09.11.20 10:33

CSCI 101
Connecting with Computer Science
Artificial Intelligence II

Jetic Gū

2020 Fall Semester (S3)

Overview

- Focus: Artificial Intelligence
- Readings: -
- Core Ideas:
    1. How the Mind Works
    2. Neural Network Introduction
    3. Limitations and Applications

P1
Mind

How the Mind Works

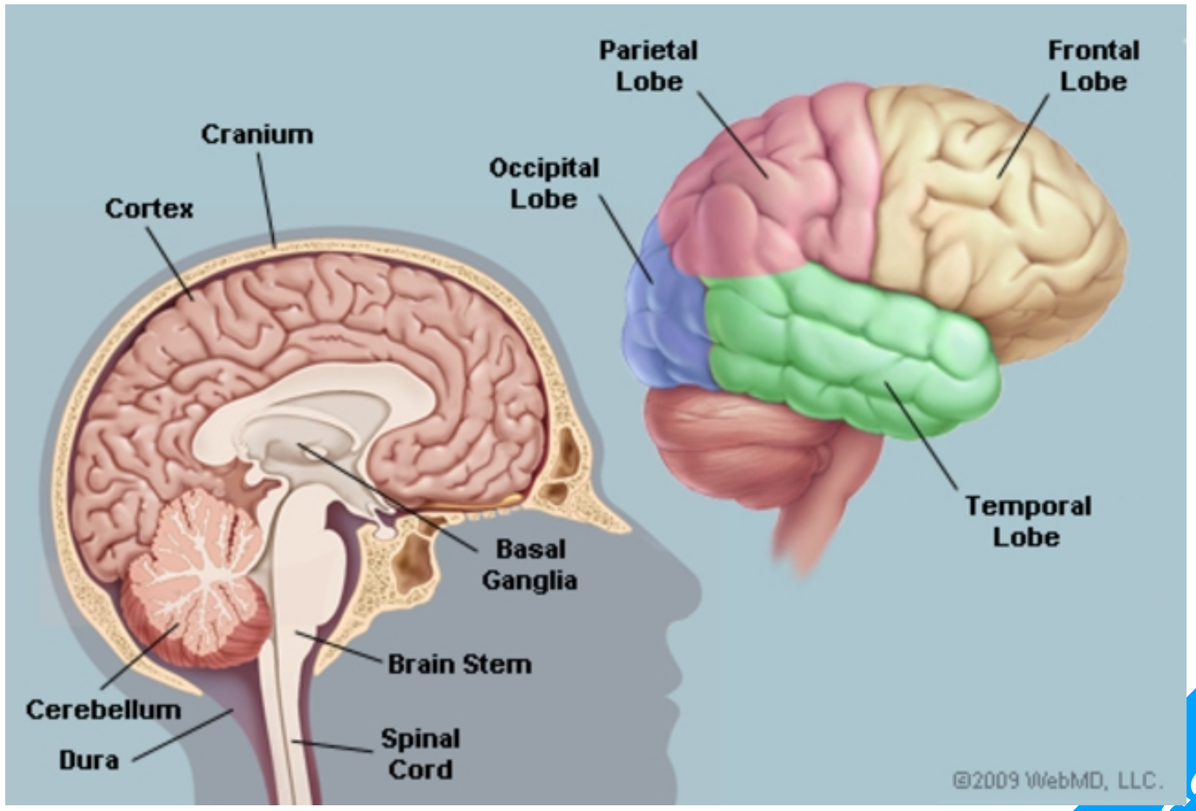
Brain! Brain! I love BRAINSSS!

Summary

P1
Mind

General Anatomy

- The brain is part of the central nervous system
  - Brain
    - Main Cortex
    - Cerebellum
    - Brain Stem
    - etc...
  - Spinal Cord



Technical

P1
Mind

General Anatomy

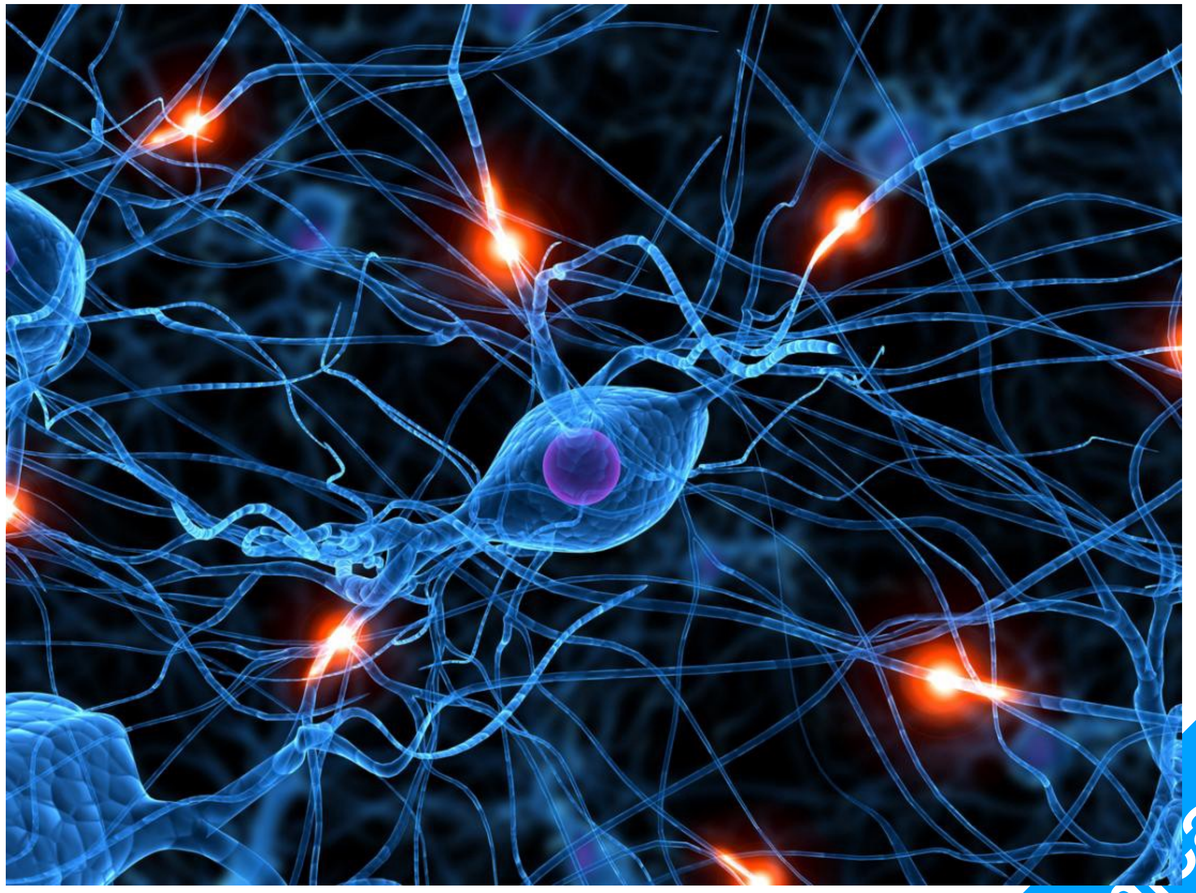
- Each region of the brain consists of massive neurones interconnected with each other (thousands of different types)

- Neurone

  - **Soma** (Cell body) + **Nucleus** (Core)

  - **Dendrites**

  - **Axons**
    connected to the dendrites of other neurones



Technical

P1
Mind

General Anatomy

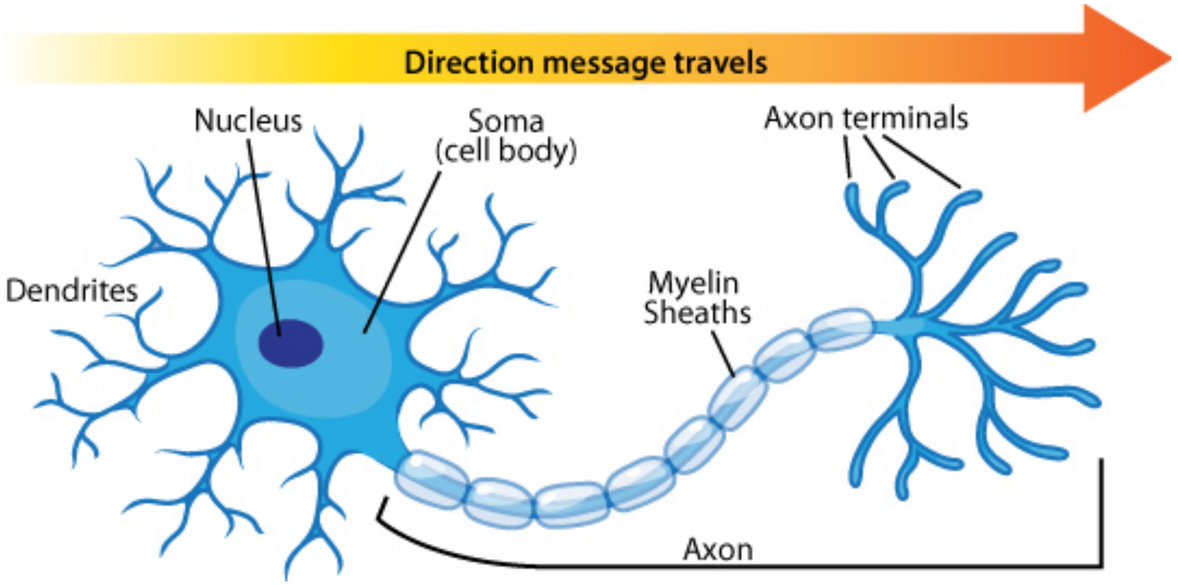
- Each region of the brain consists of massive neurones interconnected with each other (thousands of different types)

- Neurone

  - **Soma** (Cell body) + **Nucleus** (Core)

  - **Dendrites**

  - **Axons**
    connected to the dendrites of other neurones

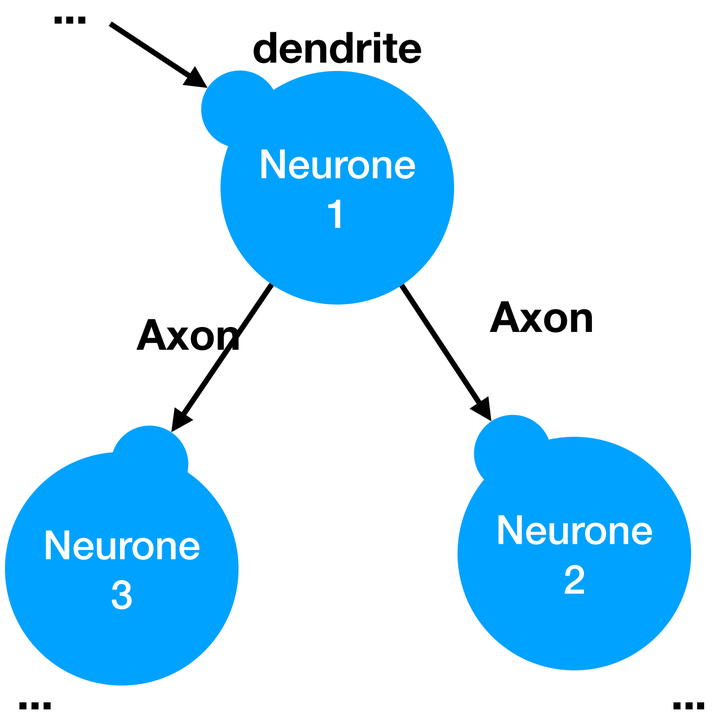


Technical

P1
Mind

General Anatomy

- Each region of the brain consists of massive neurones interconnected with each other (thousands of different types)
- Neurone
  - **Soma** (Cell body) + **Nucleus** (Core)
  - **Dendrites**
  - **Axons**
    connected to the dendrites of other neurones

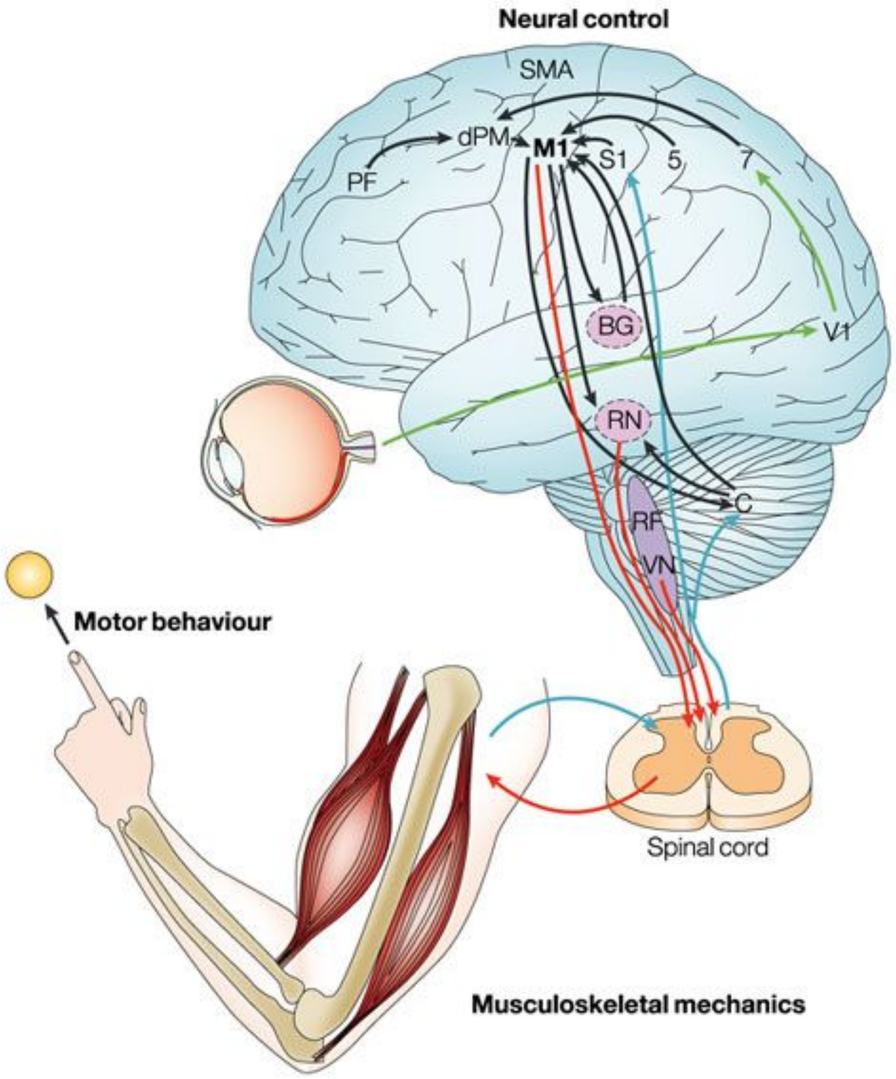


Technical

P1
Mind

General Anatomy

- Information Passage
  - Sensory information gets passed to the brain through certain **neurotransmitter ions**
  - Upon receiving certain **ions**
    - Neurones generate **Electric Potential**
    - Chooses to pass/not pass **ions** to other **connected neurones**
    - Certain ions travel to our **muscle** then cause contractions, which brings about our motor functions



Technical

P2
NN

Intro. to Artificial Neural Network

Multi-layer Perceptrons

Summary

A Mathematical Model

- We talked about basic **Perceptron** models
  - Features/Evidence X and Weights W
  - Predictions: $P(L = l_j \,|\, X) = \sigma(\Sigma_i x_i w_{ij})$, where σ is the logistic function
- A single **perceptron** unit **is like** a single **neurone**
  - Information from dendrites: $x_i \in X$
  - Pass on to its axons: $P(L = l_j \,|\, X) = \sigma(\Sigma_i x_i w_{ij})$
    to be taken as inputs to other neurones!

Review

P2
NN

A Mathematical Model

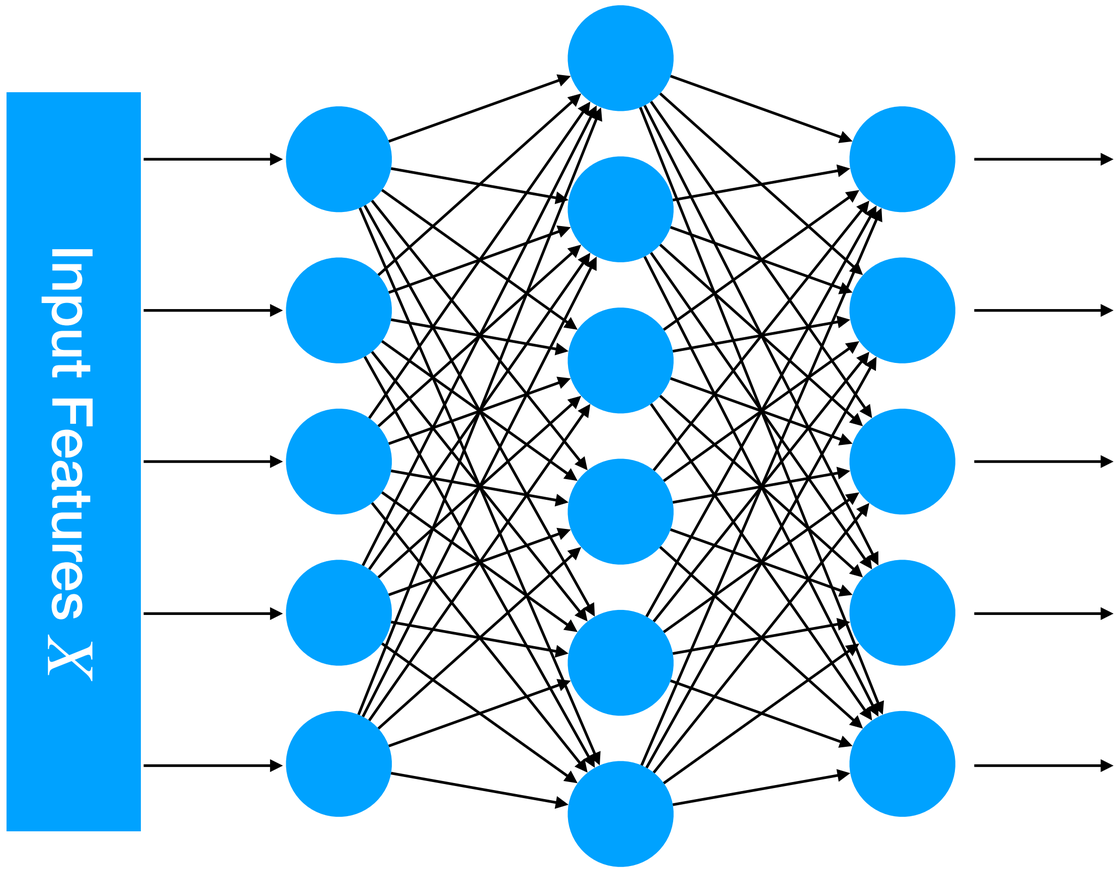
1. Rosenblatt, 1961. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms

Technical

P2
NN

A Mathematical Model

Example for a 3-layer Multi-layer Perceptron Model



$$y_0 = P(L = l_0 \mid X)$$

$$y_1 = P(L = l_1 \mid X)$$

$$y_2 = P(L = l_2 \mid X)$$

$$y_3 = P(L = l_3 \mid X)$$

$$y_4 = P(L = l_4 \mid X)$$

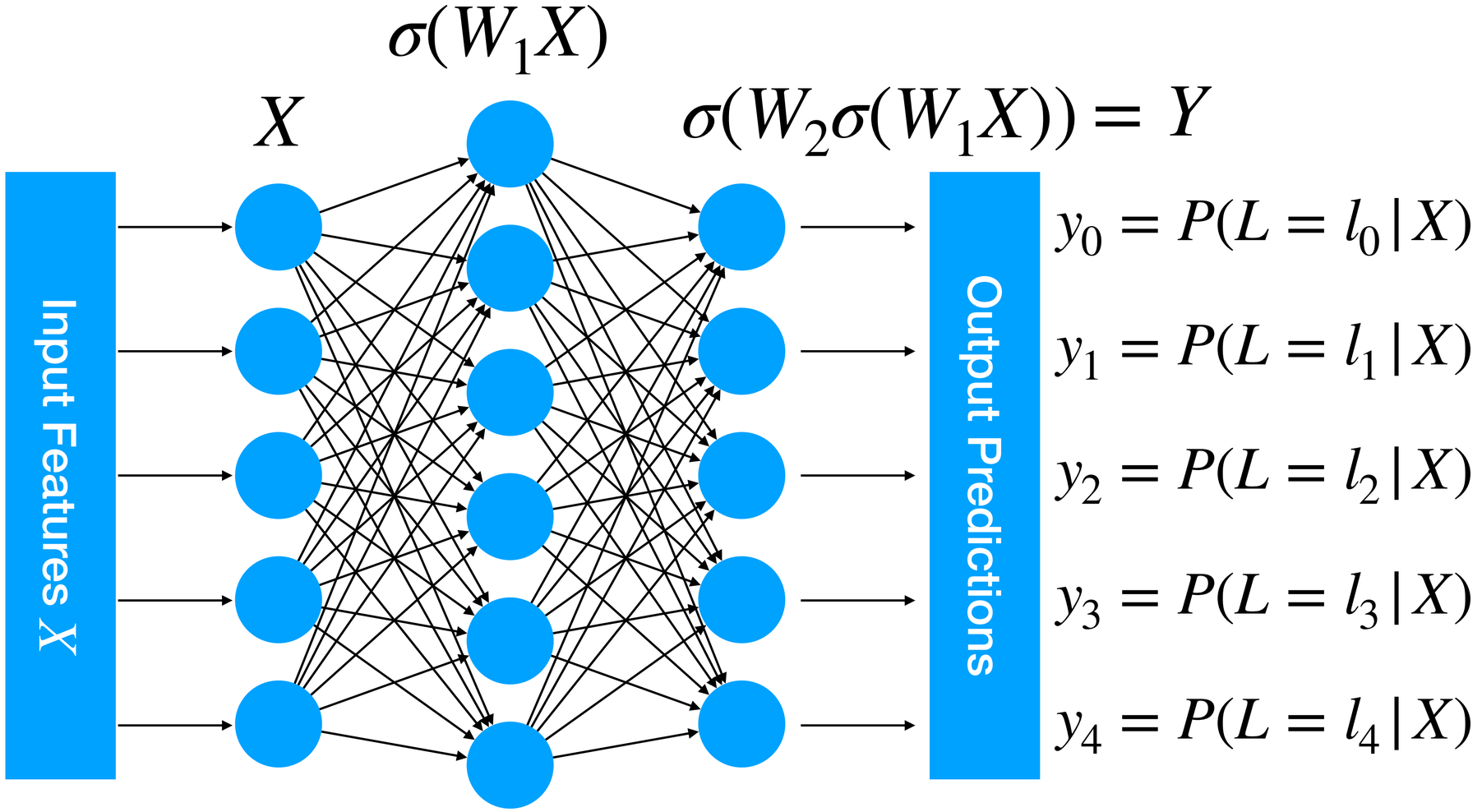
Technical

1. Rosenblatt, 1961. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms

P2
NN

A Mathematical Model

Example for a 3-layer Multi-layer Perceptron Model

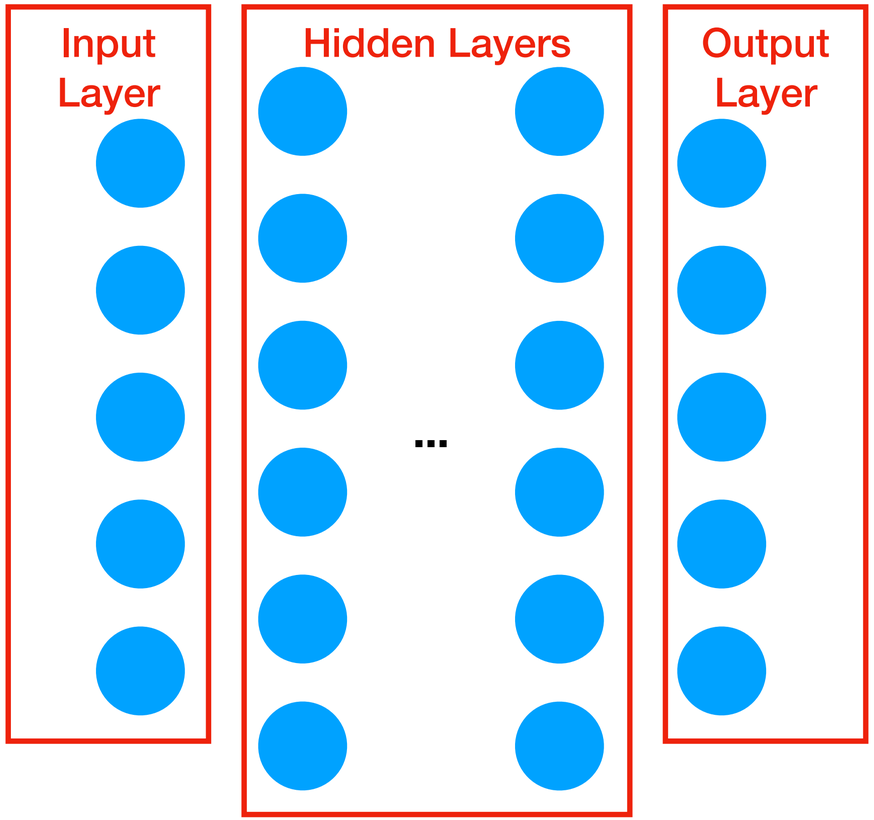


1. Rosenblatt, 1961. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms

Technical

A Mathematical Model

- Multi-layer Perceptron
  - Can have **more layers** between input layer and output layer
  - Can have **more nodes** in each layer
  - Can use **different activation functions** than Logistic σ
  - Can have **incomplete connections**
  - Can have backward connections (**recurrent neural network**)
  - Can have forward connections (**highway connections**)
  - .....



Concept

P2
NN

How can you learn the weights?

- For MLP (Multi-layer perceptron)[1]
  - Back-propagation Algorithm
  - For recurrent neural network: back-propagation-through-time (BPTT)

Concept

1. Rosenblatt, 1961. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms

P2
NN

What have we accomplished with MLP?

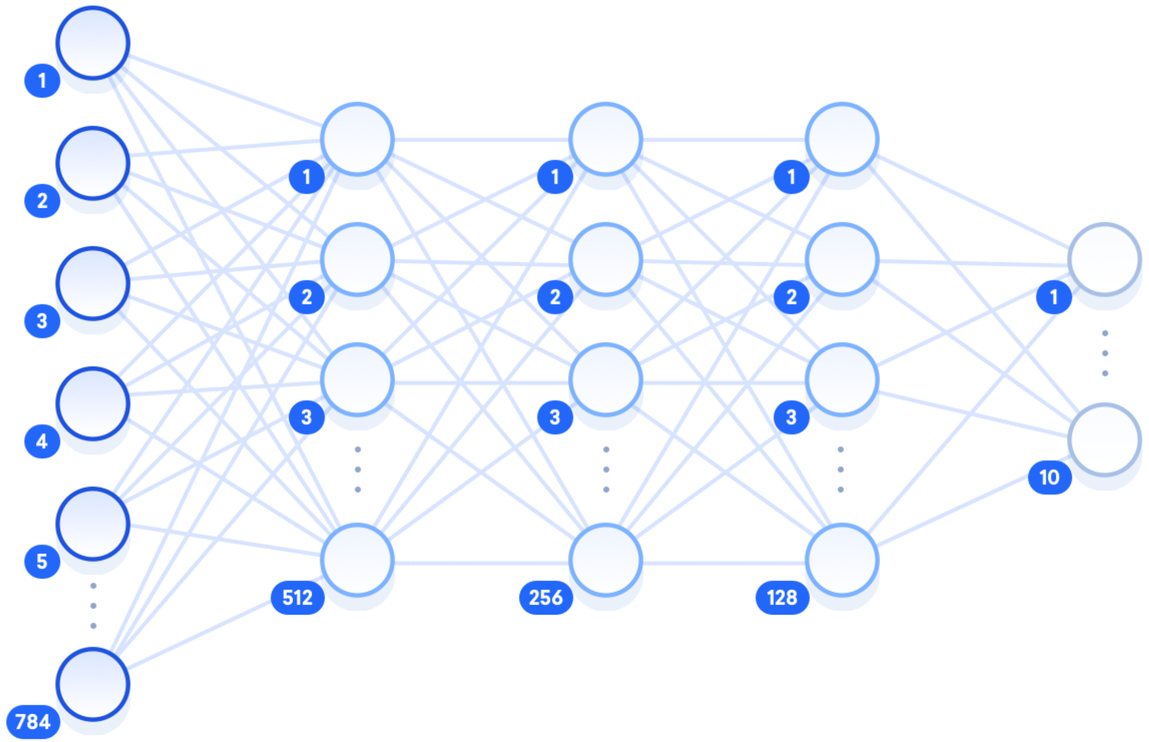
- Recognising handwritten digits
  - Input: 28 x 28 grey-scaled values, one for each pixel
    This is our $X \in \mathbb{R}^{28\times28} = \mathbb{R}^{784}$
  - Output: 1 x 10 decimal values
    This is our $(y_0, y_1, \ldots, y_9) Y \in [0,1]^{10}$, $Y \in [0,1]^{10}$,
    $y_i = P(\text{Label} = i \,|\, X)$

Application

What have we accomplished with MLP?

- Recognising handwritten digits

X



Y

- **This model can give you >90% accuracy!**

Application

1. https://www.digitalocean.com/community/tutorials/how-to-build-a-neural-network-to-recognize-handwritten-digits-with-tensorflow

P2
NN

What have we accomplished with MLP?

- Primitive autonomous cars
  - Recognise lanes
  - Control steering wheel
  - 1980s: NAVLAB

Application

Progress

- We’ve had a lot of advancement in NN research, with **better models than MLP**
- We’ve discovered a lot of useful applications of NN, such as **image processing** and **NLP**
- We’ve learn a lot about how to create NNs with more and more layers (**Deep Learning**)
- Problems
  - Learning algorithm: still **basic** back-propagation
  - Artificial neurones: **barely any progress** in the past 20 years
  - More layers of NN **does not improve performance** that much anymore
  - We are currently at the bottleneck!