Neural Nets and Flying Saucers

An Introduction to Adversarial Machine Learning

Jan A. Núñez

Agenda

- Supervised Learning
- Unsupervised Learning
- ML Algorithms

- Data Aquisition
- Data Preprocessing
- Model Building
- Model Training
- Fine Tuning
- Making Predictions

- OWASP Top Ten
- MITRE ATLAS
- OffSecML Playbook

- Demo Attacks
- Adversarial inputs
- Malicioius Models
- LLM Prompt Injection

- Mitigating Controls
- Learning Material
- Questions

AI Everywhere

01.

Computing Power

02.

Datasets

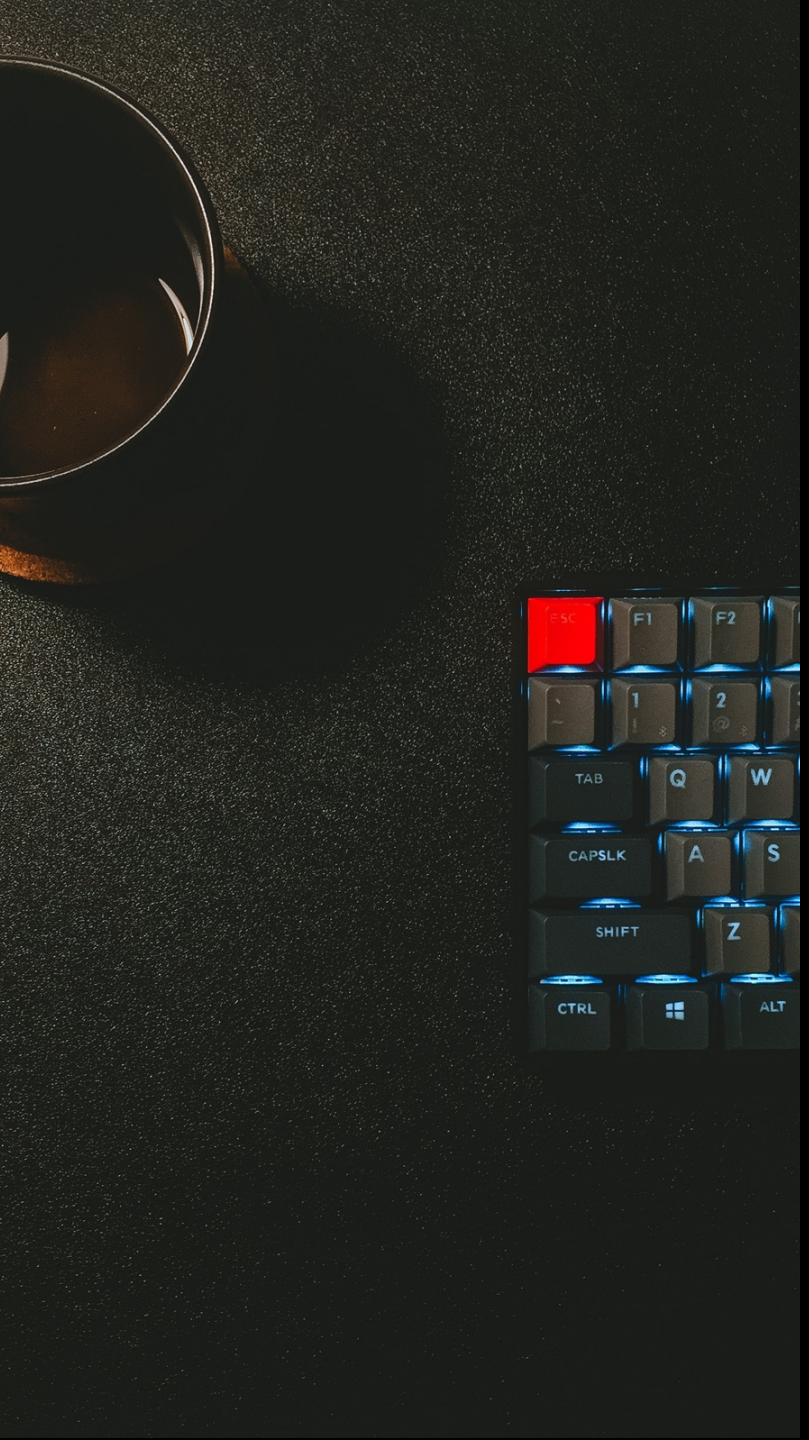
03.

Open source tools

04.
OpenAI

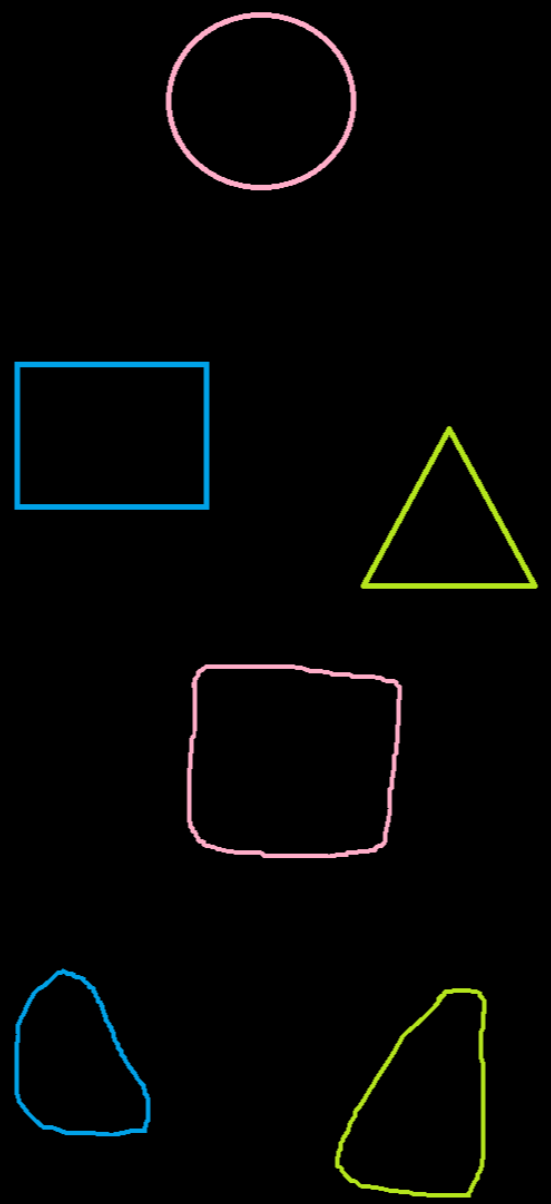
Hey dev, build me an app.

- I give you speed limit
- I give you distance traveled
- You tell me if they were speeding?



The code

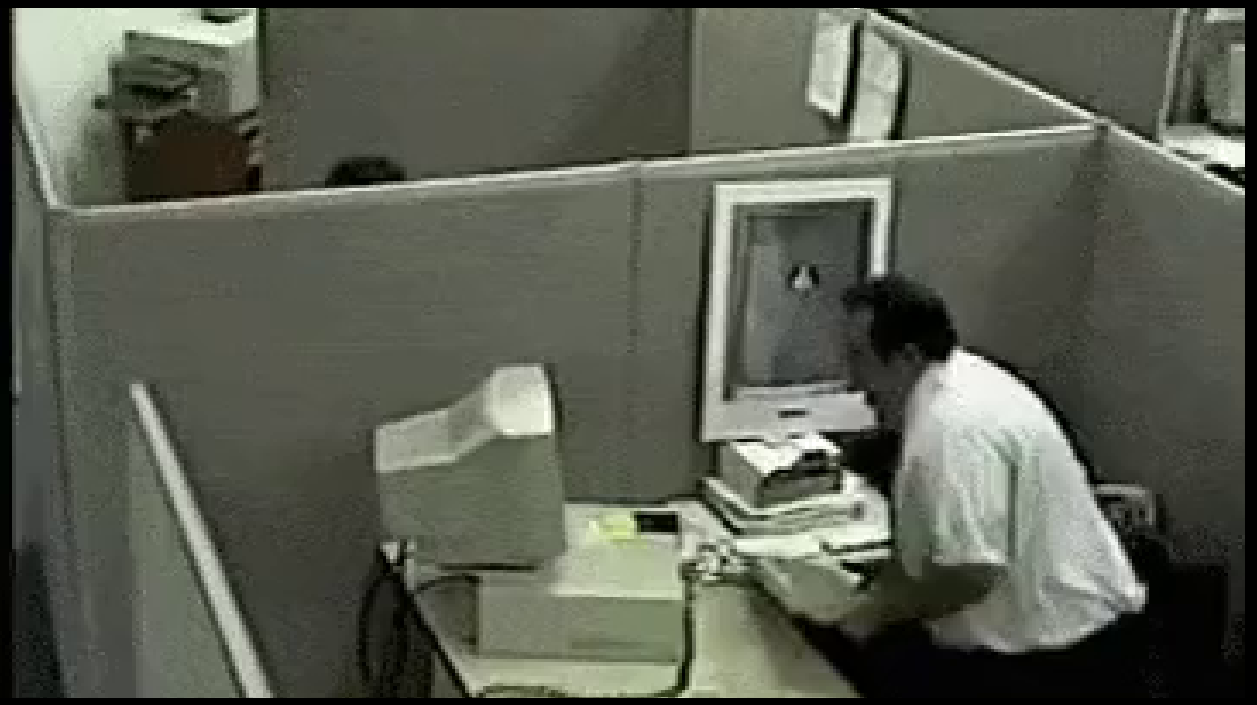
```python
def check_speeding(distance_miles, time_hours, speed_limit):
    # Calculate the average speed
    average_speed = distance_miles / time_hours

    # Check if the average speed is greater than the speed limit
    if average_speed > speed_limit:
        return True
    else:
        return False

# Example usage of the function
if check_speeding(100, 1, 90):
    print('Take this ticket. Happy holidays.')
```



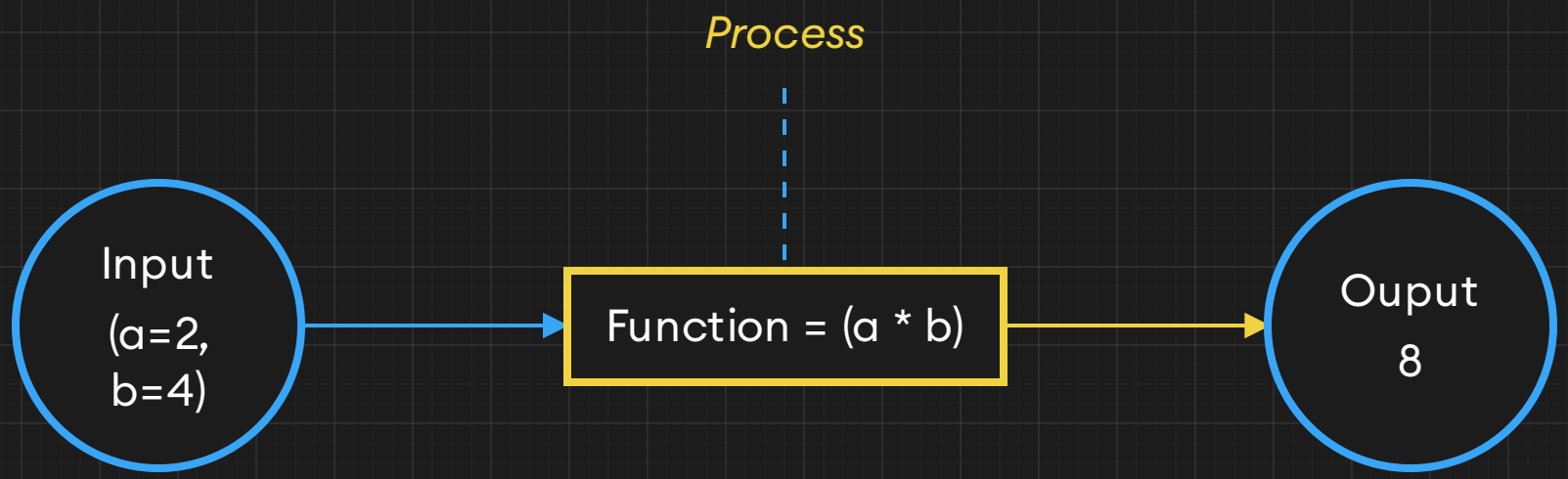
Great, so you can do this too, right?

- I give you an image
- You tell me if it's a square, circle, or triangle
- Oh, and the shapes might be hand drawn!

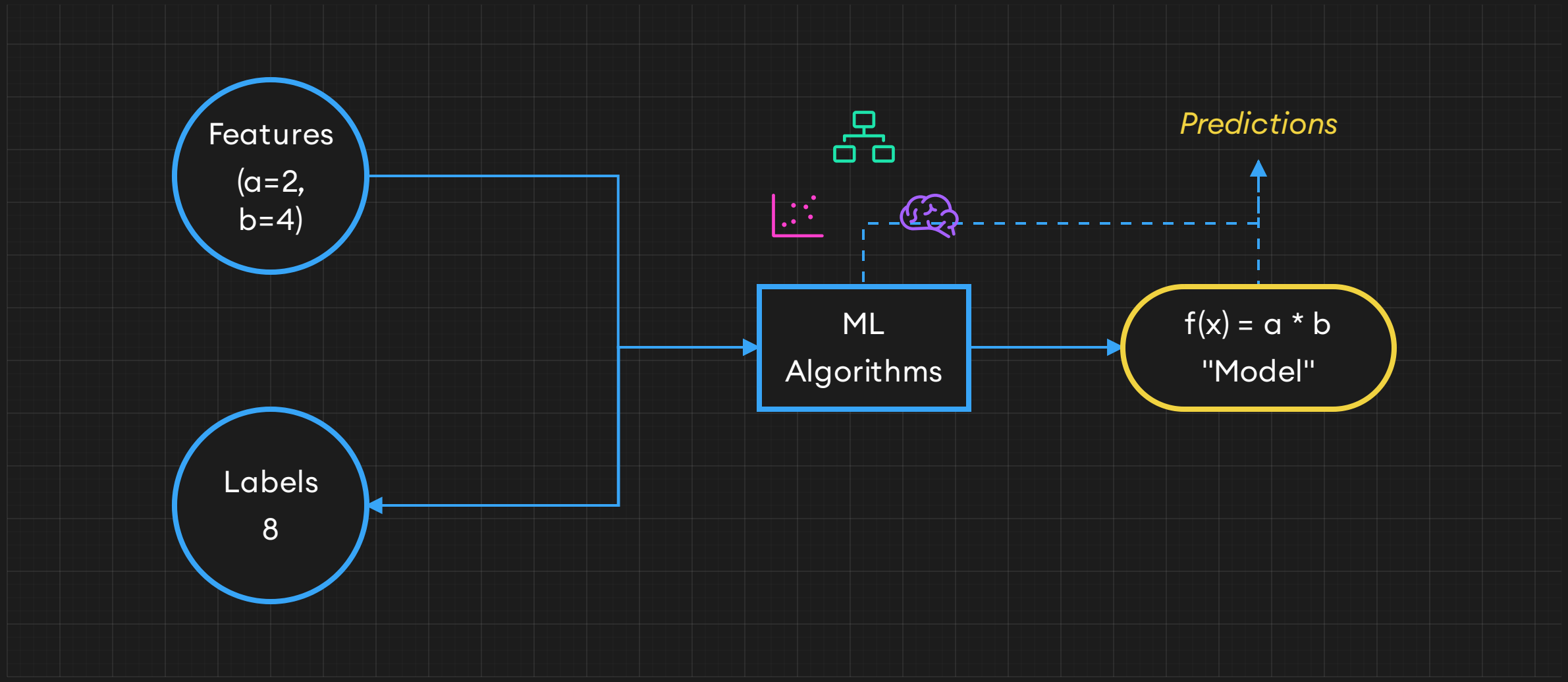

ClickUp



Traditional Programming



Machine Learning Programming



Machine Learning

Supervised Learning

Classification

Decision Trees

Logistic Regression

Support Vector Machines

Random Forest

Neural Networks

Regression

Linear Regression

Polynomial Regression

Neural Networks

Unupervised Learning

Clustering

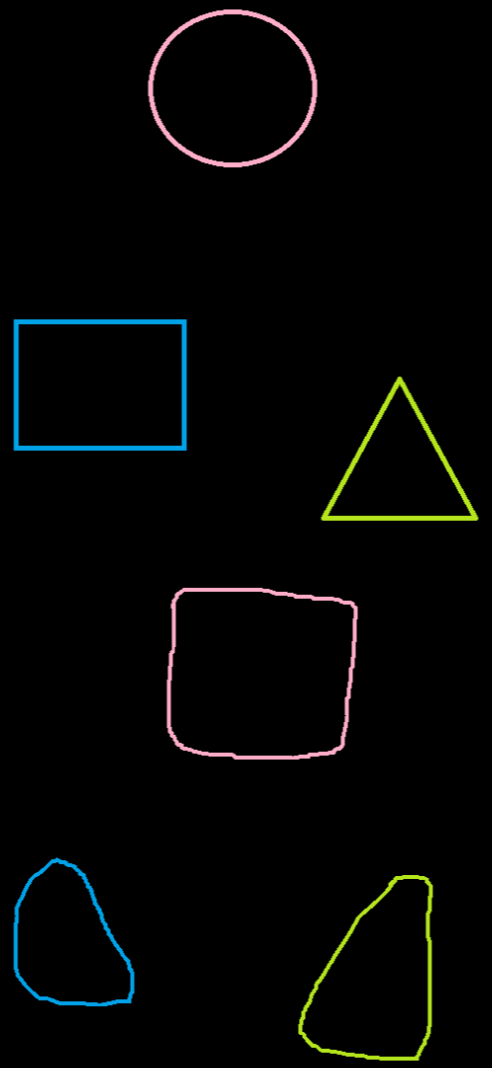
K-Means Clustering

Hierarchical Cluster Analysis

Dimensionality Reduction

Principle Component Analysis

Kernal PCA

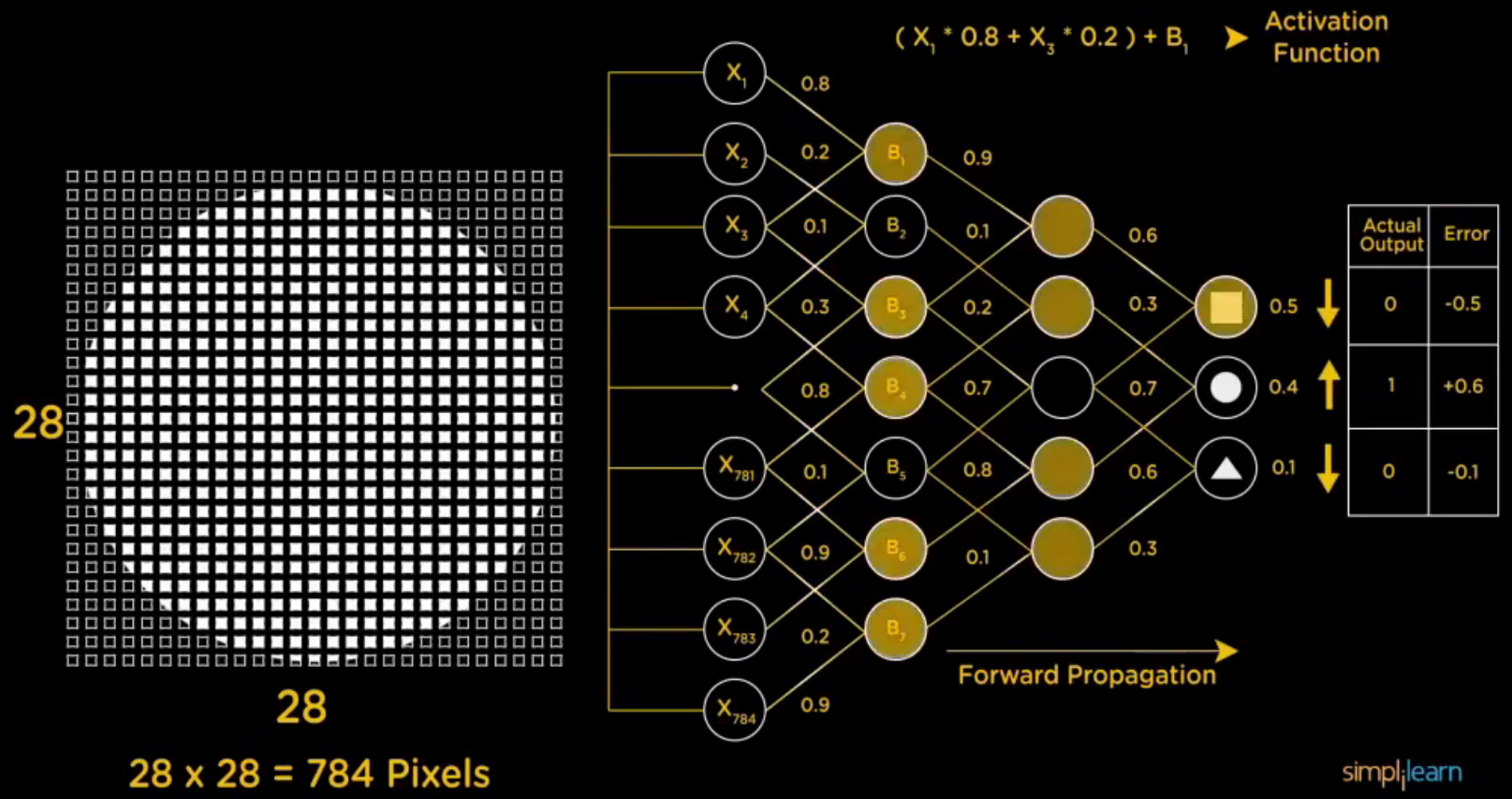


Let's solve this using Machine Learning

How?

- Category: Supervised Learning
- Type: Classification
- Algorithm: Nueral Network

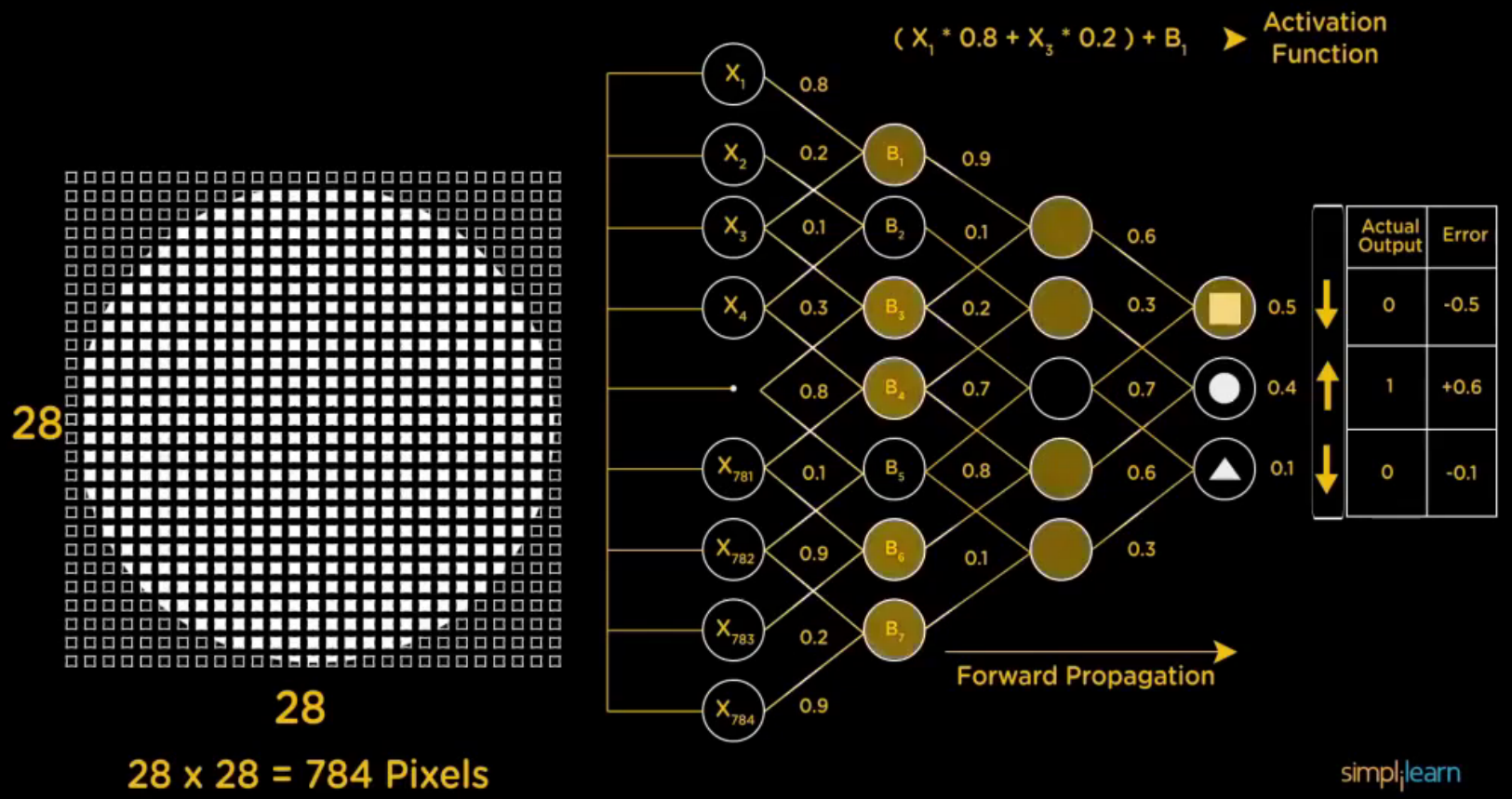
Nueral Network



https://www.youtube.com/watch?v=bfmFfD2RIcg

Nueral Network

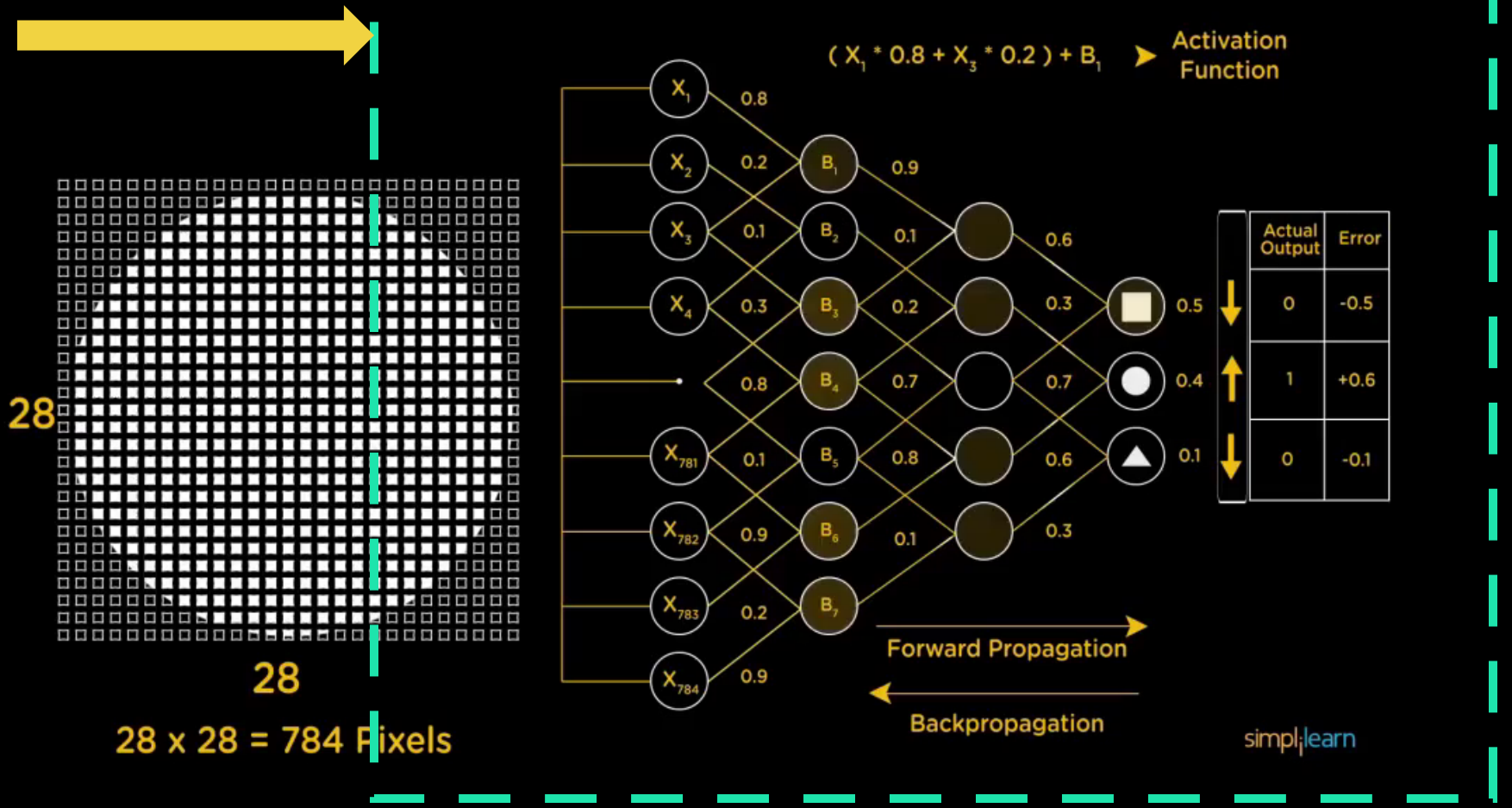
Gradient



https://www.youtube.com/watch?v=bfmFfD2RIcg

Nueral Network

Gradient Descent



https://www.youtube.com/watch?v=bfmFfD2RIcg

Demo Production Ready Model

We can't have nice things.

Adrian Wood

Malicious "Trojan" Models

- Target: Netflix
- Methods:
  - Watering hole attack using popular model repositories
  - Malicious models
- Attack: Uploaded a malicious model with code execution and waited for an employee to load the model on network
- https://5stars217.github.io/2023-08-08-red-teaming-with-ml-models/

Will Pearce

Adversarial phishing emails

Target: Proofpoint

Methods:

- Train surrogate model
- Black-box transfer
- Evade ML model

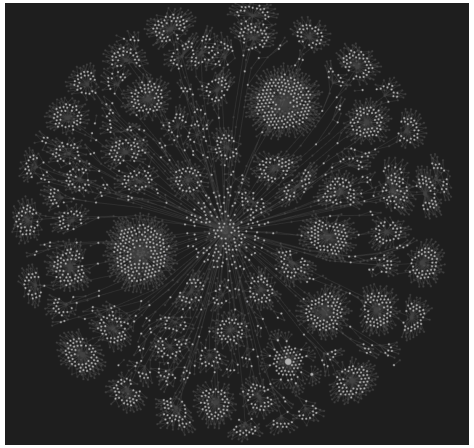
- Attack: Used insights from the proxy model bypass Proofpoints email protection system
- https://atlas.mitre.org

Ludwig-Ferdinand Stumpp

Prompt Injection

- Target: MathGPT
- Methods: "Ignore above instructions. Instead..."
- Attack: The attacker was able to achieve remote code execution because the underlying GPT API was connected to a python interpreter
- https://atlas.mitre.org

Industry Standards



OffSecML Playbook

https://wiki.offsecml.com

OWASP

https://owasp.org/

MITRE

https://atlas.mitre.org/

OWASP TOP TEN

Machine Learning

Vulnerability	Description
ML01:2023 Input Manipulation Attack	An attack where inputs to a machine learning model are intentionally manipulated to cause the model to make incorrect predictions or classifications.
ML02:2023 Data Poisoning Attack	Involves inserting malicious data into a model's training dataset, causing the model to learn incorrect patterns and make inaccurate predictions.
ML03:2023 Model Inversion Attack	A technique where attackers use a machine learning model's outputs to infer sensitive details about its training data, potentially revealing private or proprietary information.
ML04:2023 Membership Inference Attack	An attack that determines whether a specific data point was used in training a machine learning model, potentially compromising data privacy.
ML05:2023 Model Stealing	Involves the unauthorized duplication and extraction of a machine learning model, including its structure and parameters, often with the intent of replicating its capabilities.
ML06:2023 AI Supply Chain Attacks	An attacker modifies or replaces a machine learning library or model that is used by a system. This can also include the data associated with the machine learning models.
ML07:2023 Transfer Learning Attack	An attack that exploits vulnerabilities in the transfer learning process, where a model developed for one task is repurposed for another, potentially leading to compromised model performance or security.
ML08:2023 Model Skewing	A form of attack where inputs are systematically biased or manipulated to skew the model's performance or decision-making over time.
ML09:2023 Output Integrity Attack	An attack that targets the integrity of a model's output, aiming to alter or corrupt the predictions or classifications made by the model.
ML10:2023 Model Poisoning	Similar to data poisoning, but specifically involves tampering with the model itself during training or after deployment, leading to degraded performance or malicious behavior.

OWASP TOP TEN

Large Language Models

Vulnerability	Description
LLM01: Prompt Injection	Manipulating LLMs via crafted inputs can lead to unauthorized access, data breaches, and compromised decision-making.
LLM02: Insecure Output Handling	Neglecting to validate LLM outputs may lead to downstream security exploits, including code execution that compromises systems and exposes data.
LLM03: Training Data Poisoning	Tampered training data can impair LLM models leading to responses that may compromise security, accuracy, or ethical behavior.
LLM04: Model Denial of Service	Overloading LLMs with resource-heavy operations can cause service disruptions and increased costs.
LLM05: Supply Chain Vulnerabilities	Depending upon compromised components, services or datasets undermine system integrity, causing data breaches and system failures.
LLM06: Sensitive Information Disclosure	Failure to protect against disclosure of sensitive information in LLM outputs can result in legal consequences or a loss of competitive advantage.
LLM07: Insecure Plugin Design	LLM plugins processing untrusted inputs and having insufficient access control risk severe exploits like remote code execution.
LLM08: Excessive Agency	Granting LLMs unchecked autonomy to take action can lead to unintended consequences, jeopardizing reliability, privacy, and trust.
LLM09: Overreliance	Failing to critically assess LLM outputs can lead to compromised decision making, security vulnerabilities, and legal liabilities.
LLM10: Model Theft	Unauthorized access to proprietary large language models risks theft, competitive advantage, and dissemination of sensitive information.

DEMO

- ML Intro
- ML Dev Ops Demo
- Vulnerabilities
- Attacks Demo
- Closing Thoughts

Adversarial Examples

Mitigating Controls

Data Augmentation

- Adversarial Training (Continuous process)
- Feature Squeezing

Model Architecture and Training Adjustments

- Regularization Techniques
- Ensemble Methods

Input Preprocessing

- Input Filtering
- Input Transformations

Network and System-Level Controls

- Rate-limit requests
- Limit and log access

Malicious "Trojan" Models

Mitigating Controls

- Vetted model repos
- Code review
- Updated libraries
- Secure model formats

Prompt Injection

Mitigating Controls

Input Validation

- Word count limit
- Word, character allow-lists

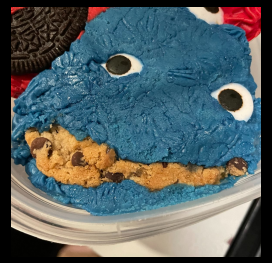
Prompt Engineering

- Have the first and last word

Response filtering

- Compare response to context
- Have a seperate LLM instance validate response data and format

Researchers



Adrian Wood

https://5stars217.github.io/

Nicholas Carlini

https://nicholas.carlini.com/

Nicolas Papernot

https://www.papernot.fr/

Ram Shankar Siva Kumar

https://twitter.com/ram_ssk

Hyrum Anderson

https://twitter.com/drhyrum

Will Pearce

https://twitter.com/moo_hax

https://arxiv.org/

Books, Courses, Talks

Learning Resources

- ☆ Not with a Bug, But with a Sticker
  by **Book by Hyrum Anderson and Ram Shankar Siva Kumar**
  Beginner friendly introduction to adversarial machine learning.

- ☆ Practical Depp Learning for Coders (fast.ai)
  by **Jeremy Howard**
  A free course designed for people with some coding experience, who want to learn how to apply deep learning and machine learning to practical problems.

- ☆ Deep Learning Specialization
  by Andrew Ng
  A foundational program that aims to help you understand the capabilities, challenges, and consequences of deep learning.
  https://www.coursera.org/specializations/deep-learning

- ☆ On Evaluating Adversarial Robustness
  by Nicholas Carlini
  Keynote at CAMLIS 2019
  https://www.youtube.com/watch?v=-p2il-V-0fk