深度学习的安全应用及隐患

肖煌 | 德国弗劳恩霍夫协会应用集成信息安全研究所

FIT · FREEBUF

Overview | 主题内容一览

- ❑ Introduction to Deep Learning
- ❑ Learning capability of Security Applications
- ❑ Deep learning for Cyber Security
- ❑ Adversarial Machine Learning
- ❑ Conclusion and take home messages

- ❑ 什么是深度学习?
- ❑ 信息安全应用的学习能力
- ❑ 深度学习的安全应用
- ❑ 对抗式的机器学习
- ❑ 总结及建议

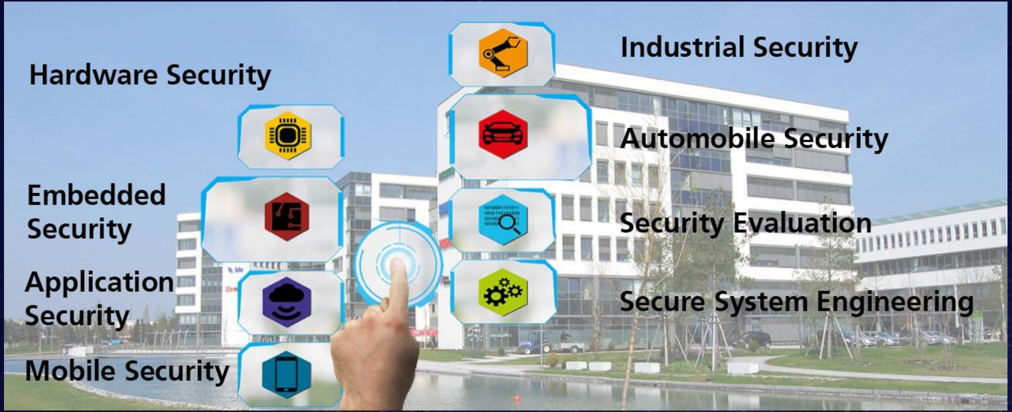
Who are we? | 我们是谁

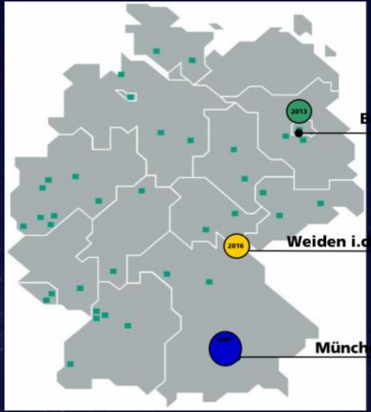
- Founded in 1949 as NPO
- Largest applied research organization in Germany and Europe
- 69 institutes, headquarter in Munich
- 24.500 researchers (2017)
- More than 2.1 billion budget per year Serves 3.000 industry clients
- More than 10.000 R&D projects per year

- 1949年成立的公益性科研机构
- 德国以及欧洲最大的产学研机构
- 69个不同的研究所(总部慕尼黑)
- 24.500 全职科研人员(2017)
- 年经费逾21亿欧元
- 年均服务3000 企业客户
- 年均完成超过10000项科研项目

Fraunhofer
AISEC

Who are we? | 我们是谁



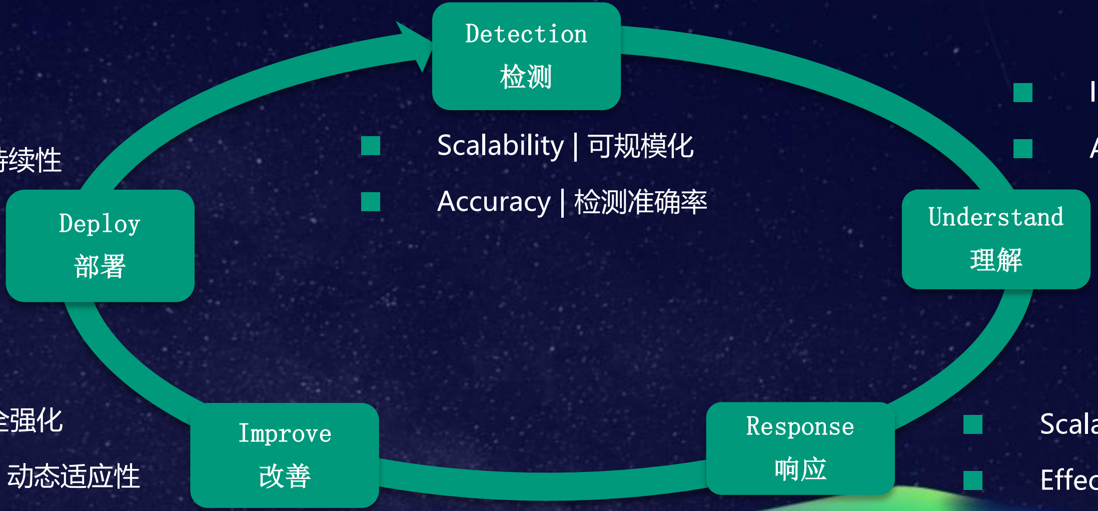


德国柏林自由大学

德国慕尼黑工业大学

Cognitive Security Technologies | 认知信息安全

FIT · FREEBUF



- Integrity | 完备性
- Sustainability | 可持续性

- Scalability | 可规模化
- Accuracy | 检测准确率

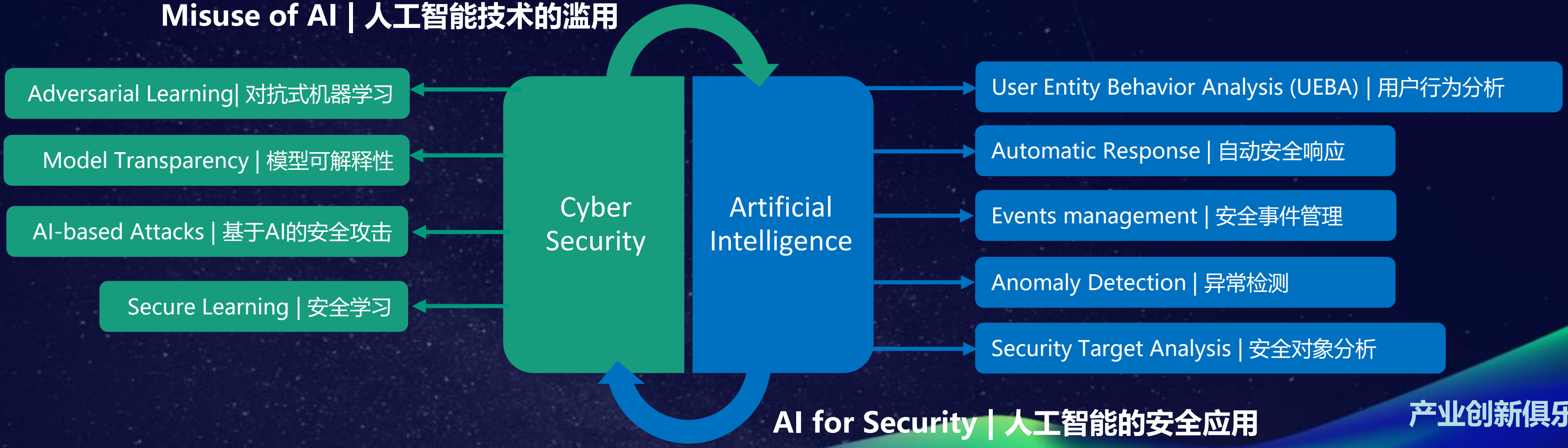
- Interpretability |可表述性
- Accountability | 可解释性

- Hardness | 安全强化
- Adaptiveness | 动态适应性

- Scalability | 可规模化
- Effectiveness | 有效性

产业创新俱乐部

Key Topics | 主要研究课题



Misuse of AI | 人工智能技术的滥用
Adversarial Learning| 对抗式机器学习
Model Transparency | 模型可解释性
AI-based Attacks | 基于AI的安全攻击
Secure Learning | 安全学习
Cyber Security
Artificial Intelligence
User Entity Behavior Analysis (UEBA) | 用户行为分析
Automatic Response | 自动安全响应
Events management | 安全事件管理
Anomaly Detection | 异常检测
Security Target Analysis | 安全对象分析
AI for Security | 人工智能的安全应用

产业创新俱乐部

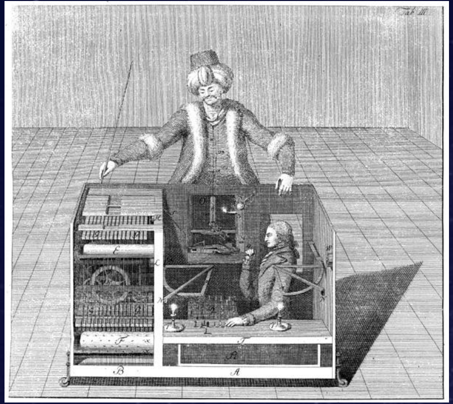
FIT · FREEBUF

深度学习简介

INTRODUCTION TO DEEP LEARNING

产业创新俱乐部

Deep Learning in History | 深度学习的发展历程



1770: The Turk **土耳其行棋傀儡**
“Fake AI”

AI? 人工智能?

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
SONGBIRDS À LA CARTE
SAFEGUARD TRANSPARENCY
WHEN GENES GOT 'SELFISH'

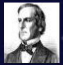
2016: DeepMind: AlphaGo

Deep Learning in History | 深度学习的发展历程

1700s

Reasoning about probability of events
Thomas Bayes, "An essay towards solving a problem in doctrine of chances."

1800s



Logical reasoning by George Boole. "An investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities. "

1900s

Emergence of **statistics** enables inferences from data.

统计学，数据分析

1950-1951

Colossus was the first electronic digital programmable computing device

Alan Turing.
1912 - 1954
Computing machinery and Intelligence. Mind 59.

1956

Dartmouth Workshop: Birth of name "AI"

John McCarthy
Marvin Minsky
Claude Shannon
Ray Solomonoff
Alan Newell
Herbert Simon
Arthur Samuel
And three others...
Oliver Selfridge (Pandemonium theory)
Nathaniel Rochester (IBM, designed 701)
Trenchard More (Natural Deduction)

可编程，机器智能

1950-1960

专家系统，神经网络

Newell & Simon - Logic Theorist. General Problem Solver. CMU
Gelerntner - Geometry Theorem Prover. IBM
Samuel - Checkers Learner. IBM
McCarthy - Lisp. Moved to Stanford

1960s

Slagle - SAINT. Symbolic integration
Evans - ANALOGY. Geometric analogy problems.
Borrow - STUDENT. Algebra story problems.
Waltz, Huffmann - Vision. Blocks Worlds.
Winston - Learning in blocks worlds.
Robinson - Resolution methods
Green - Planning
Shakey the robot.
Widrow, Rosenblatt - neural nets, Adaline, Perceptron

1970s

Walz Huffmann - Vision Blocks Worlds.
Winston - Learning in blocks worlds
Winograd - Natural language blocks worlds
AI Winter I

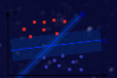
1980s

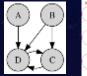
Rumelhart, Hinton - Error propagation, connectionist models for cognitive science
Bart, Sutton, Anderson - adaptive critic design for reinforcement learning
Hopfield - Nets for storage and optimisation
R1 - Commercial expert system DEC.
AI Winter II

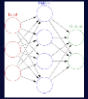
1990-2000

统计学习理论，贝叶斯

Machine learning
Vapnik, V. et al - Statistical Learning Theory, e.g., SVM, Kernel Methods
Michale I. Jordan - Probabilistic Graphical Models



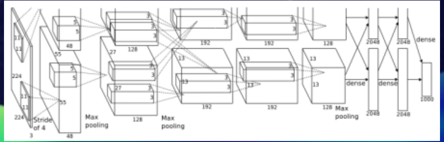




Now

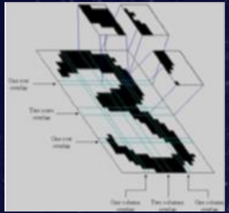
深度神经网络

Hinton, Bengio, LeCun, et al, Deep Learning

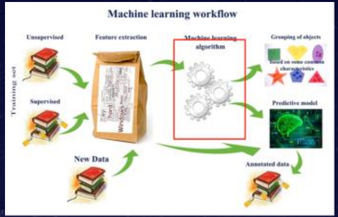


深度学习的发展历程

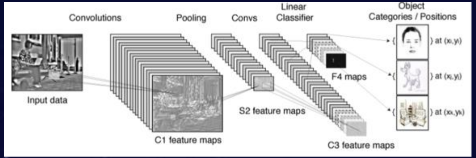
Fast pacing AI development | AI的高速发展



~70s-80s: Pattern Recognition 模式识别
- ✓ Based on rules, logic, expert knowledge 规则, 逻辑, 专家知识
- ✓ Automate tasks by program 自动化程序



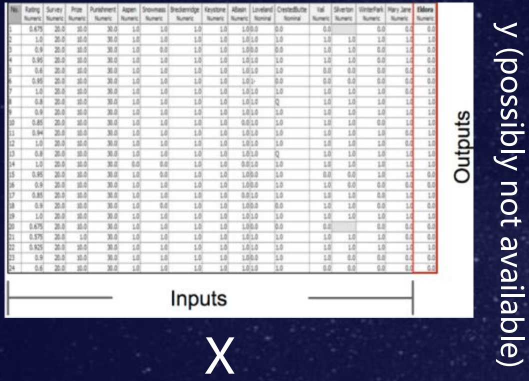
~90s - now: Machine learning 机器学习
- ✓ Uncertainty introduced by Pearl et al. 不确定性的引入
- ✓ Learning models from seen data samples (historical view) 从数据中学习模型
- ✓ Focus on prediction 预测能力
- ✓ E.g., SVM, decision tree, Naïve Bayes, etc…

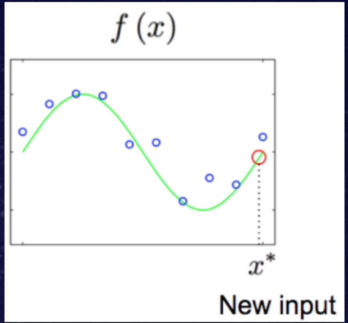


2006 - now: Deep learning 深度学习
- ✓ Renaissance of NNs
- ✓ Unsupervised feature learning
- ✓ Learning non-local patterns
- ✓ Mimic how cortex works
- ✓ No theoretical ground so far
- ✓ Works like a charm

从数据提炼知识

Given a dataset (X, y) consists of N rows（samples）, we learn a function f(x) that can predict unseen samples with minimal errors.


Inputs
Outputs
X
y (possibly not available)

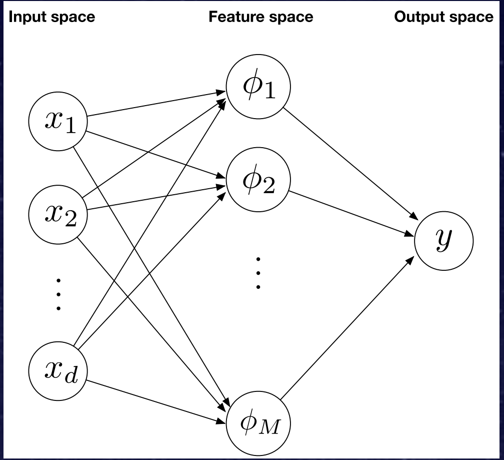


- 从数据中进行函数抽象化
- 抽象化等同于某优化过程
- 训练数据推广泛化到未知数据

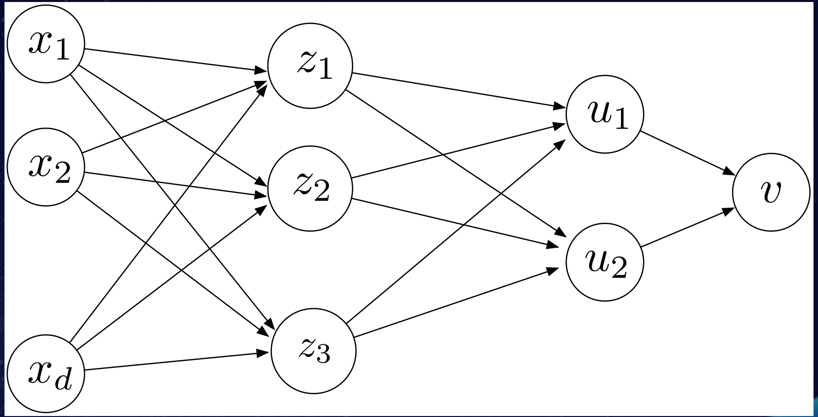
浅层模型 vs. 深度模型

FIT · FREEBUF

- 支撑向量机
- 逻辑回归
- 最大似然估计
- etc



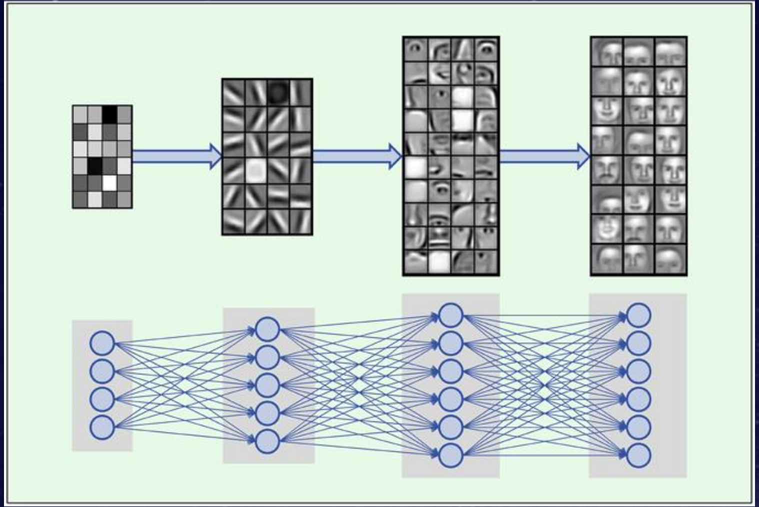
浅模型



深模型：更强的表达能力

产业创新俱乐部

深度学习的优点



- 计算机技术领域近几十年最令人兴奋的领域
- 依赖海量数据, 算法, 以及计算资源
- 极为灵活的算法架构层出不穷
- 统一的计算(编程)模式 (Backpropogation)
- 高度非线性问题的学习
- 极强的描述数据能力
- 自动化的特征学习
- 中间层的可迁移特性
- 在线学习
- 算法以及硬件的加速

FIT · FREEBUF

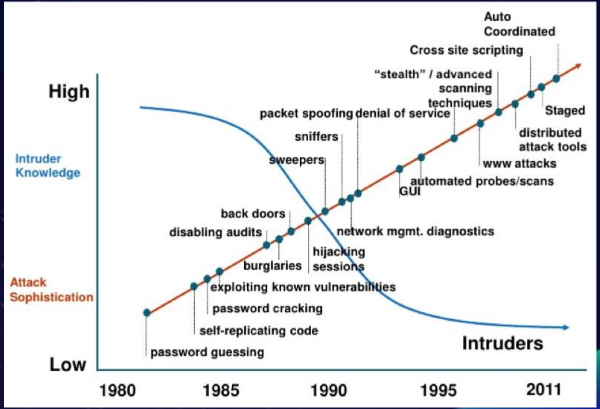
信息安全应用的学习能力

LEARNING CAPABILITY OF SECURITY APPLICATIONS

产业创新俱乐部

安全应用自主学习的动机

- Attack sophistication is increasing | 攻击复杂性越来越高
- Required expertise is decreasing | 需要的技能门槛越来越低
- Quality of tools is improved constantly | 黑客工具的质量逐步提升
- Technological trends: (技术发展态势)
  - Big data | 大数据
  - Cloud and IoT | 云计算及物联网
  - Industry 4.0 | 工业4.0 等



安全应用自主学习的动机

- Security Analysts are often overwhelmed
- Data collection & Storage & Retrieving is well underway
- Current detection techniques might fail on
  - Polymorphic malwares
  - Zero-day attacks
  - APTs
- Dissolving network perimeter:
  - BYOD/Cloud

- 训练有素的安全分析师越来越难以应对繁重的工作
- 数据的ETL技术越发成熟
- 现有的安全检测技术在以下场景未必有效
  - 多态病毒软件
  - 零日攻击
  - 高级持续性威胁
- 日益消失的网络边界:
  - 自带设备/云计算

Machine Learning for Security | 基于机器学习的安全应用

- Typical use cases
  - Botnet detection
  - Malware classification
  - Spam/Phishing detection
  - Insider attacker detection
  - Web attacks detection
  - Biometrics e.g., face, finger, iris

- 例如一些传统的安全应用场景
  - 僵尸网络监测
  - 病毒分类
  - 垃圾邮件,钓鱼网站检测
  - 内部入侵检测
  - Web攻击检测
  - 生物特征识别, 如脸部, 指纹, 虹膜等

深度学习的安全应用

DEEP LEARNING FOR CYBER SECURITY

产业创新俱乐部

Malware classification | 恶意病毒分类

FIT · FREEBUF

- More and more new distinct malware samples – how to keep up?
- Reverse engineering, malware analysis – time consuming
- Signature-based methods – already surpassed, but still often used
- However, many samples are similar: variations, families
- Motivates machine learning approaches, detection driven by data from:
    - Static analysis (malware code, PE header metadata)
    - Dynamic analysis (e.g. system calls, network traffic)

- 不同种类的病毒样本越来越多 – 如何应对?
- 逆向, 病毒分析 – **耗时过长!**
- 基于签名的方法 – 过时,但仍然广泛使用
- 然而许多病毒样本是相似的: 变种, 相似的族群
- 基于数据的机器学习算法
    - 静态分析(malware code, PE header metadata)
    - 动态分析(e.g. system calls, network traffic)

Bojan et al., **Empowering Convolutional Networks for Malware Classification and Analysis**. In 30th IJCNN.

产业创新俱乐部

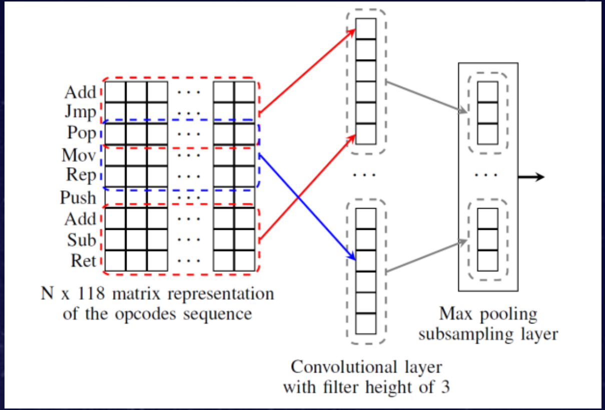
Malware classification | 恶意病毒分类

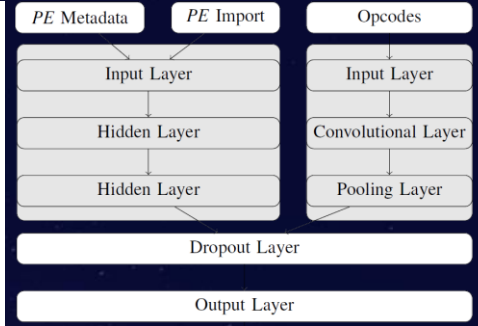
- Construct convolutional networks to capture similarity between instruction sequences
- Use improvements in feature extraction in order to improve malware detection and classification
- Combine these features with metadata from the PE Header in one neural network architecture

- 利用卷积神经网络捕捉命令序列的相似性
- 利用深度神经网络在特征提取上的优越性提高病毒检测分类的能力
- 结合PE Header 中的meta信息构造统一的神经网络架构

Malware classification | 恶意病毒分类

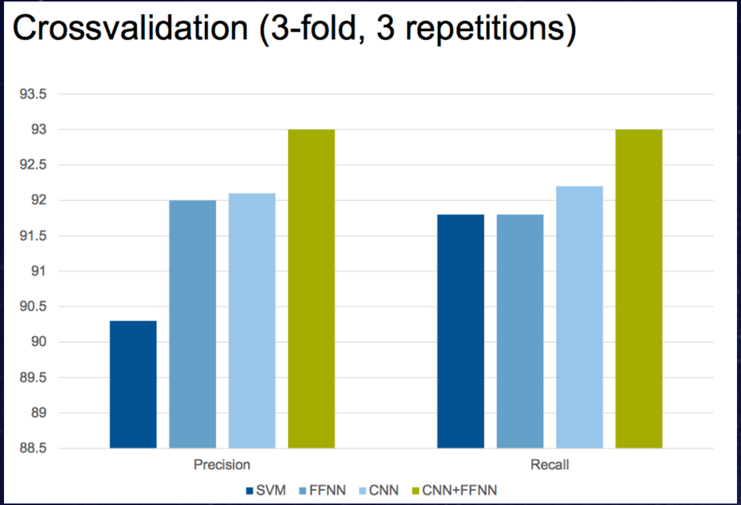
- 22,694 binaries
- 63 apps from ZDNet (benign)
- 13 malware classes (labels from virustotal)
- Instruction sequences (opcodes) using objdump
- PE header features using PEInfo
  - PE Metadata: Timestamps, offsets and sizes of image resources, section sizes (virtual and raw),…
  - PE Import features: imported functions and DLLs (one-hot encoding)


Add
Jmp
Pop
Mov
Rep
Push
Add
Sub
Ret
N x 118 matrix representation of the opcodes sequence
Convolutional layer with filter height of 3
Max pooling subsampling layer



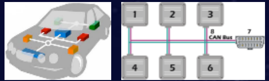
Convolutional + classic feedforward network
卷积+前馈神经网络

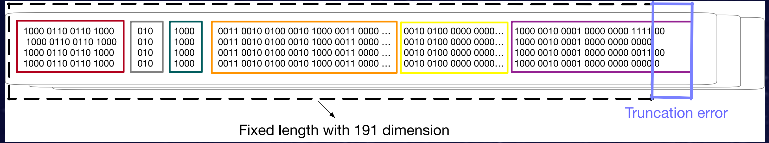
Malware classification | 恶意病毒分类



Outperform other ML methods with 93% accuracy

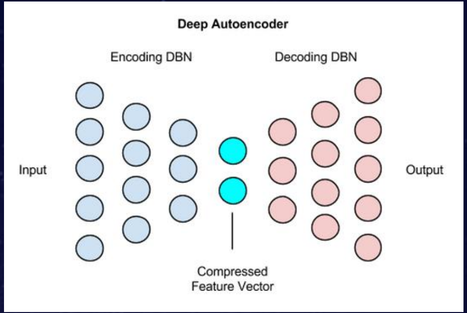
Anomaly Detection for CAN | 控制局域网络的异常检测





```
1000 0110 0110 1000   010   1000   0011 0010 0100 0010 1000 0011 0000 ...   0010 0100 0000 0000...   1000 0010 0001 0000 0000 1111 00
 1000 0110 0110 1000  010   1000   0011 0010 0100 0010 1000 0011 0000 ...   0010 0100 0000 0000...   1000 0010 0001 0000 0000 0000
1000 0110 0110 1000   010   1000   0011 0010 0100 0010 1000 0011 0000 ...   0010 0100 0000 0000...   1000 0010 0001 0000 0000 0011 00
1000 0110 0110 1000   010   1000   0011 0010 0100 0010 1000 0011 0000 ...   0010 0100 0000 0000...   1000 0010 0001 0000 0000 0000 0
```

Input 输入



Output输出

E.g., CAN Traces in Automobile
Unsupervised deep Autoencoder
Anomaly: if the distance between input and output is big
L defines a distance function

$$L(x_{input} - \hat{x}_{output}) > \epsilon$$

输入深度自编码器，如果输出和输入相差较大，即判别为CAN网络异常.

产业创新俱乐部

Anomaly Detection for CAN | 控制局域网络的异常检测

FIT · FREEBUF

深度自编码器能够从正常的数据X中学习一组嵌入(embedding)用以描述正常行为的空间向量.

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x_i} - f(\mathbf{x_i})\|^2 \text{ and } \mathbf{x_i} \in \mathbf{X}$$

学习的过程即优化上述目标函数使输入和输出的误差最小. 例如利用backpropagation,

$$f(x_i) = h^m(W^m \cdot h^{m-1}(W^{m-1} \cdot h^{m-2}(\ldots h^1(W^1 x_i + b_1) + b_{m-1}) + b_m)$$

梯度下降:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}} = \frac{2}{N}\sum_{i=1}^{N}(x_i - f(x_i))\frac{\partial f}{\partial h^m}\cdot\frac{\partial h^m}{\partial W^m}\cdots\frac{\partial h^1}{\partial W^1} \Longleftrightarrow \mathbf{W}_{k+1} = \mathbf{W}_k - \alpha\frac{\partial\mathcal{L}}{\partial\mathbf{W}}$$

产业创新俱乐部

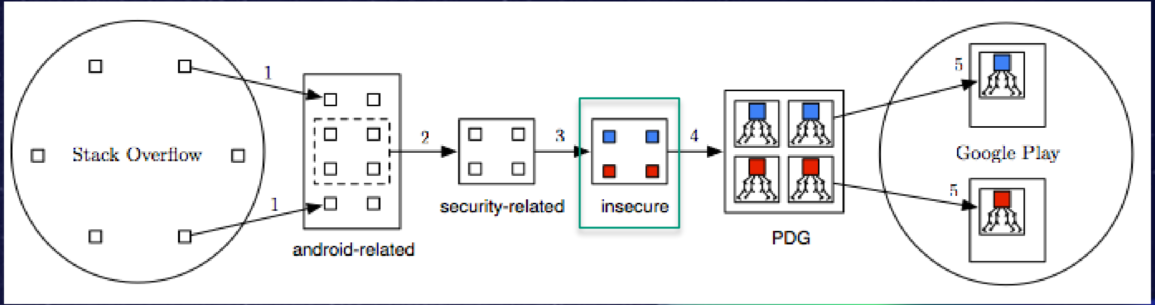
Code Security Detection | 代码安全性检测

代码从开发社区拷贝粘贴至产品开发中,从而带来的安全隐患.

Felix F., Konstantin B., Huang X., Christian S., Yasemin A., Michael B., Sascha F. **Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security.** In 38th IEEE Symposium on Security and Privacy, May. 2017.

Code Security Detection | 代码安全性检测

- **Fully automated processing pipeline | 全自动分析流程**
- **Measures the flow of secure and insecure code snippets | 测量安全及非安全代码片的流向情况**
- **Program dependency graph (PDG) based clone detection | 基于PDG的代码拷贝检测**
- **手动标注1517个代码片, 训练SVM分类模型**



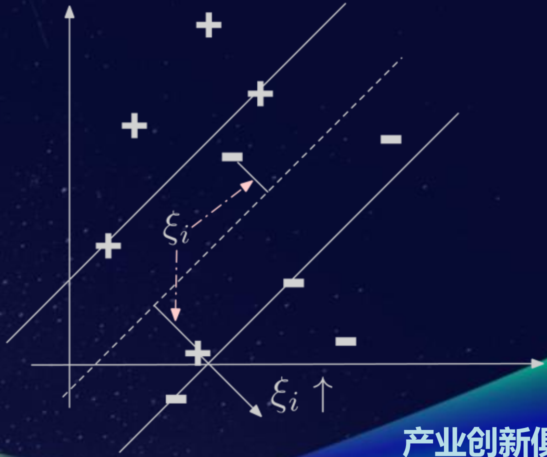
Code Security Classification | 代码片安全分类

FIT · FREEBUF

- Binary Support Vector Machine 支撑向量机
- Maximize a margin classifier 最大化边界将不同类别的数据分开
- Secure vs. Insecure
- TF-IDF 特征处理, Tokens as features
- Accuracy: 准确度 89%

Insecure: 75%

```
// Insert security-related Java code snippet
SecretKeySpec getKey() {
  final pass = "47e7717f0f37ee72cb226278279aebef".getBytes("UTF-8");
  final sha = MessageDigest.getInstance("SHA-256");
  def key = sha.digest(pass);
  key = Arrays.copyOf(key, 16);
  return new SecretKeySpec(key, AES);
}
```



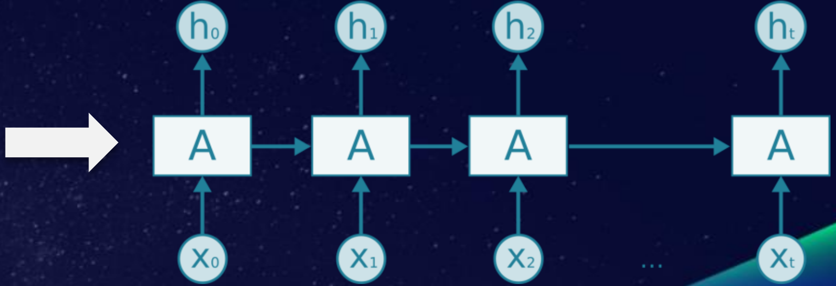
产业创新俱乐部

Code Security Classification | 代码片安全分类

- How about deep learning? 如何利用深度学习做代码分类?
- Sequence Model, e.g., LSTM | 时序模型模拟代码片, 如长短记忆网络
- Accuracy: ~72%(Overfitting) | 实际效果过拟合
- 数据量太少, 模型太复杂

```
// Insert security-related Java code snippet
SecretKeySpec getKey() {
  final pass = "47e7717f0f37ee72cb226278279aebef".getBytes("UTF-8");
  final sha = MessageDigest.getInstance("SHA-256");
  def key = sha.digest(pass);
  key = Arrays.copyOf(key, 16);
  return new SecretKeySpec(key, AES);
}
```

Source: http://colah.github.io



FIT · FREEBUF

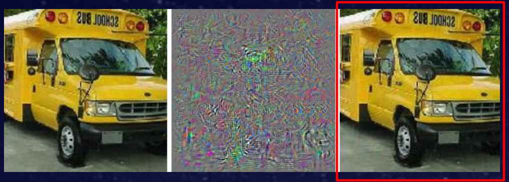
深度学习的安全隐患

ADVERSARIAL MACHINE LEARNING

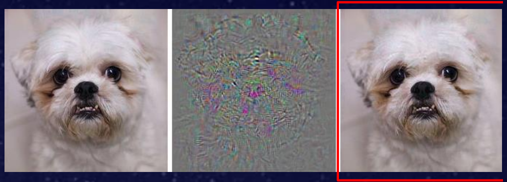
产业创新俱乐部

Deep neural network easily fool?

- 问题: 深度学习本身值得信赖吗?

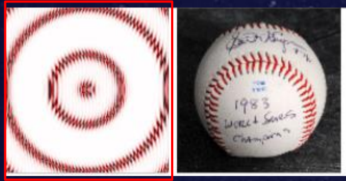


校车 + 噪音 != 校车

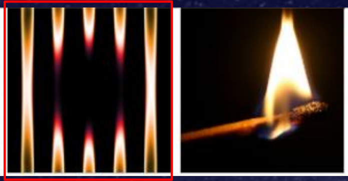


小狗 + 噪音 != 小狗

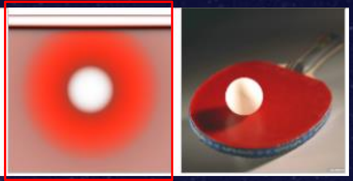
Nguyen A, Yosinski J, Clune J. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In Computer Vision and Pattern Recognition (CVPR '15), IEEE, 2015.



垒球 (99%)



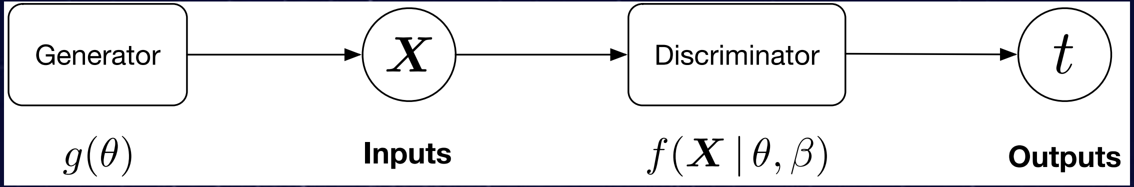
火柴(99%)



乒乓球 (99%)

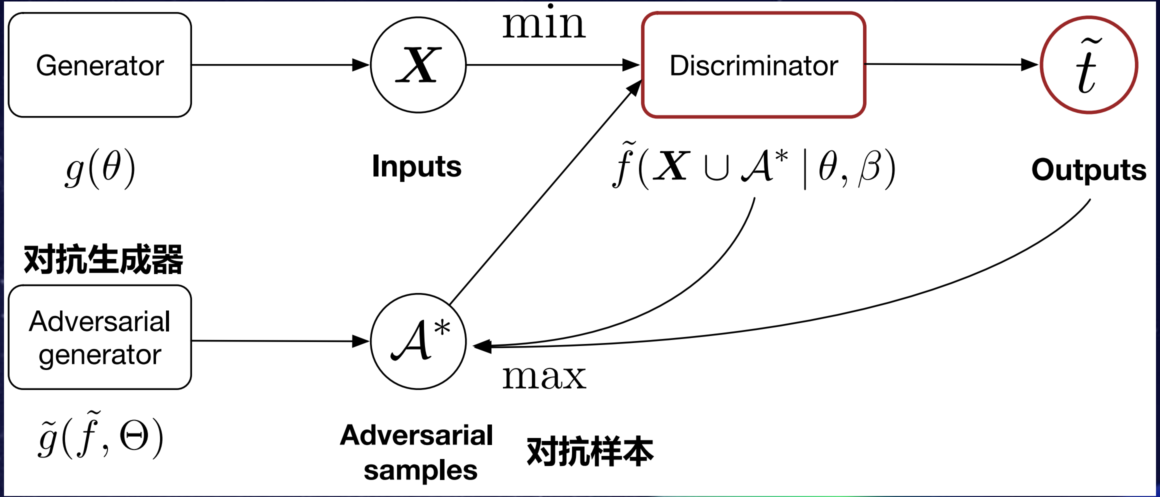
产业创新俱乐部

Functional view of machine learning | 机器学习的函数视角



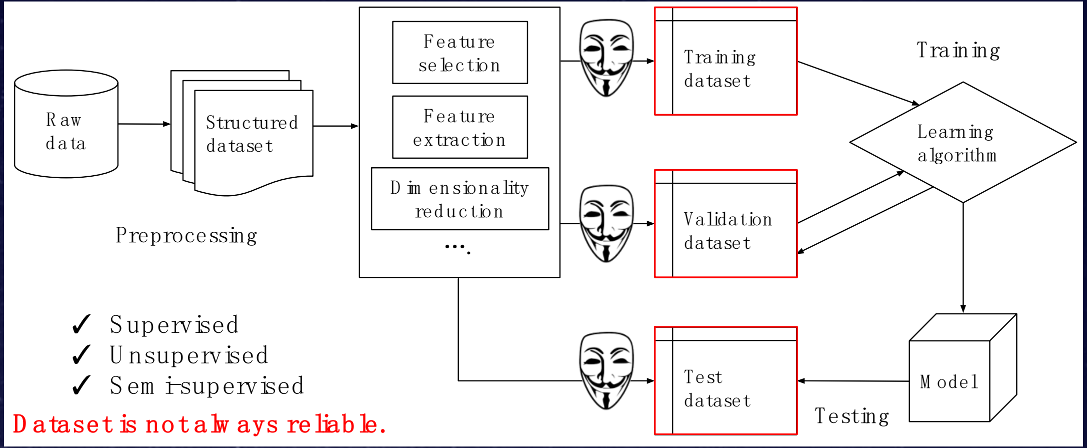
- X~g：所观察的数据X由某一生成过程产生
- f: X =>t 从数据中得到的学习模型
- 数据的一致性假设不一定成立

Machine learning under adversary | 存在对抗条件下的机器学习



产业创新俱乐部

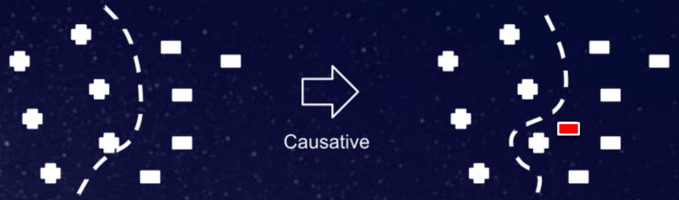
Adversarial machine learning |对抗式机器学习



安全视角下的学习过程

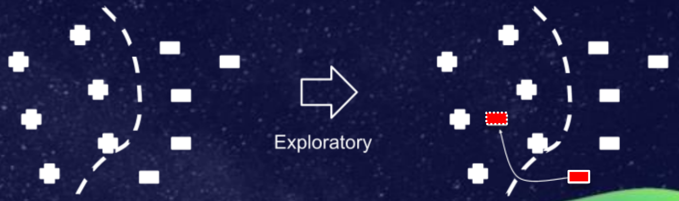
Adversarial attacks taxonomy|对抗式攻击分类

因果式攻击
- 注入恶意样本
- 破坏分类器本身



Causative attack
- Inject adversarial samples
- Compromise the whole classifier

探索式攻击
- 探索样本分类边界
- 以最小的代价绕过分类器的检测

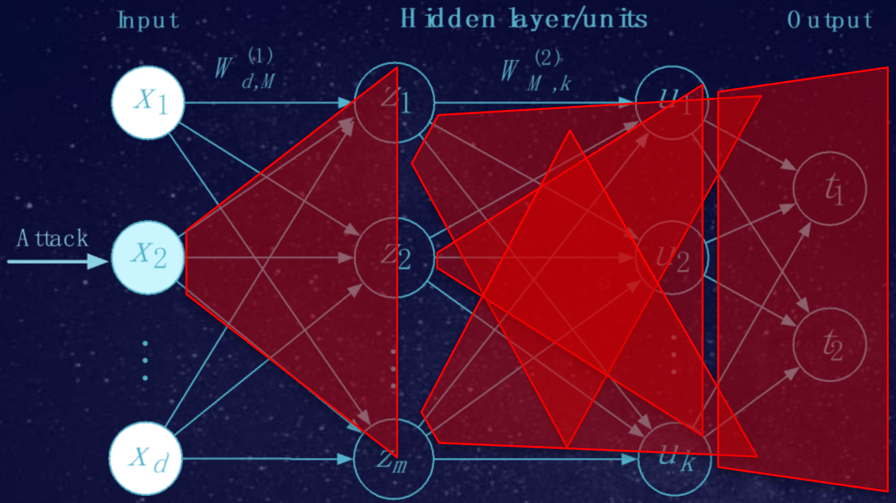


Exploratory attack
- Explore the boundary
- Circumvent the classifier with minimal cost

产业创新俱乐部

Compromise Deep NN | 攻击深度神经网络

FIT · FREEBUF



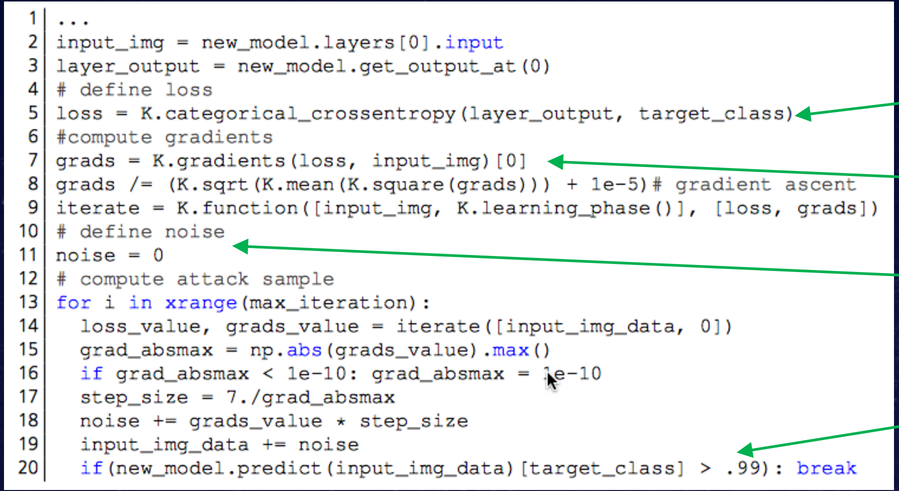
- Manipulation on single input broadcast to all hidden parameters and output.
- Much easier than before

- 改变局部输入特征广播至全局神经元
- 攻击方式比以前大大简化

产业创新俱乐部

Compromise Deep NN | 攻击深度神经网络

Now 现在



```
...
input_img = new_model.layers[0].input
layer_output = new_model.get_output_at(0)
# define loss
loss = K.categorical_crossentropy(layer_output, target_class)
#compute gradients
grads = K.gradients(loss, input_img)[0]
grads /= (K.sqrt(K.mean(K.square(grads))) + 1e-5)# gradient ascent
iterate = K.function([input_img, K.learning_phase()], [loss, grads])
# define noise
noise = 0
# compute attack sample
for i in xrange(max_iteration):
  loss_value, grads_value = iterate([input_img_data, 0])
  grad_absmax = np.abs(grads_value).max()
  if grad_absmax < 1e-10: grad_absmax = 1e-10
  step_size = 7./grad_absmax
  noise += grads_value * step_size
  input_img_data += noise
  if(new_model.predict(input_img_data)[target_class] > .99): break
```

定义攻击目标损失函数

自动计算梯度, e.g., Keras
~20 lines in Python

随机开始生成攻击样本

迭代至神经网络误差足够大收敛

产业创新俱乐部

Compromise Deep NN | 攻击深度神经网络

Past 过去

$$\max_{\boldsymbol{x}_c} \mathcal{W} = \frac{1}{m} \sum_{j=1}^{m} \ell\left(\hat{y}_j, f(\hat{\boldsymbol{x}}_j)\right) + \lambda \Omega(\boldsymbol{w}) \qquad (3)$$

$$\frac{\partial \mathcal{W}}{\partial \boldsymbol{x}_c} = \frac{1}{m} \sum_{j=1}^{m} \left(f(\hat{\boldsymbol{x}}_j) - \hat{y}_j\right) \left(\hat{\boldsymbol{x}}_j^\top \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c} + \frac{\partial b}{\partial \boldsymbol{x}_c} \right) + \lambda \boldsymbol{r} \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c},$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}^\top = \frac{1}{n} \sum_{i=1}^{n} \left(f(\hat{\boldsymbol{x}}_i) - \hat{y}_i\right) \hat{\boldsymbol{x}}_i + \lambda \boldsymbol{r}^\top = \boldsymbol{0}, \qquad (5)$$

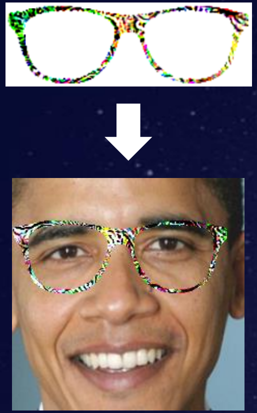
$$\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{n} \sum_{i=1}^{n} \left(f(\hat{\boldsymbol{x}}_i) - \hat{y}_i\right) = 0, \qquad (6)$$

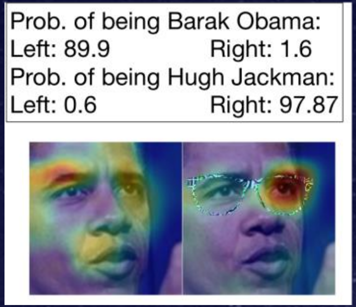
$$\begin{bmatrix} \boldsymbol{\Sigma} + \lambda \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}_c} & \boldsymbol{\mu} \\ \boldsymbol{\mu}^\top & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_c} \\ \frac{\partial b}{\partial \boldsymbol{x}_c} \end{bmatrix} = -\frac{1}{n} \begin{bmatrix} \mathbf{M} \\ \boldsymbol{w}^\top \end{bmatrix},$$

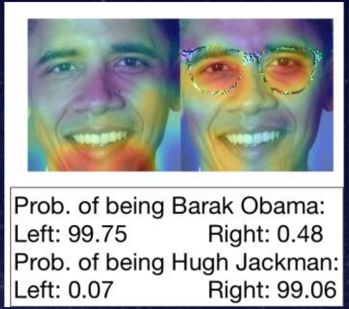
手动计算梯度，并编写代码
~300 lines in Python

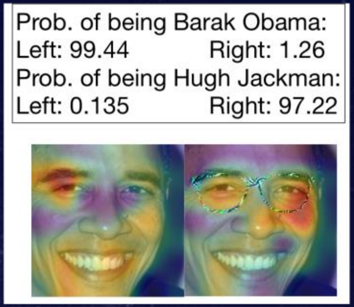
产业创新俱乐部

EvilFace | 绕过人脸识别









- Model trained by transfer learning from VGG16
- Attack target: Barack Obama -> Huge Jackman
- Face feature is misled by glass (see heatmap)

- VGG16迁移学习人脸识别模型
- 攻击目标: 奥巴马 -> 休杰克曼
- 脸部识别特征被恶意眼镜误导

Conclusion & Take Home Message | 总结与建议

FIT · FREEBUF

- Deep learning is very powerful and essential for security of the trend
- Deep learning does not work every where
- Need careful tuning
- Complexity of model must be aligned with data
- Be aware of the vulnerability of deep learning
- Design secure and reliable security service with AI

- 深度学习非常强大, 对信息安全大势所趋
- 深度学习并非总能成功
- 需要仔细的调整模型
- 模型的复杂度必须和数据(问题)匹配
- 深度学习存在潜在的隐患
- 设计基于AI的值得信赖的安全服务

产业创新俱乐部

Thanks for your attention
感谢您的聆听

Fraunhofer AISEC Institute

Dr. Huang Xiao
Director of Cognitive Security Technologies

Parkring. 4
85748 Garching b. München,
Germany

Tel. +49 89 3229986-149
Fax +49 89 3229986-299
Mobile +49 176 4290 6699

Mail: huang.xiao@aisec.fraunhofer.de