ECE499/ECE590
Machine Learning for Embedded Systems
(Fall 2021)

Lecture 1: Course Information and Introduction to Machine Learning

Weiwen Jiang, Ph.D.

Electrical and Computer Engineering

George Mason University

wjiang8@gmu.edu

Agenda

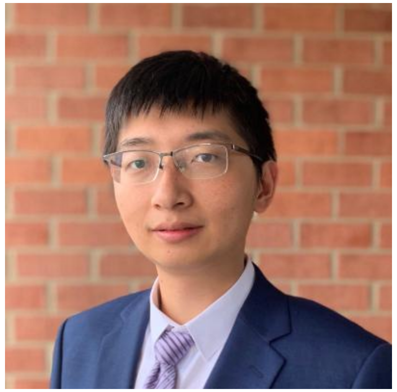
- Course Information
  - Logistics
  - Motivation
  - Overview
- Introduction to Artificial Neuron and MLP

Course Logistics

Course Resources

- **Course Website:**
  - https://jqub.github.io/2021/09/01/ML4Emb/
  - Slides, readings, and documents will be posted here!
  - Assignments will be posted here! (Until Blackboard can be used)
- **Blackboard:**
  - Assignments will be posted and submitted here!
- **Piazza:**
  - Online discussion, shared documents, announcements.
  - Do NOT upload codes.

About Me.



Dr. Weiwen Jiang

- **Background**
  - Researcher at University of Pittsburgh (2017-2019)
  - Postdoc at University of Notre Dame (2019-2021)
  - George Mason University (2021 - present)
- **Research Interests**
  - HW/SW Co-Design
  - Quantum Machine Learning
- **Contacts:**
  - wjiang8@gmu.edu
  - (412)427-0695
  - https://jqub.github.io/

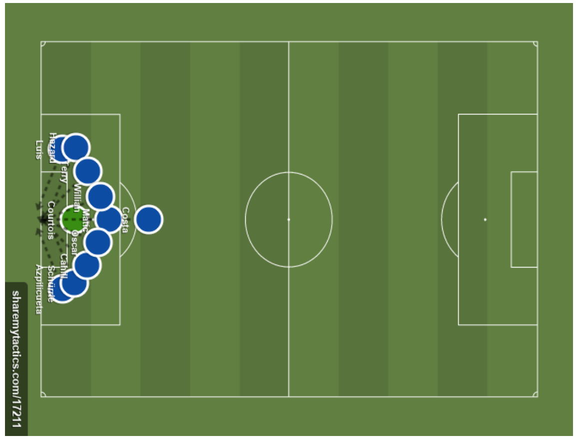
Teaching Assistant

XXX

XXX@gmu.edu

Office Hours: TBD

Lecture-Presentation-Lab Hours



10-0-0
(No!)

4-3-3
(Yes!)

Good Stuff

- No quizzes
- No mid-terms
- No finals
- Contents driven by demand and interest

"Bad" Stuff

- You'll have to make presentation or critiques
- You'll have to hand-on labs
- You'll have to work on a final project

- Eventually, they will do you good!

Grading Policy

Undergraduate (ECE 499)

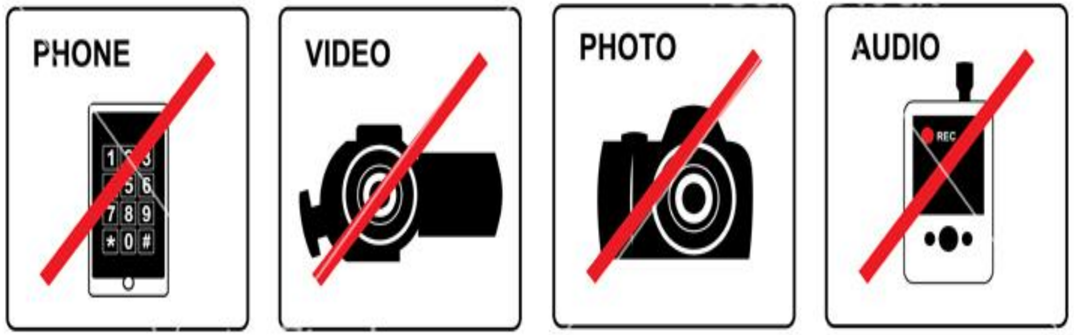
• Homework & Labs	50%
• Paper Critiques	10%
• Project progress review	10%
• Project final review	30%

Graduate (ECE 590)

• Homework & Labs	50%
• Research paper presentation	20%
• Project progress review	10%
• Project final review/report	20%

You have been warned.
Zero tolerance!

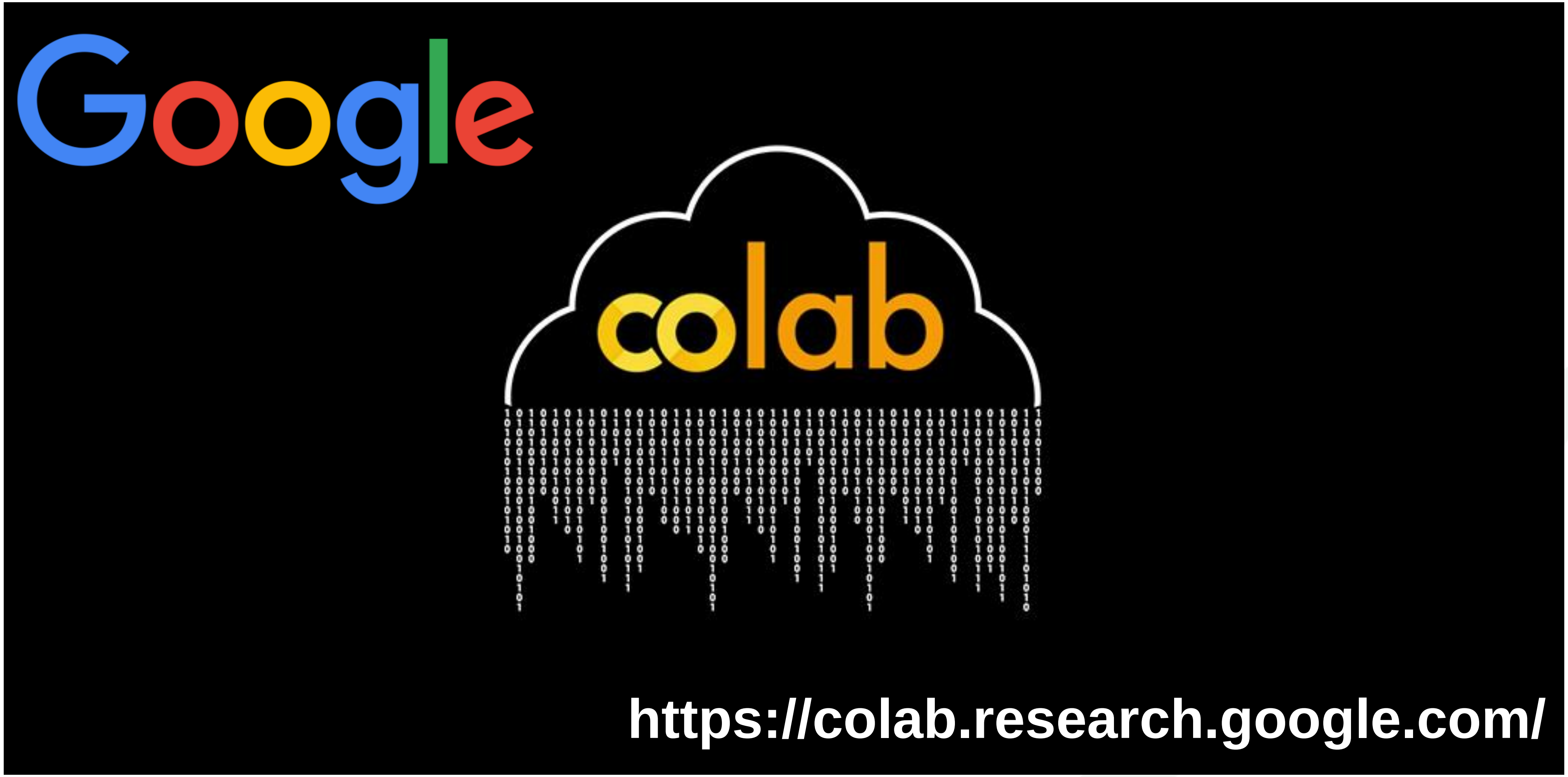
- **Lecture content and materials should NOT go online without explicit permission**



- **No plagiarism!**

The most common sense of way interpreting no plagiarism:
You need to DO your work.

ML Programming Platform



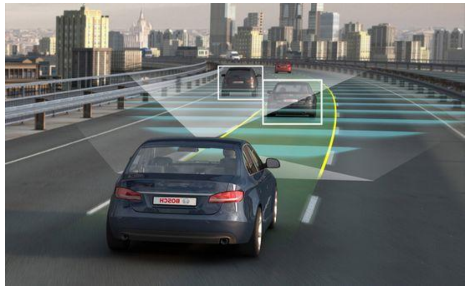
Course Motivation


"MACHINE LEARNING WILL
AUTOMATE JOBS THAT
MOST PEOPLE THOUGHT COULD ONLY BE
DONE BY PEOPLE." ~DAVE WATERS.

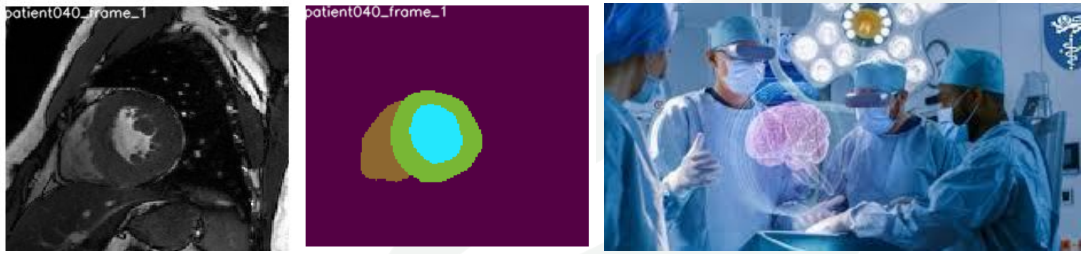
ML Applications

Game Play

Autonomous Driving

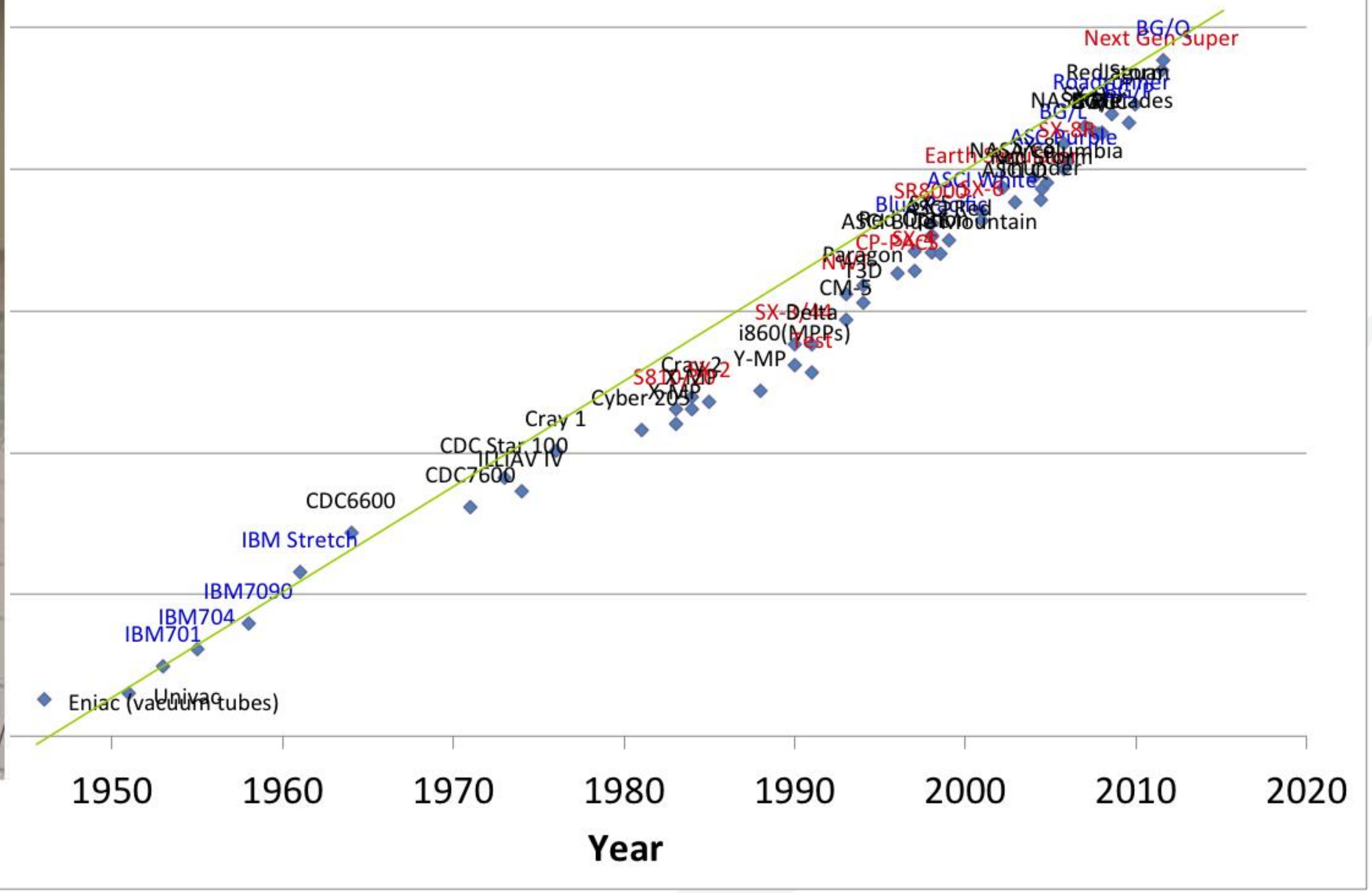


Medical AI

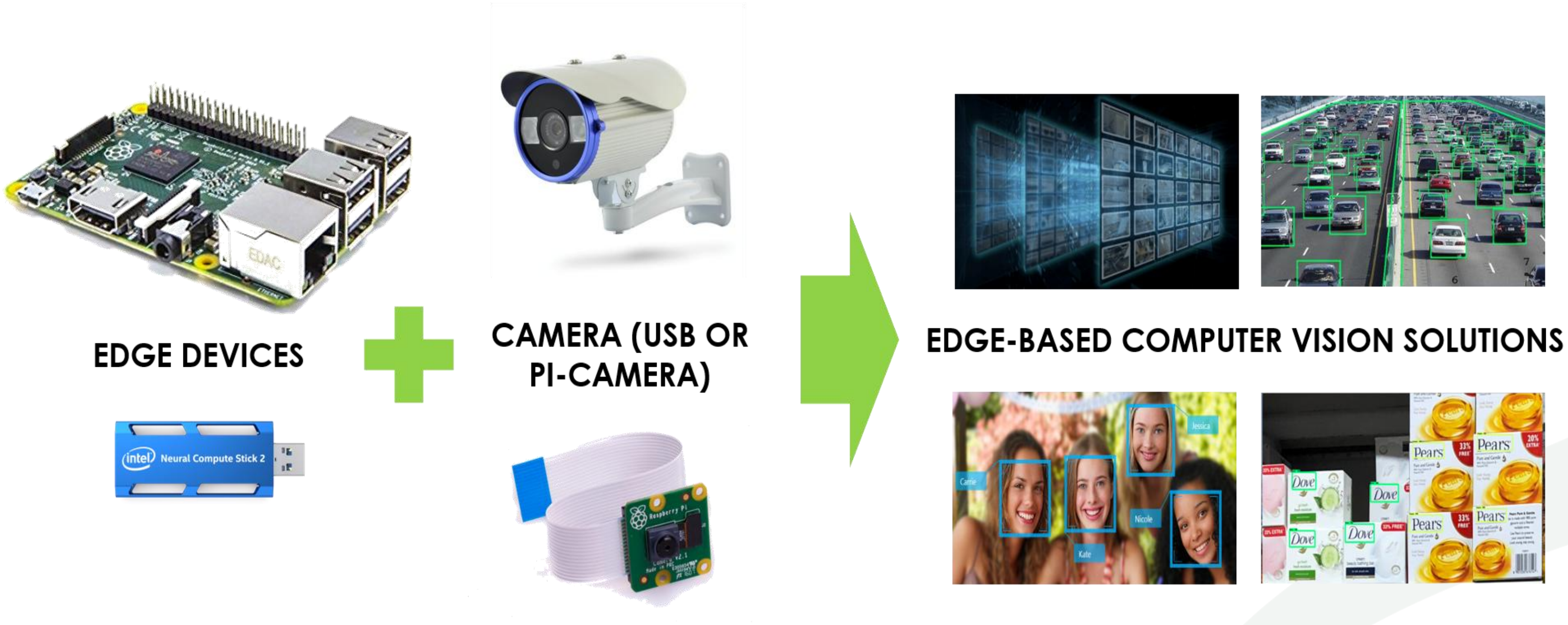


Race of Computer Powers Enables ML

Credit: IBM Blue Gene/Q Supercomputer



Machine Learning on the Edge



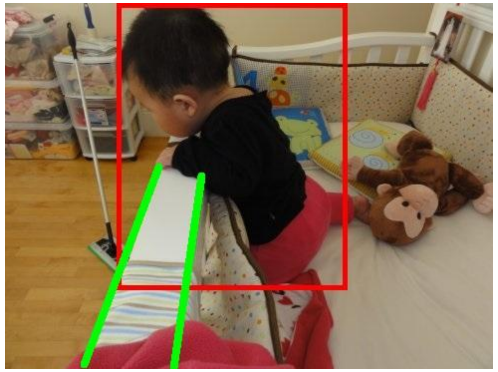
StrataHive
Enhancing the Digital Experience

Why on the Edge?

- Latency Problem

- Delay & Latency
- Speed
- WiFi Access

- Privacy Leakage



- Data uploaded to the server
- Privacy concerns

- Cost/energy efficiency considerations

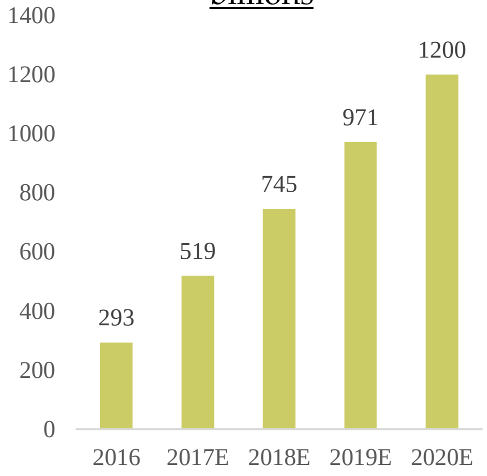
Why on the Edge?

AI chip bearing artificial intelligence algorithm, billion dollar market opportunity

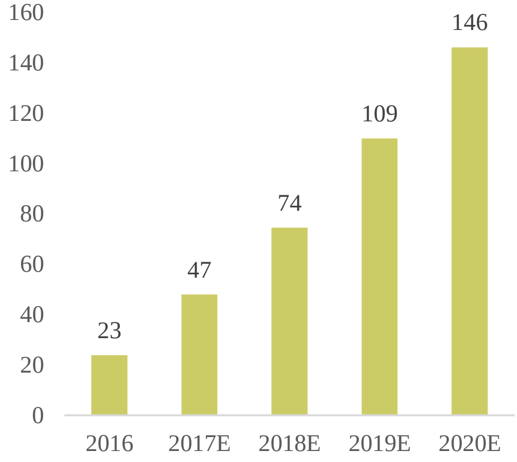
Big data, Maturing algorithm,
Core processor for AI Chip is the key

- **Data**
  - Massie data and frequent human computer interaction
- **Calculation**
  - Engineering methods and simulation methods require the use of convolutions.
- **Hardware**
  - Insufficient calculation, AI chips provide computing power: GPU, FPGA, ASIC, TPU

Global AI market exceeds US$100 billions



AI chip market will exceed US$10 billions



14.6 billions

Smart end devices

Apple, Qualcomm, Spreadtrum, HiSilicon, Mediatek, annual volume

9 billions

Home appliance

Smart appliance, digital TV, set top box, game console, VR/AR annual volume

200+ billions

Autopilot

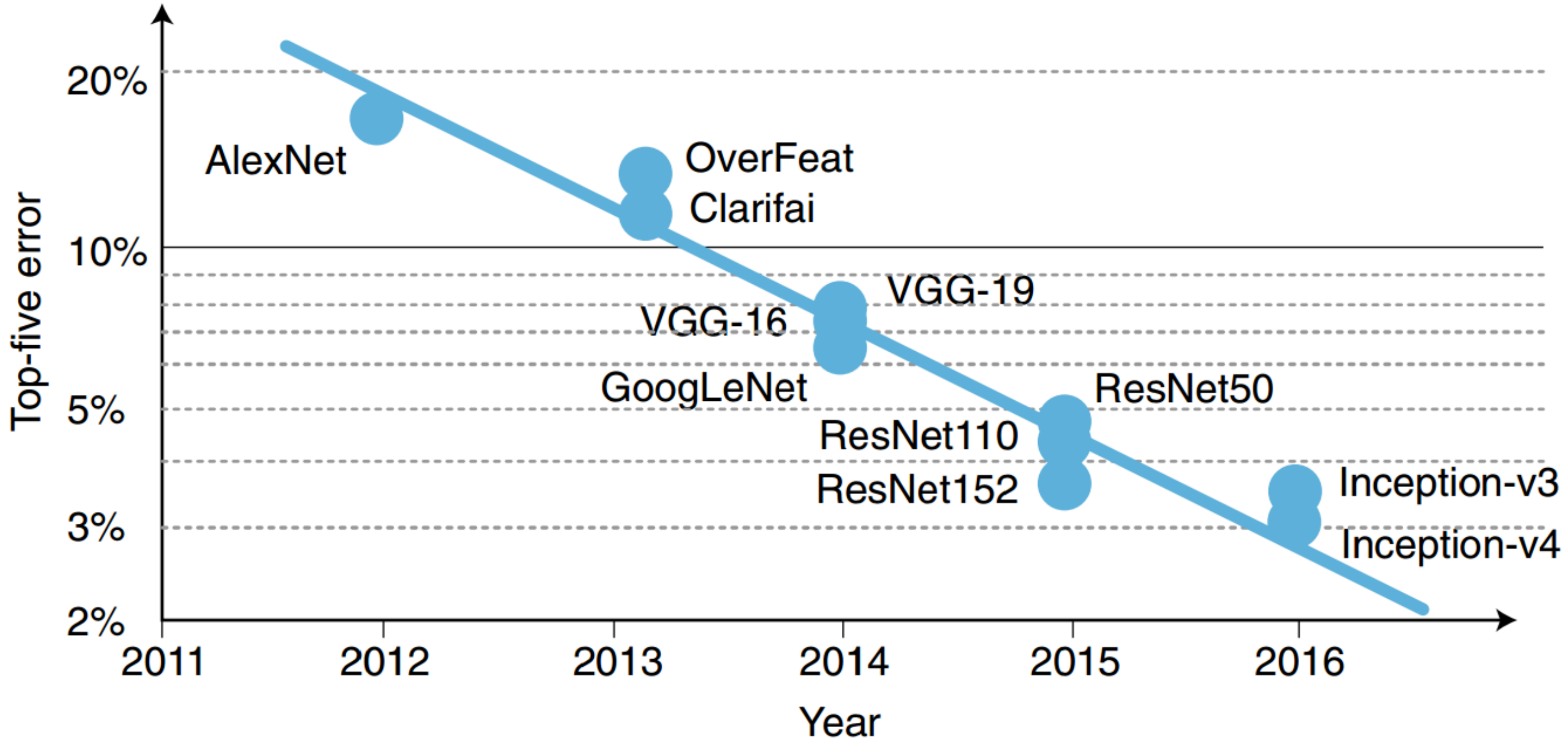
ADAS chip market potential

Global AI Chip Market is Expanding!

Source: CCID, NVIDIA, Intel, gartner, CITIC Securities

Challenges of ML on Edge

Error rate improved exponentially

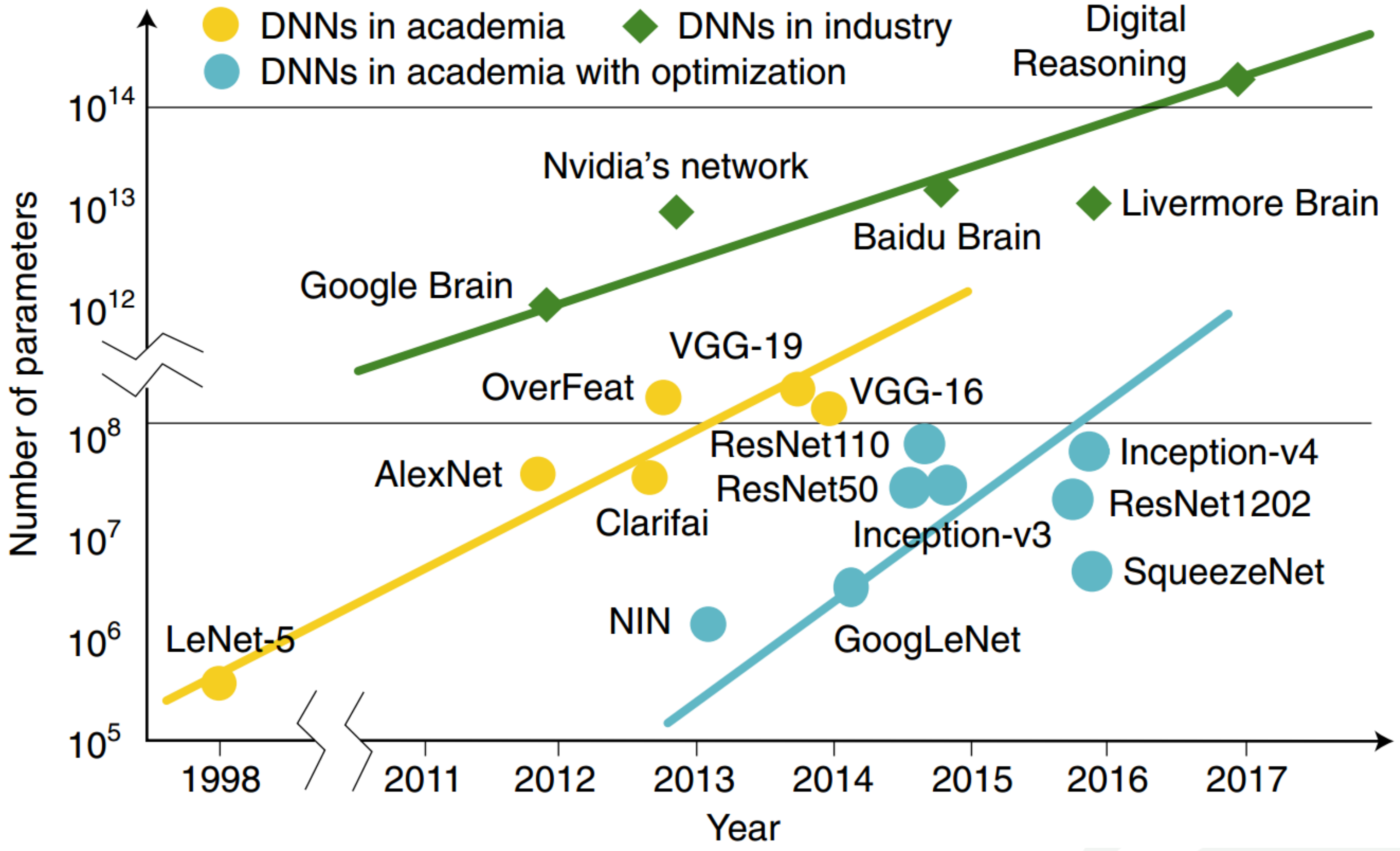


Error rate decreases by approximately 30% each year

Xu, Xiaowei, et al. "Scaling for edge inference of deep neural networks." Nature Electronics 1.4 (2018): 216.

Challenges of ML on Edge

Size of machine learning model also increases exponentially

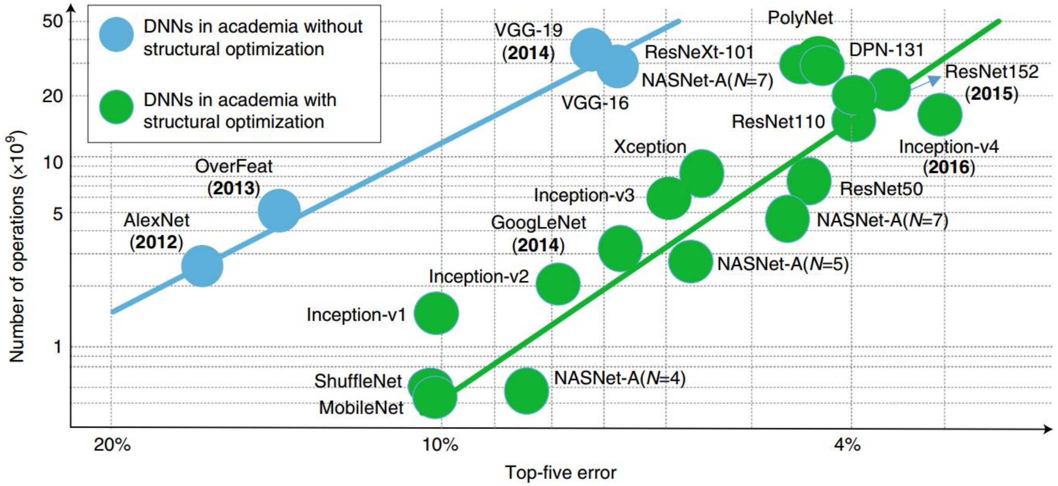


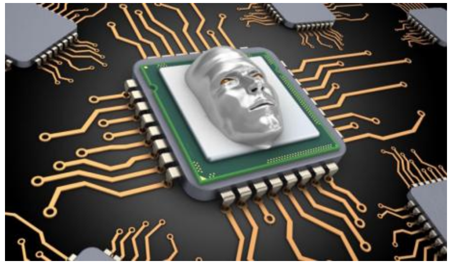
Xu, Xiaowei, et al. "Scaling for edge inference of deep neural networks." Nature Electronics 1.4 (2018): 216.

Challenges of ML on Edge

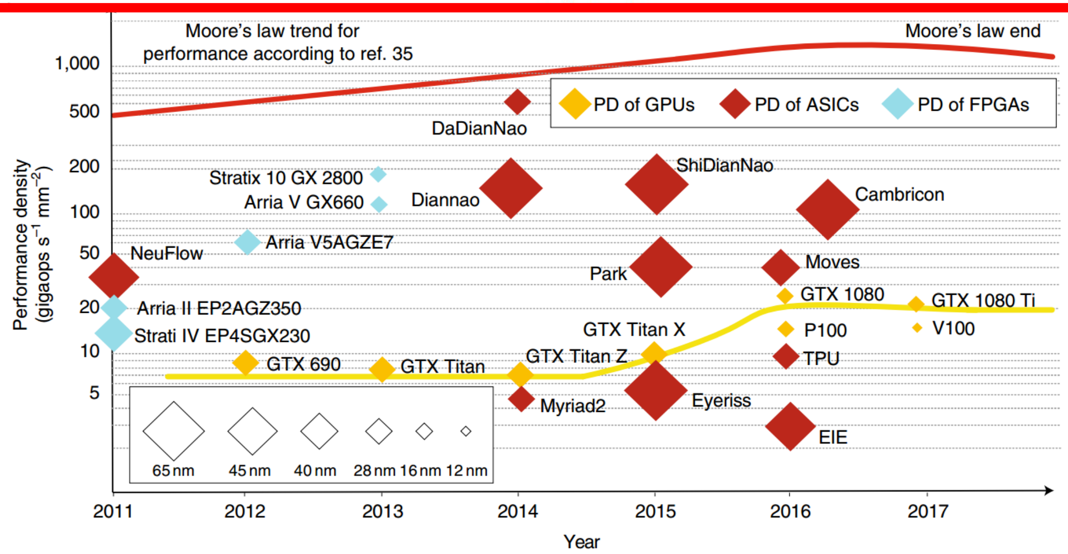
Computing performance gap

Number of DNN **operations** increases exponentially





Performance density almost stops increasing

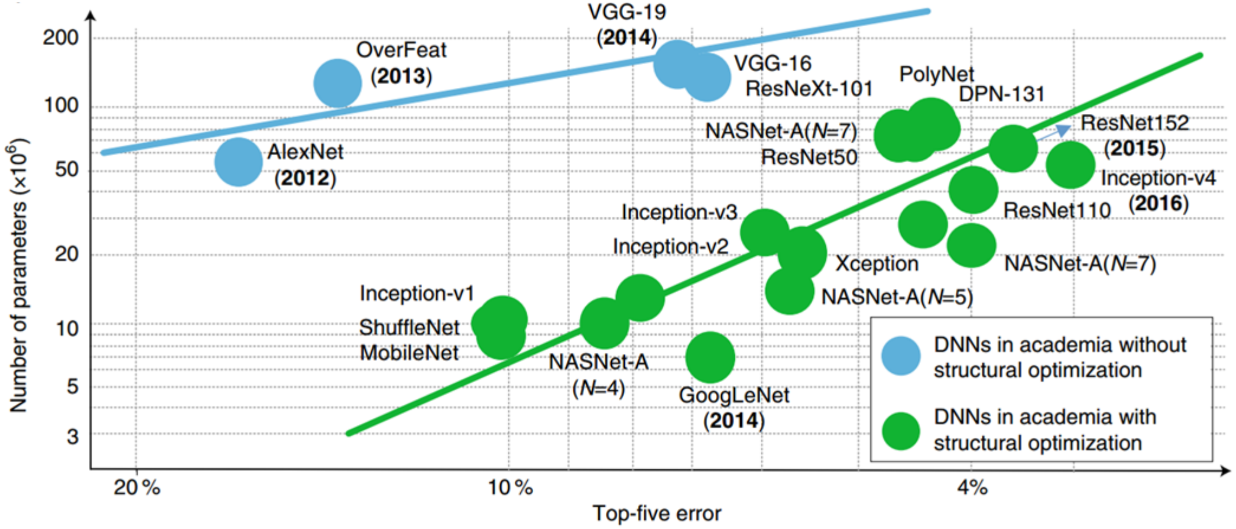


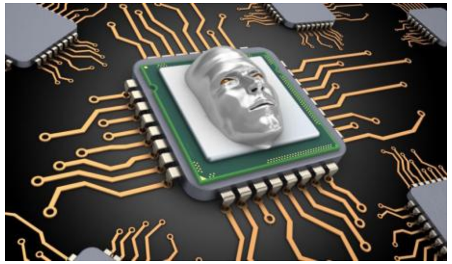
Xu, Xiaowei, et al. "Scaling for edge inference of deep neural networks." Nature Electronics 1.4 (2018): 216.

Challenges of ML on Edge

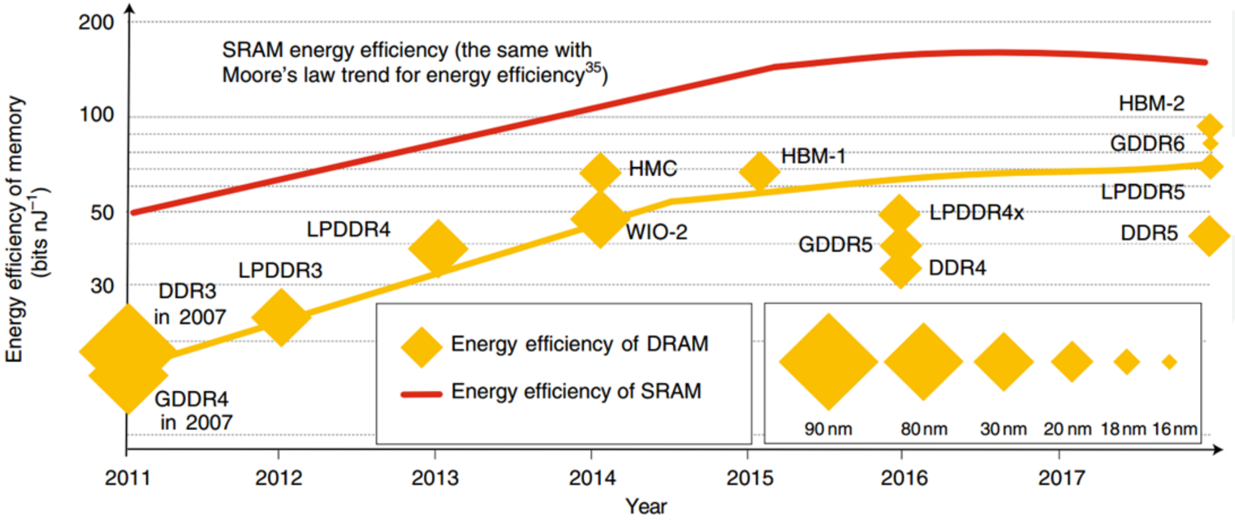
Storage energy efficiency gap

Number of DNN **parameters** increases exponentially





Energy efficiency of memory almost stops increasing

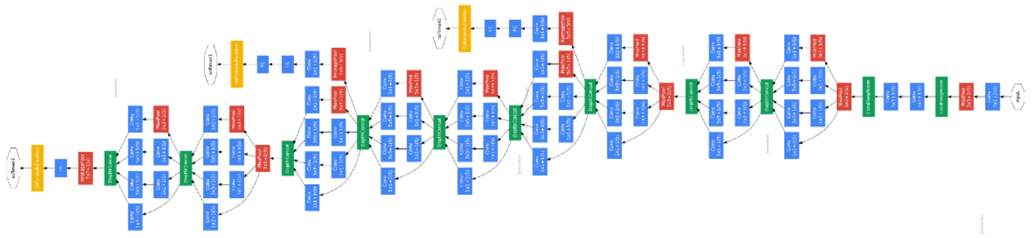


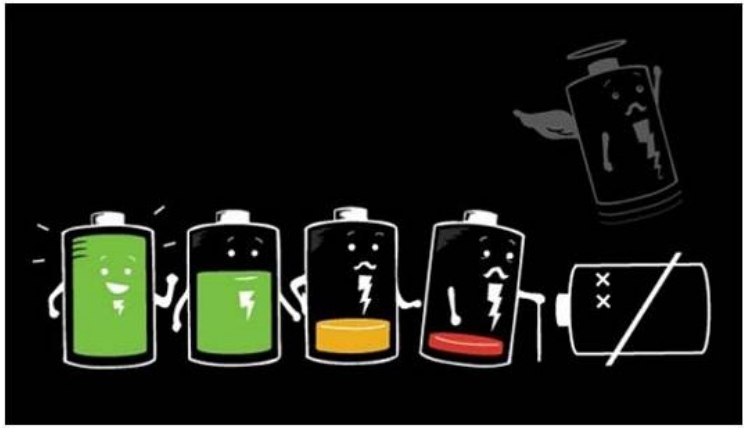
Xu, Xiaowei, et al. "Scaling for edge inference of deep neural networks." Nature Electronics 1.4 (2018): 216.

Course Overview

What is This Course About?

Open question on Machine Learning for Embedded Systems!





Machine Learning

- High computation complexity
- High storage complexity

V.S.

Embedded Systems

- Low power
- Small on-chip memory
- Low bandwidth
- Real-time requirements

How to overcome the limitations of embedded systems?

What is This Course About?

Software side: AI/ML/DL?

Artificial Intelligence (AI)

[Definition] AI is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals, which involves consciousness and emotionality.

What is This Course About?

Software side: AI/ML/DL?

Artificial Intelligence (AI)

Machine Learning (ML)

[Definition] ML is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of AI.

ECE 527: Learning From Data

What is This Course About?

Software side: AI/ML/DL?

Artificial Intelligence (AI)

Machine Learning (ML)

Deep Learning (DL)

[Definition] DL is a class of ML Algorithms that uses multiple layers to progressively extract higher-level features from the raw input.

- Computer Vision
- Natural Language
- Games
- …

What is This Course About?

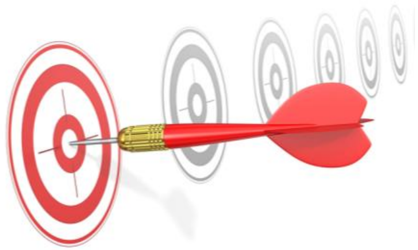
Overview: software side

Software

Application
- Computer Vision
- Natural Language
- Games
- …

ML/DL Algorithms
- AN
- MLP
- CNN
- RNN
- LSTM
- RL
- Transformer
- MLP-Mixer
- …

Optimization
- Network Inference
- Network Training



High Accuracy

What is This Course About?

Hardware side: from cloud to edge

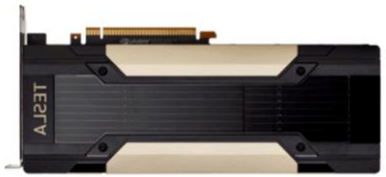
ECE 350: Embedded Systems and Hardware Interfaces

Mobile Device

Microcontroller

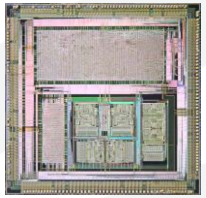
General Purpose Computing

Cloud GPU/CPU



Customized Computing

FPGA
Field-Programmable Gate Array



ASIC
Application Specifical Integrate Circuit

ECE 231: Digital System Design

What is This Course About?

Overview

Software

Application

- Computer Vision
- Natural Language
- Games
- …

ML/DL Algorithms

AN | MLP | CNN
RNN | LSTM | RL
Transformer | MLP-Mixer
…

Optimization

- Network Inference
- Network Training
- Model Compression
- Network Design

Hardware

Embedded Systems

ECE 699: Hardware Accelerators for Machine Learning

Low-Power 

 **Low-Latency**

What is This Course About?

Overview

Software

Hardware

Application

- Computer Vision
- Natural Language
- Games
- …

ML/DL Algorithms

- AN
- MLP
- CNN
- RNN
- LSTM
- RL
- Transformer
- MLP-Mixer
- …

Optimization

- Network Inference
- Network Training
- Model Compression
- Network Design

Embedded Systems

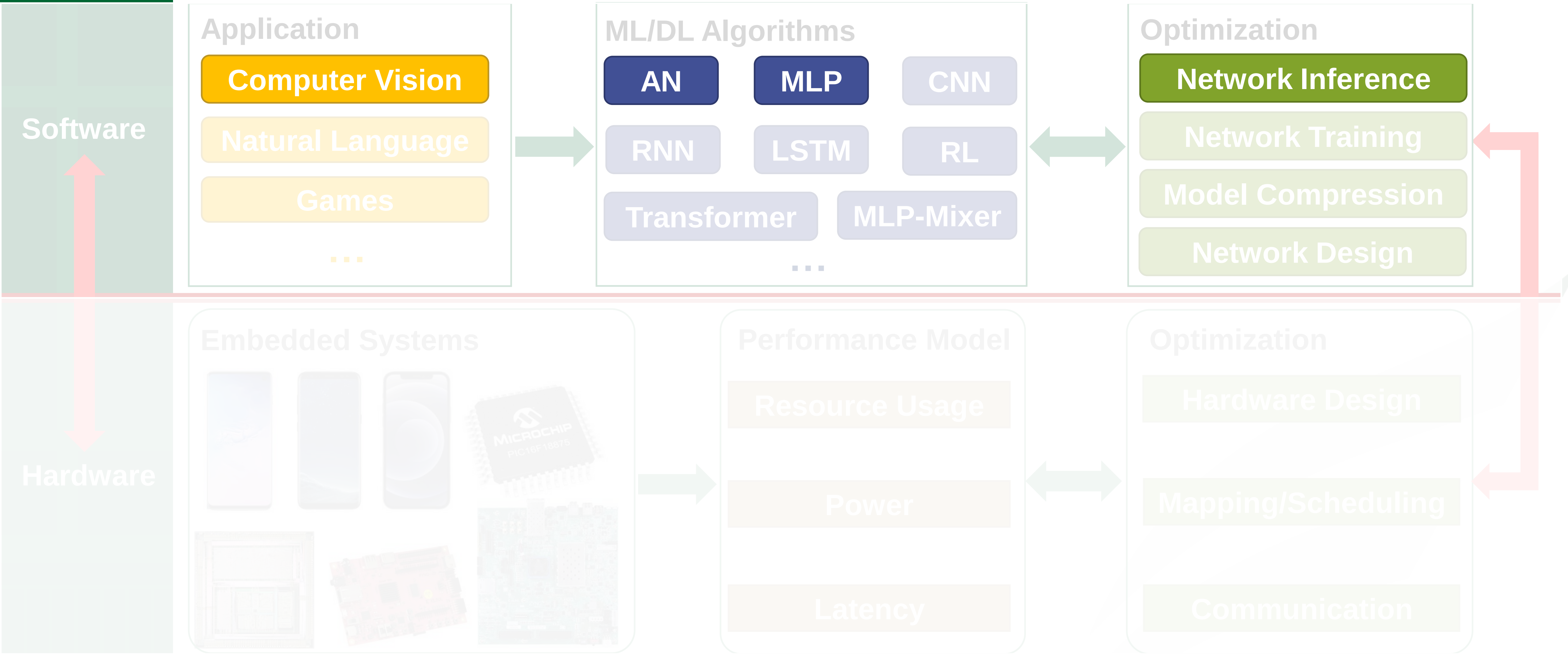
Performance Model

- Resource Usage
- Power
- Latency

Optimization

- Hardware Design
- Mapping/Scheduling
- Communication

Week 1: Introduction to Artificial Neuron and MLP



Week 2: From Inference to Training, From MLP to CNN

Software

Hardware

Application
- Computer Vision
- Natural Language
- Games
- ...

ML/DL Algorithms
- AN
- MLP
- CNN
- RNN
- LSTM
- RL
- Transformer
- MLP-Mixer
- ...

Optimization
- Network Inference
- Network Training
- Model Compression
- Network Design

Embedded Systems

Performance Model
- Resource Usage
- Power
- Latency

Optimization
- Hardware Design
- Mapping/Scheduling
- Communication

Week 3-4: From CV to NLP

Software

Application
- Computer Vision
- **Natural Language**
- Games
- ...

ML/DL Algorithms
- AN
- MLP
- CNN
- **RNN**
- **LSTM**
- RL
- **Transformer**
- MLP-Mixer
- ...

Optimization
- **Network Inference**
- **Network Training**
- Model Compression
- Network Design

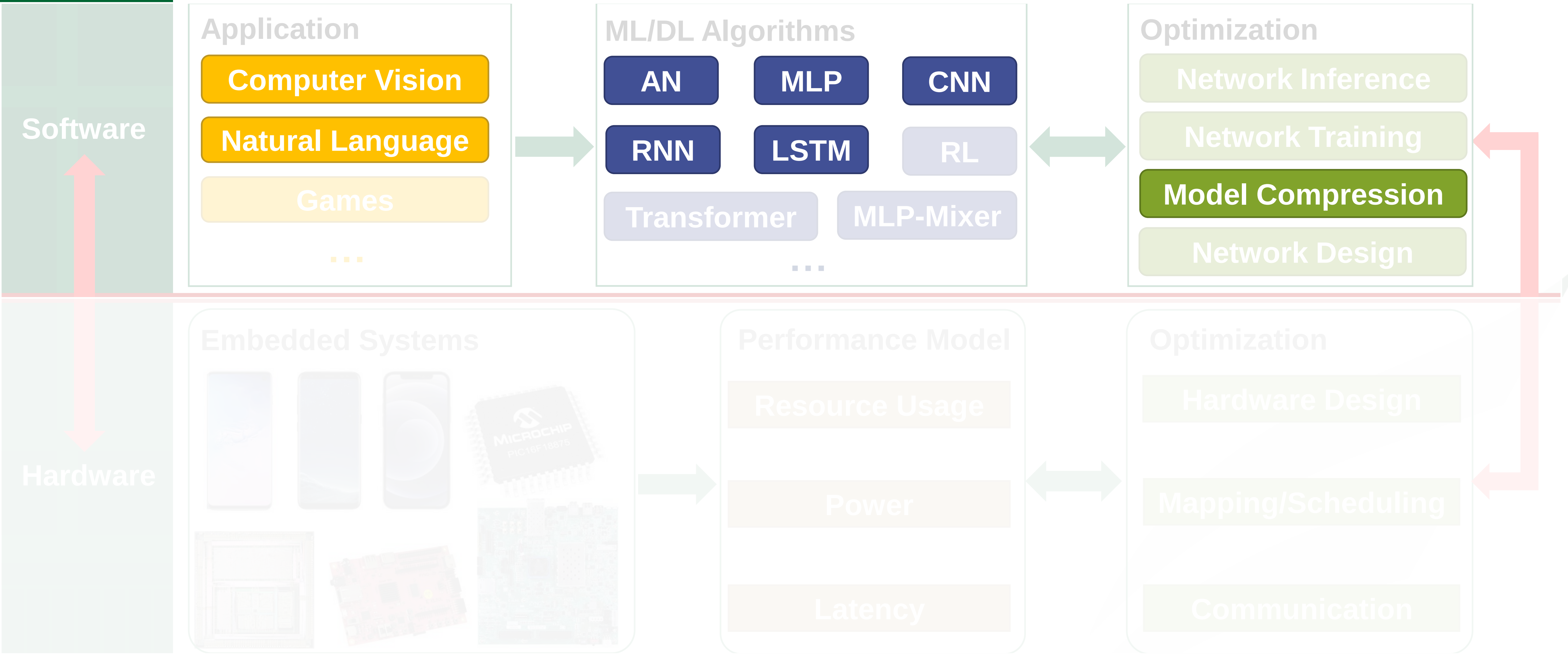
Hardware

Embedded Systems

Performance Model
- Resource Usage
- Power
- Latency

Optimization
- Hardware Design
- Mapping/Scheduling
- Communication

Week 5-6: Model Compression



Week 7: From Deep Learning to Deep Reinforcement Learning

Software

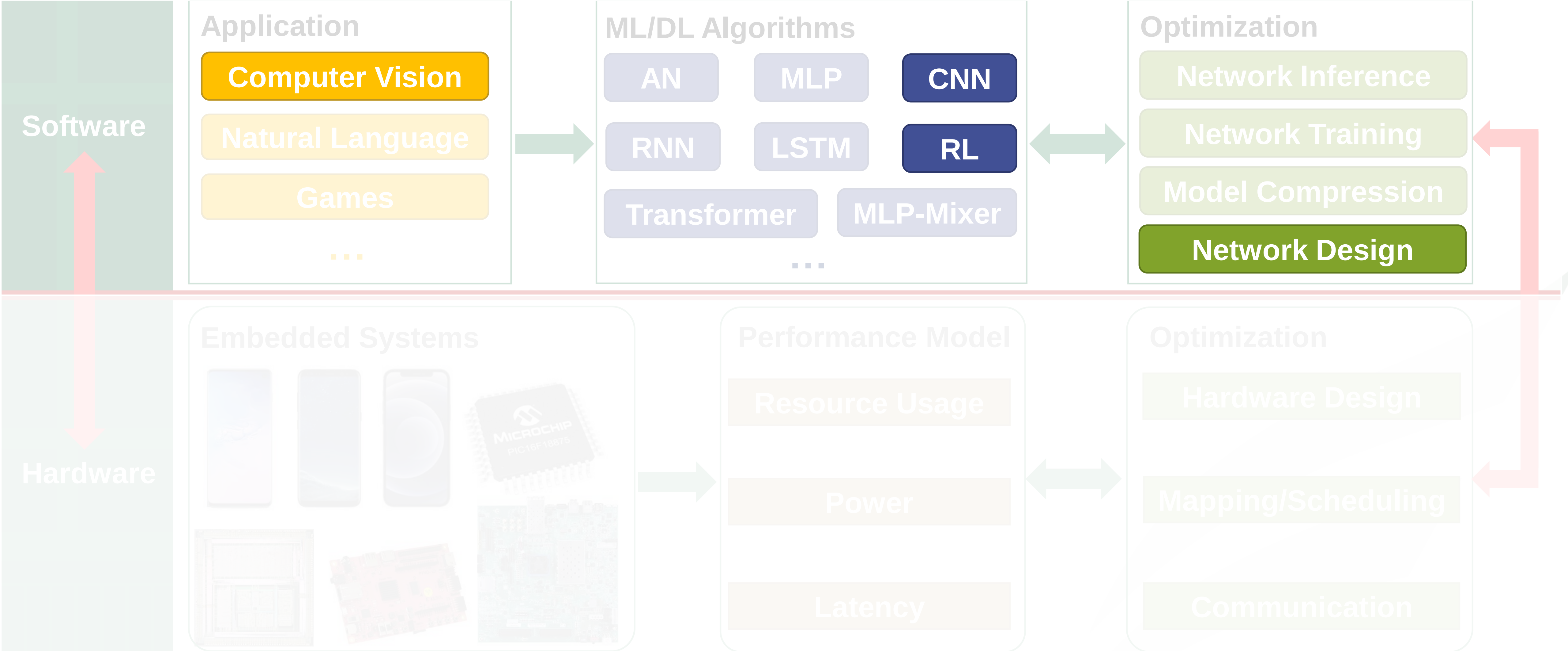
- Application
  - Computer Vision
  - Natural Language
  - **Games**
  - ...
- ML/DL Algorithms
  - AN
  - MLP
  - CNN
  - RNN
  - LSTM
  - **RL**
  - Transformer
  - MLP-Mixer
  - ...
- Optimization
  - **Network Inference**
  - **Network Training**
  - Model Compression
  - Network Design

Hardware

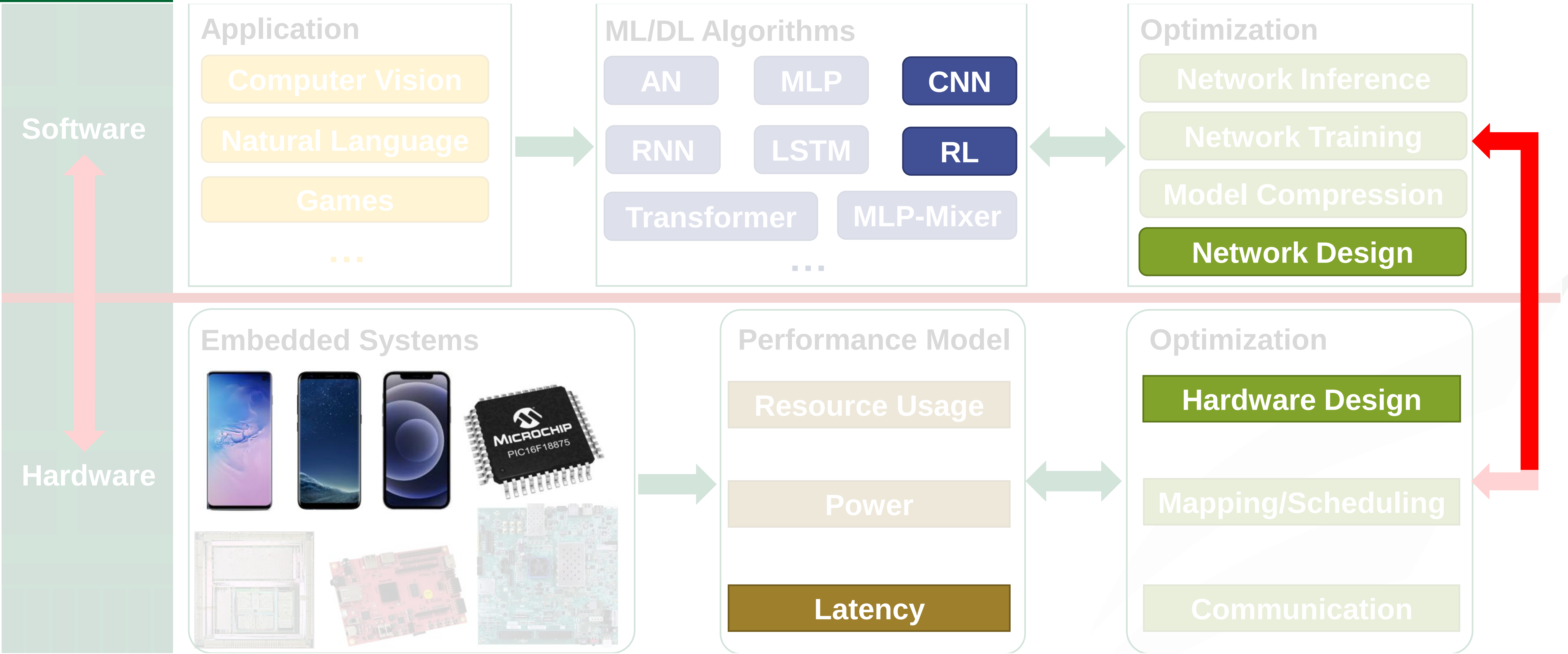
- Embedded Systems

- Performance Model
  - Resource Usage
  - Power
  - Latency
- Optimization
  - Hardware Design
  - Mapping/Scheduling
  - Communication

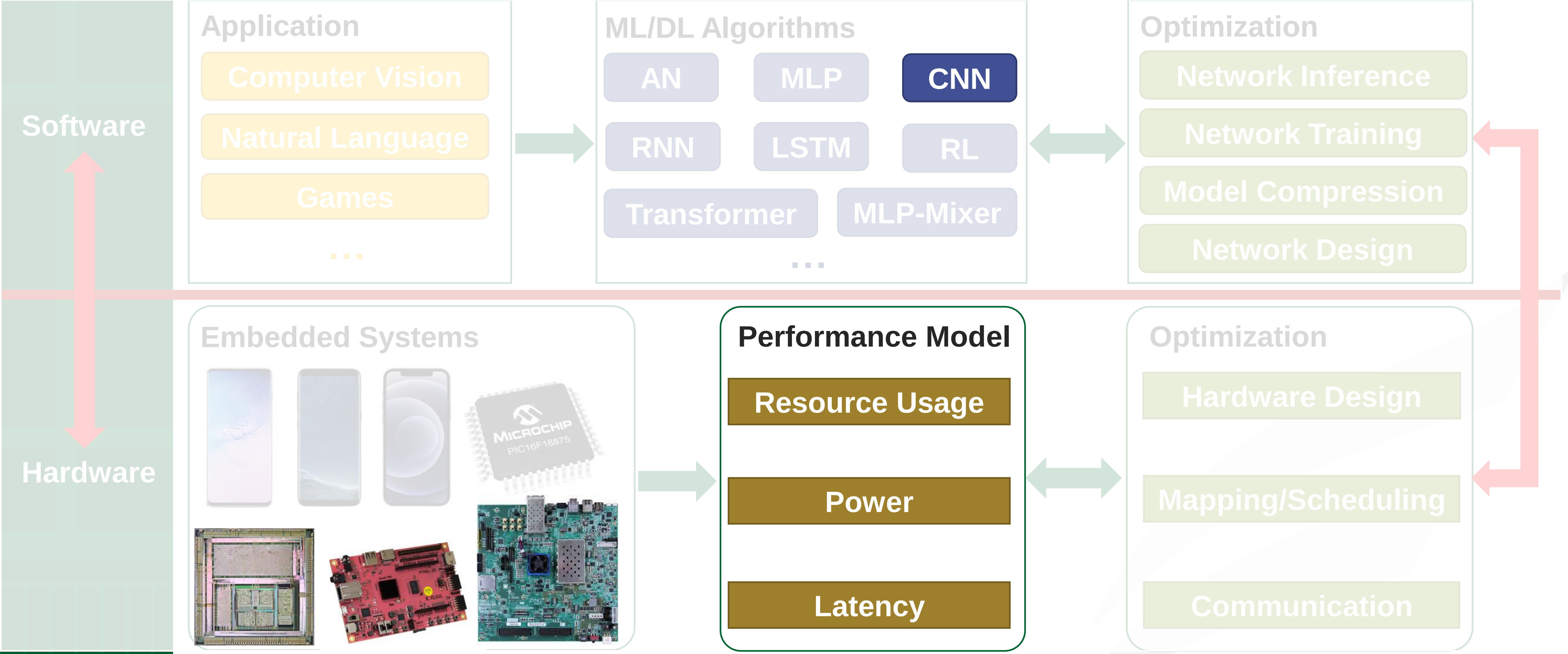
Week 8: RL-based Network Design (Neural Architecture Search)



Week 9: Hardware-Aware Neural Architecture Search



Week 10-11: ML Accelerator Design



Week 12-14: HW/SW Co-Design with Neural Architecture Search

Software

Hardware

Application

Computer Vision

Natural Language

Games

...

ML/DL Algorithms

AN | MLP | CNN

RNN | LSTM | RL

Transformer | MLP-Mixer

...

Optimization

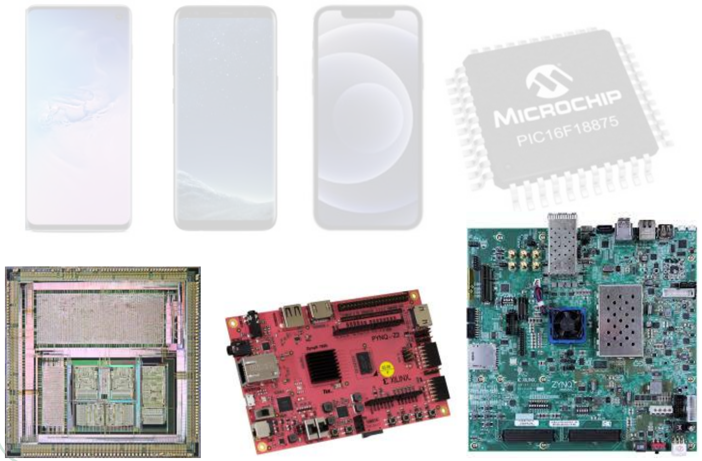
Network Inference

Network Training

Model Compression

Network Design

Embedded Systems



Performance Model

Resource Usage

Power

Latency

Optimization

Hardware Design

Mapping/Scheduling

Communication

Introduction to Artificial Neuron and MLP

Week 1: Introduction to Neural Network

Software

Application
- Computer Vision
- Natural Language
- Games
- ...

ML/DL Algorithms
- AN
- MLP
- CNN
- RNN
- LSTM
- RL
- Transformer
- MLP-Mixer
- ...

Optimization
- Network Inference
- Network Training
- Model Compression
- Network Design

Hardware

Embedded Systems

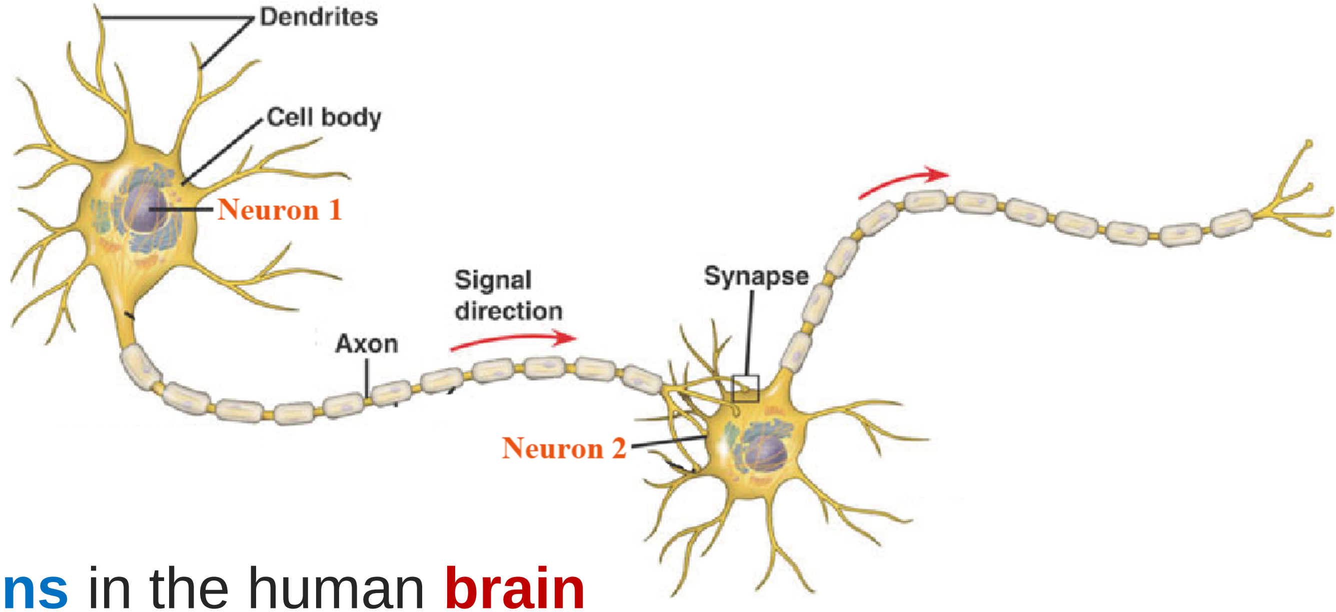
Performance Model
- Resource Usage
- Power
- Latency

Optimization
- Hardware Design
- Mapping/Scheduling
- Communication

Why Neural Networks

- **An emulation of the biological neural systems**
  - Parallel computation
  - Adaptive connections
- **Very different style from sequential computation**
  - Should be good for things that brains are good at (e.g., vision)
  - Should be bad for things that brains are bad at (e.g., 23 x 71)
- **To solve practical problems by using novel learning algorithms inspired by the brain**
  - Learning algorithms can be very useful even if they are not how the brain actually works.

Biological Neuron



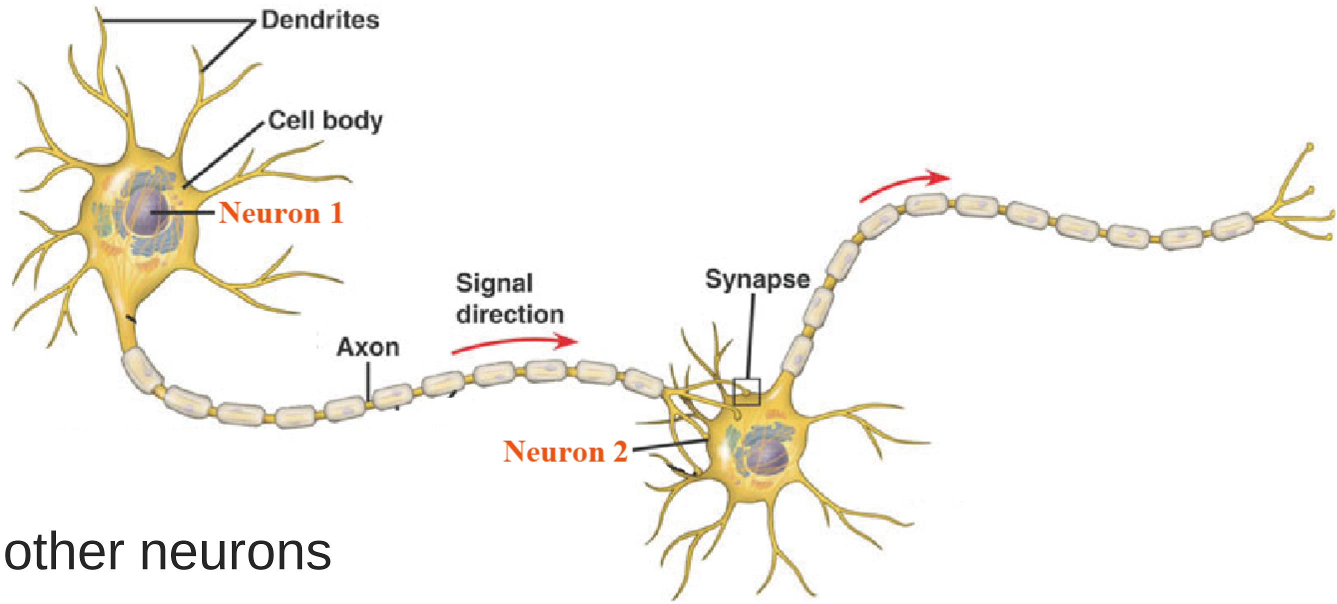
Human intelligence reside in the brain:

- Approximately **86 billion neurons** in the human **brain**
- The brain is a **network** of **neurons**, connected with nearly $10^{14} - 10^{15}$ **synapses**

How to equip intelligence in the machine?

- **To understand how the brain network is constructed**
- **To mimic the brain**

Biological Neuron



Neurons work together:

- **Cell body** process the information
- **Dendrites** receive messages from other neurons
- **Axon** transmit the output to many smaller branches
- **Synapses** are the contact points between axon (Neuron 1) and dendrites (Neuron 2) for message passing

Cell body receives input signal from **dendrites** and produce output signal along **axon**, which interact with the next neurons via **synaptic weights**

Synaptic weights are **learnable** to perform useful computations

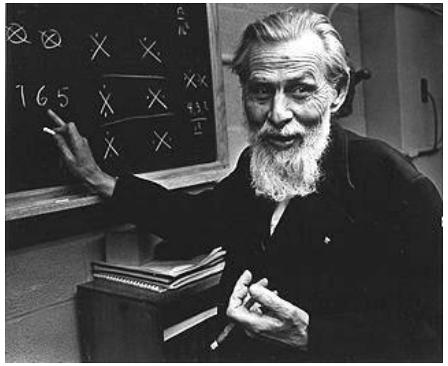
(e.g., Recognizing objects, understanding language, making plans, controlling the body.)

Artificial Neuron Design

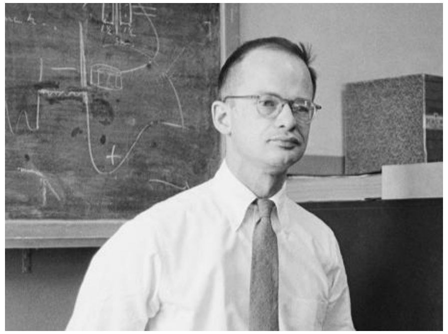
- **Idealized neuron models**
  - Idealization removes complicated details that are not essential for understanding the main principles.
  - It allows us to apply mathematics and to make analogies.

McCulloch-Pitts Neuron

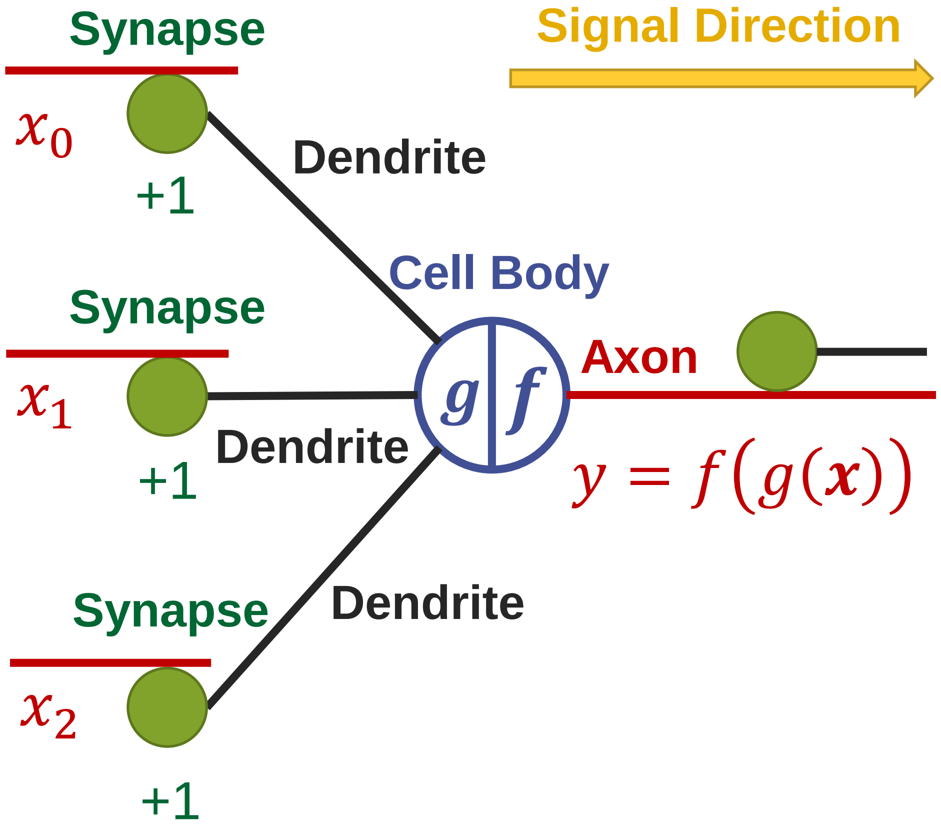
The first computational model of a biological neuron @ 1943



Warren McCulloch

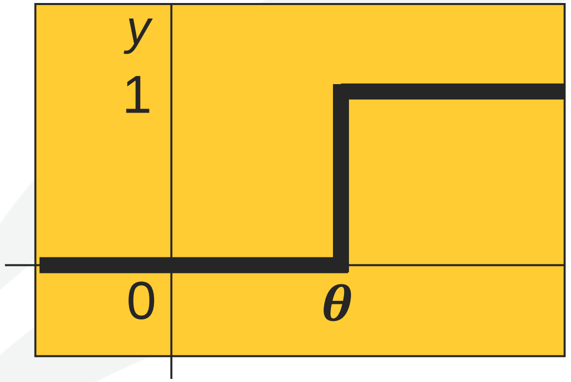


Walter Pitts



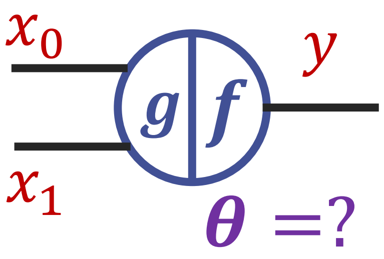
Assumptions:

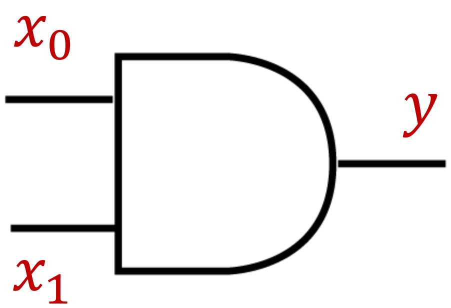
- Binary devices (i.e., $x_i \in \{0,1\}$ and $y \in \{0,1\}$)
- Identical synaptic weights (i.e., +1)
- Activation function f has a fixed threshold $\boldsymbol{\theta}$



McCulloch-Pitts Neuron

Boolean function 'AND' can be implemented by using MP Neuron



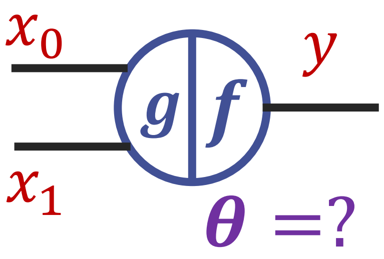


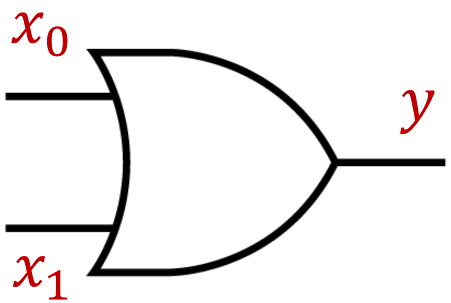
AND Gate

x_0	x_1	y
0	0	0
0	1	0
1	0	0
1	1	1

McCulloch-Pitts Neuron

Boolean function ‘OR’ can be implemented by using MP Neuron



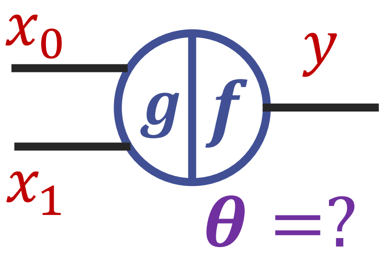


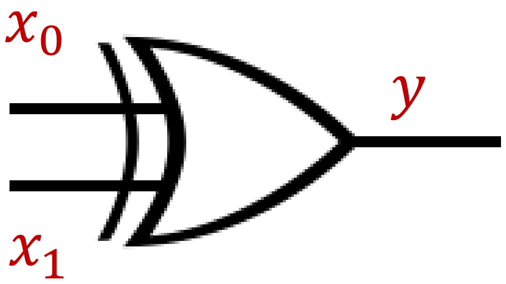
OR Gate

x_0	x_1	y
0	0	0
0	1	1
1	0	1
1	1	1

McCulloch-Pitts Neuron

Boolean function XOR cannot be implemented by using MP Neuron





XOR Gate

x_0	x_1	y
0	0	0
0	1	1
1	0	1
1	1	0

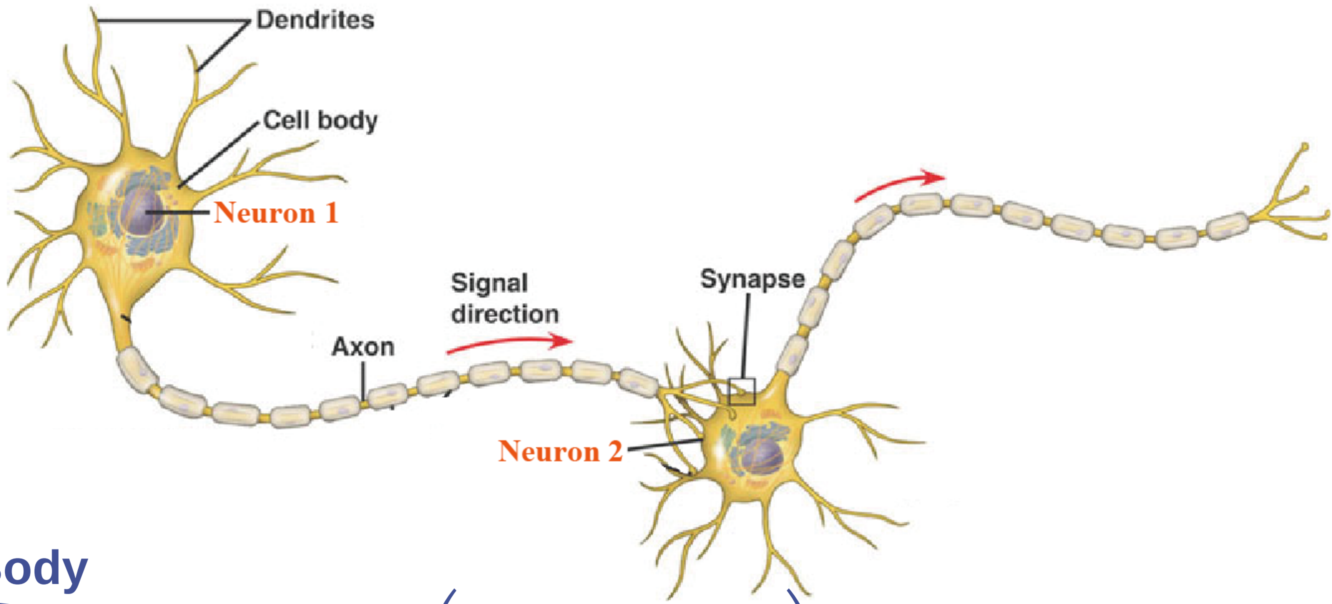
MP Neuron is limited to only solve linearly separable functions!

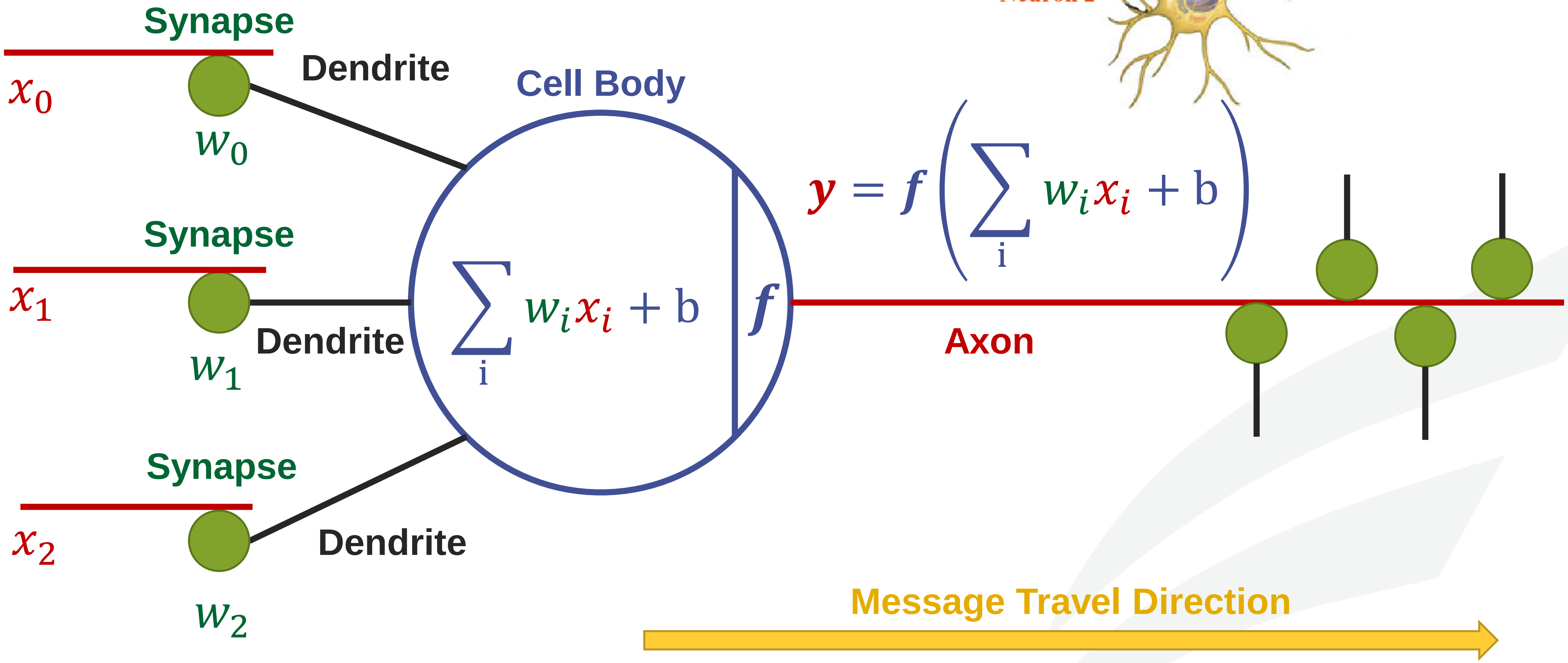
Artificial Neuron Design

- **Idealized neuron models**
  - Idealization removes complicated details that are not essential for understanding the main principles.
  - It allows us to apply mathematics and to make analogies.
- **Break the limitations on MP Neuron**
  - What about non-boolean inputs (say, real number)?
  - What if we want to assign more weight (importance) to some inputs?
  - Do we always need to hand code the threshold?
  - What about activation functions other than threshold step function?
  - What about functions which are not linearly separable ?

Perceptron

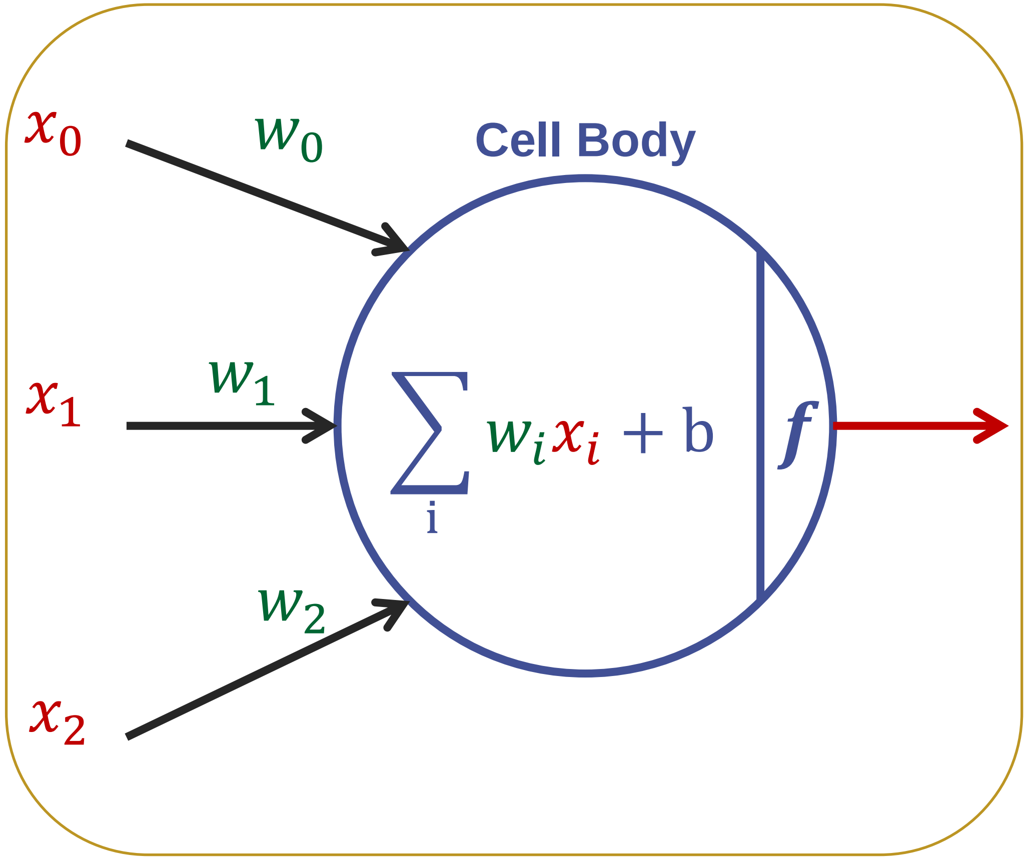
Frank Rosenblatt @ 1958

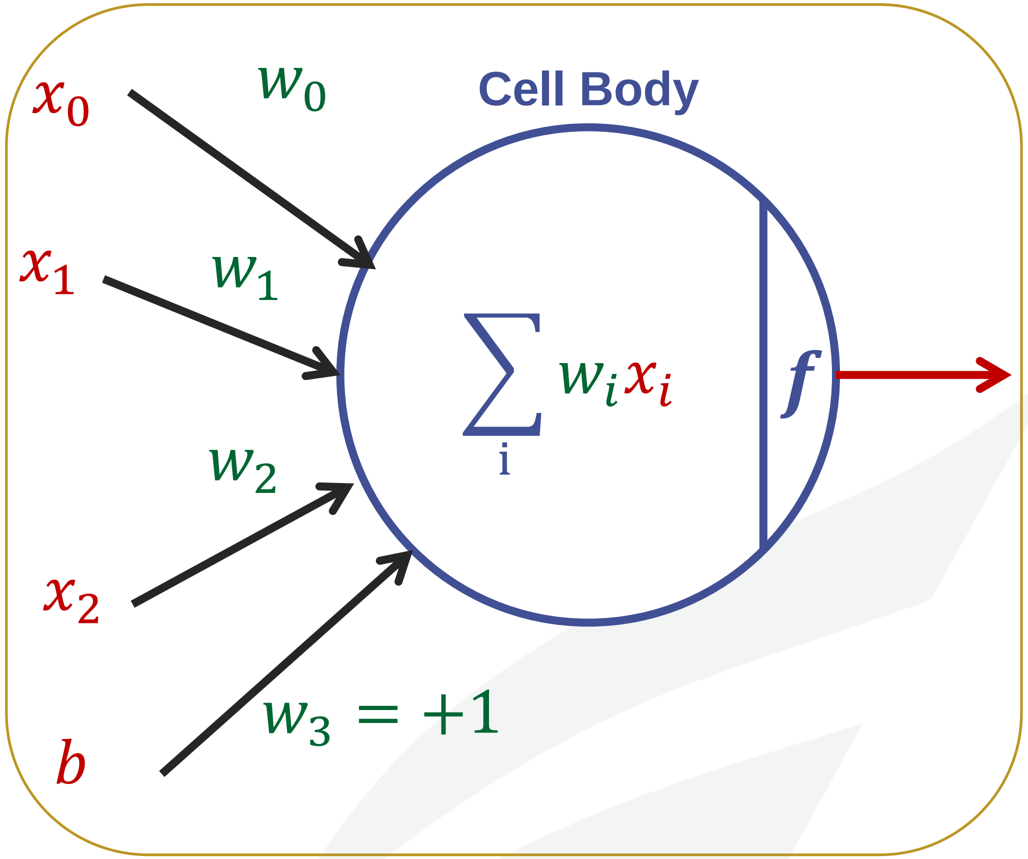




Perceptron

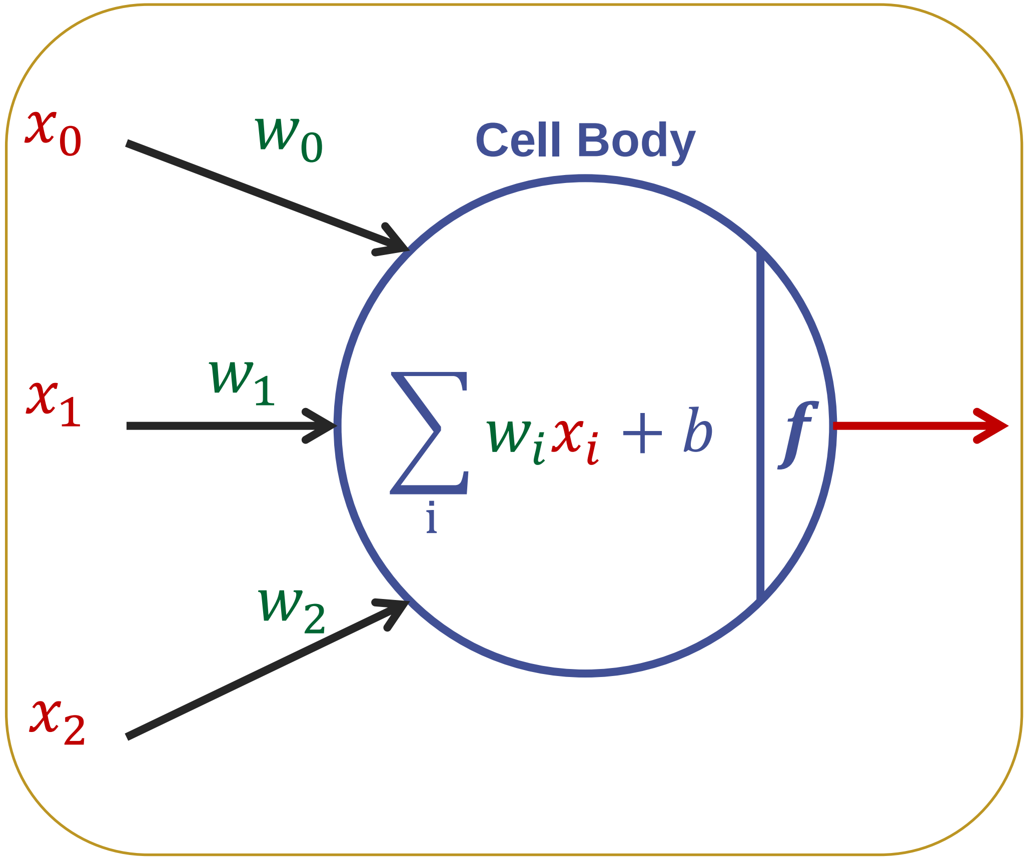
What is Bias b?

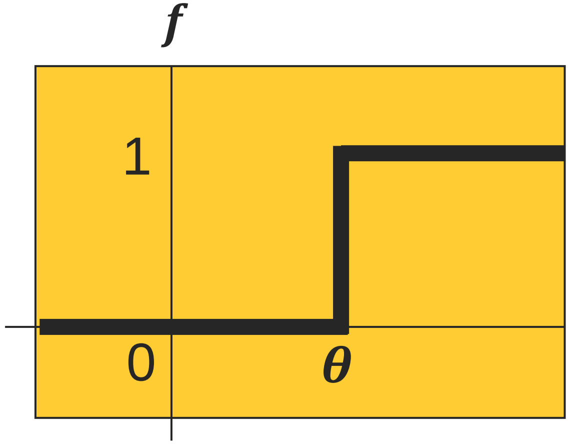




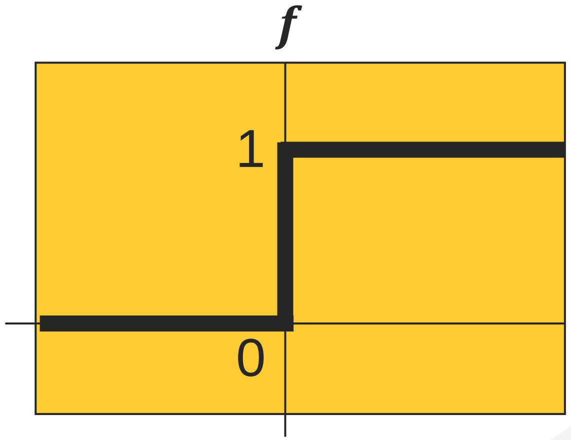
Perceptron

Effect of bias b on Threshold Step activation function.





$$b = 0$$



$$b = -\theta$$

$$z = \sum_i x_i w_i$$

$$z = \sum_i x_i w_i - \theta$$

$$y = \begin{cases} 1 & if\ z > \theta \\ 0 & otherwise \end{cases}$$

$$y = \begin{cases} 1 & if\ z > \mathbf{0} \\ 0 & otherwise \end{cases}$$

Perceptron v.s. MP Neuron

Perceptron

$$y = \begin{cases} 1 & if \sum_{i} x_i w_i + b > \mathbf{0} \\ 0 & otherwise \end{cases}$$

MP Neuron

$$y = \begin{cases} 1 & if \sum_{i} x_i > \theta \\ 0 & otherwise \end{cases}$$

In Perceptron: the inputs can be **real numbers;** the weights (including threshold) can be **learned/trained.**

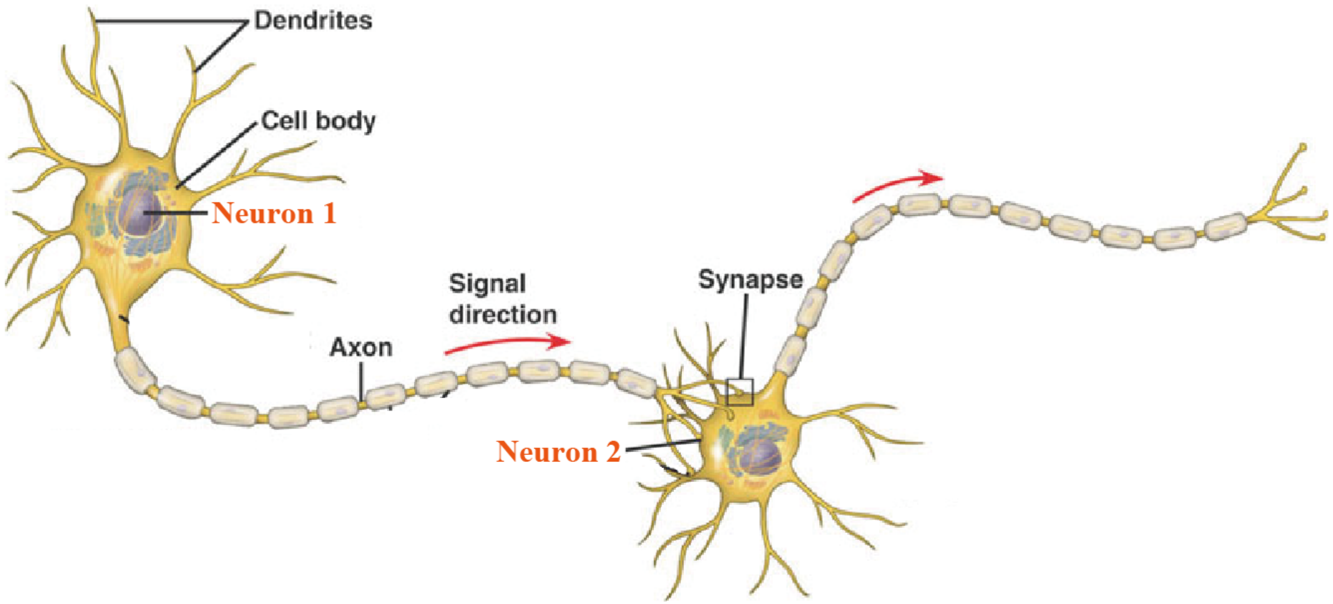
In Perceptron: like MP Neuron, the Perceptron separates the input space into two halves. However, all inputs producing 1 lie on one side, and those producing 0 lie on the other side.
===> A single perceptron can still **only used to implement linearly separable functions**, but not for XOR-like function.

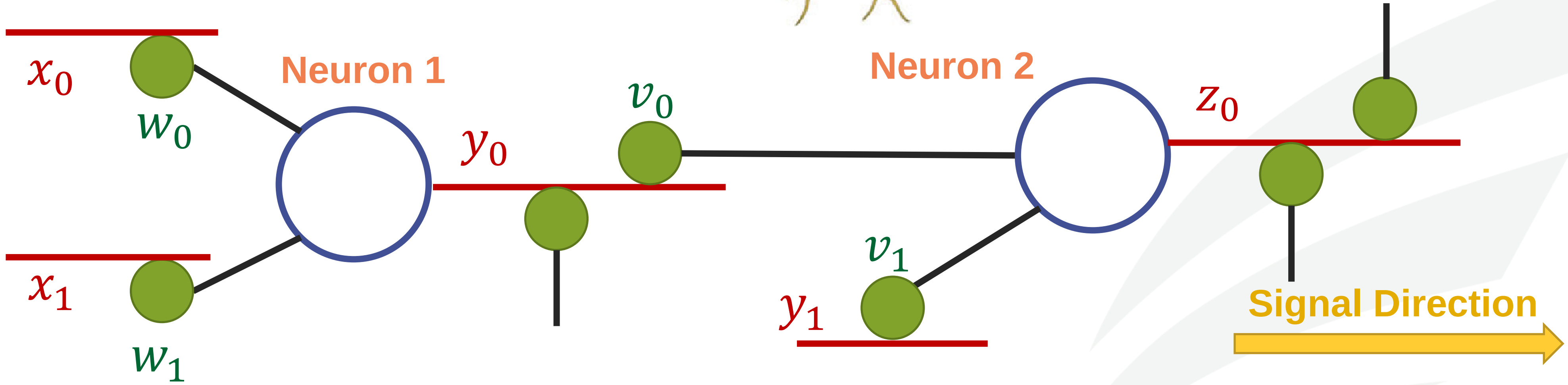
Artificial Neuron Design

- **Idealized neuron models**
  - Idealization removes complicated details that are not essential for understanding the main principles.
  - It allows us to apply mathematics and to make analogies.
- **Break the limitations on MP Neuron**
  - What about non-boolean inputs (say, real number)? ✔
  - What if we want to assign more weight (importance) to some inputs? ✔
  - What about functions which are not linearly separable ? ❓ **=> MLP**
  - Do we always need to hand code the threshold? ❓ **=> Training**

Multi-Layer Perceptron (MLP)

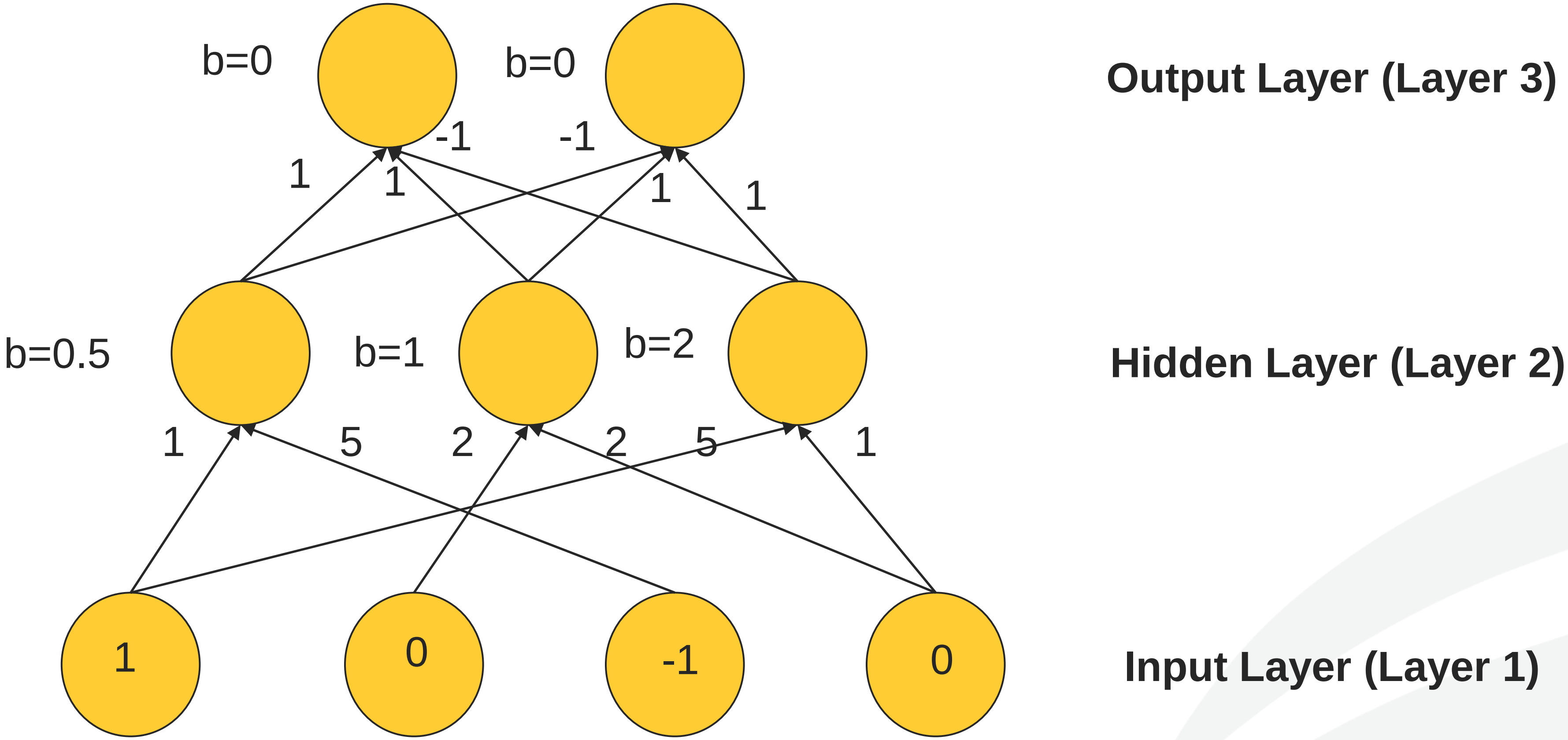
Connect two neurons





Multi-Layer Perceptron (MLP)

Connect more neurons and more layers



Lab 1: Implementing XOR using MLP on Google Colab

Assignments and Related Documents:

- https://jqub.github.io/2021/09/01/ML4Emb/

Due Date: This Friday by 1 PM

- Please take this chance to evaluate the required programming background and the required bandwidth to decide whether keep or drop this course.

GMU.EDU

George Mason University

4400 University Drive
Fairfax, Virginia 22030

Tel: (703)993-1000