Perceptron

Nan Ye

School of Mathematics and Physics
The University of Queensland

Schedule

A tentative schedule is available on BlackBoard

- Week 1-2: machine learning basics
- Week 3-4: neural network basics
- Week 5-6: deep architectures
- Week 7-8: optimization
- Week 8-10: improving generalization
- Week 10-11: generative models
- Week 11-12: reinforcement learning, applications, review

Neural Computation

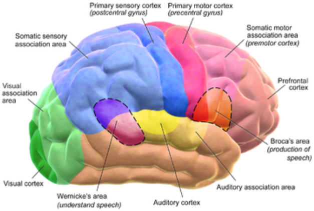
- The human brain performs complex information processing tasks.
  - Perception: five senses, object recognition, speech processing
  - Control: coordinate body parts and move
  - Reasoning: logical deductions
- This stimulates interest to better understand and make use of this.
  - How the brain completes all these tasks via neural computations?

    Connectionism studies this question using computer simulations
  - Can we develop parallel brain-inspired computers?

    Neuromorphic chips: IBM's TrueNorth chip, Intel Loihi chip, SpiNNaker
  - Can we develop brain-inspired learning systems?

    Many ideas in artificial neural network are inspired by the brain

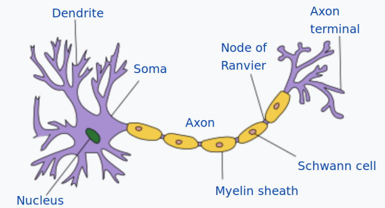
The Human Brain



https://en.wikipedia.org/wiki/Human_brain

- Brain activity is made possible by the neurons, the interconnections between them, and their release of neurotransmitters in response to nerve impulse.
- There are more than 86 billion neurons in the human brain.
- Neurons connect to form neural pathways, neural circuits, and complex functional areas.

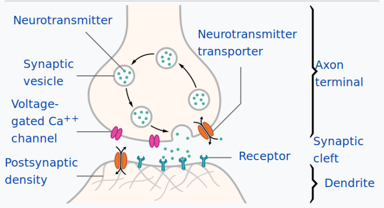
A typical neuron



https://en.wikipedia.org/wiki/Neuron

- A typical neuron consists of a cell body (soma), dendrites, and a single axon.
  - Dendrites: receive information from other neurons.
  - Soma: join signals from the dendrites and pass them on.
  - Axon: transmits information to other neurons.

Synapse



https://en.wikipedia.org/wiki/Synapse

- Synapses are functional neural connections, typically from the axon terminals of neurons to the dendrites of other neurons.
- A typical neuron makes a few thousand connections.
- Synapses are slow, but very small, very low-power, and adaptive.

Activation of the synapse

- When the neuron voltage changes rapidly, the neuron generates an *all-or-none* electrochemical pulse that activates the synapse
    - Synaptic vesicles release transmitter chemicals
    - Transmitter chemicals diffuse across the synaptic cleft
    - Transmitter chemicals bind to receptors of the post-synaptic neuron

Synapses are adaptive

- Synaptic plasticity: synapses change depending on how active or inactive they were.

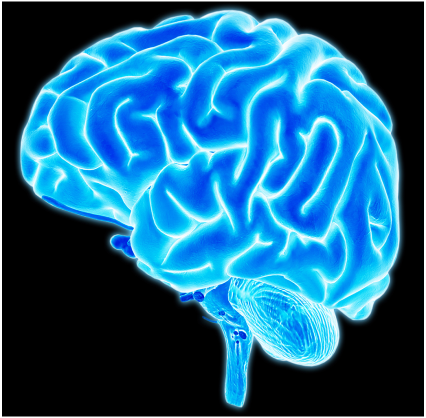
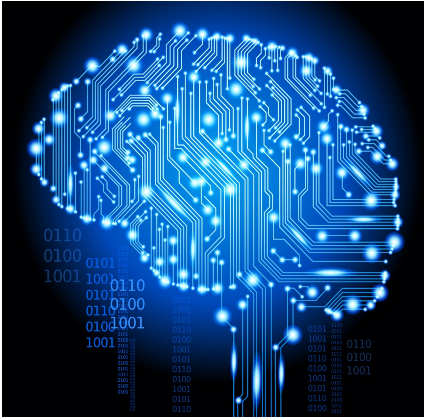
  Synaptic strength can be strengthened or weakened due to changes in the number of synaptic vesicles, number of receptor molecules.

- Synaptic plasticity is one of the important neurochemical foundations of learning and memory.

Two theories

- There are two main theories on how the brain works: the theory of modularity, and distributive processing.
- The theory of modularity assumes that the brain is divided into several functional areas.
  - e.g. visual area V4 and V5 are specifically involved in the perception of color and vision motion respectively.
  - Local damages have specific effects, but functions sometimes relocate upon damage.
- Distributive processing assumes that the brain is interactive, and its regions are functionally interconnected rather than specialised.

Brain vs. Computer

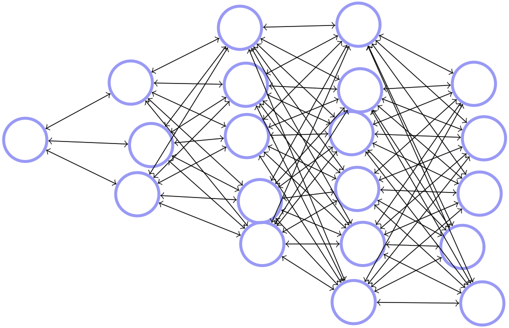
Strengths and weaknesses

- Computers are good at numerical and symbolic problems, but very vulnerable.
- Human brains are good at perceptual problems, and robust.

Architectural differences

	von Neumann computer	**human brain**
processor	complex	simple
	high speed	low speed
	one or a few	many
memory	separate from processor	integrated into processor
	localized	distributed
	non-content addressable	content addressable
computation	centralized	distributed
	sequential	parallel
	stored programs	self-learning

An abstract view of the brain



- An information processing system consisting of simple connected neurons.
- An information processing task is achieved by neurons receiving inputs from some neurons and sending their outputs to some neurons.

Recall

1943	computational model for neural networks
	McCulloch and Pitts, A logical calculus of the ideas immanent in nervous activity
1949	Hebbian learning (cells that fire together wire together)
	Hebb, The organization of behavior: A neuropsychological theory
1960	single layer and multilayer neural nets (ADALINE and MADALINE)
	Widrow and Hoff, *Adaptive switching circuits*
1962	Perceptron
	Rosenblatt, *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms*
1969	limitations of artificial neural nets
	Minsky and Papert, *Perceptrons; an introduction to computational geometry*

McCulloch-Pitts' neuron

- McCulloch and Pitts (1943) proposed linear threshold neurons as a mathematical model for biological neurons.
- Each neuron computes a weighted sum of the inputs, and sends out a spike of activity if the weighted sum exceeds a threshold.
- They thought of each spike as the truth value of a proposition, and each neuron combines truth values to compute the truth value of another proposition.

The perceptron

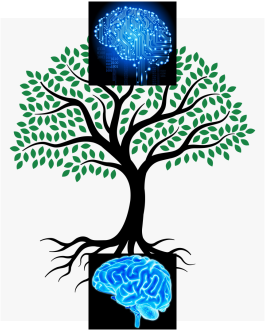
- The perceptron is a supervised classification algorithm proposed by Frank Rosenblatt.
- The model consists of a single McCulloch-Pitt's neuron.
- The learning algorithm is still used for tasks with enormous feature vectors that contain many millions of features.

A setback

- Minsky and Papert's *Perceptrons* book showed what perceptrons could do and their limitations.
- However, many people misinterpretted the limitations as limitations to all neural network models.

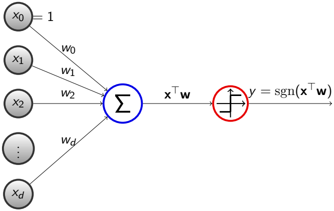
Why So Much about the Biology

Return to the root and you will find the meaning. Sengcan



...and the root provides the nourishment for growth.

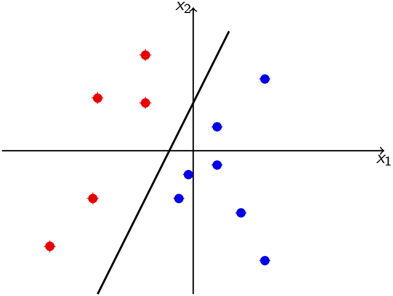
Linear Threshold Unit



- McCulloch-Pitts' linear threshold neuron computes $f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^\top \mathbf{x})$, where $\mathbf{x} \in \mathbf{R}^{d+1}$ is the input vector, and $\mathbf{w} \in \mathbf{R}^{d+1}$ consists of the weights for the inputs.
- $\mathrm{sgn}(x)$ takes value -1 and 1 when x is non-positive and positive, respectively.
- We include a dummy variable 1 to account for the bias.

Decision boundary

A linear threshold unit defines a linear decision boundary.



Weight space view

- Each example $(\mathbf{x}, y)$ defines a half space of correct weight $\mathbf{w}$ satisfying $y\mathbf{x}^\top \mathbf{w} > 0$.
- The intersection of all these half spaces defines a cone containing weights that correctly classify all examples.
  - The solution space is convex, i.e. if $\mathbf{w}_1$ and $\mathbf{w}_2$ are solutions, then for any $\lambda \in [0, 1]$, $\lambda \mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2$ is a solution.

The Perceptron

Require: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbf{R}^{d+1} \times \{-1, +1\}$, $\eta \in (0, 1]$.
Ensure: Weight vector $\mathbf{w}$.

Randomly or smartly initialize $\mathbf{w}$.
while there is any misclassified example **do**
  Pick a misclassified example $(\mathbf{x}_i, y_i)$.
  $\mathbf{w} \leftarrow \mathbf{w} + \eta y_i \mathbf{x}_i$.

N.B. Each input vector $\mathbf{x}_i$ has a dummy feature with value 1 so that a bias term can be learned.

Why the update rule $\mathbf{w} \leftarrow \mathbf{w} + \eta y_i \mathbf{x}_i$?

- This is an error correction learning algorithm that only updates a model when it makes a mistake.
- $\mathbf{w}$ classifies $(\mathbf{x}_i, y_i)$ correctly if $y_i \mathbf{x}_i^\top \mathbf{w} > 0$.
- If $\mathbf{w}$ classifies $(\mathbf{x}_i, y_i)$ wrongly, then the update rule moves $y_i \mathbf{x}_i^\top \mathbf{w}$ towards positive, because

$$y_i \mathbf{x}_i^\top (\mathbf{w} + \eta y_i \mathbf{x}_i) = y_i \mathbf{x}_i^\top \mathbf{w} + \eta ||\mathbf{x}_i||^2 > y_i \mathbf{x}_i^\top \mathbf{w}.$$

Perceptron Convergence Theorem

Assume that the data is linearly separable, that is, for some $\mathbf{w}^*$ we have $y_i \mathbf{x}_i^\top \mathbf{w}^* > 0$ for all i.

$$R = \max_i ||\mathbf{x}_i||,$$
$$\gamma = \min_i y_i {\mathbf{w}^*}^\top \mathbf{x}_i / ||\mathbf{w}^*||.$$

Suppose the initial weights are 0. Then the perceptron algorithm finds a separating hyperplane, and the number of updates required is at most

$$R^2/\gamma^2.$$

Remark. R is the radius of an enclosing ball centered at 0, and γ is the minimum distance from a data point to the hyperplane $\mathbf{w}^*\mathbf{x} = 0$.

Proof. Let $\mathbf{w}_t$ be the weight vector after t updates, with the initial weight vector being $\mathbf{w}_0 = 0$. The idea is to show that the angle θ_t between $\mathbf{w}_t$ and $\mathbf{w}^*$ decreases in general, or that $\cos\theta_t = \frac{\mathbf{w}_t^\top \mathbf{w}^*}{\|\mathbf{w}_t^\top\|\|\mathbf{w}^*\|}$ increases in general.
First, we have that $\mathbf{w}_t^\top \mathbf{w}^* \geq t\eta\gamma||\mathbf{w}^*||$ because

$$\mathbf{w}_{t+1}^\top \mathbf{w}^* = \mathbf{w}_t^\top \mathbf{w}^* + \eta y_i \mathbf{x}_i^\top \mathbf{w}^* \geq \mathbf{w}_t^\top \mathbf{w}^* + \eta\gamma||\mathbf{w}^*||,$$

where we use $\gamma \leq y_i \mathbf{w}^{*\top} \mathbf{x}_i/||\mathbf{w}^*||$ in the inequality.
In addition, we have $||\mathbf{w}_t||^2 \leq t\eta^2 R^2$ because

$$||\mathbf{w}_{t+1}||^2 = ||\mathbf{w}_t + \eta y_i \mathbf{x}_i||^2 = ||\mathbf{w}_t||^2 + 2\eta y_i \mathbf{w}_t \mathbf{x}_i + \eta^2||\mathbf{x}_i||^2 \leq ||\mathbf{w}_t||^2 + \eta^2 R^2,$$

where we use the fact that $(\mathbf{x}_i, y_i)$ is misclassified and thus $y_i \mathbf{w}_t \mathbf{x}_i \leq 0$.
Using the above two inequalities, we have

$$1 \geq \cos\theta_t \geq \frac{t\eta\gamma||\mathbf{w}^*||}{\sqrt{t}\eta R||\mathbf{w}^*||} = \frac{\sqrt{t}\gamma}{R}.$$

Hence $t \leq R^2/\gamma^2$.

Convergence for arbitrary initial weights

- In the above analysis, we assume that the perceptron algorithm starts from 0 in the perceptron convergence theorem.
- In fact, as long as the data is linearly separable, the perceptron algorithm converges irrespective of the initial weights.
- We may converge faster/slower if we start with a good/bad guess of the weights.
- The learning rate η has an effect on how fast the algorithm converges, but it is difficult to choose a good η.

Remark on bias

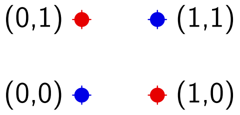
- We often need to include a bias in a linear classifier.
- This has already been taken into account using the dummy variable trick: add an extra input with constant value 1.
- This allows the perceptron algorithm to learn the weights and the bias using the same update rule.
- Note that while we often use the dummy variable trick to simplify notations, sometimes we need to treat the bias with care (e.g. in ridge regression).

Weakness of the algorithm

- When the data is separable, the hyperplane found by the perceptron algorithm depends on the initial weight and is thus arbitrary.
- Convergence can be very slow, especially when the gap between the positive and negative examples is small.
- When the data is not separable, the algorithm does not stop, but this can be difficult to detect.

The Limitations of Perceptrons

- A perceptron cannot learn even simple functions like XOR (exclusive or).
- Consider the case with 2 binary variables only, then our data points look like the following



  where red is 1, and blue is -1.
- If a perceptron can be used to perfectly classify the data, then the decision boundary is a straight line.
- But there is no way to draw a straight line that separates the data!

Your Turn

Which of the following statement is correct? (Multiple choice)

(a) A perceptron has a quadratic decision boundary in the feature space.

(b) The perceptron learning algorithm always converges on non-linearly separable dataset.

(c) The perceptron algorithm can only learn a linear function without a bias term.

(d) The XOR function cannot be represented by a perceptron.

What You Need to Know

- Biological inspiration of artificial neural networks
- Linear threshold units
- The perceptron algorithm
- Perceptron convergence theorem
- Limitations of perceptrons