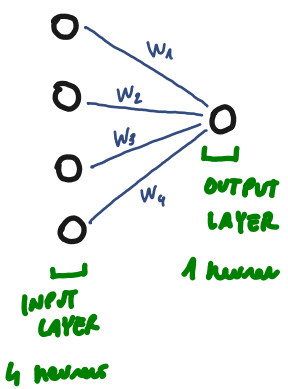
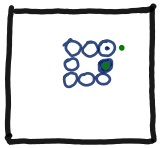
NEURAL NETWORKS:



Fully connected NN (Neural Network), each node for each layer is connected with each node of the other layer

Each connection has a weight

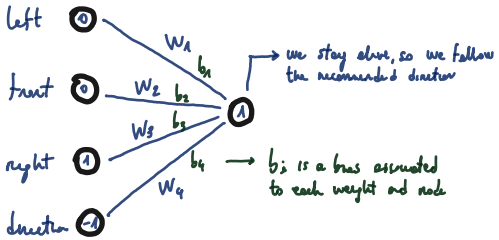
Example: SNAKE GAME → How do we expect the NN to perform?

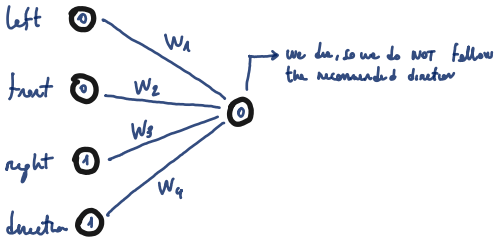


Train a NN that keeps the snake alive

Input: Do we have something in FRONT, LEFT or RIGHT of the snake + recommended direction
↳ encoded as: -1 = left, 0 = front, 1 = right

Output: what direction to go for
→ encoded as: 1 = we follow the recommended direction
0 = we do not follow the recommended direction





How do we design something like this?

Let's say y is the output and V_i the inputs: $y = \sum_{i=0}^{n} W_i V_i + b_i$ (b_i: bias)

If I play 1000 snake games, the NN will adjust W_i and b_i to obtain the desired input associated with each possible move

But how are the weights and biases tweaked?

Each weight and bias is associated with a specific move, so it's easy to know what parameters to tweak

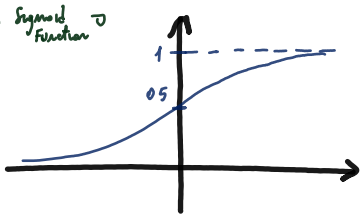
⇑

⇒ Let's say for example the output is 1 for a certain move, then the NN won't adjust W_i or b_i

⇒ If I get $y=0$, then the NN will adjust in order to make the output vary

Activation functions:

Non-linear function to add a degree of complexity to our NN:



⇒ Maps y from 0 to 1

for $y \to \infty$, $\sigma(y) \to 1$

So: $y = \left[\sum (V_i W_i + b_i)\right]$

$\sigma(y) = \sigma\left[\sum (V_i W_i + b_i)\right]$

⇒ Provides a complex output compared to the linear sum that is provided by the weighted sum of the $(W_i V_i + b_i)$

Loss Function:

How do we know how far off we are? How much to tweak the parameters? MSE usually