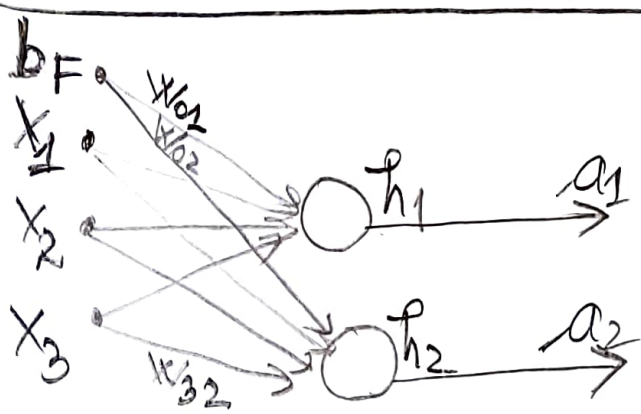
⊐ ≡ no conn. between hidden nodes

Restricted Boltzmann Machines (= symm., bipartite, bidirectional graph)

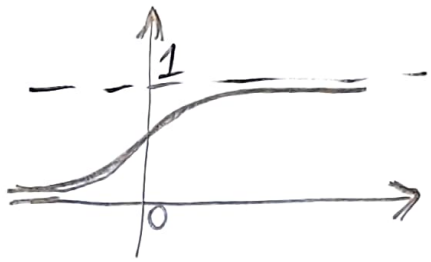


	Elephant	Dog
	1	0
	0	1

input (ex. pixels of images) ↑ visible layer

hidden layer

$a_1 = \sigma(\vec{w} \cdot \vec{x}) = \frac{1}{1+e^{-\vec{w}\cdot\vec{x}}}$ ← output after activation fct.



Step 1 Forword pass (like in FFNN)

↳ captures $P(a|x)$

↳ Q Given the input pixels, should my weights send a signal to Eleph. or Dog node?

Step 2 Backward pass = reconstruct input x_i from output a_i

↳ captures $P(x|a)$

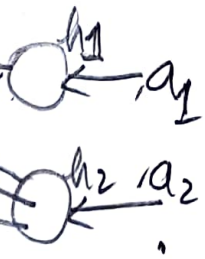
↳ Q Given an Eleph., which distrib. of pixels should I expect?

new biases ↓

$r_1 = b_{B1} + w_1$

$r_2 = b_{B2} + w_2$

$r_3 = b_{B3} + w_3$



ip. ← op. (reconstructions of inputs)

Reconstr. error = $\vec{r} - \vec{x}$ ⇒ update $\vec{w}$ to min. it

(uses the "contrastive divergence")

↳ all $\vec{w}$, including those for b_F (bias fowd) & b_B (bias backward)

RBM
- can be used for
  - supervised
  - unsupervised
  - > learning
- can be used for
  - feature learning
  - dimens. reduction
  - collaborative filtering
  - classif.
  - regression

Obs.) We need the extra bias b_B (bias at Backward pass) to model a_i into x_i.