

# CSEE5590/CS490-0004: AI for Cybersecurity

Gharib Gharibi

Summer 2019

# Instructor

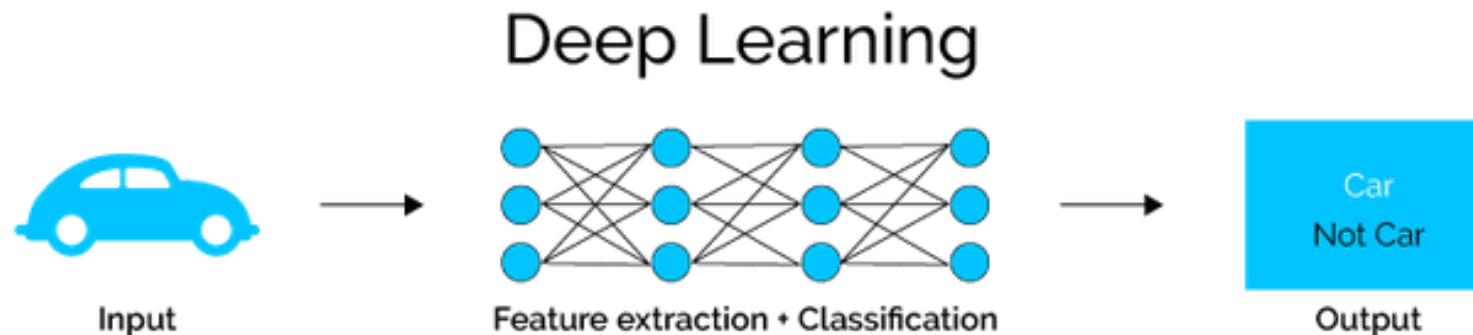
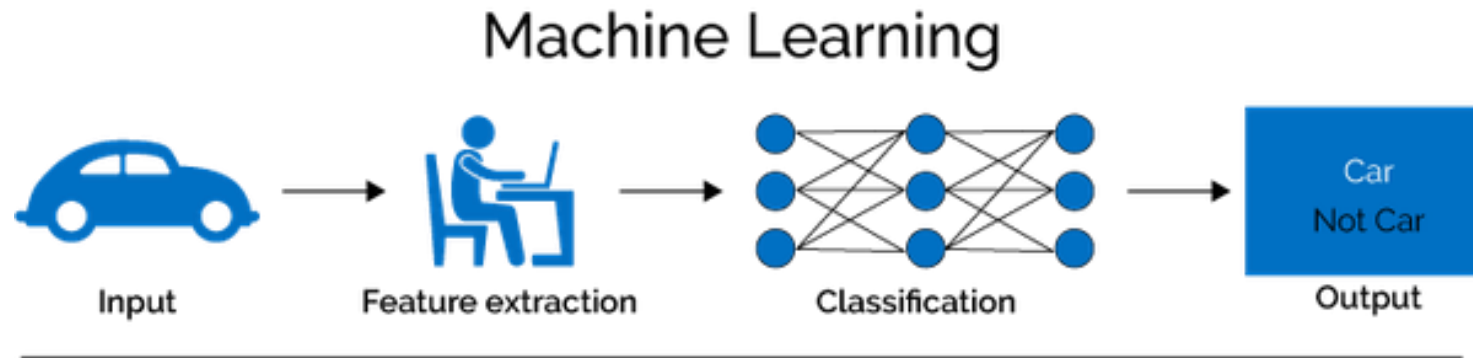
- Gharib Gharibi
- [ggk89@mail.umkc.edu](mailto:ggk89@mail.umkc.edu) (preferred)
- Office: FH 519 (TuTh 3-4pm) or by appointment

# Course: AI for Cybersecurity

- AI, and specifically Deep Learning, has advanced the SOTA in an growing number of domains.
- Thus, there is a high demand to develop DL models, but the privacy and security of these models have been ignored for some extent.
- We study how to secure DL models and preserve data privacy when developing and implementing DL models.

# What is Deep Learning?

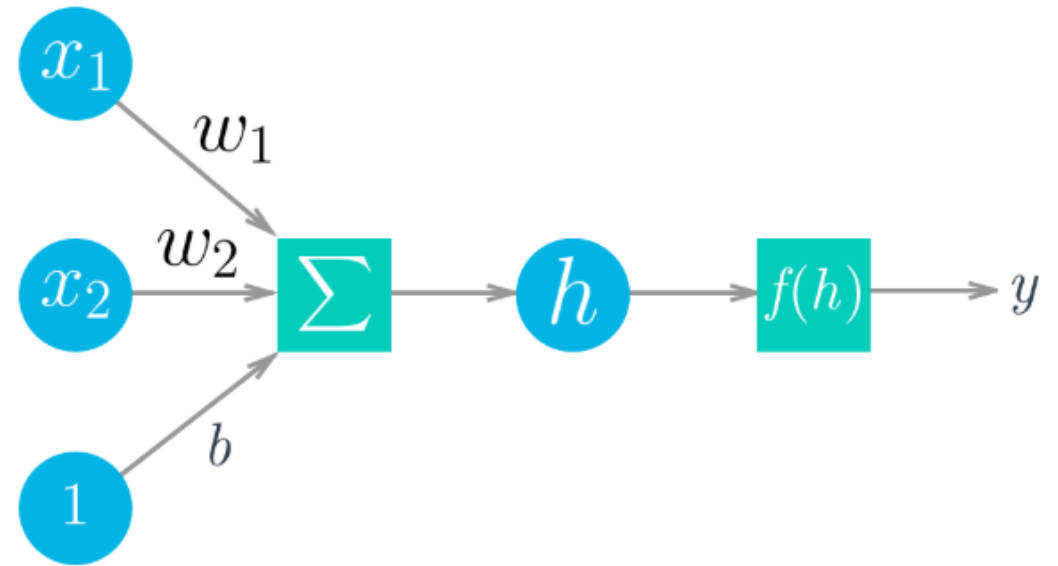
- A subfield of Machine Learning that can automatically detect the features/patterns of data without user intervention.



# Neural Networks

Deep Learning is based on ***artificial neural networks*** which have been around in some form since the late 1950s.

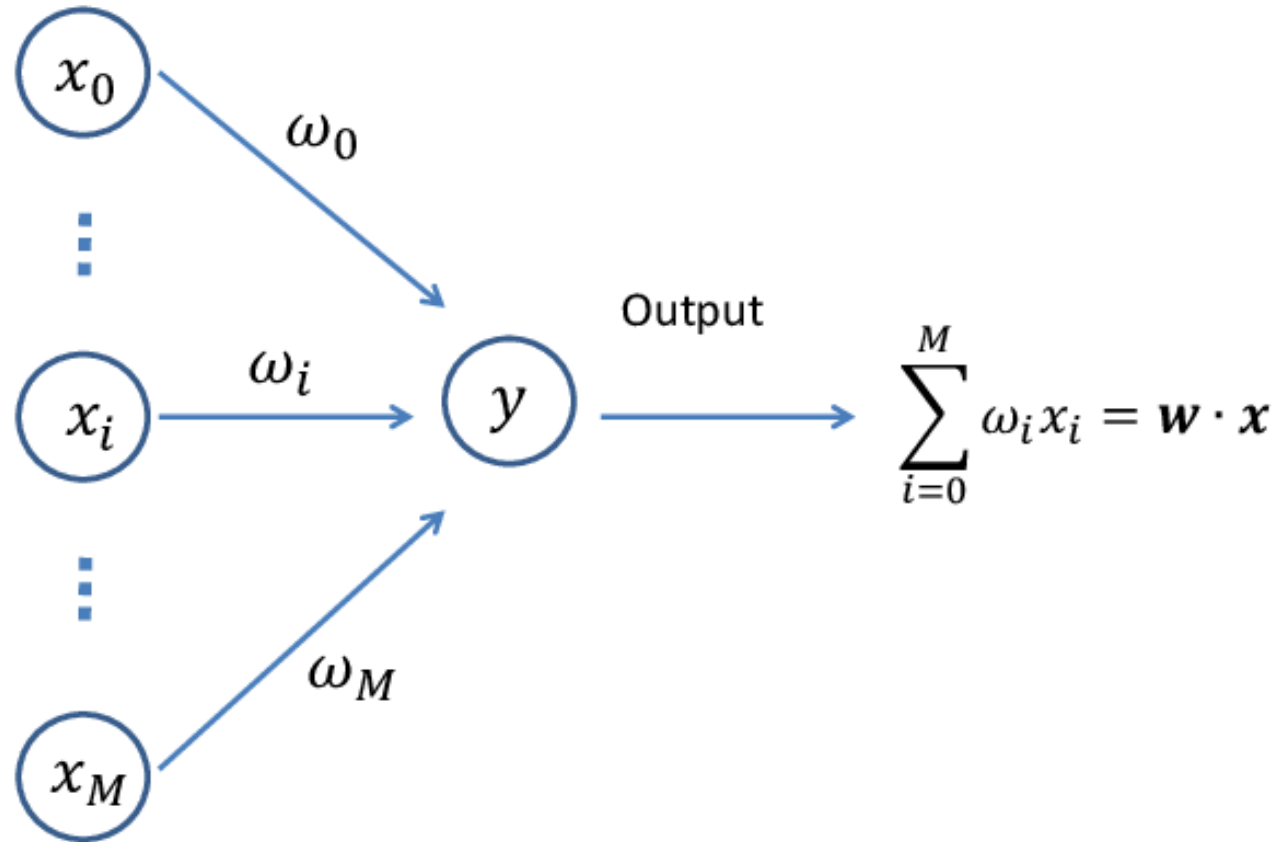
The networks are built from individual parts approximating neurons, typically called units or simply "neurons."



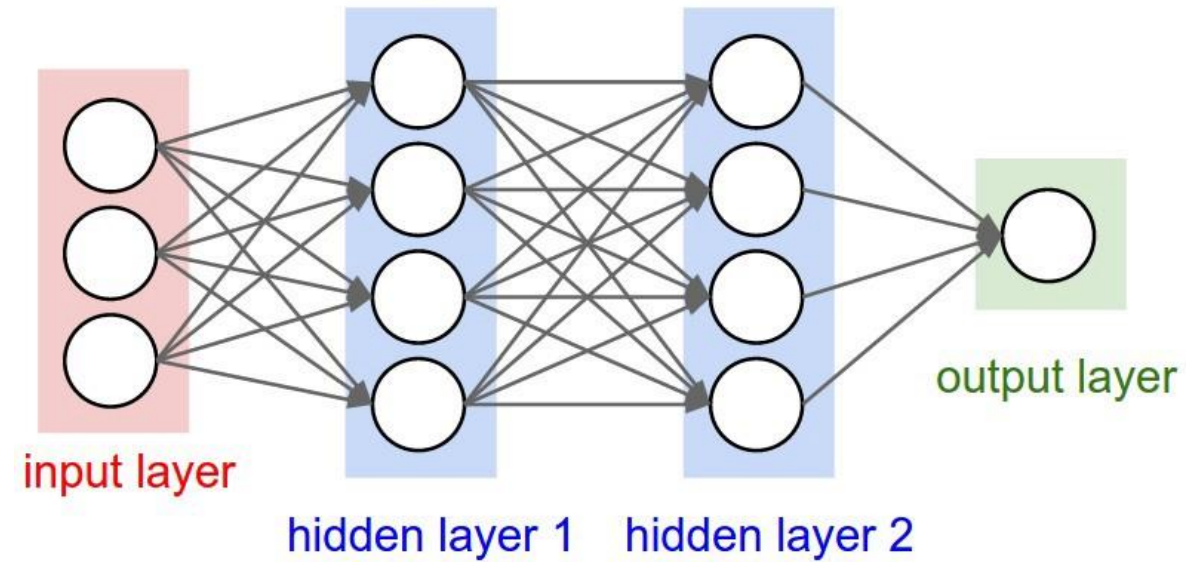
$$y = f(w_1x_1 + w_2x_2 + b)$$

$$y = f\left(\sum_i w_i x_i + b\right)$$

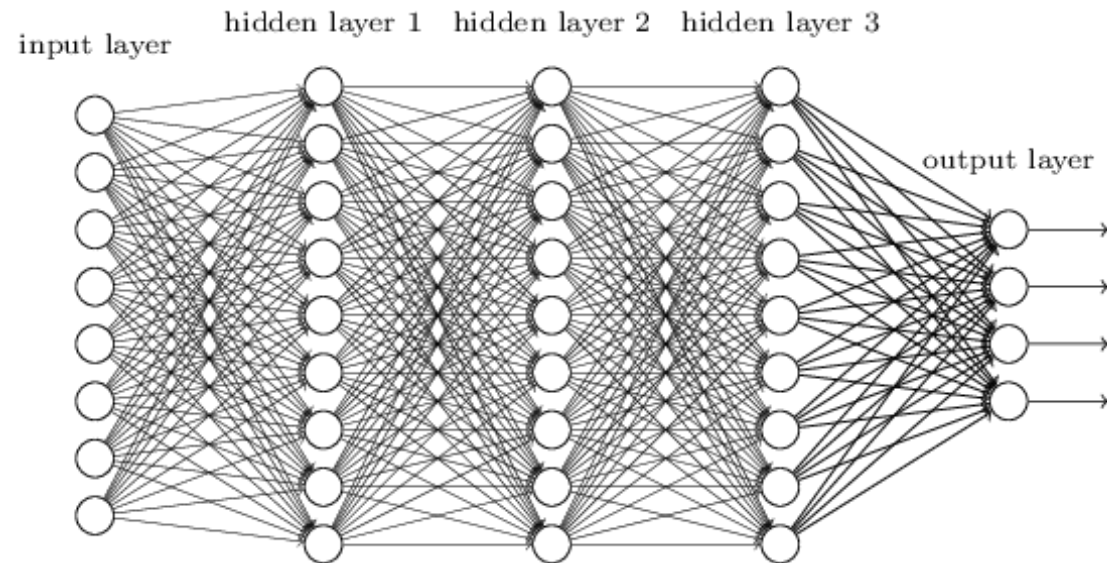
# One-Layer Neural Network: How does it work?



## The General Architecture of a Deep Neural Network



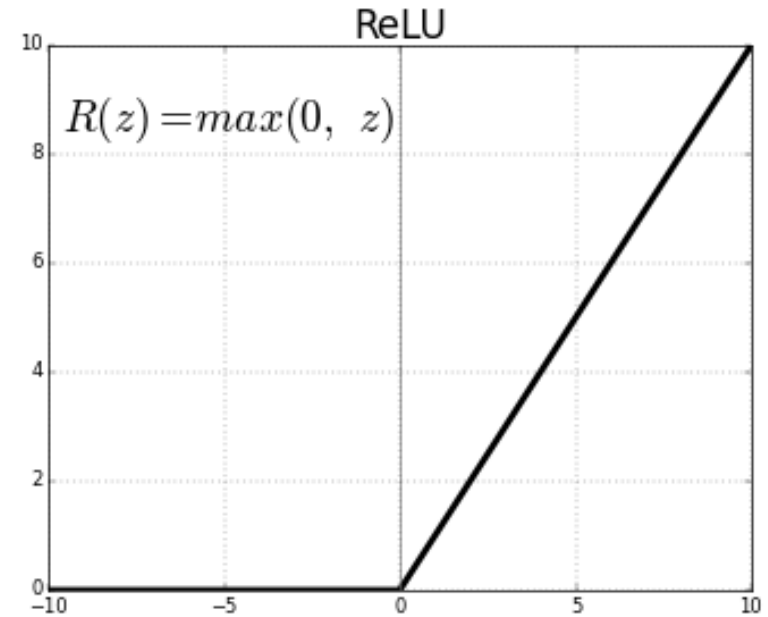
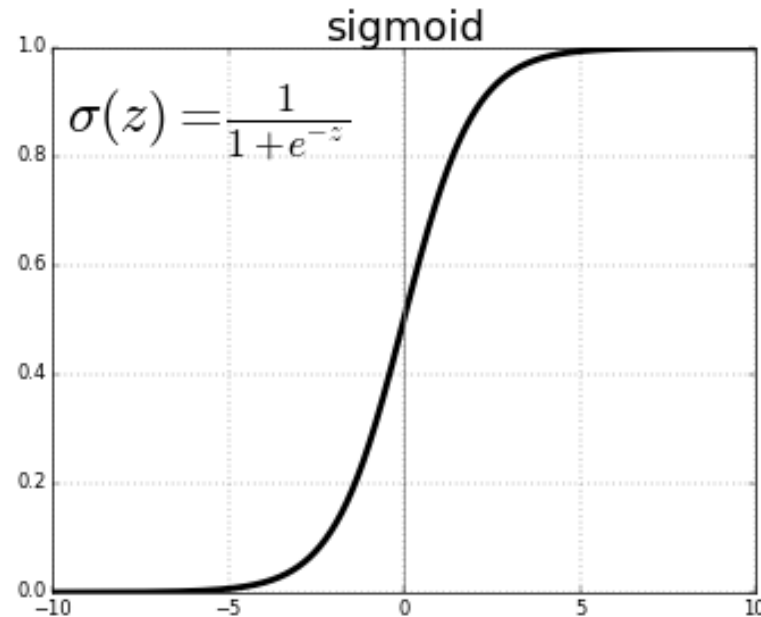
More hidden layers for more complicated tasks (NLP, Computer Vision, etc.)



# Activation Functions

Used to break the linearity of the matrix computations by compressing the network values between 0 and 1 or -1 and 1.

- ReLU
- Sigmoid
- Tanh
- Softmax





# More on Deep Neural Networks

Do not worry about these concepts, we will study them in more details as the course progress (top-down approach):

- A loss function (e.g., mean square error)
- Backpropagation
- Stochastic Gradient Descent
- Hyperparameters
- ...and others

# Google Colaboratory

- Google Colab is an online Notebook service with that provides a GPU for up to 12 hours per session for free.
- You only need to create a Google account (gmail) and access the GC from your Google Drive.
- Here's the formal "getting-Started" tutorial from Google:  
<https://colab.research.google.com/notebooks/welcome.ipynb>
- We will run this step-by-step in the class

# Introduction to PyTorch

## What is PyTorch?

### PyTorch Locally:

- To install **PyTorch** Locally, follow the instructions here:  
<https://pytorch.org/get-started/locally/>
- You should be able to manage Virtual Environment on your own system. I recommend using **Conda**. Follow the instructions here:  
<https://conda.io/en/latest/>
- Install **NumPy** and **Jupyter Notebooks** (should be easy to install from within conda –if not already installed with the package)

# Introduction to PyTorch

If you do not have a local GPU, I recommend using Google Colaboratory.

<https://colab.research.google.com/notebooks/welcome.ipynb>

If you would like to use other paid services, check:

- AWS: <https://docs.aws.amazon.com/dlami/latest/devguide/gpu.html>
- GCP: <https://cloud.google.com/gpu/>
- FolydHub: <https://www.floydhub.com/>

# Introduction to PyTorch

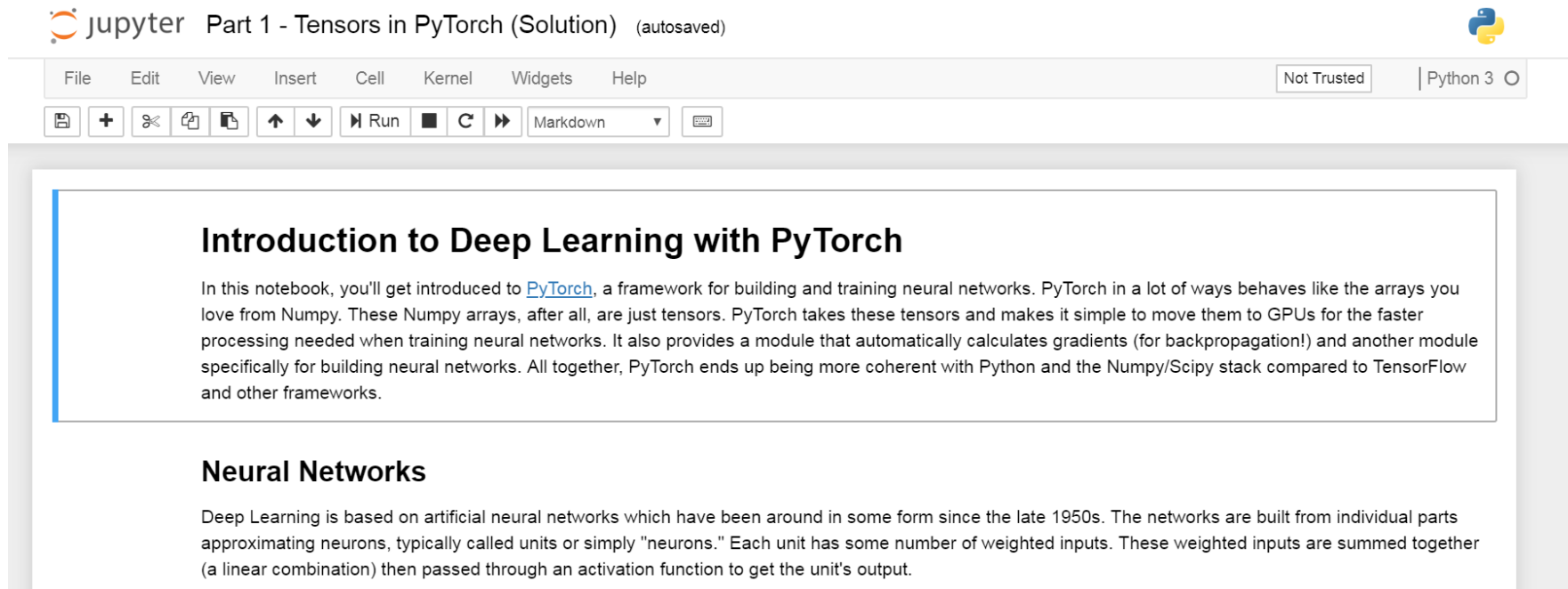
- The best resource to start learning PyTorch, is to learning by example from the following Notebooks on GitHub (these are used for teaching a Deep Learning Nanodegree in Udacity). Following is a modified version to fit our classroom.
- Clone the following repository (It includes more teaching materials and ICP 1):

<https://app.box.com/file/469880222759>

# Building a Deep Neural Network (DNN) from scratch:

## Notebook1:

*deep-learning-v2-pytorch >> intro-to-PyTorch >> Part1 (tensors in PyTorch)*



jupyter Part 1 - Tensors in PyTorch (Solution) (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run

### Introduction to Deep Learning with PyTorch

In this notebook, you'll get introduced to [PyTorch](#), a framework for building and training neural networks. PyTorch in a lot of ways behaves like the arrays you love from Numpy. These Numpy arrays, after all, are just tensors. PyTorch takes these tensors and makes it simple to move them to GPUs for the faster processing needed when training neural networks. It also provides a module that automatically calculates gradients (for backpropagation!) and another module specifically for building neural networks. All together, PyTorch ends up being more coherent with Python and the Numpy/Scipy stack compared to TensorFlow and other frameworks.

### Neural Networks

Deep Learning is based on artificial neural networks which have been around in some form since the late 1950s. The networks are built from individual parts approximating neurons, typically called units or simply "neurons." Each unit has some number of weighted inputs. These weighted inputs are summed together (a linear combination) then passed through an activation function to get the unit's output.

# ICP1 (Exercise Part1 + Part2)

**Part 1:** Build a DNN using Tensors

**Part2:** Build a DNN using PyTorch nn module  
and download the MNIST dataset