基于机器学习的物理不可克隆函数（PUF）建模攻击

一、 实验目的

1) 学习及掌握机器学习的基础算法（线性回归，逻辑回归）；
2) 学习 SVM，ANN，CNN，CMA-ES 等算法；
3) 学习 Python 的基本语法以及掌握其 Tensorflow 包的使用；
4) 掌握 PUF 的相关知识和原理；
5) 使用机器学习对 Arbiter PUF 进行建模攻击。

二、 实验原理

2.1、机器学习

机器学习是什么？

- 1+1 等于几 ？
- 50
- 笨，多了

- 1+2 等于几？
- 20
- 笨，多了

- 3+4 等于几？
- 7
- 真聪明，对了

- 6+9 等于几？
- 13
- 笨，少了

久而久之……

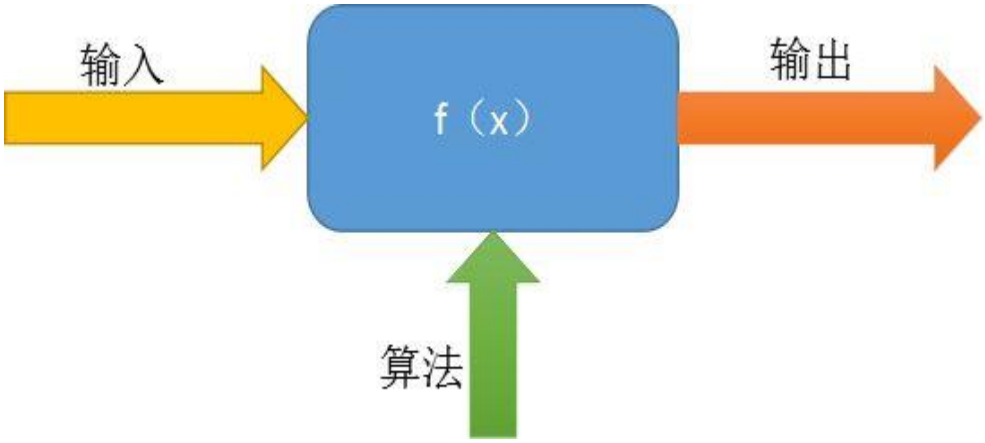
- 2+2 等于几？
- 4

- 4+5 等于几？
- 9

这就是机器学习，准确来说是最常见的一种，监督学习。最开始的几步是对于模型的训练，“多了”或“少了”可以理解为训练时的误差，模型根据误差调整自身参数，这就是机器学习里常用的反向传播(Backpropagation)的简单的解释。

现实生活中，我们会碰到两类问题：

1.一类问题，是给定输入，通过施加一定条件（或算法），得到最终的输出。就像下图这样：



典型的例子，在用计算机解决问题的时候很常见，比如给定一个数的集合（输入），通过编写算法实现数组从小到大排序。输出是一个有序列表。

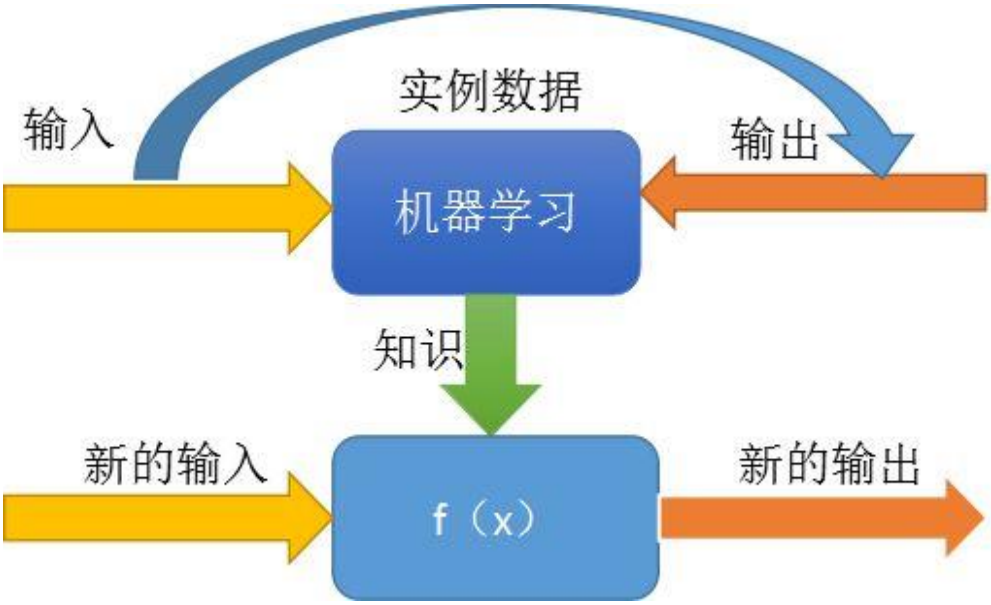
对于这类问题，人类能够自己设定一种模式（函数），把输入映射成想要的输出。

2.另一类问题，人类找不到这样的模式。以 OCR 字符识别为例，输入是手写体（数字）图片，输出是 0-9 字符串，我们并不知道怎么把输入转换成输出，因为手写体因人而异，随机性很大。

换句话说，这个时候，我们缺的是知识（如何映射），不过幸运的是，我们有（实例）数据。

而把这个知识通过机器（计算机）学出来的过程，叫做机器学习。

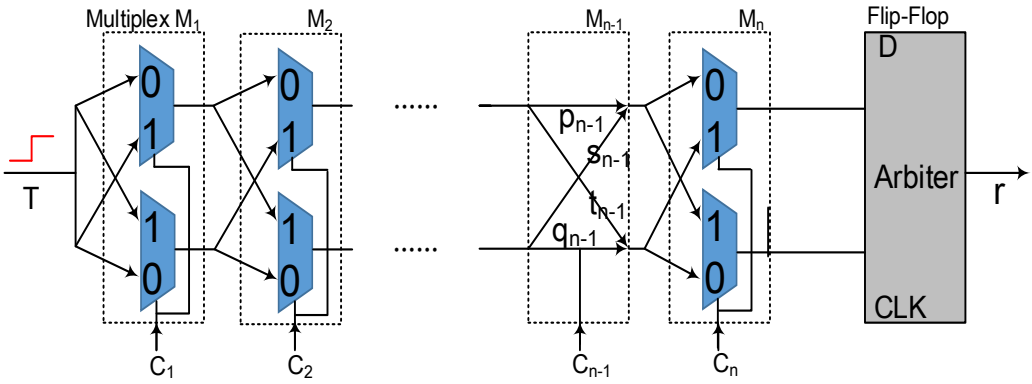
这个学出来的知识（或经验），可以用于新的输入，产生新的输出。



无论哪种问题，产生我们想要的输出才是目的，机器学习或计算机程序只是手段。

2.2、PUF

物理不可克隆函数（Physical Unclonable Function，PUF）是一种新的轻量级硬件安全原语。当输入一个激励时，PUF 利用芯片制造过程中难以预测的工艺偏差（Process Variation），输出依赖于芯片的不可克隆的响应，非常适合资源受限环境下的设备认证。然而，攻击者可以收集一定数量的激励响应对将 PUF 进行建模，因此，PUF 易受基于机器学习建模攻击。下图是一种典型的 PUF——Arbiter PUF，其中$\{C_1, C_2, \ldots, C_{n-1}, C_n\}$共同组成激励，r 为响应。其原理是：一个脉冲信号 T 会在 Arbiter PUF 上下两条路径同时传播，通过激励$\{C_1, C_2, \ldots, C_{n-1}, C_n\}$改变路径（如$C_1$=1 时，在$M_1$阶段交叉传播；$C_1$=0 时，在$M_1$阶段平行传播），由于工艺偏差会影响不同路径的传播快慢，最终导致上下两条路径信号传播产生快慢差异，比较传播快慢生成激励响应 r（0 或 1）。



2.3、建模攻击

线性回归和逻辑回归是入门级的机器学习算法，本实验需自行掌握这两个算法，另外如支持向量机（SVM），人工神经网络（ANN），卷积神经网络（CNN），协方差矩阵自适应进化策略（CMAS-ES）需根据分组情况学习其一，最后使用逻辑回归以及分得的四种算法（SVM, ANN, CNN, CMA-ES）其一对 Arbiter PUF 进行建模攻击，本实验只需对仿真的 Arbiter PUF 进行建模即可。

在我们的仿真实验中，建模主要分为三个步骤：

1. 在仿真的 Arbiter PUF 上获取一定量的激励响应对作为训练数据，再获取一定量的激励响应对（要求 10000 对，不能与训练数据的激励相应对相同）作为测试数据。
2. 使用训练数据的激励响应对构建攻击模型。
3. 使用测试数据的激励响应对测试攻击模型的准确率。（实验要求准确率达到 98%以上，并报告使用了多少对训练数据到达该效果）

三、 实验环境

操作系统：Windows

编程语言：Python 并安装 Tensorflow 包

四、 实验简要介绍

本实验给出一个 csv 文件“仿真 Arbiter_PUF.csv”，可用 Excel 打开，实验中可用 python 读入。

实验步骤：

1. 安装 python 以及 Tensorflow 包；
2. 使用 csv_file = csv.reader(open('仿真 Arbiter_PUF.csv','r'))读入数据；
3. 编写程序获取训练数据与测试数据，并保存为文件形式；
4. 读入训练数据以及测试数据，使用 Tensorflow 对机器学习模型进行训练；
5. 不断扩大训练数据集，使得训练的模型达到准确率要求。

五、 实验要求

实验要求使用机器学习模型对 Arbiter PUF 进行建模攻击，要求建模准确率达到 98%以上，最后需提交完成的实验报告至少包括但不仅限于以下内容：

- 实验的目的和意义
- 实验的原理
- 实验的操作过程（请截取必要步骤截图）
- 建模成功的准确率截图
- 实验的心得体会，不少于 500 字（由于本实验是分组完成，每人都需要写 500 字，需包含个人分工内容）
- 报告以组的形式打包（实验报告+程序）发给粟海翰（QQ 即可）

编写人：粟海翰

时间 2018 年 4 月