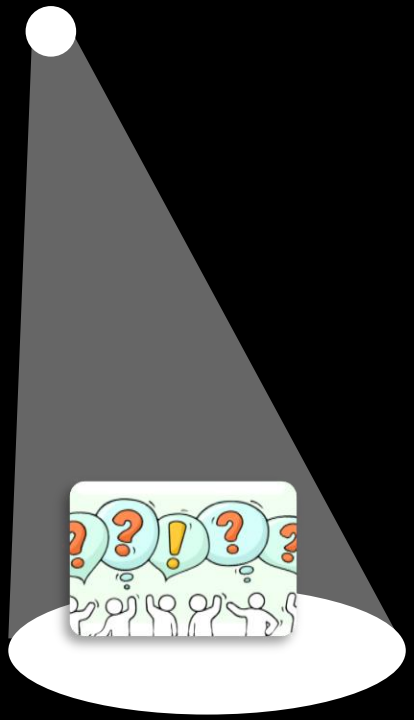
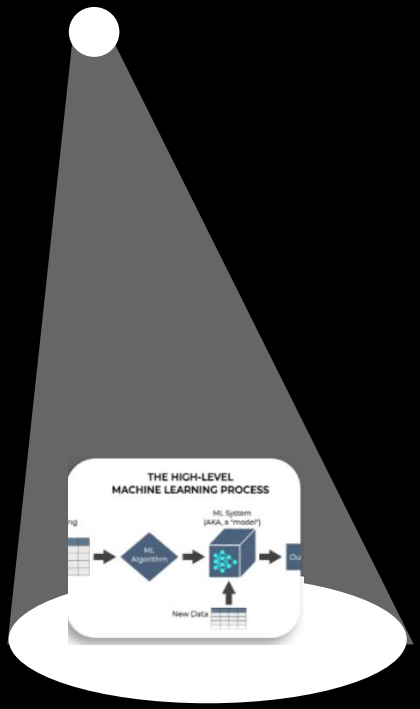
Machine Learning For Security Professionals

Building And Hacking ML Systems

Seasides Goa 2023



Intro



ML for Security

Security for ML

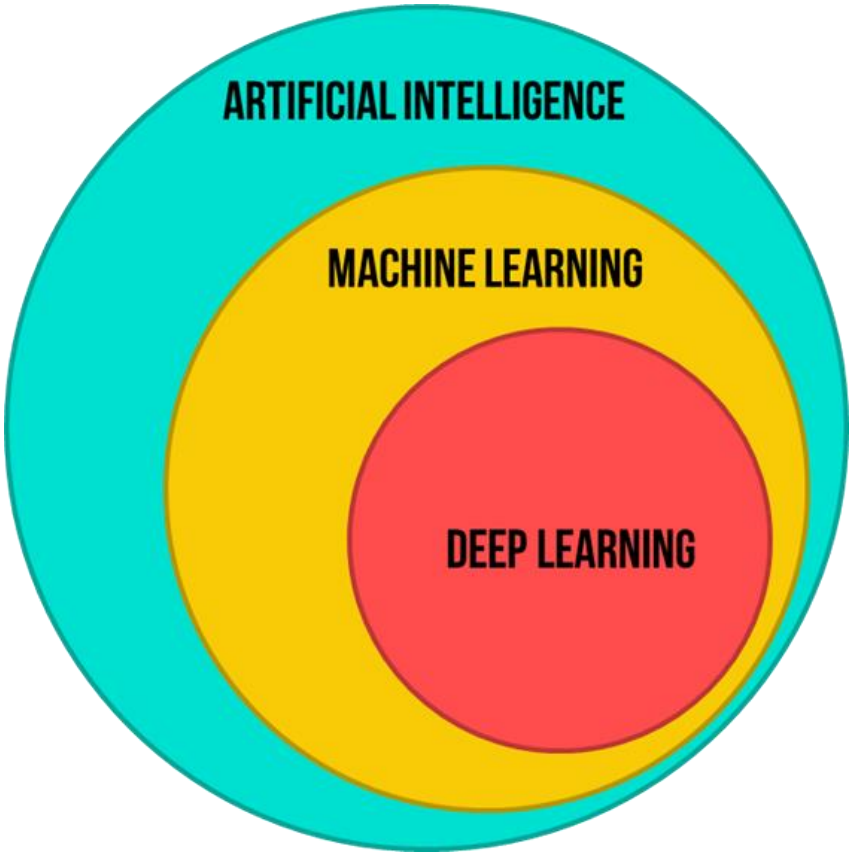
$whoami

SAGAR BHURE

Software Engineer | OWASP Project Lead | Blackhat Speaker | Mentor

Intro

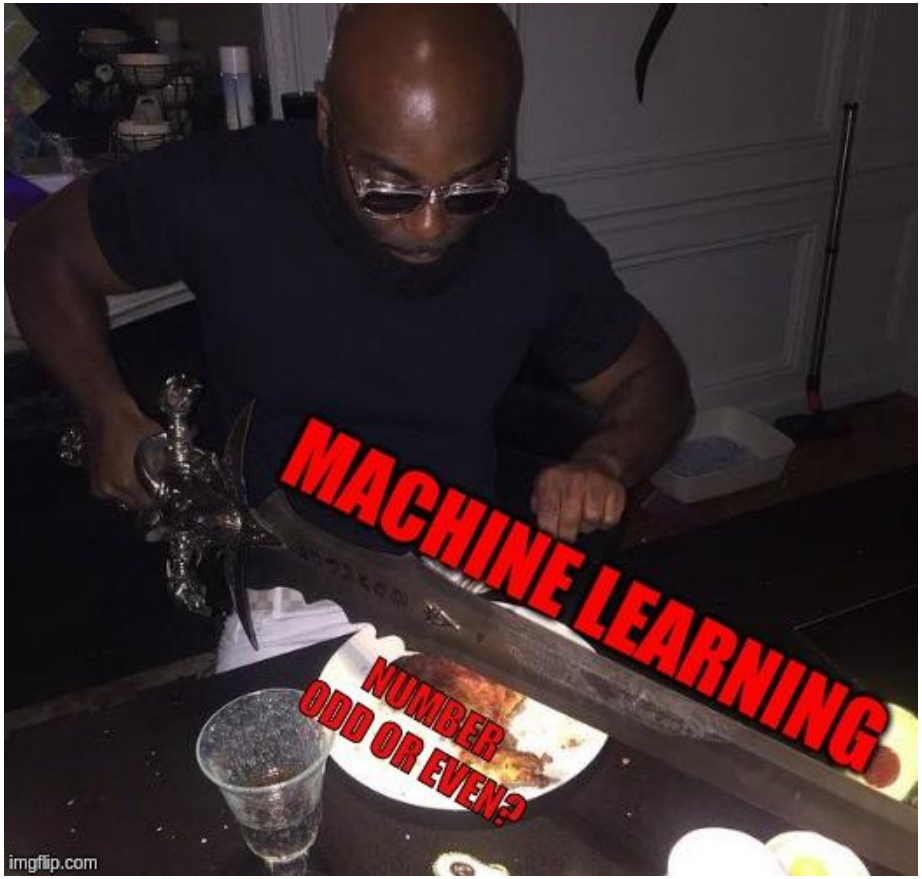
DL ≠ ML ≠ AI



ML/DL Intro: What for?

- Pattern matching
- Pattern search
- Classification
- Regression
- Clustering
- Forecasting
- etc.

Where not to use ML?



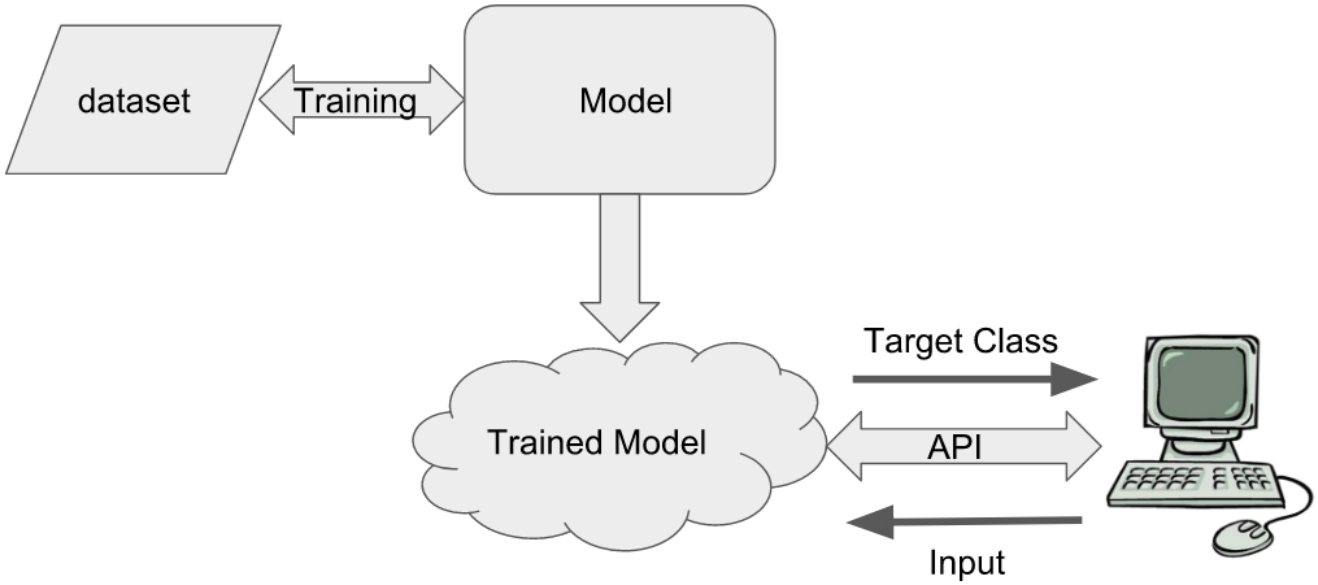
Applications in security

- WAF
- IDS/IPS
- Malware detection
- Antiviruses
- Spam filters
- ...

Libs

Tools

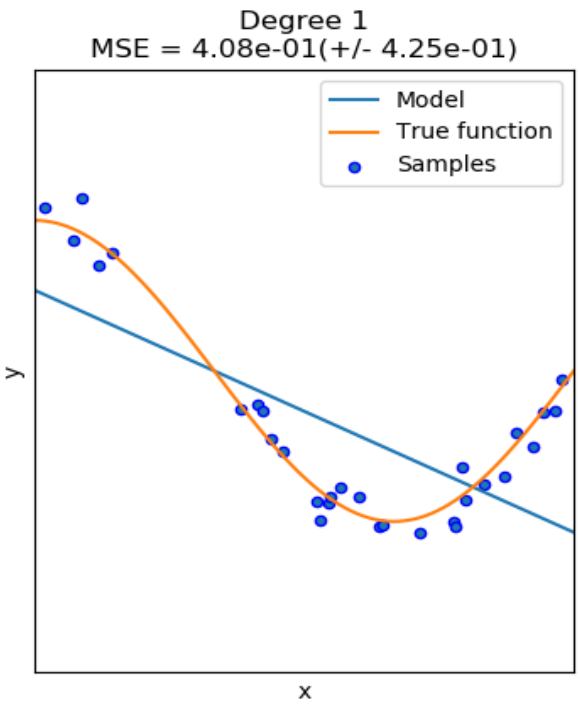
MlOps

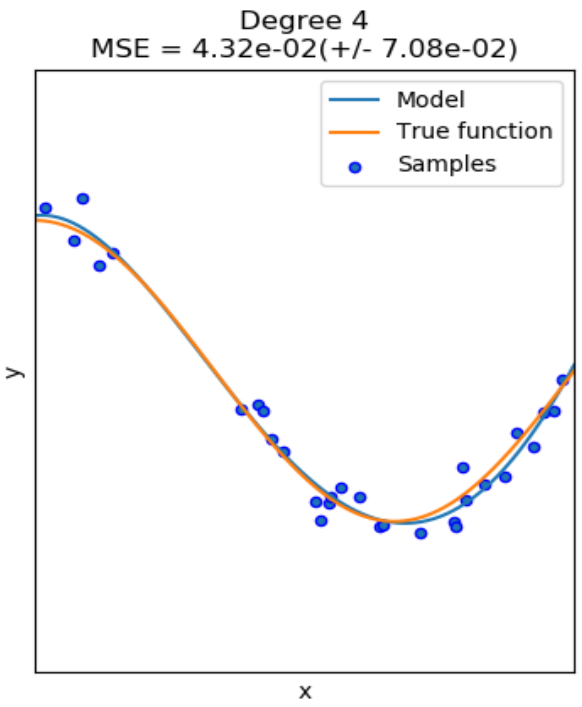

dataset
Training
Model
Trained Model
Target Class
API
Input

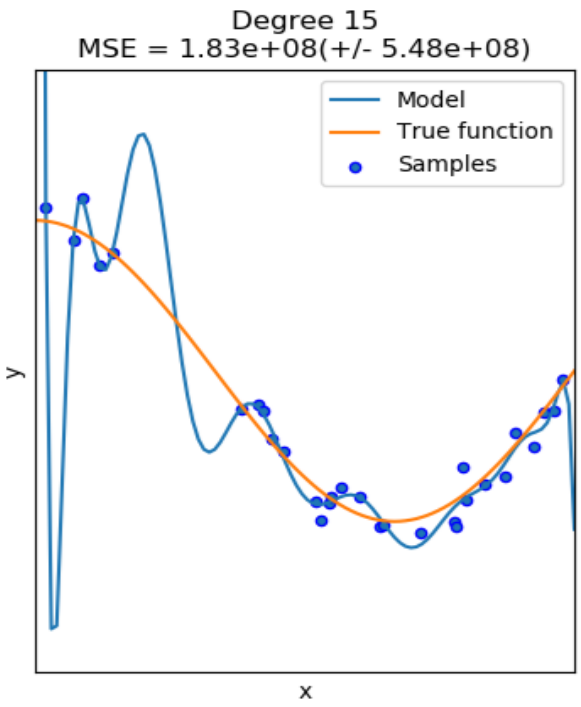
Basic Terminology

- Dataset
- Features and Labels
- Training set and Testing set
- Probability
- Model
- Loss function
- Optimization
- Overfitting
- Underfitting

Underfitting vs Overfitting



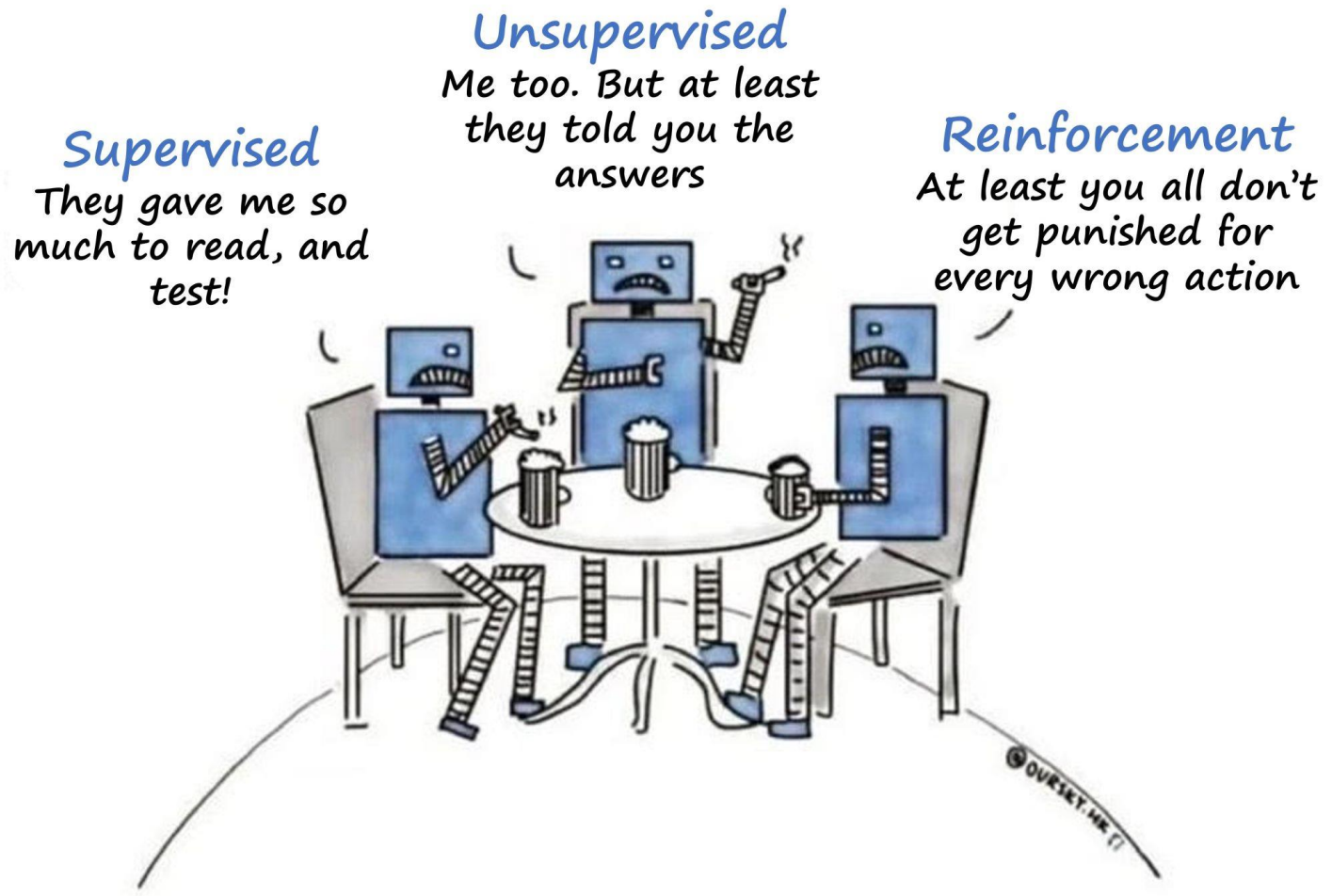




Basic Terminology

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning

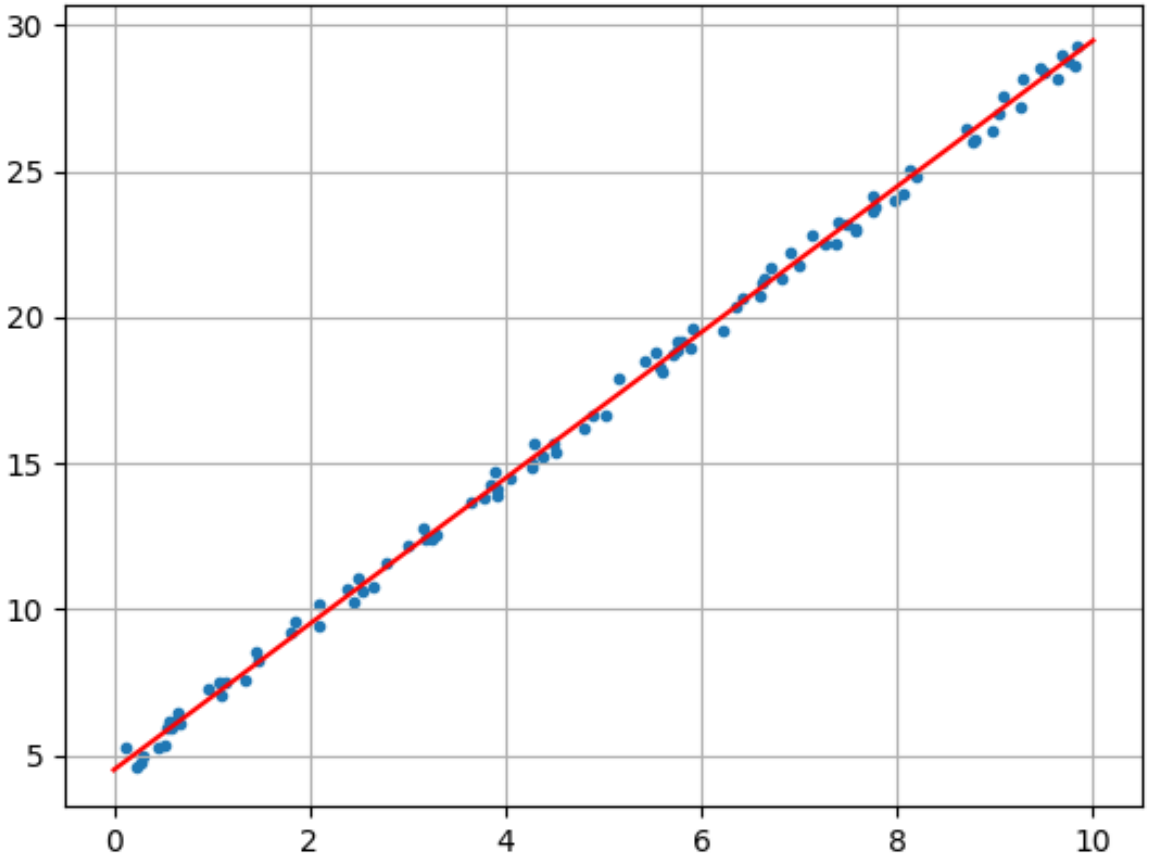
Basic Terminology



I know, right


CS Freshie
Machine Learning
Math prerequisites
Linear Algebra
Calculus

Linear Regression



Linear Regression

Demo 1: Linear Regression

<Linear Regression End>

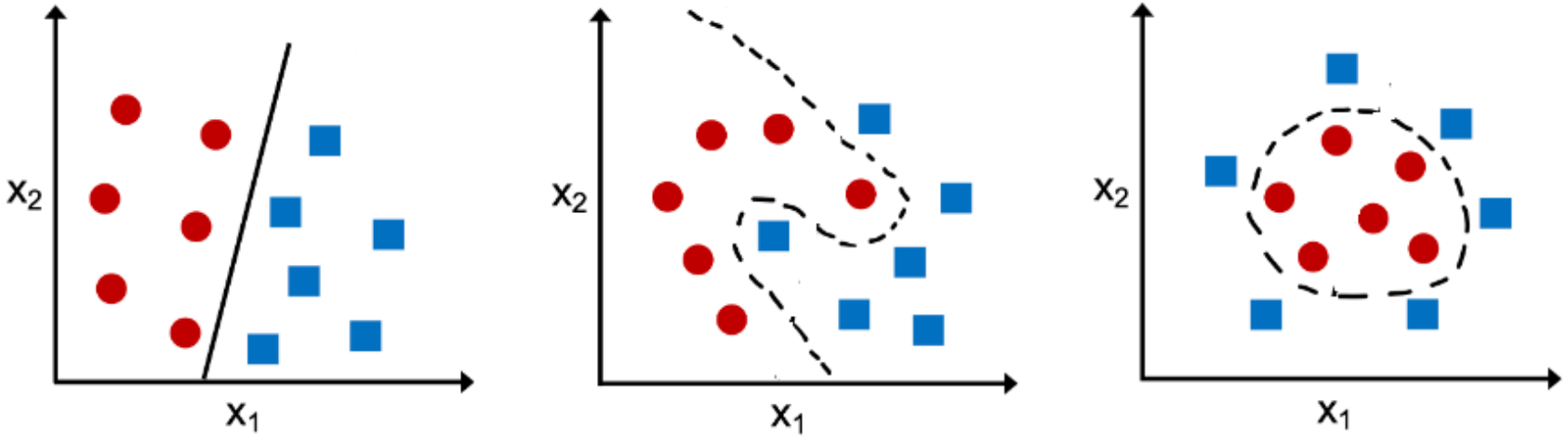
Before we begin

Teachable Machine

https://teachablemachine.withgoogle.com/

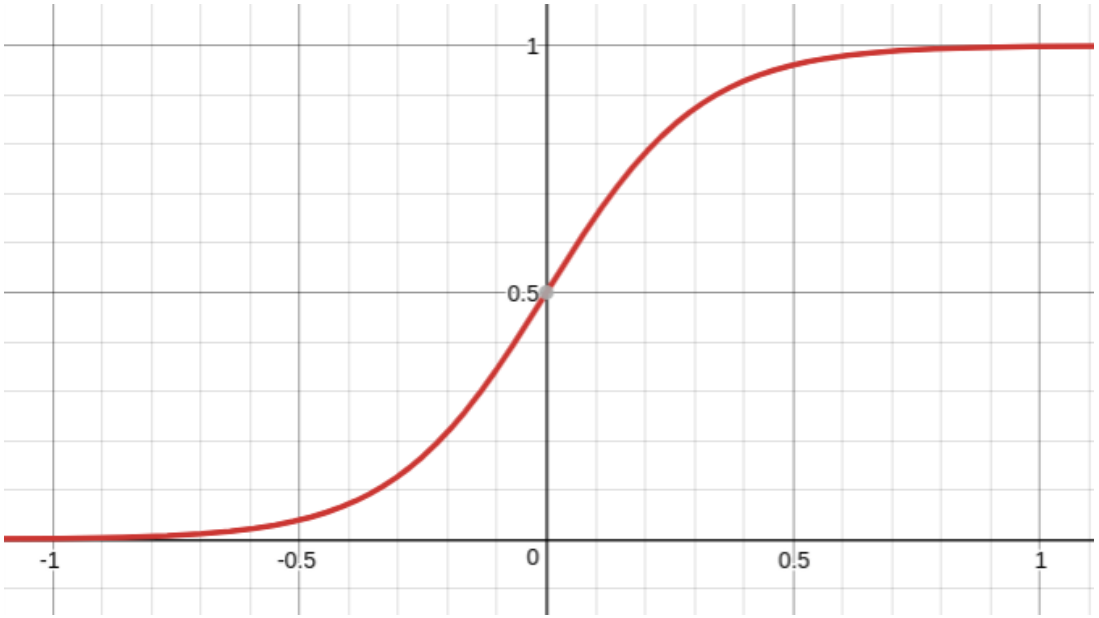
{ref} https://www.tensorflow.org/js/demos

Classification

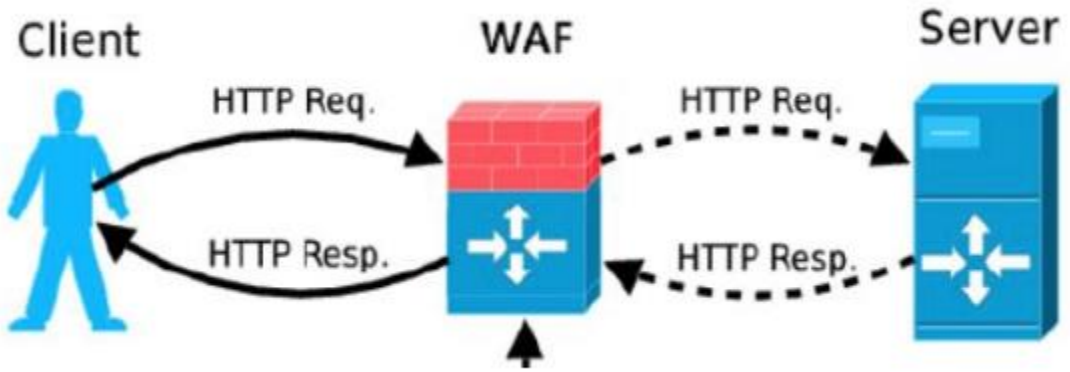


Logistic Regression

$$\frac{1}{1+e^{-(wx+b)}}$$



WAF using Logistic Regression



Ref: https://github.com/faizann24/Fwaf-Machine-Learning-driven-Web-Application-Firewall

DataSet

Good queries

```
/final_warning/
/javascript/resize.cs
/javascript/msg.rtf
/javascript/cat.rb
/buy_phente/
//yske4gpe.asp
/javascript/menu.meta
/avalanche-research_0/
/privacy-gnw/
/176232/
/examples/jsp/checkbox/webc.cgi/
/cb_line/
/bmw_3er/
/155362/
/javascript/shopping.nsf
```

Bad queries

```
/<script>document.cookie="testnvxc=4301;"</script>
/cacti/base_local_rules.php?dir=<script>alert('base_l
ocal_rules_xss.nasl-1331905098')</script>
/scripts/comersus_customerregistrationform.asp
/cgi-bin/search.pl?form=../../../../../../etc/passwd\
x00
/help.php?q='uname >q23433168 #
/u8yiqilu.jsp?<meta http-equiv=set-cookie content="te
stswjp=6010">
/scripts/forum.php3?id_article=1&id_forum=-1/**/union
/**/select 1376526276--
/core/adodb/server.php?sql='adodb_sql_sql_injection.n
asl
/xsqln7eb.aspx?<meta http-equiv=set-cookie content="t
estpcbb=9254">
```

Ref: https://github.com/faizann24/Fw

Part one

Preprocessing data

TF-IDF

$$TF_{ij} = \frac{f_{ij}}{n_j} \tag{1}$$

Where f_{ij} is the frequency of term i in document j. n_j is the total number of words in document j.

$$IDF_i = 1 + log(\frac{N}{c_i}) \tag{2}$$

Where N is the total number of documents in the corpus. c_i is the number of documents that contain word i.

$$w_{ij} = TF_{ij} \times IDF_i \tag{3}$$

Where w_{ij} is the TF-IDF score of term i in document j.

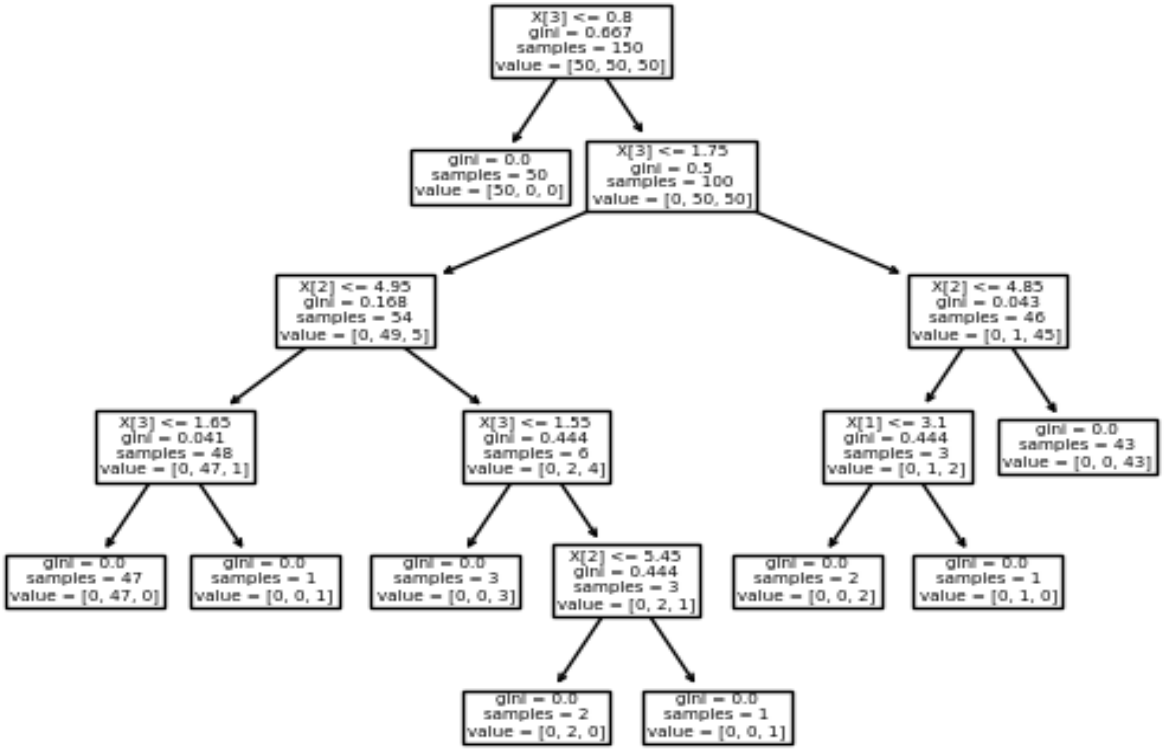
TF-IDF

- Term Frequency: This summarizes how often a given word appears within a document.
- Inverse Document Frequency: This downscales words that appear a lot across documents.

Demo 2: WAF using Logistic regression

<Logistic Regression End>

Decision Tree & Random Forest



Decision Tree & Random Forest

- Gini Impurity: Gives you a likelihood of incorrect classification

$$Gini\ impurity = 1 - \sum_{j=1}^{f} p_j^2$$

- Calculate Gini impurity for all possible features
- Add the node to tree if improved Gini impurity value is seen
- Else do not fork the tree

Decision Tree & Random Forest

- Precision: What proportion of positive identifications was actually correct?

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall: What proportion of actual positives was identified correctly?

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1 score:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Dataset : Malware Classifier

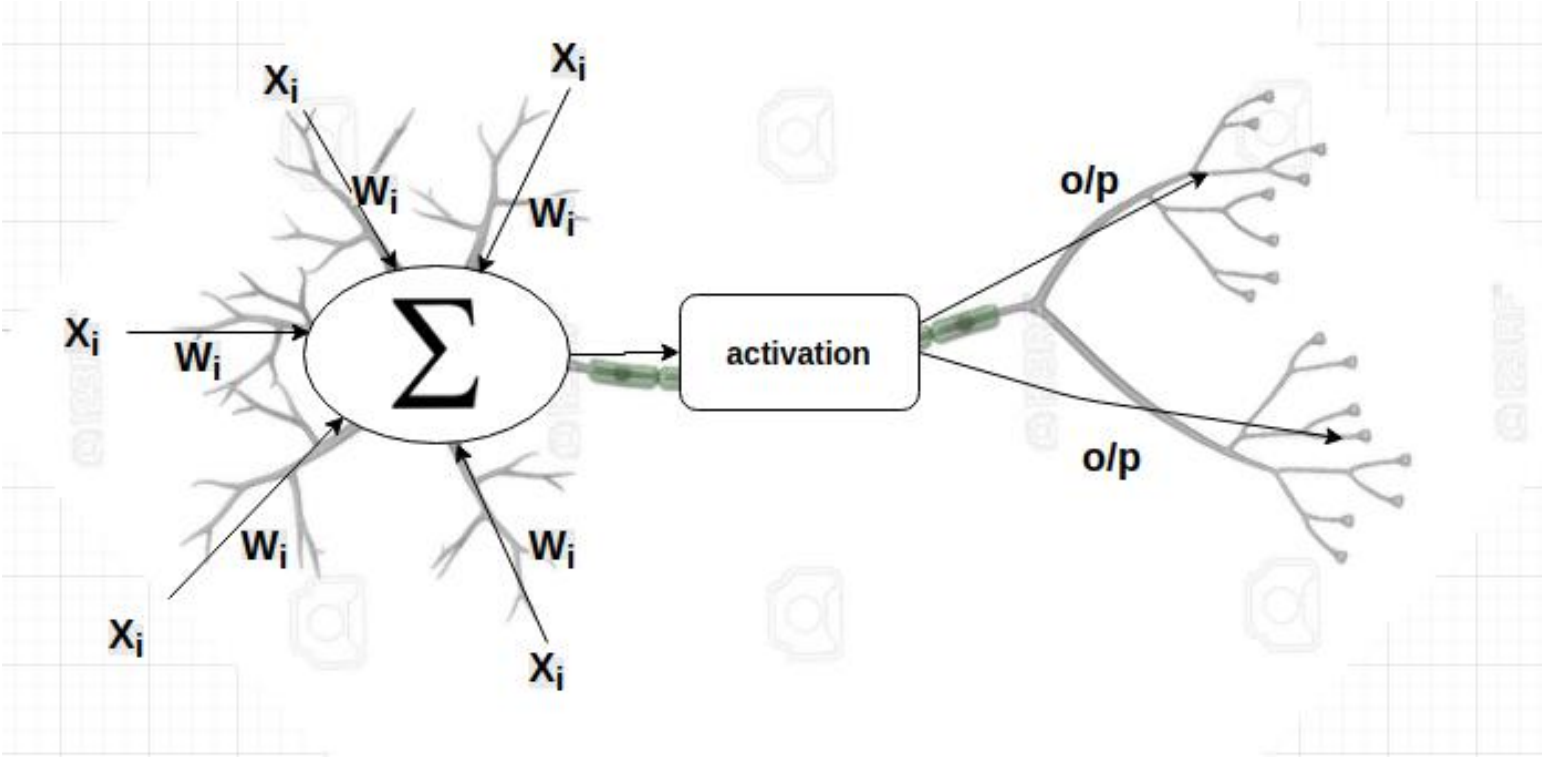
- 56 features extracted from PE files using pefile
- 2 classes
- 138047 samples

{ref}: https://github.com/Te-k/malware-classification

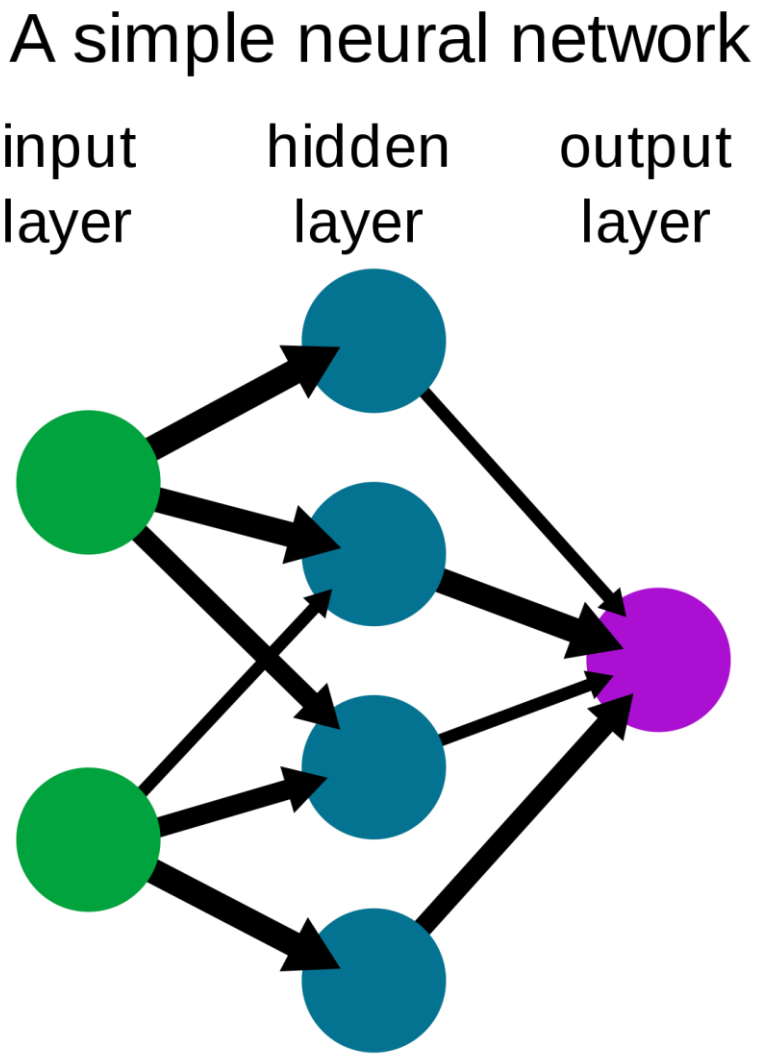
Demo 3: Malware Classifier

<Decision Tree End>

Neural Network



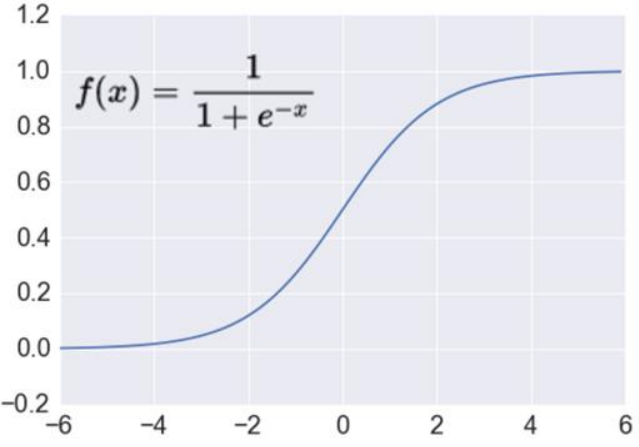
A simple NN


A simple neural network
input layer
hidden layer
output layer

Activation Function

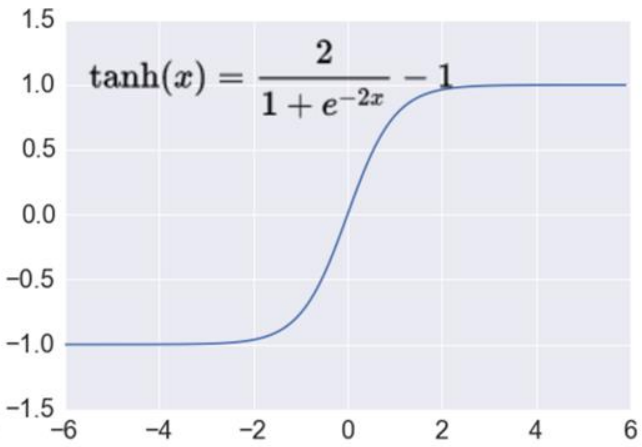
Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$



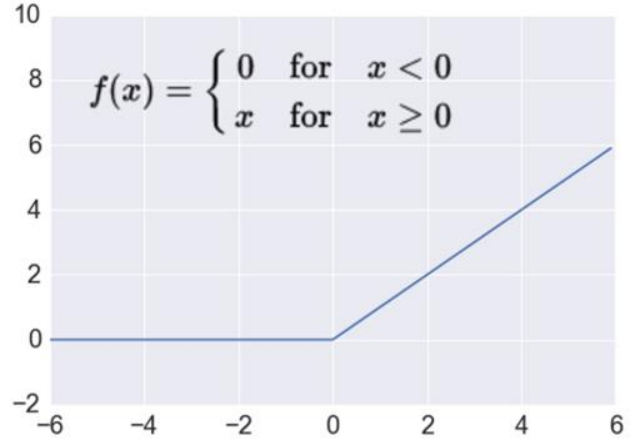
TanH

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

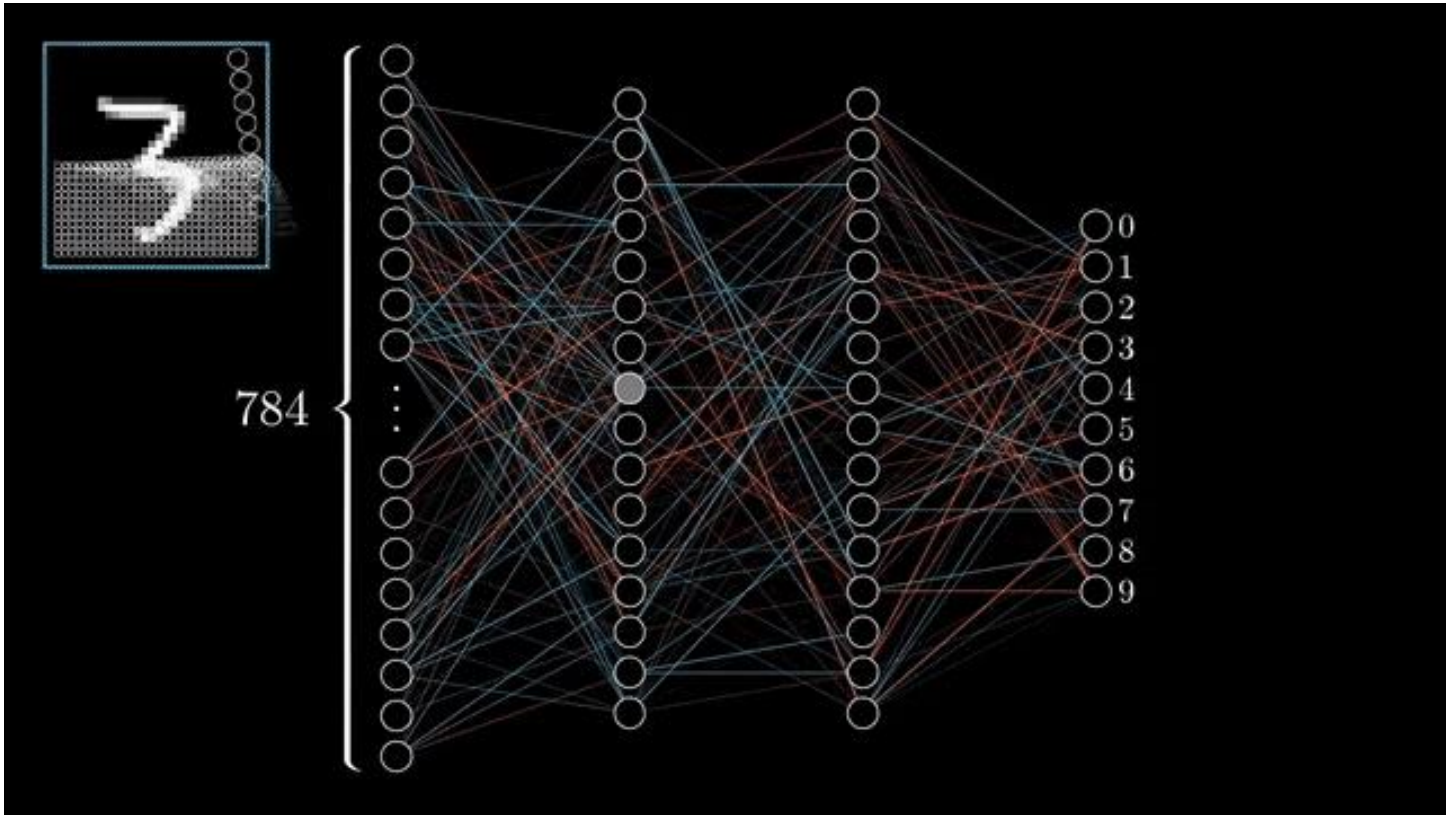


ReLU

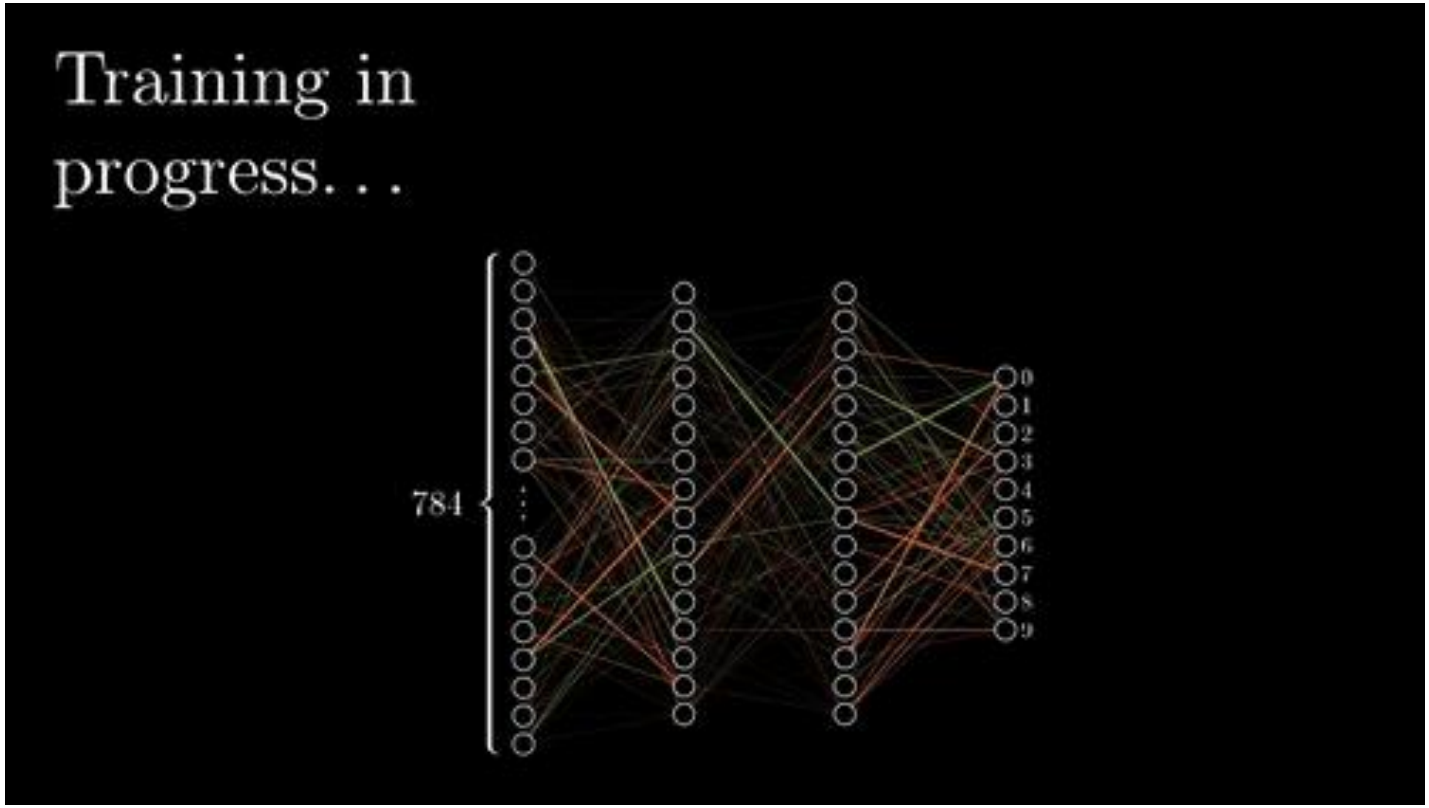
$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$$



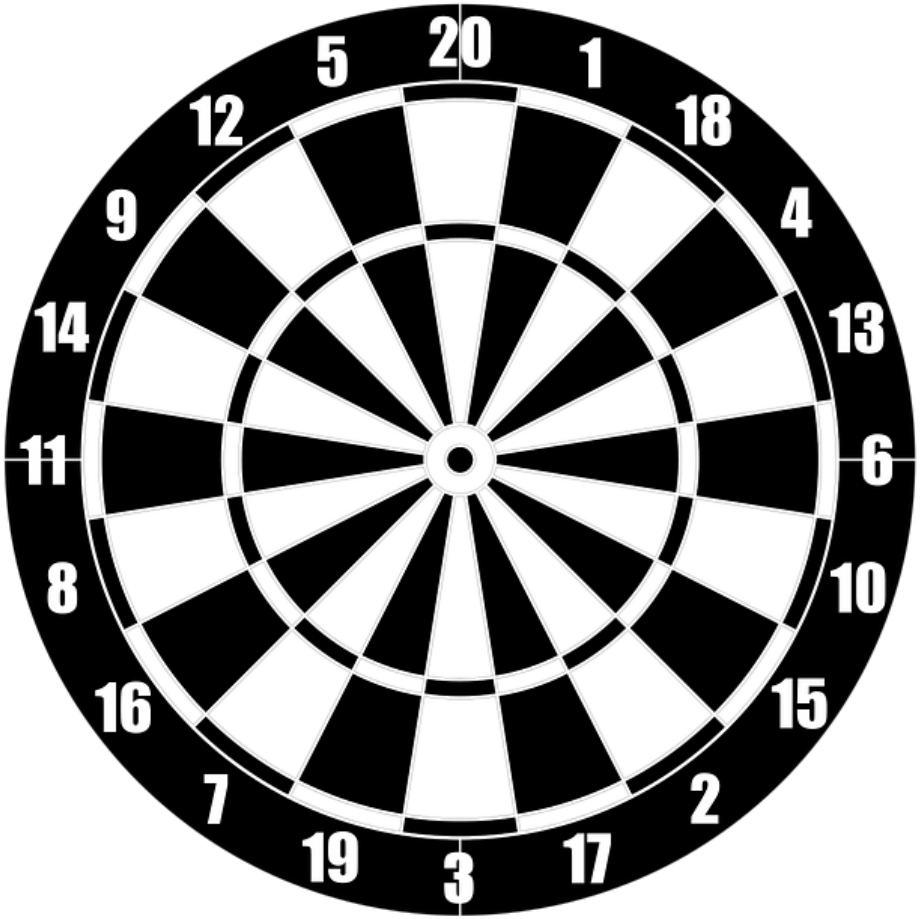
Initialization



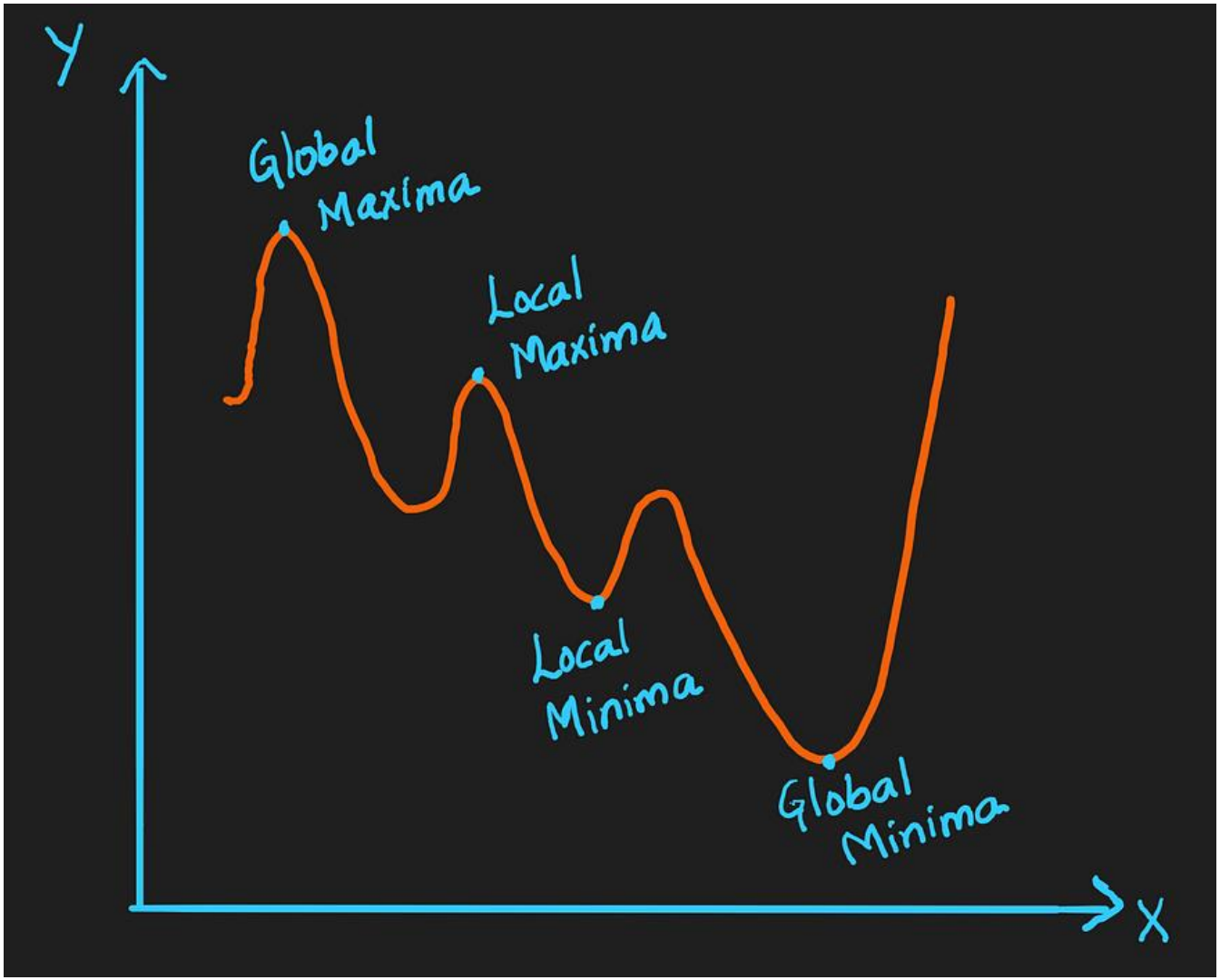
Training



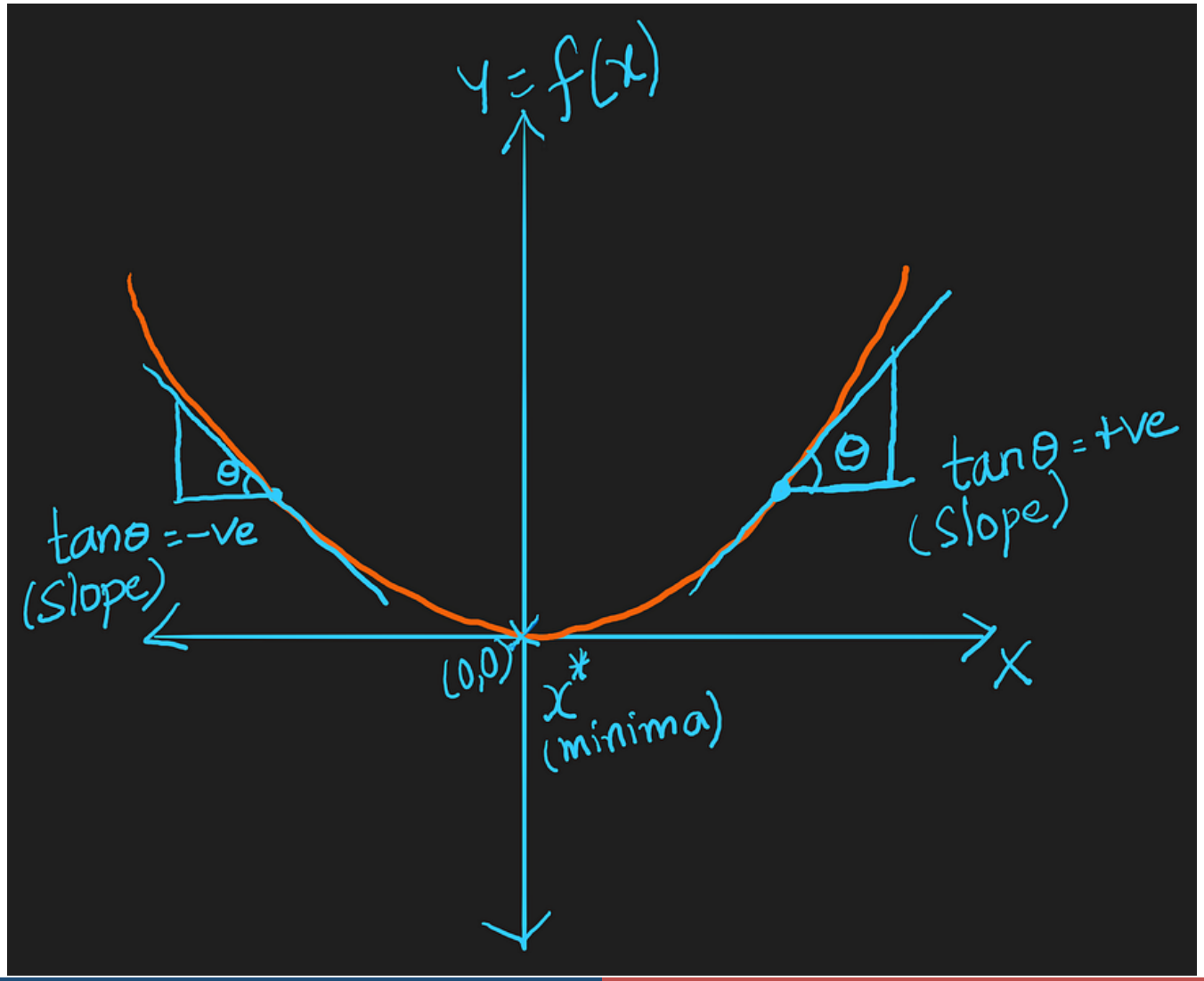
Learning



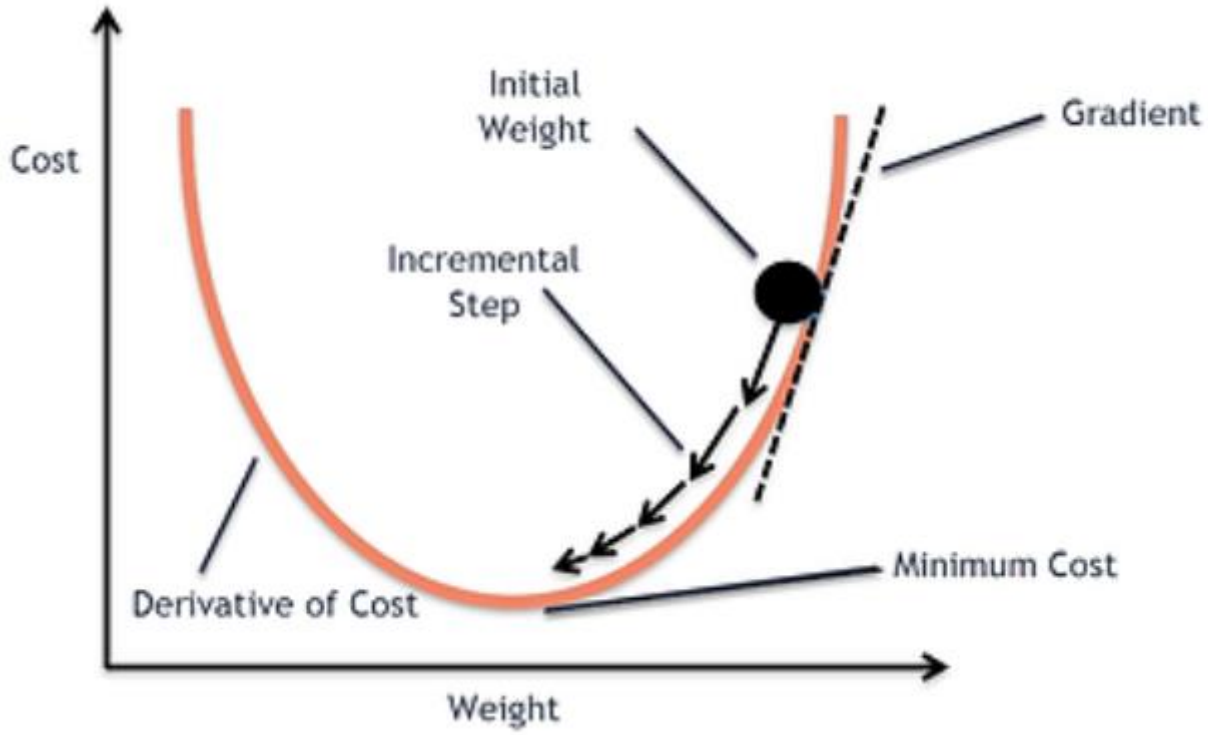
Maths behind Learning – Maxima/Minima



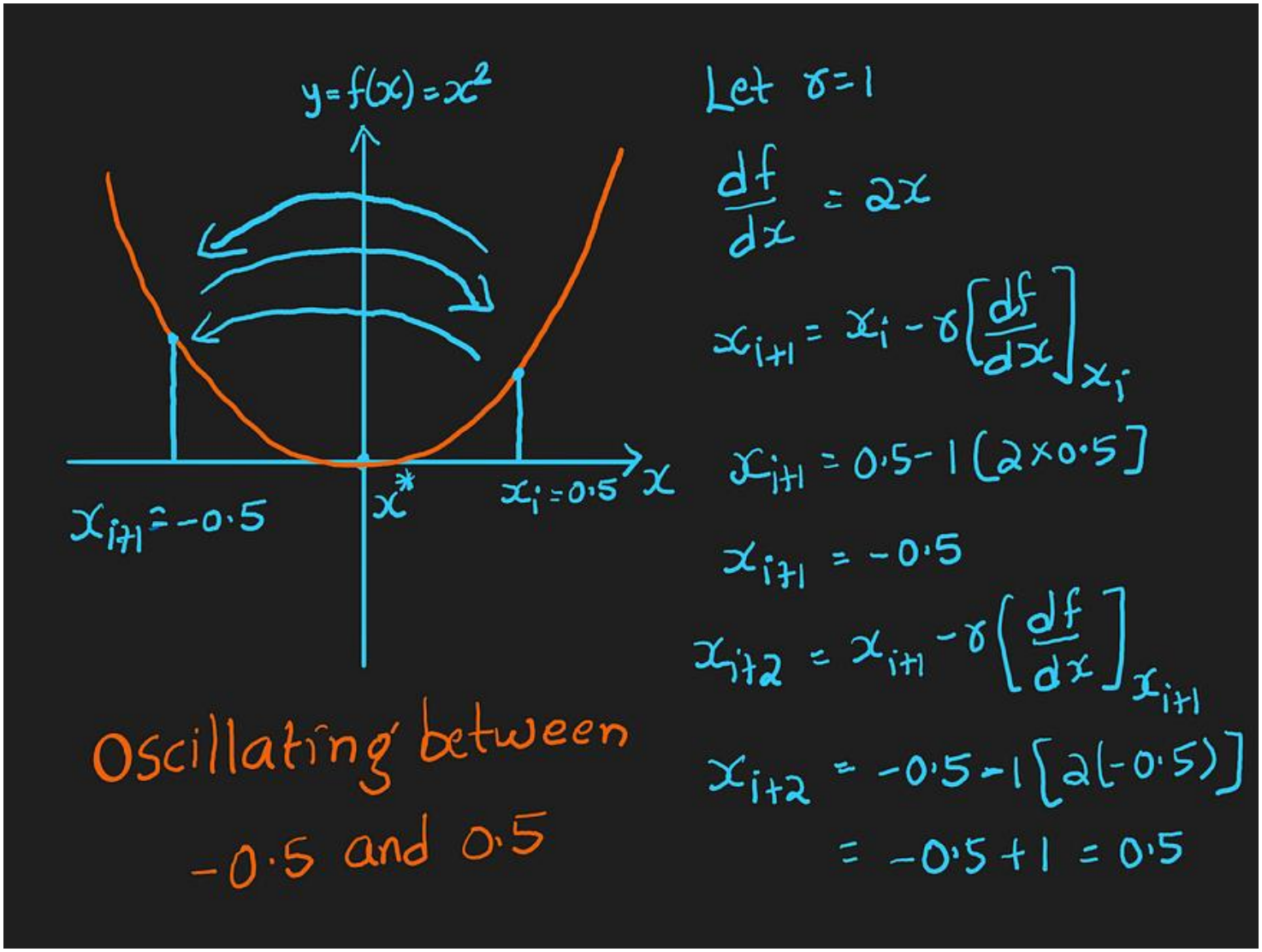
Maths behind Learning – Gradient/Descent



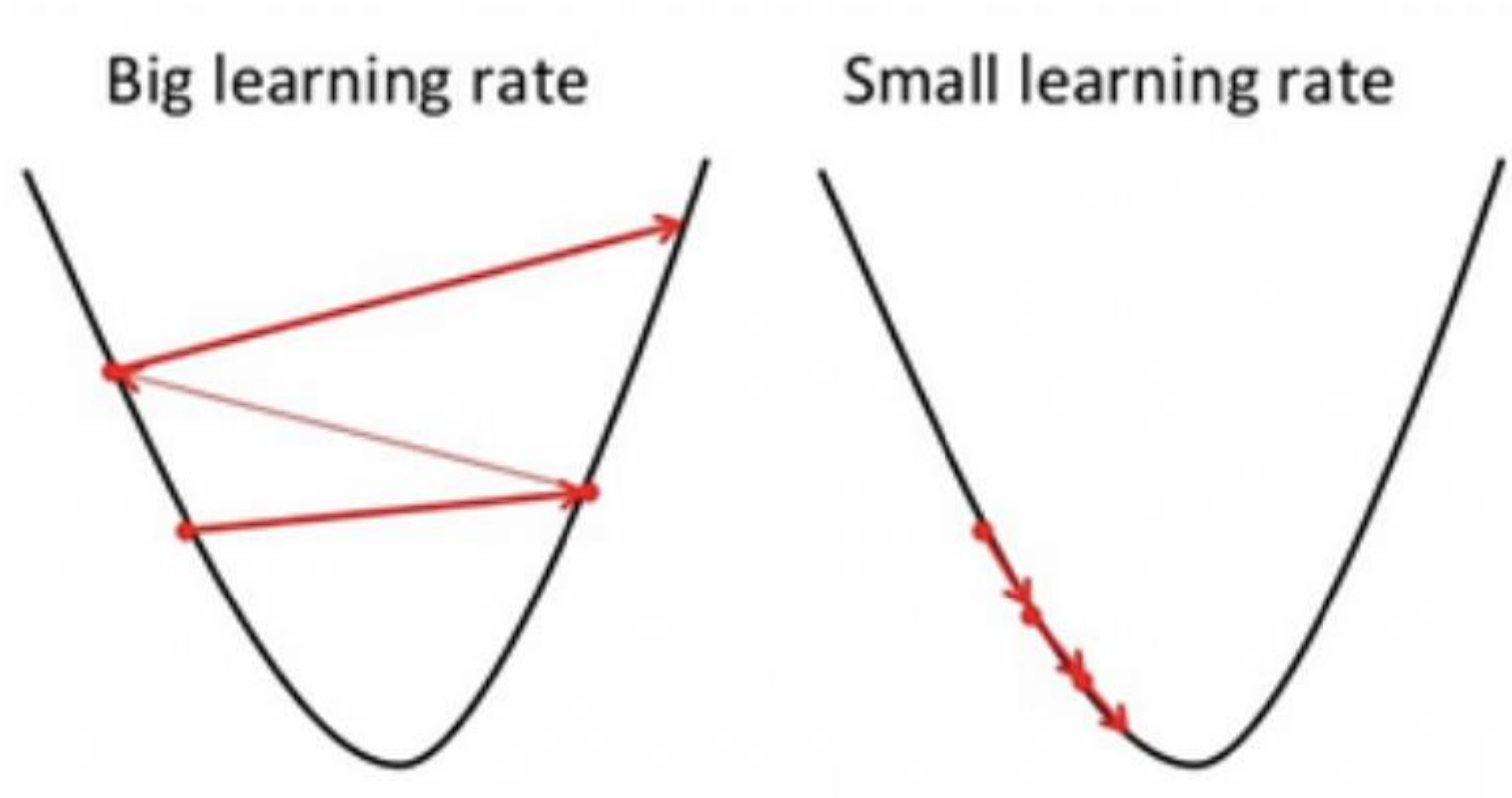
Maths behind Learning – Gradient/Descent



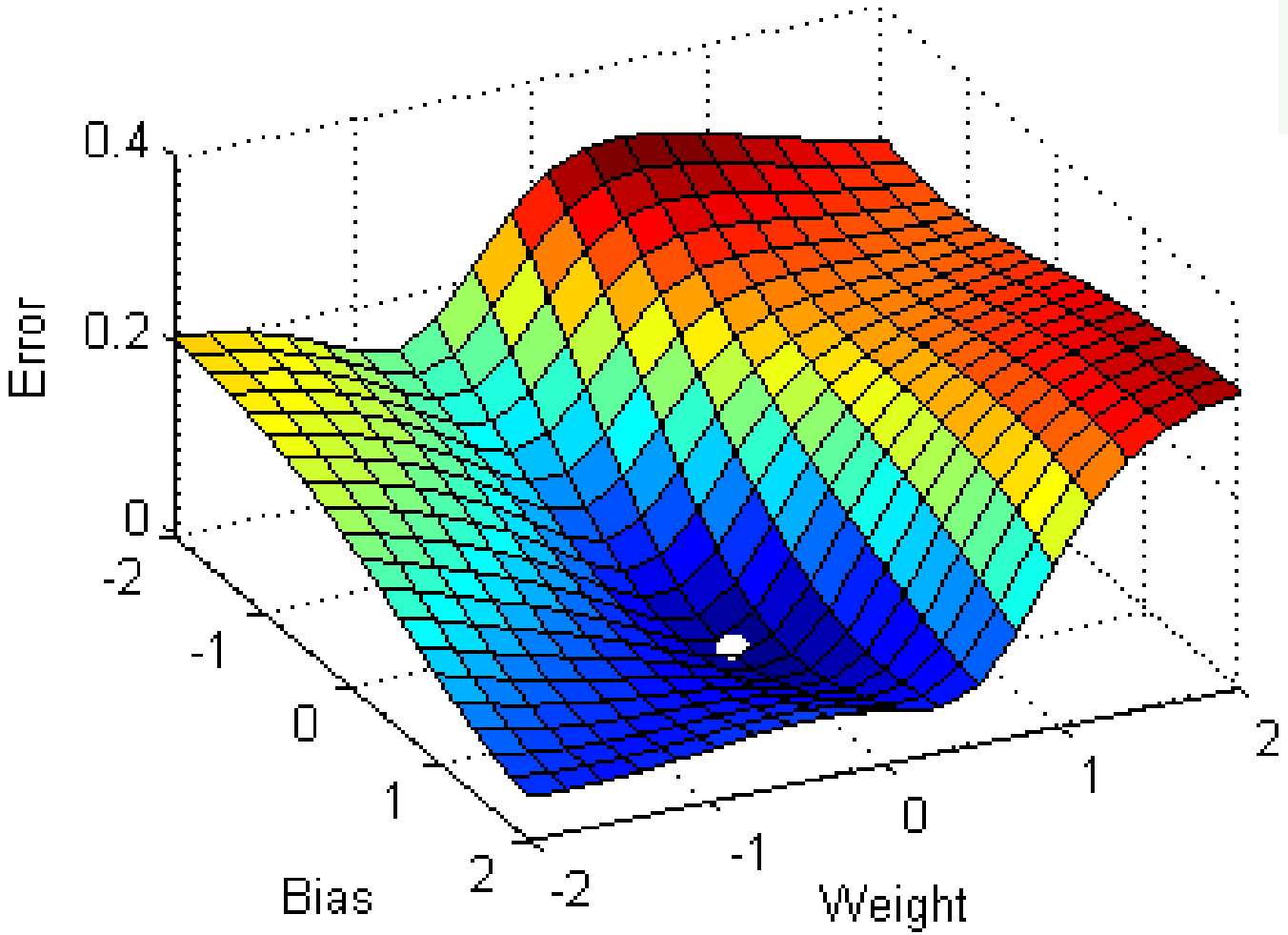
Maths behind Learning – Learning Rate



Maths behind Learning – Learning Rate

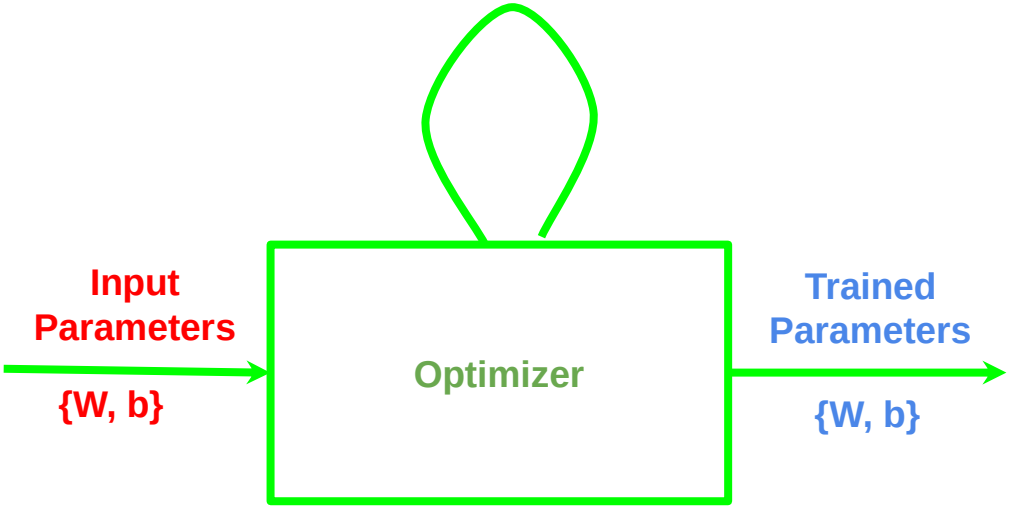


Maths behind Learning - ALL



$$W_{i+1} = W_i + \alpha \frac{\partial}{\partial w} \mathbf{C}$$

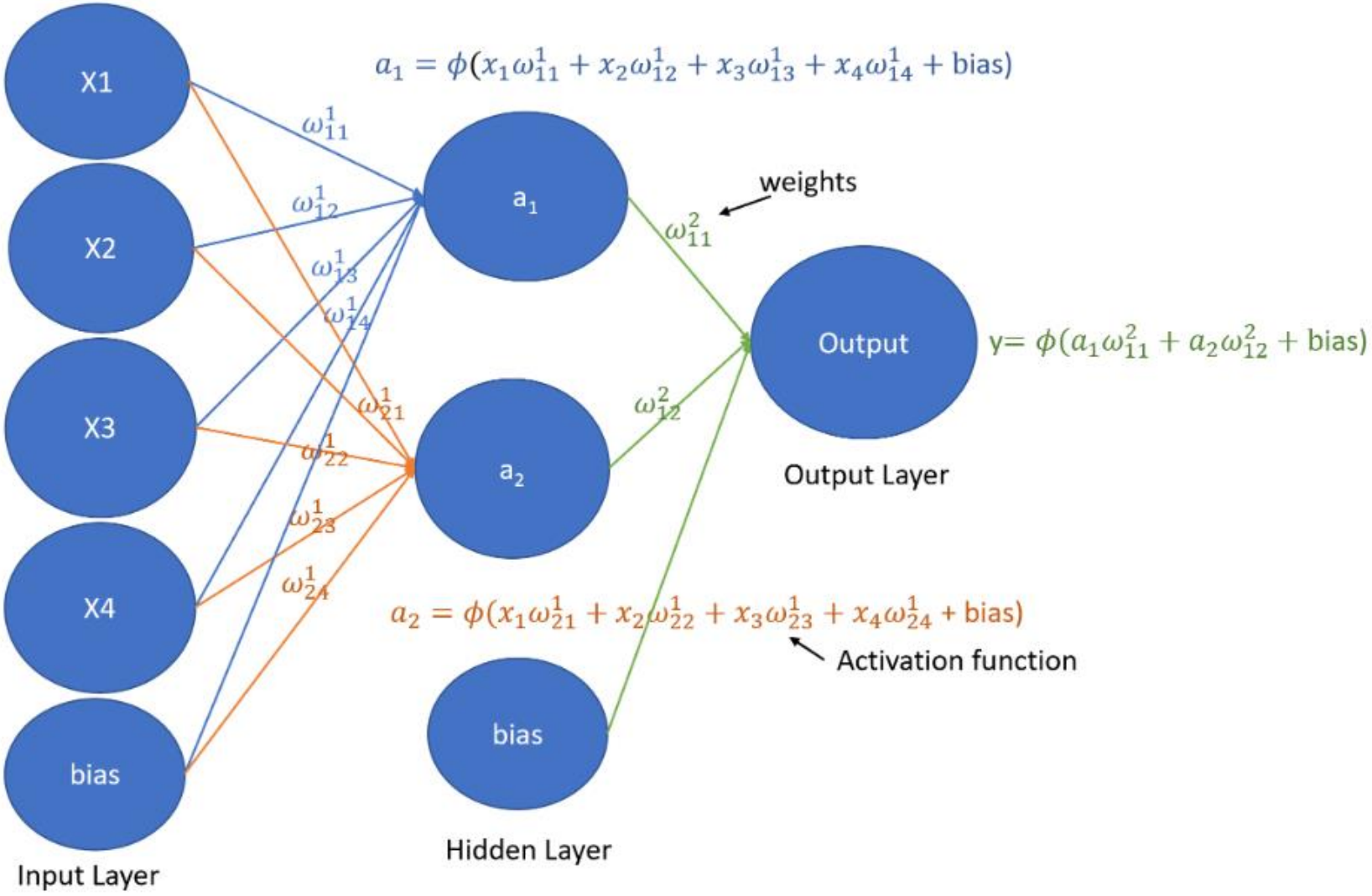
Optimization Algorithms



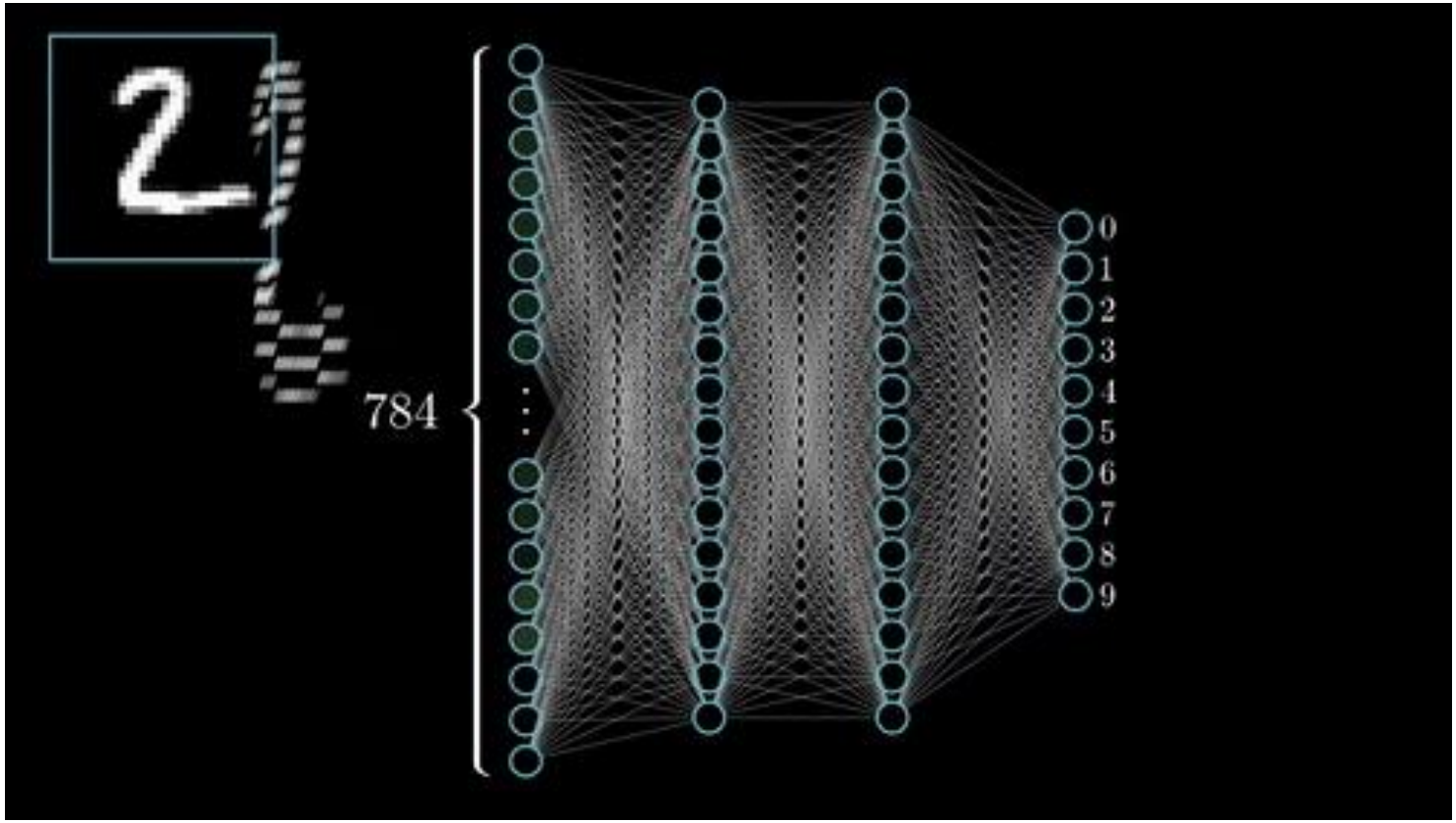
Optimization Algorithms

Method	Update equation
	$g_t = \nabla_{\theta_t} J(\theta_t)$
SGD	$\Delta\theta_t = -\eta \cdot g_t$
	$\theta_t = \theta_t + \Delta\theta_t$
Momentum	$\Delta\theta_t = -\gamma\, v_{t-1} - \eta g_t$
NAG	$\Delta\theta_t = -\gamma\, v_{t-1} - \eta \nabla_\theta J(\theta - \gamma v_{t-1})$
Adagrad	$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$
Adadelta	$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$
RMSprop	$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$
Adam	$\Delta\theta_t = -\frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$

Summarizing the MATH



Prediction



Neural Net for Spam classification

```
spam  FreeMsg Hey there darling it's been 3 week's n...
 ham  Even my brother is not like to speak with me. ...
 ham  As per your request 'Melle Melle (Oru Minnamin...
spam  WINNER!! As a valued network customer you have...
spam  Had your mobile 11 months or more? U R entitle...
 ham  I'm gonna be home soon and i don't want to tal...
spam  SIX chances to win CASH! From 100 to 20,000 po...
spam  URGENT! You have won a 1 week FREE membership ...
 ham  I've been searching for the right words to tha...
 ham                  I HAVE A DATE ON SUNDAY WITH WILL!!
spam  XXXMobileMovieClub: To use your credit, click ...
 ham                           Oh k...i'm watching here:)
 ham  Eh u remember how 2 spell his name... Yes i di...
 ham  Fine if thatåÕs the way u feel. ThatåÕs the wa...
spam  England v Macedonia - dont miss the goals/team...
 ham        Is that seriously how you spell his name?
```

Neural Net for Spam classification

Preprocessing data

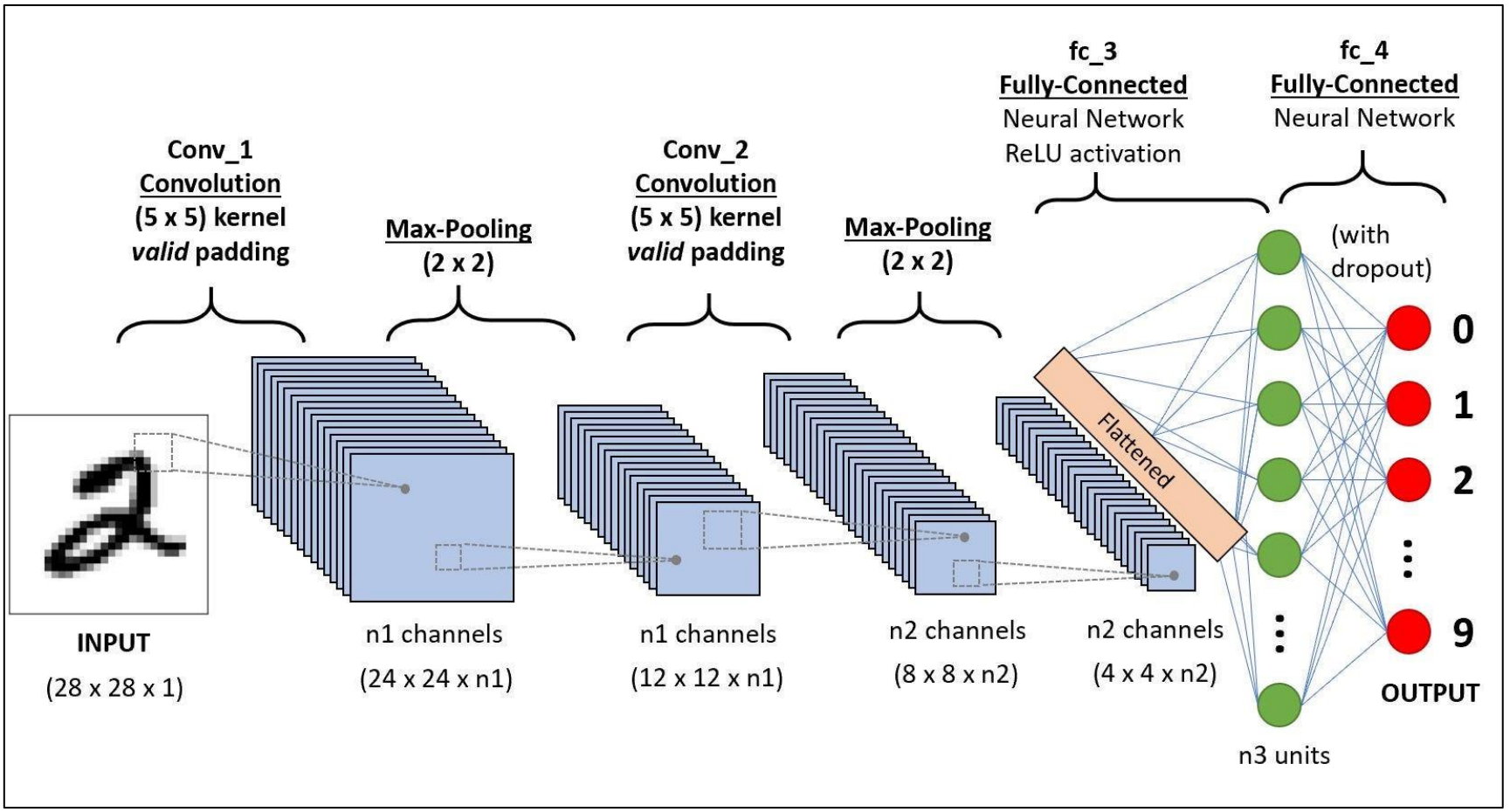
Neural Net for Spam classification

Demo 4: Neural Net for Spam classification

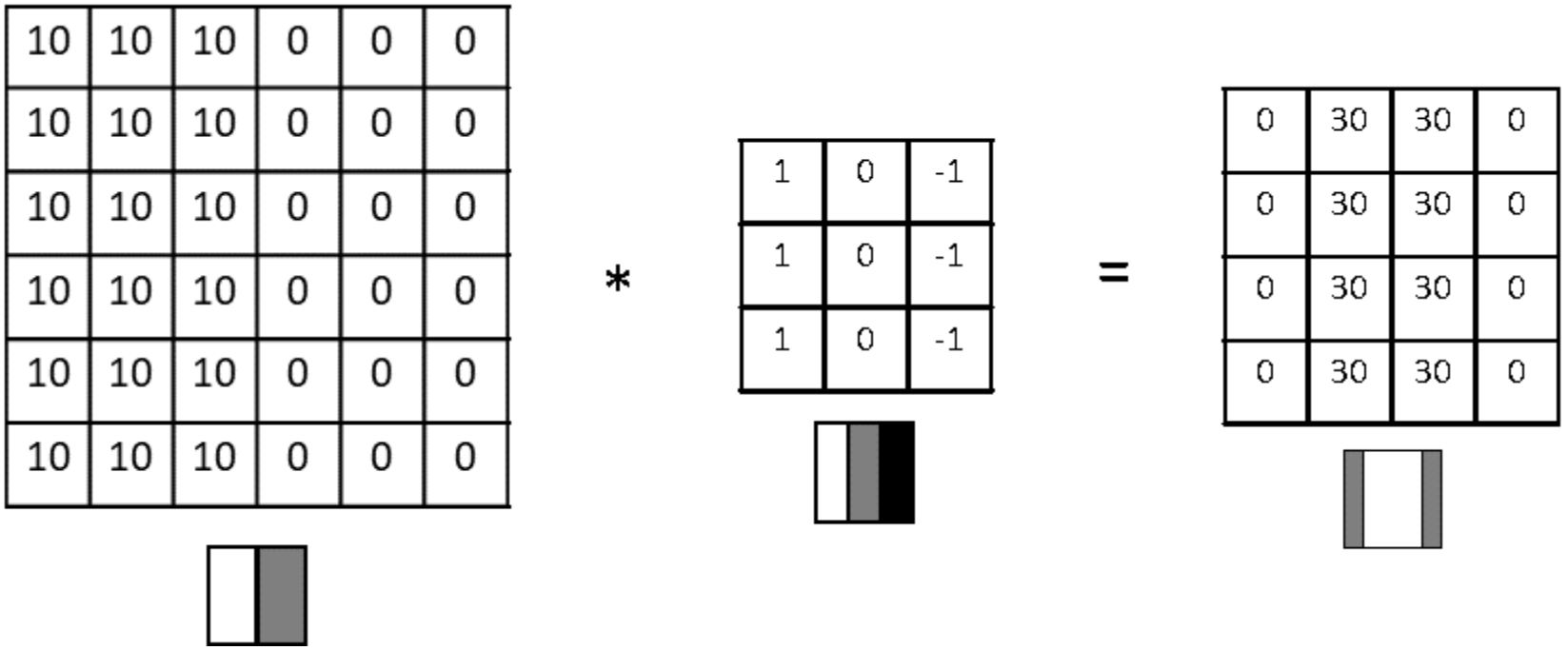
<NN for spam classification End>

Convolutional Neural Net

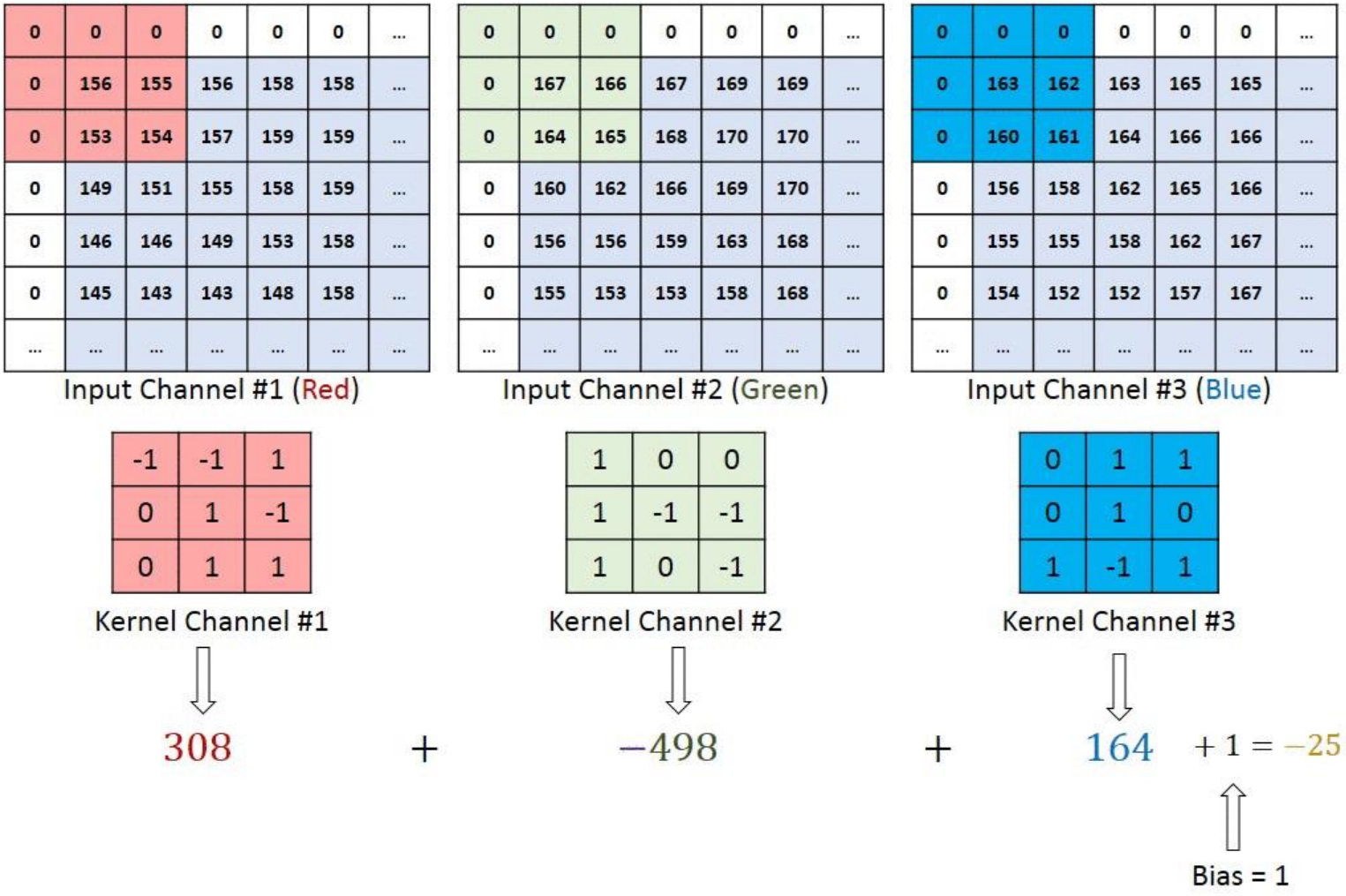
Convolution layer | Pooling layer | Dense layer



Convolutional Neural Net - Convolution Layer



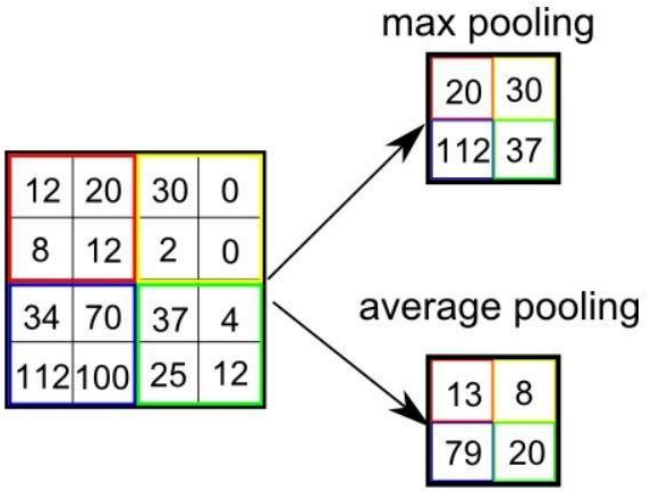
Convolutional Neural Net - Convolution Layer



Convolutional Neural Net - Pooling Layer

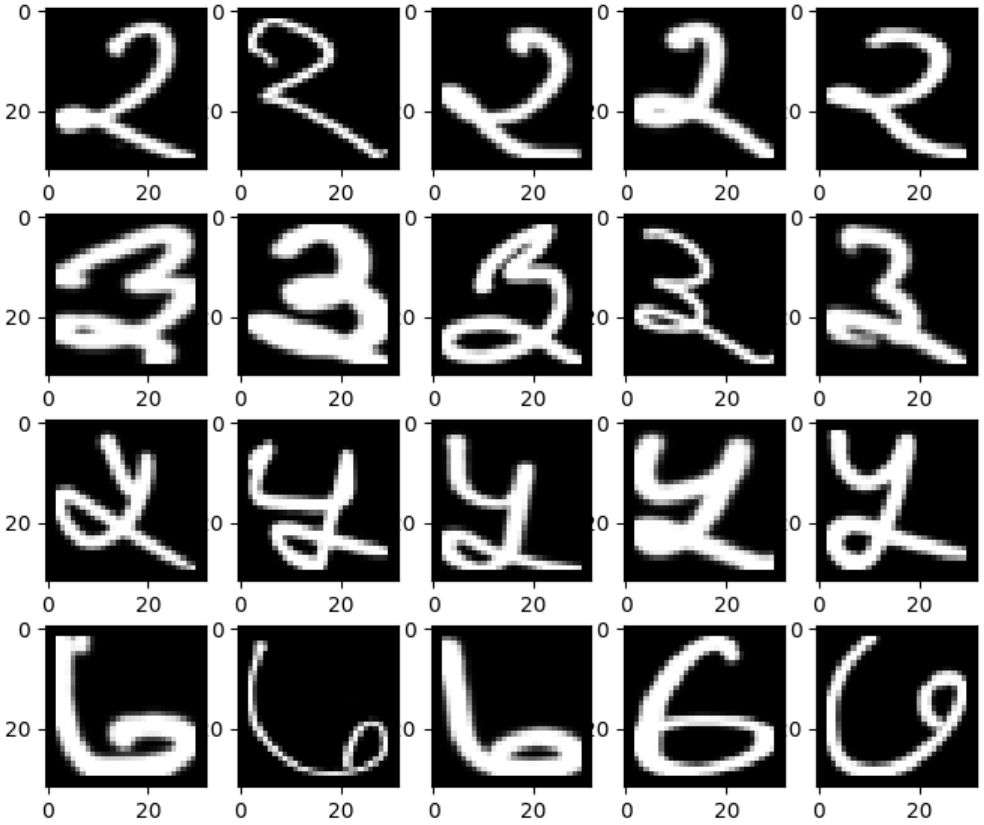
3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1



Modified National Institute of Standards and Technology database

CNN for Devanagari digit classification



About Dataset

- Grayscale image of size (32x32)
- There are 1025 columns and 1200 rows.
- Columns 1-1024 : contains the pixel values in range 0 to 255.
- column 1025: Label

Demo 5: CNN for Devanagari digit classification

Phishing

- Fraudulent attempt to obtain sensitive/personal information
- Multiple vectors
- Mails, DMs, SMS, Social network, chat forums, etc.
- Spear Phishing

Spear Phishing: The problem

- Targeted
- Understand interests of target
- Takes hours of manual efforts
- Higher success rate

Spear Phishing: How Deep Learning helps

- Build a huge corpus of text from target's social media feeds
- Train a model to generate similar text using words from corpus

ML based Spear Phishing: Results

[+]20: Much awaited talk given by @imohitch at @PydataPune @PythonPune for AI"<link>

[+]12: Thanks!! <link> CFP deadline: Jun 20 - 12 #PyDataPune <link>

[+]5: It was a comment. <link>
#python #pydatapune #HouseOfResearch @RedHat. Worked on xarray - R and action plan.
@RedHat #MachineLearning #DataScience #PyDataPuneMeetups

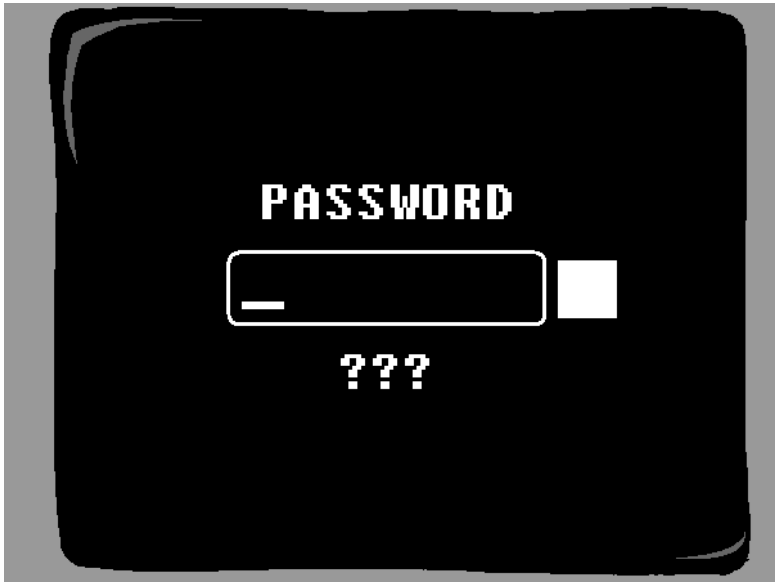
[+]9: Yo! I find the Newsletter.
@PydataPune @PyData @NumFOCUSpic.twitter.com/T5ydVfF5PR

[+]13: <link> May 19, 2018. We are PyDaters.
Thanks @katyhuff
@PyData @NumFOCUS #python #pune #machinelearning #dataviz #openscience #nonprofit #swagpic.twitter.com/j9JhQy9Pus

[+]7: Interested in today’s @WiDSPune2019 on numpy and show your Attention please!!!
The goal is going @pranshutople14 conclude the maths behind NeuralNets <link>
#PyData #NumFOCUS @PyData #PyData_Meetup @Pune @machinelearnflx @Pune

Dictionary attacks

- Dictionary contains commonly used password
- Try every password from dictionary unless you succeed
- Attack may work if the password is similar to
  - Pydata_123$$
  - ##pydata##Pun3
  - PyData@@098



Dictionary attacks: data collection

1.4 Billion Clear Text Credentials Discovered in a Single Database

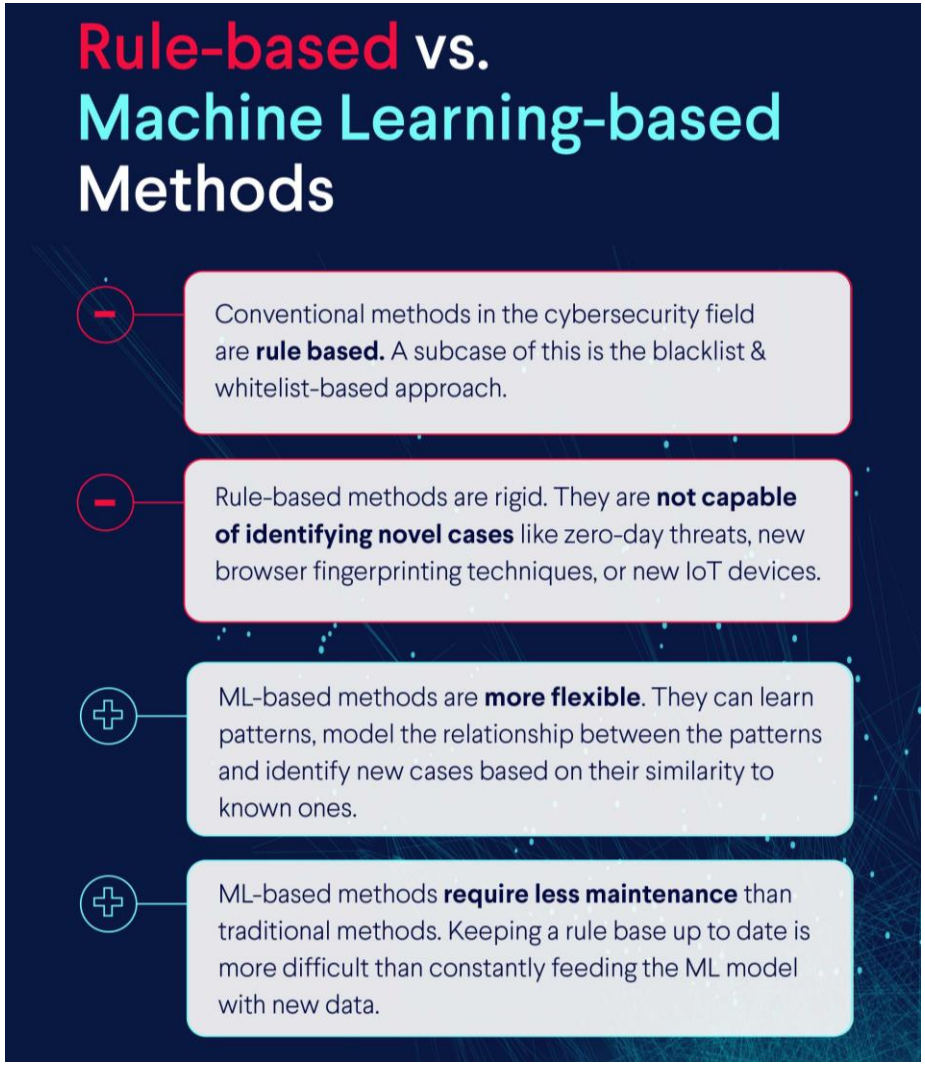
Julio Casal Follow

Dec 9, 2017 · 5 min read

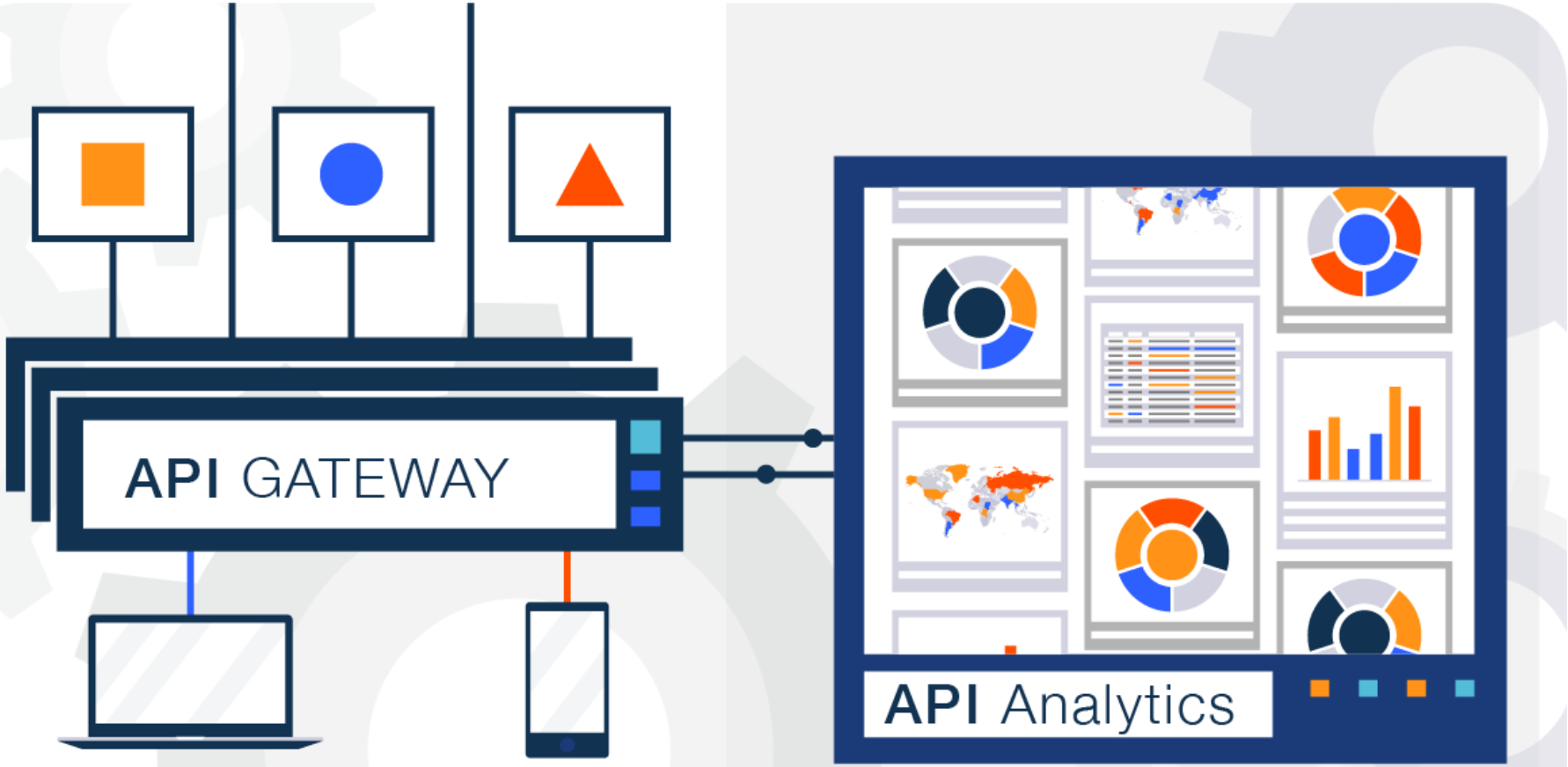
Dictionary attacks: Approach

- Skeletonize the passwords
  - <first_name>_<last_name>##123
  - <dob><mob><First_name>
  - <dob><mob>_<Lastname>$$
  - <last_name>##<random_num>
  - Etc ...
- Train generative models on corpus of skeletons
- Basic personal info + skeletons = passwords

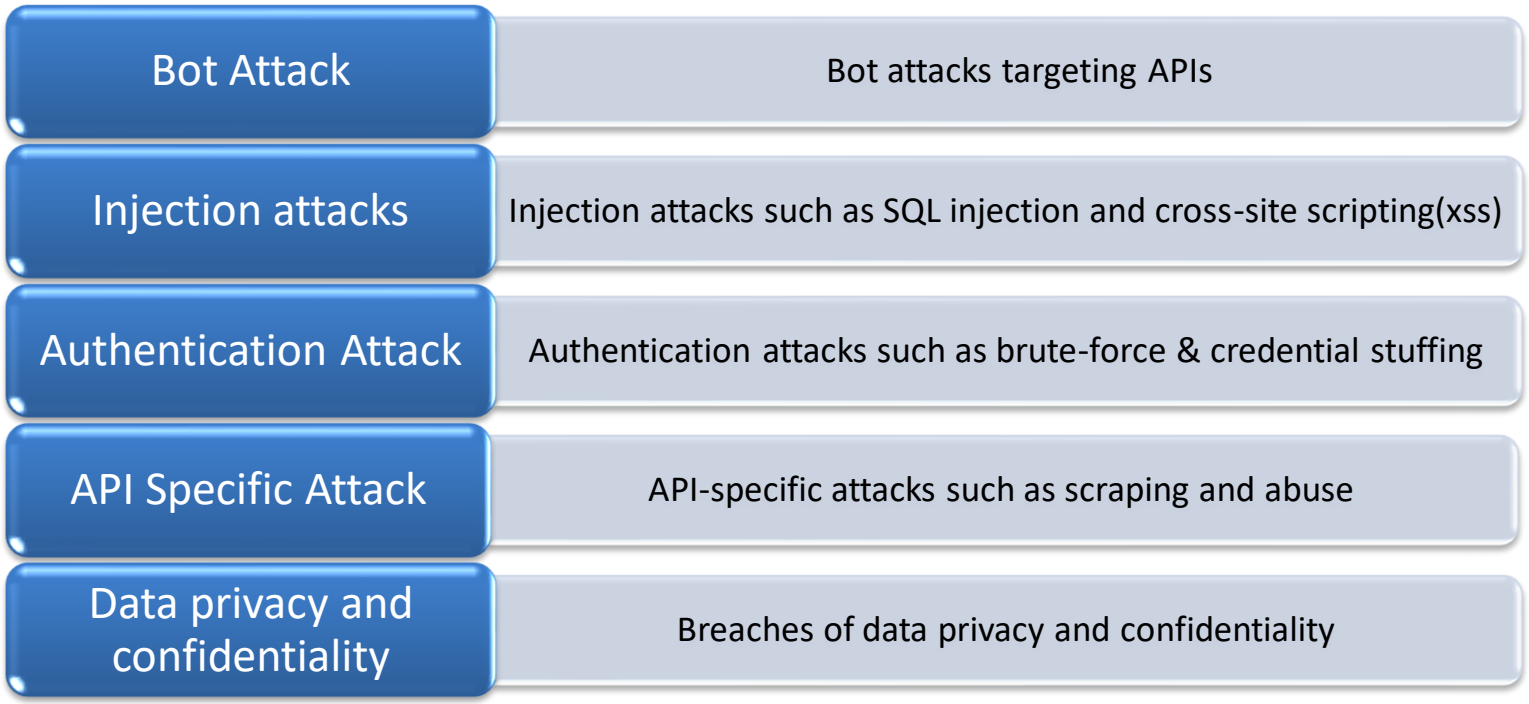
Traditional API Security



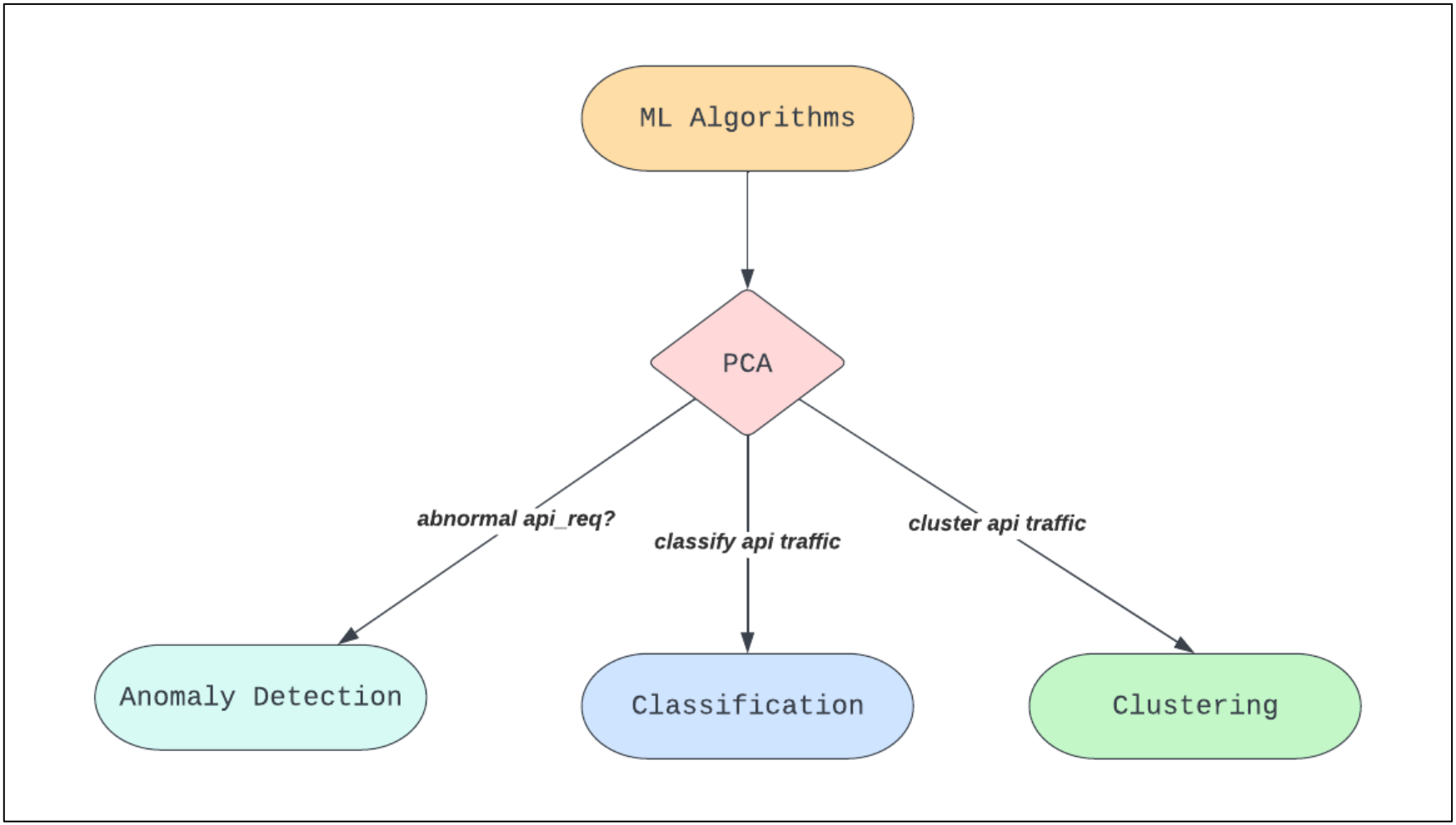
Adding ML to API Security



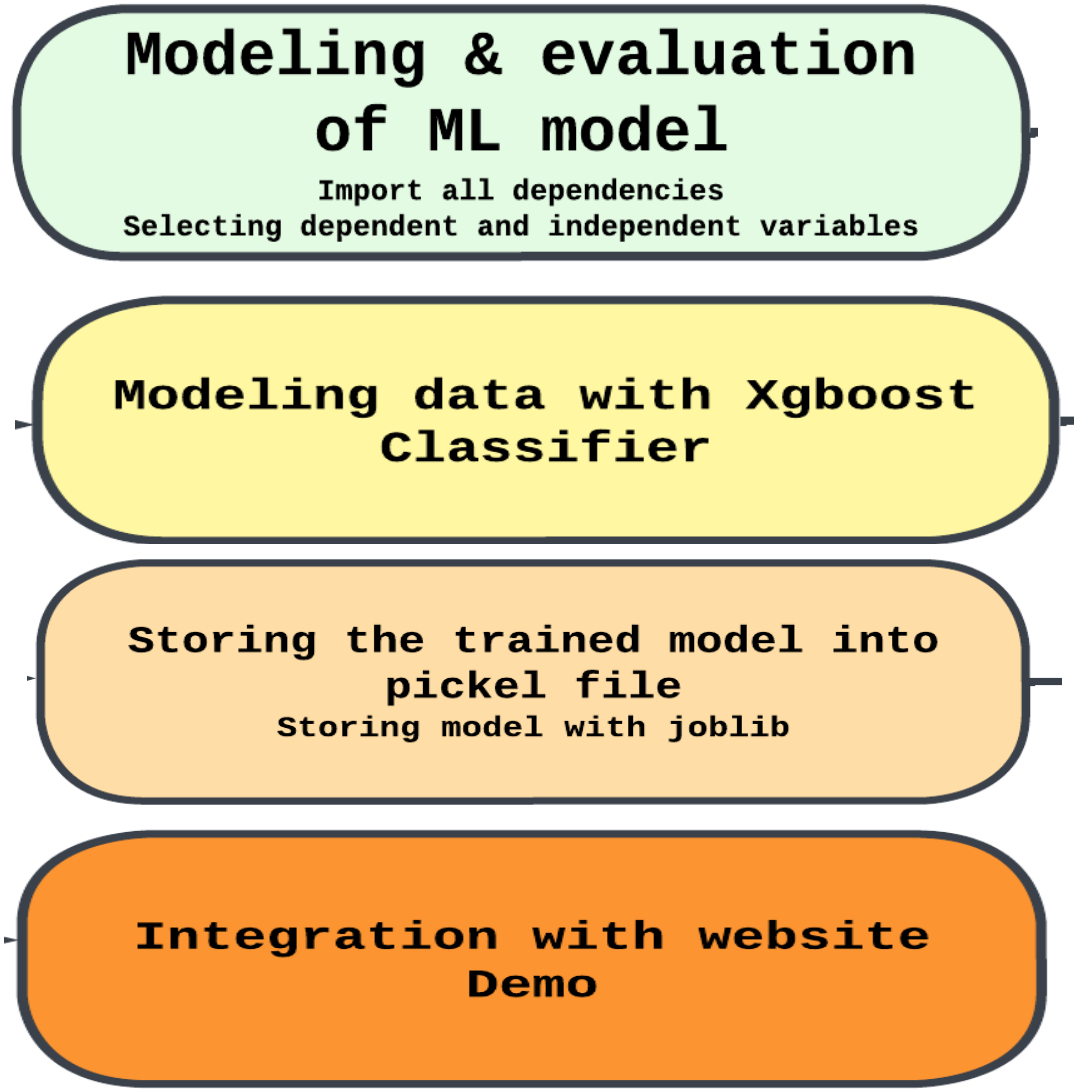
Type of API Security Threat



Applying ML to API Security



DEMO: Before the break



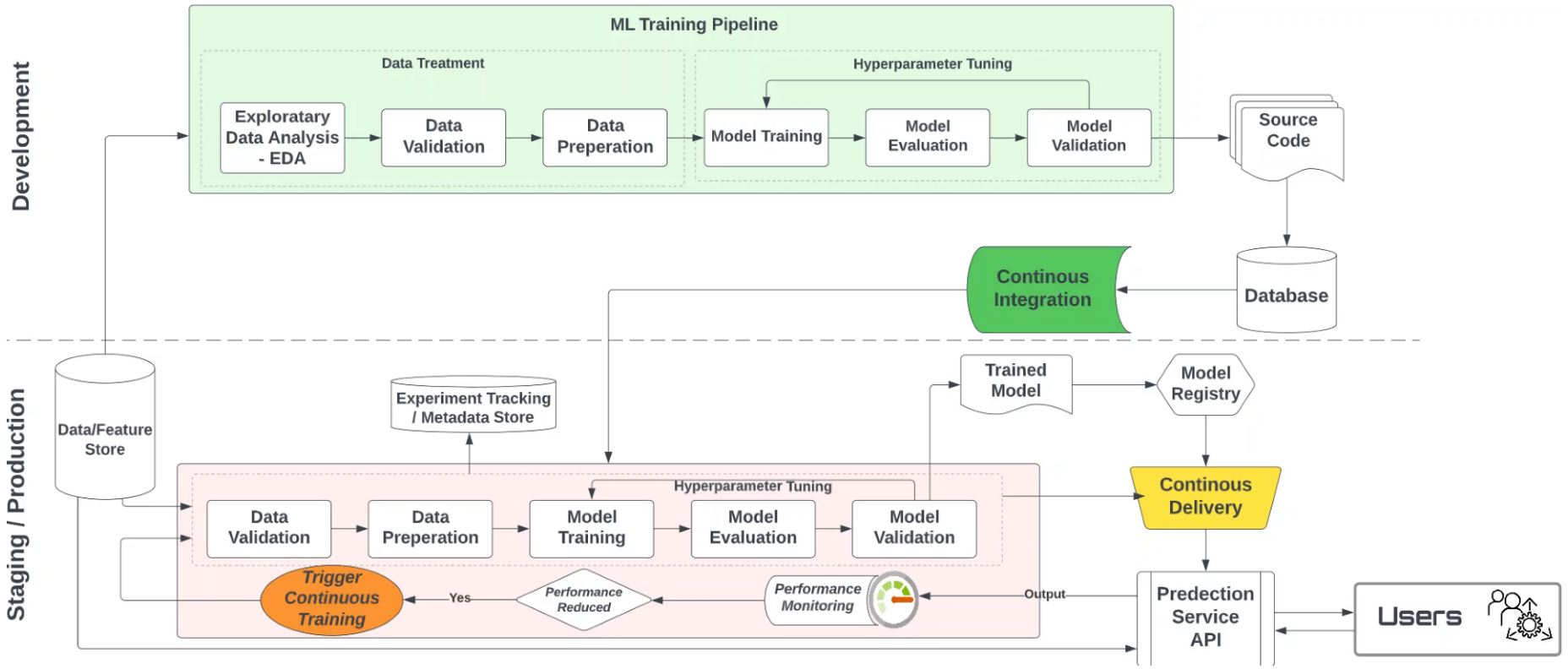
LINK

<Interval>

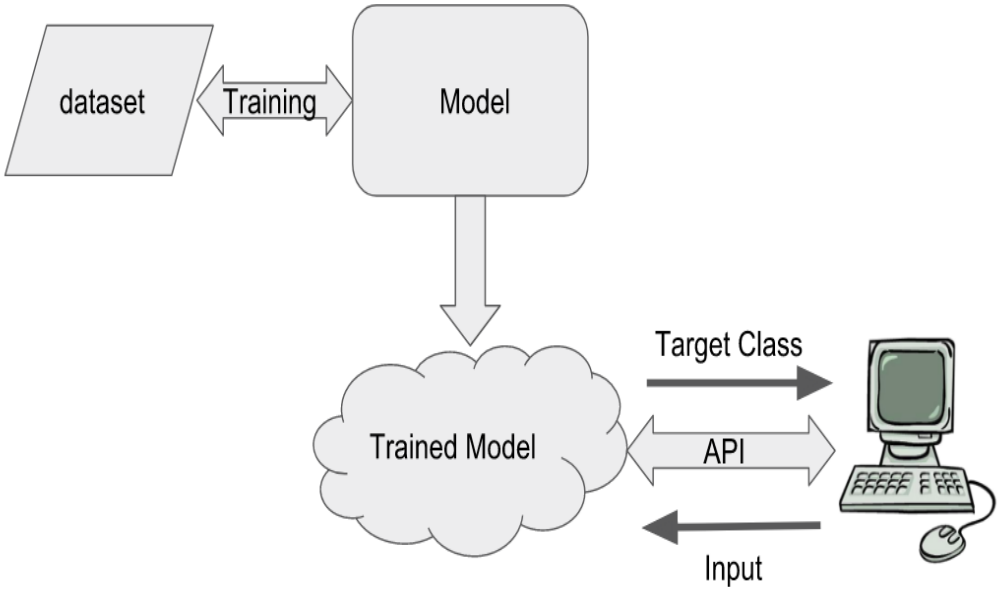
Part two: Security for ML

ML in Production

"As MlOps hits maturity, time to consider security"



Possible Attack Targets: Almost Everything



- Poison Dataset
- Skewed Models
- Adversarial Attack
- Steal Models
- I/P Data inference
- Framework Exploits
- Network attacks

AI Security Assessment

- Recon
- Attack surface identification
- Threat modeling
- Adversarial Learning Attack
- Model Stealing
- Model Skewing
- Input data inference
- Model inversion
- Hybrid attacks

The Recon

- Intel on Application and Model internals
- Research Papers, Blogs and product pages
- Conference talks, demos, social media content for target application/appliance
- Helps to plan efficient attack scenarios
- Get your hands on target application

Adversarial Learning

Is our model actually "learning" what we expect it to learn?

Adversarial Learning

Specially craft an input with an intention to produce desired prediction from target model.

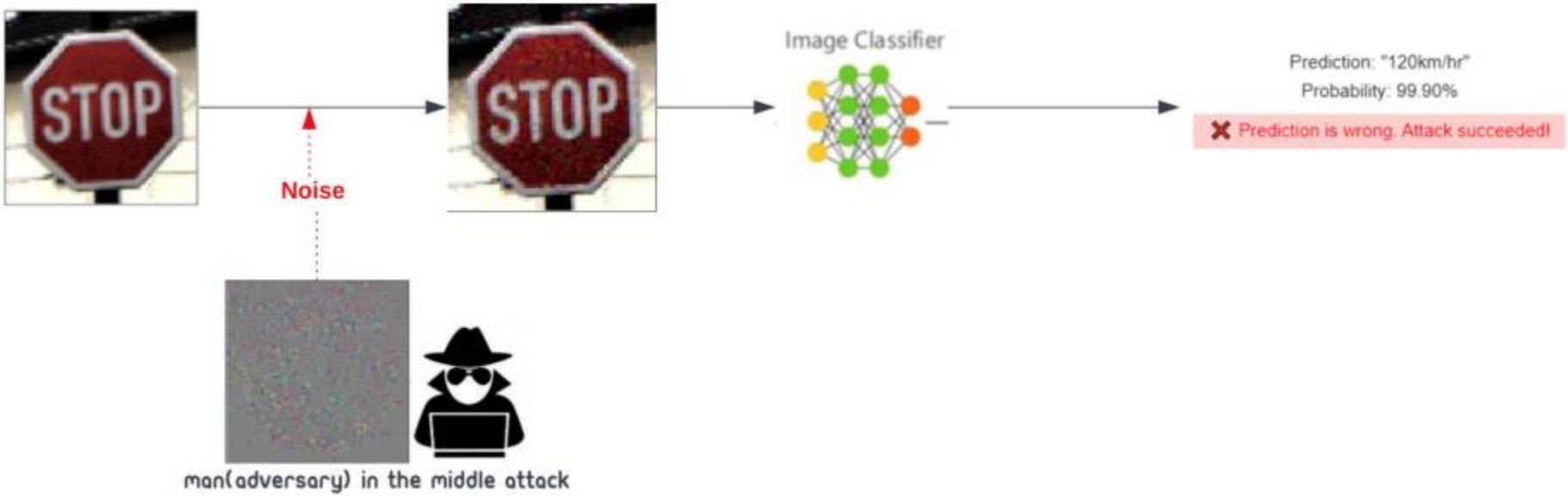
Street Sign

Perturbations

Mail Box

Adversarial Learning

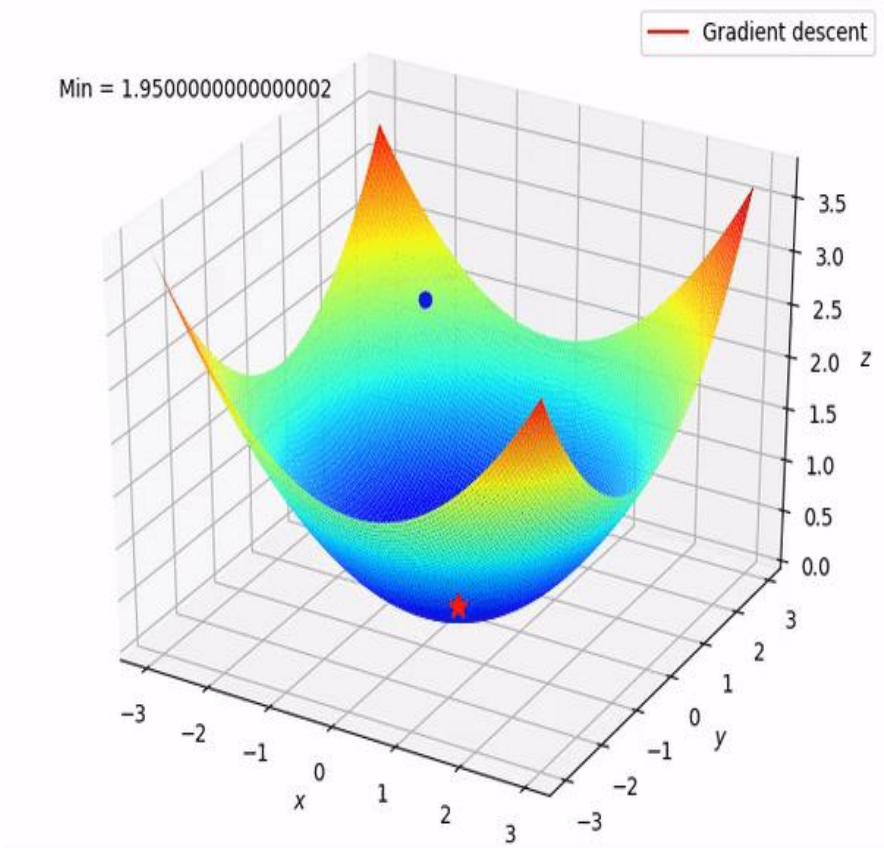
Demo 6: Hack the ML



https://hacktheml.web.app

Adversarial Learning

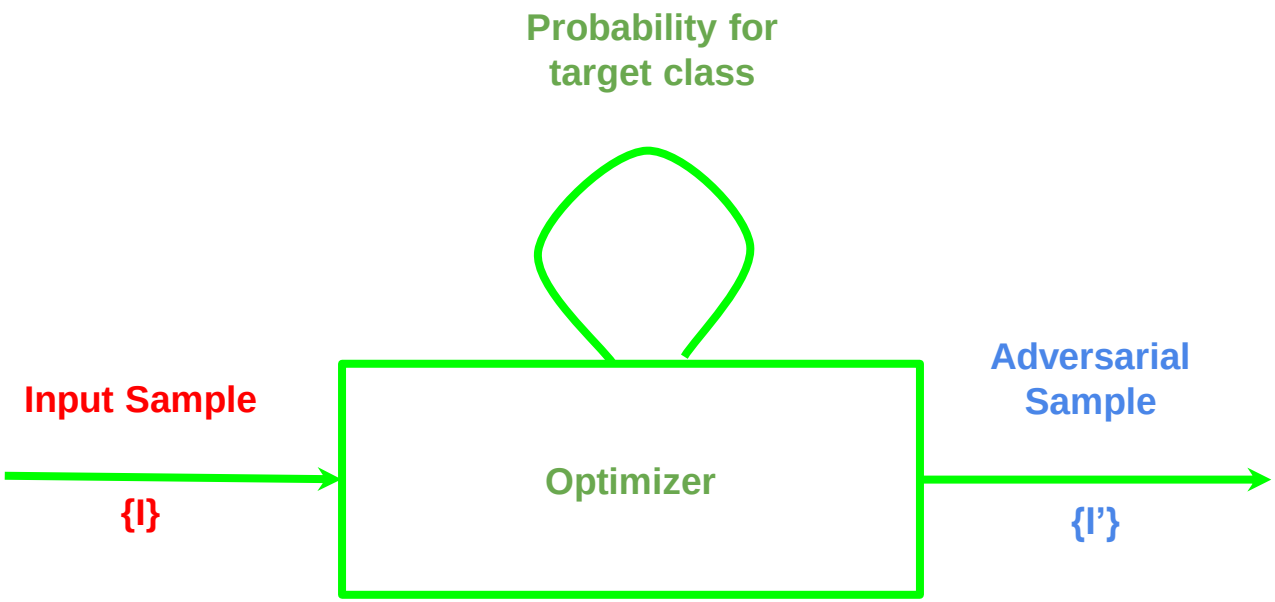
Adversarial attack as an optimization problem



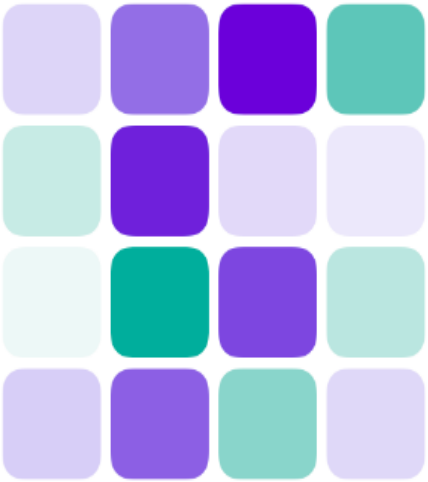
$$I_{i+1} = I_i + \alpha \frac{\partial}{\partial I} C$$

Adversarial Learning Attacks

Specially craft an input with an intention to produce desired prediction from target model.



Adversarial Learning Attacks



Foolbox

Ref: https://github.com/bethgelab/foolbox

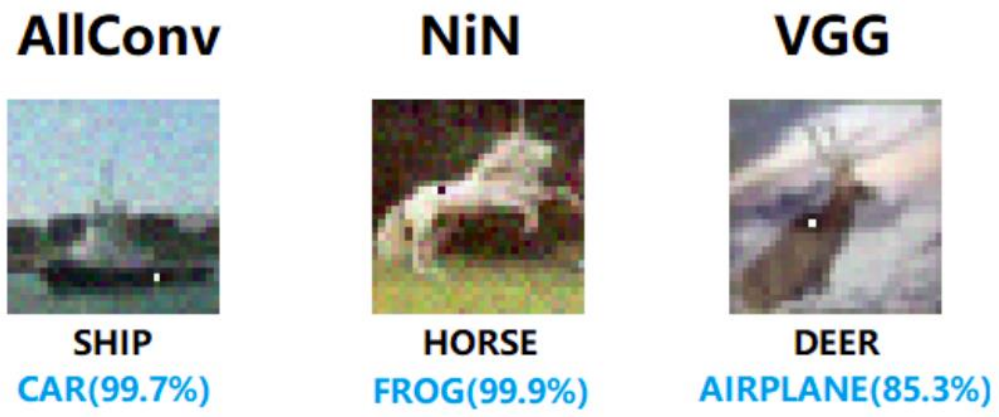
Cleverhans

Ref: https://github.com/cleverhans-lab/cleverhans

Experimental Adversarial attacks

One Pixel Attack for Fooling Deep Neural Networks

Jiawei Su*, Danilo Vasconcellos Vargas* and Kouichi Sakurai



Experimental Adversarial attacks

A General Framework for Adversarial Examples with Objectives

Fig. 5. An example of digital dodging. Left: An image of actor Owen Wilson (from the PubFig dataset [40]), correctly classified by VGG143 with probability 1.00. Right: Dodging against VGG143 using AGN's output (probability assigned to the correct class <0.01).

Realworld Attacks

How do we deal with copying content, or the first adversarial attack in prod

Avito company blog , Programming , algorithms , Image processing , Machine learning

Realworld Attacks

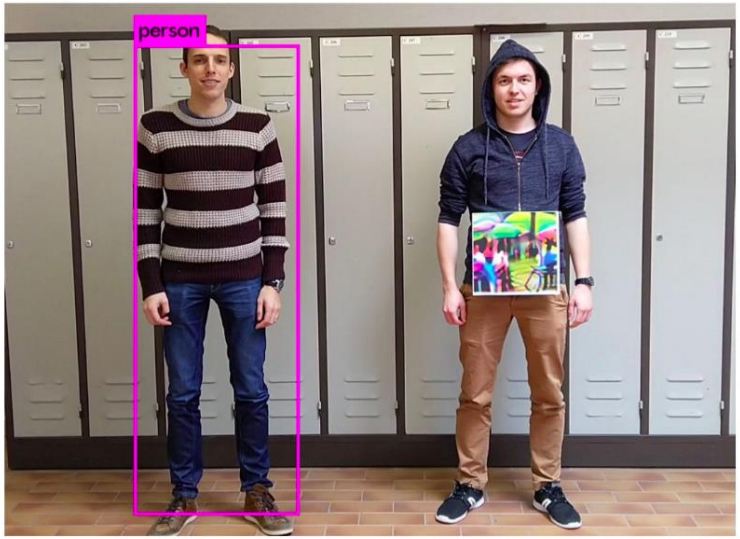
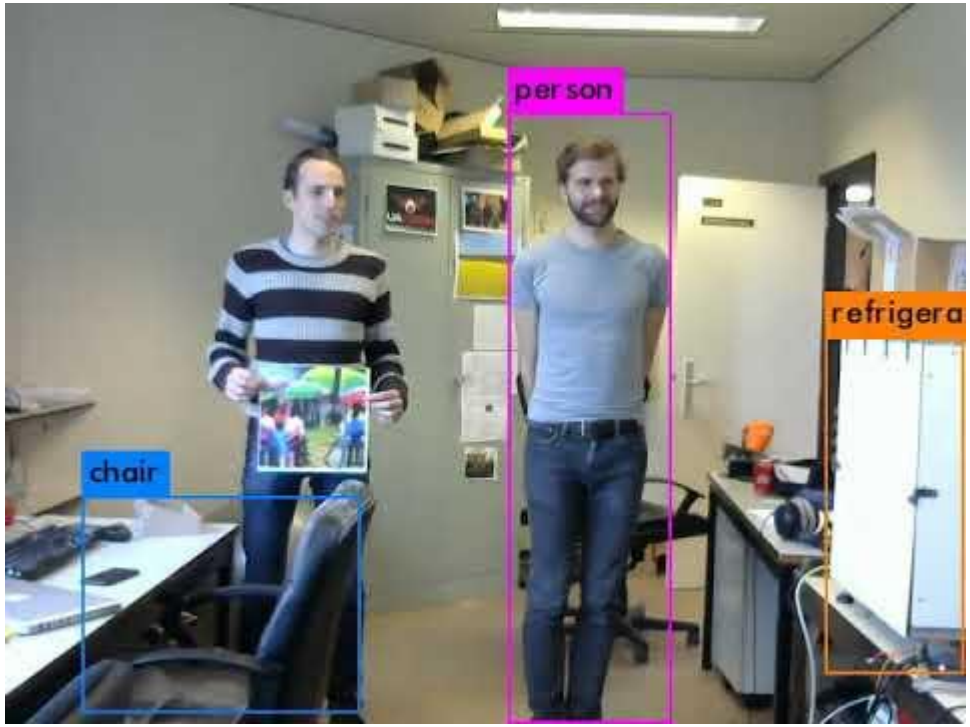
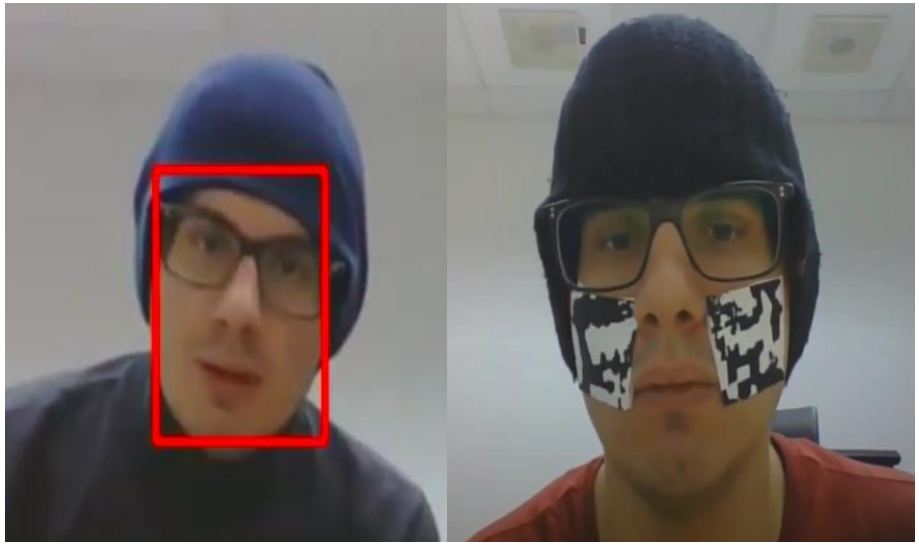


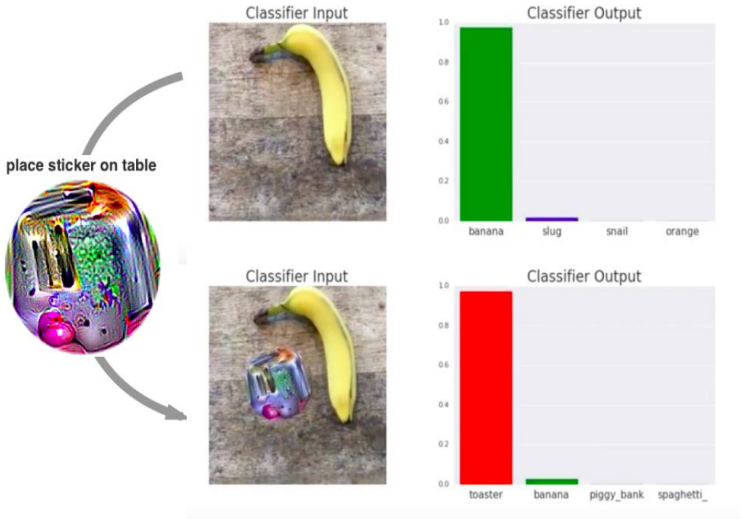
Figure 1: We create an adversarial patch that is successfully able to hide persons from a person detector. Left: The person without a patch is successfully detected. Right: The person holding the patch is ignored.



Adversarial Attacks in Physical Domain

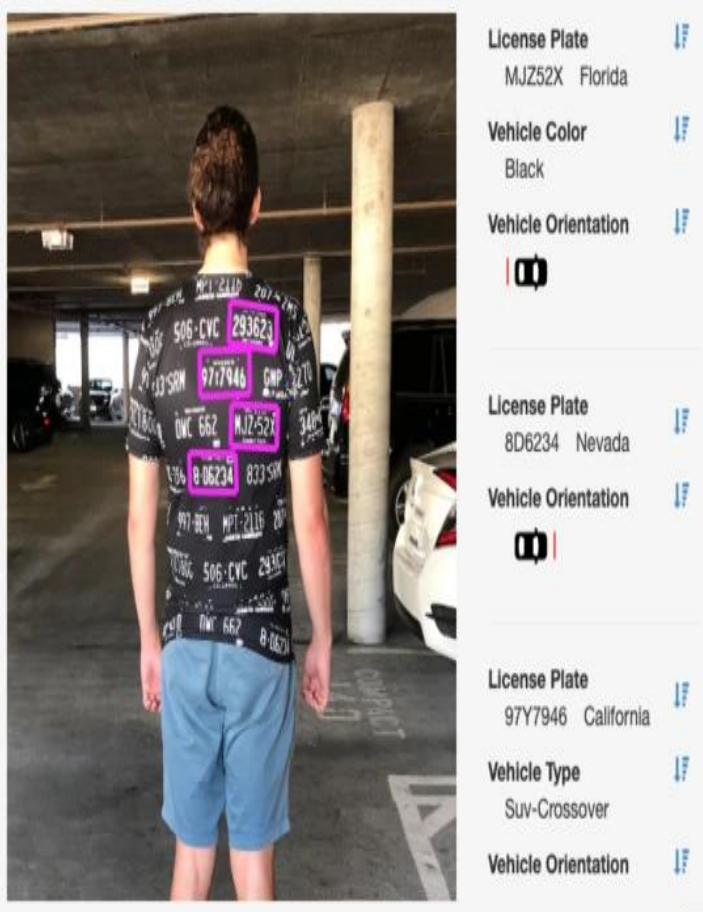


https://arxiv.org/pdf/1910.06261.pdf



https://arxiv.org/pdf/1712.09665.pdf

Mis-classification attacks Without any Fancy Maths



https://adversarialfashion.com/

Distorted Inputs

Mis-classification attacks Without any Fancy Maths

Attack text label iPod

Granny Smith	85.6%
iPod	0.4%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.1%

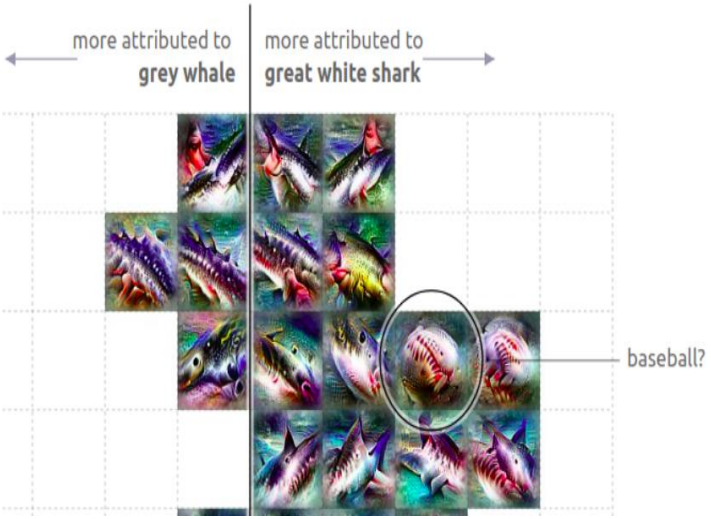
Granny Smith	0.1%
iPod	99.7%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.0%

piggy bank	52.5%
Standard Poodle	23.8%
Miniature Poodle	2.3%
Pyrenean Mountain Dog	1.1%
military cap	0.7%
Chow Chow	0.7%

piggy bank	35.9%
horse chestnut seed	13.6%
acorn	5.0%
abacus	2.6%
pretzel	2.3%
golf ball	1.9%

https://openai.com/blog/multimodal-neurons/

Adversarial attacks Without any Fancy Maths



1.	grey whale	91.0%
2.	killer whale	7.5%

1.	great white shark	66.7%
2.	baseball	7.4%

https://distill.pub/2019/activation-atlas/

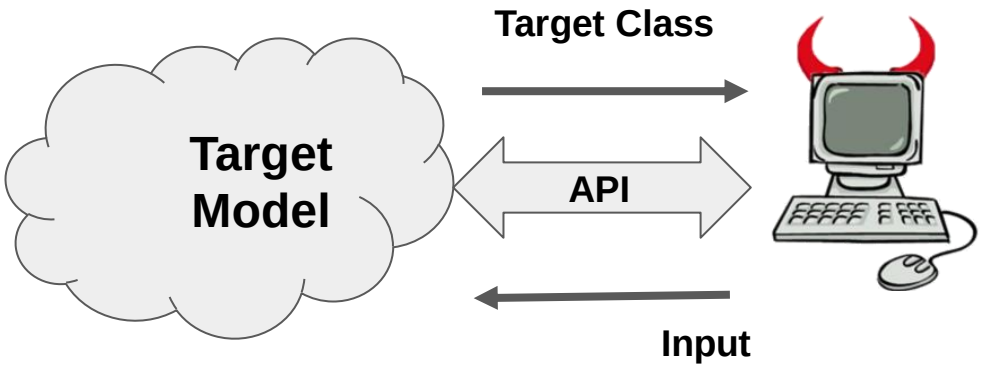
The patient has a history of back pain and chronic alcohol abuse and more recently has been seen in several...

Adversarial text substitution

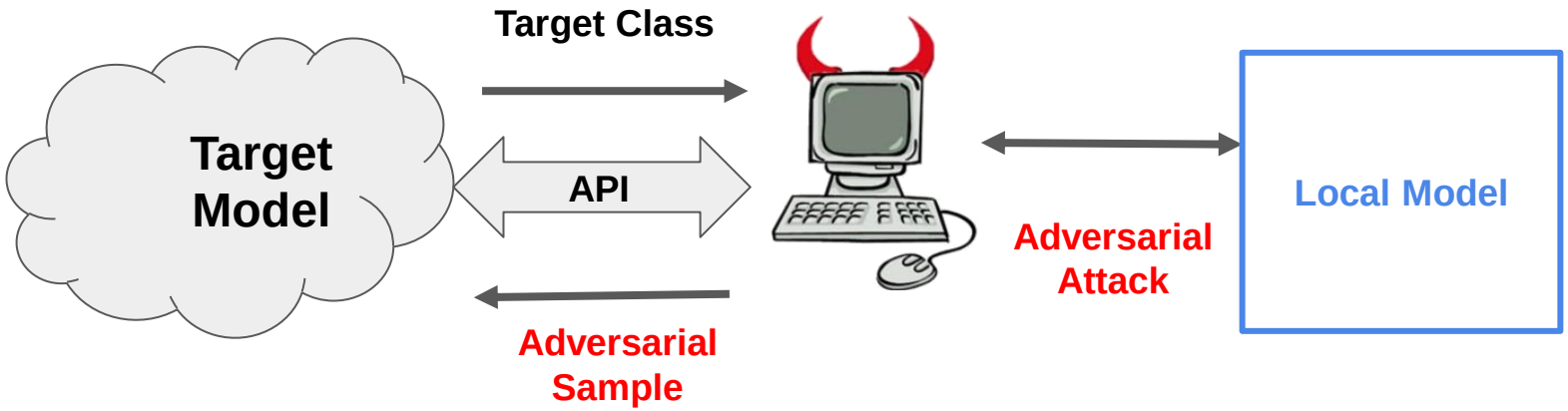
The patient has a history of lumbago and chronic alcohol dependence and more recently has been seen in several...

https://science.sciencemag.org/content/363/6433/1287.full?ijkey=OXnSsEp.Iagl6&keytype=ref&siteid=sci

Adversarial Attacks on Black-Box Models

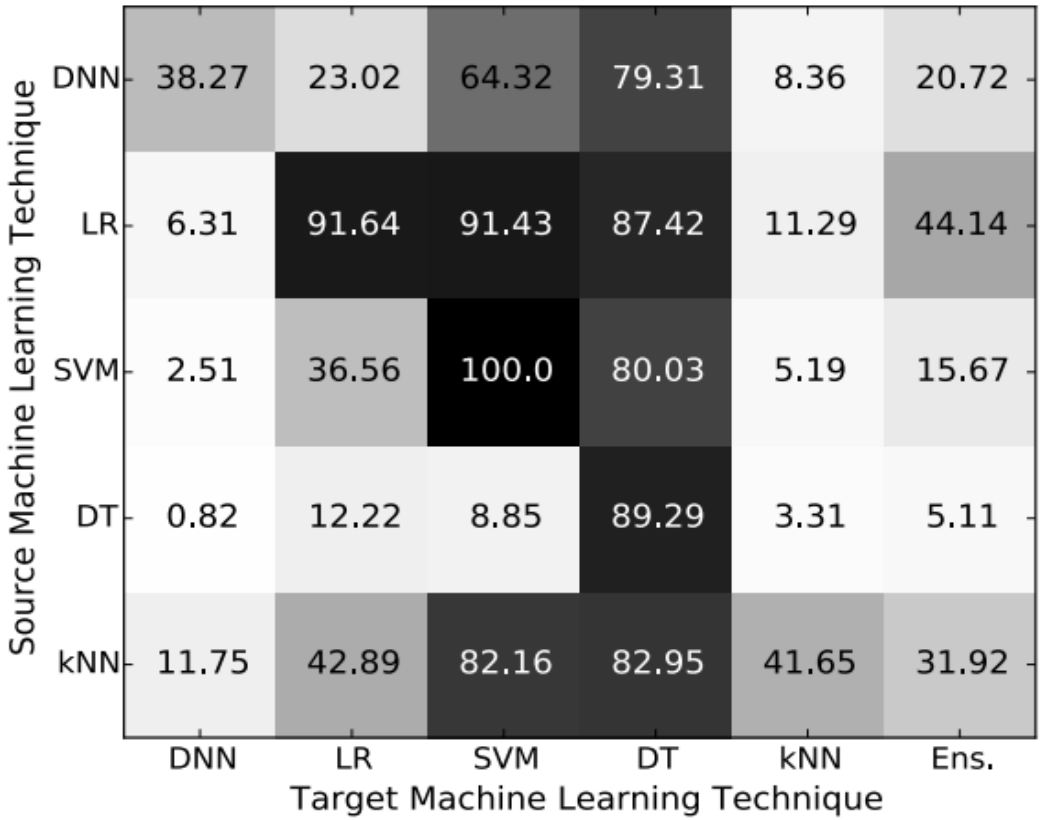


Adversarial Attacks on Black-Box Models



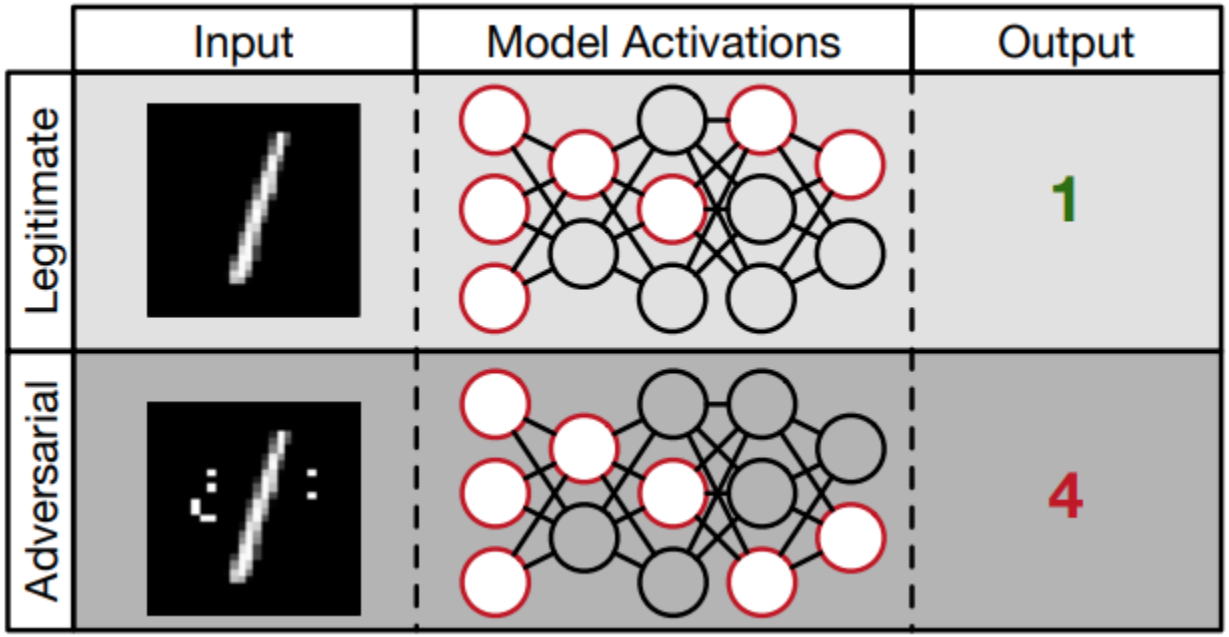
Adversarial Attacks on Black-Box Models

Cross-technique Transferability



Ref: https://arxiv.org/pdf/1605.07277.pdf

Adversarial Attacks on Black-Box Models



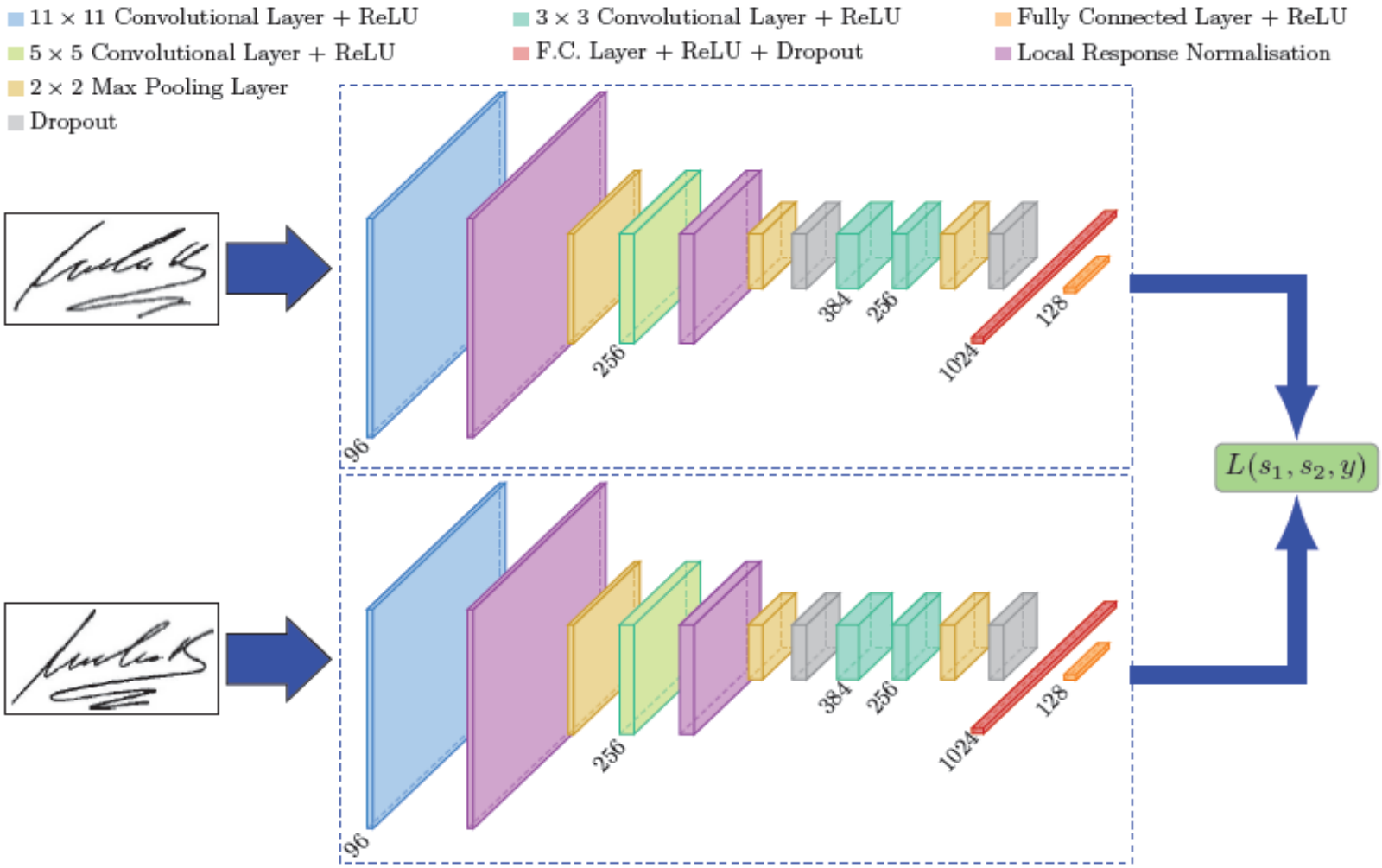
Ref: https://arxiv.org/pdf/1605.07277.pdf

Demo 7: Adversarial Attacks

Mitigation: Adversarial Attacks

Adversarial Attack on FaceNet

Siamese Neural Network



https://arxiv.org/pdf/1503.03832.pdf

FaceNet Architecture

layer	size-in	size-out	kernel	param
conv1	220×220×3	110×110×64	7×7×3, 2	9K
pool1	110×110×64	55×55×64	3×3×64, 2	0
rnorm1	55×55×64	55×55×64		0
conv2a	55×55×64	55×55×64	1×1×64, 1	4K
conv2	55×55×64	55×55×192	3×3×64, 1	111K
rnorm2	55×55×192	55×55×192		0
pool2	55×55×192	28×28×192	3×3×192, 2	0
conv3a	28×28×192	28×28×192	1×1×192, 1	37K
conv3	28×28×192	28×28×384	3×3×192, 1	664K
pool3	28×28×384	14×14×384	3×3×384, 2	0
conv4a	14×14×384	14×14×384	1×1×384, 1	148K
conv4	14×14×384	14×14×256	3×3×384, 1	885K
conv5a	14×14×256	14×14×256	1×1×256, 1	66K
conv5	14×14×256	14×14×256	3×3×256, 1	590K
conv6a	14×14×256	14×14×256	1×1×256, 1	66K
conv6	14×14×256	14×14×256	3×3×256, 1	590K
pool4	14×14×256	7×7×256	3×3×256, 2	0
concat	7×7×256	7×7×256		0
fc1	7×7×256	1×32×128	maxout p=2	103M
fc2	1×32×128	1×32×128	maxout p=2	34M
fc7128	1×32×128	1×1×128		524K
L2	1×1×128	1×1×128		0

https://arxiv.org/pdf/1503.03832.pdf

Demo 8 : Adversarial Attack on FaceNet

Model Stealing Attacks

Offline Attacks

Stealing models that are deployed on device

Online Attacks

Stealing models deployed on cloud with black box access

Offline Model Stealing Attacks

- Applications like WAF, Object detection, Antiviruses, IDS/IPS etc. require to work without internet connectivity
- Models are deployed on device to make them available for prediction
- Deployment may or may not employ any protection mechanism

Offline Model Stealing Attacks

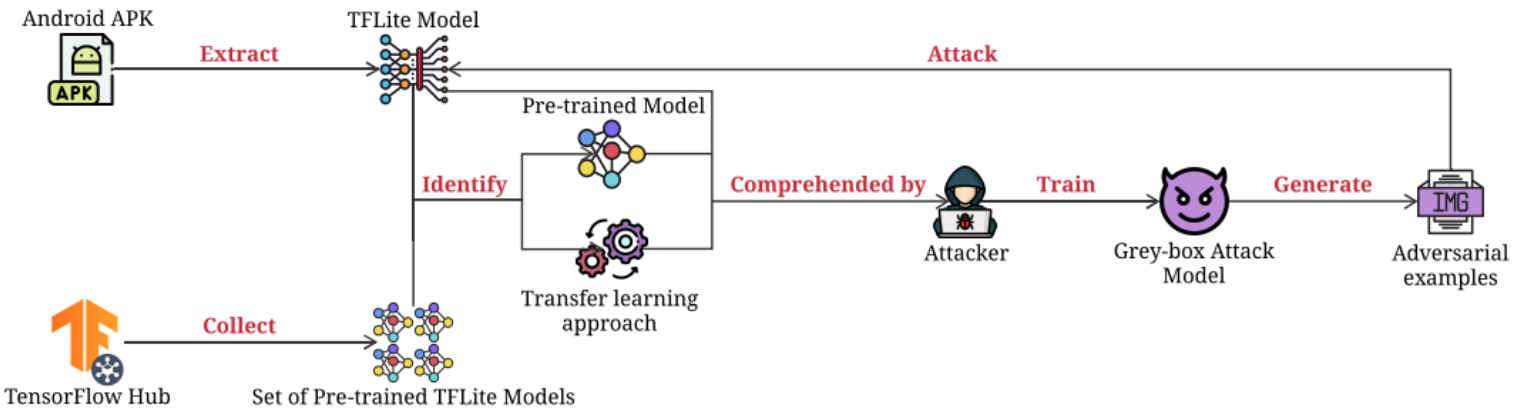
- Model could be deployed on device
- Extracting from
  - firmware/flash storage
  - APK, executables
  - Disk
  - Transmitted over Network
- Analyze the serialized model
- Leverage recon to understand internals (like framework, algorithm, architecture, etc) of the model
- Get predictions from model

Offline Model Stealing Attacks

- Model could be deployed on device
- Extracting from
  - firmware/flash storage
  - APK, executables
  - Disk
  - Transmitted over Network
- Analyse the serialized model
- Leverage recon to understand internals (like framework, algorithm, architecture, etc) of the model
- Get predictions from model

Offline Model Stealing Attacks

Demo 9 : Offline Model Stealing Attack [H/W]



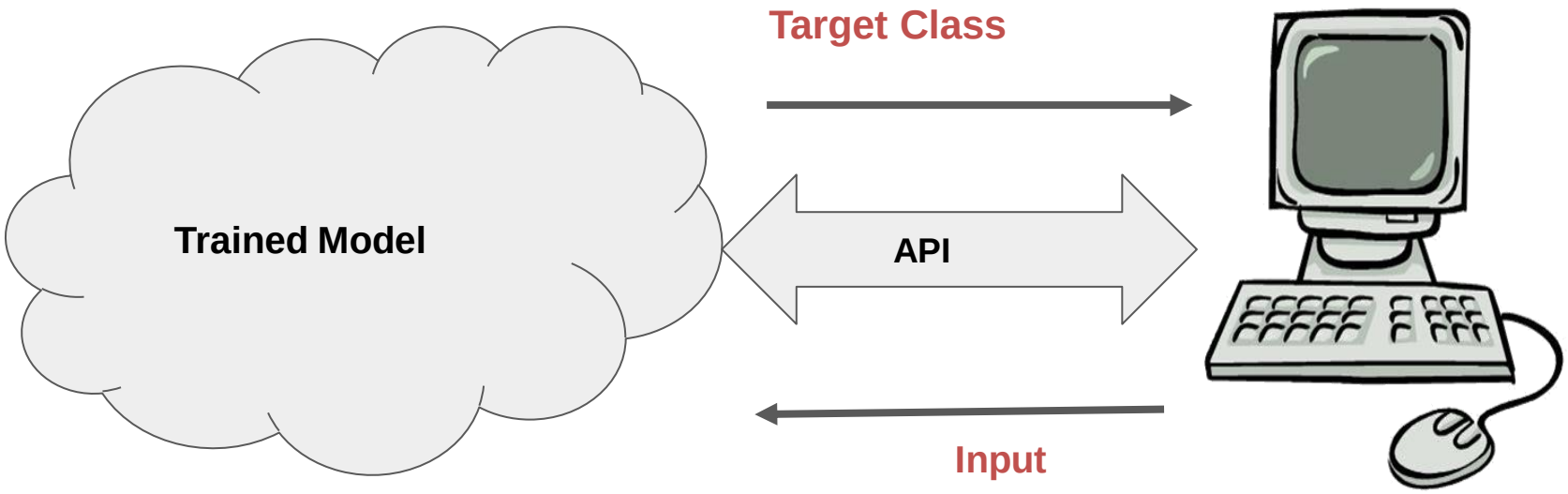
Offline Model Stealing Attacks

Demo 9 : Offline Model Stealing Attack [H/W]

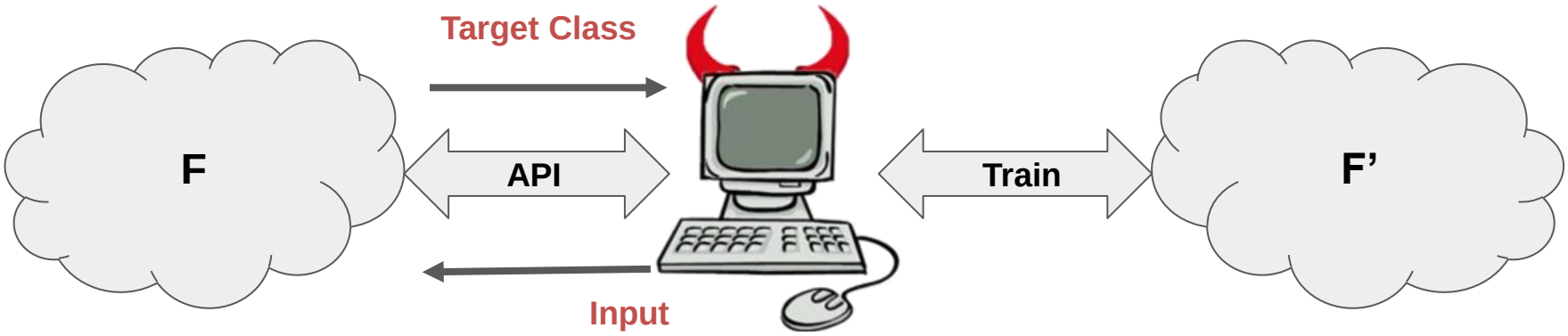
Online Black Box model stealing

MLaaS

MLaaS



Model Stealing attack methodology

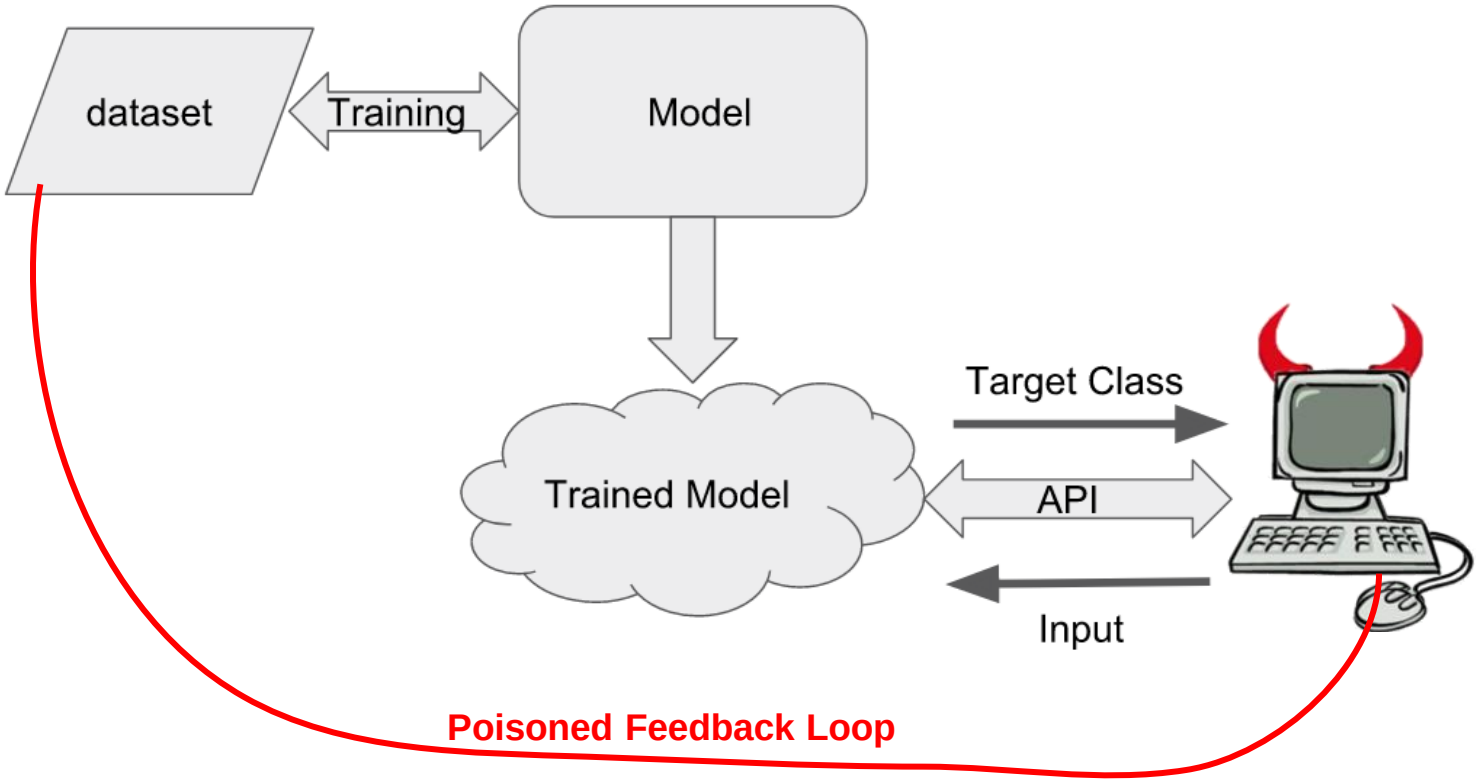


Model Stealing attack methodology

- Collect data
- Query prediction APIs to generate labels from target Model
- Use generated Labels and collected data to train a duplicate model
- Optimization algorithms like Gradient descent are used to train the

Model skewing Attacks

Model skewing Attacks

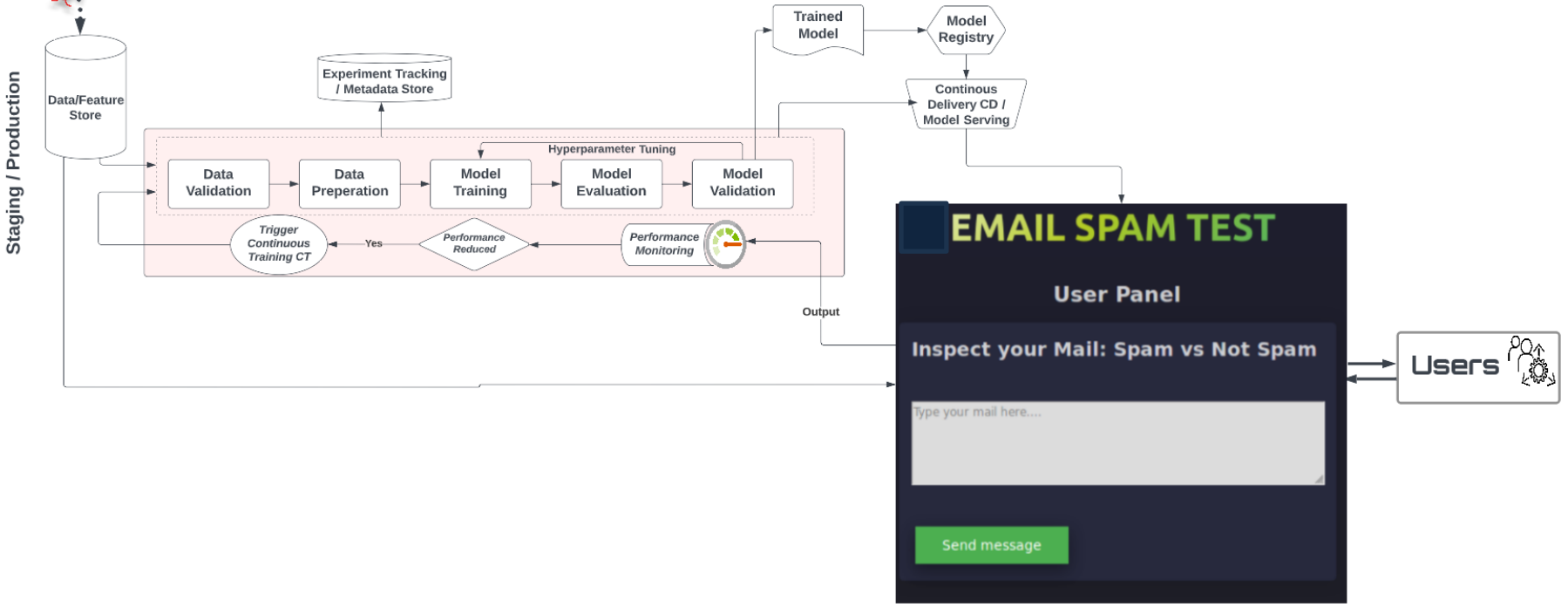


Demo 10: Skewing Spam Filters

Data Poisioning Attack

Data Poisioning Attack

Attack Type: Label Flipping

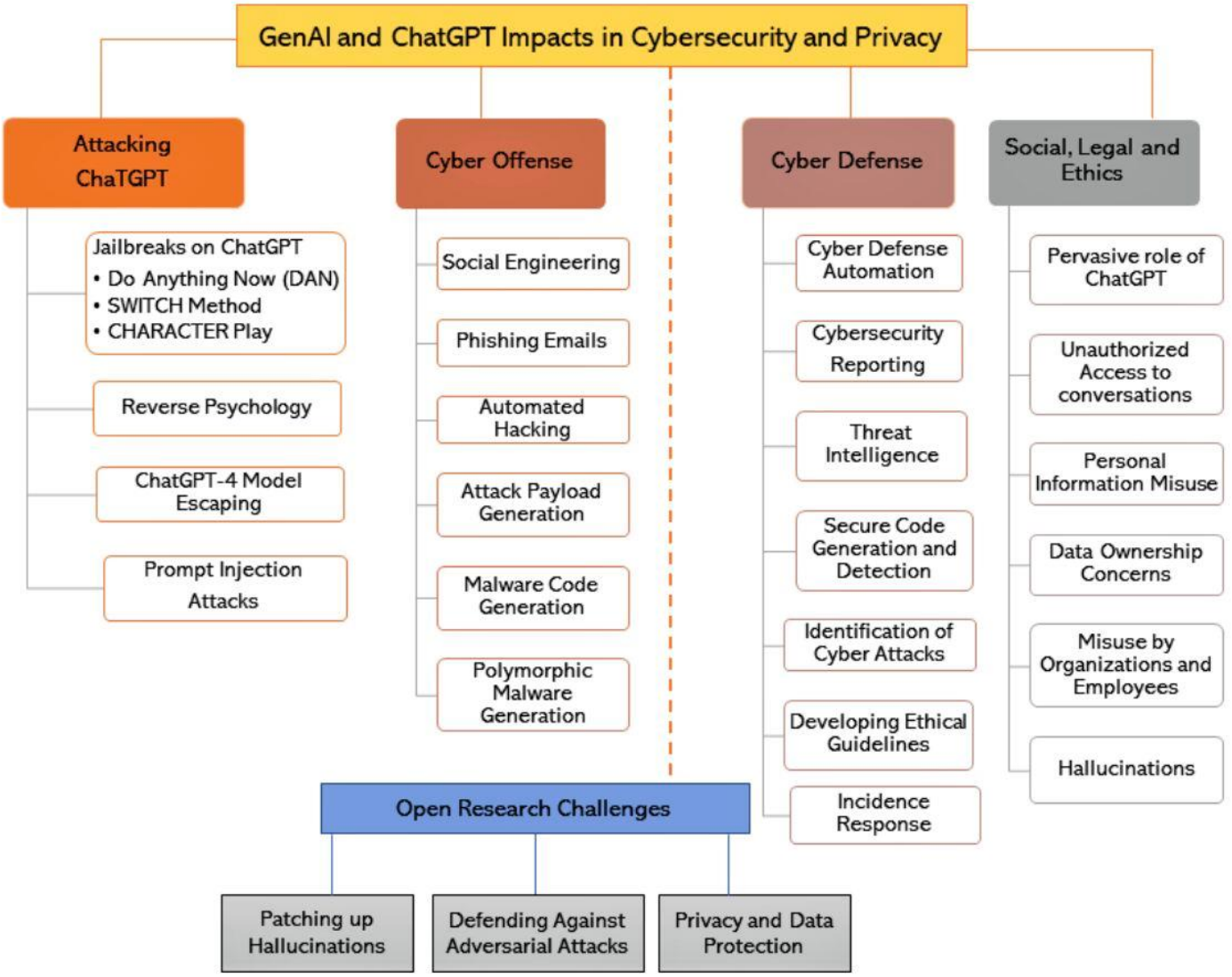


Data Poisioning Attack

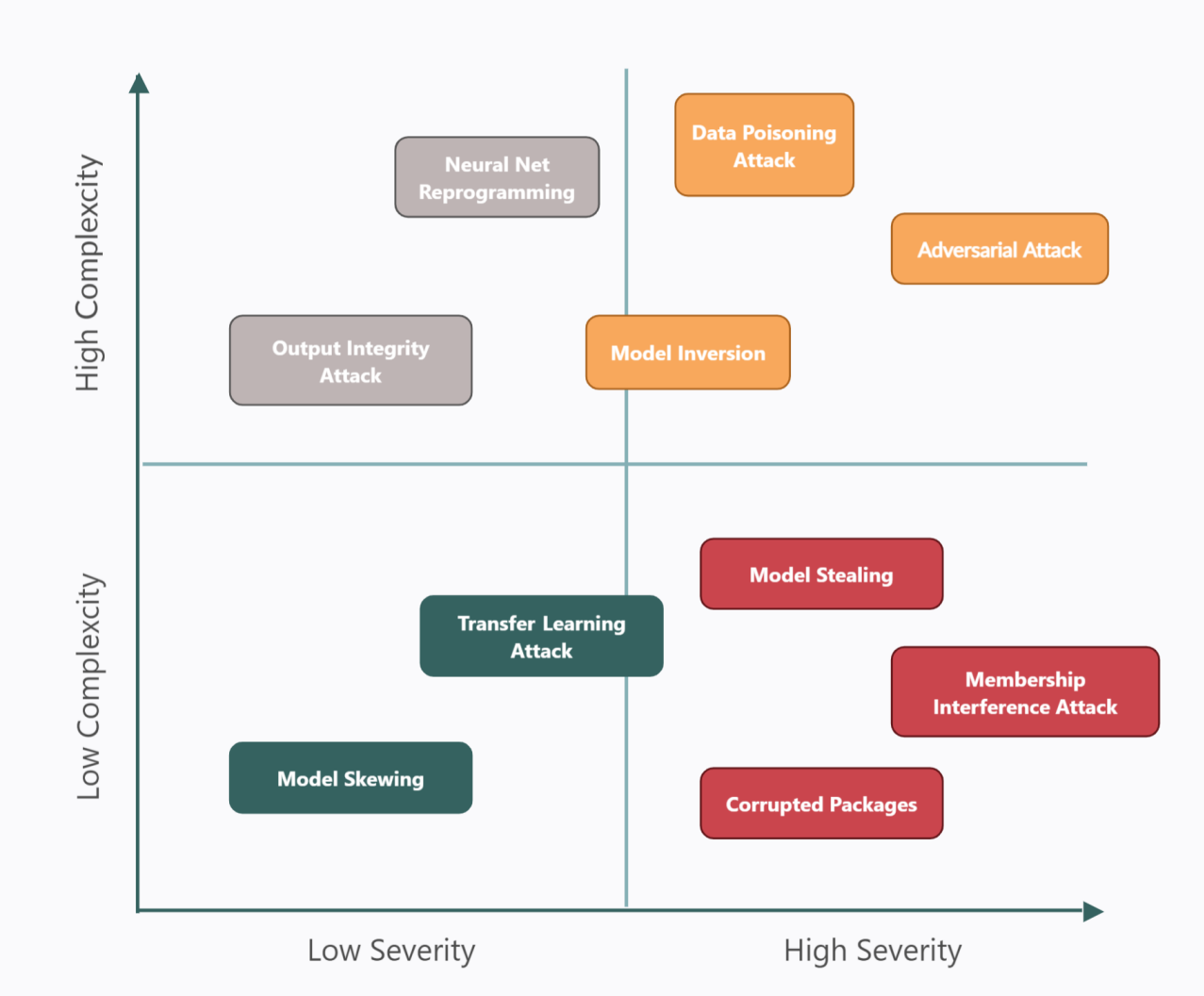
Demo X:label flipping data poisoning attack

Play Video: /seasides/Data Poisoning.mp4

Generative AI



ML Top 10: Risk Assessment Matrix



OWASP “Machine Learning Security top 10”

https://mltop10.info/

OWASP Machine Learning Security Top Ten

Main | Charter | Related | Glossary

owasp incubator License CC BY-SA 4.0

Important Information

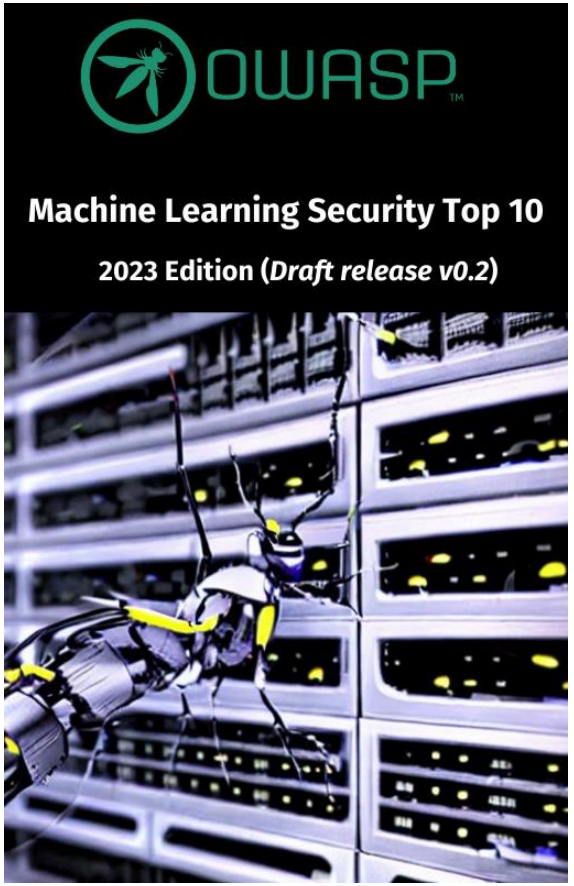
The current version of this work is in draft and is being modified frequently. Please refer to the project wiki for information on how to contribute and project release timelines.

Overview

Welcome to the repository for the OWASP Machine Learning Security Top 10 project! The primary aim of the OWASP Machine Learning Security Top 10 project is to deliver an overview of the top 10 security issues of machine learning systems. More information on the project scope and target audience is available in our project working group charter

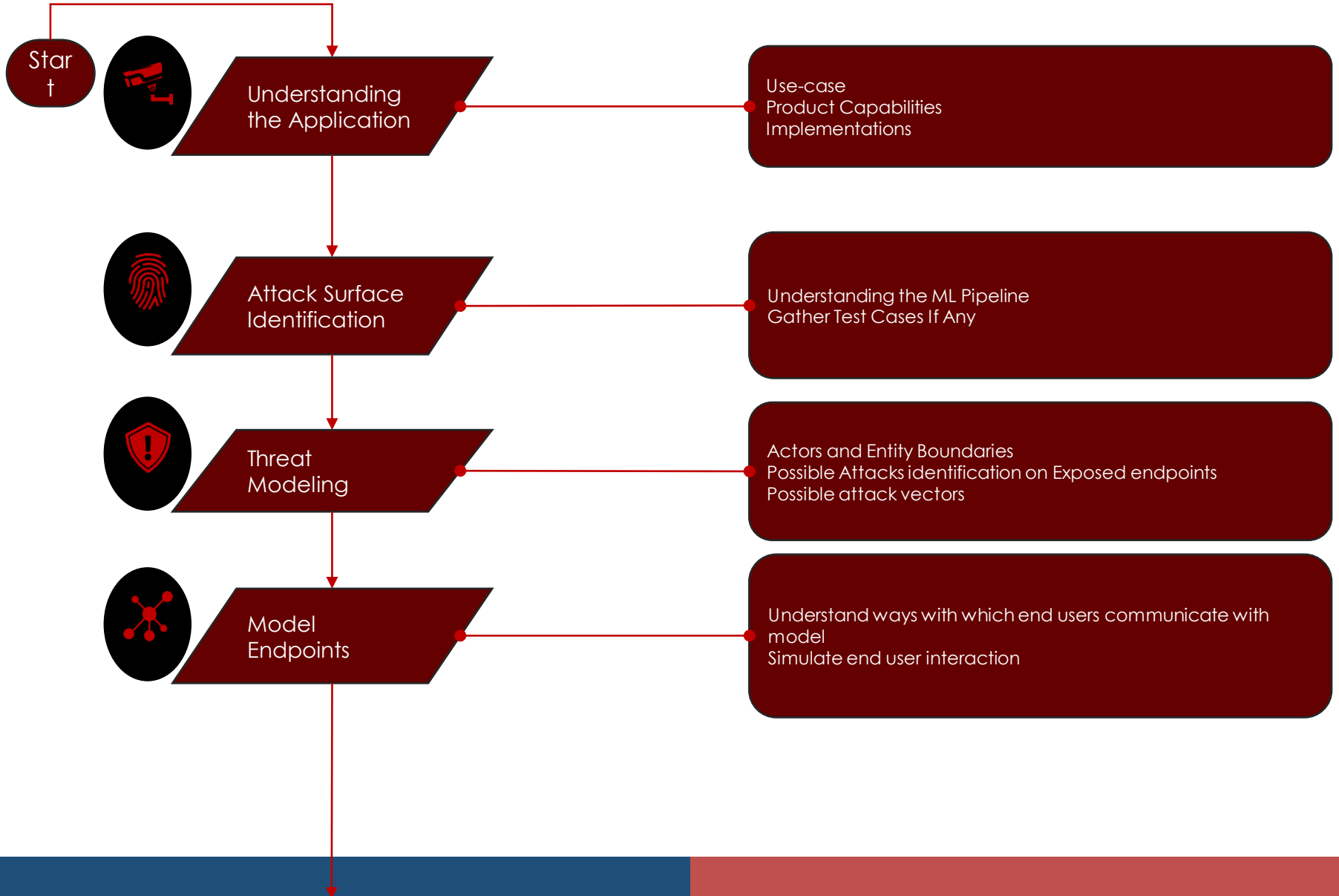
Top 10 Machine Learning Security Risks

- ML01:2023 Input Manipulation Attack
- ML02:2023 Data Poisoning Attack
- ML03:2023 Model Inversion Attack
- ML04:2023 Membership Inference Attack
- ML05:2023 Model Stealing
- ML06:2023 Corrupted Packages
- ML07:2023 Transfer Learning Attack
- ML08:2023 Model Skewing
- ML09:2023 Output Integrity Attack
- ML10:2023 Model Poisoning

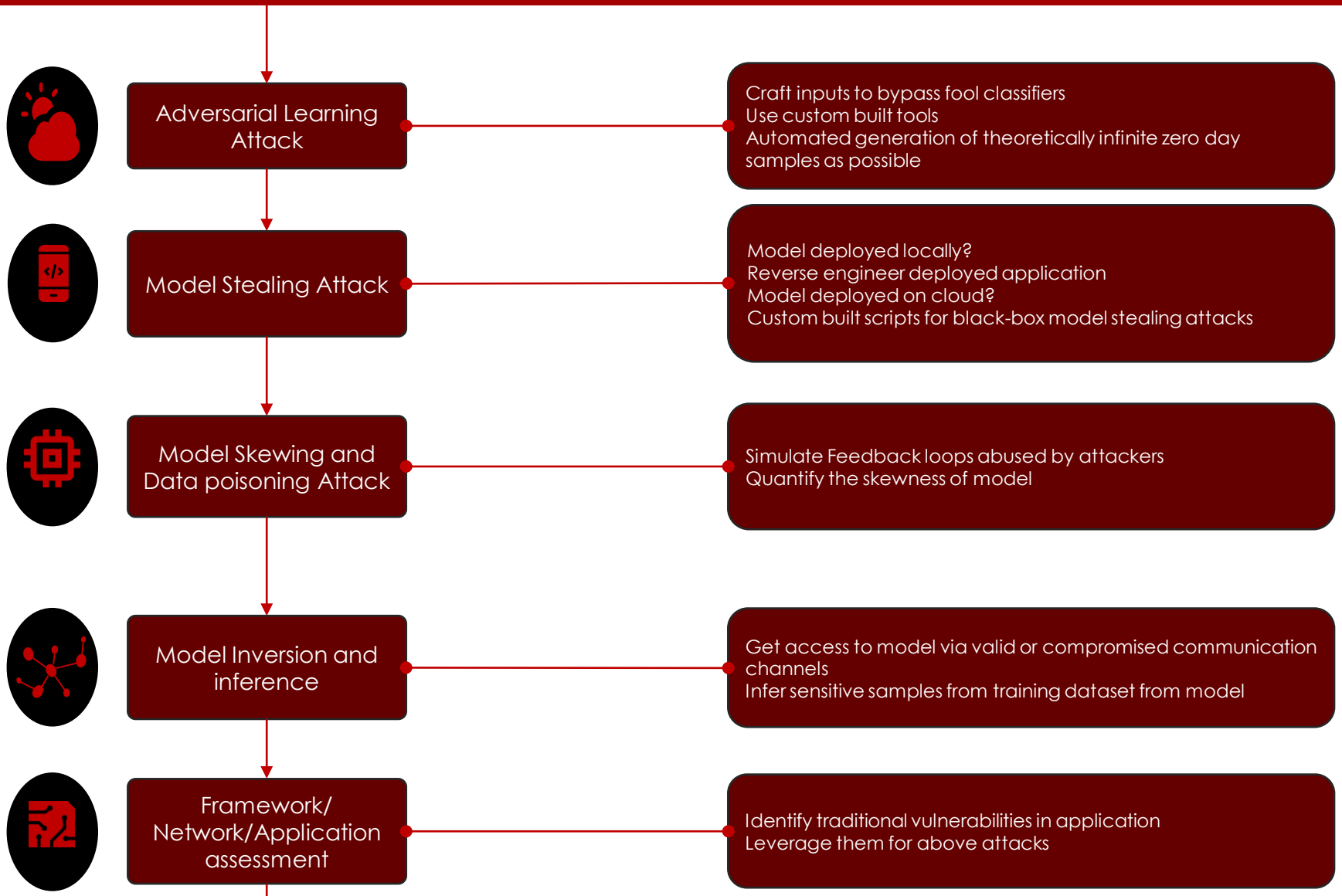


{ref} https://owasp.org/www-project-machine-learning-security-top-10/

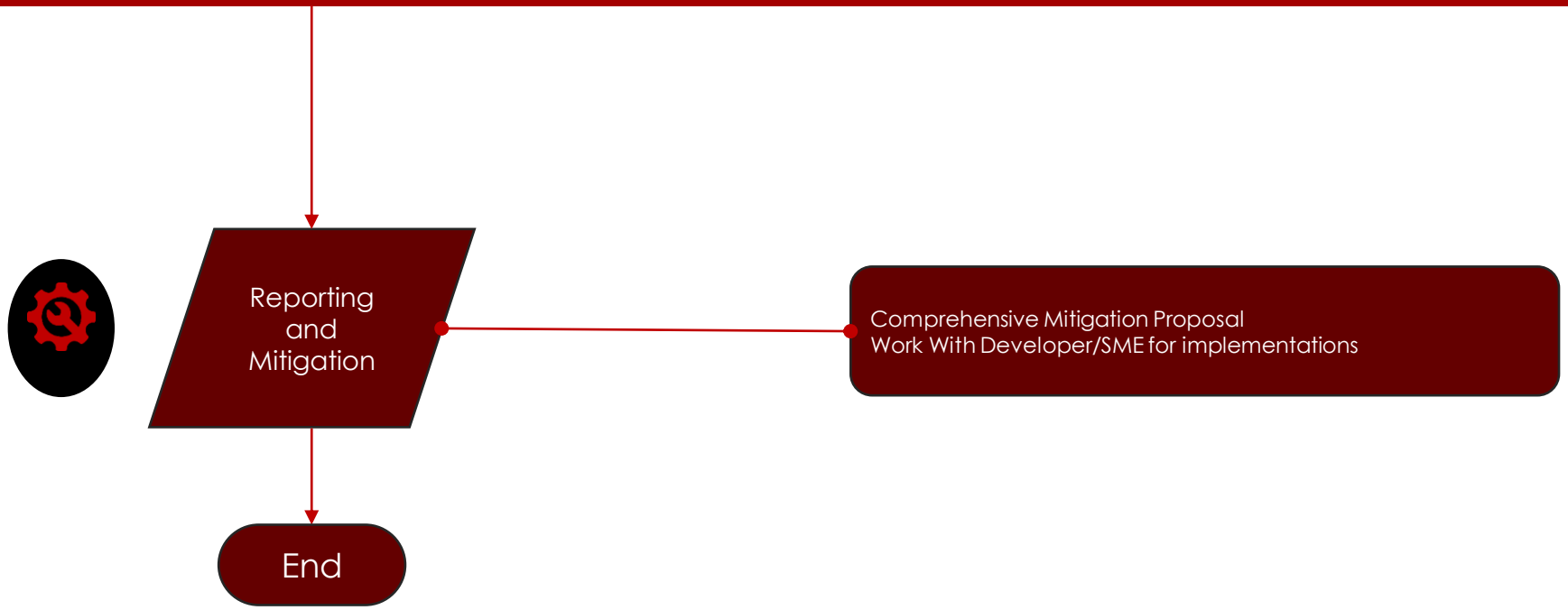
ML Security Assessment Process



ML Security Assessment Process



ML Security Assessment Process



CTF

An organisation has implemented an authentication system "mlAuth" using machine learning, which is 99.9% accurate. Every employee has a profile(represented by a string on 784 hex values). mlAuth is trained using these profiles to predict the probability of authenticity for an employee. System grants access only if the predicted probability is higher than 0.99. Hence, your aim is to generate a fake profile that will trick the 99.9% accurate mlAuth in granting you access.

```
[Hint]
# You can make use of dumped machine learning model to conduct your targated
attack. Are you smart enough to fool the "intelligent" mlAuth?
```

Next Steps