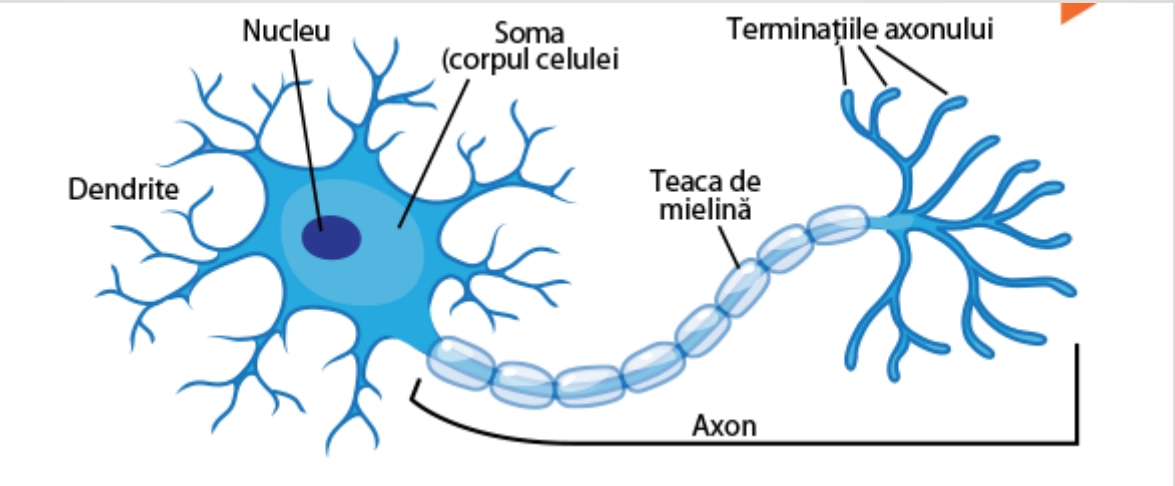
LABORATOR 7

RETELE NEURONALE ARTIFICIALE

RETELE NEURONALE ARTIFICIALE

- O alta metoda de identificare a functiei f
- Inspirata din biologie



“BUNNY TEST”

- Un iepuras clipeste daca ii dai cu spray in ochi.

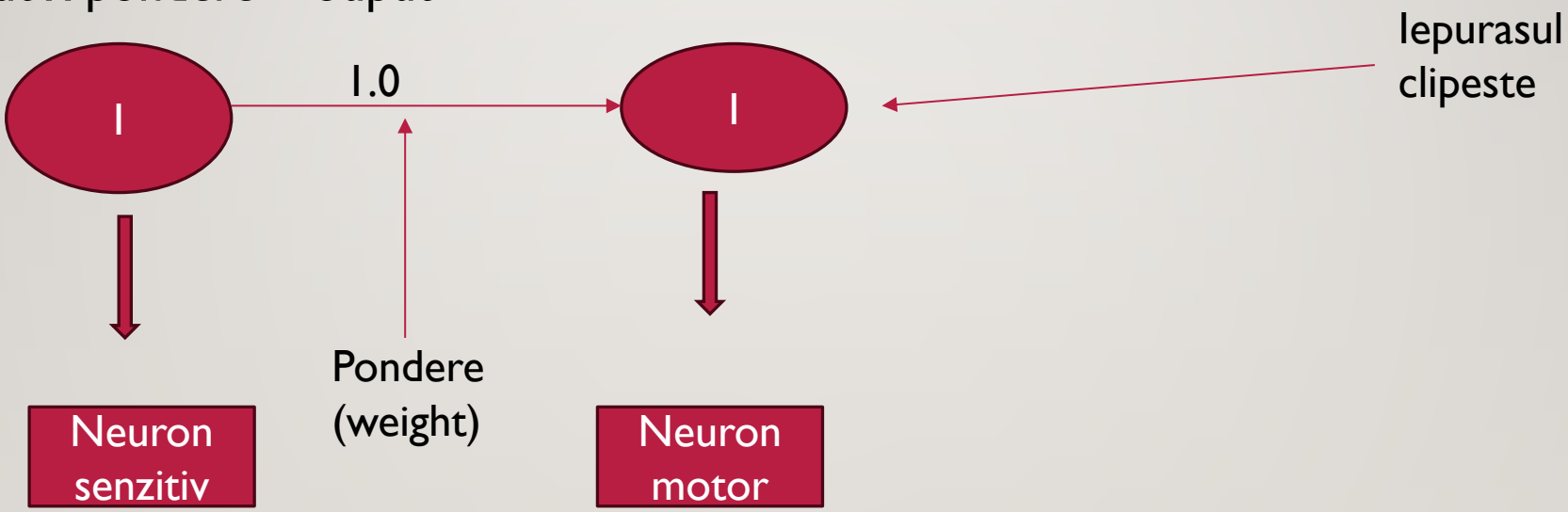
Clipit → 1
Neclipit → 0

Folosim spray→1
Nu folosim spray→0

Stimul

"BUNNY TEST"

- input x pondere = ouput



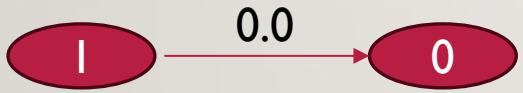
- 1 x 1.0 = 1 → iepurasul clipeste

“BUNNY TEST”

- Vrem ca iepurasul sa clipeasca la zgomot puternic

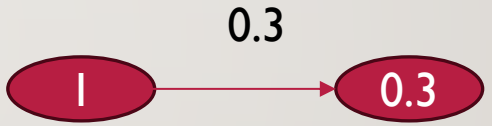
“BUNNY TEST”

- Faza de antrenare → output >=1 clipire
- Stare initiala



T1
(zgomot)

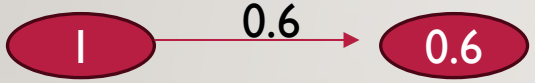
Dupa primul experiment



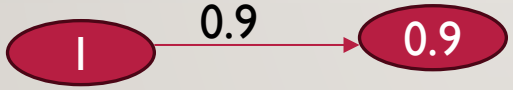
1 x 0.3 =0.3 <1 Iepurasul inca nu clipeste

“BUNNY TEST”

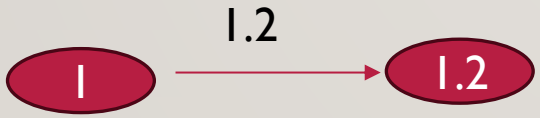
- Dupa al doilea experiment



- Dupa al treilea experiment

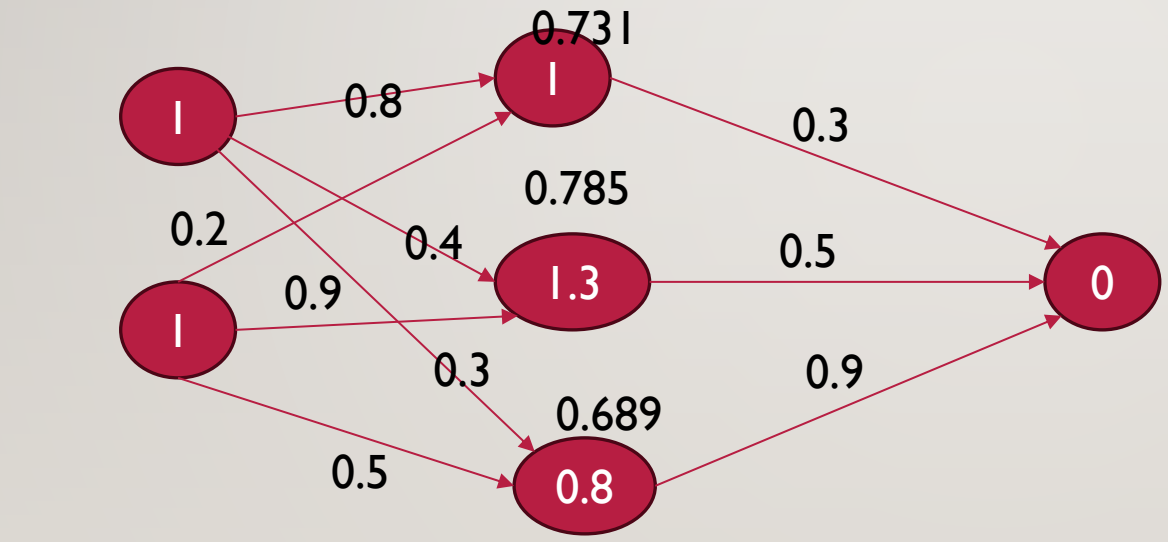


- Dupa 4 experimente – am invatat iepurasul sa recunoasca zgomotul si sa clipeasca



MODELARE RETEA NEURONALA ALGORITMUL FORWARD PROPAGATION

x1	x2	Y
1	1	0



Input layer

Hidden layer

Output layer

$H_{1net}=1*0.8+1*0.2=1$
$H_{2net}=1*0.4+1*0.9=1.3$
$H_{3net}=1*0.3+1*0.5=0.8$

Informatia ce intra in hidden layer

Neuronul proceseaza informatia→ se foloseste functia de activare

- Functia sigmoid $y=\frac{1}{1+e^{-x}}$

MODELARE RETEA NEURONALA-FORWARD PROPAGATION

- f(1)=$\frac{1}{1+e^{-1}} = 0.731$
- f(1.3)=$\frac{1}{1+e^{-1.3}} = 0.785$
- f(0.8)=0.689
- O_{net}=0.73*0.3+0.79*0.5+0.69*0.9=1.235
- Aplicam functia sigmoid f(1.235)=0.774
- Target: 0
- Calculat: 0.774

FORWARD PROPAGATION

- Structura retelei:
  - STRAT DE INPUT – are atatia neuroni cate attribute are un set de date
  - STRAT DE OUTPUT- are un neuron pentru un singur output
  - STRATURILE ASCUNSE-pot avea oricati neuroni

- Hidden layer (strat ascuns)- transforma inputul in ceva ce poate fi folosit de output
- Output layer transforma activarile layerului hidden
- Eroarea = target- calculat=0-0.774=-0.774

BACKWARD PROPAGATION

- Backpropagation = metoda prin care rețeaua neuronală **învață din greșeli**
- Eroare → Înapoi prin rețea → Actualizare ponderi
- [Sigmoid(output)]' = output *(1-output)
- $\delta = (y - output) \cdot output(1 - output) - pt\ output\ layer$
- $\delta_j = (\sum_k w_{jk}\, \delta_k) \cdot output_j(1 - output_j) - pt\ hidden\ layer$
- Actualizare ponderi:
  - $w = w + \text{learningRate} \cdot \delta \cdot x(\ unde\ x\ este\ input)$