

Inteligência Artificial

Regressão Neural para Síntese de Texto (RNA)

Aplicação da Lógica Matemática – Sistemas Especialistas

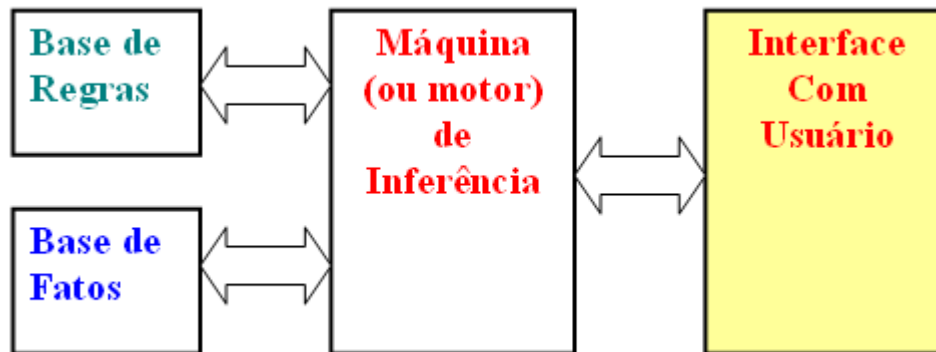
- Um exemplo de aplicação da **Lógica Matemática** encontra-se nos **Sistemas Especialistas** – também conhecidos como **Sistemas Baseados em Conhecimento**.
- Tais sistemas fazem parte da **Inteligência Artificial (IA) simbólica**.

Aplicação da Lógica Matemática – Sistemas Especialistas

- Geralmente **especialistas** de uma área específica do **conhecimento** são **entrevistados** e sua *expertise* é **armazenada** na forma de **proposições lógicas**.
- Em nosso **exemplo simplificado** de um **sistema especialista** o conhecimento abordado será o de algumas **relações de parentesco** (não todas).

Aplicação da Lógica Matemática – Sistemas Especialistas

- A arquitetura de um sistema especialista pode ser representada da seguinte maneira:



Aplicação da Lógica Matemática – Sistemas Especialistas

- A **Base de Regras** armazena proposições lógicas: o **conhecimento específico** é descrito na forma de um **conjunto de regras**.
- As proposições utilizam predicados:
 - **PAI(“Marcus”, “Selina”)**
 - Se lê “**Marcus é PAI de Selina**”.
 - $\text{PAI}(x, y) \rightarrow \text{Filho}(y, z)$
 - Se lê “Se **x é PAI de y** Então **y é FILHO de x**” ou “**x é PAI de y** implica **em y é FILHO de x**”.

Aplicação da Lógica Matemática – Sistemas Especialistas

- Eis um exemplo simplificado da Base de Regras:

Base de Regras	
Nº	Regra
1	$\text{PAI}(x,y) \rightarrow \text{FILHO}(y,x)$
2	$\text{IRMAO}(x,y) \rightarrow \text{IRMAO}(y,x)$
3	$\text{IRMAO}(x,y) \rightarrow [\text{PAI}(w,x) \wedge \text{PAI}(w,y)]$ $\vee [\text{MAE}(z,x) \wedge \text{MAE}(z,y)]$
4	$\text{MAE}(x,y) \rightarrow \text{FILHO}(y,x)$

Aplicação da Lógica Matemática – Sistemas Especialistas

- A **Base de Fatos** armazena **informações** consideradas **verdadeiras**:

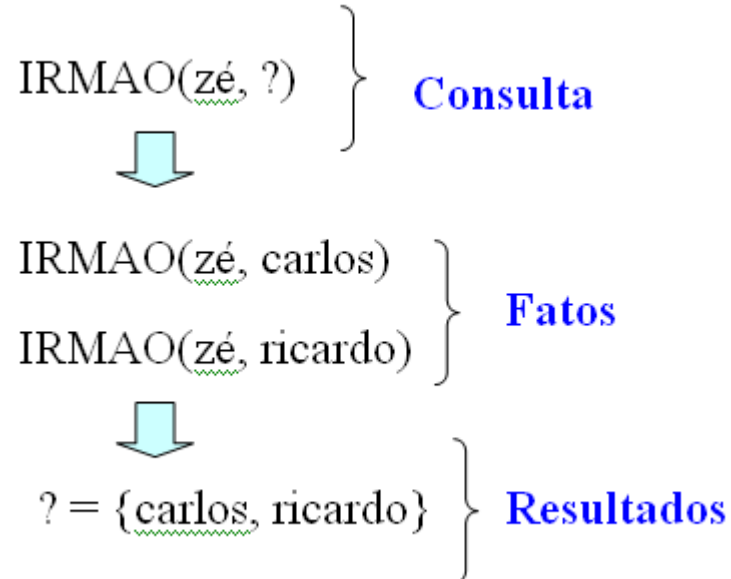
Base de Fatos	
Nº	Fato
1	IRMAO(zé, carlos)
2	IRMAO(marcio, zé)
3	IRMAO(zé, ricardo)
4	MAE(vanusa, zé)
5	MAE(vanusa, socrates)
6	MAE(vanusa, ricardo)
7	PAI(fabiano, zé)
8	PAI(fabiano, ricardo)
9	PAI(antonio, sócrates)

Aplicação da Lógica Matemática – Sistemas Especialistas

- A **Máquina (ou Motor) de Inferência** confronta os dados da **Base de Regras** com os dados da **Base de Fatos** para gerar **novo conhecimento**.
- Suponha que um **usuário humano** deseje saber quem são os **irmãos de zé**.
- Ele faz uma **consulta** com o seguinte **comando: IRMAO(zé, ?)**

Aplicação da Lógica Matemática – Sistemas Especialistas

- O **Motor de Inferência** recebe essa **consulta** e começa procurando por **fatos** que se encaixem diretamente nos **parâmetros** da **consulta**:



Aplicação da Lógica Matemática – Sistemas Especialistas

- Nessa primeira etapa, o motor de inferência identificou “carlos” e “ricardo” como irmãos de “zé”.
- Porém, essa **busca** não teve nada de “**inteligente**” em particular. Fez o que qualquer **Banco de Dados** faria sem precisar de **inteligência artificial**.
- Note que ainda não identificou “márcio” como um dos irmãos de “zé”. Por que?

Aplicação da Lógica Matemática – Sistemas Especialistas

- O **fato número 2** - **IRMAO(márcio, zé)** – não se encaixa perfeitamente na consulta **IRMAO(zé, ?)**.
- Ocorre que “zé” não é o primeiro parâmetro (ou argumento) do predicado dentro do parênteses, mas sim o segundo.
- Significa que nosso sistema especialista jamais descobriria que “márcio” e “zé” são irmãos?
- Ao usar da **lógica matemática** armazenada na **base de regras**, ele descobre isso e outras coisas. Mas como?

Aplicação da Lógica Matemática – Sistemas Especialistas

Regra N° 2:

$\text{IRMAO}(\underline{x}, y) \rightarrow \text{IRMAO}(y, x)$

Fato N°2:

$\text{IRMAO}(\underline{\text{marcio}}, \underline{\text{zé}})$

Consulta:

$\text{IRMAO}(\underline{\text{zé}}, ?)$

Resultado:

$\text{IRMAO}(\underline{\text{zé}}, y) \rightarrow \text{IRMAO}(y, \underline{\text{zé}})$

$y = \underline{\text{marcio}}$

$? = \{\underline{\text{carlos}}, \underline{\text{ricardo}}, \underline{\text{marcio}}\}$

Aplicação da Lógica Matemática – Sistemas Especialistas

- O **Motor de Inferência** percebe uma combinação entre o **Fato Nº 2** e a **Regra Nº 2**.
- Assim infere que “**marcio**” é irmão de “**zé**” também.
- O **Motor de inferência** continua a procurar possíveis combinações entre **regras** e **fatos**, até que todos as combinações existentes sejam esgotadas.
- Assim, por exemplo, encontra outra combinação entre a **Regra Nº 3** e os **Fatos 4, 5, 7 e 9**.

Aplicação da Lógica Matemática – Sistemas Especialistas

Regra 3:

$\text{IRMAO}(x,y) \rightarrow [\text{PAI}(w,x) \wedge \text{PAI}(w,y)] \vee [\text{MAE}(z,x) \wedge \text{MAE}(z,y)]$

Fato 4:

$\text{MAE}(\text{vanusa}, \text{zé})$

Fato 5:

$\text{MAE}(\text{vanusa}, \text{socrates})$

Fato 7:

$\text{PAI}(\text{fabiano}, \text{zé})$

Fato 9:

$\text{PAI}(\text{antonio}, \text{socrates})$

Aplicação da Lógica Matemática – Sistemas Especialistas

- Após inúmeras combinações:

$x = zé$ (conforme a consulta)

$IRMAO(zé, y) \rightarrow [PAI(w, zé) \wedge PAI(w, y)] \vee [MAE(z, zé) \wedge MAE(z, y)]$

$z = vanusa$ (conforme Fato 4)

$IRMAO(zé, y) \rightarrow [PAI(w, zé) \wedge PAI(w, y)] \vee [MAE(vanusa, zé) \wedge MAE(vanusa, y)]$

$w = fabiano$ (conforme Fato 7)

$IRMAO(zé, y) \rightarrow [PAI(fabiano, zé) \wedge PAI(fabiano, y)] \vee [MAE(vanusa, zé) \wedge MAE(vanusa, y)]$

$y = socrates$ (conforme Fato 5)

$IRMAO(zé, socrates) \rightarrow [PAI(fabiano, zé) \wedge PAI(fabiano, socrates)] \vee [MAE(vanusa, zé) \wedge MAE(vanusa, socrates)]$

Aplicação da Lógica Matemática – Sistemas Especialistas

- Agora que o **Motor de Inferência** substituiu todas as **variáveis** na **Regra Nº 3**, a **proposição lógica** deve ser resolvida (seu **valor lógico** deve ser encontrado). Resolvendo apenas o lado direito da proposição:

$[PAI(\text{fabiano}, zé) \wedge PAI(\text{fabiano}, \text{socrates})] \vee$
 $[MAE(\text{vanusa}, zé) \wedge MAE(\text{vanusa}, \text{socrates})]$

$[\text{verdadeiro} \wedge PAI(\text{fabiano}, \text{socrates})] \vee [MAE(\text{vanusa}, zé) \wedge$
 $MAE(\text{vanusa}, \text{socrates})]$

$[\text{verdadeiro} \wedge \text{falsidade}] \vee [MAE(\text{vanusa}, zé) \wedge MAE(\text{vanusa},$
 $\text{socrates})]$

Aplicação da Lógica Matemática – Sistemas Especialistas

- Note que o valor lógico de **PAI(fabiano, socrates)** é falsidade, pois “fabiano” não é pai de “socrates”.
- O pai de “socrates” é “antonio” (**Fato 9**).

$[verdadeiro \wedge falsidade] \vee [verdade \wedge MAE(vanusa, socrates)]$

$[verdadeiro \wedge falsidade] \vee [verdade \wedge verdade]$

$falsidade \vee [verdade \wedge verdade]$

$falsidade \vee verdade$

$verdade$

Aplicação da Lógica Matemática – Sistemas Especialistas

- Assim, o sistema especialista infere também que “sócrates” é irmão de “zé”.
- Assim, o resultado final apresentaria a seguinte lista de irmãos:
 - Carlos,
 - Márcio,
 - Ricardo e
 - Sócrates.

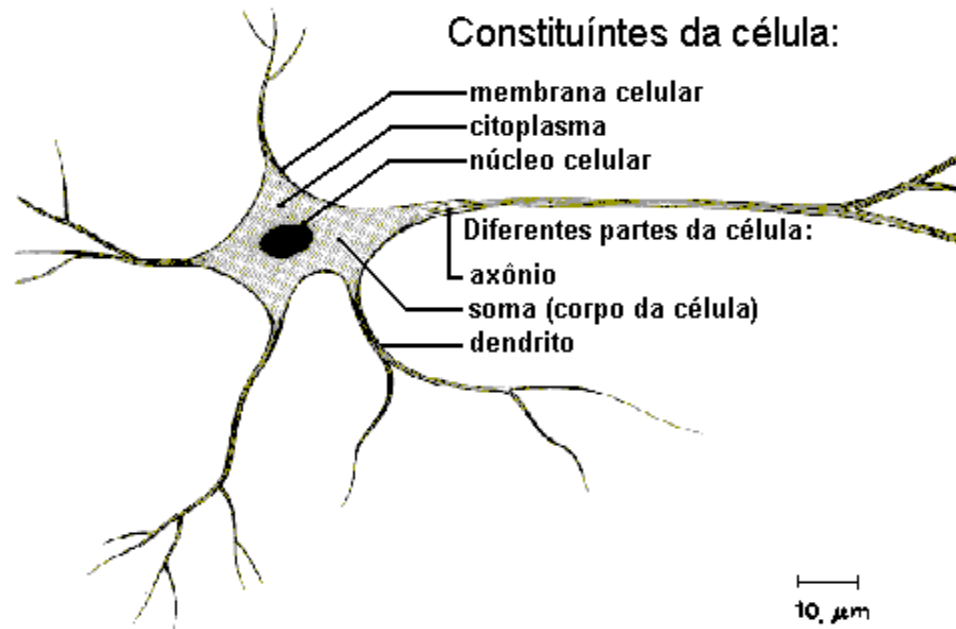
Redes Neurais Artificiais

- A **Lógica de Predicados** é a base sobre a qual se assenta a **Inteligência Artificial (IA) Simbólica** dos **Sistemas Especialistas**.
- Mas a **Lógica** também tem sua parcela de contribuição para a **IA Conexionista** das **Redes Neurais Artificiais (RNA)**.
- Todavia, tal contribuição é mais indireta.

Redes Neurais Artificiais

- **Redes Neurais Artificiais** são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.
- Uma grande **rede neural artificial** pode ter centenas ou milhares de unidades de processamento; já o cérebro de um mamífero pode ter muitos bilhões de neurônios.

Redes Neurais Artificiais

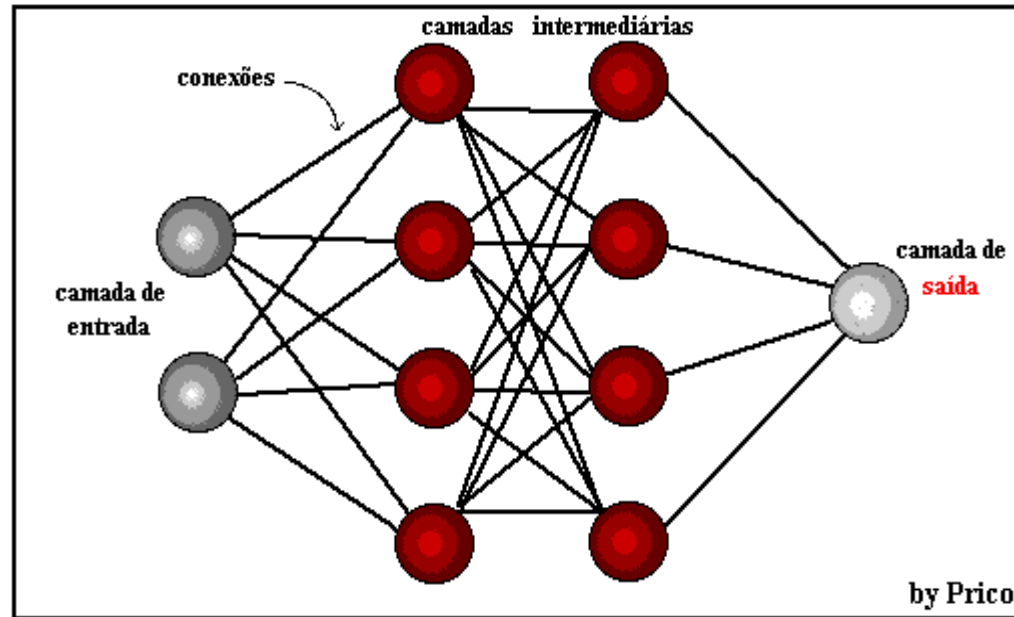


- Os **neurônios** se **comunicam** através de **sinapses**.
- **Sinapse** é a **região** onde **dois** **neurônios** **entram** **em** **contato** e

Redes Neurais Artificiais

- Os **impulsos** recebidos por um **neurônio A**, em um determinado momento, são **processados**, e atingindo um dado limiar de ação, o **neurônio A** dispara, produzindo uma substância neurotransmissora que flui do **corpo celular** para o **axônio**, que pode estar **conectado** a um **dendrito** de um **outro neurônio B**.
- O **neurotransmissor** pode diminuir ou aumentar a polaridade da **membrana**

Redes Neurais Artificiais



Organização em camadas.

- Os sinais provenientes da camada de entrada são propagados para a camada intermediária mais próxima.

Rede Neural Artificial

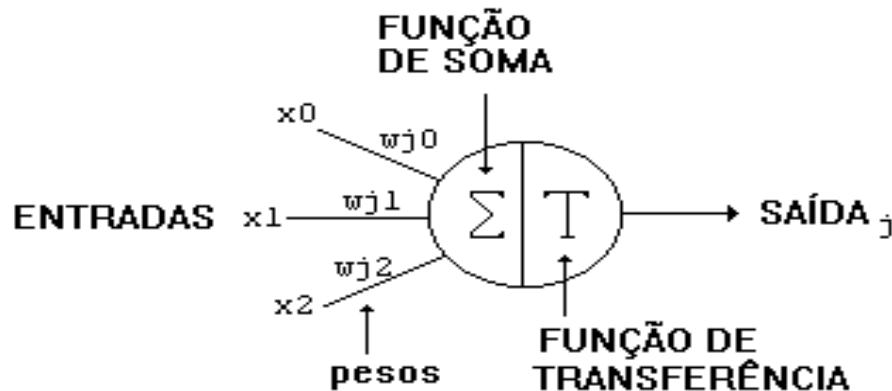
- A propriedade mais importante das **redes neurais** é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho.
- Isso é feito através de um **processo iterativo de ajustes aplicado a seus pesos, o treinamento**.
- O **aprendizado** ocorre quando a rede neural atinge uma **solução generalizada para uma classe de**

RNA Perceptron



- O desenho acima representa uma **RNA simplificada**, formada apenas por um neurônio, com **três valores de entrada** (X_0 , X_1 e X_2), seus **respectivos pesos** (w_{j0} , w_{j1} e w_{j2}).

RNA Perceptron



- O Neurônio apresenta uma função de soma: $\Sigma = (x_0 * w_{j0}) + (x_1 * w_{j1}) + (x_2 * w_{j2})$

RNA Perceptron

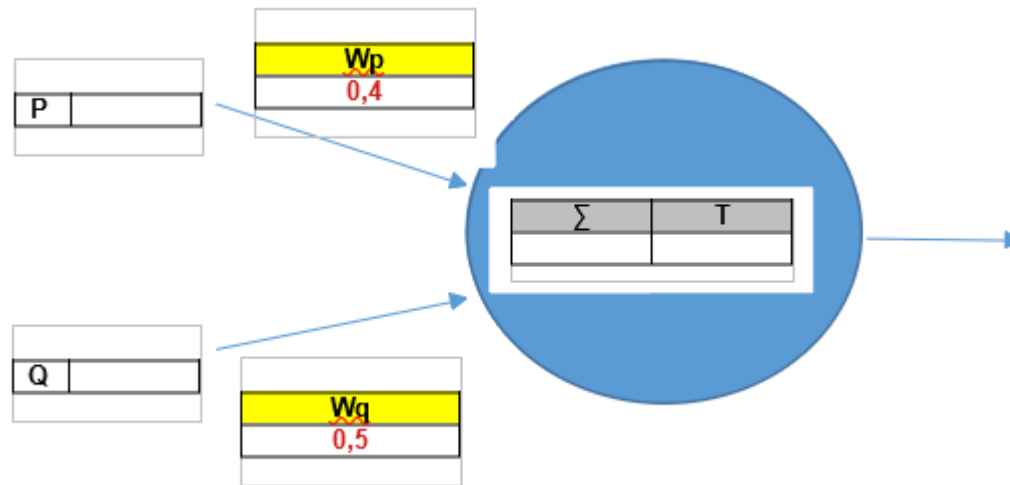
- O resultado da função soma (Σ) é repassado para a função de transferência(T):
- **Se ($\Sigma \geq \text{Patamar}$) Então Saída = 1 Senão Saída = 0 fim-se.**
- Caso saída = 1 então o neurônio foi excitado/ativado. Mas quando saída = 0 dizemos que o neurônio foi inibido/desativado.

RNA Perceptron

- Tomemos como exemplo uma **rede neural artificial Perceptron** para o aprendizado do operador OU (V) da Lógica.
- Essa RNA terá como entrada duas variáveis binárias(P e Q): **1 = Verdadeiro** e **0 = Falso**.

RNA Perceptron

- As duas entradas (P e Q) estão conectadas ao único Neurônio através de duas sinapses.
- Cada sinapse apresenta um peso (W_p e W_q).



RNA Perceptron

Configuração da RNA Perceptron:

- Taxa de Aprendizado (Learning Rate) = $TA = 0,2$.
- Patamar (Sinapse Threshold) = $0,5$
- Valores Iniciais dos pesos:
 - $W_p = 0,4$
 - $W_q = 0,5$

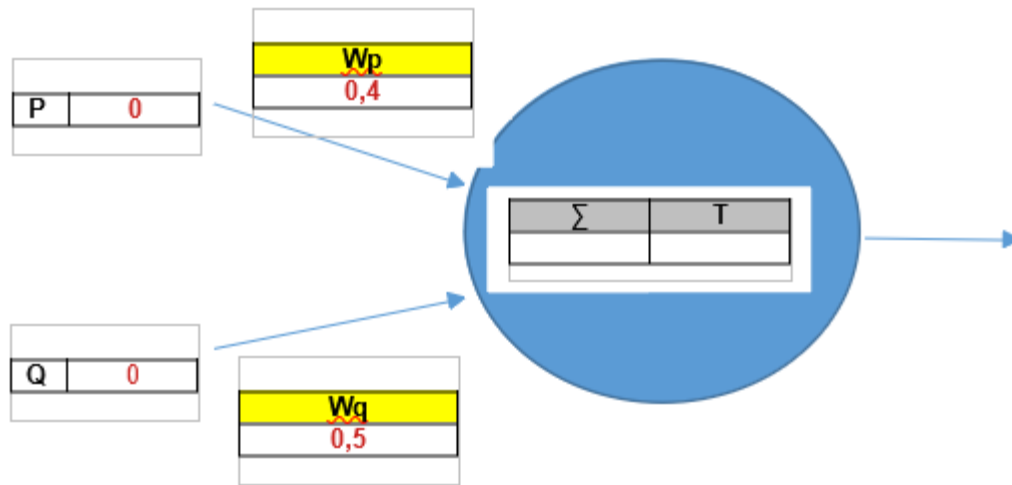
RNA Perceptron

Lições a serem aprendidas pela RNA Perceptron:

Operador OU			
Lição	P	Q	P OU Q
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

RNA Perceptron

Início da **fase de Treinamento** da RNA: a **1ª Lição** é repassada para a RNA.



Treinamento
Supervisionado:

1ª Lição:

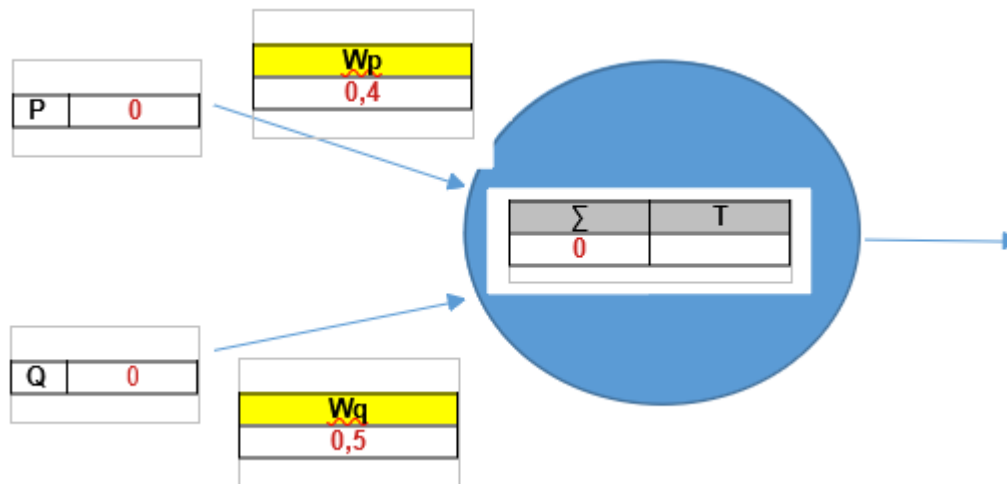
$P = 0$

$Q = 0$

$P \text{ OU } Q = 0$
(saída esperada)

RNA Perceptron

- $\Sigma = (P \times W_p) + (Q \times W_q)$
- $\Sigma = (0 \times 0,4) + (0 \times 0,5) = 0$
- $\Sigma = 0.$



RNA Perceptron

Se ($\Sigma \geq \text{Patamar}$)

Então Saída = 1

Senão Saída = 0

Fim-se

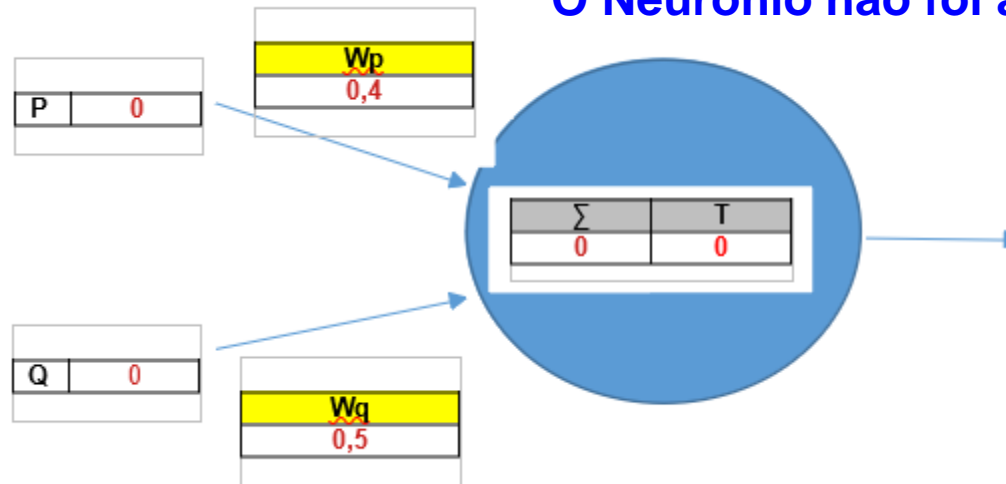
Patamar = 0,5

$\Sigma < \text{Patamar}$

$0 < 0,5$

Saída = 0

O Neurônio não foi ativado.



RNA Perceptron

Erro = SaídaEsperada – Saída

Erro = 0 – 0 = 0

Não houve erro na 1ª Lição.

Portanto não há necessidade de ajuste nos pesos (W_p e W_q).

RNA Perceptron

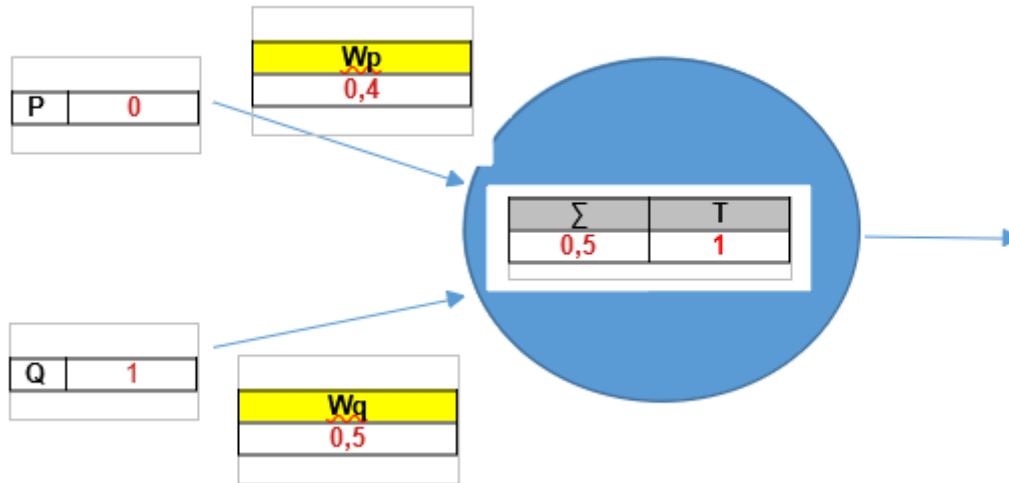
2ª Lição: $P = 0$ $Q = 1$ $P \text{ OU } Q = 1$.

$$\Sigma = (P \times W_p) + (Q \times W_q)$$

$$\Sigma = (0 \times 0,4) + (1 \times 0,5) = 0,5$$

- Como $\Sigma \geq \text{Patamar}$ ($0,5 = 0,5$) a **SaídaEsperada** é 1 (Neurônio é Ativado).

RNA Perceptron



$\text{Erro} = \text{SaídaEsperada} - \text{Saída} = 1 - 1 = 0$

Não ocorreu erro.

Não há necessidade de Ajuste dos Pesos.

RNA Perceptron

3ª Lição: $P = 1$ $Q = 0$ P OU $Q = 1$.

$$\Sigma = (P \times W_p) + (Q \times W_q)$$

$$\Sigma = (1 \times 0,4) + (0 \times 0,5) = 0,4$$

$$\Sigma = 0,4$$

Como $\Sigma < \text{Patamar}$ ($0,4 < 0,5$) a Saída será 0 (o neurônio não foi ativado).

RNA Perceptron

Erro = SaídaEsperada – Saída

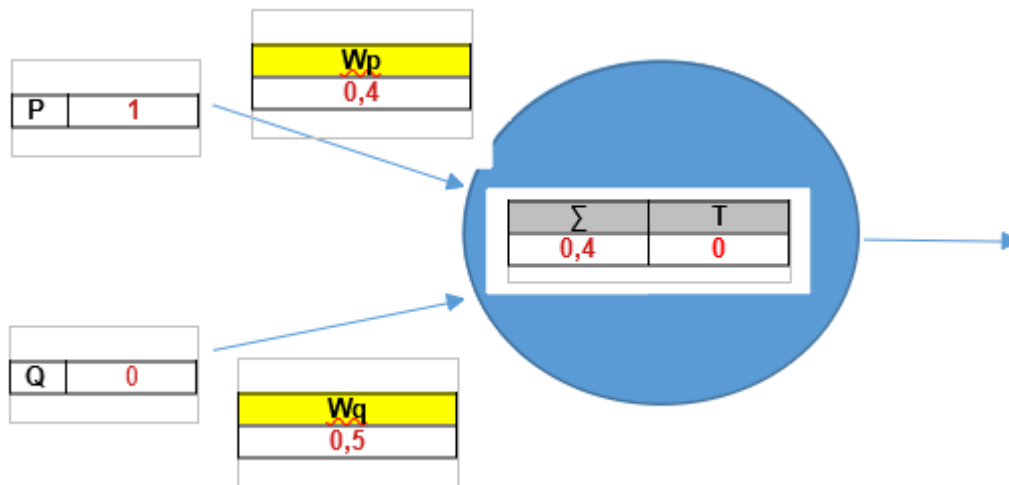
Erro = 1 – 0 = +1

Ocorreu um erro: o gabarito da Lição é 1 mas a RNA produziu 0.

Necessidade de ajuste nos valores dos pesos (W_p e W_q).

RNA Perceptron

3ª Lição:



RNA Perceptron

TA: Taxa de Aprendizagem

$$W_p = W_p + (\text{Erro} \times P \times \text{TA})$$

$$W_p = 0,4 + (1 \times 1 \times 0,2)$$

$$W_p = 0,4 + 0,2 = 0,6$$

$$W_q = W_q + (\text{Erro} \times Q \times \text{TA})$$

$$W_q = 0,5 + (1 \times 0 \times 0,2)$$

$$W_q = 0,5 + 0 = 0,5$$

RNA Perceptron

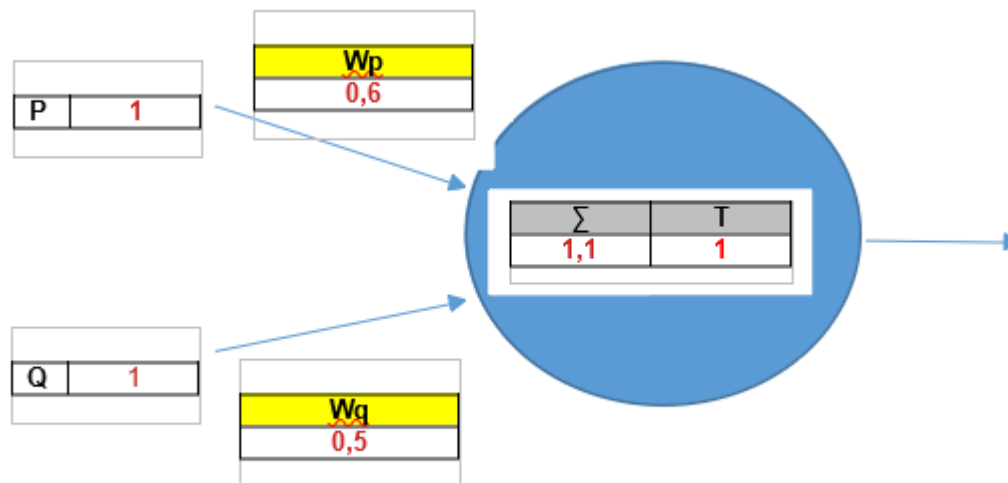
O **ajuste dos pesos** (W_p e W_q)
corresponde ao “**aprendizado**” da
RNA.

Dada a natureza do erro, apenas **W_p**
mudou de 0,4 para **0,6**.

W_q continua com o valor **0,5**.

RNA Perceptron

4ª Lição:



RNA Perceptron

4ª Lição: $P = 1$ $Q = 1$ $P \text{ OU } Q = 1$.

$$\Sigma = (P \times W_p) + (Q \times W_q)$$

$$\Sigma = (1 \times 0,6) + (1 \times 0,5) = 0,6 + 0,5$$

$$\Sigma = 1,1$$

Como $\Sigma > \text{Patamar}$ ($1,1 > 0,5$) a Saída será 1 (o neurônio foi ativado).

RNA Perceptron

$\text{Erro} = \text{SaídaEsperada} - \text{Saída}$

$\text{Erro} = 1 - 1 = 0$

Não ocorreu um ERRO!

Não há necessidade de ajuste dos pesos.

RNA Perceptron

Como a **totalidade das Lições** foram repassadas para a RNA podemos considerar a **1ª seção de treinamento** como encerrada.

Iniciada a **2ª seção de treinamento** repassaremos novamente todas as quatro lições existentes para a RNA.

RNA Perceptron

O **algoritmo** termina sua execução quando em uma **seção de treinamento não ocorrerem mais erros**.

Isso já acontece agora na **segunda seção**.

Note que em nosso exemplo aconteceu apenas um **ajuste de pesos para cima** (**W_p aumentou de 0,4 para 0,6**).

RNA Perceptron

Todavia, **ajustes de peso** podem ocorrer para baixo.

Exemplo:

$\text{Erro} = \text{SaídaEsperada} - \text{Saída}$

$\text{Erro} = 0 - 1 = -1$

$W = W + (\text{Erro} \times Q \times \text{TA})$

$W = 0,7 + (-1 \times 1 \times 0,2) = 0,7 - 0,2 = 0,5$

RNA Perceptron

- **Redes Neurais Artificiais** do tipo **Perceptron** foram propostas originalmente por ***Frank Rosenblatt*** em **1958**.
- Em **1969**, ***Marvin Minsky*** e ***Seymour Papert*** publicaram **Perceptrons** - um texto histórico que alteraria o curso da pesquisa em inteligência artificial por décadas.

RNA Perceptron

- No texto, *Minsky* e *Papert* provaram que um único **perceptron** - um avô das unidades computacionais que compõem as **redes neurais modernas** - **era incapaz de aprender a função exclusiva ou (também conhecida como XOR).**

RNA Perceptron

A **função ou exclusivo**, ou **XOR**, retornará "1" se e somente se uma das duas entradas for "0" e a outra entrada for "1".

Se a entrada for "0, 0" ou "1, 1", o **XOR** produz

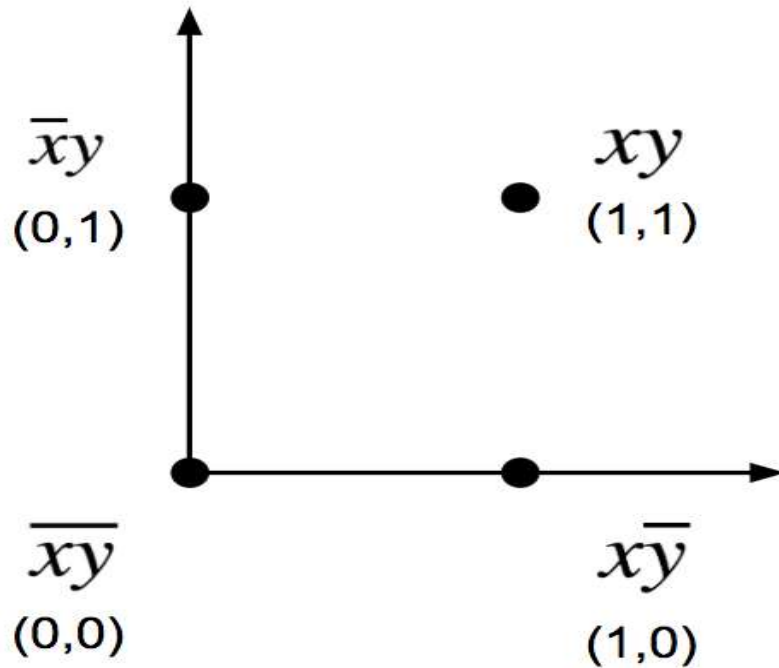
Operador OU EXCLUSIVO			
Lição	P	Q	P OU EXCLUSIVO Q
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Limitações do Perceptron

Questão 1: Por que um **modelo linear** semelhante ao de um **perceptron não pode aprender a função XOR?**

Por definição, um **modelo linear** vai produzir algum tipo de **função linear** (uma **linha 2D**, um **plano 3D** ou uma **superfície dimensional superior que é reta e não tem curvas**).

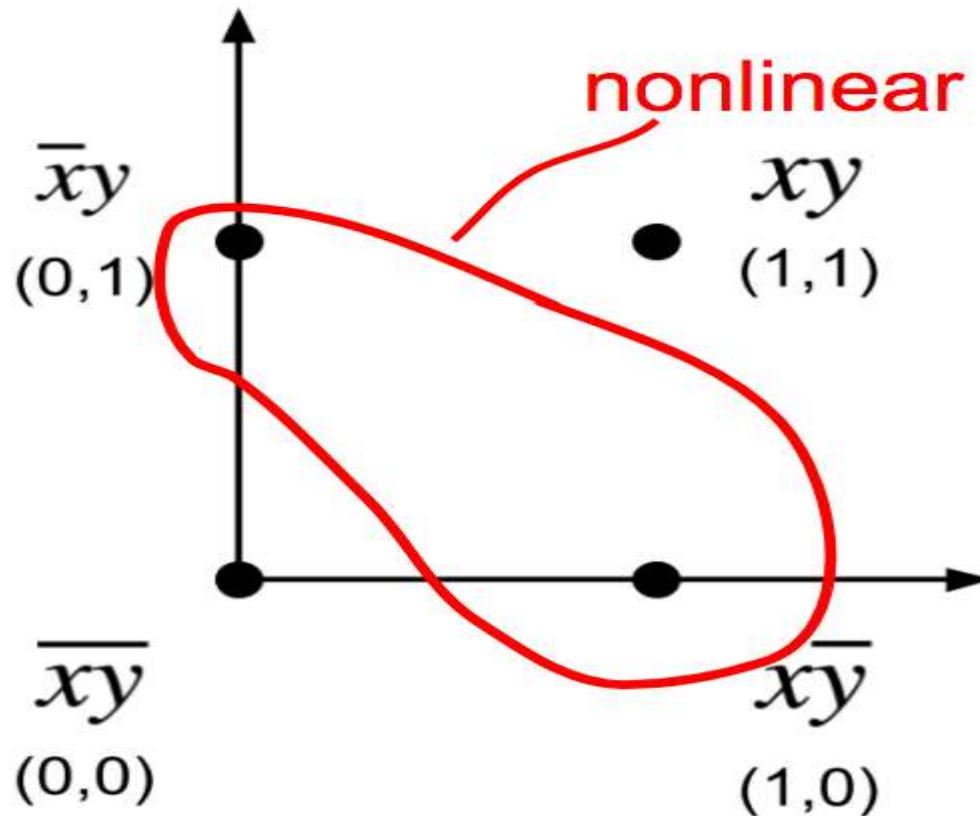
Limitações do Perceptron



No gráfico ao lado, os pontos $\bar{x}y$ e $x\bar{y}$ são as entradas que devem ser sinalizadas pela função XOR como “Verdadeiro” ou “1”.

O desafio é que

Limitações do Perceptron



Limitações da Perceptron

Portanto, se a função XOR separa efetivamente $\bar{x}y$ e $x\bar{y}$, com que se parece a função?

P ou-Exclusivo Q =

(P OU Y) E NÃO(P E Y) =

(P OU Y) E (NÃO P OU NÃO Q)

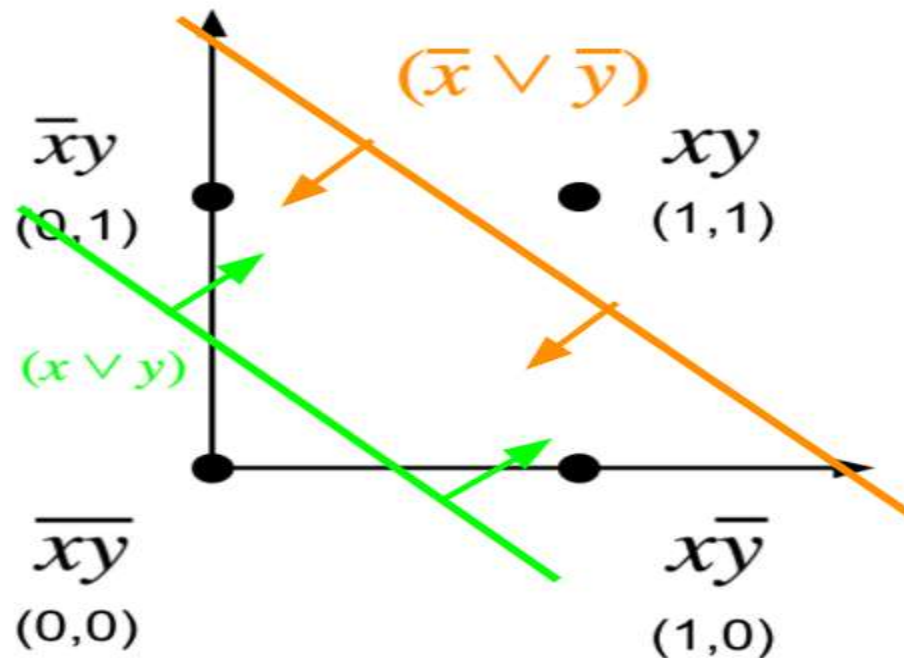
Limitações da Perceptron

Usando a **notação lógica formal**, vemos aqui que **XOR** é na verdade **uma conjunção de duas funções booleanas separadas** (também conhecidas como **predicados**).

O **primeiro predicado**, $(x \vee y)$, retorna Verdadeiro se x ou y ou ambos forem verdadeiros.

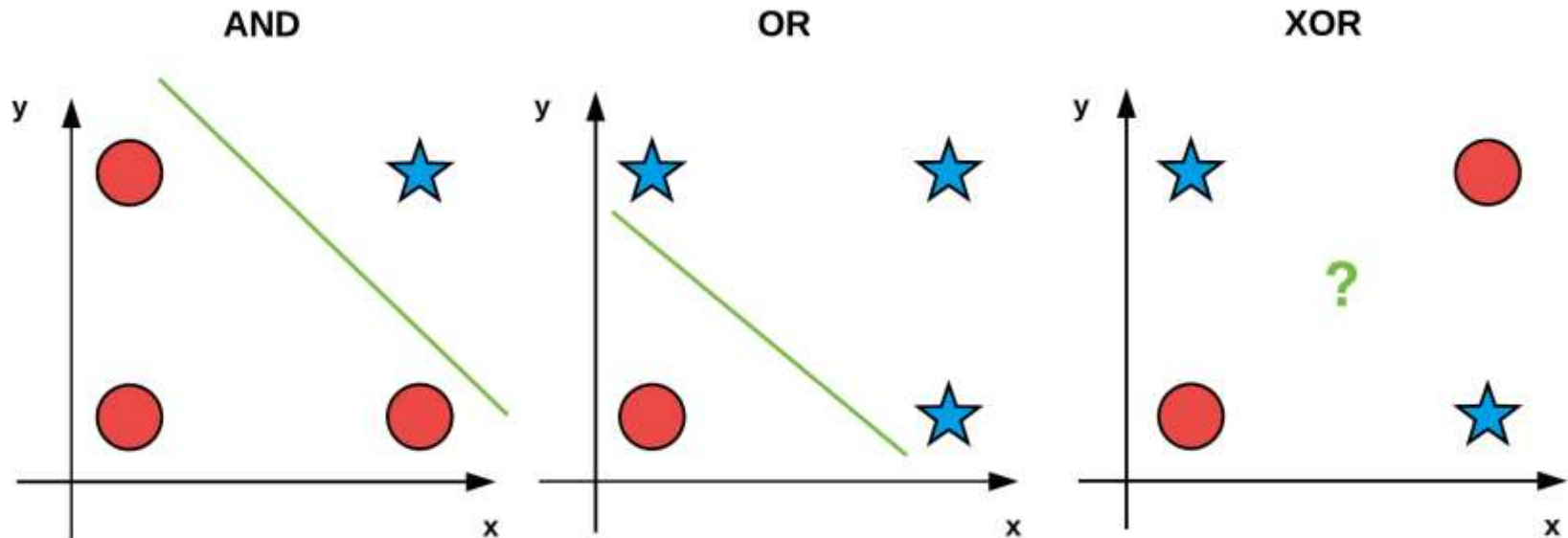
O **segundo predicado**, $(\bar{x} \vee \bar{y})$, retorna Verdadeiro se x não for verdadeiro, y

Limitações da Perceptron



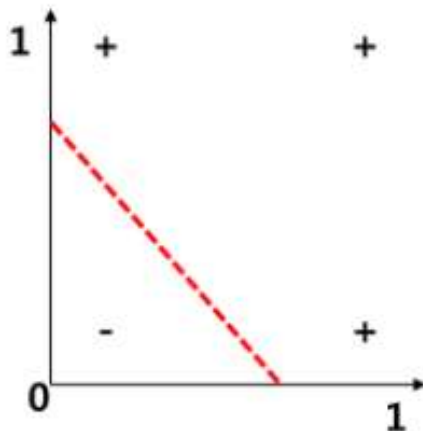
Como mostrado acima, **XOR** e seu inverso, **XNOR**, são únicos entre as 16 funções booleanas porque **não podem ser capturadas com uma única linha.**

Limitações da Perceptron



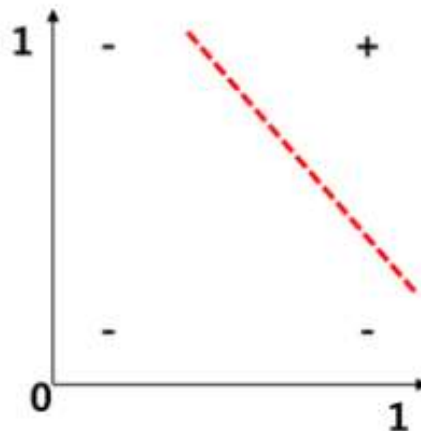
Limitações da Perceptron

OR



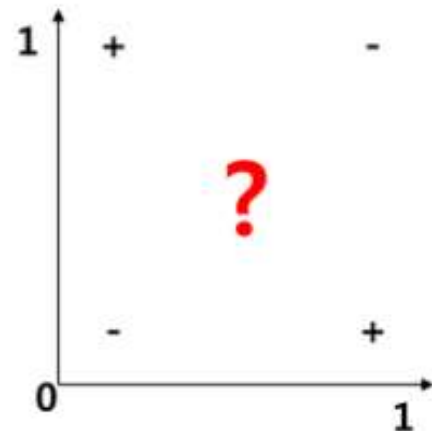
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

AND



x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

XOR



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Redes Neurais Sem Peso

Como o próprio nome deixa antever, as Redes Neurais Sem Peso, não utilizam pesos em suas sinapses.

Elas também trabalham segundo um esquema de endereçamento em que os neurônios de uma camada anterior formam um endereço de um padrão nos neurônios das camadas seguintes.

Redes Neurais Sem Peso

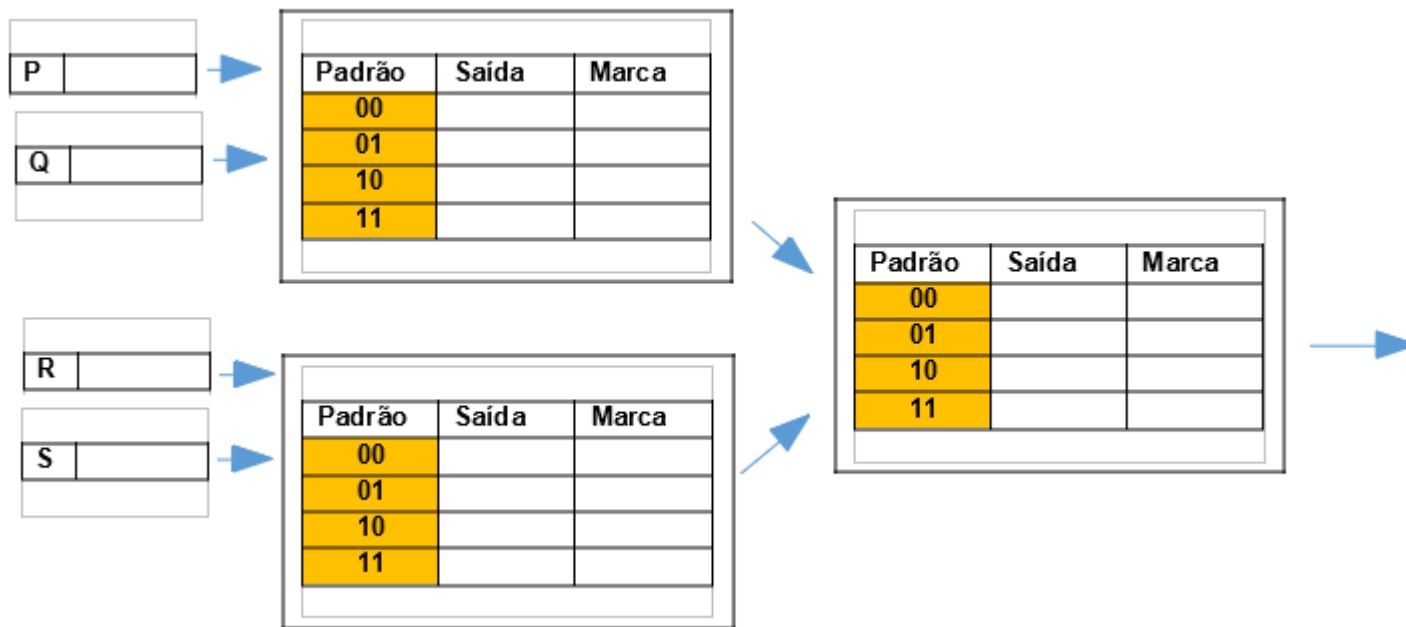
Explicaremos por meio de um exemplo.

Considere uma **RNA** que deve **aprender** a seguinte **proposição composta**:

$((P \wedge Q) \vee (R \vee S))$

Rede Neural Sem Peso

Essa é a arquitetura de nossa RNA:



Rede Neural Sem Peso

Lições							
Lição	P	Q	R	S	$P \wedge Q$	$R \vee S$	$(P \wedge Q) \vee (R \vee S)$
1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	1
3	0	0	1	0	0	1	1
4	0	0	1	1	0	1	1
5	0	1	0	0	0	0	0
6	0	1	0	1	0	1	1
7	0	1	1	0	0	1	1
8	0	1	1	1	0	1	1
9	1	0	0	0	0	0	0
10	1	0	0	1	0	1	1
11	1	0	1	0	0	1	1
12	1	0	1	1	0	1	1
13	1	1	0	0	1	0	1
14	1	1	0	1	1	1	0
15	1	1	1	0	1	1	0
16	1	1	1	1	1	1	0

Rede Neural Sem Peso

1ª Lição:

Lições

Lição	P	Q	R	S	$P \wedge Q$	$R \vee S$	$(P \wedge Q) \vee (R \vee S)$
1	0	0	0	0	0	0	0

P	0
Q	0

Padrão	Saída	Marca
00		
01		
10		
11		

R	0
S	0

Padrão	Saída	Marca
00		
01		
10		
11		

Padrão	Saída	Marca
00		
01		
10		
11		

Rede Neural Sem Peso

Na 1ª Lição temos:

$$P = 0$$

$$Q = 0$$

$$R = 0$$

$$S = 0$$

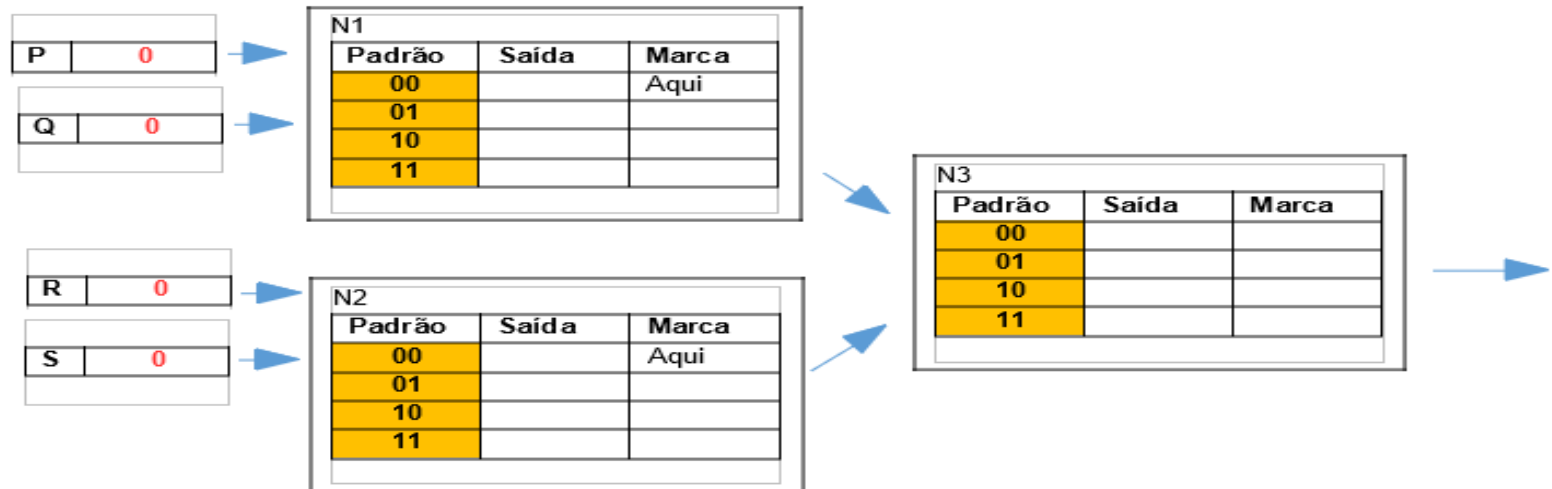
$$\text{Assim: } (P \wedge Q) = (0 \wedge 0) = 0$$

$$(R \vee S) = (0 \vee 0) = 0$$

Rede Neural Sem Peso

Finalmente: $(0 \underline{\vee} 0) = 0$

Portanto, a **Saída Esperada** é **0** em nosso **treinamento supervisionado**.



Rede Neural Sem Peso

Observe que a coluna “Marca” nos neurônios da 1ª camada (N1 e N2) sofreu uma atualização: receberam o valor “Aqui” na linha correspondente ao padrão “00”.

Por que?

As entradas $P (=0)$ e $Q (=0)$ apontam para o Neurônio N1 e formam o padrão $(PQ) = (00)$.

Rede Neural Sem Peso

As entradas $R(=0)$ e $S(=0)$ apontam para o Neurônio N2 e formam o padrão $(RS) = (00)$.

Por essa razão a coluna Marca da linha correspondente ao padrão 00 foi atualizada com “Aqui”.

Essas são as linhas ativadas nos neurônios N1 e N2.

Rede Neural Sem Peso

O conteúdo das colunas Saída das linhas marcadas em N1 e N2 é que vão apontar para o endereço do padrão em N3 (Na camada seguinte).



Rede Neural Sem Peso

Contudo, a Saída de N1 indica “?” (endereço desconhecido).

O mesmo é válido para a Saída de N2.

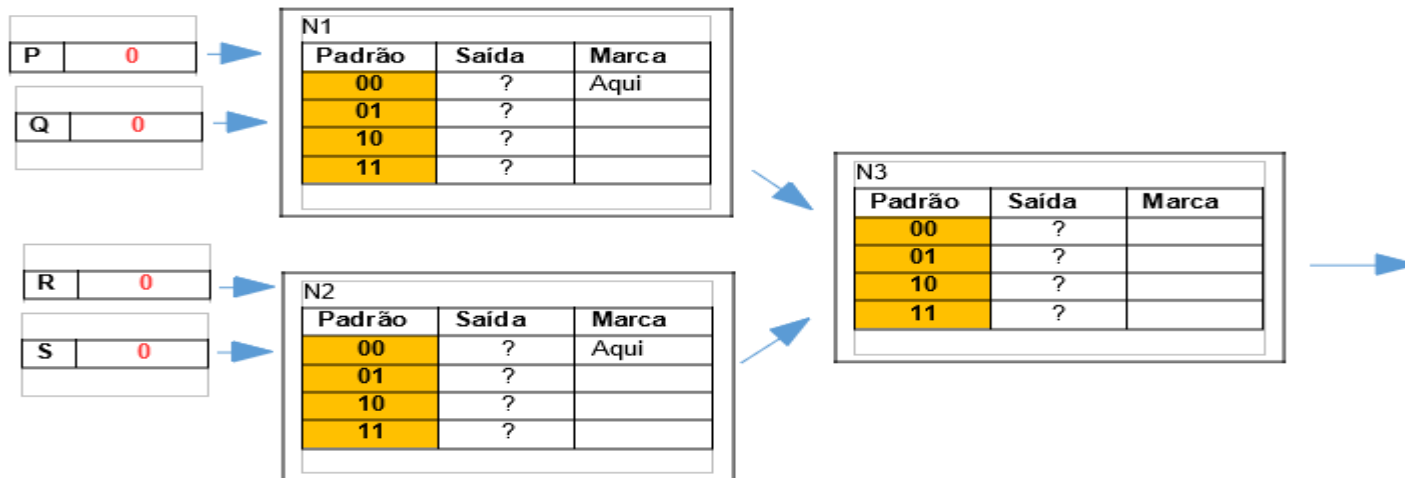
Acontece que a combinação das saídas (N1 N2) = (??) deve apontar para um padrão válido em N3 (00, 01, 10 ou 11).

(??) significa que qualquer padrão é possível em N3.

Rede Neural Sem Peso

N3 já é o Neurônio da camada de saída da RNA sem peso.

Por essa razão devemos verificar se já existe algum padrão em N3 com Saída igual à Saída Esperada da Lição (= 0).



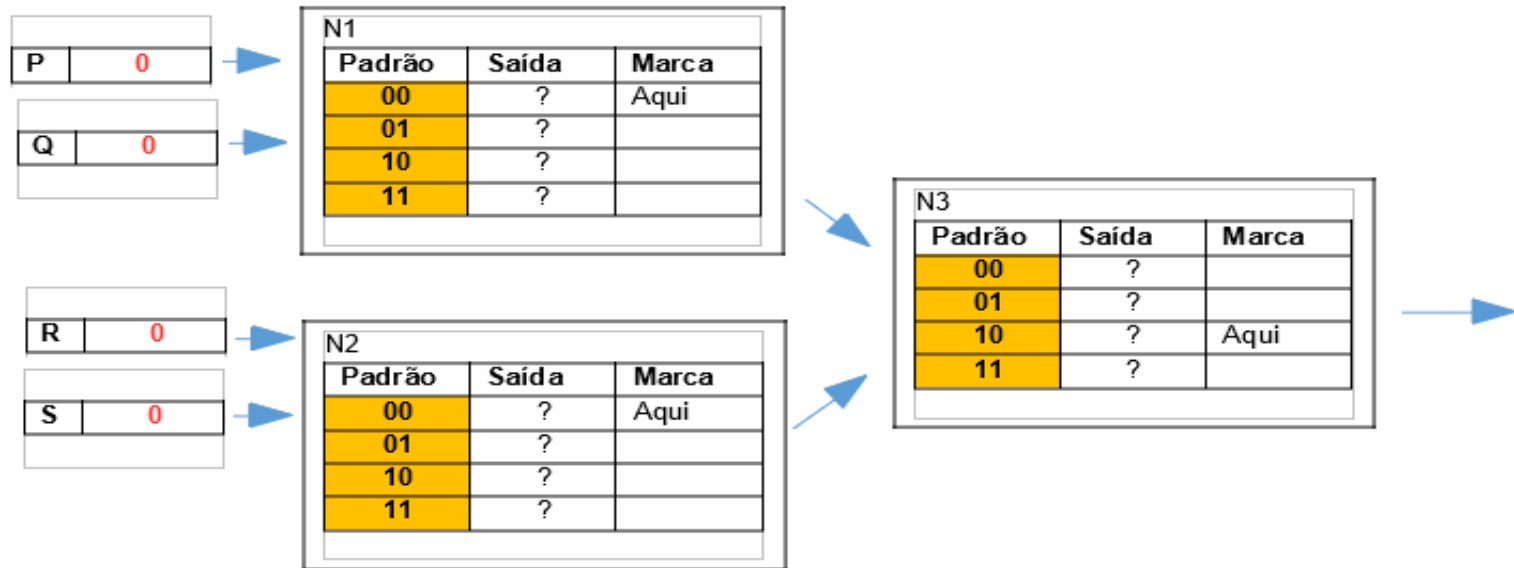
Rede Neural Sem Peso

Todas as Saídas em N3 têm valor “?”
(desconhecido).

Portanto, uma linha será escolhida
aleatoriamente em N3.

Digamos que foi a linha relativa ao padrão
(10).

Rede Neural Sem Peso



Por essa razão, a linha selecionada aleatoriamente em N3 têm sua marca atualizada para “Aqui”.

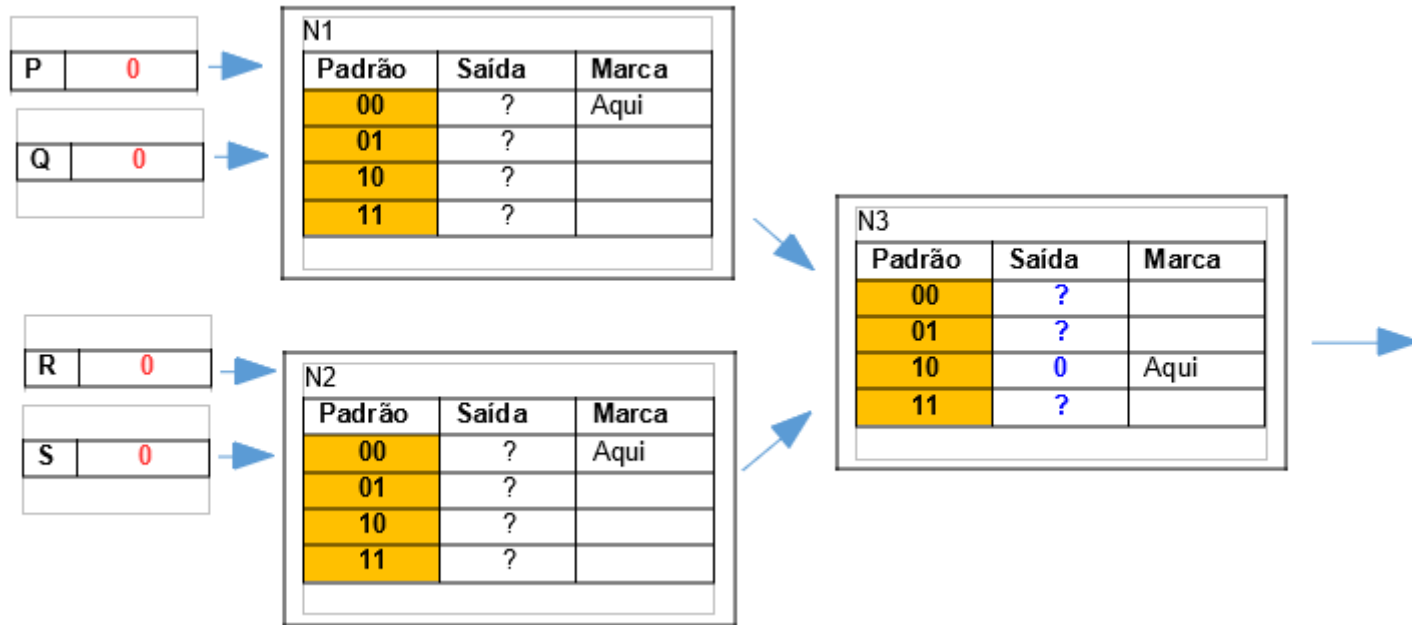
Rede Neural Sem Peso

O que significa tudo isso?

Que nossa rede ainda não aprendeu como lidar com esse conjunto de entradas $(P, Q, R, S) = (0, 0, 0, 0)$.

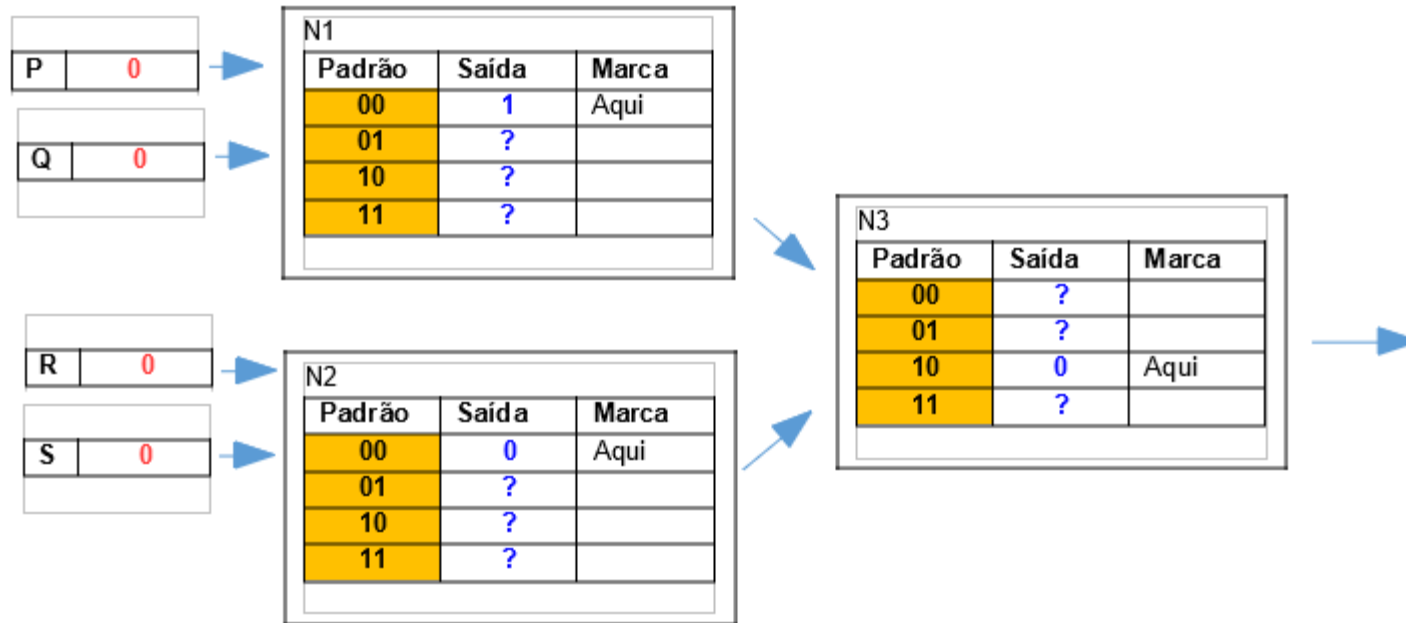
Como o treinamento é supervisionado a Saída Esperada ($= 0$) é informada para a RNA.

Rede Neural Sem Peso



Agora ficou definido que o padrão “10” em N3 corresponde à saída “0” (= Falso).

Rede Neural Sem Peso



Em seguida atualizamos a Saída do padrão marcado “00” de N1 com o valor “1”.

E a Saída do padrão marcado “00” de N2 com “0”

Rede Neural Sem Peso

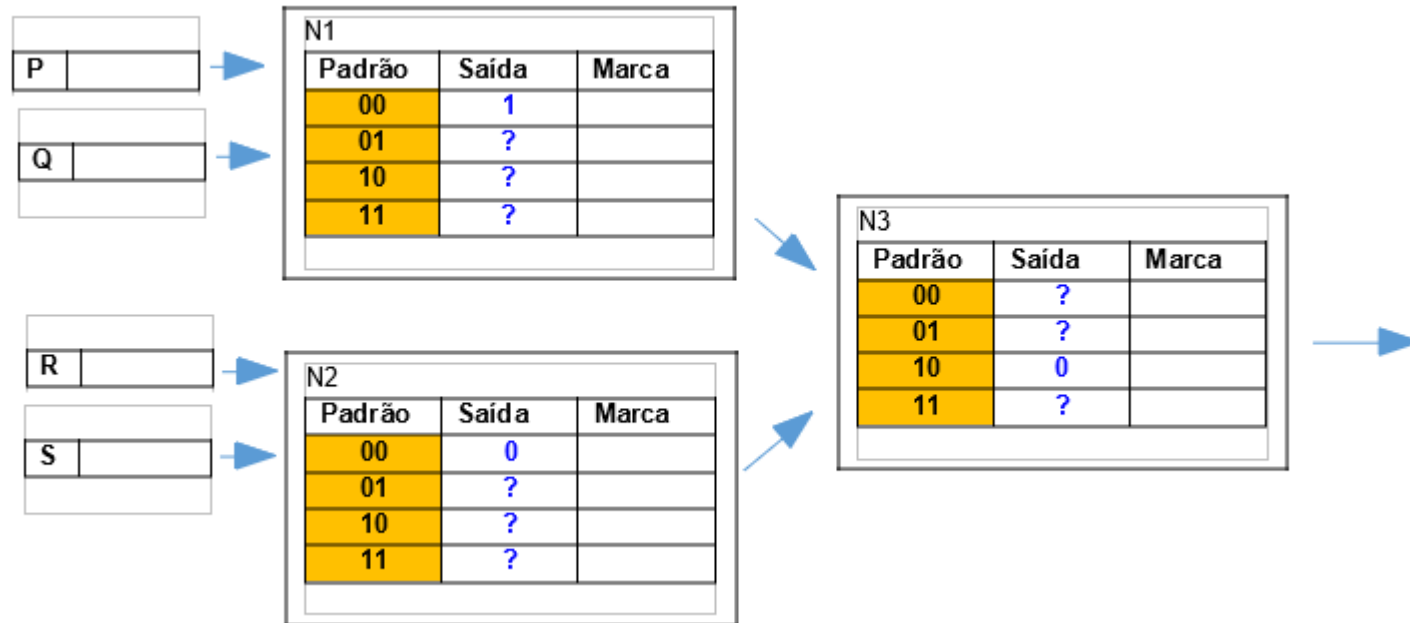
Por que?

Para que as Saídas de N1 e N2 (nessa ordem) formem o endereço do padrão em N3.

O “1” de N1 é a dezena do padrão em N3.

O “0” de N2 é a unidade do padrão em N3.

Rede Neural Sem Peso



As **Marcas** e as **Entradas** da RNA são “**limpas**” antes de passarmos para a 2ª lição.

Rede Neural Sem Peso

2ª Lição:

Lições							
Lição	P	Q	R	S	$P \wedge Q$	$R \vee S$	$(P \wedge Q) \vee (R \vee S)$
1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	1

P	0
Q	0

N1		
Padrão	Saída	Marca
00	1	Aqui
01	?	
10	?	
11	?	

R	0
S	1

N2		
Padrão	Saída	Marca
00	0	
01	?	Aqui
10	?	
11	?	

N3		
Padrão	Saída	Marca
00	?	
01	?	
10	0	
11	?	

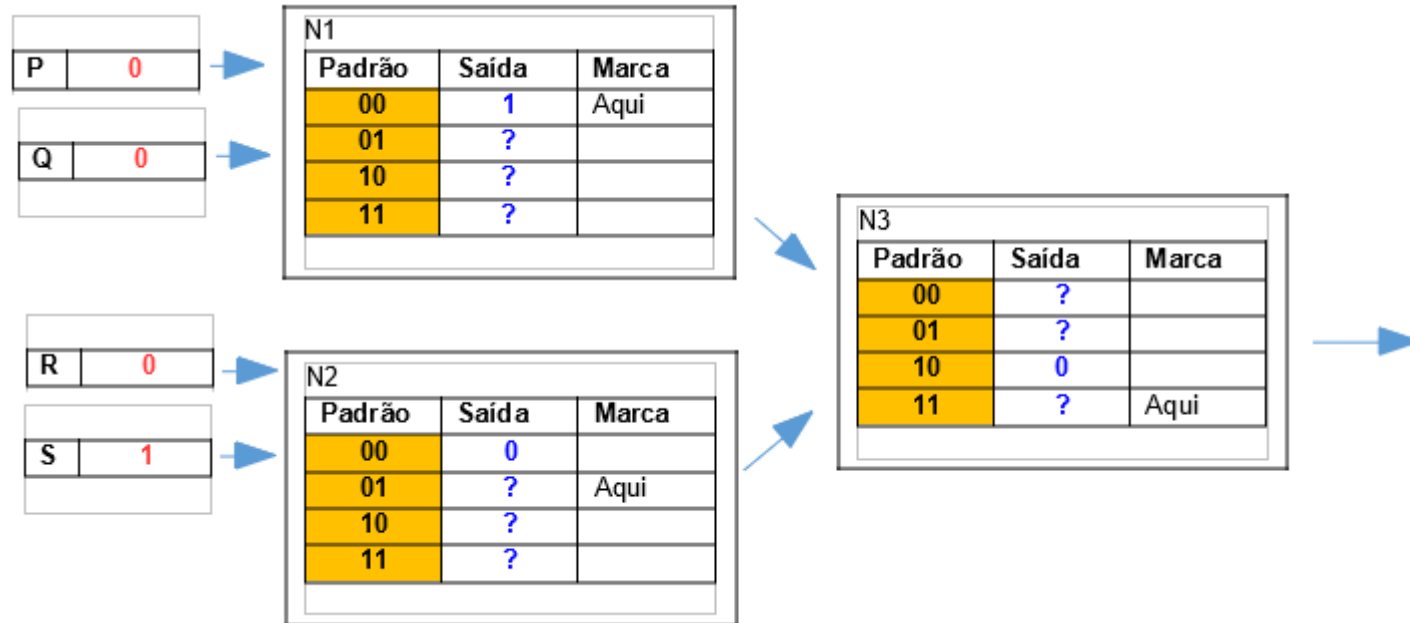
Rede Neural Sem Peso

N1 e **N2** apontam para o endereço de padrão “**1?**”: forçosamente ele deve começar com “1”, mas pode terminar com “0” ou “1”.

E a Saída Esperada (=1) é ainda inédita em N3.

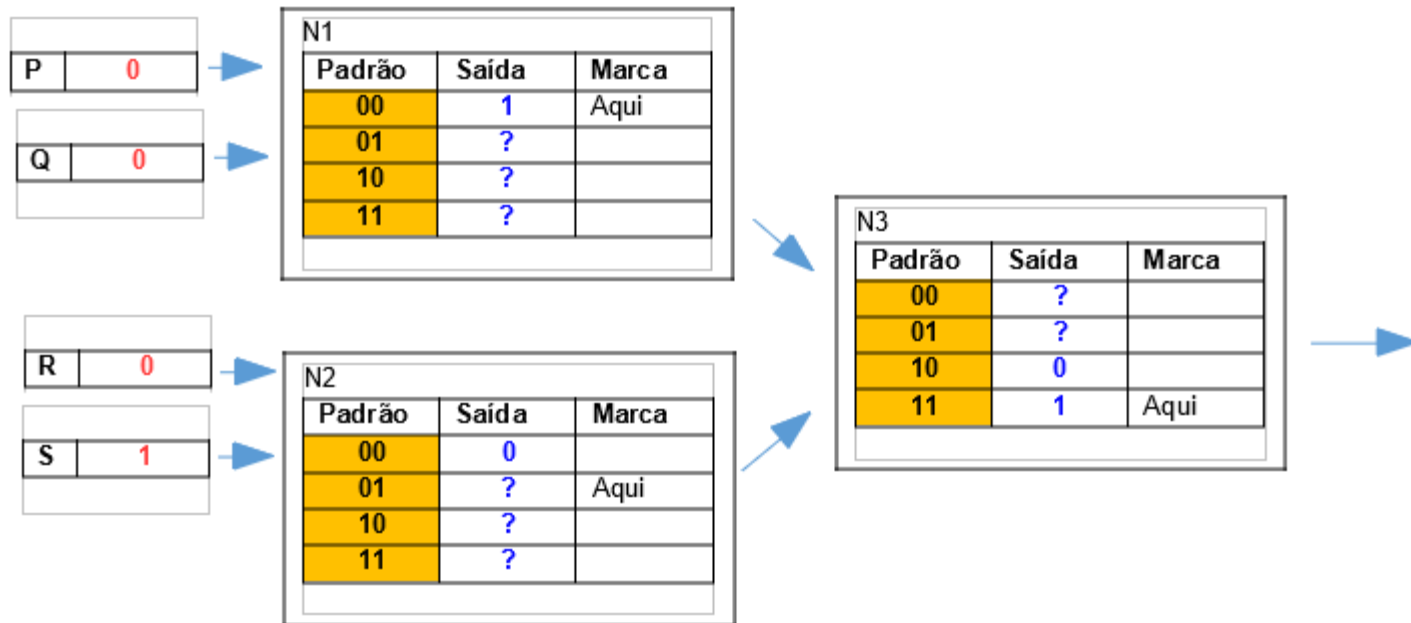
Significa que devemos escolher um padrão aleatoriamente em N3 para esse valor.

Rede Neural Sem Peso



O padrão de N3 deve começar com “1”, mas não pode ser “10” (que já está sendo utilizado): sobrou “11”.

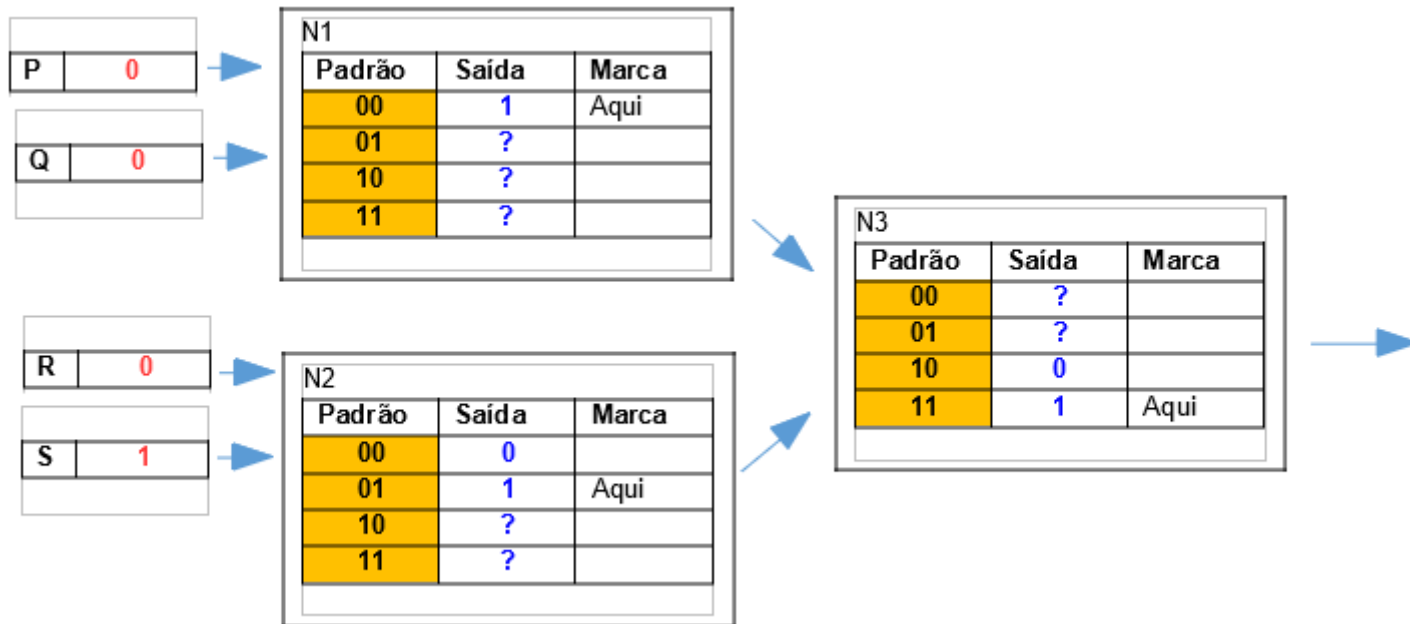
Rede Neural Sem Peso



N1 e N2 devem apontar para N3(padrão “11”). N1 já fornece a dezena “1”, mas N2 não aponta ninguém na unidade “?”.

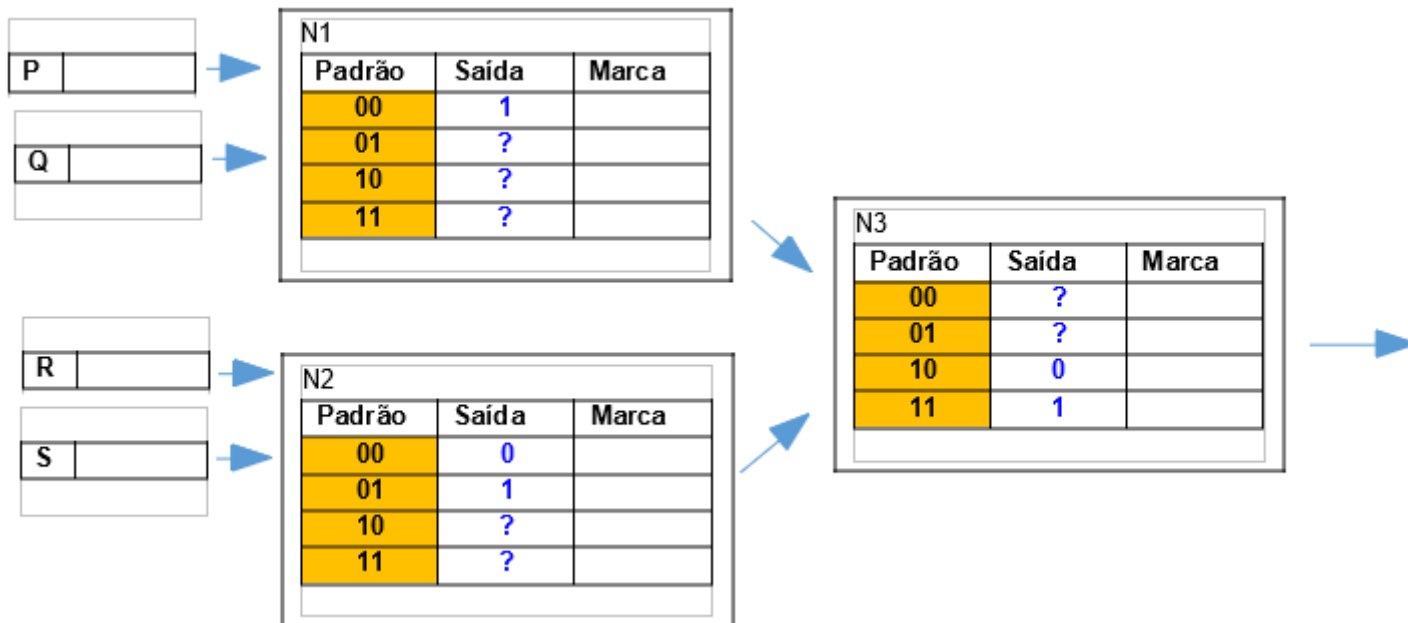
Rede Neural Sem Peso

Por essa razão a Saída de N2 deve ser atualizada para “1”.



Rede Neural Sem Peso

Limpando Entradas e Marcas antes de prosseguirmos para a 3ª Lição.



Rede Neural Sem Peso

3ª Lição:

Lições							
Lição	P	Q	R	S	$P \wedge Q$	$R \vee S$	$(P \wedge Q) \vee (R \vee S)$
1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	1
3	0	0	1	0	0	1	1

P	0
Q	0

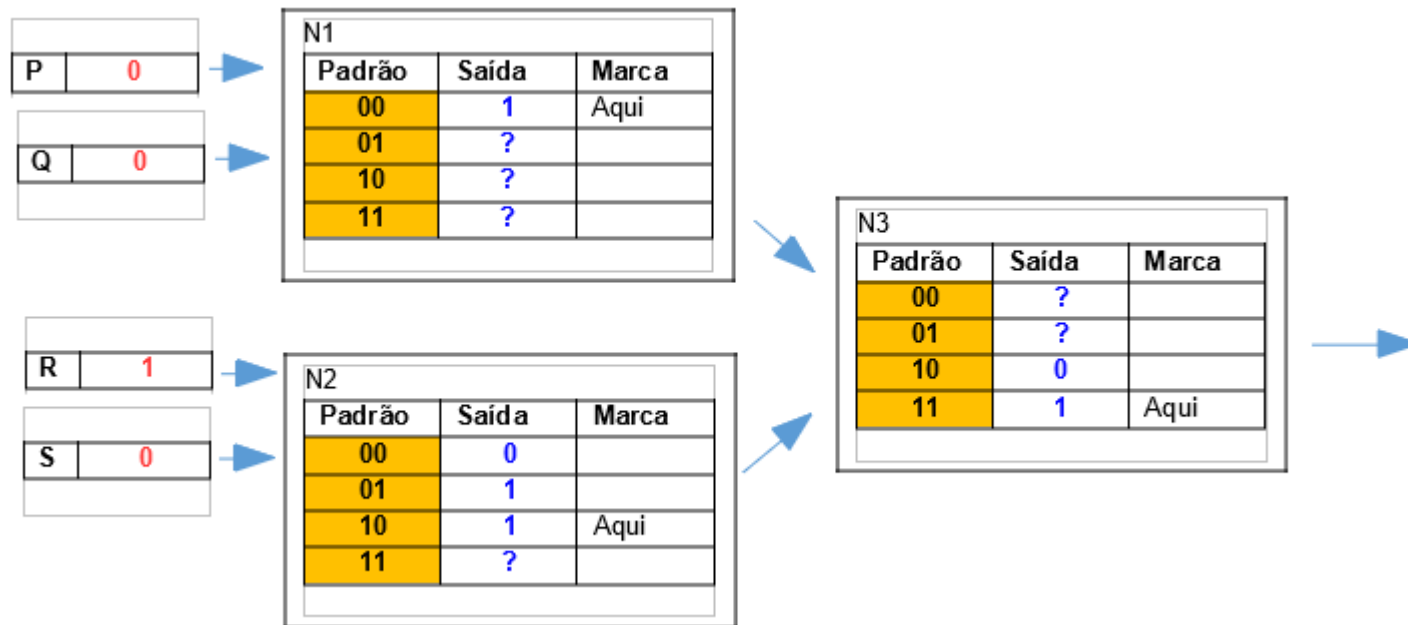
N1		
Padrão	Saída	Marca
00	1	Aqui
01	?	
10	?	
11	?	

R	1
S	0

N2		
Padrão	Saída	Marca
00	0	
01	1	
10	?	Aqui
11	?	

N3		
Padrão	Saída	Marca
00	?	
01	?	
10	0	
11	1	Aqui

Rede Neural Sem Peso



Até aqui nossa RNA nada fez de notável.
Nem mesmo se comportou como uma
Inteligência Artificial (IA).

Rede Neural Sem Peso

4ª Lição:

Lições							
Lição	P	Q	R	S	$P \wedge Q$	$R \vee S$	$(P \wedge Q) \vee (R \vee S)$
1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	1
3	0	0	1	0	0	1	1
4	0	0	1	1	0	1	1

P	0
Q	0

N1		
Padrão	Saída	Marca
00	1	Aqui
01	?	
10	?	
11	?	

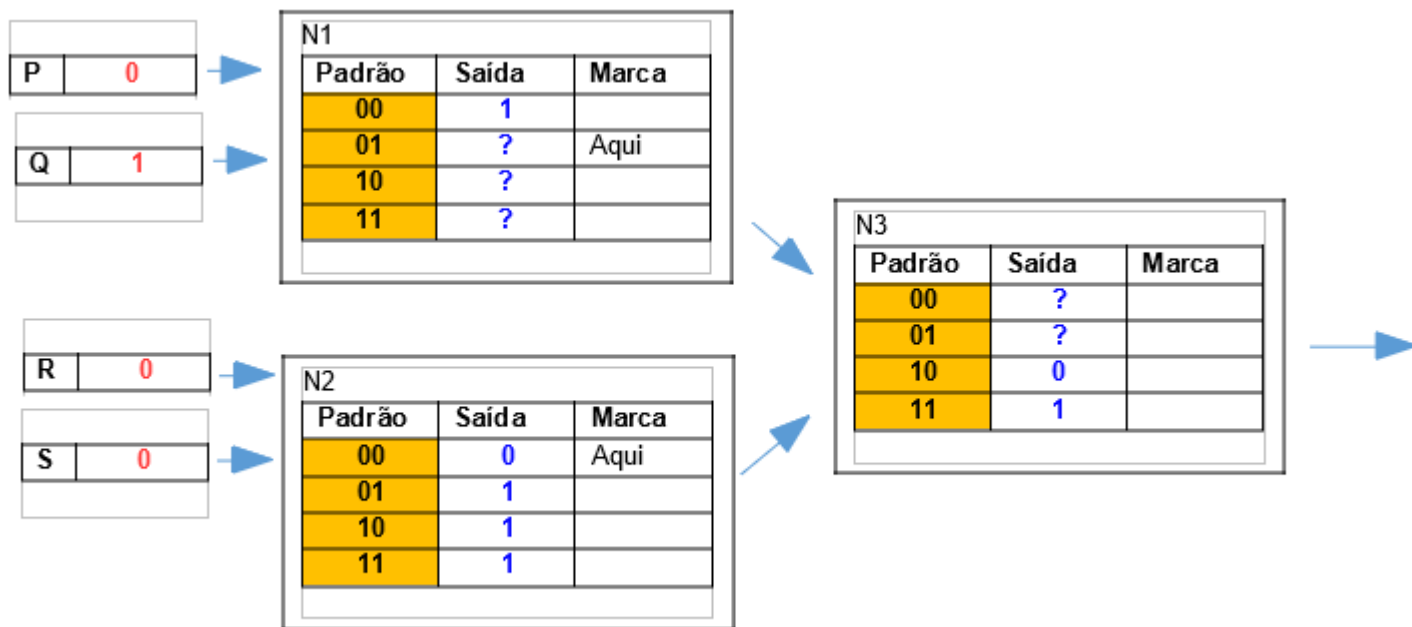
R	1
S	1

N2		
Padrão	Saída	Marca
00	0	
01	1	
10	1	
11	?	Aqui

N3		
Padrão	Saída	Marca
00	?	
01	?	
10	0	
11	1	Aqui

Rede Neural Sem Peso

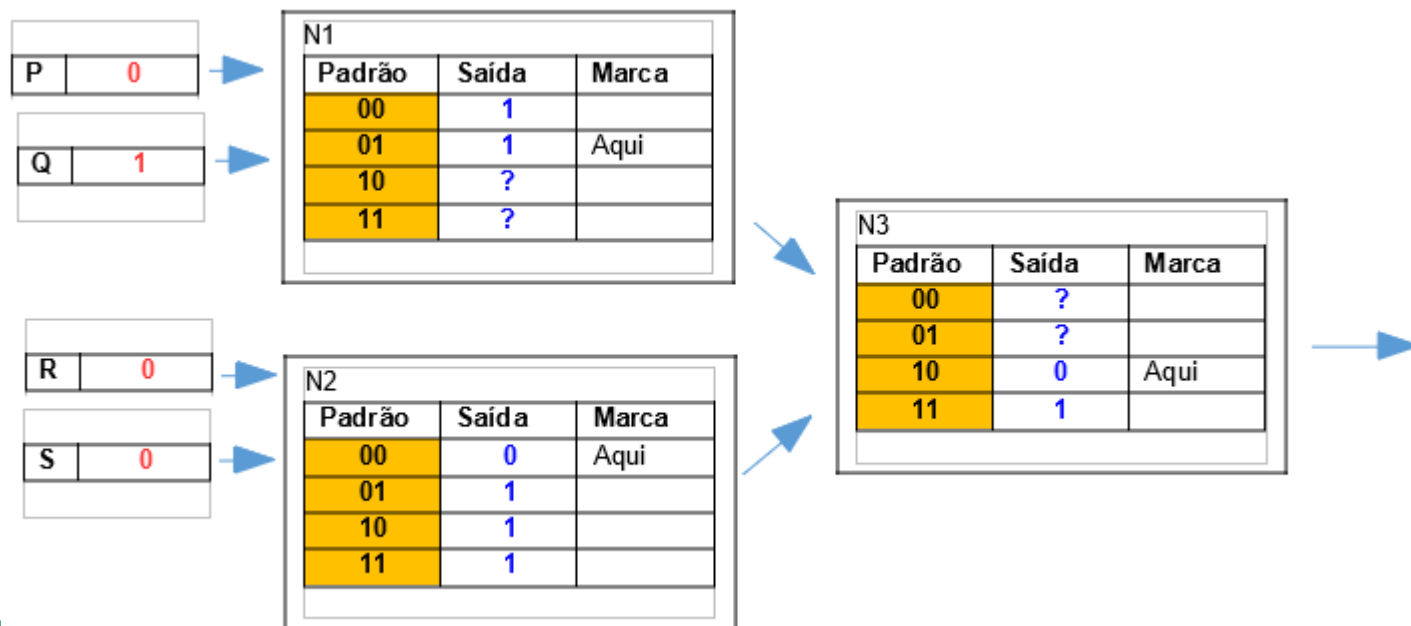
5ª Lição: Saída Esperada = 0.



Rede Neural Sem Peso

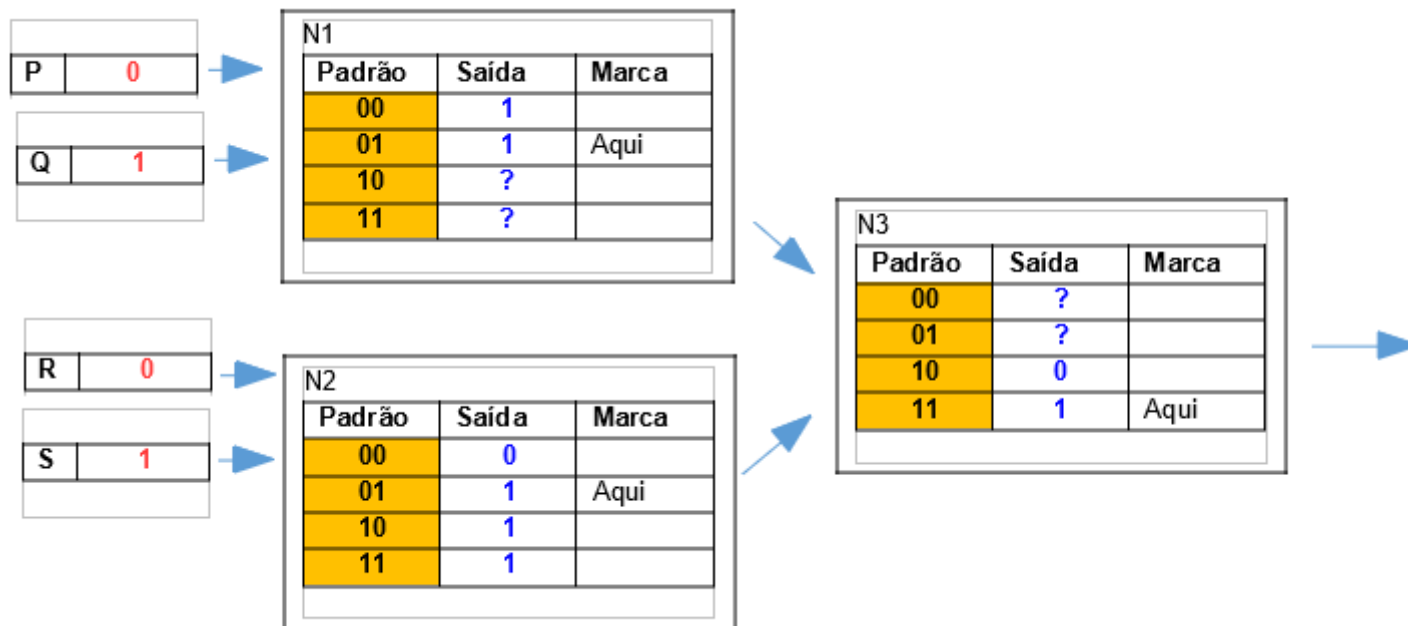
Em N3 já existe algum padrão terminado em “0” (“?0”) cuja saída seja “0”?

Sim, existe o padrão “10”.



Rede Neural Sem Peso

6ª Lição: Saída Esperada = 1.



Rede Neural Sem Peso

Aqui, na 6ª Lição, algo inédito acontece.

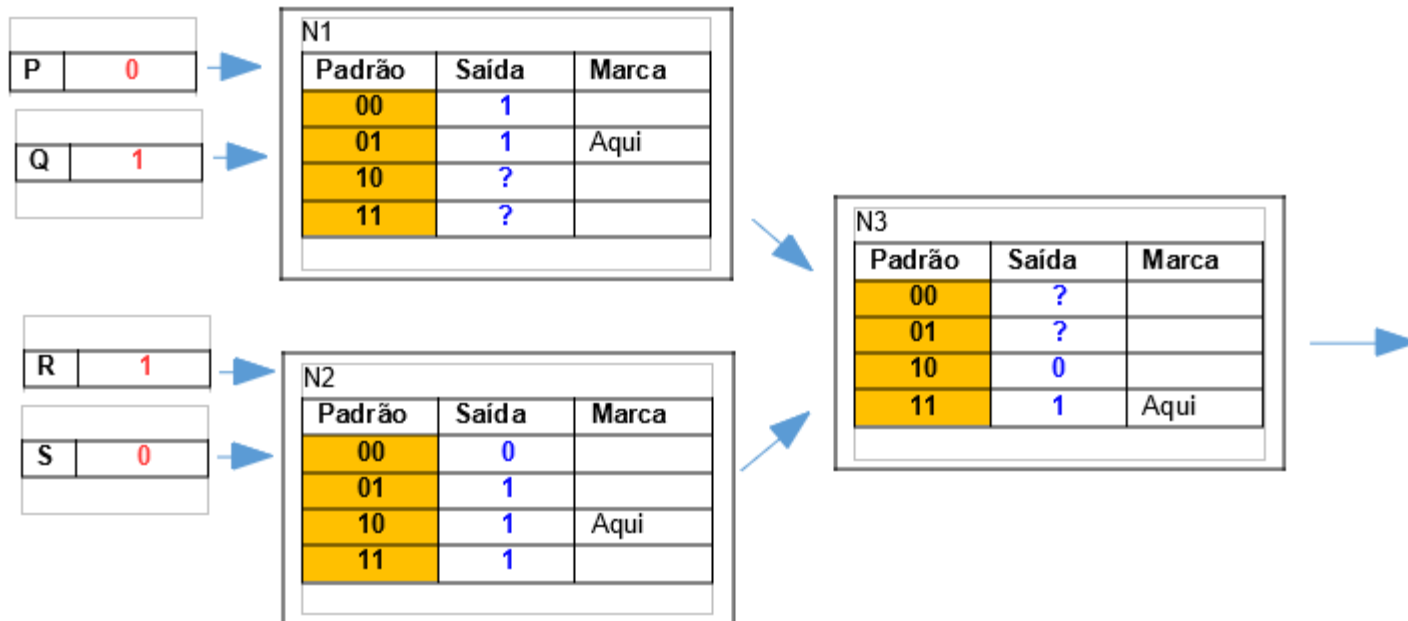
A entrada $(P, Q, R, S) = (0, 1, 0, 1)$ é inédita – A RNA não a “conhecia” ainda.

Mesmo assim, a RNA não perguntou pela Saída que N3 deveria apresentar.

Com base no aprendizado anterior a RNA respondeu que a Saída de N3 deveria ser “1”.

Rede Neural Sem Peso

7ª Lição: Saída Esperada = 1.



Rede Neural Sem Peso

Aqui na 7ª Lição a RNA inferiu sozinha a resposta.

