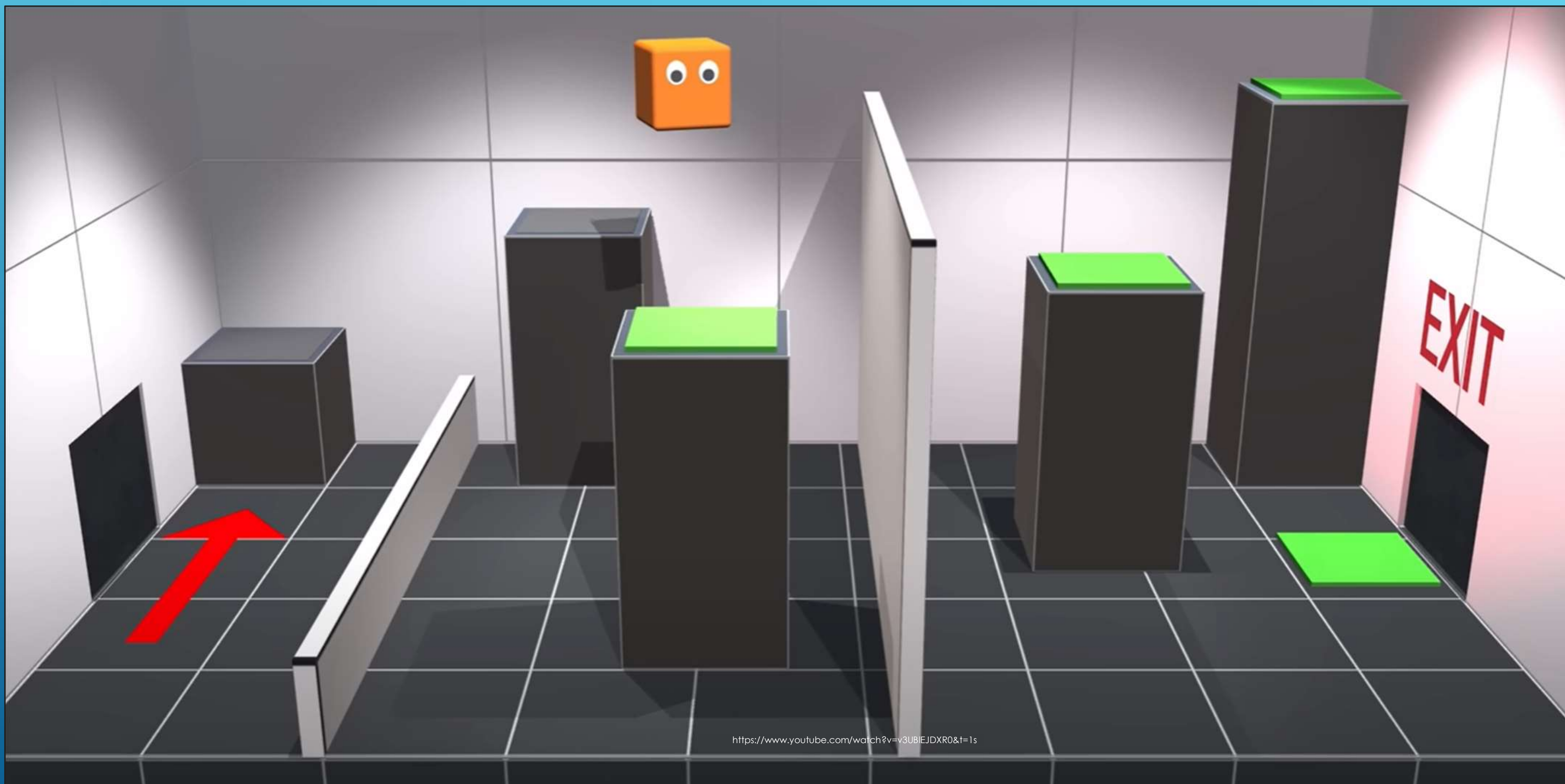
DISCUSSION QUESTION

Is this type of training the same as fully mapping all possibilities?





ALBERT ESCAPES: LONG SEQUENCE LEARNING



DISCUSSION QUESTION

Goal:

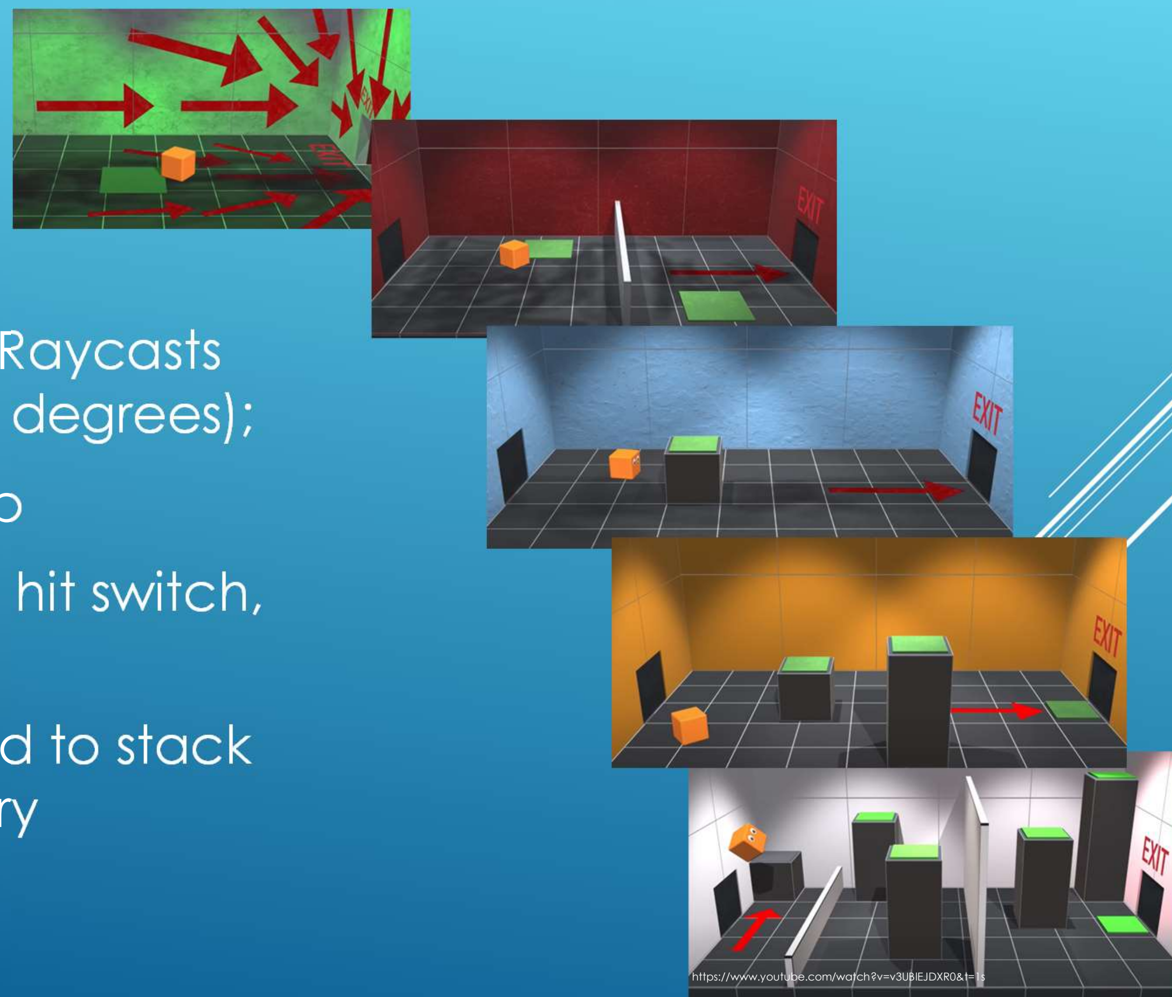
How Environment is Perceived:

Actions:

Rewards:

Unintended Consequences:

ALBERT ESCAPES: LONG SEQUENCE LEARNING

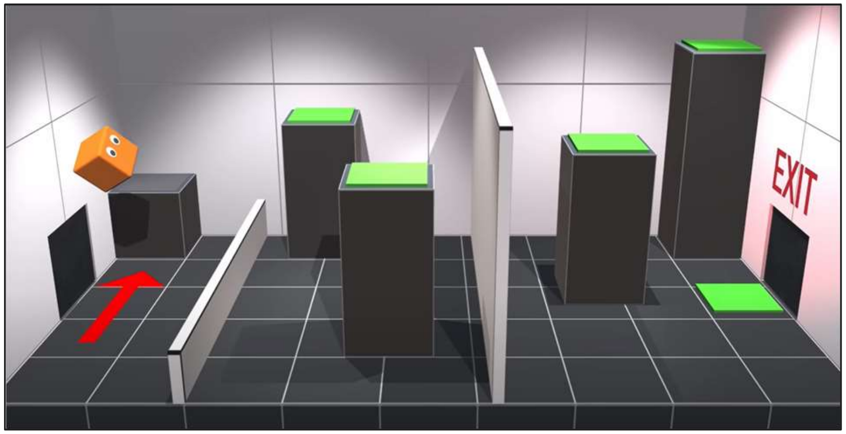


- **Goal:** Escape!
- **How Environment is Perceived:** Raycasts down, ahead, above head (70 degrees);
- **Actions:** Move horizontally, jump
- **Rewards:** Time (neg), Fall (neg), hit switch, escape room
- **Unintended Consequences:** Had to stack vision to build short term memory

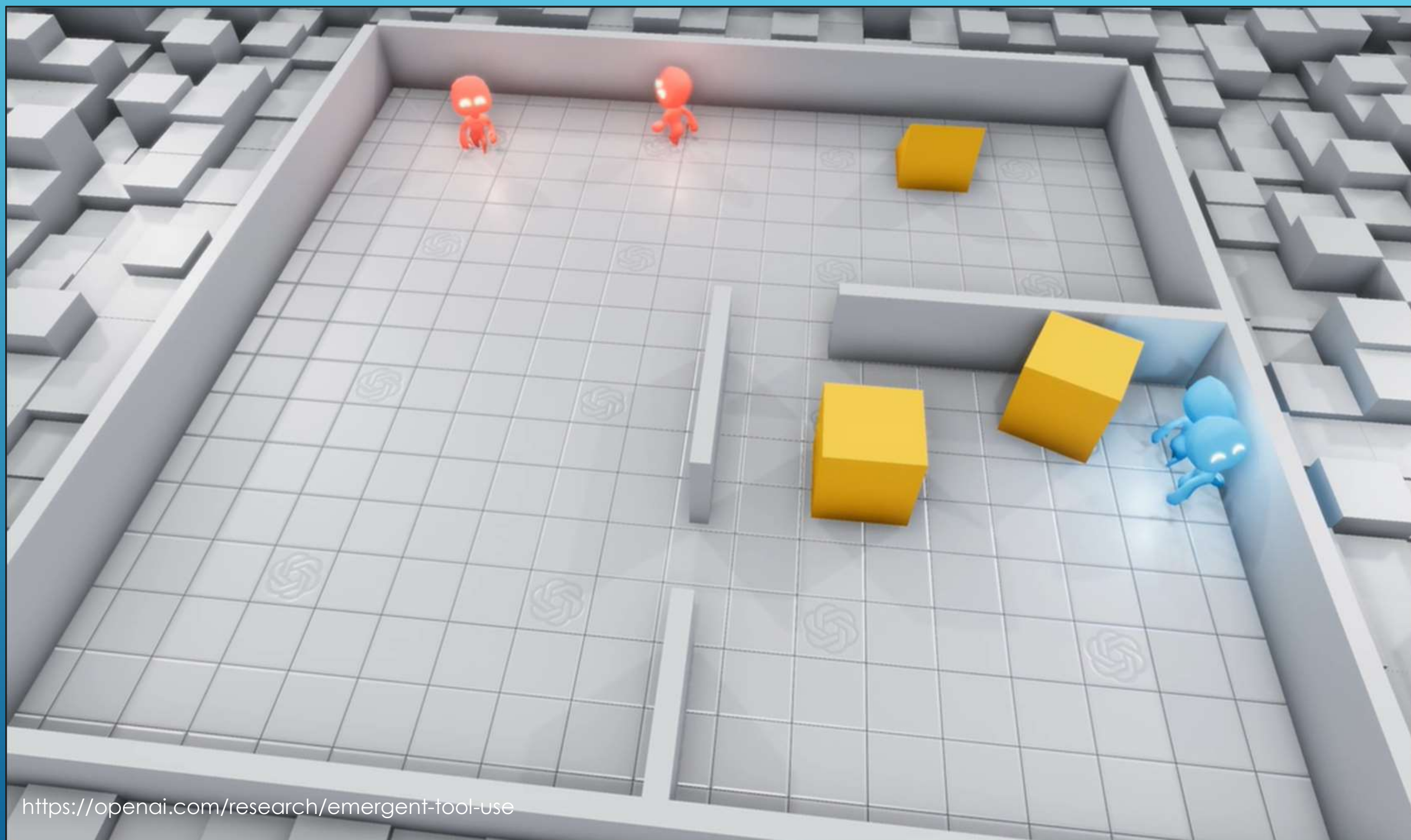
DISCUSSION QUESTION

How might have Albert gotten stuck in Room 5?

What could we adjust to fix it?



HIDE AND SEEK: DISTURBINGLY CREATIVE



DISCUSSION QUESTION

Goal:

How Environment is Perceived:

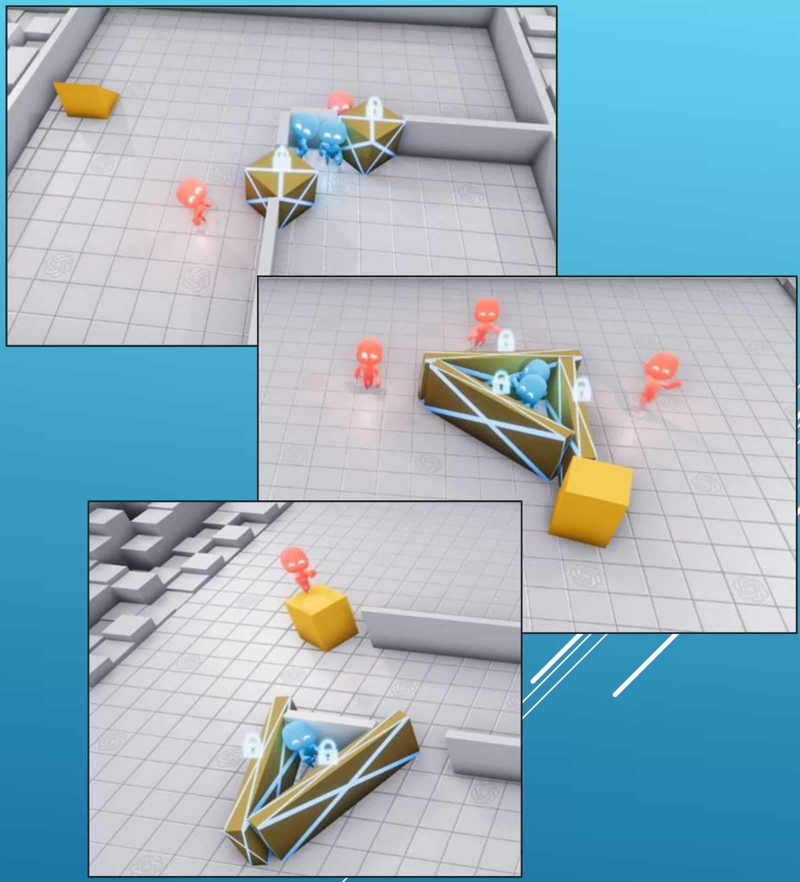
Actions:

Rewards:

Unintended Consequences:

HIDE AND SEEK: DISTURBINGLY CREATIVE

- **Goal:** Hide (or seek)
- **How Environment is Perceived:** Raycasts to see and identify objects
- **Actions:** Run, grab objects, lock objects, move objects
- **Rewards:** Points won/lost as time goes on, reward/punishment for result of game
- **Unintended Consequences:** When multi-agent problems are allowed to evolve, very creative behaviour can emerge.

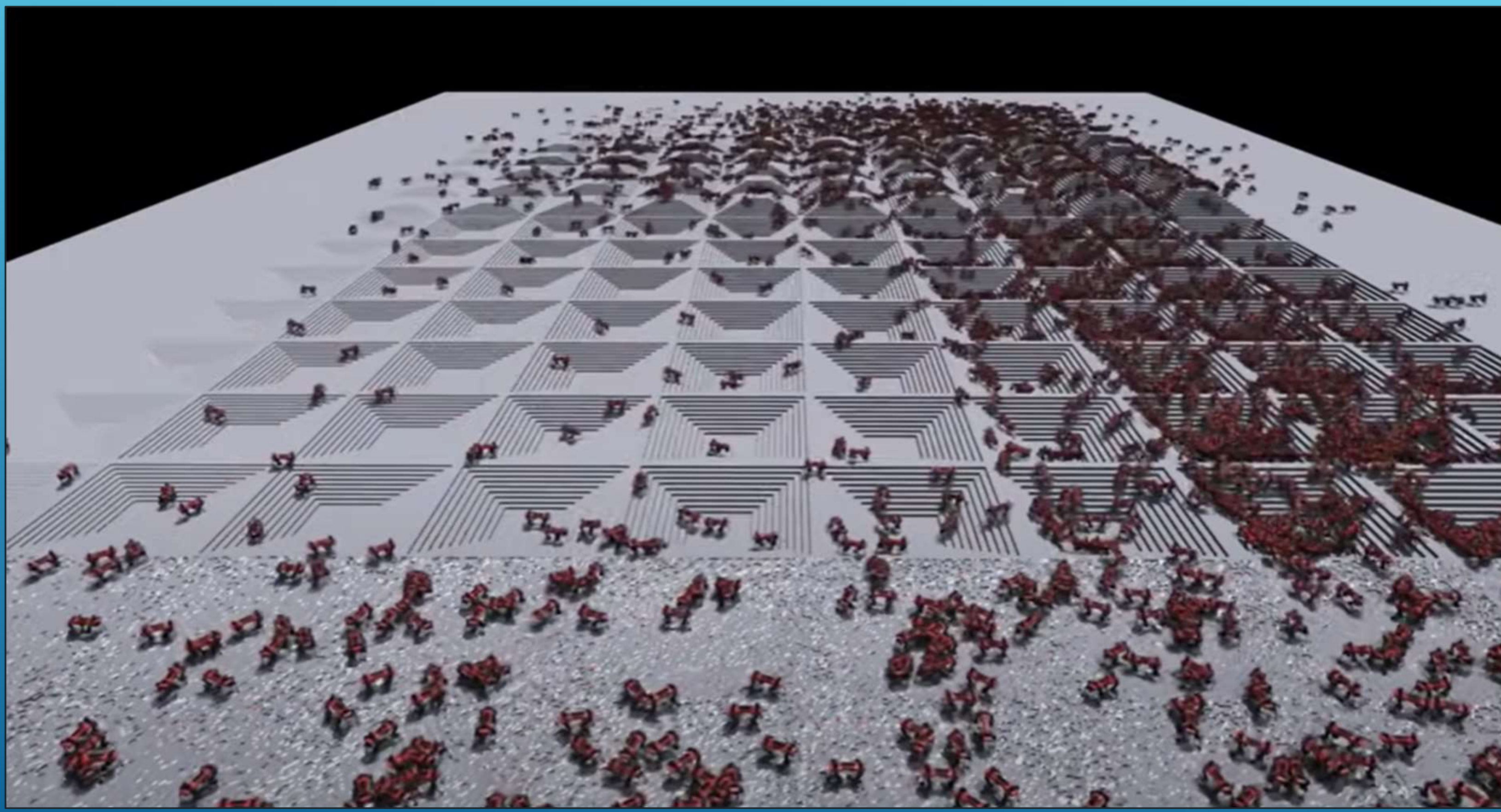


https://openai.com/research/emergent-tool-use

DISCUSSION QUESTION

How could you use these unintended consequences to your advantage when implementing a new application?

BRINGING RL TO THE REAL WORLD



https://leggedrobotics.github.io/legged_gym/

DISCUSSION QUESTION

Goal:

How Environment is Perceived:

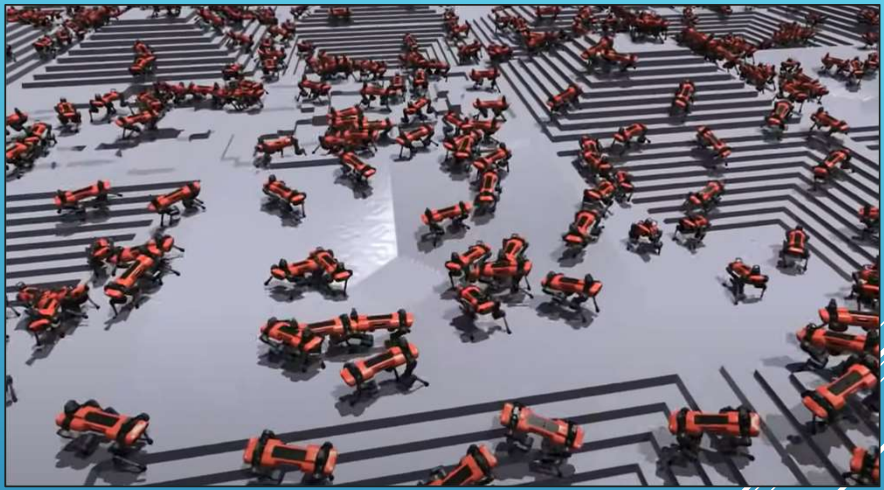
Actions:

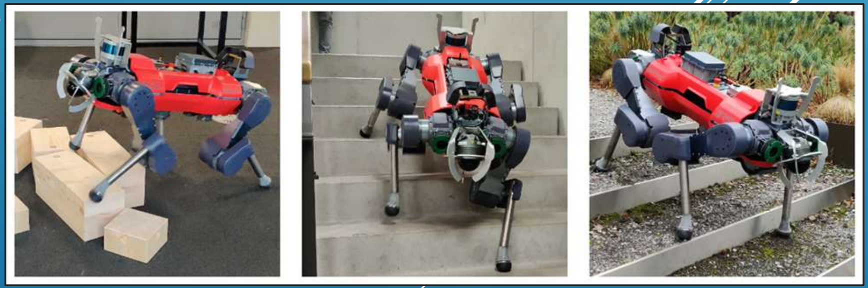
Rewards:

Unintended Consequences:

BRINGING RL TO THE REAL WORLD

- **Goal:** Learn to walk
- **How Environment is Perceived:** Various positions, surrounding environment
- **Actions:** Articulate limbs
- **Rewards:** Distance travelled
- **Unintended Consequences:** Challenges when converting from sim to real world with added noise





https://leggedrobotics.github.io/legged_gym/

TEAM CHALLENGE:

CAN RL SOLVE THE CANDLE PROBLEM?

SETUP: THE PROBLEM

Solve the problem of a candle dripping wax onto a table using only the items seen here.

SETUP: THE PROBLEM

Alternate set up; easier?

SETUP: THE "SOLUTION"

CHALLENGE:

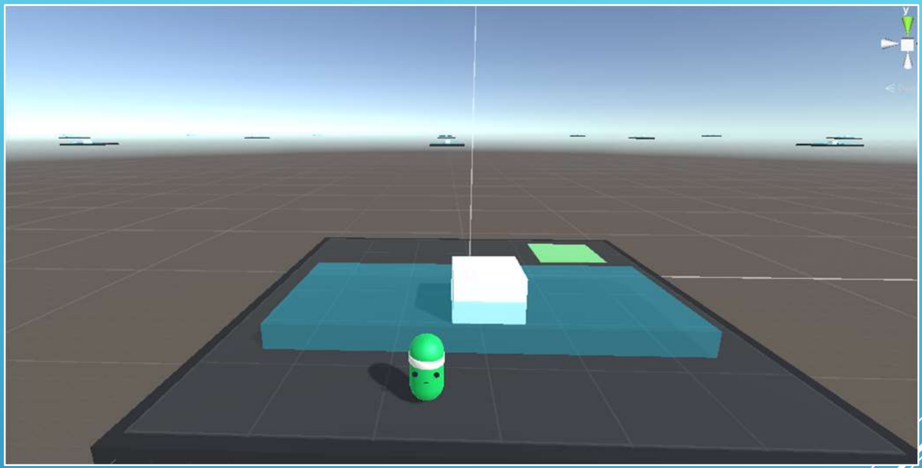
Create a reinforcement learning model where an agent can learn to solve the candle problem

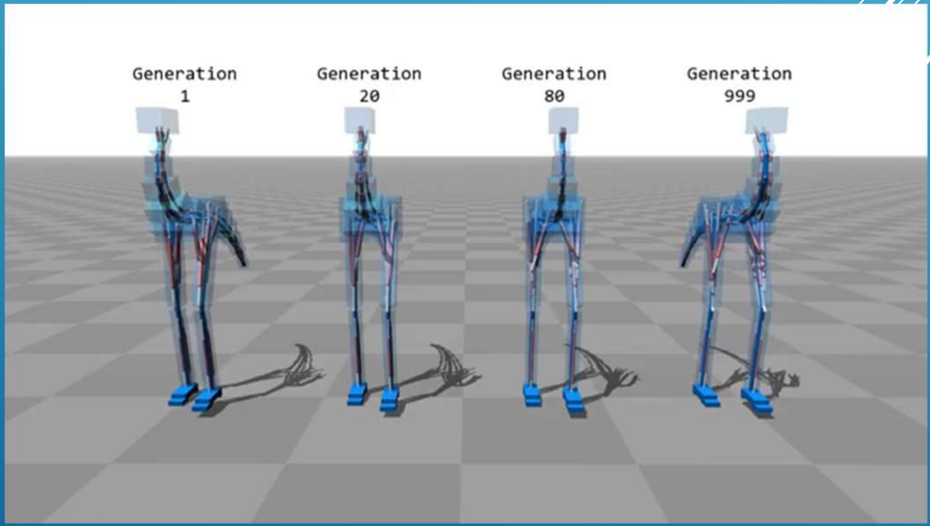
- **Goal:**
- **How Environment is Perceived:**
- **Actions:**
- **Rewards:**
- **Unintended Consequences:**

AEROSPACE APPLICATIONS

PROJECT 1: ROBOTIC OPTIMIZATION OF A SET TASK

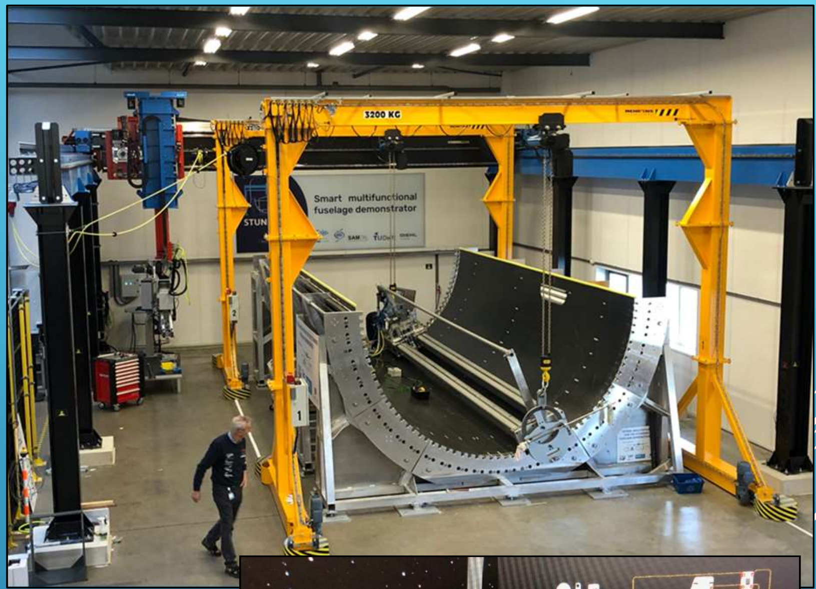
- Problem:
  robot programming is not easily scalable
- Solution: reinforcement learning to optimize the correct movements to complete a set task
- Exponential benefit: any time the process is changed, the simulation can be changed and retrained with zero code





PROJECT 2: ROBOTIC OPTIMIZATION OF A FLUID TASK

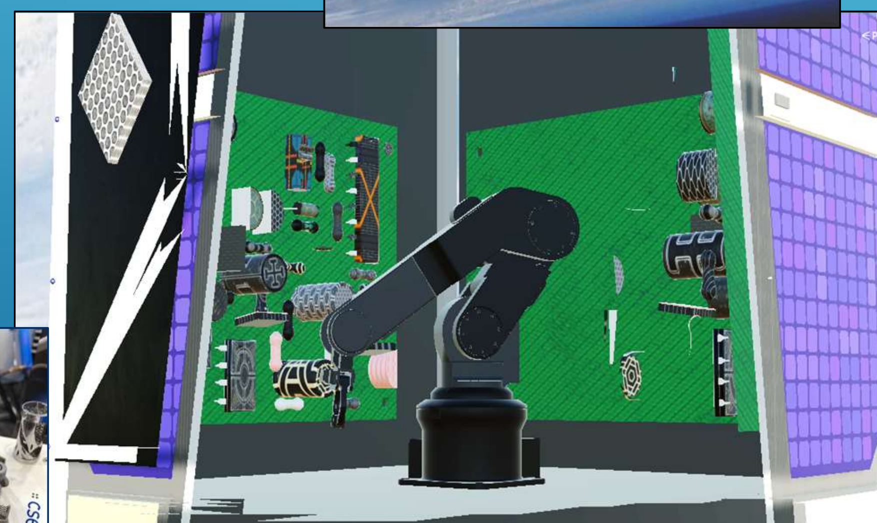
- Problem:
  traditional robot programming cannot handle fluid (uncertain) tasks
- Solution: reinforcement learning to simulate/optimize highly variable process; deploy on a physical robot
- Exponential benefit: robot can handle uncertainties with zero explicit code



PROJECT 3: UTILIZE AI-BASED TRAINING FOR SPACE MANUFACTURING

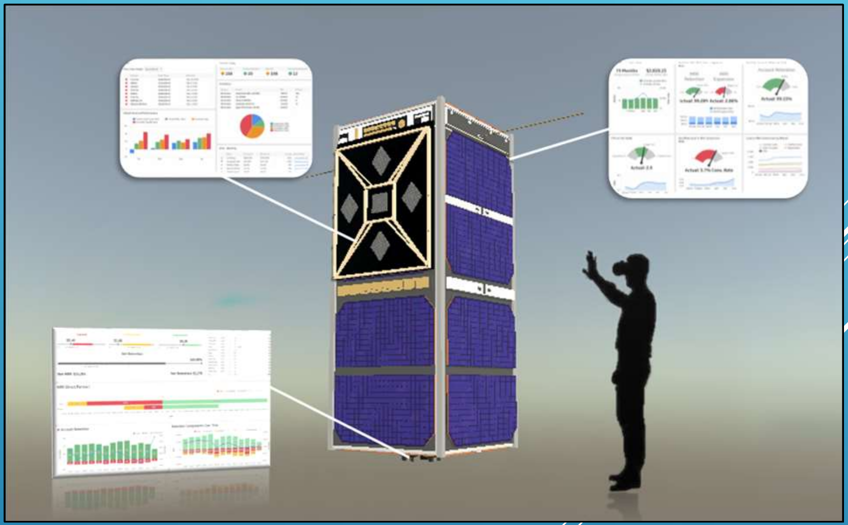
- Problem:
  robots not trusted to perform remote tasks independently
- Solution: pre-train robot in simulated microgravity for component replacement
- Exponential benefit: validates/improves micro G simulations, demonstrates spacecraft life extension / autonomous space manufacturing

Project TESSERACT



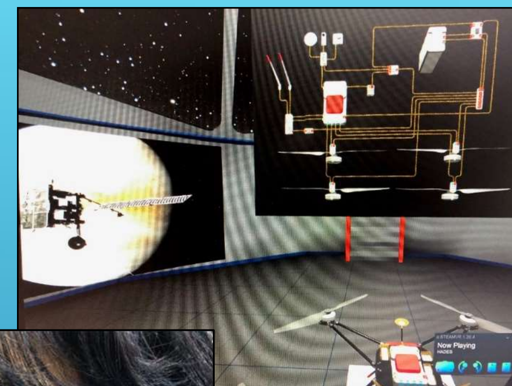
PROJECT 4: IMMERSIVE, INTERACTIVE DIGITAL TWIN

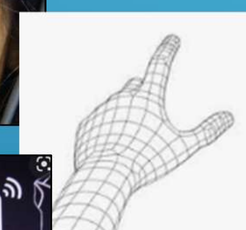
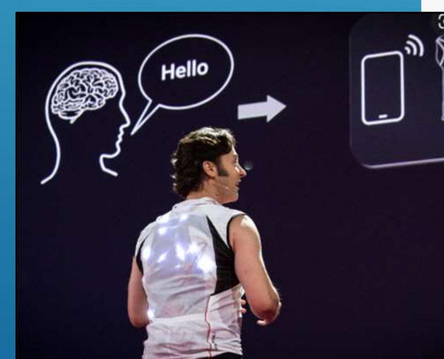
- Problem:
  digital twins offer key metrics but often little else. Key measure of success should be "time to action"
- Solution: develop immersive digital twin with many metrics, ai-driven analytics / recommendations (pm), vr interaction
- Exponential benefit:
  innovates key time to action,
  real-time interaction with digital twin
  (reaction wheel manipulation with project tesseract)



PROJECT 5: UTILIZE AI-BASED TRAINING FOR HUMANS

- Problem:
  training humans to complete a complex task with limited repetitions is inefficient
- Solution: optimize a task for a virtual human, then “train” a human in real time using AR and haptic guidance.
- Exponential benefit: training humans to perform complex tasks in real time has monumental benefits across all industries.



DISCUSSION QUESTION

What other RL applications do you think are possible in the aerospace industry?

CONCLUSION

- Reinforcement Learning is a powerful tool that can be used to teach long, complex tasks
- RL relies on an agent, environment, actions, rewards, and the ability to create a policy
- RL can have unintended consequences, be computationally very expensive, and can hit dead ends
- RL will change the aerospace industry

QUESTIONS?

Nathan Eskue
n.d.eskue@tudelft.nl

RESEARCH SOURCES

https://www.frontiersin.org/articles/10.3389/fpsyg.2018.02001/full

http://dma.ing.uniroma1.it/users/ls_sas/MATERIALE/Design%20criteria%20and%20procedures%20of%20space%20structures.pdf

https://www.ted.com/talks/julie_chang_why_we_need_to_design_cameras_for_robots

https://acerta.ai/blog/artificial-intelligence-industry-4-0-5-manufacturing-applications-for-ai/

https://7t.co/blog/the-top-3-artificial-intelligence-applications-in-manufacturing/

https://medium.com/@venkat34.k/10-examples-of-artificial-intelligence-in-manufacturing-to-inspire-your-smart-factory-3c780be46624

https://emerj.com/ai-sector-overviews/machine-learning-in-manufacturing/

https://www.forbes.com/sites/louiscolumbus/2020/05/18/10-ways-ai-is-improving-manufacturing-in-2020/?sh=7bbffe391e85

2019 Manufacturing Trends Report, Microsoft (PDF, 72 pp., no opt-in)

Accenture, Manufacturing The Future, Artificial intelligence will fuel the next wave of growth for industrial equipment companies (PDF, 20 pp., no opt-in)

Anderson, M. (2019). Machine learning in manufacturing. Automotive Design & Production, 131(4), 30-32.

Bruno, J. (2019). How the IIoT can change business models. Manufacturing Engineering, 163(1), 12.

Enabling a digital and analytics transformation in heavy-industry manufacturing, McKinsey & Company, December 19, 2019

Governance and Management Economics, 7(2), 31-36.

Greenfield, D. (2019). Advice on scaling IIoT projects. ProFood World

Hayhoe, T., Podhorska, I., Siekelova, A., & Stehel, V. (2019). Sustainable manufacturing in industry 4.0: Cross-sector networks of multiple supply chains, cyber-physical production systems, and AI-driven decision-making. Journal of Self-

Industry's fast-mover advantage: Enterprise value from digital factories, McKinsey & Company, January 10, 2020

Kazuyuki, M. (2019). Digitalization of manufacturing process and open innovation: Survey results of small and medium-sized firms in japan. St. Louis: Federal Reserve Bank of St Louis.

Machine Learning in Manufacturing – Present and Future Use-Cases, Emerj Artificial Intelligence Research, last updated May 20, 2019, published by Jon Walker

Machine learning, AI are most impactful supply chain technologies. (2019). Material Handling & Logistics

'Lighthouse' manufacturers lead the way—can the rest of the world keep up? McKinsey & Company, January 7, 2019

MAPI Foundation, The Manufacturing Evolution: How AI Will Transform Manufacturing & the Workforce of the Future by Robert D. Atkinson, Stephen Ezell, Information Technology and Innovation Foundation (PDF, 56 pp., opt-in)

RESEARCH SOURCES

Mapping heavy industry's digital-manufacturing opportunities, McKinsey & Company, September 24, 2018

McKinsey, 'Lighthouse' manufacturers, lead the way—can the rest of the world keep up?,by Enno de Boer, Helena Leurent, and Adrian Widmer; January, 2019.

McKinsey, AI in production: A game changer for manufacturers with heavy assets, by Eleftherios Charalambous, Robert Feldmann, Gérard Richter, and Christoph Schmitz

McKinsey, Digital Manufacturing – escaping pilot purgatory (PDF, 24 pp., no opt-in)

McKinsey, Driving Impact and Scale from Automation and AI, February 2019 (PDF, 100 pp., no opt-in).

McKinsey, Manufacturing: Analytics unleashes productivity and profitability, by Valerio Dilda, Lapo Mori, Olivier Noterdaeme, and Christoph Schmitz, March, 2019

McKinsey/Harvard Business Review, Most of AI's business uses will be in two areas,

Morey, B. (2019). Manufacturing and AI: Promises and pitfalls. Manufacturing Engineering, 163(1), 10.

Preparing for the next normal via digital manufacturing's scaling potential, McKinsey & Company, April 10, 2020

Reducing the barriers to entry in advanced analytics. (2019). Manufacturing.Net,

Scaling AI in Manufacturing Operations: A Practitioners Perspective, Capgemini, January, 2020

Seven ways real-time monitoring is driving smart manufacturing. (2019). Manufacturing.Net,

Siemens, Next Level AI – Powered by Knowledge Graphs and Data Thinking, Siemens China Innovation Day, Michael May, Chengdu, May 15, 2019

Smart Factories: Issues of Information Governance Manufacturing Policy Initiative School of Public and Environmental Affairs Indiana University, March 2019 (PDF, 68 pp., no opt-in)

Smartening up with Artificial Intelligence (AI) - What's in it for Germany and its Industrial Sector? (52 pp., PDF, no opt-in) McKinsey & Company.

Team predicts the useful life of batteries with data and AI. (2019, March 28). R & D.

The Future of AI and Manufacturing, Microsoft, Greg Shaw (PDF, 73 pp., PDF, no opt-in).

The Rise of the AI-Powered Company in the Postcrisis World, Boston Consulting Group, April 2, 2020

Top 8 Data Science Use Cases in Manufacturing, ActiveWizards: A Machine Learning Company Igor Bobriakov, March 12, 2019

Walker, M. E. (2019). Armed with analytics: Manufacturing as a martial art. Industry Week

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. Journal of Manufacturing Systems, 48, 144–156.

Zulick, J. (2019). How machine learning is transforming industrial production. Machine Design

RESEARCH SOURCES

"A Genetic Algorithm for the Optimization of Cable Routing". Xuan Ma, Kazuhiro Lida, Mengchun Xie, Junji Nishino, Tomohiro Odaka, and Hisakazu Ogura. Systems and Computers in Japan, Vol. 37, No. 7, 2006

Earley, M. W., Murray, R. H., Caloggero, J. M., & O'Connor, J. A. (1993). National electric code handbook 1993. Quincy, MA: National Fire Protection Association.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison{Wesley.

Goldberg, D. E., & Deb, K. (1990). A comparative analysis of selection schemes used in genetic algorithms (TCGA Report No. 90007). Tuscaloosa: The University of Alabama

Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor: The University of Michigan Press.

Intergraph (1993). Electrical engineering cable management system (EECms). Marketing Document DDAEC94A0.

The Coding Train Youtube Channel, Genetic Algorithm Series

Ruhlen, L. H., & Shire, P. R. (1967). Optimal circuit routing by dynamic programming. In Proceedings of the Power Industrial Computer Applications Conference (PICA), 291{301.

Smith, R. E., Goldberg, D. E., & Earickson, J. (1991). SGA-C: A C-language implementation of a simple genetic algorithm (TCGA Report No. 91002). Tuscaloosa: University of Alabama, The Clearinghouse for Genetic Algorithms.

http://boxcar2d.com/

9.1 – 9.7: Genetic Algorithm Series. The Coding Train Channel, Youtube.

Sinha P, Zoltners AA. The multiple-choice knapsack problem. Oper Res 1979;27:503–614.

Armatrong RD, Kung DS, Sinha P, Zoltners AA. A computational study of a multiple-choice knapsack algorithm. ACM Trans Math Software 1983;9:184–198.

Leung Y, Li G, Xu ZB. A genetic algorithm for the multiple destination routing problems. IEEE Trans Evol Comput 1998;2:150–161.

https://github.com/gmichaelson/GA_in_python

"BULK CABLE ROUTING USING GENETIC ALGORITHMS". David A. Kloske, Robert E. Smith. TCGA Report No. 94001; March 22, 1994. The University of Alabama

https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35