Universidade Estadual do Rio de Janeiro
Redes Neuronais
Raeli Wiedlmann

Neuronal: é relativo ao neurônio que compõe o sistema cerebral e não relativo ao sistema nervoso central.

Bibliografia:

- Introduction to the theory of Neural Computation, J. Hertz, A. Krogh R.G. Palmer, Lecture Notes Vol I Santa Fe Institute, Studies in the Science of Complexity, Perseus Books.

o mais importante

- Neural Networks A comprehensive Foundation, S. Haykin, Macmillan Publishing Co.

- Data Mining, A mineração de Dados no Marketing, Medicina Economia, Engenharia e Administração, Luis Alfredo Vidal de Carvalho, Editora Ciência moderna.

→ Fazer a disciplina de Algoritimos (IMAG

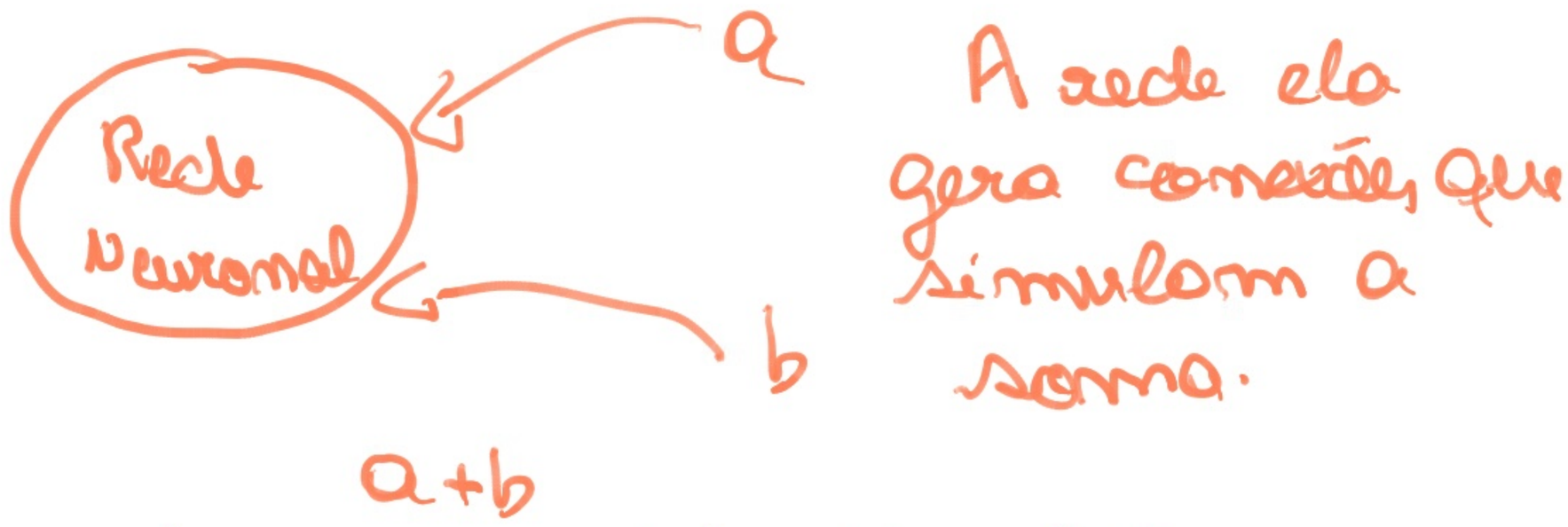
1. Introduction:

O cérebro humano pode realizar muitas tarefas melhor do que sistemas de IA. Por exemplo, o processamento de imagens

Computadores, por sua vez, realizam melhor tarefas aritméticas.

1.1. Aprendizado simbólico

1.2 - Aprendizado conexionistas: o paradigma conexionista



Diferentemente no modelo simbólico que pega o simbolo "a" e "b" e gera a soma "a+b".

1.3 - Neurocomputação

1.4 - Redes criativas

1.5 - Computação coletiva:

Processamento distribuído X Processamento seqüencial

O processamento distribuído realiza diversas tarefas em paralelo, a exemplo do cérebro, que realiza o processamento de imagens em paralelo

com a respiração e batimentos cardíacos. Já o processamento sequencial, nos computadores, realizam somente uma tarefa de cada vez.

A computação em paralelo é uma forma de se tentar aproximar a capacidade de processamento do cérebro humano. Nossa evolução biológica convergiu o nosso cérebro p/ um tipo de processamento em paralelo.

A grande vantagem da Rede neuronal Artificial é que ela tem a capacidade de resolver sistemas físicos não-lineares.

A evolução de sistemas no tempo, a exemplo da colônia de formigas são exemplos de modelagem de redes neuronais

Robustez e falha de tolerância. Células nervosas morrem todo o dia e a performance do cérebro não é significativamente afetado.

Os ajustes e flexibilidades do ambiente pelo aprendizado

O cérebro consegue ajustar a informação nebulosa, probabilística, ruidosa e inconsistente.

É altamente paralelo

Ele é compacto, pequeno, e dissipa pouca energia.

Pois, então, a computação neuronal é uma alternativa a computação de Von Neumann ou paradigma do (programação sequencial) comumente usadas em computadores.

Logo as RNA's são inspiradas na neurociência, apesar de não ser realístico biologicamente.

Os métodos advêm da física estatística.

As aplicações são em ciências da computação e engenharia. É usado como um paradigma de modelagem em neurociência, sensorial e psicologia.

Mecânica estatística

A inspiração da neurociência

Os modelos são extremamente simplificados

Rio, 16/08/17

Aula 2:

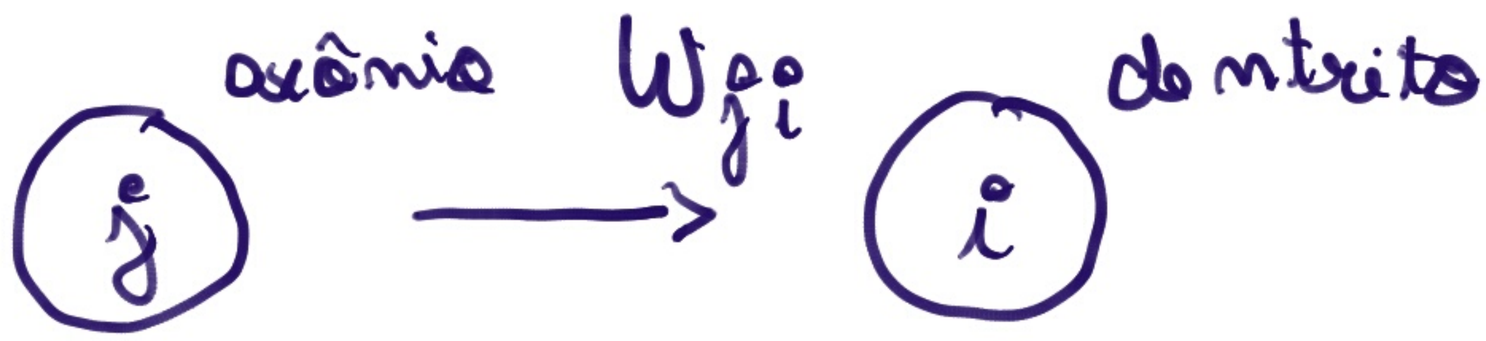
Dra Roseli : (21) 98895-2180

Cérebro ≈ 10^{11} neurônios

O final do axônio contém os botões transissores dos impulsos elétricos que se chamam sinapses.

Junção sináptica. Conjuntos de lipídio os chamados canais iônicos abem "portas" que dêxa passar íons, por meio de substâncias, suscetíveis à campos elétricos, que por sua vez são estimulados por correntes elétricas (Principles of neuroscience - Eric Kandel). Esse procedimento gera uma DDP que realiza a transmissão do sinal

Analogamente a RNA faz uma representação simplista desses processos e neurorecepções por W_{ji}.

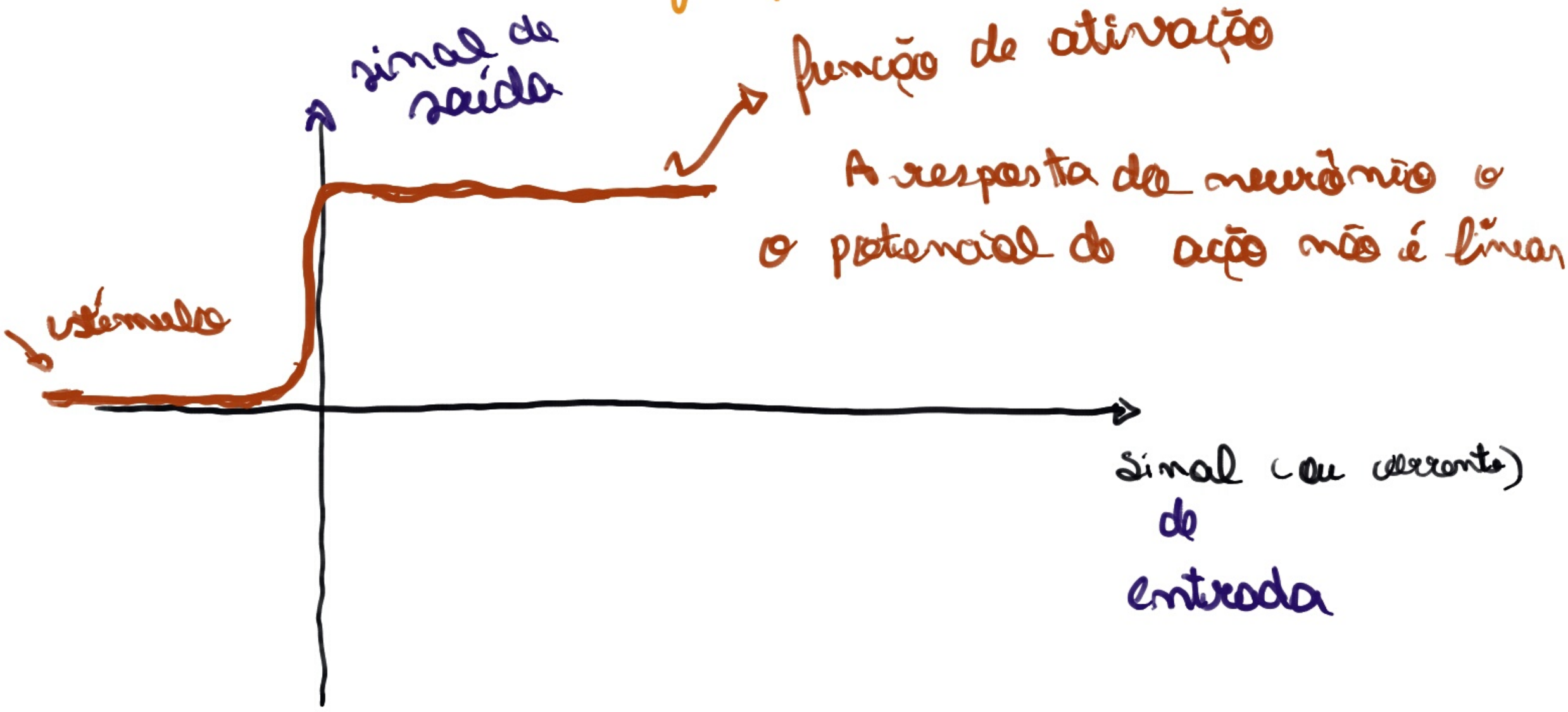


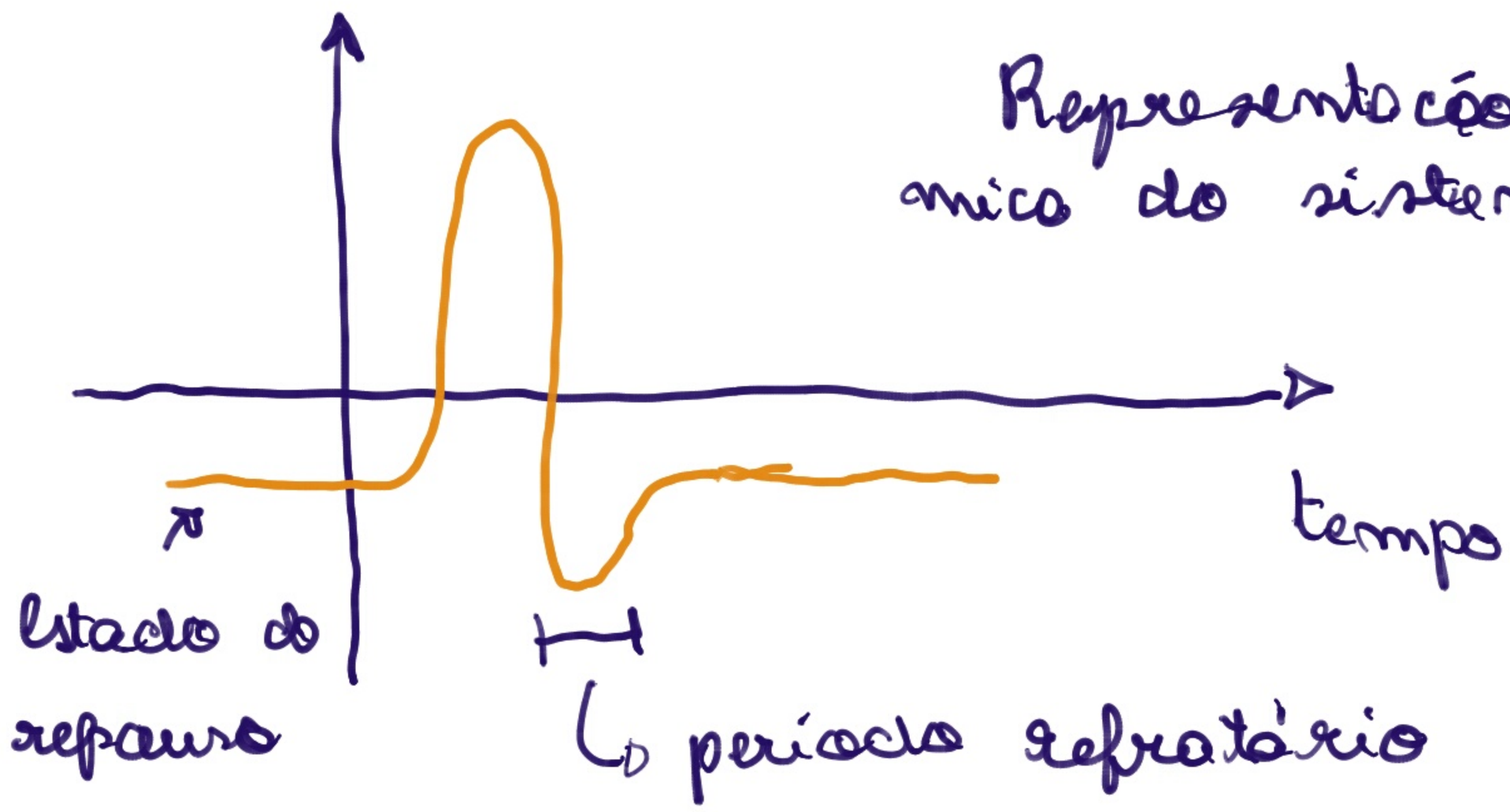
neurossubstâncias
- neurotransmissores: são as substâncias que são transmitidas e Atuam no curto espaço de tempo.
- neuromoduladores: alteram a capacidade de transmissão de informação de longo prazo.

Buscar na internet as Palestras de Charles Rose

Parâmetro do modelo é o peso sináptico! Ele é descrito como W_{ji} (onde j é a origem e i o destino)

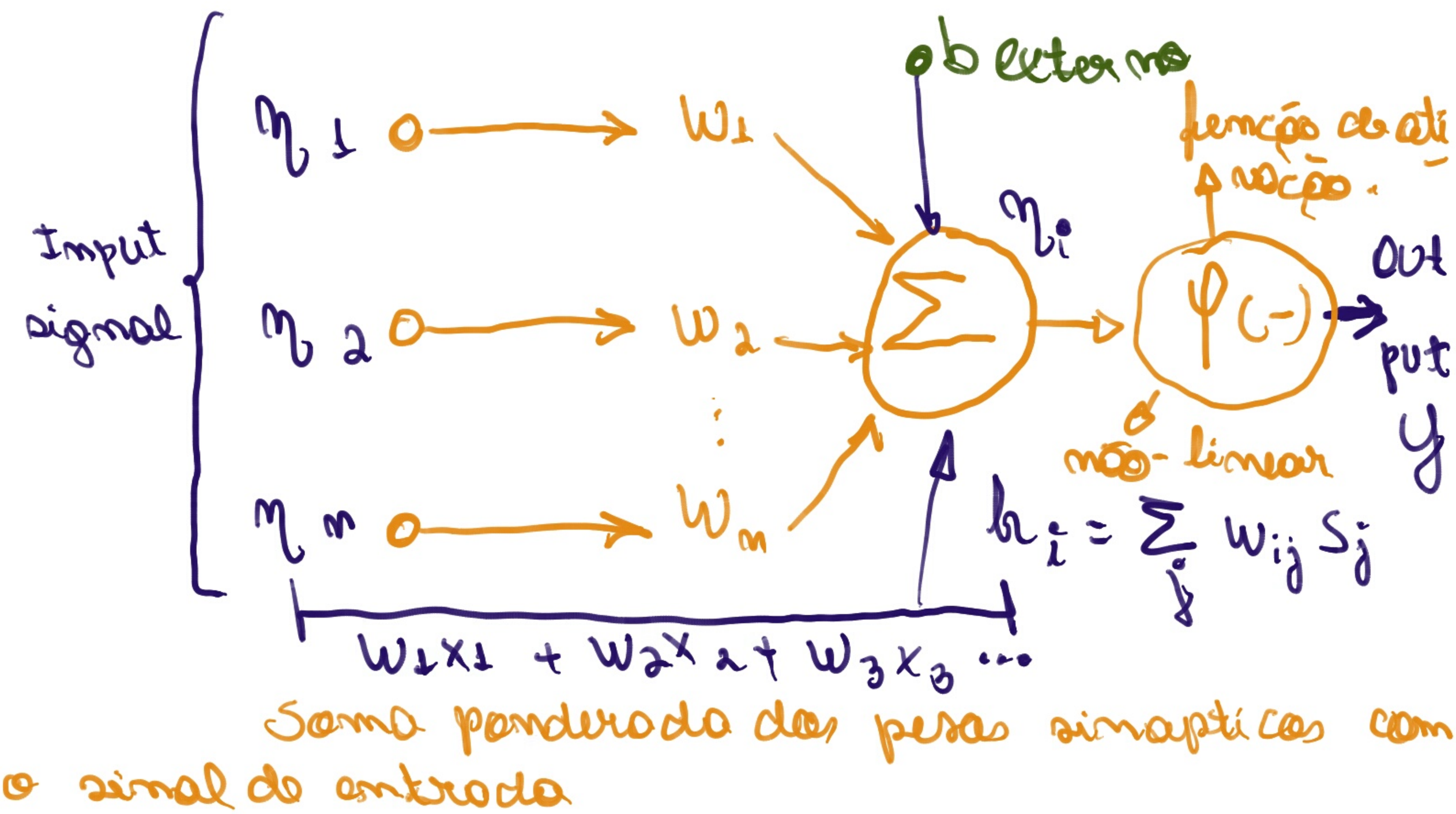
Pesquisar o que um grafo!





Ao término do período refratário (após o disparo) a célula está pronta para receber um novo estímulo.

O modelo computacional de um Neurônio



Soma ponderada das pesas sinapticas com o sinal de entrada

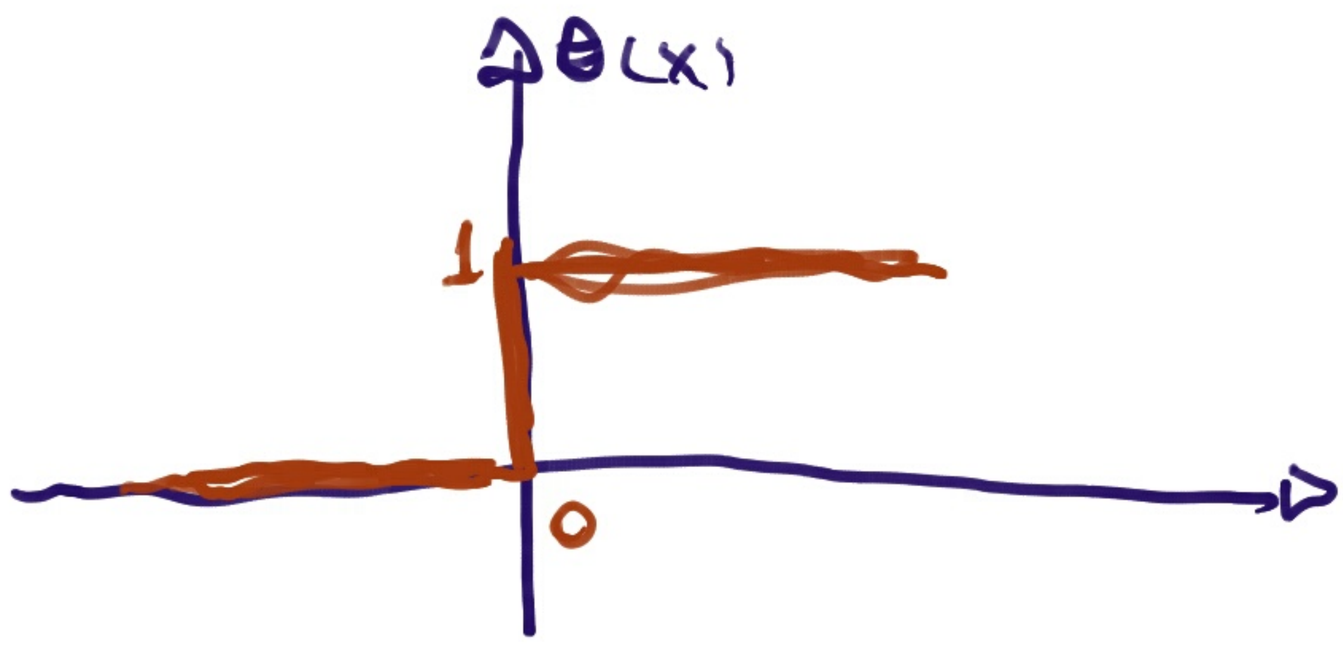
McCulloch-Pitts (1943)

Foram os primeiros a modelar matematicamente um neurônio

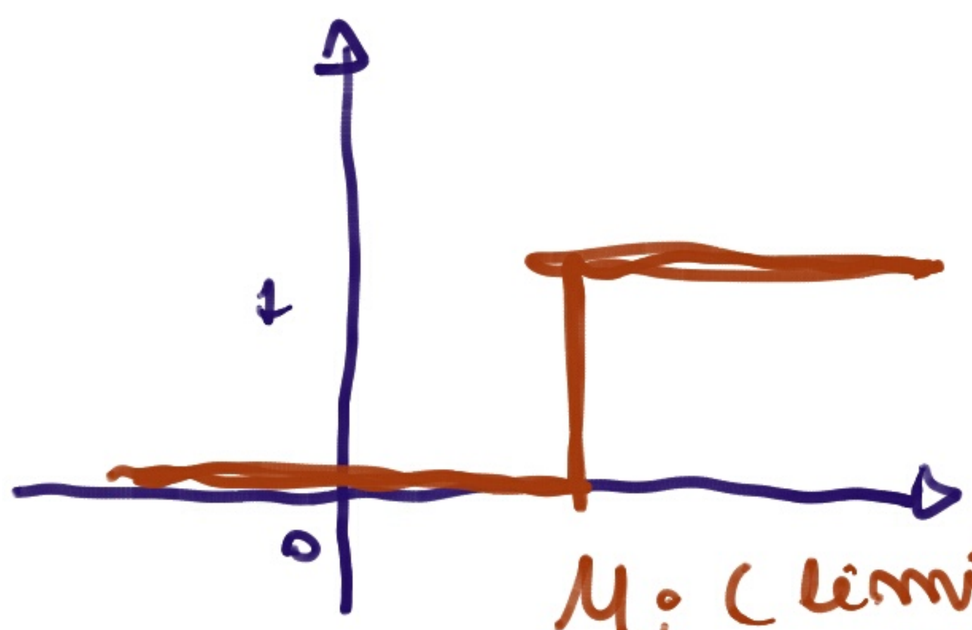
$$h_i^{(t)} = \sum_j w_{ij} \, \eta_j(t)$$

$$\eta_i(t+1) = \underbrace{\Theta}_{\text{função degrau}}(h_i - \underbrace{\mu_i}_{\text{limiar de ativação}})$$

$$\Theta(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{quando contrário} \end{cases}$$



O μ_i vai regular aonde a função vai ser descontínua.



μ_i (limiar de ativação ou threshold)

$$\eta_i(t+1) = \Theta(\underbrace{\sum w_{ij}\,\eta_j(t)}_{h_i} - \mu_i)$$

Quando $w_{ij}(+)$ é chamado de sinapse excitatória e $w_{ij}(-)$ é chamado de sinapse inibitória.

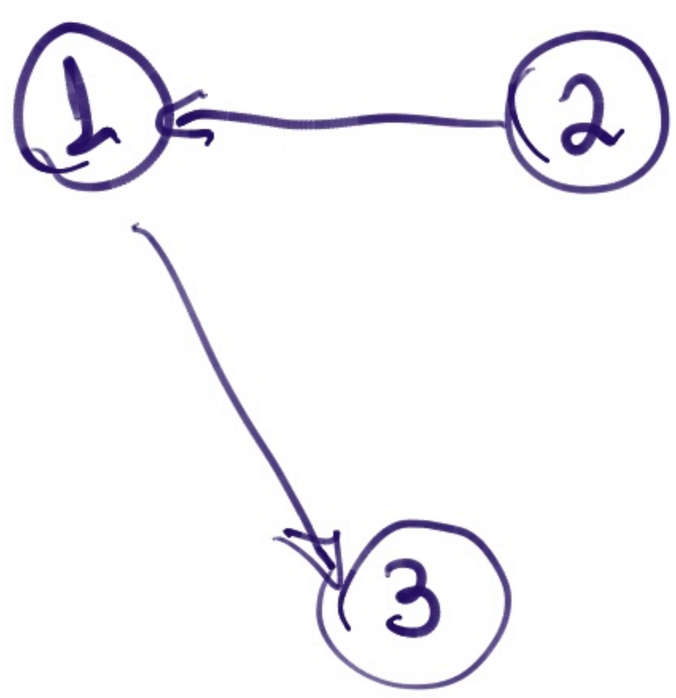
Somas ponderadas devem atingir ou exceder o μ_i para um neurônio disparar.

Artigo: Wedemann, Avalanche: Physica A
↳ buscar a referência do Arcdrangelis. => replica a modelagem sinapses inibitórias. (OBS: funciona como um tipo de filtro)

Modelo matemático simples, não-linear.

Eles provaram que uma assembléia síncrona é capaz de realizar computação universal. Ou seja, dependendo da topologia do grafo você será capaz de selecionar qualquer função.

Definindo um Grafo: são definidos pelos items I, II, e III



I - Vértices = {1, 2, 3}

II Arestas = {(2,1), (1,3)}

III $W_{(peso)} = \{W_{12}, W_{31}\}$

A saída é proporcional a entrada.
A entrada ponderada é o que as células reais fazem. $h_i = W_{ij} \eta_j$

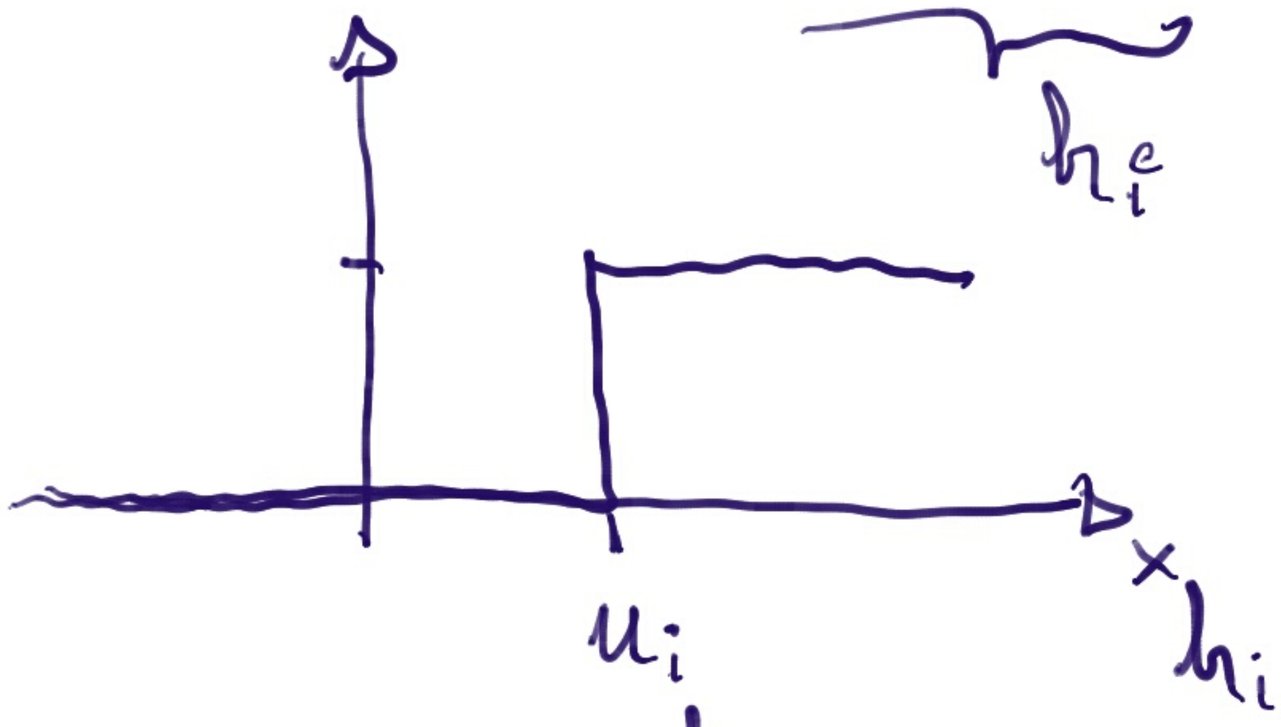
Um neurônio real produz uma seqüência de pulsos. Os modelos que se aproximam dessa realidade é chamada de Spiking neural Network.

Rio, 23/08/2017

Recapitulando:

Estado do neurônio é a intensidade do sinal

$$\eta_i(t+1) = \Theta\Big(\underbrace{\sum_j \omega_{ij}\, n_j(t) - \mu_i}_{h_i^e}\Big)$$



$$\underbrace{\overset{h_i}{x} > \mu_i}$$

$$\underbrace{\overset{h_i}{x} - \mu_i > 0}$$

$$\eta_i(t+1) = \Theta(h_i(t) - \mu_i)$$

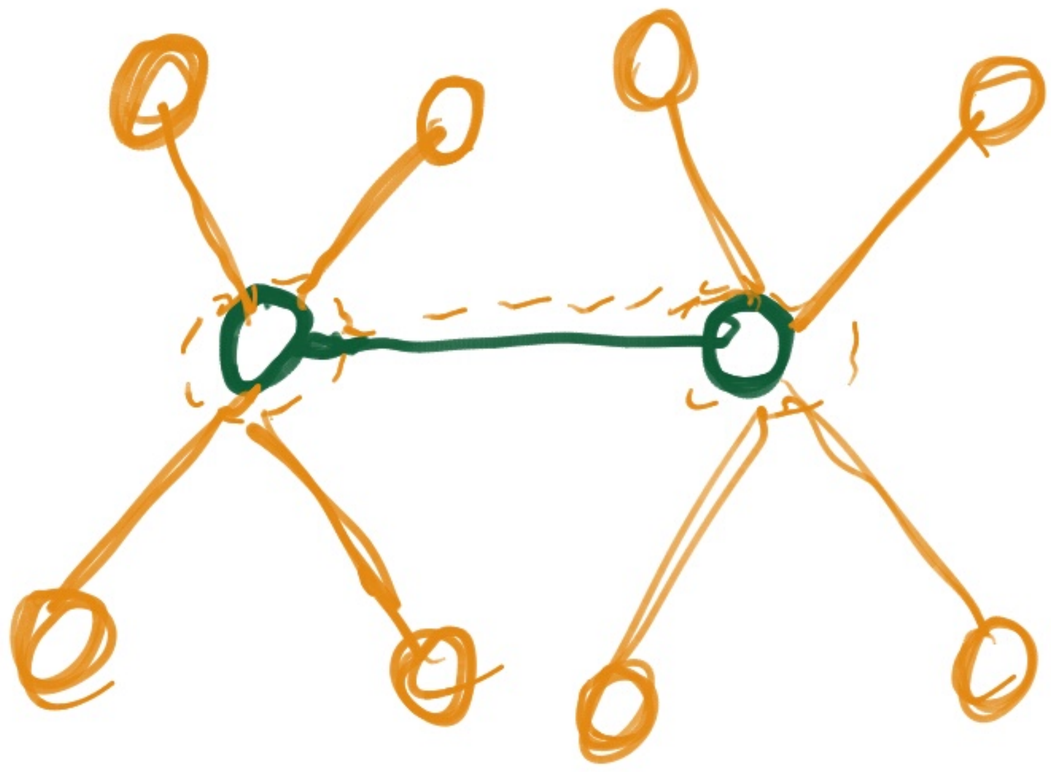
degrau, discreta, a derivada no ponto μ_i tende a infinito.

$$n_i := g\Big(\sum_j \omega_{ij}\, n_j - \mu_i\Big)$$

atribuição, em algoritimos o sinal de atribuição diz que a variável n_i assumirá os valores de g (função de ativação ela te dá o dinâmico da mudança de estado). n_i é chamado de estado.

O tempo que a informação leva para ser transferida pode ser modelado de maneira síncrona ou assíncrona.

O princípio é chamado **detecção de neurônios dependentes**.

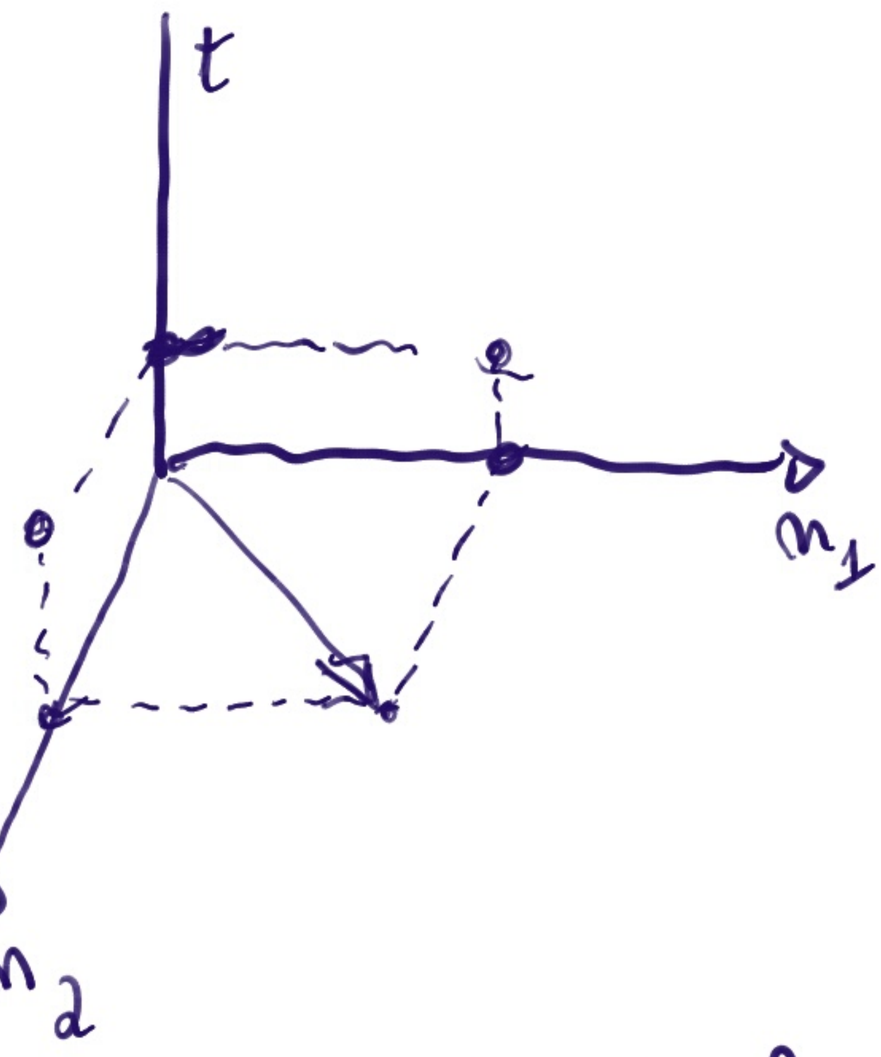


ativação nível 1
ativação nível 2

Em algoritmos sequenciais a atualização não é síncrona.

Como se mapeia o caminho do sinal na rede?

Ele é uma função do estado do neurônio em função do tempo. O estado da rede, portanto, é um conjunto $\{n_1, n_2, n_3\}$, ou seja dos estados dos neurônios.

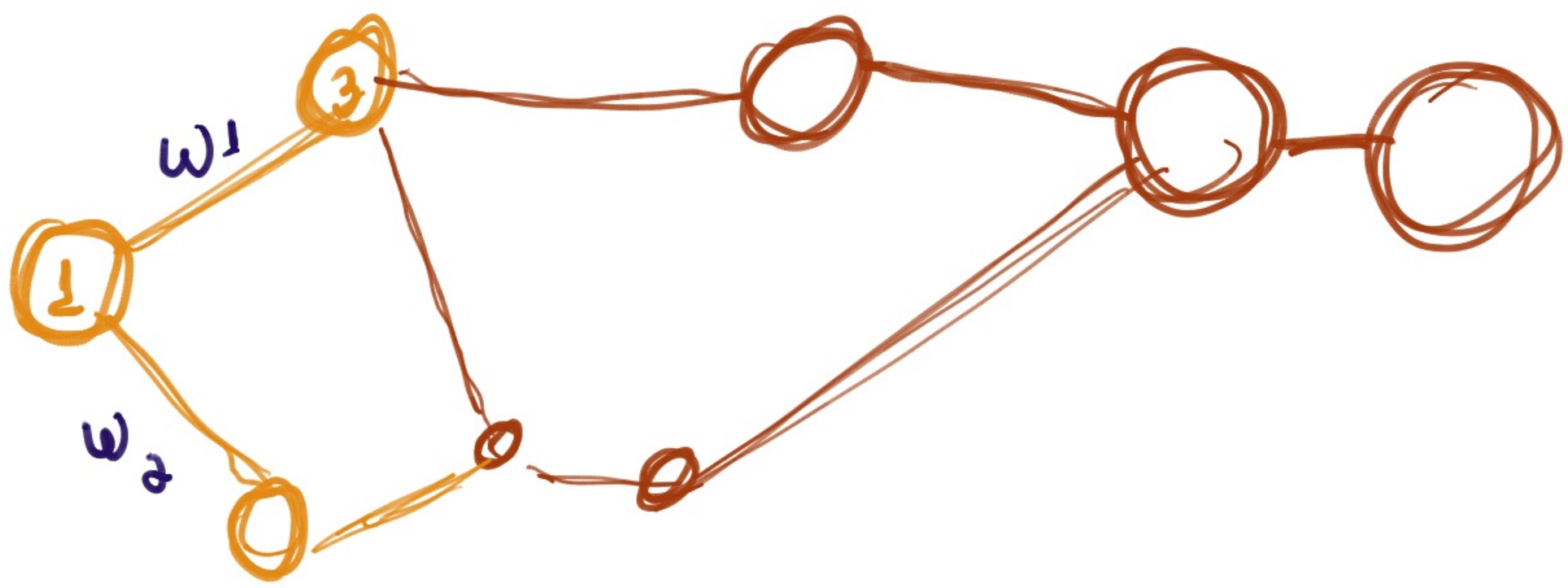


Representam-se a rede através de grafos:
Quando eles possuem somente uma direção entre os estados dos neurônios, eles são chamados de feedforward network. Quando estão não ligados entre os estados através de laços entre neurônios

ou neurônios com alimentação bi-direcional.

Em RNA's modelos de neurônios são chamados de unidade de processamento, enquanto que as sinapses são tratadas como pesos ou conexões.

$$n_i := g(h_i - \mu_i) \; ; \; h_i = \sum_j w_{ij} n_j$$



I - $$h_i = \sum_{j \in \text{vizinhos de } i} w_{ij} n_j$$

II - Robustez, o sistema não se degrada com a retirada de neurônios. Capacidade de se adaptar a ruídos do meio.

ler sobre lógica de Von Neumann

ler o parte 1 e 2 e 1.3 do livro

próximo aula o modelo de Hopfield.

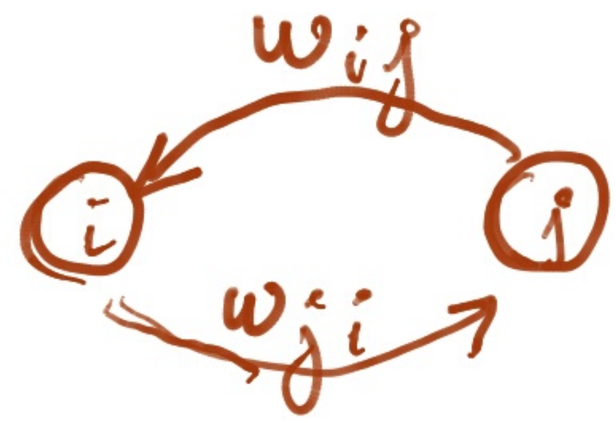
Rio, 04/10/2017

O modelo de McCulloch-Pitts (1943)

$$n_i(t+1) = \Theta\left(\sum_j w_{ij}\, n_j(t) - \mu_i\right)$$

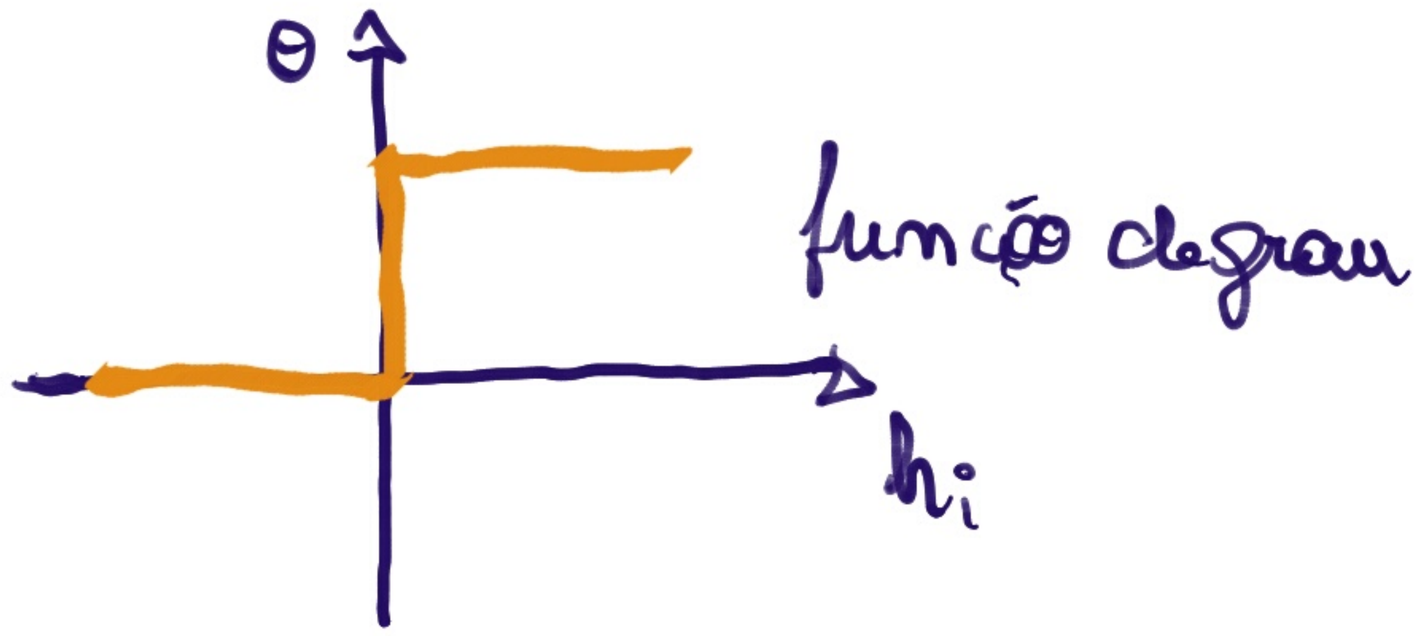
$$n_i \in \{0,1\} \text{ ou } \{-1,+1\}$$

$$h_i = \sum_{j \in \text{vizinhanças}} w_{ij}\, n_j(t)$$



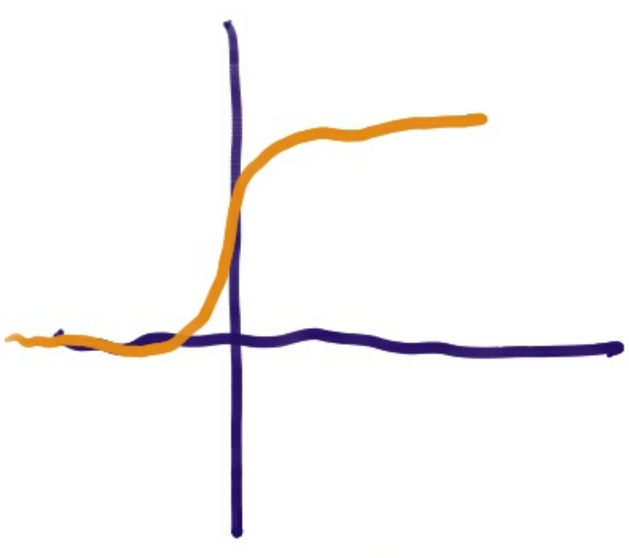
$$\eta_i(t+1) = \Theta(h_i - \mu_i) \qquad [\text{seção } 1.1]$$

Θ → função de ativação



função degrau

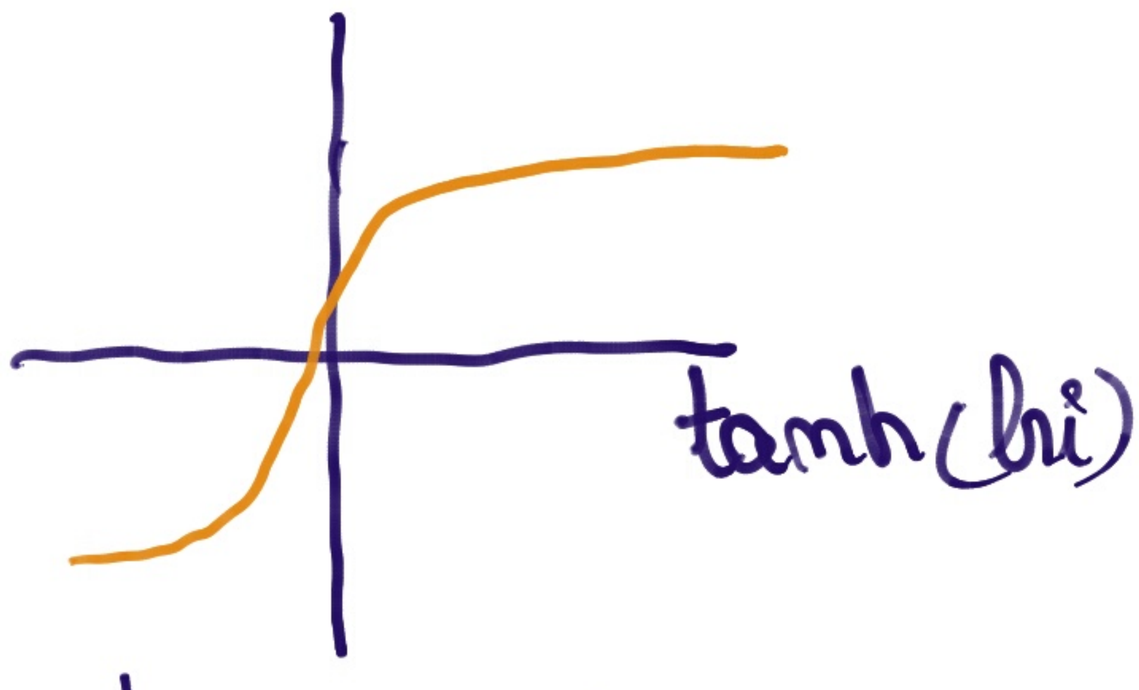
O Θ pode ser substituído por modelos contínuos.



sigmóide

$$\frac{1}{1+e^{-\beta(h_i-\mu_i)}}$$

ou



tangente hiperbólica

Na matemática neuronal as sinapses são chamadas de unidades e as conexões são chamadas de pesos.

O modelo de Hopfield

Ele foi o primeiro modelo a armazenar "memória".

I- O problema da memória associativa.

p: nº de padrões

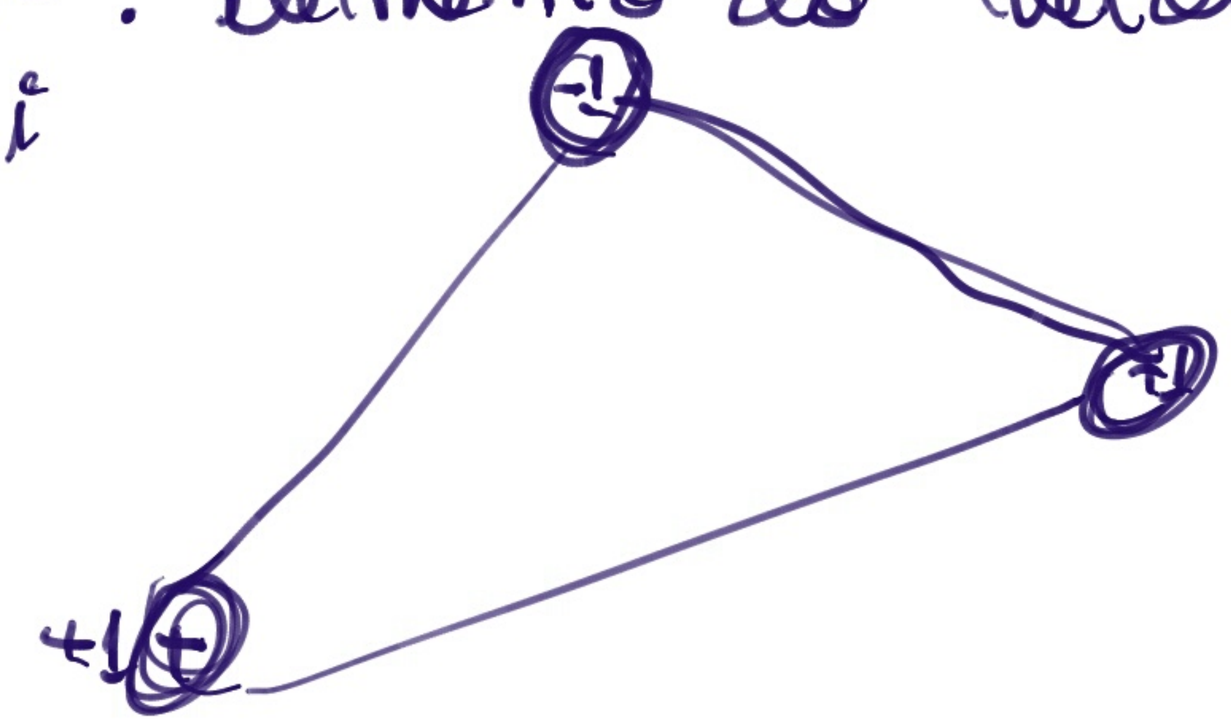
N: nº de neurônio

$\mu \in \{1, 2, 3, \ldots, p\}$

$i \in \{1, 2, \ldots N\}$

$\vec{\xi}^{\mu}$: padrão da rede

ξ_i^{μ}: elemento do vetor



$\vec{\xi}^{1} = (-1, +1, +1)$

O problema pode ser resolvido através do armazenamento de uma lista de parâmetros através do cômputo da distância de Hamming

Zeta

$$\sum_i \left[\xi_i^{\mu} (1 - \zeta_i) + (1 - \xi_i^{\mu}) \zeta_i \right]$$

ζ_i → padrão que se apresenta p/ rede.

ξ_i^{μ} → ζ_i

Essa distância te dará um quanto um neto se diferencia um do outro.

Memória associativa:

é um processo aonde dado $\vec{\xi}^{\mu}$ padrões ele vai te retornar um $\vec{\zeta}^{k}$ padrões de saída

Medida de semelhança:

é a menor distância encontrada.

$\vec{\xi}^{\mu_0} = \vec{\zeta}$ → padrão de entrada

↓

padrão

armazenado

Memória associativa também é chamada de memória não endereçável ou memória endereçável por conteúdo.

Atrator

Sumidouro

Exercício:

atribuição (isto recebe aquilo)

$$(1.1) \qquad \eta_i(t+1) := \Theta\left(\sum_j w_{ij}\, n_j(t) - \mu_i\right)$$

$$\Theta(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

$$\eta_i \in \{0, 1\}$$

$$S_i(t+1) := \mathrm{Sgn}\left(\sum w_{ij}\, S_j(t) - \theta_i\right)$$

$$S_i = 2n_i - 1 \qquad S_i \in \{-1, +1\}$$

Faça a transformação de variáveis entre S_i e n_i.

$$\mathrm{Sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

$$\mathrm{Sgn}\left(\sum_j w_{ij}(2n_j - 1) - \theta_i\right) =$$

$$= \mathrm{Sgn}\left(\sum_j (2n_j w_{ij} - w_{ij}) - \theta_i\right)$$

$$= \mathrm{Sgn}\left(\sum_j 2n_j\right.$$

$$s_i := \mathrm{sgn}\left(\sum_j w_{ij}(2m_j - 1) - \Theta_i\right) =$$

$$= \mathrm{sgn}\left(\sum_j (2m_j w_{ij} - w_{ij}) - \Theta_i\right)$$

$$= \mathrm{sgn}\left(\sum_j 2m_j w_{ij} - \sum_j w_{ij} - \Theta_i\right)$$

$$= \mathrm{sgn}\left(2\sum_j m_j w_{ij} - \sum_j w_{ij} - \Theta_i\right)$$

$s_i = +1$ ze:

$$2\sum_j m_j w_{ij} - \sum_j w_{ij} - \Theta_i \geq 0$$

$$2\sum_j m_j w_{ij} - \sum_j w_{ij} > \Theta_i$$

$$2\sum_j m_j w_{ij} > \Theta_i + \sum_j w_{ij}$$

$$\sum_j w_{ij} m_j > \frac{\Theta_i}{2} + \frac{\sum_j w_{ij}}{2}$$

$$\mu_i = \frac{\Theta_i}{2} + \frac{1}{2}\sum_j w_{ij}$$

$s_i = -1$

$$\sum_j w_{ij} m_j - \sum_j w_{ij} - \Theta_i < 0$$

$$2\sum_j w_{ij} m_j < \sum_j w_{ij} + \Theta_i$$

$$\sum_j w_{ij} m_j < \frac{1}{2}\sum_j w_{ij} + \frac{\Theta_i}{2} \qquad (**)$$

$$\mu_i = \frac{1}{2}\sum_j w_{ij} + \frac{\Theta_i}{2}$$

$$2\mu_i - \sum m_j = \Theta_i$$

$m_j = 0$

$$\sum_j w_{ij} m_j < \mu_i \qquad (*$$

(1.1) $$\eta_i(t+1) := \Theta\left(\sum_j w_{ij}\eta_j(t) - \mu_i\right)$$

$$\Theta(x) = \begin{cases} 1, & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Rio, 11/10/17

S_i: variável de estado
ξ_i: padrão armazenado

Através da mudança de variável

$$S_i := \operatorname{sgn}\Big(\overbrace{\sum_j w_{ij} S_j}^{h_i = \sum_j w_{ij} S_j} - \theta_i\Big)$$

limiar de ativação

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ -1, & \text{se } x < 0 \end{cases}$$

θ passa a ser chamado de Eq. de movi., que representa o estado do neurônio.

Considerando $\theta = 0$

$$S_i := \operatorname{sgn}\Big(\sum_j w_{ij} S_j\Big)$$

tipos de atualização

Atualização síncrona: todos os neurônios se atualizam sincronicamente no tempo.

$$S_{i(t+1)} = \operatorname{sgn}\Big(\sum_j w_{ij} S_{j(t)}\Big)$$

$$S_2(t+1) = \mathrm{sgn} \sum S_j^2(t)$$

$$S_3(t+1) = \mathrm{sgn} \sum S_j(t)$$

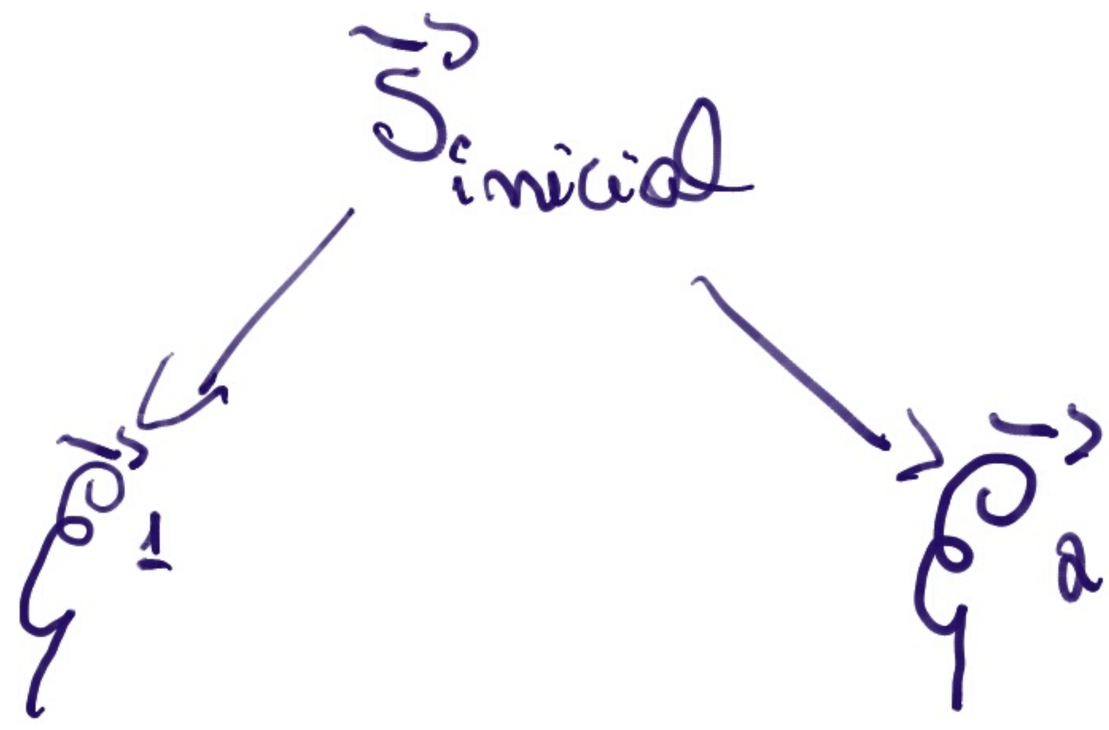
Assíncrona: a atualização dos neurônios se dá de acordo com o estado do vizinho no tempo futuro

$$S_2(t+2) = \mathrm{sgn} \sum S_j(t+1)$$

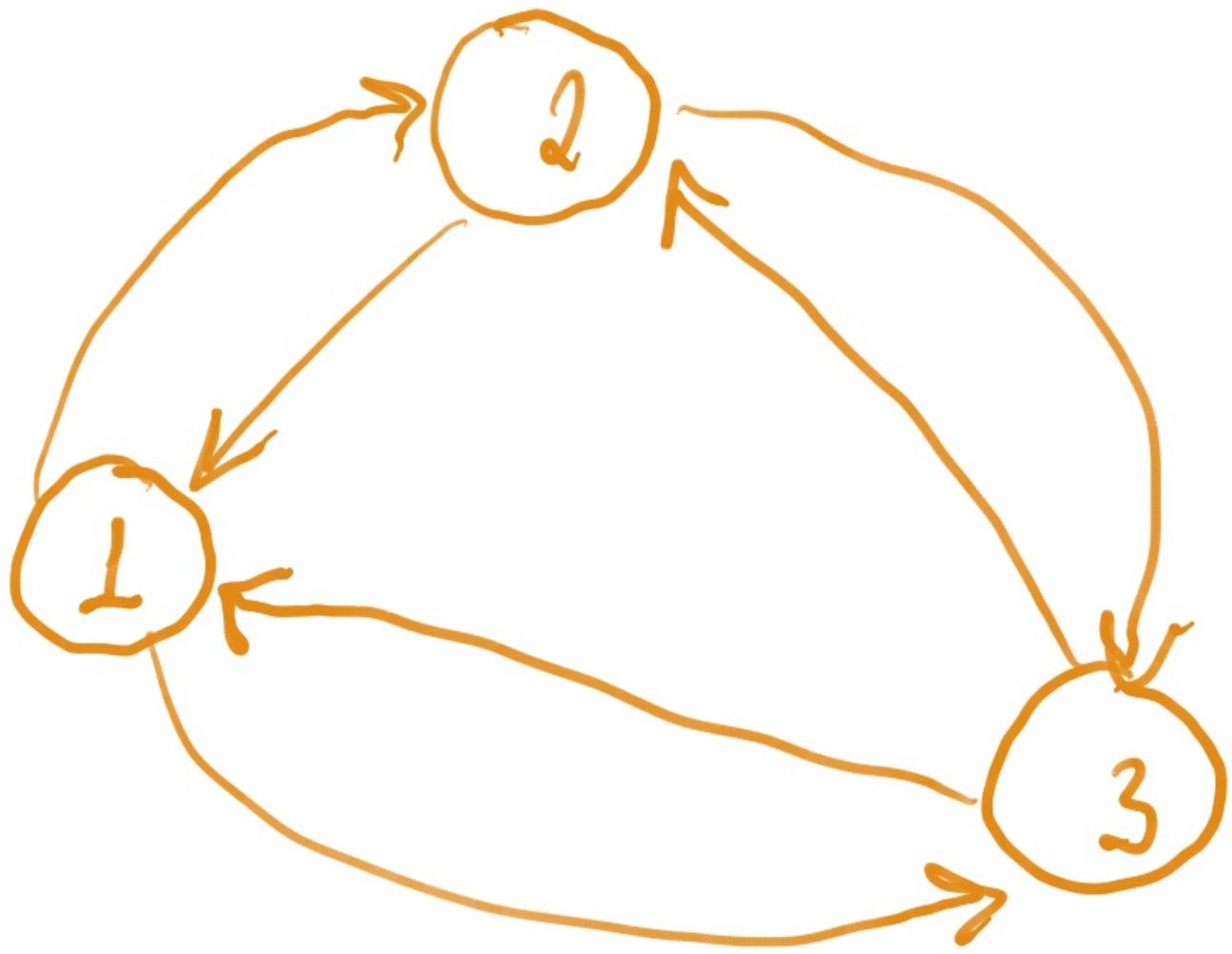
Assíncrona A: um sorteio decide qual o neurônio que irá se atualizar

" B: o neurônio sorteio um valor que irá atrinvéto ou não.

Na atualização assíncrona o modelo de atrator se dá quando os S_i não se atualiza mais. (convergência)



$\vec{\xi}_i = (-1, +1, +1)$ são atividades de maneira aleatória



$$\vec{\xi}^{\,1} = (\xi^1_1, \xi^1_2, \xi^1_3, \dots, \xi^1_N)$$

A dinâmica da escolha dos "ξi" se dará por meio das seus dados.

Por exemplo, ...

$$\vec{\xi}_1 = (-1, +1, +1) \qquad \vec{\xi}_2 = (-1, +1\ -1)$$

Dado $S = (-1, -1, +1)$

Resposta: A sua rede tenderia para $\vec{\xi}^{\,1}$.

Notação correta: $\vec{\xi}^{\,\mu}$ $\qquad 1 \leq \mu \leq p$

$p = 1$ (1 padrão)

O S que convergiu é chamado de estado estável da rede

Para o caso de ξ_i ser um atrator ou ser um estado estável:

$$\mathrm{Sgn}\left(\sum_j w_{ij}\vec{\xi}_j\right) = \vec{\xi}_j \,, \quad \forall i$$

para todo i

(Atrator = estado estável = padrão armazenado).

tomando a eq acima como verdade temos que:

$$w_{ij} \propto \xi_i \xi_j$$

$$w_{ij} = b\,\xi_i \xi_j$$

$$\mathrm{Sgn}\left(\sum_j b\,\xi_i \xi_j \xi_j\right)$$

$$= \mathrm{Sgn}\left(b\,\xi_i \sum_j \xi_j\right)$$

$$= \mathrm{Sgn}\left(b\,\xi_i \sum_j^N 1\right)$$

$$w_{ij} = \mathrm{Sgn}(\xi_i) = \xi_i = w_{ij}$$

Dado o exemplo anterior temos que

$$w_{12} = \xi_1 \xi_2 = -1$$

$w_{21} = -1$

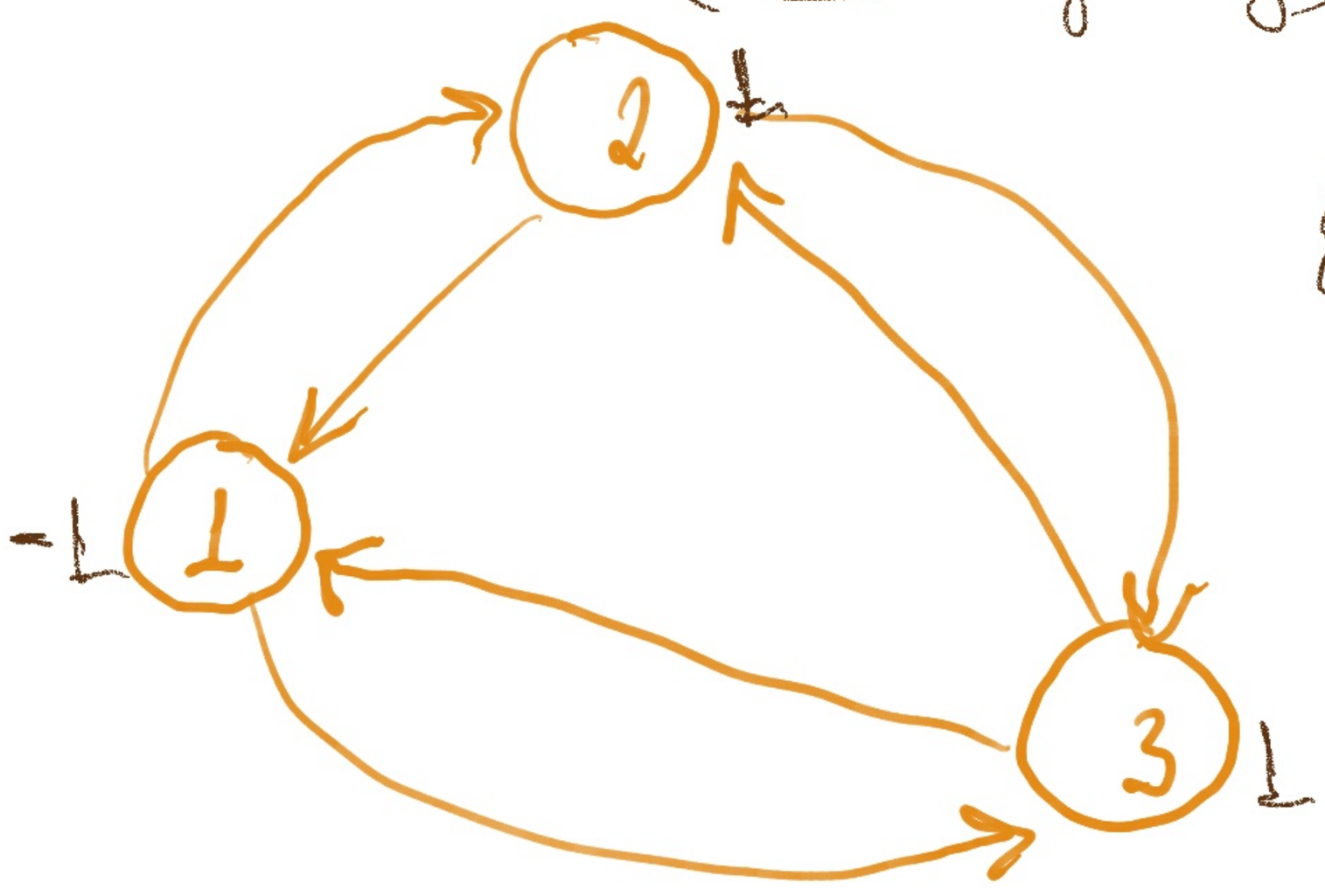
$w_{13} = -1$

Problema: construa uma rede de 3 neurônios usando o $\xi(-1,1,1)$ aplicando a eq 2.7 do livro do Valmir. Mostrar o que acontece ao se aplicar a Eq. 2.4 (S_i) e apresente a atualização na forma síncrona e assíncrona.

2.7

$$W_{ij} = \begin{cases} \frac{1}{n} \xi_i \xi_j & , \text{se } i \neq j \\ 0 & , \text{se } i = j \end{cases}$$

2.4 $\quad S_i = \text{sgn}\left(\sum w_{ij} S_j\right)$



$\xi(-1,1,1)$

$w_{12} = \frac{1}{3}(-1)(1) = -\frac{1}{3}$

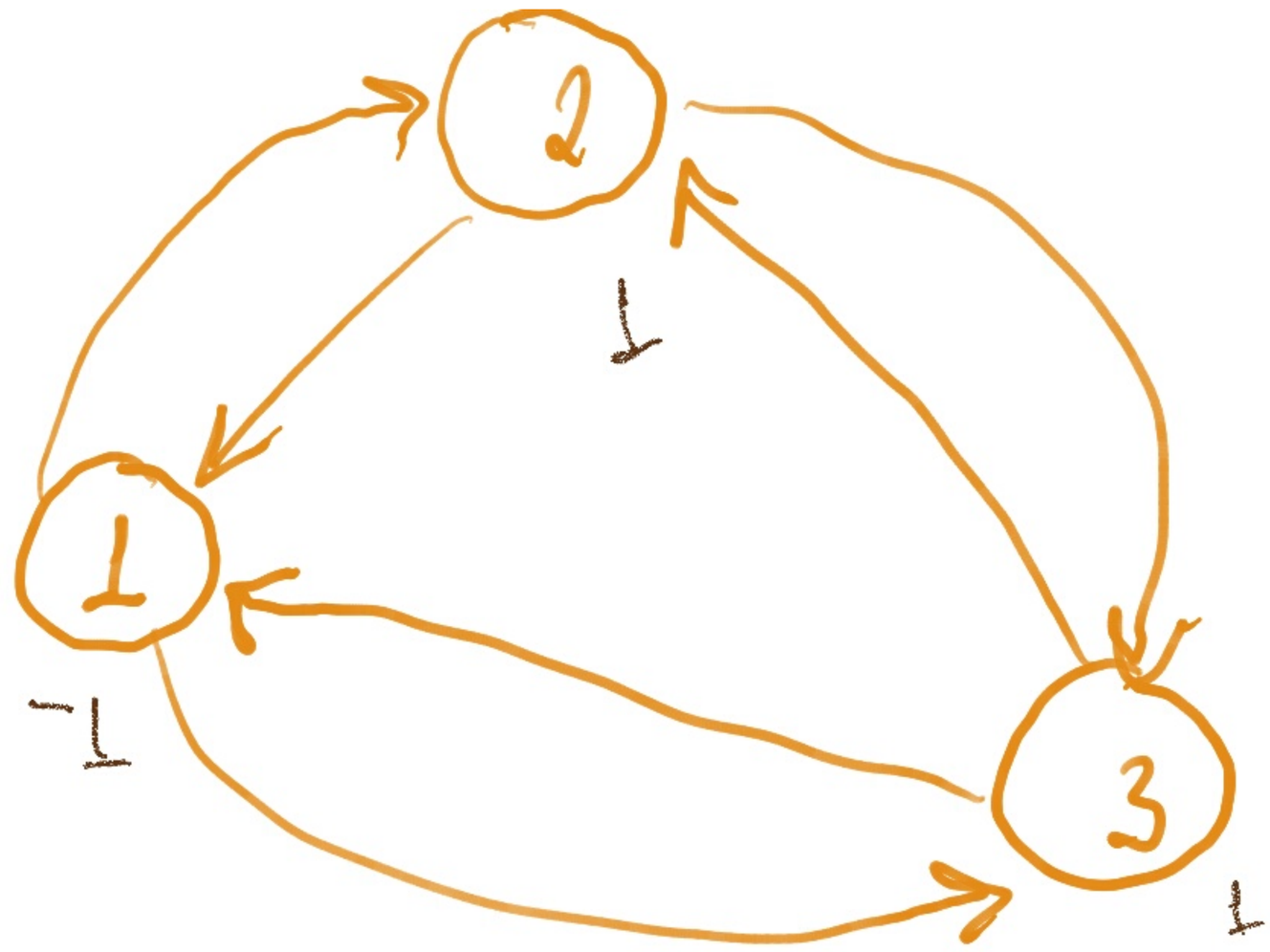
$w_{13} = -\frac{1}{3}$

$w_{21} = -1/3$

$w_{23} = 1/3$

$w_{31} = -1/3$

$w_{32} = 1/3$



$$S_i \leftarrow \operatorname{sgn}\left(\sum w_{ij} S_j\right)$$

$$S_0 = (-1, 1, 1)$$

$$w_{ij} = \frac{1}{n} \xi_i \xi_j$$

Assinorma

$$w_{12} = \frac{1}{3}(-1)(1) = -\frac{1}{3}$$

$$w_{13} = \frac{1}{3}(-1)(1) = -\frac{1}{3}$$

$$S_1 = \operatorname{sgn}(w_{12} S_2 + w_{13} S_3)$$

$$= \operatorname{sgn}\left[(-1/3)(1) + (-1/3)(1)\right]$$

$$S_1 = \operatorname{sgn}[-2/3] = -1$$

$$S_1(-1, 1, 1)$$

para o S

$$w_{31} = 1/3\,(1)(-1) = -1/3$$

$$w_{32} = 1/3\,(1)(1) = 1/3$$

$$S_3 = \operatorname{sgn}\left[w_{31} S_1 + w_{32} S_2 \right]$$

$$\operatorname{sgn}\left[-1/3\,(-1) + 1/3\,(1) \right]$$

$$S_3 = \operatorname{sgn}\left[-2/3 \right] = 1$$

$$S_3 = (-1, 1, 1) \;//$$

p) o S_2)

$$w_{21} = -1/3$$

$$w_{23} = 1/3$$

$$S_2 = \operatorname{sgn}\left[w_{21} S_1 + w_{23} S_3 \right]$$

$$\operatorname{sgn}\left[-1/3\,(-1) + 1/3\,(1) \right]$$

$$\operatorname{sgn}\left[2/3 \right] = 1 \;//$$

$$S_2 = [-1, 1, 1] \;//$$

Sincrono

$$w_{13} = 1/3\,(-1)\,(1) = -1/3$$

$$w_{12} = 1/3\,(-1)\,(1) = -1/3$$

$$w_{21} = 1/3\,(1)\,(-1) = -1/3$$

$$w_{23} = 1/3\,(1)\,(1) = 1/3$$

$$w_{31} = 1/3\,(1)\,(-1) = -1/3$$

$$w_{32} = 1/3\,(1)\,(1) = 1/3$$

$$S_1 = \mathrm{sgn}\left[\overset{-1/3}{w_{12}}\,\overset{1}{S_2} + \overset{-1/3}{w_{13}}\,\overset{1}{S_3}\right] = -1$$

$$S_2 = \mathrm{sgn}\left[\overset{-1/3}{w_{21}}\,\overset{-1}{S_1} + \overset{1/3}{w_{23}}\,\overset{1}{S_3}\right] = 1$$

$$S_3 = \mathrm{sgn}\left[\overset{-1/3}{w_{31}}\,\overset{-1}{S_1} + \overset{1/3}{w_{32}}\,\overset{1}{S_2}\right] = 1$$

$$S = (-1, 1, 1)$$

Obs: atualização síncrona $S_1^{(t+1)} = \mathrm{sgn}(w_{12} S_2^{(t)} + w_{13} S_3^{(t)})$

$$= S_2 = S_3$$

atualização assíncrona $S_2^{(t+2)} = \mathrm{sgn}(w_{21} S_1^{(t+1)} + w_{23} S_3^{(t)})$

$$S_3^{(t+1)} = \mathrm{sgn}(w_{31} S_1^{(t+1)} + w_{32} S_2^{(t+2)})$$

Rio, 18/10/2017

Quando queremos armazenar vários padrões...

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu, & \text{se } i \neq j \\ 0, & \text{se } i = j \end{cases} \quad \forall i, j$$

$$w_{ij} = \xi_i^1 \xi_1^1 + \dots + \xi_i^p \xi_j^p$$

Hebb, detectou que a regra de aprendizado entre neurônios que estão conectados por sinapse e foram estimulados simultaneamente, terão um peso maior.

O modelo matemático não é exatamente igual. Pois quando os estados estão em acordo; exemplo (1, 1) (-1, -1) matematicamente ambas as sinapses terão mais força. E isso não acontece biologicamente, pois quando os neurônios em sinapse não são estimulados eles não terão força na mudança de estado de outros neurônios.

A condição de estabilidade é dada por

$$\text{sgn}(h_i^\nu) = \xi_i^\nu, \quad \forall i$$

$$h_i^\nu = \sum_{j=1}^{N} w_{ij} \xi_j^\nu = \frac{1}{N} \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu \right)$$

Exercício: substituir o w_{ij} na eq. anterior.

Ler cap 2

Rio, 25/10/17

$$\vec{S}(t+1) = \mathrm{sgn}\left(\sum_{j=1}^{N} w_{ij} S_j\right)$$

$$w_{ij} \begin{cases} \xi_i \xi_j & i \neq j \\ 0 & i = j \end{cases}$$

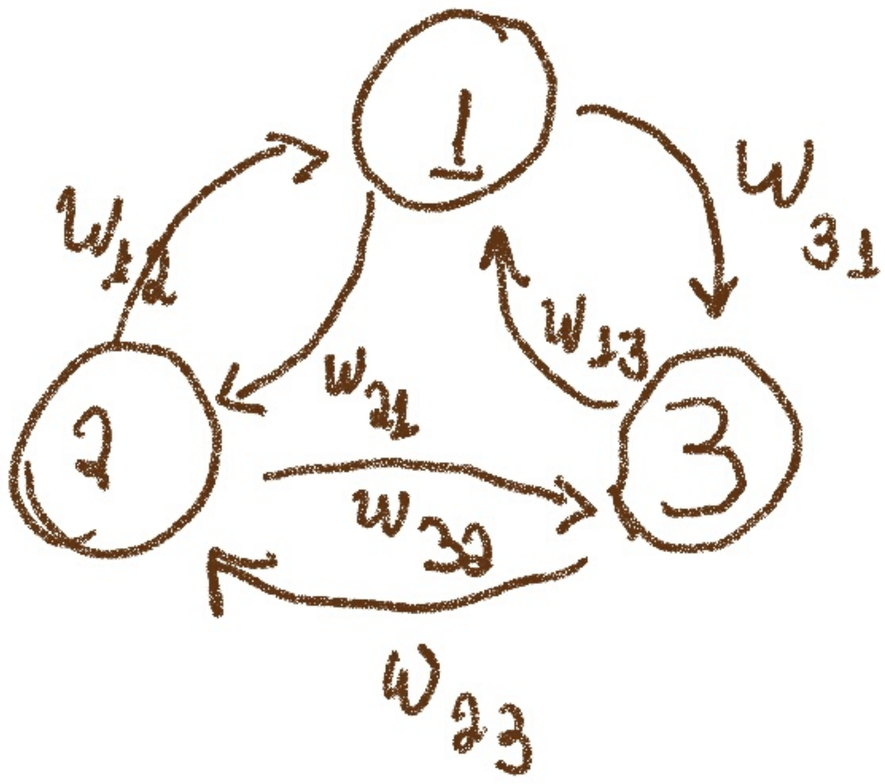
$N = 3$

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \cdot \begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} w_{11} S_1 + w_{12} S_2 + w_{13} S_3 \\ w_{21} S_1 + w_{22} S_2 + w_{23} S_3 \\ w_{31} S_1 + w_{32} S_2 + w_{33} S_3 \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{j=1}^{N} w_{1j} S_j \\ \sum_{j=2}^{N} w_{2j} S_j \\ \sum_{j=3}^{N} w_{3j} S_j \end{pmatrix}$$

→ O registro de dois ou mais padrões na rede se dá através da regra de Hebb.

Rio, 08/11/17



$$w = \begin{pmatrix} 0 & -1/3 & -1/3 \\ -1/3 & 0 & 1/3 \\ -1/3 & 1/3 & 0 \end{pmatrix}$$

$$S_j(t) = \mathrm{sgn}\left[\sum_{j=1}^{N} w_{ij} \; S_j\right]$$

ℓ c

Rio, 8/11/17

Relembrando ...

Múltiplos padrões

O estado espúrio da rede é o padrão - ξ armazenado por construção da rede.

A regra de Hebb diz que quando dois neurônios disparam ao mesmo tempo ocorre um reforço da sinapse.

O modelo de Hopfield é um modelo de memória associativa que utiliza a regra de Hebb para todas as possíveis pares ij.

$$h_i^\nu = \frac{N-1}{N}\xi_i^\nu + \underbrace{\frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{p}\xi_i^\mu\,\xi_j^\mu\,\xi_j^\nu}_{\text{termo cruzado (crosstalk)}}$$

faz o sinal mudar dependendo do tamanho ou valor do termo.

A memória endereçada por conteúdo

Este tipo de memória associa a informação completa, p.ex. uma imagem.

Capacidade de Armazenamento

$$C_i^\nu \equiv -\xi_i^\nu \frac{1}{N} \sum_j \sum_{\mu=\nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu$$

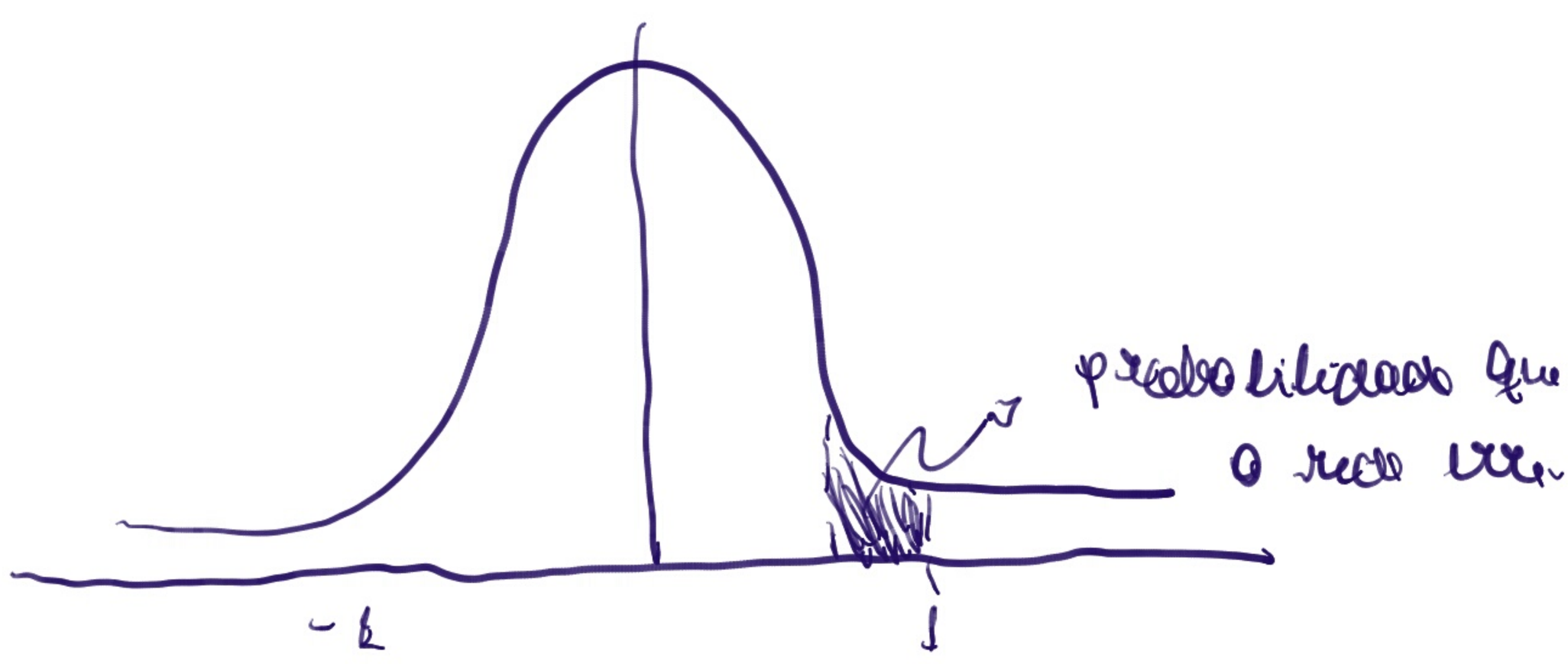
$C_i^\nu < 0$, não altera o sinal de ξ_i

$0 < C_i^\nu < \frac{N-1}{N}$, não altera o sinal "

$C_i^\nu > \frac{N-1}{N} \approx 1$, Problema! Altera o valor de ξ_i

$\lim_{N \to \infty}$

$P_{erro} = \text{prob}(C_i^\nu > 1)$



$$P_{erro} = \frac{1}{\sqrt{2\pi}\sigma} \int_{x}^{\infty} e^{-x^2/2\sigma^2} dx = \frac{1}{2}\left[1 - erf\left(1/\sqrt{2\sigma^2}\right)\right]$$

$$= \frac{1}{2}\left[1 - erf\left(\sqrt{N/2p}\right)\right] \quad (2)$$

onde a função erro é definida como

$$erf(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x \exp(-u^2)\, du \quad (1)$$

exercício: fazer a mudança de variáveis

$$u = \frac{x^2}{2\sigma^2}$$

$$du =$$

e provar 1 e chegar em 2 e comparar c/ a tabela de probabilidade do final do livro.

Rio de Janeiro, 22 de novembro de 2017.

Provar que

$$\log(P_{erro}) \approx -\log 2 - \frac{N}{2p}\log\pi - \frac{1}{2}\log\left(\frac{N}{2p}\right)$$

Considere $= \frac{1}{2}\left[\sqrt{\frac{e^{-(\sqrt{N/2p})^2}}{\sqrt{\pi}\,(\sqrt{N/2p})}}\right] = P_{erro}$

$$\log\left\{\frac{1}{2}\left[\sqrt{\frac{e^{-(\sqrt{N/2p})^2}}{\sqrt{\pi}\,(\sqrt{N/2p})}}\right]\right\}$$

Relembrando

$\log(ab) = \log a \log b$

$\log(a/b) = \log a - \log b$

$\log_b a = \frac{\log_c a}{\log_c b}$

$K \log x = \log(x^K)$

$$P_{error} = \log \frac{1}{2} + \log \left[\frac{e^{-\left(\sqrt{\frac{N}{2P}}\right)^2}}{\sqrt{\pi}\left(\sqrt{\frac{N}{2P}}\right)} \right]$$

$$P_{error} = \left(\log \overset{\nearrow 0}{1} - \log 2\right) + \left(\log e^{-\left(\sqrt{\frac{N}{2P}}\right)} - \log\left(\sqrt{\pi}\left(\sqrt{\frac{N}{2P}}\right)\right)\right)$$

$$= -\log 2 + \left(-\frac{N}{2P}\right)\log e - \log \pi - \log \sqrt{\frac{N}{2P}}$$

$$= -\log 2 - \frac{N}{2P} \cdot \log e - \log \sqrt{\pi} - \log \sqrt{\frac{N}{2P}}$$

$$= -\log 2 - \frac{N}{2P} \cdot \log e - \log \pi^{1/2} - \log \left(\frac{N}{2P}\right)^{1/2}$$

$$= \log 2 - \frac{N}{2P} \cdot \log e - \frac{1}{2}\log \pi - \frac{1}{2}\log\left(\frac{N}{2P}\right)$$

considerando $\boxed{\log = \ln}$ $\boxed{\ln e = 1}$

$$= \ln 2 - \frac{N}{2P} - \frac{1}{2}\ln \pi - \frac{1}{2}\ln\left(\frac{N}{2P}\right) \quad //$$

$$-\ln 2 - \frac{N}{2P} - \frac{1}{2}\ln \pi - \frac{1}{2}\ln\left(\frac{N}{2P}\right) < \log\left(\frac{0{,}01}{N}\right)$$

tomando N um valor muito grande

$$-\cancel{\ln 2}^{\nabla \ll N} - \frac{N}{2P} - \frac{1}{2}\cancel{\ln \pi}^{\nabla \ll N} - \frac{1}{2}\cancel{\ln\left(\frac{N}{2P}\right)}^{\nabla \ll N} < \cancel{\log 0{,}01}^{\ll N}$$

$$-\log N$$

$$-\frac{N}{2p} < -\log N \Rightarrow \boxed{\frac{N}{2p} > \log N}$$

Reescrevendo

$$p < \frac{N}{2\log(Np)}$$

p só pode ser igual ao número de neurônios

$$\frac{N}{2p} > \log(N^2)$$

$$\frac{N}{2p} > 2\log N$$

$$\frac{N}{4p} > \log N \Rightarrow p < \frac{N}{4\log N}$$

$$p_{máx} = \frac{N}{4\log N}$$

Caso os padrões sejam ortogonais

$$\sum_{j} \xi_j^{\mu} \xi_j^{\nu} = 0$$

Neste caso o padrão não será aleatório

O termo Crosstalk

$$C_i^{\nu} = -\xi_i^{\nu} \frac{1}{N} \left(\sum_{\substack{j=1 \\ j \neq i}} \sum_{\substack{\mu=1 \\ \mu \neq \nu}} \xi_i^{\mu} \xi_j^{\mu} \xi_j^{\nu} \right)$$

$$\begin{matrix} i=2 \\ \nu=2 \end{matrix} \quad \xi^{1}_{2}\, \xi^{1}_{1}\, \eta^{2}_{1} + \eta^{1}_{2}\, \eta^{1}_{3}\, \eta^{2}_{3}$$

Ou seja você pode armazenar P padrões desde que os padrões sejam ortogonais.