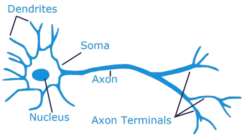
Neural Networks

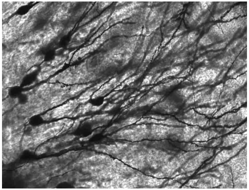
Learning of hierarchical concepts - Joint work with Nancy Lynch (MIT)

Frederik Mallmann-Trenn
King's College London

Biological Inspiration

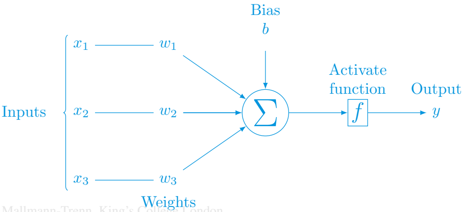
- A neuron in the brain (there are 86 billion neurons):





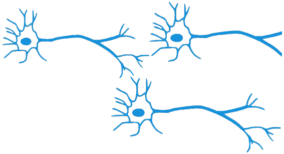
  - Dendrites have receptors that pick up on neurotransmitter (chemical signals) and interpret them as electrical signals
  - Those signals are interpreted in the SOMA (cell body)
  - Nucleus contains genetic material of the cell
  - If the signal is strong enough, it will be send to the axon
  - Axon terminals release neurotransmitter

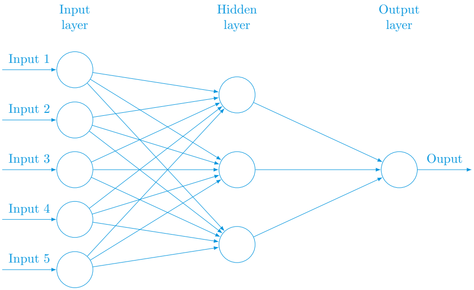
- A neuron in an artificial neutral network (ANN):



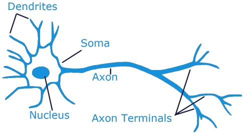
Biological Inspiration

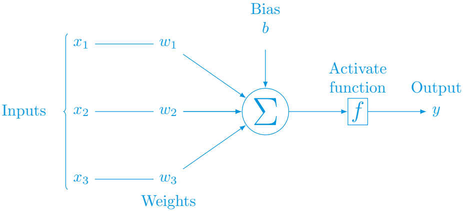
- Of course in both cases they can be connected to other neurons





Difference between the brain and artificial neural networks

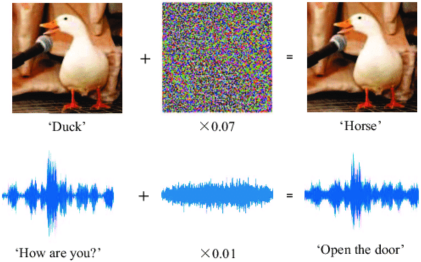




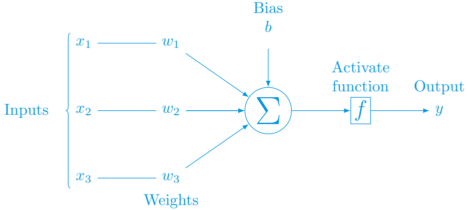
- There are many differences: synchronous rounds (ANN), fault tolerance, learning, signals/activation functions

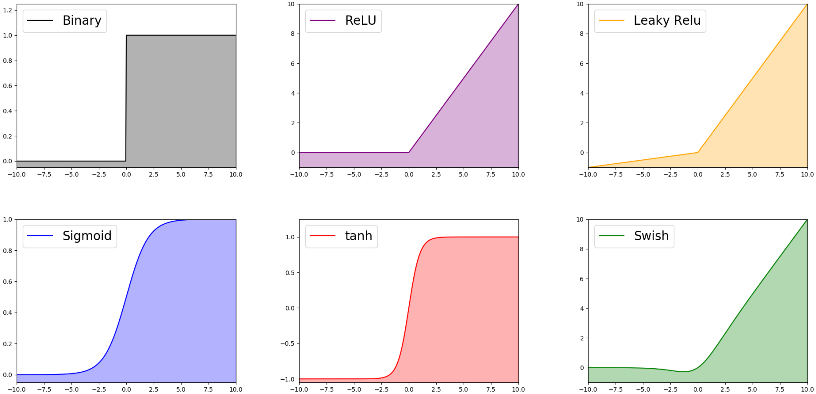
Error tolerance

A little bit of noise can completely throw off ANN, whereas brains are more resilient

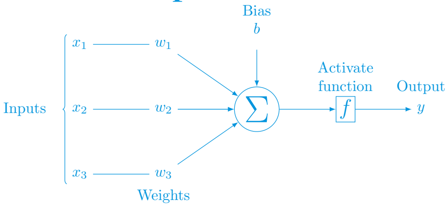


Activation Functions



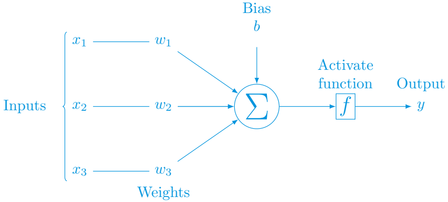


Our assumptions



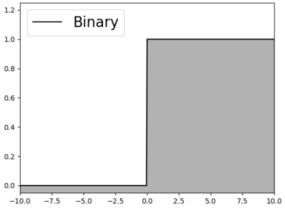
- Input vector $\mathbf{x}$
- Weights $\mathbf{w}$
- Bias b
- Activation function f
- Example: let's say f is binary

Our assumptions



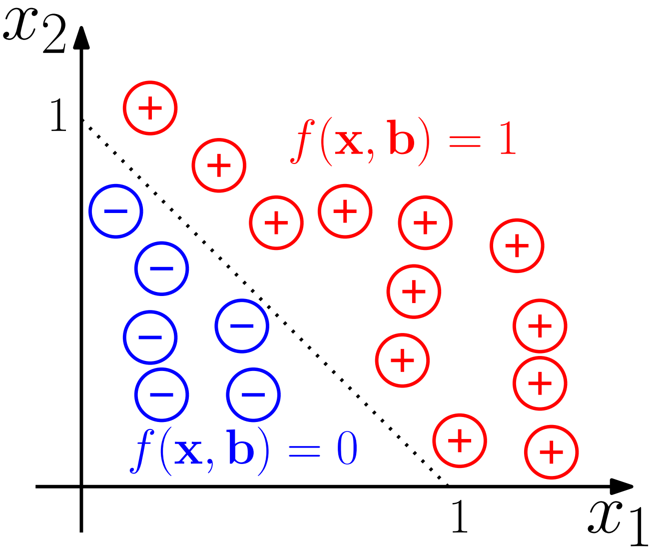
- Input vector $\mathbf{x}$
- Weights $\mathbf{w}$
- Bias b
- Activation function f

- Example: let's say f is binary
- $y = f_{binary}(\mathbf{x}, \mathbf{w}) = \begin{cases} 1 & \text{if } \mathbf{x}^T \cdot \mathbf{w} \geqslant b \\ 0 & \text{otherwise} \end{cases}$

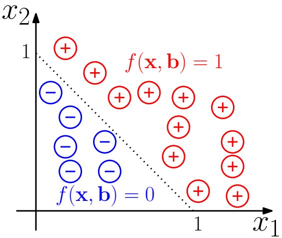


Note, that $\mathbf{x}^T \cdot \mathbf{w} = \sum_i x_i \cdot w_i$

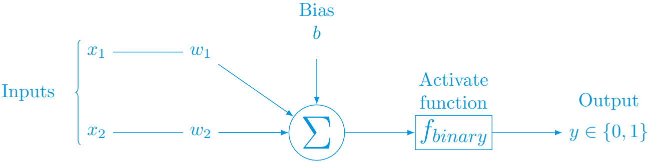
Classification



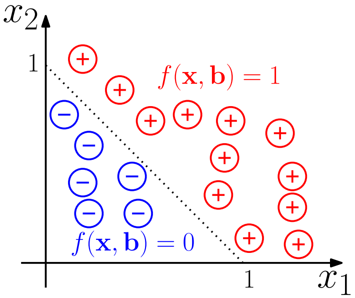
Classification

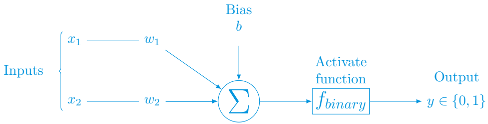


- Goal: Adjust the parameters (w_1, w_2 and b) of the neural network to classify correctly!
- Note that you don't have any influence over x_1 and x_2.
- Think of this as some game. You have to pick w_1, w_2 and b so that for any x_1 and x_2 you output the correct y!



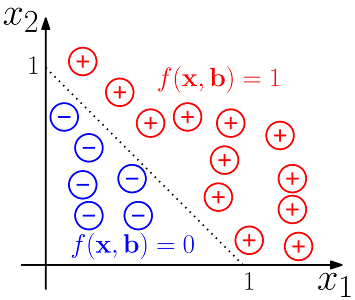
Classification

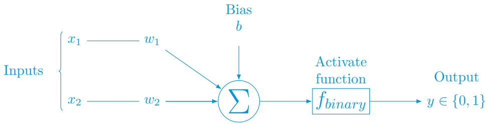


- 

- Recall $f(\mathbf{x}, \mathbf{w}) = \begin{cases} 1 & \text{if } x_1 w_1 + x_2 w_2 \geqslant b \\ 0 & \text{otherwise} \end{cases}$
- How do you choose to b, w_1 and w_2 to classify correctly?

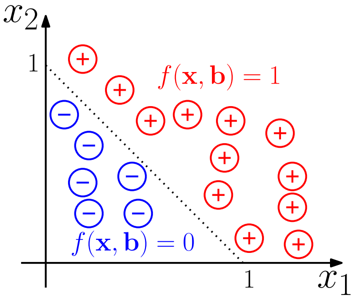
Classification

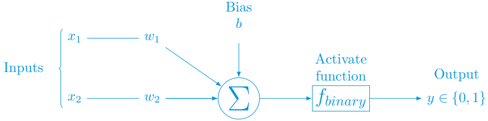


- 

- Recall $f(\mathbf{x}, \mathbf{w}) = \begin{cases} 1 & \text{if } x_1 w_1 + x_2 w_2 \geqslant b \\ 0 & \text{otherwise} \end{cases}$
- How do you choose to b, w_1 and w_2 to classify correctly?
- The ideal (dashed) line is given by $x_1 + x_2 = 1$

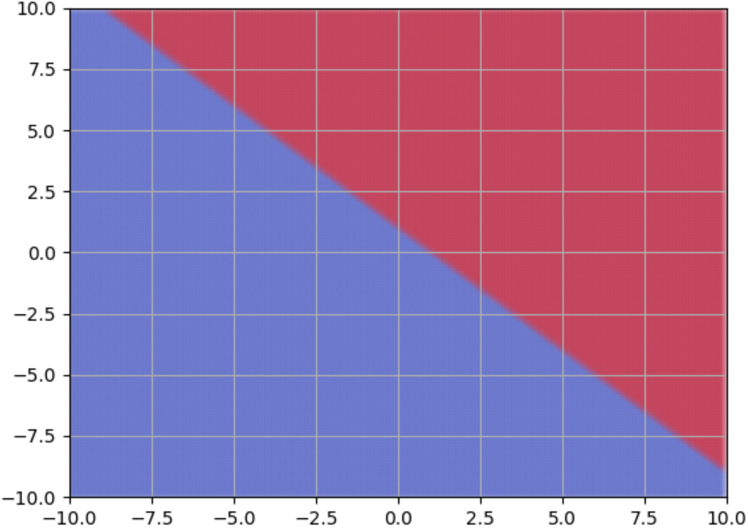
Classification



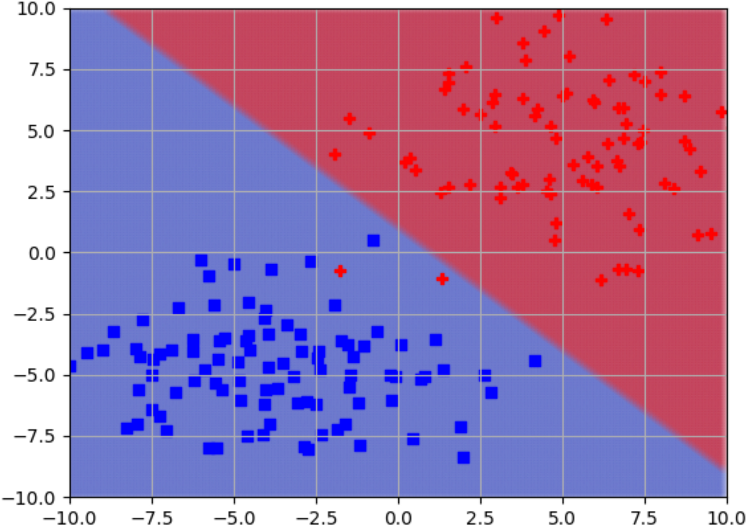
- 

- Recall $f(\mathbf{x}, \mathbf{w}) = \begin{cases} 1 & \text{if } x_1 w_1 + x_2 w_2 \geqslant b \\ 0 & \text{otherwise} \end{cases}$
- How do you choose to b, w_1 and w_2 to classify correctly?
- The ideal (dashed) line is given by $x_1 + x_2 = 1$
- Thus, choose $w_1, w_2, b = 1$

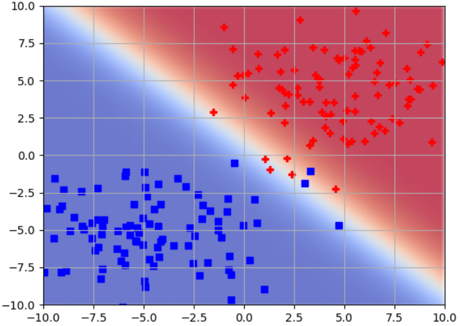
Space divided



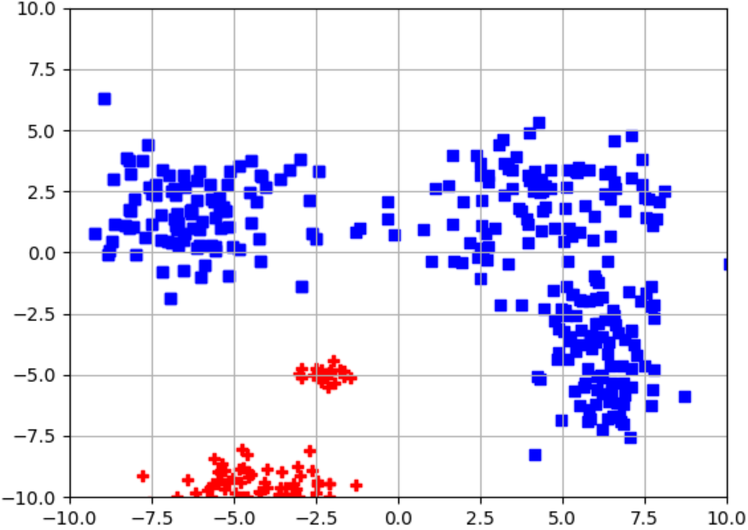
Problem With Binary



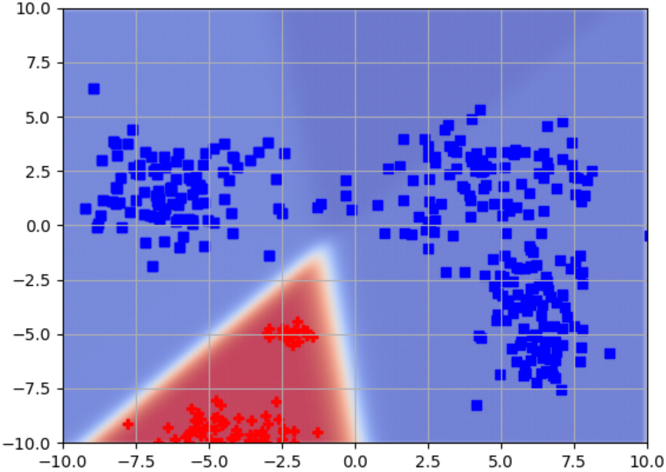
Better Sigmoid



What now?



More Neurons



- You cannot do this with one single neuron

Learning of hierarchical concepts - Joint work with Nancy Lynch (MIT)

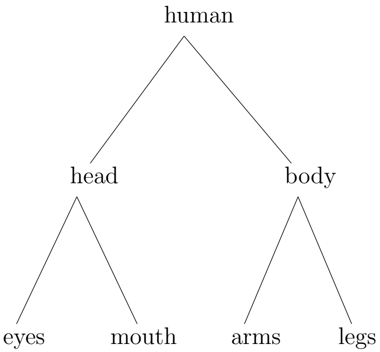
- What do you see here?

Learning of hierarchical concepts - Joint work with Nancy Lynch (MIT)

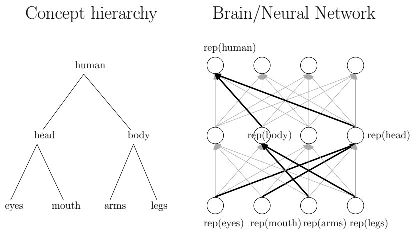
- What is a human?

Learning of hierarchical concepts - Joint work with Nancy Lynch (MIT)

Concept hierarchy



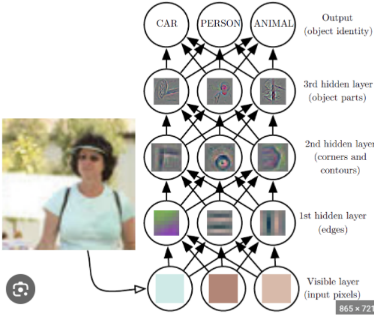
Learning of hierarchies



- Spoiler alert: We show, under some assumptions, that a mapping between the concepts and the neurons will naturally emerge.

Learning of hierarchies

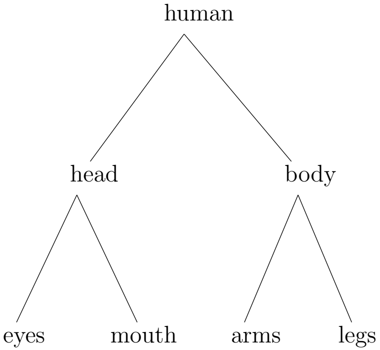
- It turns out that hierarchies are naturally imbedded in ANNs:



- What about the brain?

Learning of hierarchical concepts - Joint work with Nancy Lynch (MIT)

Concept hierarchy

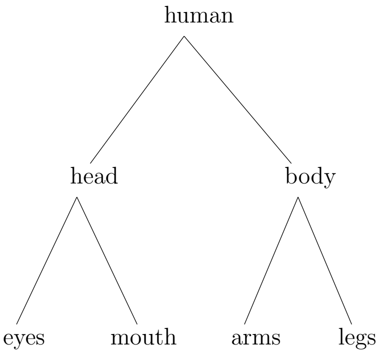


Prepare mentally

- Data Model (concept hierarchy)
- Network Model
- Neurons
- Noisy Recognition
- Learning
- Presenting Concepts

Data Model

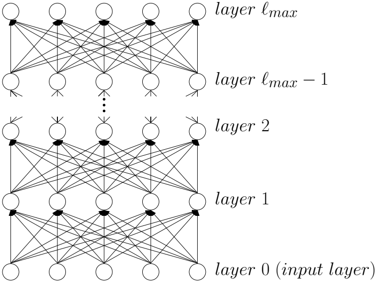
Concept hierarchy



- $\ell_{max} \in \mathbb{N}^+$: maximum level number for the concepts.
  Lowest level is 0. Here, $\ell_{max} = 2$
- $n \in \mathbb{N}^+$: total number of lowest-level concepts. Here, $n = 4$.
- $k \in \mathbb{N}^+$: number of sub-concepts per level. Here, $k = 2$.
- $r_1, r_2 \in [0, 1]$ with $r_1 \leqslant r_2$: Noise thresholds for recognition.

Network Model

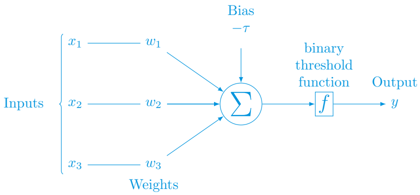
- We assume network (brain) has dimensions at least $n \times \ell_{max}$



- Fully connected between layers

Neuron Model

- Each neuron has the following states
  1. Firing ($\in \{0, 1\}$): If the neuron is currently firing
  2. Weight vector ($\in \mathbb{R}^n$): representing the incoming weights
  3. Engaged ($\in \{0, 1\}$): if the current is currently ready to learn



Neuron Model

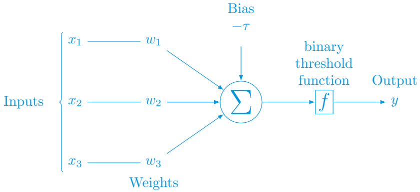
- To determine if a neuron will fire:

$$pot^u(t) = w^u(t-1)^T \cdot x^u(t-1) = \sum_{j=1}^{n} w_j^u(t-1)x_j^u(t-1).$$

The activation function, which defines whether or not neuron u fires at time t, is then defined by:

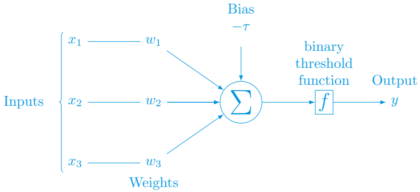
$$y^u(t) = \begin{cases} 1 & \text{if } pot^u(t) \geqslant \tau, \\ 0 & \text{otherwise,} \end{cases}.$$

where τ is the assumed firing threshold.



Neuron Mode - Example

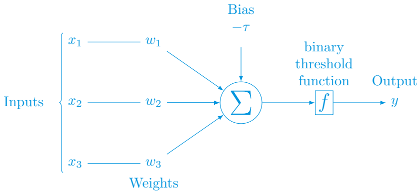
- Assume $w_1, w_2, w_3 = 1$ and $x_1(t-1) = 1, x_2(t-1) = 1$ and $x_3(t-1) = 0$
- Assume $\tau = 2$.
- Recall $pot^u(t) = \sum_{j=1}^{n} w_j^u(t-1)x_j^u(t-1)$.



- Does the neuron fire?

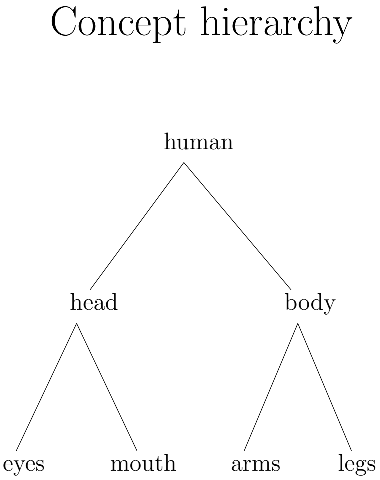
Neuron Mode - Example

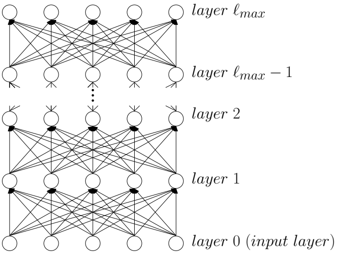
- Assume $w_1, w_2, w_3 = 1$ and $x_1(t-1) = 1, x_2(t-1) = 1$ and $x_3(t-1) = 0$
- Assume $\tau = 2$.
- Recall $pot^u(t) = \sum_{j=1}^{n} w_j^u(t-1)x_j^u(t-1)$.



- Does the neuron fire? **yes!**

Learning of hierarchies

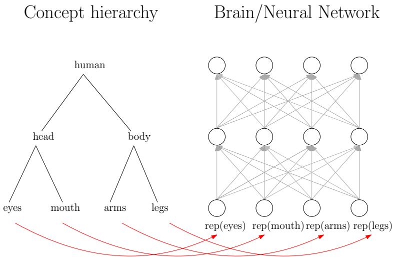




- Terminology: A concept hierarchy has **levels** and a NN has **layers**.

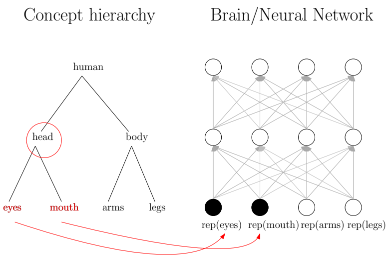
Network Model

- We assume the lowest layer represents the most elementary concepts. In our example: eye, mouth, leg, arm



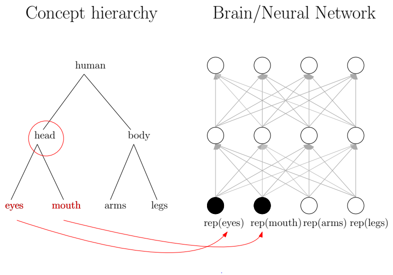
- Input: $\mathbf{v} \in \{0,1\}^n$ of layer-0 firings
- Output: $\mathbf{F} \in \{0,1\}^{n \times \ell_{max}}$ firing state of all neurons in the network.

Presenting a concept



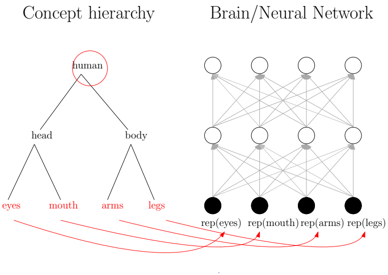
- When we **present** a concept, we assume that representatives of the bottom-level concepts are fed to the network.
- Example: When we present head, then rep(eyes) and rep(mouth) fires.

Presenting a concept



- Example: What if we present ”human”?

Presenting a concept

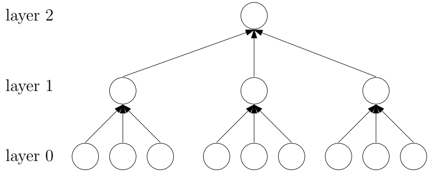


- Example: What if we present ”human”?

Goal

- What is the goal of our NN network?
- We want to see if we can embedded a concept hierarchy in the brain
- If a high-level concept is presented, we want one designated neuron to fire.
- All of that even when parts are missing?!

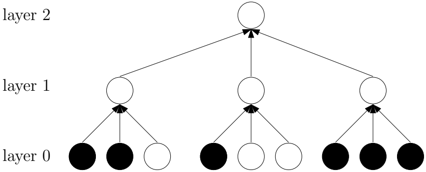
Noisy recognition



Noisy recognition:

- If $\geqslant 2/3$ of the sub-concepts are present, then the concept should be recognised
- If $\leqslant 1/3$ of the concepts are present, then the concept should NOT be recognised

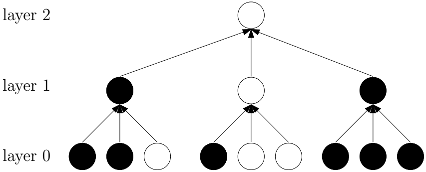
Noisy recognition



Noisy recognition:

- If $\geqslant 2/3$ of the sub-concepts are present, then the concept should be recognised
- If $\leqslant 1/3$ of the concepts are present, then the concept should NOT be recognised

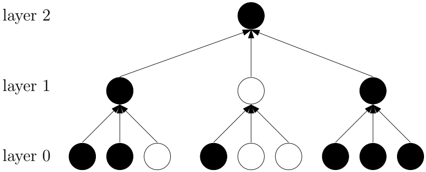
Noisy recognition



Noisy recognition:

- If $\geqslant 2/3$ of the sub-concepts are present, then the concept should be recognised
- If $\leqslant 1/3$ of the concepts are present, then the concept should NOT be recognised

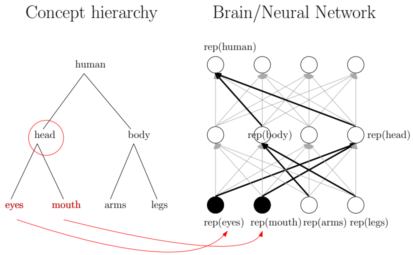
Noisy recognition



Noisy recognition:

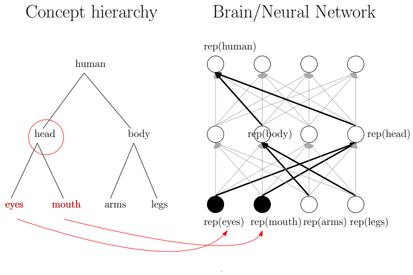
- If $\geqslant 2/3$ of the sub-concepts are present, then the concept should be recognised
- If $\leqslant 1/3$ of the concepts are present, then the concept should NOT be recognised

Noisy recognition - Proof that there exists an embedding



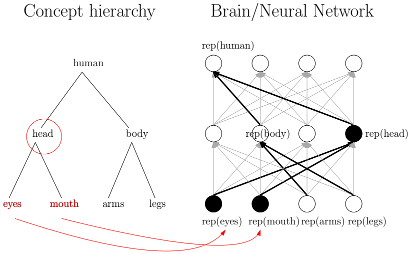
- Define a 1to1 mapping between the concepts and nodes (reps) in the network (at the correct layer)
- Set all weights (of edges) to either 1 if it connects two reps. Otherwise, set it to 0.
- Set the firing thresholds to $\tau = k/2$
- Let's go through the example

Presenting a concept



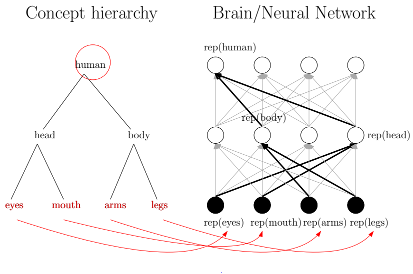
- When we **present** a concept, we assume that representatives of the bottom-level concepts fire
- Example: When we present head, then rep(eyes) and rep(mouth) fires.

Presenting a concept - Example



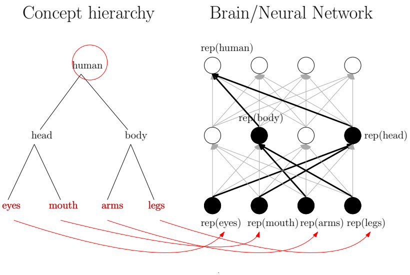
- When we **present** a concept, we assume that representatives of the bottom-level concepts fire
- Example: When we present head, then rep(eyes) and rep(mouth) fires.
- Then in the next step, we want rep(head) to fire.

Presenting a concept



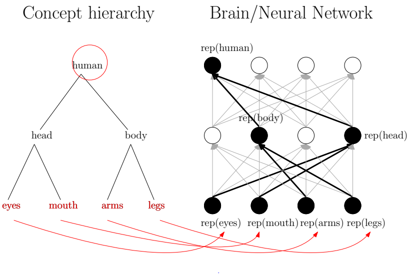
- Likewise, if we present the concept human, we want all reps of level 0 concepts to fire.

Presenting a concept



- In the next time step the reps of one layer higher fire

Presenting a concept



- And finally the rep of human fires.

Presenting a concept

- (Proposition) For every concept hierarchy $\mathcal{C}$ there exists a SNN that recognises $\mathcal{C}$.

Presenting a concept

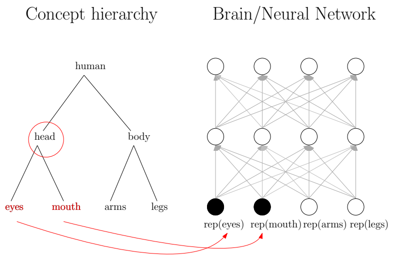
- (Proposition) For every concept hierarchy $\mathcal{C}$ there exists a SNN that recognises $\mathcal{C}$.
- Can we learn it?

Presenting a concept

- (Proposition) For every concept hierarchy $\mathcal{C}$ there exists a SNN that recognises $\mathcal{C}$.
- Can we learn it?
- Yes!

How should we update the weights?

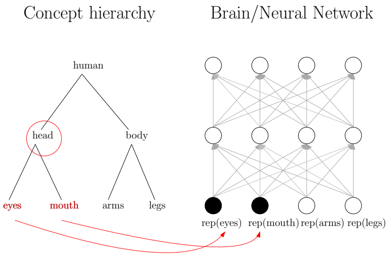
- Assume all weights are 1 at the beginning (empty slate / the mind is completely blank at birth).



-
- How should we update the weights?

WTA

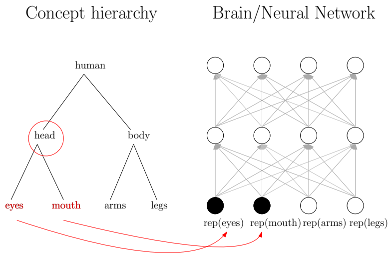
- Assume all weights are 1 at the beginning (empty slate).
- Which neuron should learn/change its weights?



-
- Problem: all weights are initially the same, we need a tie breaker.

WTA

- Assume all weights are 1 at the beginning (empty slate).
- Which neuron should learn/change its weights?



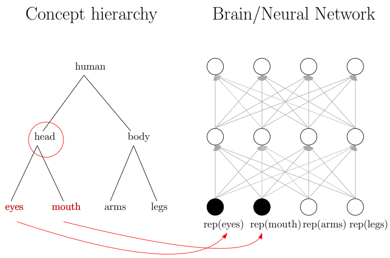
-
- Problem: all weights are initially the same, we need a tie breaker.
- Solution: Need Winner-Take-All Module that breaks the ties.

WTA

- Essential idea: select the neuron with the highest potential, ties broken arbitrarily, and set the **engaged flag to** 1**.**
- **Only this neuron is ready to learn.**

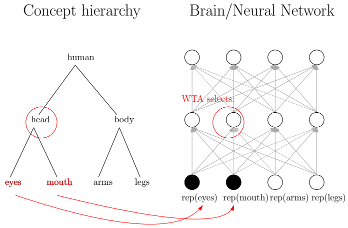
WTA

- Essential idea: select the neuron with the highest potential, ties broken arbitrarily, and set the **engaged flag to 1.**
- **Only this neuron is ready to learn.**



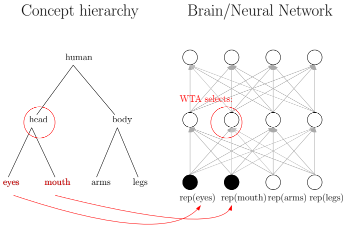
WTA

- WTA selects one neuron

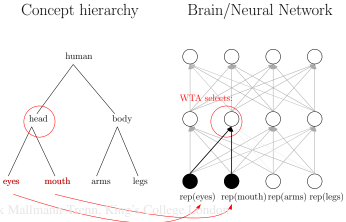


WTA

- WTA selects one neuron



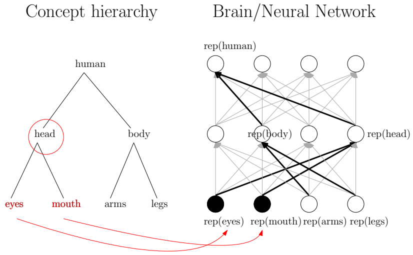
- The edges corresponding to the firing children will be strengthened, the other weakened — ”Neurons that fire together, wire together.”



Weight Update

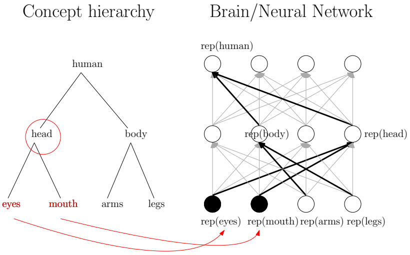
- We use Oja's rule to update the weights:
- Let $w \in \mathbb{R}^k$ be the current weight vector. Let $x \in \{0,1\}^n$ be the firing vector of the layer below.
- The updated weights are $w' = w' + \eta w^\top x(x - w^\top x x)$
- Is has the "neurons that fire together, wire together property"

Learning of hierarchies - The order matters



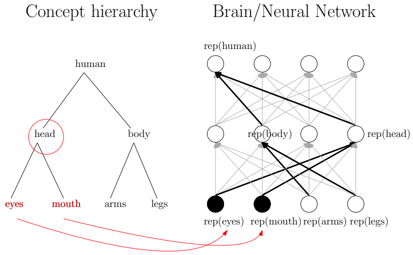
- Remember, when we **present** a concept, we assume that representatives of the bottom-layer concepts fire
- Example: When we present head, then rep(eyes) and rep(mouth) fires.

Learning of hierarchies - The order matters



- Remember, when we **present** a concept, we assume that representatives of the bottom-layer concepts fire
- Example: When we present head, then rep(eyes) and rep(mouth) fires.
- We assume that before presenting a concept c we present first all its children.
- Valid order: mouth, arms, legs, body, eyes, head, human

Learning of hierarchies - The order matters



- Remember, when we **present** a concept, we assume that representatives of the bottom-layer concepts fire
- Example: When we present head, then rep(eyes) and rep(mouth) fires.
- We assume that before presenting a concept c we present first all its children.
- Valid order: mouth, arms, legs, body, eyes, head, human
- Invalid order: mouth, head, arms, leg, body, mouth, human

Theorem

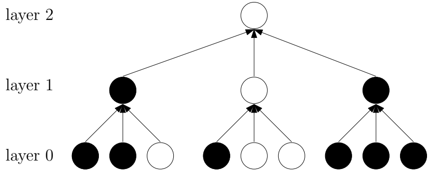
- (Theorem) Under all the assumptions above, the SNN will learn any concept hierarchy in a robust way.

Theorem

- (Theorem) Under all the assumptions above, the SNN will learn any concept hierarchy in a robust way.
- This even holds when the input for learning is noisy!

Noisy Learning

- Starting at the root, pick 70 percent of the neurons randomly and mark them as ”good”.
- Recurse on all ”good” neurons until the bottom level is reached
- The good neurons of the bottom level are then fed to the network



Proof Idea

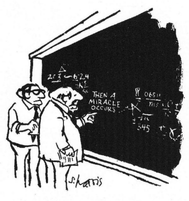
- Due the noise things will go wrong, we just consider intervals of size n^6 and argue that during it things will be fine w.h.p.
- We need to decrease the learning speed to make sure we have concentration around the expectation
- It turns out that the way the weights change depends highly on the other weights, which makes the analysis non-trivial.
- For this reason, we refrain from showing convergence of each weight separately.

Proof Idea

- We use the following potential function ψ. to show that the max and min weight convergence towards $\bar{w} = \frac{1}{\sqrt{pk+1-p}}$ and 0 respectively.
- Fix an arbitrary time t and let $w_{min}(t)$ and $w_{max}(t)$ be the minimum and maximum weights among $w_1(t), w_k(t), \ldots, w_k(t)$, respectively.
- Let $\psi(t) = \max\left\{\frac{w_{max}(t)}{\bar{w}}, \frac{\bar{w}}{w_{min}(t)}\right\}$.

Proof Idea

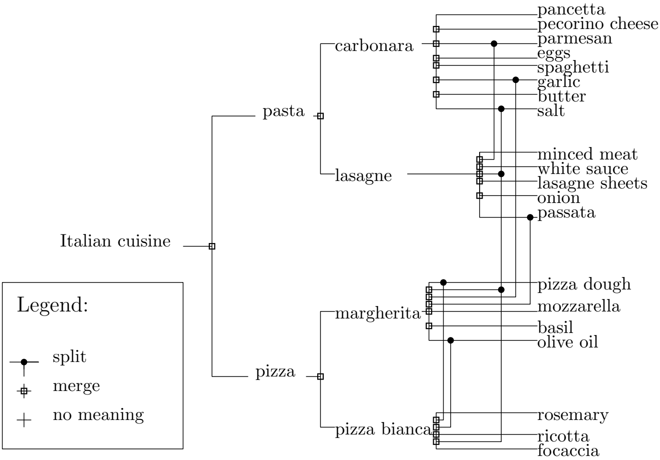
- Let $\psi(t) = \max\left\{\frac{w_{max}(t)}{\bar{w}}, \frac{\bar{w}}{w_{min}(t)}\right\}$.
- We show that the above potential decreases quickly until it is very close to 1.
- Showing that the potential decreases is involved, since one cannot simply use a worst-case approach, due to the terms in Oja's rule being non-linear
- potentially having a high variance, depending on the distribution of weights.
- Instead, the key to showing that ψ decreases is to carefully use the randomness over the input vector and to carefully bound the non-linear terms.
- Bounding these non-linear terms tightly presents a major challenge.
- To overcome it, we show that the changes of the weights form a Doob martingale allowing us to use Azuma-Hoeffding inequality to get asymptotically almost tight bounds on the change of the weights



"I think you should be more explicit here in step two."

Allow overlap

- Overlap



- requires new WTA!

Feedback

- Sometimes you can draw knowledge by looking at the super-concept to identify the current input
- Example: Cat

Recap

- Biological Inspiration
- Difference between ANN and SNN
- Data model (concept)
- Network model
- Noisy recognition
- Learning

Future Work

- Flexibility in order of learning
- Is-not relationship
- Different k
- level overarching concepts
- How is the WTA implemented?
- Are there simpler learning rules than (Oja's) - Let's work on this!

Thank you!

Questions?