Perceptron

Gaspar J. Machado

Departamento de Matemática, Universidade do Minho

março de 2024

Tipos de aprendizagem

- **Aprendizagem supervisionada** (**supervised learning**): algoritmos que têm por objetivo aprender uma relação entre variáveis dependentes e variáveis independentes a partir de um conjunto de dados em que esta relação é conhecida, chamado **training set**. Por exemplo, técnicas de **regressão** e **classificação**.

Tipos de aprendizagem

- **Aprendizagem supervisionada** (**supervised learning**): algoritmos que têm por objetivo aprender uma relação entre variáveis dependentes e variáveis independentes a partir de um conjunto de dados em que esta relação é conhecida, chamado **training set**.
  Por exemplo, técnicas de **regressão** e **classificação**.
- **Aprendizagem não-supervisionada** (**Unsupervised learning**): algoritmos que têm por objetivo identificar padrões em dados não rotulados.
  Por exemplo, técnicas de **Clustering** e **redução de dimensionalidade**.

Tipos de aprendizagem

- **Aprendizagem supervisionada** (**supervised learning**): algoritmos que têm por objetivo aprender uma relação entre variáveis dependentes e variáveis independentes a partir de um conjunto de dados em que esta relação é conhecida, chamado **training set**.
  Por exemplo, técnicas de **regressão** e **classificação**.
- **Aprendizagem não-supervisionada** (**Unsupervised learning**): algoritmos que têm por objetivo identificar padrões em dados não rotulados.
  Por exemplo, técnicas de **Clustering** e **redução de dimensionalidade**.
- **Aprendizagem por reforço** (**Reinforcement Learning**): algoritmos que têm por objetivo aprender qual a melhor ação perante diferentes situações que vão surgindo de acordo com um processo de teste e erro em que se recompensam as boas decisões e se punem as más decisões.
  Por exemplo, técnicas de **Q Learning** e **Deep Q Network**.

Aprendizagem supervisionada

- O algoritmo baseia-se num *training set* (dados para treino) rotulado, ou seja, os eventos da base de dados incluem o valor das variáveis independentes e da variável dependente.

Aprendizagem supervisionada

- O algoritmo baseia-se num *training set* (dados para treino) rotulado, ou seja, os eventos da base de dados incluem o valor das variáveis independentes e da variável dependente.
- O modelo é treinado por forma a obter uma função que tenta construir a relação entre as variáveis independentes e a variável dependente do *training set*.

Aprendizagem supervisionada

- O algoritmo baseia-se num *training set* (dados para treino) rotulado, ou seja, os eventos da base de dados incluem o valor das variáveis independentes e da variável dependente.
- O modelo é treinado por forma a obter uma função que tenta construir a relação entre as variáveis independentes e a variável dependente do *training set*.
- Para um novo conjunto de valores das variáveis independentes o modelo faz uma predição (inferência) do valor da variável dependente.

Algoritmos de aprendizagem supervisionada (i)

- **Regressão**: problemas em que a variável dependente é contínua. Exemplos: qual o preço de uma casa em Braga com $150\,\mathrm{m}^2$? qual o peso de uma criança com 2 anos?

Algoritmos de aprendizagem supervisionada (i)

- **Regressão**: problemas em que a variável dependente é contínua. Exemplos: qual o preço de uma casa em Braga com $150\,\mathrm{m}^2$? qual o peso de uma criança com 2 anos?
- **Classificação**: problemas em que a variável dependente é categórica (classes).

Algoritmos de aprendizagem supervisionada (i)

- **Regressão**: problemas em que a variável dependente é contínua. Exemplos: qual o preço de uma casa em Braga com $150\,\mathrm{m}^2$? qual o peso de uma criança com 2 anos?
- **Classificação**: problemas em que a variável dependente é categórica (classes).
  - **Classificador binário**: a variável dependente é do tipo Sim/Não. Exemplos: este e-mail é *spam*? hoje vai chover? esta imagem é um cão? esta imagem é um cão ou um gato? este doente tem gripe?

Algoritmos de aprendizagem supervisionada (i)

- **Regressão**: problemas em que a variável dependente é contínua. Exemplos: qual o preço de uma casa em Braga com $150\,\mathrm{m}^2$? qual o peso de uma criança com 2 anos?
- **Classificação**: problemas em que a variável dependente é categórica (classes).
  - **Classificador binário**: a variável dependente é do tipo Sim/Não. Exemplos: este e-mail é *spam*? hoje vai chover? esta imagem é um cão? esta imagem é um cão ou um gato? este doente tem gripe?
    - **Classificador binário linear**: classes separadas por uma "reta".

Algoritmos de aprendizagem supervisionada (i)

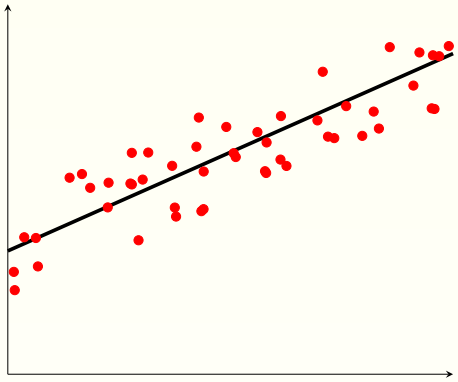
- **Regressão**: problemas em que a variável dependente é contínua. Exemplos: qual o preço de uma casa em Braga com $150\,\mathrm{m}^2$? qual o peso de uma criança com 2 anos?
- **Classificação**: problemas em que a variável dependente é categórica (classes).
  - **Classificador binário**: a variável dependente é do tipo Sim/Não. Exemplos: este e-mail é *spam*? hoje vai chover? esta imagem é um cão? esta imagem é um cão ou um gato? este doente tem gripe?
    - **Classificador binário linear**: classes separadas por uma "reta".
    - **Classificador binário não linear**: classes separadas por uma "curva".

Algoritmos de aprendizagem supervisionada (i)

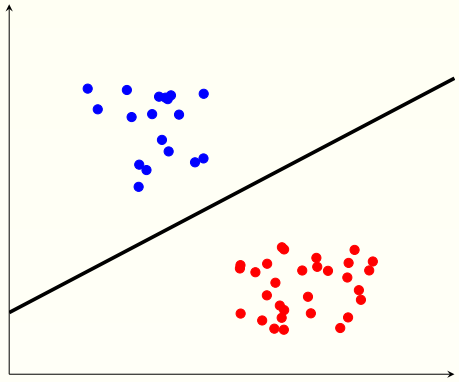
- **Regressão**: problemas em que a variável dependente é contínua.
  Exemplos: qual o preço de uma casa em Braga com $150\,\mathrm{m}^2$? qual o peso de uma criança com 2 anos?
- **Classificação**: problemas em que a variável dependente é categórica (classes).
  - **Classificador binário**: a variável dependente é do tipo Sim/Não.
    Exemplos: este e-mail é *spam*? hoje vai chover? esta imagem é um cão? esta imagem é um cão ou um gato? este doente tem gripe?
    - **Classificador binário linear**: classes separadas por uma "reta".
    - **Classificador binário não linear**: classes separadas por uma "curva".
  - **Classificador multi-classe**: a variável dependente pode assumir um de J valores, $J \geqslant 3$.
    Exemplos: que dígito é este? em que língua oficial da União Europeia está este texto escrito?

Algoritmos de aprendizagem supervisionada (ii)

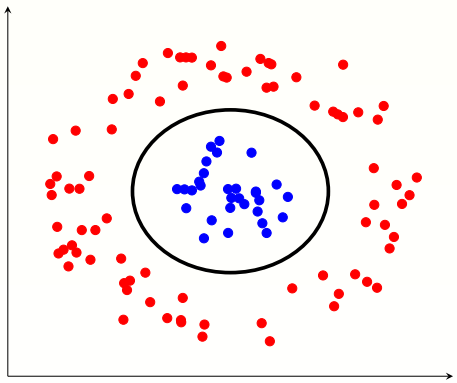
regressão



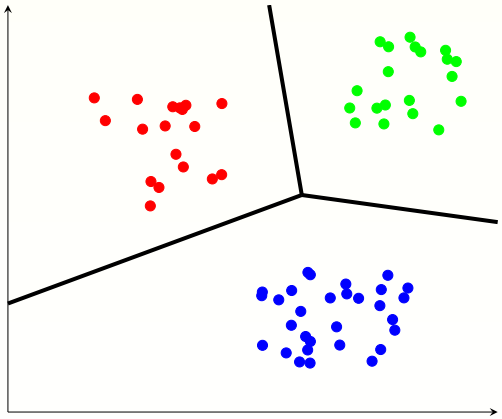
classificador binário linear



classificador binário não linear



classificador multi-classe



Base de dados | notação

- **Input space**: $\mathcal{A} = \mathbb{R}^I$.

Base de dados | notação

- **Input space**: $\mathcal{A} = \mathbb{R}^I$.
- **Output space** (espaço das classes): $\mathcal{C} = \{-1, +1\}$.

Base de dados | notação

- **Input space**: $\mathcal{A} = \mathbb{R}^I$.
- **Output space** (espaço das classes): $\mathcal{C} = \{-1, +1\}$.
- **Base de dados** (**dataset**): $D = (x^n, y^n)_{n=1}^N$:

Base de dados | notação

- **Input space**: $\mathcal{A} = \mathbb{R}^I$.
- **Output space** (espaço das classes): $\mathcal{C} = \{-1, +1\}$.
- **Base de dados** (**dataset**): $D = (x^n, y^n)_{n=1}^N$:
  - Evento ou ocorrência: $\rho^n = (x^n, y^n)$.

Base de dados | notação

- **Input space**: $\mathcal{A} = \mathbb{R}^I$.
- **Output space** (espaço das classes): $\mathcal{C} = \{-1, +1\}$.
- **Base de dados** (**dataset**): $D = (x^n, y^n)_{n=1}^N$:
  - Evento ou ocorrência: $\rho^n = (x^n, y^n)$.
  - $x^n = (x_1^n, \ldots, x_I^n)^\top \in \mathcal{A} = \mathbb{R}^I$: **atributos** do evento n.

Base de dados | notação

- **Input space**: $\mathcal{A} = \mathbb{R}^I$.
- **Output space** (espaço das classes): $\mathcal{C} = \{-1, +1\}$.
- **Base de dados** (**dataset**): $D = (x^n, y^n)_{n=1}^N$:
  - Evento ou ocorrência: $\rho^n = (x^n, y^n)$.
  - $x^n = (x_1^n, \ldots, x_I^n)^\top \in \mathcal{A} = \mathbb{R}^I$: **atributos** do evento n.
  - $y^n \in \mathcal{C} = \{-1, +1\}$: **rótulo** ou **etiqueta** (**label**, **target**) do evento n.

Base de dados | notação

- **Input space**: $\mathcal{A} = \mathbb{R}^I$.
- **Output space** (espaço das classes): $\mathcal{C} = \{-1, +1\}$.
- **Base de dados (dataset)**: $D = (x^n, y^n)_{n=1}^N$:
  - Evento ou ocorrência: $\rho^n = (x^n, y^n)$.
  - $x^n = (x_1^n, \ldots, x_I^n)^\top \in \mathcal{A} = \mathbb{R}^I$: **atributos** do evento n.
  - $y^n \in \mathcal{C} = \{-1, +1\}$: **rótulo** ou **etiqueta** (**label**, **target**) do evento n.

Exemplo de uma base de dados. $D = (x^n, y^n)_{n=1}^5$ com

$$
\begin{aligned}
x^1 &= (0.3, 0.7, 0.5)^\top \quad y^1 = +1 \\
x^2 &= (0.1, 0.4, 0.1)^\top \quad y^2 = -1 \\
x^3 &= (0.3, 0.2, 0.5)^\top \quad y^3 = -1 \\
x^4 &= (0.2, 0.2, 0.3)^\top \quad y^4 = +1 \\
x^5 &= (0.9, 0.9, 0.2)^\top \quad y^5 = -1
\end{aligned}
$$

Hiperplano | definição

- **Definição.** Sejam $w = (w_1, \ldots, w_I)^\top \in \mathbb{R}^I - \{\underline{0}\}$ e $b \in \mathbb{R}$.
  Chama-se **hiperplano** em $\mathbb{R}^I$ ao conjunto de pontos

$$\mathcal{X}_{w,b} = \{(x_1, \ldots, x_I)^\top \in \mathbb{R}^I : \underbrace{w_1 x_1 + \cdots + w_I x_I + b}_{p(x;w,b)} = 0\}.$$

- na definição anterior, w é um vetor normal ao hiperplano $\mathcal{X}_{w,b}$
- a definição de hiperplano generaliza a definição de plano
- exemplos
  - $I = 1$: um hiperplano é um ponto
  - $I = 2$: um hiperplano é uma reta
  - $I = 3$: um hiperplano é um plano

Hiperplano | notação

- Versão afim — w, b

Hiperplano | notação

- Versão afim — w, b
  - variável: $x = (x_1, \ldots, x_I)^\top \in \mathbb{R}^I$

Hiperplano | notação

- Versão afim — w, b
  - variável: $x = (x_1, \ldots, x_I)^\top \in \mathbb{R}^I$
  - parâmetros: $w = (w_1, \ldots, w_I)^\top \in \mathbb{R}^I$, $b \in \mathbb{R}$

Hiperplano | notação

- Versão afim — w, b
  - variável: $x = (x_1, \ldots, x_I)^\top \in \mathbb{R}^I$
  - parâmetros: $w = (w_1, \ldots, w_I)^\top \in \mathbb{R}^I$, $b \in \mathbb{R}$
  - hiperplano:

$$p(x; w, b) = w_1 x_1 + \cdots + w_I x_I + b = w^\top x + b = w \cdot x + b$$

$$\mathcal{X}_{w,b} = \{(x_1, \ldots, x_I)^\top \in \mathbb{R}^I : w \cdot x + b = 0\}.$$

Hiperplano | notação

- Versão afim — w, b
  - variável: $x = (x_1, \ldots, x_I)^\top \in \mathbb{R}^I$
  - parâmetros: $w = (w_1, \ldots, w_I)^\top \in \mathbb{R}^I$, $b \in \mathbb{R}$
  - hiperplano:

$$p(x; w, b) = w_1 x_1 + \cdots + w_I x_I + b = w^\top x + b = w \cdot x + b$$
$$\mathcal{X}_{w,b} = \{(x_1, \ldots, x_I)^\top \in \mathbb{R}^I : w \cdot x + b = 0\}.$$

- Versão vetorial — $\tilde{w}$ (mais compacta)

Hiperplano | notação

- Versão afim — w, b
  - variável: $x = (x_1, \ldots, x_I)^\top \in \mathbb{R}^I$
  - parâmetros: $w = (w_1, \ldots, w_I)^\top \in \mathbb{R}^I$, $b \in \mathbb{R}$
  - hiperplano:

$$p(x; w, b) = w_1 x_1 + \cdots + w_I x_I + b = w^\top x + b = w \cdot x + b$$
$$\mathcal{X}_{w,b} = \{(x_1, \ldots, x_I)^\top \in \mathbb{R}^I : w \cdot x + b = 0\}.$$

- Versão vetorial — $\tilde{w}$ (mais compacta)
  - variável: $\tilde{x} = (1, x_1, \ldots, x_I)^\top \in \mathbb{R}^{I+1}$

Hiperplano | notação

- Versão afim — w, b
  - variável: $x = (x_1, \ldots, x_I)^\top \in \mathbb{R}^I$
  - parâmetros: $w = (w_1, \ldots, w_I)^\top \in \mathbb{R}^I$, $b \in \mathbb{R}$
  - hiperplano:

$$p(x; w, b) = w_1 x_1 + \cdots + w_I x_I + b = w^\top x + b = w \cdot x + b$$
$$\mathcal{X}_{w,b} = \{(x_1, \ldots, x_I)^\top \in \mathbb{R}^I : w \cdot x + b = 0\}.$$

- Versão vetorial — $\tilde{w}$ (mais compacta)
  - variável: $\tilde{x} = (1, x_1, \ldots, x_I)^\top \in \mathbb{R}^{I+1}$
  - parâmetros: $\tilde{w} = (w_0, w_1, \ldots, w_I)^\top \in \mathbb{R}^{I+1}$

Hiperplano | notação

- Versão afim — w, b
  - variável: $x = (x_1, \ldots, x_I)^\top \in \mathbb{R}^I$
  - parâmetros: $w = (w_1, \ldots, w_I)^\top \in \mathbb{R}^I$, $b \in \mathbb{R}$
  - hiperplano:

$$p(x; w, b) = w_1 x_1 + \cdots + w_I x_I + b = w^\top x + b = w \cdot x + b$$
$$\mathcal{X}_{w,b} = \{(x_1, \ldots, x_I)^\top \in \mathbb{R}^I : w \cdot x + b = 0\}.$$

- Versão vetorial — $\tilde{w}$ (mais compacta)
  - variável: $\tilde{x} = (1, x_1, \ldots, x_I)^\top \in \mathbb{R}^{I+1}$
  - parâmetros: $\tilde{w} = (w_0, w_1, \ldots, w_I)^\top \in \mathbb{R}^{I+1}$
  - hiperplano:

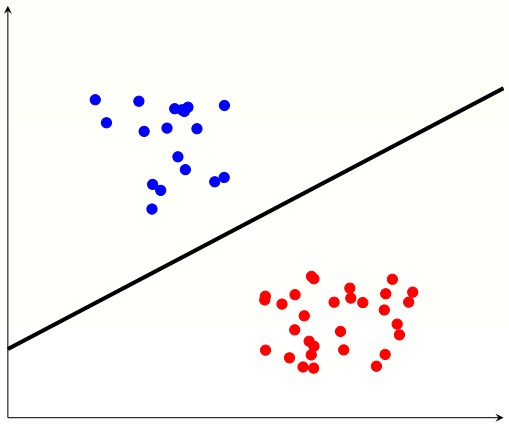
$$p(x; \tilde{w}) = w_0 \times 1 + w_1 x_1 + \cdots + w_I x_I = \tilde{w}^\top \tilde{x} = \tilde{w} \cdot \tilde{x}$$
$$\mathcal{X}_{\tilde{w}} = \{(x_1, \ldots, x_I)^\top \in \mathbb{R}^I : \tilde{w} \cdot \tilde{x} = 0\}.$$

Base de dados linearmente separável | definição

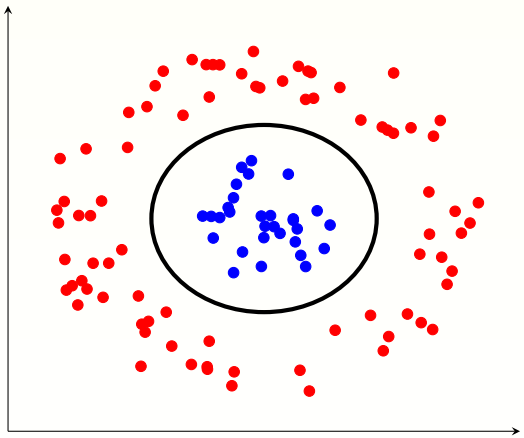
- **Definição.** A base de dados $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^I$, $y^n \in \{-1, +1\}$, diz-se **linearmente separável** se existir $\tilde{w} \in \mathbb{R}^{I+1}$ tal que
  - se $y^n = +1$, então $\tilde{w} \cdot \tilde{x} > 0$ e
  - se $y^n = -1$, então $\tilde{w} \cdot \tilde{x} < 0$,

  que são equivalentes à condição única
  - $y^n(\tilde{w} \cdot \tilde{x}) > 0, n = 1, \ldots, N.$

linearmente separável



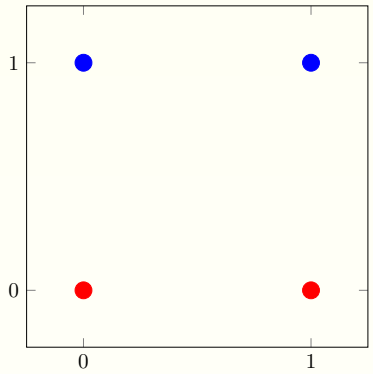
não linearmente separável



Base de dados linearmente separável | sim e não

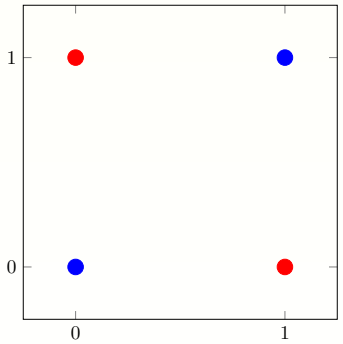
Exemplo. Exemplo de uma base de dados que é linearmente separável: a base de dados $D = (x^n, y^n)_{n=1}^4$,

$$x^1 = (0,0)^\top \quad y^1 = +1$$
$$x^2 = (1,0)^\top \quad y^2 = +1$$
$$x^3 = (0,1)^\top \quad y^3 = -1$$
$$x^4 = (1,1)^\top \quad y^4 = -1$$



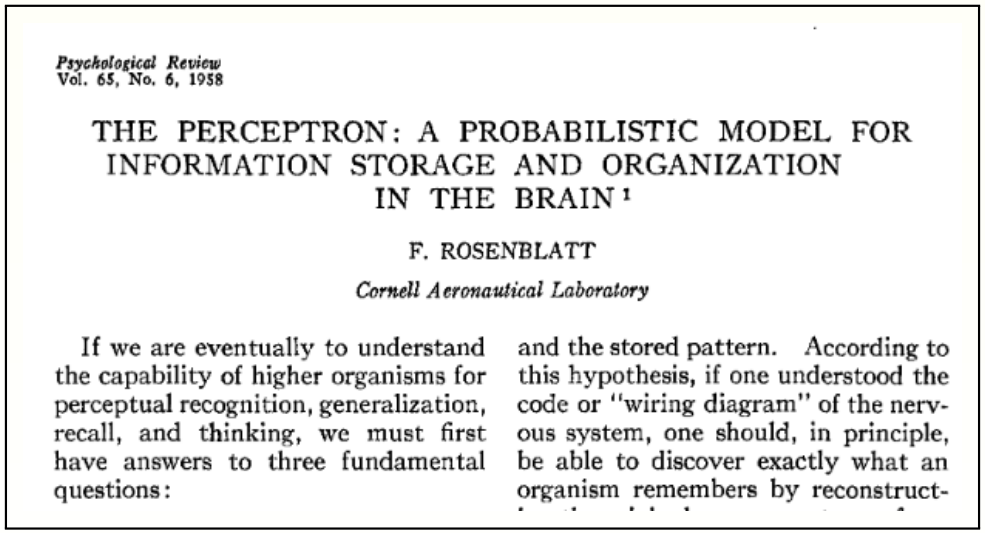
Exemplo. (operador lógico XOR) Exemplo de uma base de dados que não é linearmente separável: a base de dados $D = (x^n, y^n)_{n=1}^4$ com

$$x^1 = (0,0)^\top \quad y^1 = -1$$
$$x^2 = (1,0)^\top \quad y^2 = +1$$
$$x^3 = (0,1)^\top \quad y^3 = +1$$
$$x^4 = (1,1)^\top \quad y^4 = -1$$



Perceptron | objetivo

- Seja a base de dados $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^I$, $y^n \in \{-1, +1\}$.
- Se D é linearmente separável, como procurar $\tilde{w}$?
- E se D não é linearmente separável, como procurar $\tilde{w}$ tal que a condição $y^n(\tilde{w} \cdot \tilde{x}) > 0$ seja satisfeita na maioria dos eventos?
- Uma primeira resposta: algoritmo *Perceptron* (Percetrão), introduzido por F. Rosenblatt em 1958, que é considerado o primeiro modelo de redes neuronais.



Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN[1]

F. ROSENBLATT

Cornell Aeronautical Laboratory

If we are eventually to understand the capability of higher organisms for perceptual recognition, generalization, recall, and thinking, we must first have answers to three fundamental questions:

and the stored pattern. According to this hypothesis, if one understood the code or "wiring diagram" of the nervous system, one should, in principle, be able to discover exactly what an organism remembers by reconstruct-

Perceptron | ideia

- O *Perceptron* é um modelo matemático que se inspira no comportamento dos neurónios (células do sistema nervoso responsáveis pela transmissão de informações na forma de sinais elétricos).

Perceptron | ideia

- O *Perceptron* é um modelo matemático que se inspira no comportamento dos neurónios (células do sistema nervoso responsáveis pela transmissão de informações na forma de sinais elétricos).
- Um neurónio recebe um conjunto de sinais de entrada (dendrites) e produz um sinal de saída.

Perceptron | ideia

- O *Perceptron* é um modelo matemático que se inspira no comportamento dos neurónios (células do sistema nervoso responsáveis pela transmissão de informações na forma de sinais elétricos).
- Um neurónio recebe um conjunto de sinais de entrada (dendrites) e produz um sinal de saída.
- Os sinais de entrada são pesados e combinados produzindo um sinal de saída.

Perceptron | ideia

- O *Perceptron* é um modelo matemático que se inspira no comportamento dos neurónios (células do sistema nervoso responsáveis pela transmissão de informações na forma de sinais elétricos).
- Um neurónio recebe um conjunto de sinais de entrada (dendrites) e produz um sinal de saída.
- Os sinais de entrada são pesados e combinados produzindo um sinal de saída.
- Existe um *bias* (viés), que representa o nível de ativação.

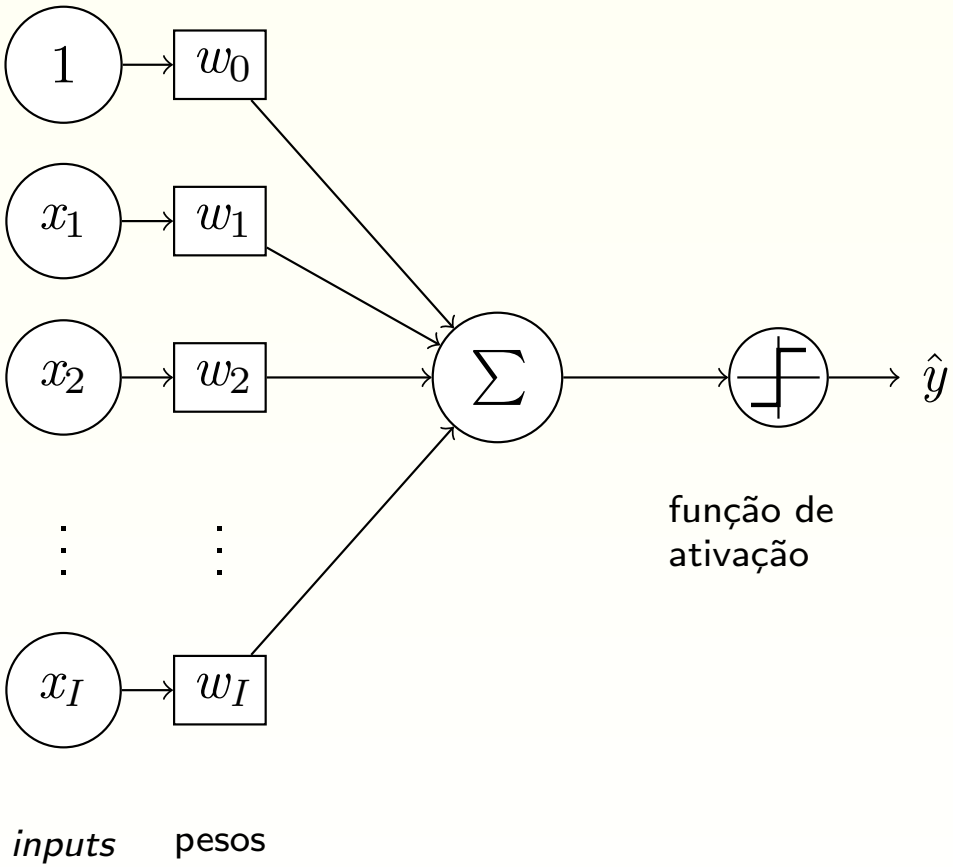
Perceptron | ideia

- O *Perceptron* é um modelo matemático que se inspira no comportamento dos neurónios (células do sistema nervoso responsáveis pela transmissão de informações na forma de sinais elétricos).
- Um neurónio recebe um conjunto de sinais de entrada (dendrites) e produz um sinal de saída.
- Os sinais de entrada são pesados e combinados produzindo um sinal de saída.
- Existe um *bias* (viés), que representa o nível de ativação.
- Existe uma função de transferência sendo o sinal de saída -1 (inibição) ou $+1$ (ativação).

Perceptron | ideia

- O *Perceptron* é um modelo matemático que se inspira no comportamento dos neurónios (células do sistema nervoso responsáveis pela transmissão de informações na forma de sinais elétricos).
- Um neurónio recebe um conjunto de sinais de entrada (dendrites) e produz um sinal de saída.
- Os sinais de entrada são pesados e combinados produzindo um sinal de saída.
- Existe um *bias* (viés), que representa o nível de ativação.
- Existe uma função de transferência sendo o sinal de saída -1 (inibição) ou $+1$ (ativação).
- Não usa técnicas de otimização mas antes um processo (muito simples mas) muito engenhoso.

Perceptron | esquema



Perceptron | ingredientes (i)

- Base de dados

$$D = (x^n, y^n)_{n=1}^N, x^n \in \mathbb{R}^I, y^n \in \{-1, +1\}.$$

Perceptron | ingredientes (i)

- Base de dados

$$D = (x^n, y^n)_{n=1}^N, x^n \in \mathbb{R}^I, y^n \in \{-1, +1\}.$$

- Arquitetura da *Machine Learning*

$$h(x; \tilde{w}) = \operatorname{sgn}(\tilde{w} \cdot \tilde{x}).$$

Perceptron | ingredientes (i)

- Base de dados

$$D = (x^n, y^n)_{n=1}^N, x^n \in \mathbb{R}^I, y^n \in \{-1, +1\}.$$

- Arquitetura da *Machine Learning*

$$h(x; \tilde{w}) = \operatorname{sgn}(\tilde{w} \cdot \tilde{x}).$$

- Aprendizagem: determinar $\tilde{w}$ (um elemento do espaço das hipóteses $\mathcal{H} = \mathbb{R}^{I+1}$).

Perceptron | ingredientes (i)

- Base de dados

$$D = (x^n, y^n)_{n=1}^N, x^n \in \mathbb{R}^I, y^n \in \{-1, +1\}.$$

- Arquitetura da *Machine Learning*

$$h(x; \tilde{w}) = \operatorname{sgn}(\tilde{w} \cdot \tilde{x}).$$

- Aprendizagem: determinar $\tilde{w}$ (um elemento do espaço das hipóteses $\mathcal{H} = \mathbb{R}^{I+1}$).
- Recorde — função sinal: seja $a \in \mathbb{R}$. Então,

$$\operatorname{sgn}(a) = \begin{cases} +1 & \text{se } a > 0, \\ -1 & \text{se } a \leqslant 0. \end{cases}$$

Note-se que se considera $\operatorname{sgn}(0) = -1$ por uma simples questão de convenção.

Perceptron | ingredientes (ii)

- Confrontação entre o valor real y^n do evento n e o seu valor inferido $\hat{y}^n = h(x^n; \tilde{w})$.

Perceptron | ingredientes (ii)

- Confrontação entre o valor real y^n do evento n e o seu valor inferido $\hat{y}^n = h(x^n; \tilde{w})$.
- Erro relativo ao evento n

$$|y^n - \hat{y}^n| = \begin{cases} 0 & \text{se } y^n = \hat{y}^n, \\ 2 & \text{se } y^n \neq \hat{y}^n. \end{cases}$$

Perceptron | ingredientes (ii)

- Confrontação entre o valor real y^n do evento n e o seu valor inferido $\hat{y}^n = h(x^n; \tilde{w})$.
- Erro relativo ao evento n

$$|y^n - \hat{y}^n| = \begin{cases} 0 & \text{se } y^n = \hat{y}^n, \\ 2 & \text{se } y^n \neq \hat{y}^n. \end{cases}$$

- Objetivo: minimizar função custo $(\tilde{w}, D) \mapsto E(\tilde{w}, D) \in \mathbb{R}_0^+$ tal que

Perceptron | ingredientes (ii)

- Confrontação entre o valor real y^n do evento n e o seu valor inferido $\hat{y}^n = h(x^n; \tilde{w})$.
- Erro relativo ao evento n

$$|y^n - \hat{y}^n| = \begin{cases} 0 & \text{se } y^n = \hat{y}^n, \\ 2 & \text{se } y^n \neq \hat{y}^n. \end{cases}$$

- Objetivo: minimizar função custo $(\tilde{w}, D) \mapsto E(\tilde{w}, D) \in \mathbb{R}_0^+$ tal que
  - $E(\tilde{w}, D) = 0$ se todos os eventos da base de dados são bem classificados (pode haver outros critérios) e

Perceptron | ingredientes (ii)

- Confrontação entre o valor real y^n do evento n e o seu valor inferido $\hat{y}^n = h(x^n; \tilde{w})$.
- Erro relativo ao evento n

$$|y^n - \hat{y}^n| = \begin{cases} 0 & \text{se } y^n = \hat{y}^n, \\ 2 & \text{se } y^n \neq \hat{y}^n. \end{cases}$$

- Objetivo: minimizar função custo $(\tilde{w}, D) \mapsto E(\tilde{w}, D) \in \mathbb{R}_0^+$ tal que
  - $E(\tilde{w}, D) = 0$ se todos os eventos da base de dados são bem classificados (pode haver outros critérios) e
  - Quantos mais eventos mal classificados existirem, maior deve ser $E(\tilde{w}, D)$ (pode haver outros critérios).

Perceptron | ingredientes (ii)

- Confrontação entre o valor real y^n do evento n e o seu valor inferido $\hat{y}^n = h(x^n; \tilde{w})$.
- Erro relativo ao evento n

$$|y^n - \hat{y}^n| = \begin{cases} 0 & \text{se } y^n = \hat{y}^n, \\ 2 & \text{se } y^n \neq \hat{y}^n. \end{cases}$$

- Objetivo: minimizar função custo $(\tilde{w}, D) \mapsto E(\tilde{w}, D) \in \mathbb{R}_0^+$ tal que
  - $E(\tilde{w}, D) = 0$ se todos os eventos da base de dados são bem classificados (pode haver outros critérios) e
  - Quantos mais eventos mal classificados existirem, maior deve ser $E(\tilde{w}, D)$ (pode haver outros critérios).
  - Exemplo:

$$\begin{aligned} E(\tilde{w}, D) &= \frac{\text{n}^{\underline{\text{o}}} \text{ de eventos mal classificados}}{N} \\ &= \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} |y^n - \hat{y}^n| . \end{aligned}$$

Perceptron | regra de aprendizagem (i)

- Para determinar

$$\tilde{w}^* = \left(\underset{\tilde{w} \in \mathbb{R}^{I+1}}{\arg\min}\, E(\tilde{w}, D) \right) \in \mathbb{R}^{I+1},$$

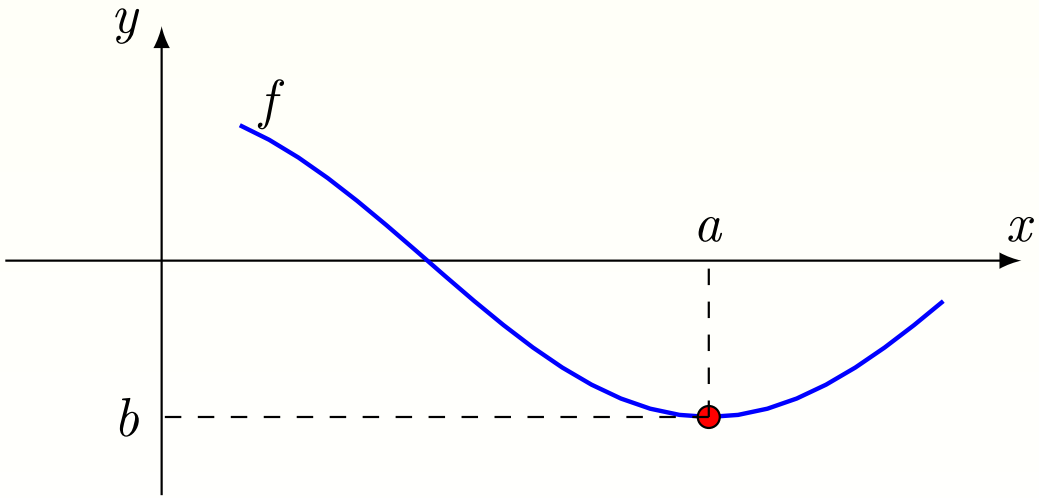
constrói-se uma sequência $\left(\tilde{w}_{(t)}\right)_{t \in \mathbb{N}_0}$ tal que $w_{(t)} \to w^*$ (quanto mais rápida for a convergência, melhor, idealmente num número finito de iterações).

- Para determinar

$$\tilde{w}^* = \left(\underset{\tilde{w} \in \mathbb{R}^{I+1}}{\arg\min}\, E(\tilde{w}, D) \right) \in \mathbb{R}^{I+1},$$

constrói-se uma sequência $\left(\tilde{w}_{(t)}\right)_{t \in \mathbb{N}_0}$ tal que $w_{(t)} \to w^*$ (quanto mais rápida for a convergência, melhor, idealmente num número finito de iterações).

- Recorde:



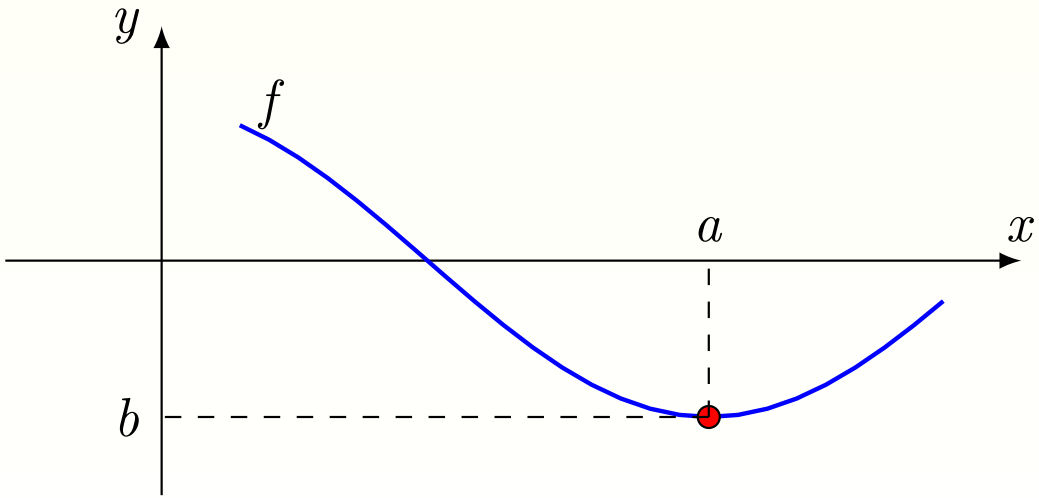
Perceptron | regra de aprendizagem (i)

- Para determinar

$$\tilde{w}^* = \left(\underset{\tilde{w} \in \mathbb{R}^{I+1}}{\arg\min} \, E(\tilde{w}, D) \right) \in \mathbb{R}^{I+1},$$

constrói-se uma sequência $\left(\tilde{w}_{(t)}\right)_{t \in \mathbb{N}_0}$ tal que $w_{(t)} \to w^*$ (quanto mais rápida for a convergência, melhor, idealmente num número finito de iterações).

- Recorde:



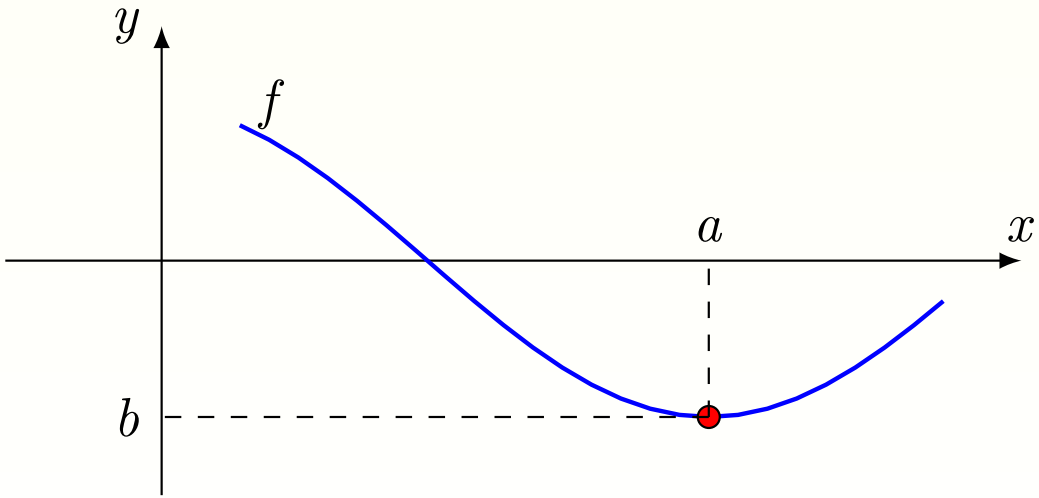
  - $\min f = b (= f(a))$

- Para determinar

$$\tilde{w}^* = \left(\underset{\tilde{w} \in \mathbb{R}^{I+1}}{\arg\min} \, E(\tilde{w}, D) \right) \in \mathbb{R}^{I+1},$$

constrói-se uma sequência $\left(\tilde{w}_{(t)}\right)_{t \in \mathbb{N}_0}$ tal que $w_{(t)} \to w^*$ (quanto mais rápida for a convergência, melhor, idealmente num número finito de iterações).

- Recorde:



  - $\min f = b (= f(a))$
  - $\arg\min f = a$

Perceptron | regra de aprendizagem (ii)

- Dado um hiperplano caracterizado por $\tilde{w}_{(t)}$ e uma ocorrência $(x^n, y^n) \in D$, calcula-se $\hat{y}^n = \operatorname{sgn}(\tilde{w}_{(t)} \cdot \tilde{x}^n)$.

Perceptron | regra de aprendizagem (ii)

- Dado um hiperplano caracterizado por $\tilde{w}_{(t)}$ e uma ocorrência $(x^n, y^n) \in D$, calcula-se $\hat{y}^n = \operatorname{sgn}(\tilde{w}_{(t)} \cdot \tilde{x}^n)$.
- Ocorrência bem classificada

$$\left.\begin{array}{l} y^n = +1, \hat{y}^n = +1 \Rightarrow \underbrace{y^n}_{=+1} \underbrace{(\tilde{w}_{(t)} \cdot \tilde{x}^n)}_{>0} > 0 \\ y^n = -1, \hat{y}^n = -1 \Rightarrow \underbrace{y^n}_{=-1} \underbrace{(\tilde{w}_{(t)} \cdot \tilde{x}^n)}_{\leqslant 0} \geqslant 0 \end{array}\right\} \Rightarrow \boxed{y^n (\tilde{w}_{(t)} \cdot \tilde{x}^n) \geqslant 0}$$

Perceptron | regra de aprendizagem (ii)

- Dado um hiperplano caracterizado por $\tilde{w}_{(t)}$ e uma ocorrência $(x^n, y^n) \in D$, calcula-se $\hat{y}^n = \operatorname{sgn}(\tilde{w}_{(t)} \cdot \tilde{x}^n)$.
- Ocorrência bem classificada

$$\left.\begin{array}{l} y^n = +1, \hat{y}^n = +1 \Rightarrow \underbrace{y^n}_{=+1} \underbrace{(\tilde{w}_{(t)} \cdot \tilde{x}^n)}_{>0} > 0 \\ y^n = -1, \hat{y}^n = -1 \Rightarrow \underbrace{y^n}_{=-1} \underbrace{(\tilde{w}_{(t)} \cdot \tilde{x}^n)}_{\leqslant 0} \geqslant 0 \end{array}\right\} \Rightarrow \boxed{y^n (\tilde{w}_{(t)} \cdot \tilde{x}^n) \geqslant 0}$$

- Ocorrência mal classificada

$$\left.\begin{array}{l} y^n = +1, \hat{y}^n = -1 \Rightarrow \underbrace{y^n}_{=+1} \underbrace{(\tilde{w}_{(t)} \cdot \tilde{x}^n)}_{\leqslant 0} \leqslant 0 \\ y^n = -1, \hat{y}^n = +1 \Rightarrow \underbrace{y^n}_{=-1} \underbrace{(\tilde{w}_{(t)} \cdot \tilde{x}^n)}_{>0} < 0 \end{array}\right\} \Rightarrow \boxed{y^n (\tilde{w}_{(t)} \cdot \tilde{x}^n) \leqslant 0}$$

Perceptron | regra de aprendizagem (iii)

- Se a ocorrência está bem classificada, o vetor dos pesos não precisa de ser atualizado (faz-se $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)}$).

Perceptron | regra de aprendizagem (iii)

- Se a ocorrência está bem classificada, o vetor dos pesos não precisa de ser atualizado (faz-se $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)}$).
- Se a ocorrência está mal classificada, o vetor dos pesos deve ser atualizado por forma a que

Perceptron | regra de aprendizagem (iii)

- Se a ocorrência está bem classificada, o vetor dos pesos não precisa de ser atualizado (faz-se $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)}$).
- Se a ocorrência está mal classificada, o vetor dos pesos deve ser atualizado por forma a que
  - $y^n \left(\tilde{w}_{(t+1)} \cdot \tilde{x}^n\right) > 0$

Perceptron | regra de aprendizagem (iii)

- Se a ocorrência está bem classificada, o vetor dos pesos não precisa de ser atualizado (faz-se $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)}$).
- Se a ocorrência está mal classificada, o vetor dos pesos deve ser atualizado por forma a que
  - $y^n \left(\tilde{w}_{(t+1)} \cdot \tilde{x}^n\right) > 0$
  - ou pelo menos que $y^n \left(\tilde{w}_{(t+1)} \cdot \tilde{x}^n\right)$ seja menos negativo do que $y^n \left(\tilde{w}_{(t)} \cdot \tilde{x}^n\right)$,

Perceptron | regra de aprendizagem (iii)

- Se a ocorrência está bem classificada, o vetor dos pesos não precisa de ser atualizado (faz-se $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)}$).
- Se a ocorrência está mal classificada, o vetor dos pesos deve ser atualizado por forma a que
  - $y^n \left(\tilde{w}_{(t+1)} \cdot \tilde{x}^n\right) > 0$
  - ou pelo menos que $y^n \left(\tilde{w}_{(t+1)} \cdot \tilde{x}^n\right)$ seja menos negativo do que $y^n \left(\tilde{w}_{(t)} \cdot \tilde{x}^n\right)$,
  - ou seja, pretende-se que

$$y^n \left(\tilde{w}_{(t+1)} \cdot \tilde{x}^n\right) > \underbrace{y^n (\tilde{w}_{(t)} \cdot \tilde{x}^n)}_{\leqslant 0}.$$

Perceptron | regra de aprendizagem (iv)

- Como

$$\tilde{x}^n \cdot \tilde{x}^n = (1, x_1, \ldots, x_I) \cdot (1, x_1, \ldots, x_I) = 1 + x_1^2 + \cdots x_I^2 > 0,$$

tem-se que

$$\underbrace{y^n y^n}_{=1} \underbrace{(\tilde{x}^n \cdot \tilde{x}^n)}_{>0} > 0,$$

pelo que

$$y^n(\tilde{w}_{(t)} \cdot \tilde{x}^n) + y^n y^n(\tilde{x}^n \cdot \tilde{x}^n) > y^n(\tilde{w}_{(t)} \cdot \tilde{x}^n) \Leftrightarrow$$
$$y^n\left(\left(\tilde{w}_{(t)} + y^n \tilde{x}^n\right) \cdot \tilde{x}^n\right) > y^n(\tilde{w}_{(t)} \cdot \tilde{x}^n).$$

Perceptron | regra de aprendizagem (iv)

- Como

$$\tilde{x}^n \cdot \tilde{x}^n = (1, x_1, \ldots, x_I) \cdot (1, x_1, \ldots, x_I) = 1 + x_1^2 + \cdots x_I^2 > 0,$$

tem-se que

$$\underbrace{y^n y^n}_{=1} \underbrace{(\tilde{x}^n \cdot \tilde{x}^n)}_{>0} > 0,$$

pelo que

$$y^n(\tilde{w}_{(t)} \cdot \tilde{x}^n) + y^n y^n(\tilde{x}^n \cdot \tilde{x}^n) > y^n(\tilde{w}_{(t)} \cdot \tilde{x}^n) \Leftrightarrow$$
$$y^n\left(\left(\tilde{w}_{(t)} + y^n \tilde{x}^n\right) \cdot \tilde{x}^n\right) > y^n(\tilde{w}_{(t)} \cdot \tilde{x}^n).$$

- Fazendo $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)} + y^n \tilde{x}^n$, tem-se, como pretendido, que

$$y^n\left(\tilde{w}_{(t+1)} \cdot \tilde{x}^n\right) > y^n(\tilde{w}_{(t)} \cdot \tilde{x}^n).$$

Algoritmo Perc–v1

Input: $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^I$, $y^n \in \{-1, +1\}$, $\tilde{w}_{(0)} \in \mathbb{R}^{I+1}$
Output: $\tilde{w}^* \in \mathbb{R}^{I+1}$

$t \leftarrow 0$;
while V **do**
  for $n \leftarrow 1$ **to** N **do**
    $E_{(t)} \leftarrow \frac{1}{N} \sum_{p=1}^{N} \frac{1}{2} \left| y^p - \underbrace{\operatorname{sgn}\left(\tilde{w}_{(t)} \cdot \tilde{x}^p\right)}_{\hat{y}^p} \right|$;
    if $E_{(t)} = 0$ **then**
      $\tilde{w}^* \leftarrow \tilde{w}_{(t)}$;
      return $\tilde{w}^*$;
    else
      $\hat{y}^n \leftarrow \operatorname{sgn}\left(\tilde{w}_{(t)} \cdot \tilde{x}^n\right)$;
      if $y^n \neq \hat{y}^n$ **then**
        $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)} + y^n \tilde{x}^n$;
      else
        $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)}$;
      $t \leftarrow t + 1$;

Algoritmo Perc–v1 | convergência

- **Teorema.** Seja $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^I$, $y^n \in \{-1, +1\}$, uma base de dados linearmente separável. Então, o algoritmo *Perceptron* v1 termina.

Algoritmo Perc–v1 | convergência

- **Teorema.** Seja $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^I$, $y^n \in \{-1, +1\}$, uma base de dados linearmente separável. Então, o algoritmo *Perceptron* v1 termina.
- Note-se que o algoritmo Perc–v1 não usa técnicas de otimização, não considerando, por exemplo derivadas.

Algoritmo Perc–v1 | convergência

- **Teorema.** Seja $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^I$, $y^n \in \{-1, +1\}$, uma base de dados linearmente separável. Então, o algoritmo *Perceptron* v1 termina.
- Note-se que o algoritmo Perc–v1 não usa técnicas de otimização, não considerando, por exemplo derivadas.
- Se a base de dados é linearmente separável, então há uma infinidade de soluções (umas melhores do que outras). O algoritmo Perc–v1 encontra uma delas em que o único critério é o valor da função custo ser 0.

Ex1 (AND) | D

- Base de dados “AND”

Ex1 (AND) | D

- Base de dados "AND" : $D = (x^n, y^n)_{n=1}^4$ com

Ex1 (AND) | D

- Base de dados "AND" : $D = (x^n, y^n)_{n=1}^4$ com

$x^1 = (0,0)^\top \quad y^1 = -1$ (F) $\qquad x^2 = (0,1)^\top \quad y^2 = -1$ (F)

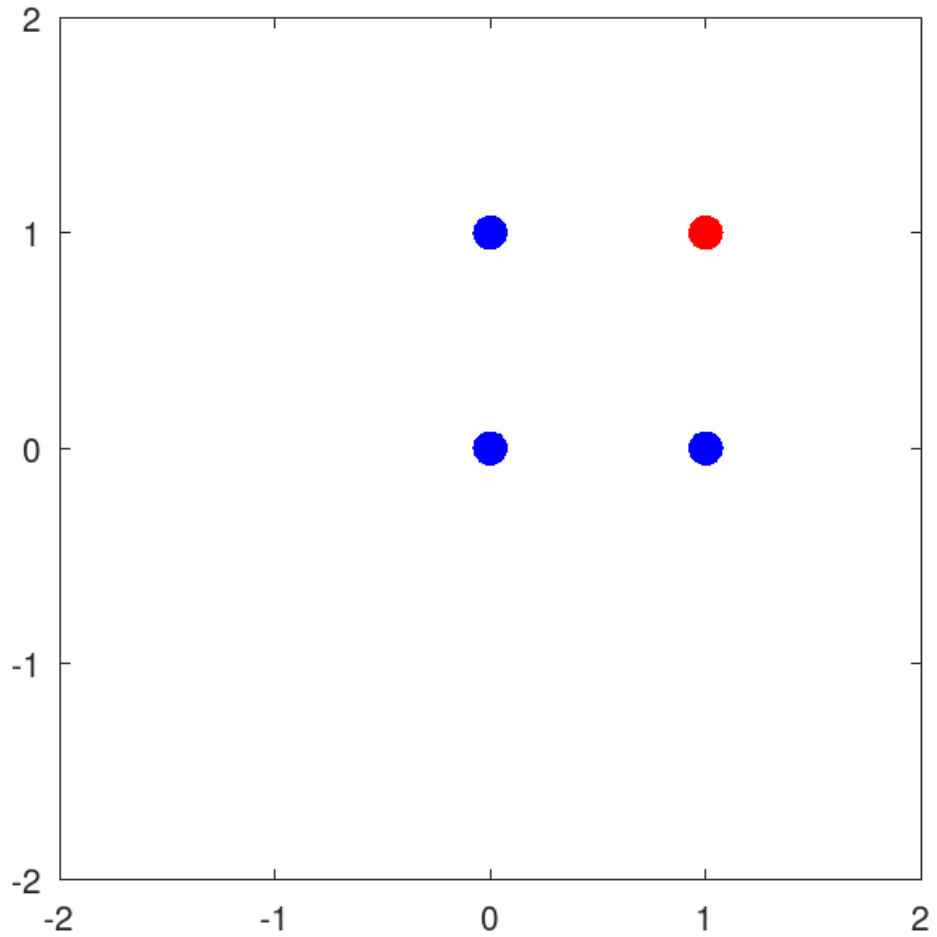
$x^3 = (1,0)^\top \quad y^3 = -1$ (F) $\qquad x^4 = (1,1)^\top \quad y^4 = +1$ (V)

Ex1 (AND) | D

- Base de dados "AND" : $D = (x^n, y^n)_{n=1}^4$ com

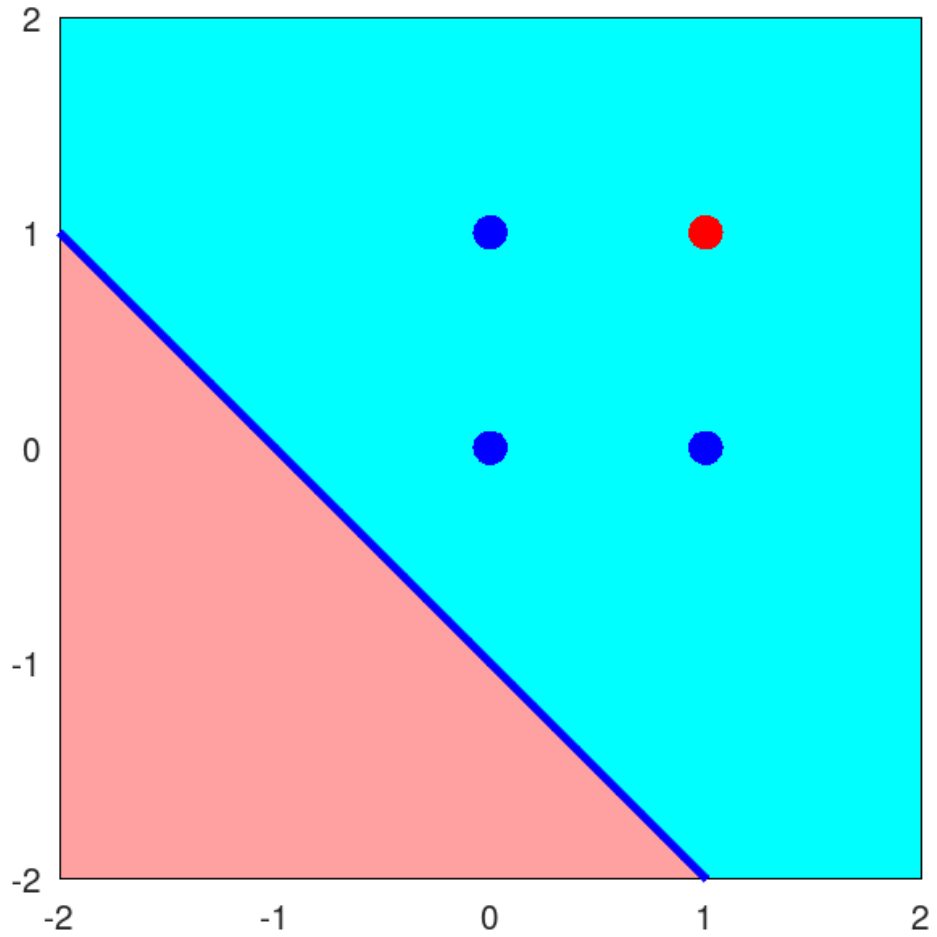
$x^1 = (0,0)^\top \quad y^1 = -1$ (F) $\qquad x^2 = (0,1)^\top \quad y^2 = -1$ (F)

$x^3 = (1,0)^\top \quad y^3 = -1$ (F) $\qquad x^4 = (1,1)^\top \quad y^4 = +1$ (V)



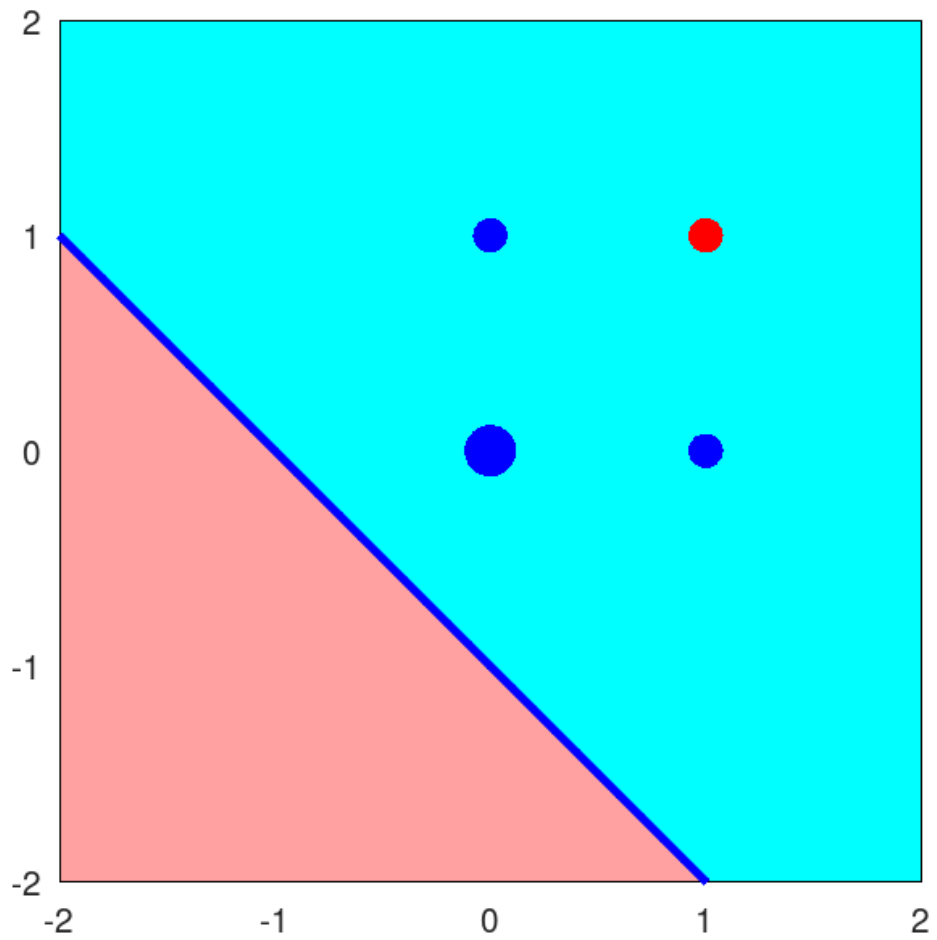
Ex1 (AND) | Perc–v1 | $t = 0$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top$, $E_{(0)} = 0.25$



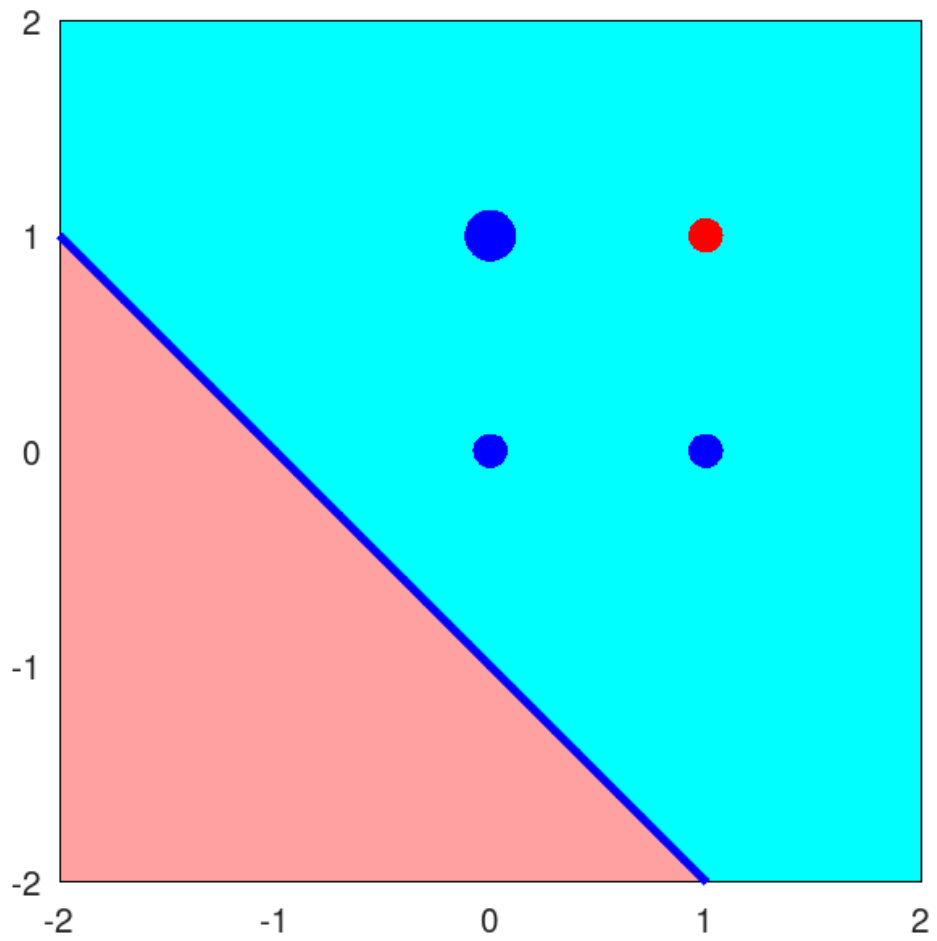
Ex1 (AND) | Perc–v1 | $t = 1$

- $n = 1, \tilde{w}_{(1)} = (-0.25, -0.25, -0.25)^\top, E_{(1)} = 0.25$



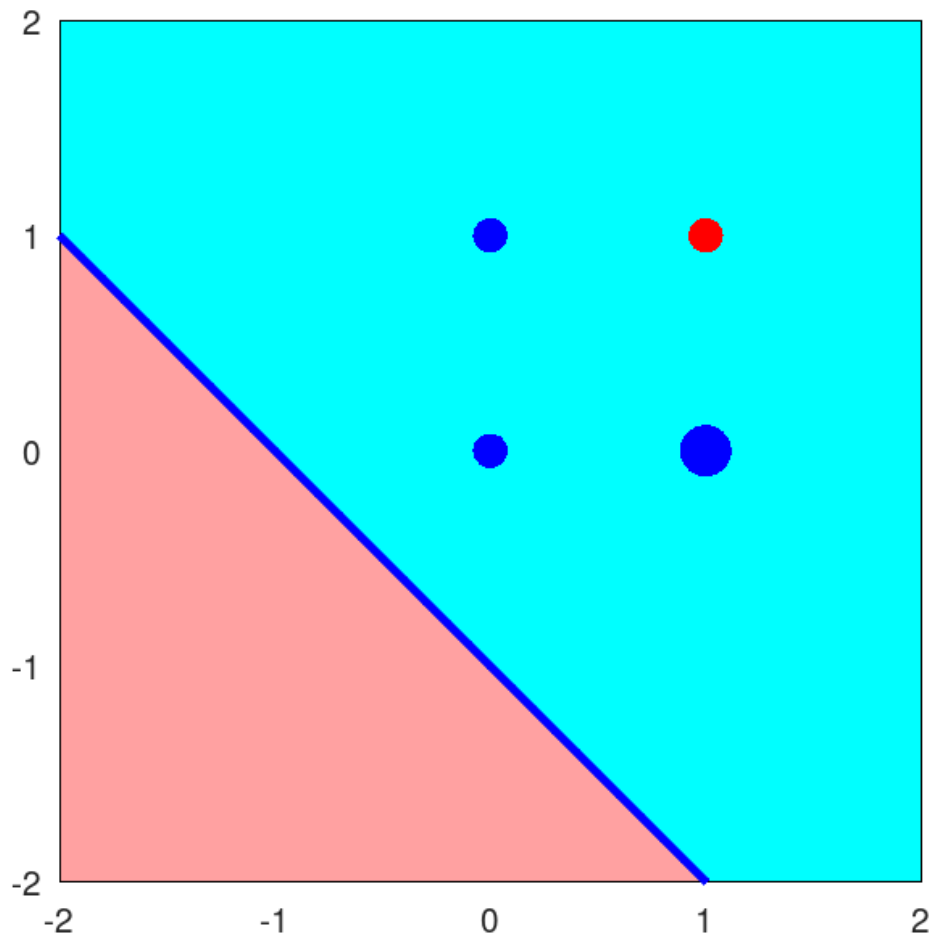
Ex1 (AND) | Perc–v1 | $t = 2$

- $n = 2, \tilde{w}_{(2)} = (-0.25, -0.25, -0.25)^\top, E_{(2)} = 0.25$



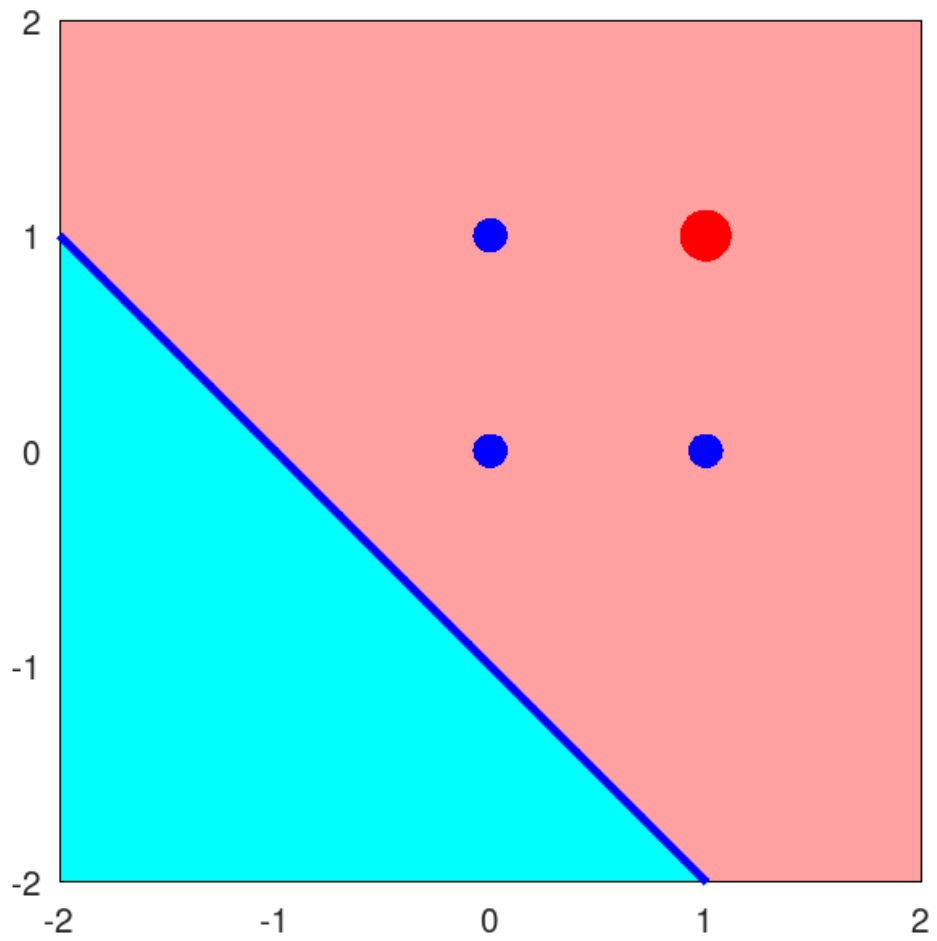
Ex1 (AND) | Perc–v1 | $t = 3$

- $n = 3, \tilde{w}_{(3)} = (-0.25, -0.25, -0.25)^\top, E_{(3)} = 0.25$



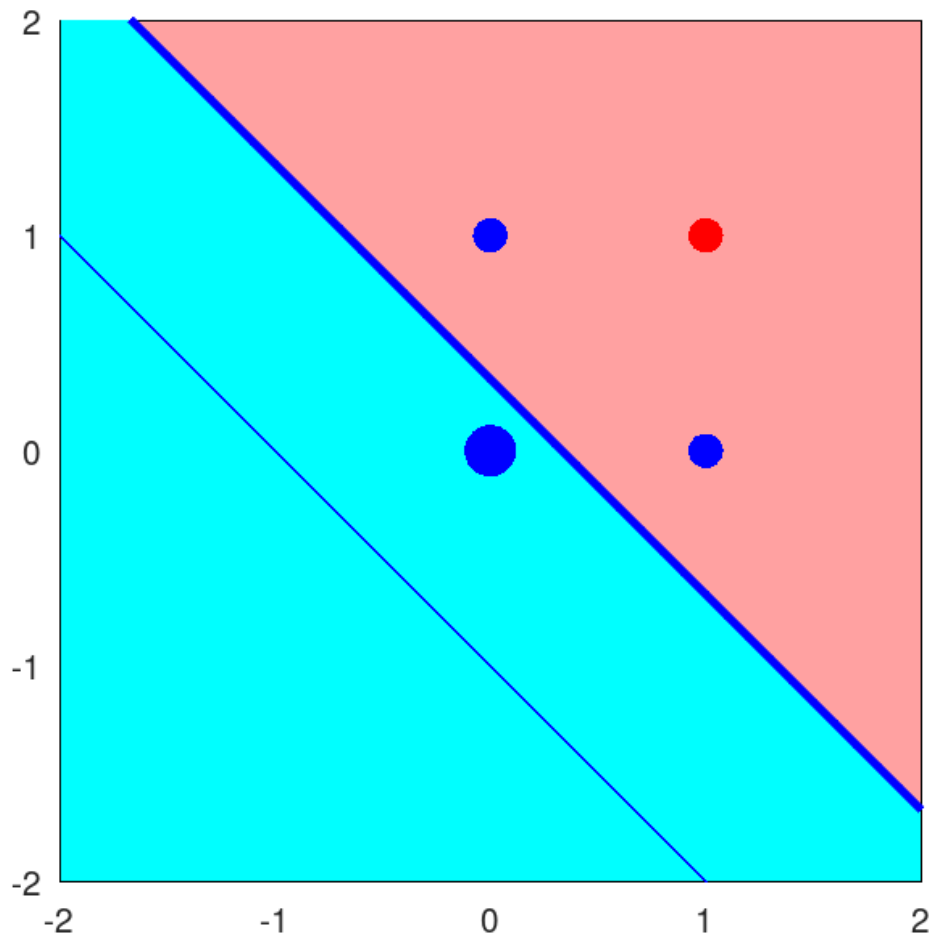
Ex1 (AND) | Perc–v1 | $t = 4$

- $n = 4$, $\tilde{w}_{(4)} = (0.75, 0.75, 0.75)^\top$, $E_{(4)} = 0.75$



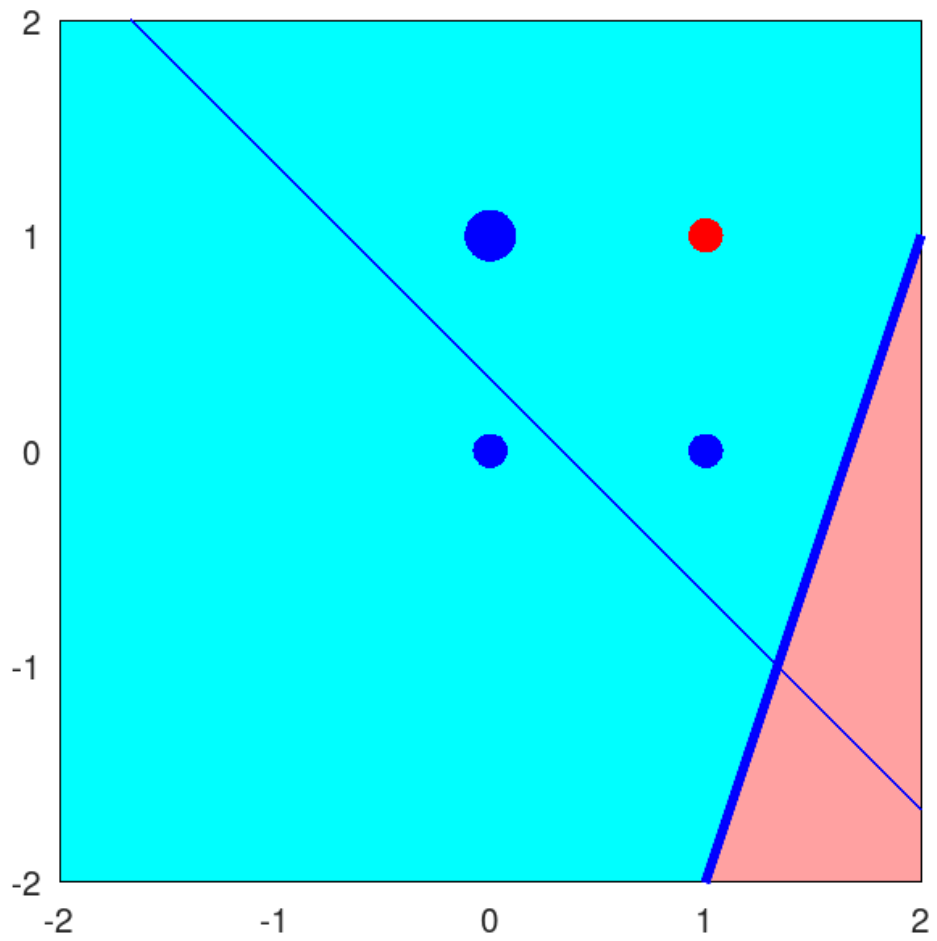
Ex1 (AND) | Perc–v1 | $t = 5$

- $n = 1, \tilde{w}_{(5)} = (-0.25, 0.75, 0.75)^\top, E_{(5)} = 0.5$



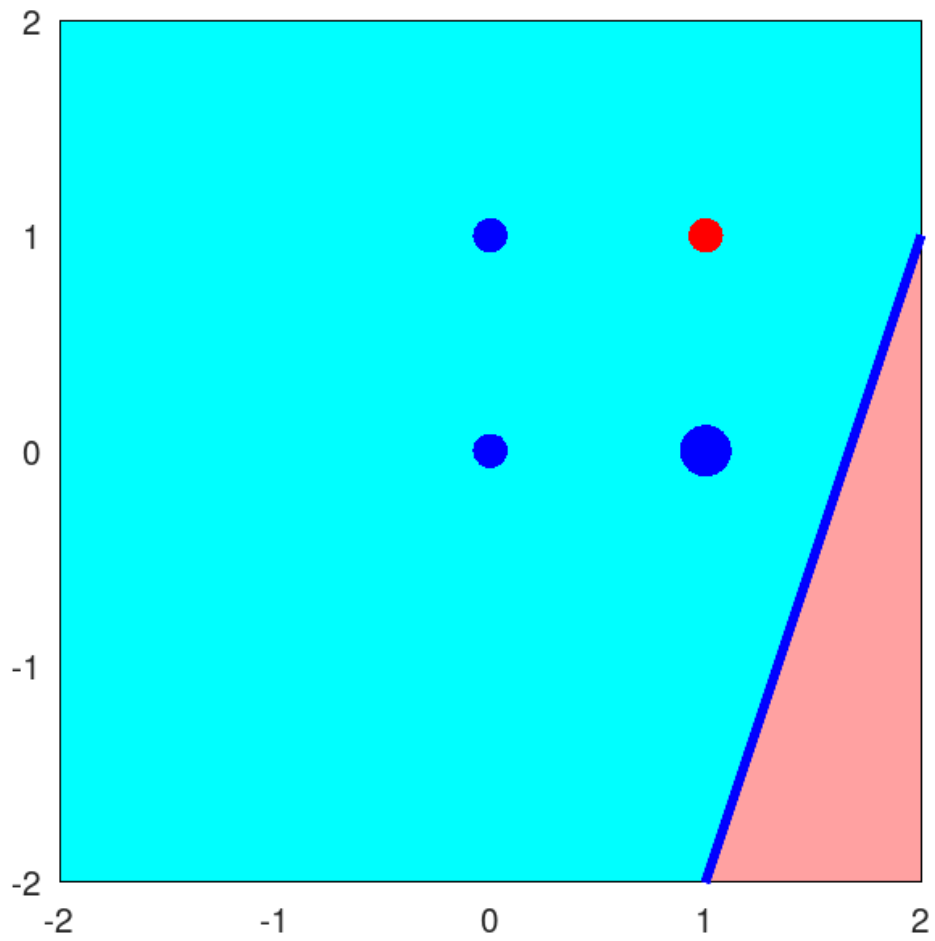
Ex1 (AND) | Perc–v1 | $t = 6$

- $n = 2, \tilde{w}_{(6)} = (-1.25, 0.75, -0.25)^\top, E_{(6)} = 0.25$



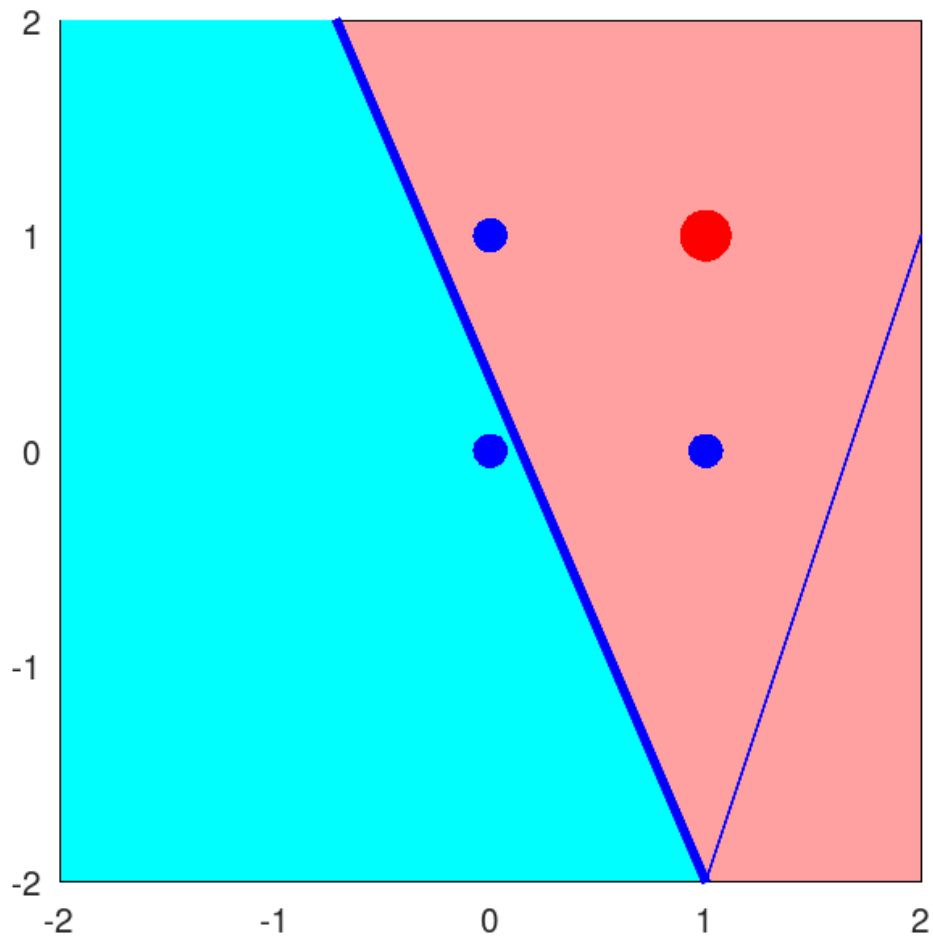
Ex1 (AND) | Perc–v1 | $t = 7$

- $n = 3, \tilde{w}_{(7)} = (-1.25, 0.75, -0.25)^\top, E_{(7)} = 0.25$



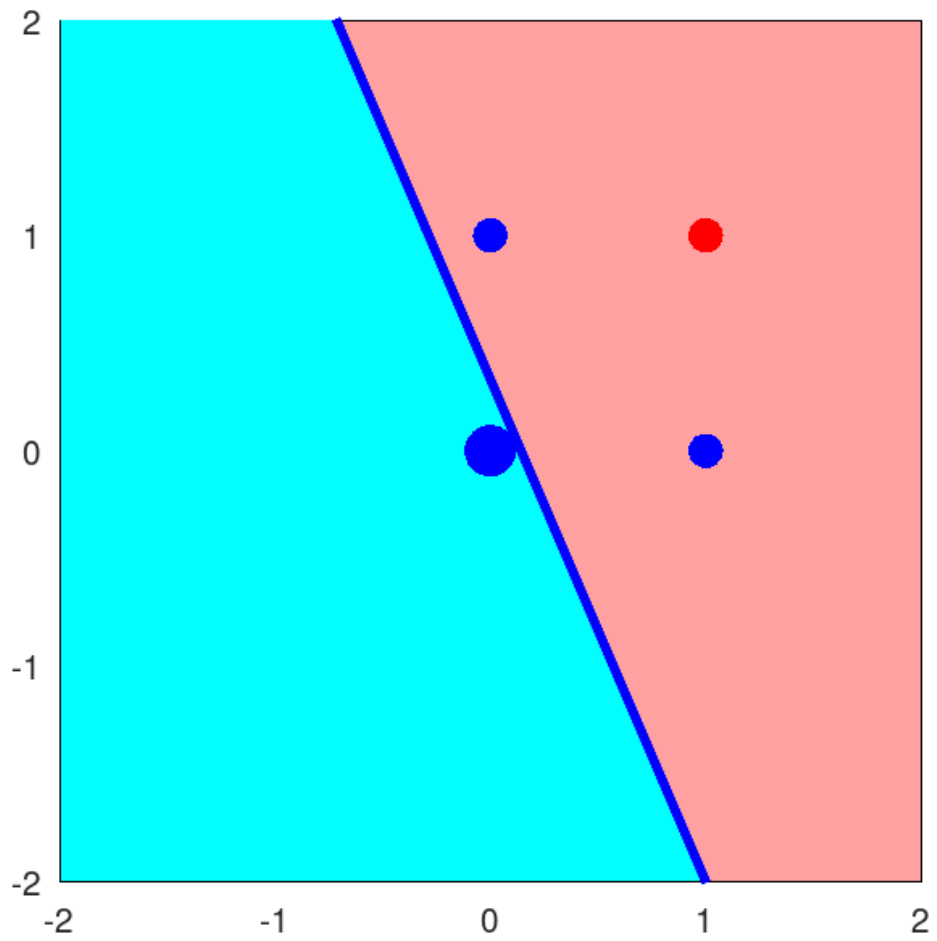
Ex1 (AND) | Perc–v1 | $t = 8$

- $n = 4, \tilde{w}_{(8)} = (-0.25, 1.75, 0.75)^\top, E_{(8)} = 0.5$



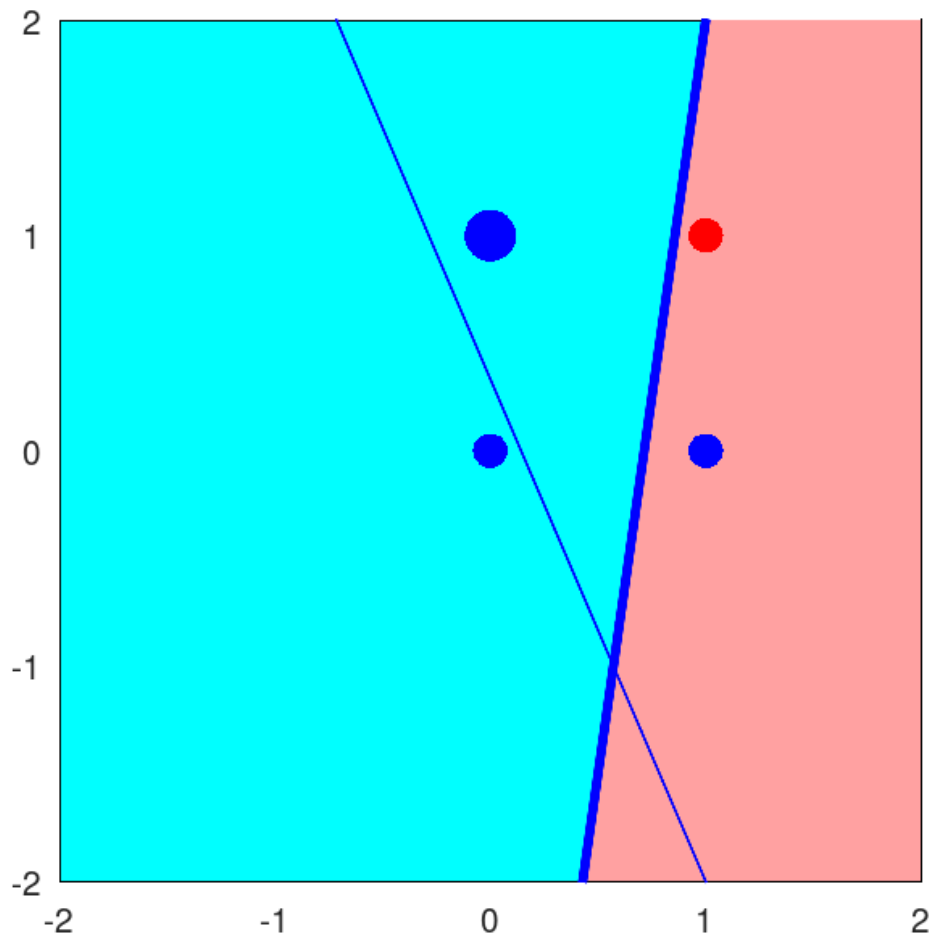
Ex1 (AND) | Perc–v1 | $t = 9$

- $n = 1, \tilde{w}_{(9)} = (-0.25, 1.75, 0.75)^\top, E_{(9)} = 0.5$



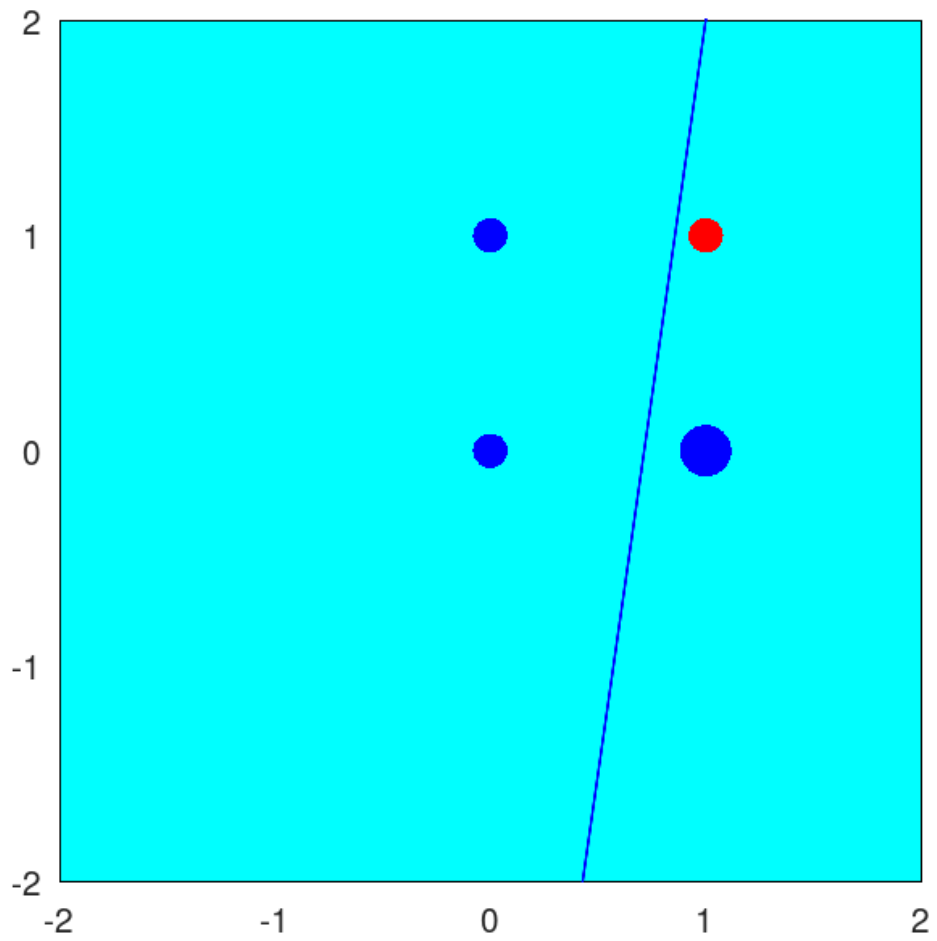
Ex1 (AND) | Perc–v1 | $t = 10$

- $n = 2, \tilde{w}_{(10)} = (-1.25, 1.75, -0.25)^\top, E_{(10)} = 0.25$



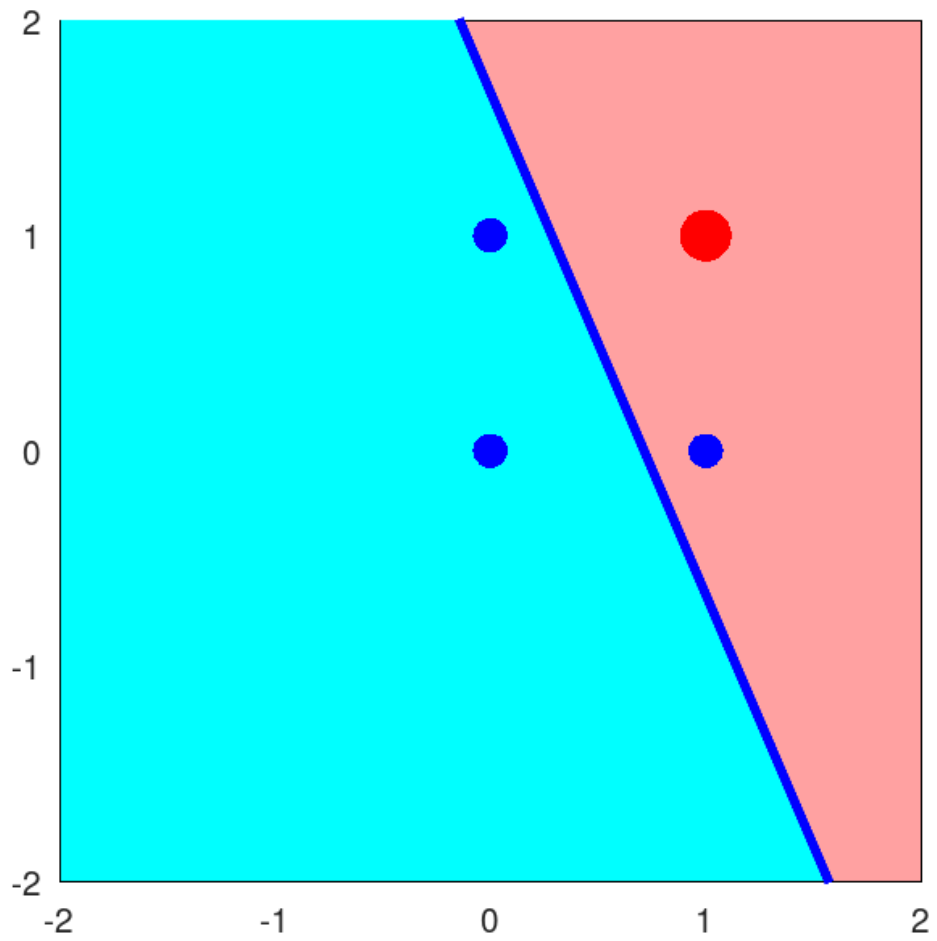
Ex1 (AND) | Perc–v1 | $t = 11$

- $n = 3, \tilde{w}_{(11)} = (-2.25, 0.75, -0.25)^\top, E_{(11)} = 0.25$



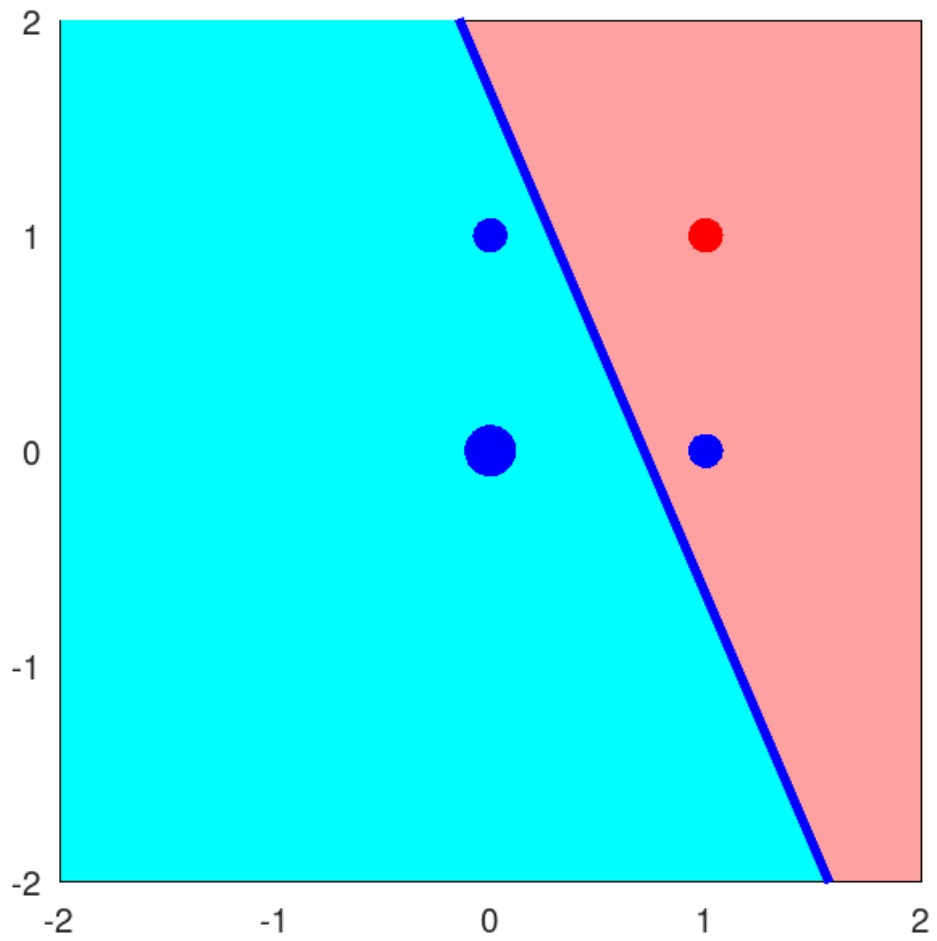
Ex1 (AND) | Perc–v1 | $t = 12$

- $n = 4, \tilde{w}_{(12)} = (-1.25, 1.75, 0.75)^\top, E_{(12)} = 0.25$



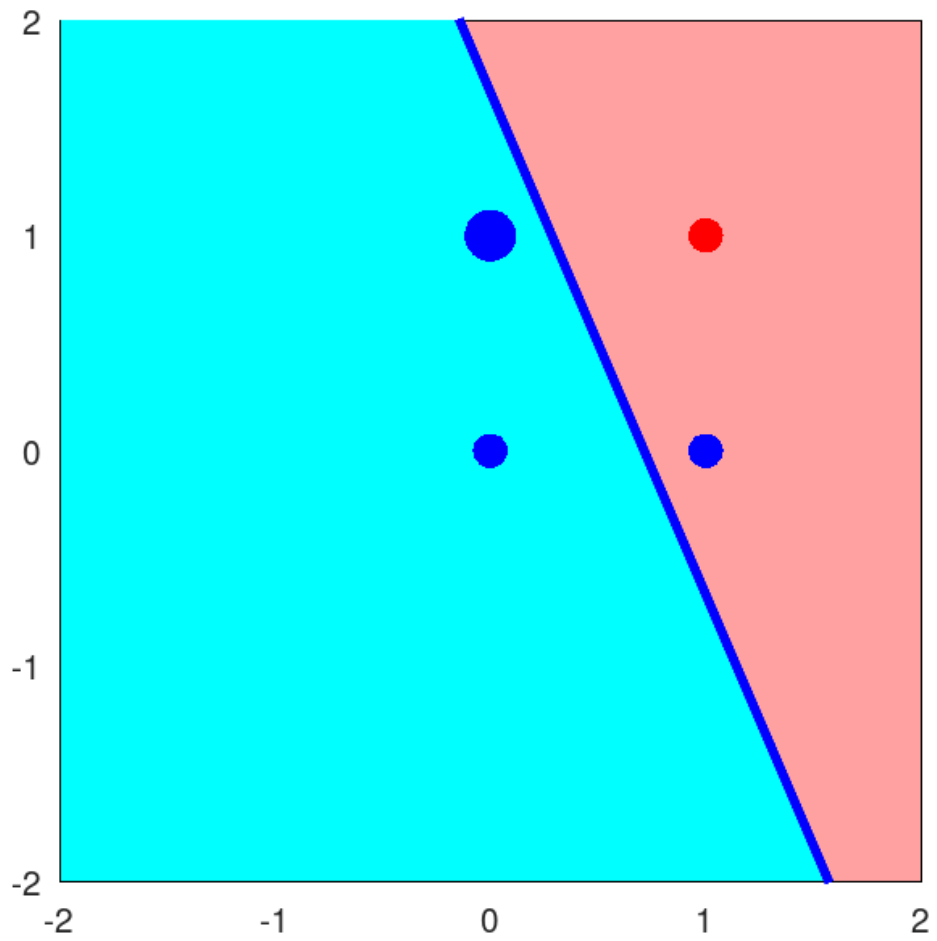
Ex1 (AND) | Perc–v1 | $t = 13$

- $n = 1, \tilde{w}_{(13)} = (-1.25, 1.75, 0.75)^\top, E_{(13)} = 0.25$



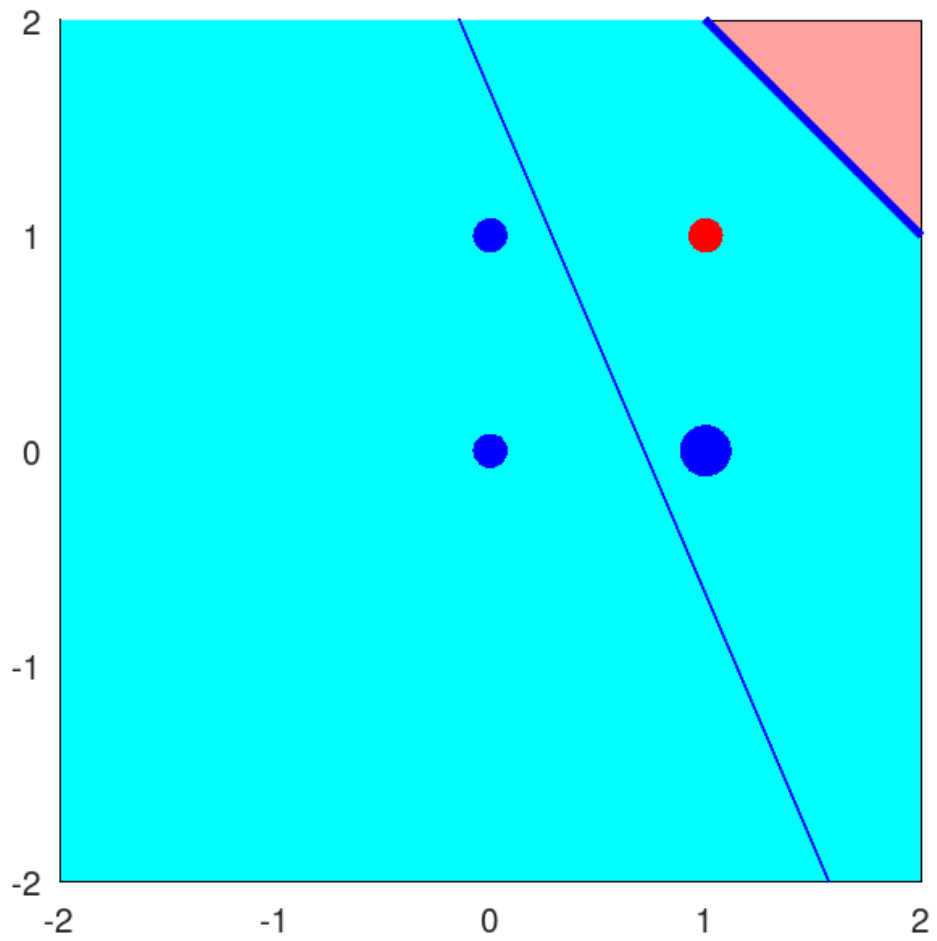
Ex1 (AND) | Perc–v1 | $t = 14$

- $n = 2, \tilde{w}_{(14)} = (-1.25, 1.75, 0.75)^\top, E_{(14)} = 0.25$



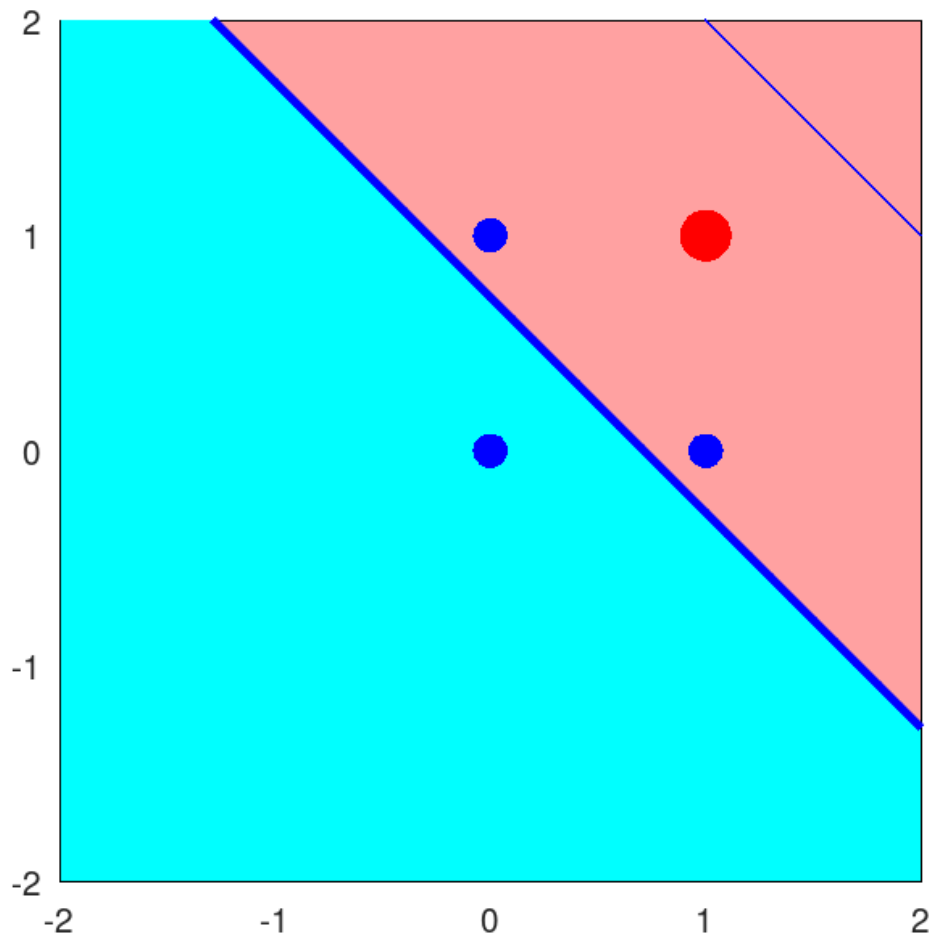
Ex1 (AND) | Perc–v1 | $t = 15$

- $n = 3, \tilde{w}_{(15)} = (-2.25, 0.75, 0.75)^\top, E_{(15)} = 0.25$



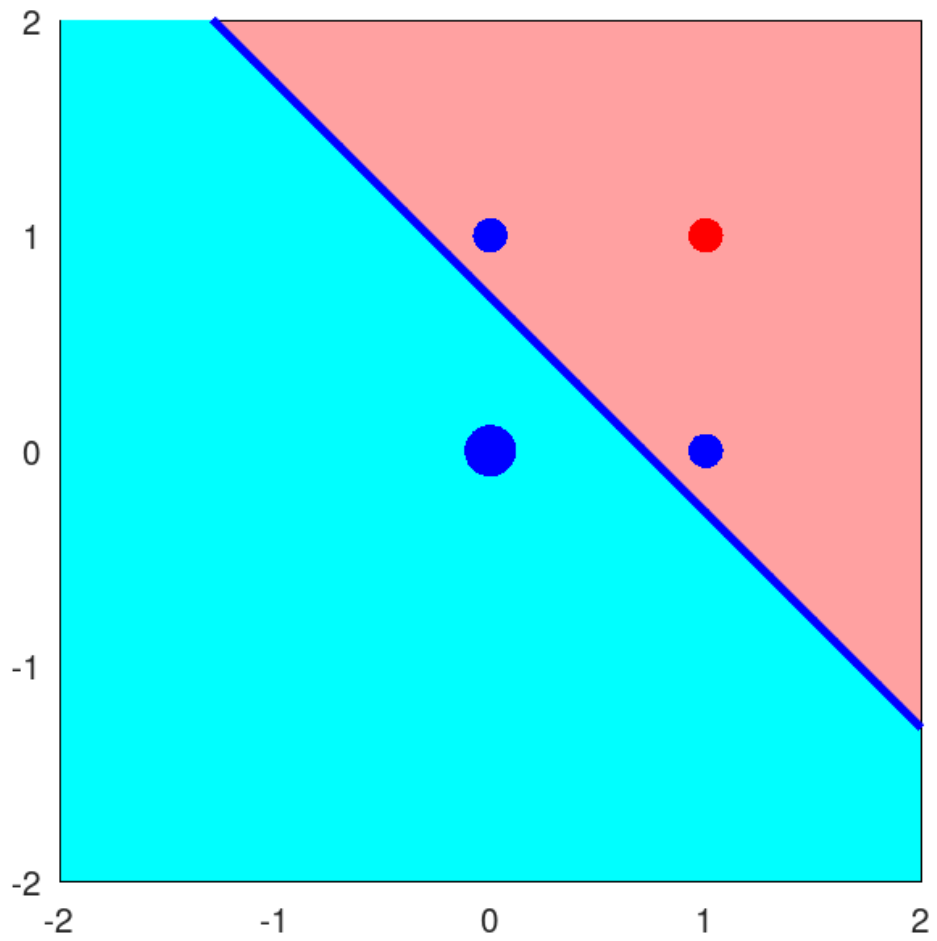
Ex1 (AND) | Perc–v1 | $t = 16$

- $n = 4, \tilde{w}_{(16)} = (-1.25, 1.75, 1.75)^\top, E_{(16)} = 0.5$



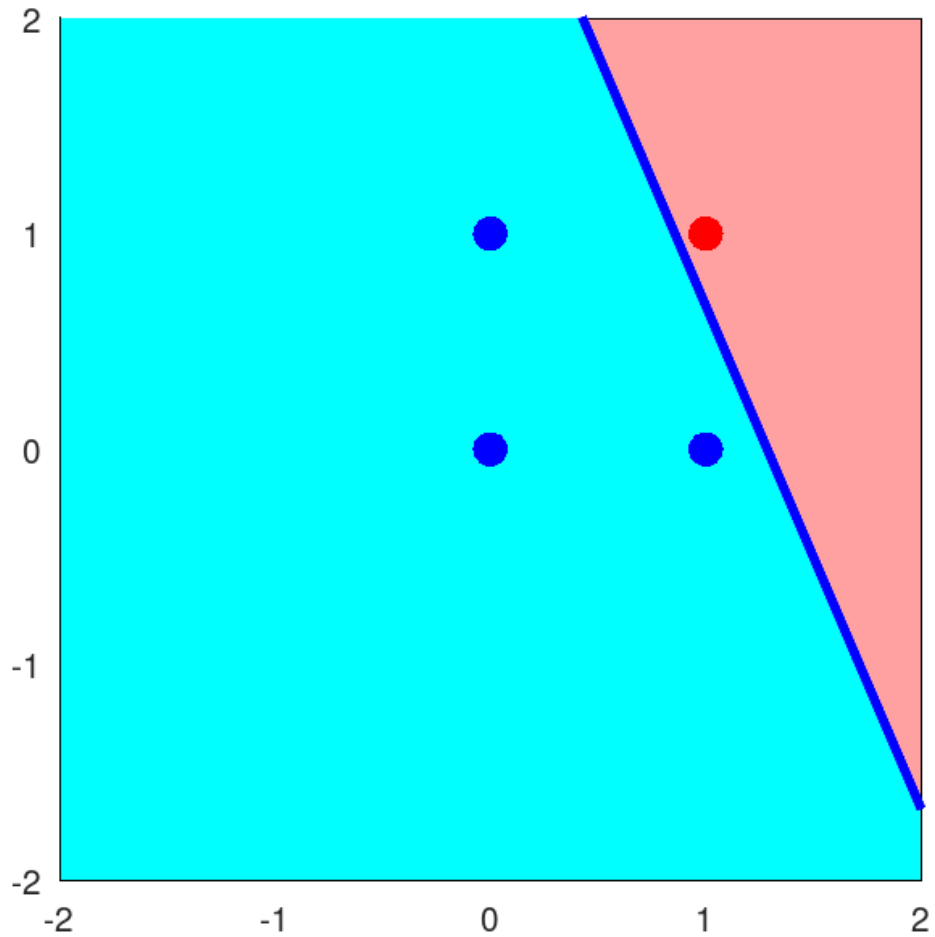
Ex1 (AND) | Perc–v1 | $t = 17$

- $n = 1, \tilde{w}_{(17)} = (-1.25, 1.75, 1.75)^\top, E_{(17)} = 0.5$



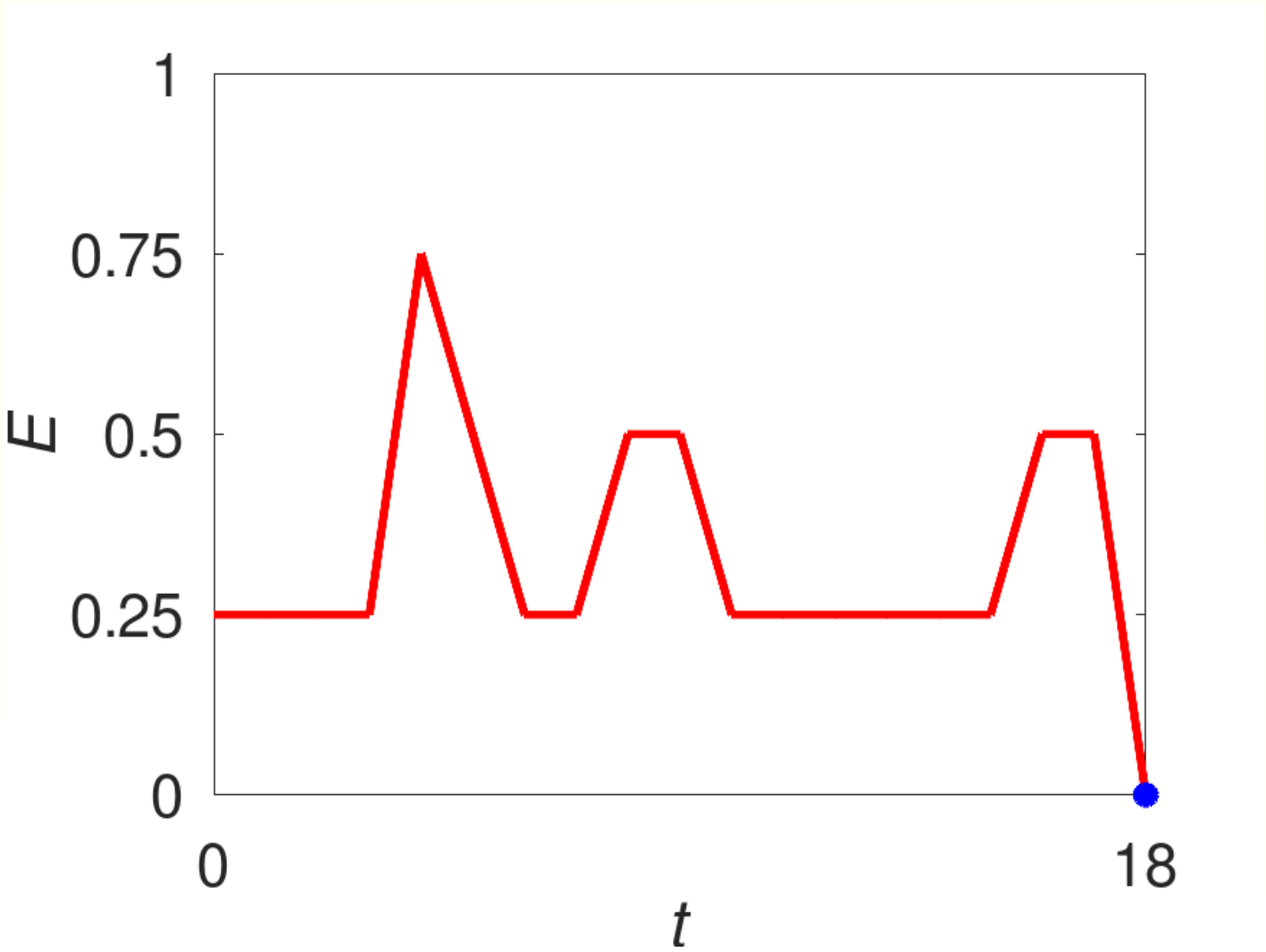
Ex1 (AND) | Perc–v1 | $t = 18$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top, E_{(0)} = 0.25$
- $\tilde{w}^* = (-2.25, 1.75, 0.75)^\top, E_{(18)} = 0$



Ex1 (AND) | Perc–v1 | E

Ex1 (AND) | Perc–v1 | E



Ex2 (OR) | D

- Base de dados “OR”

Ex2 (OR) | D

- Base de dados "OR" : $D = (x^n, y^n)_{n=1}^4$ com

Ex2 (OR) | D

- Base de dados "OR" : $D = (x^n, y^n)_{n=1}^4$ com

$x^1 = (0,0)^\top \quad y^1 = -1$ (F) $\qquad x^2 = (0,1)^\top \quad y^2 = +1$ (V)

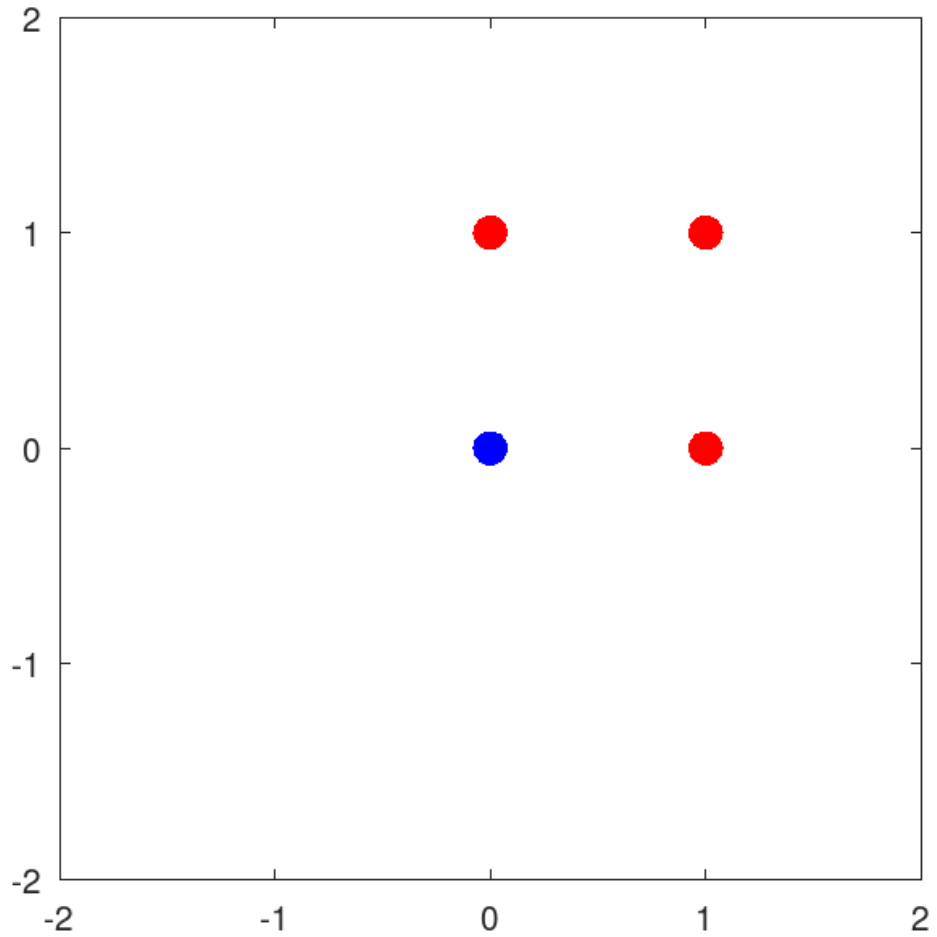
$x^3 = (1,0)^\top \quad y^3 = +1$ (V) $\qquad x^4 = (1,1)^\top \quad y^4 = +1$ (V)

Ex2 (OR) | D

- Base de dados "OR" : $D = (x^n, y^n)_{n=1}^4$ com

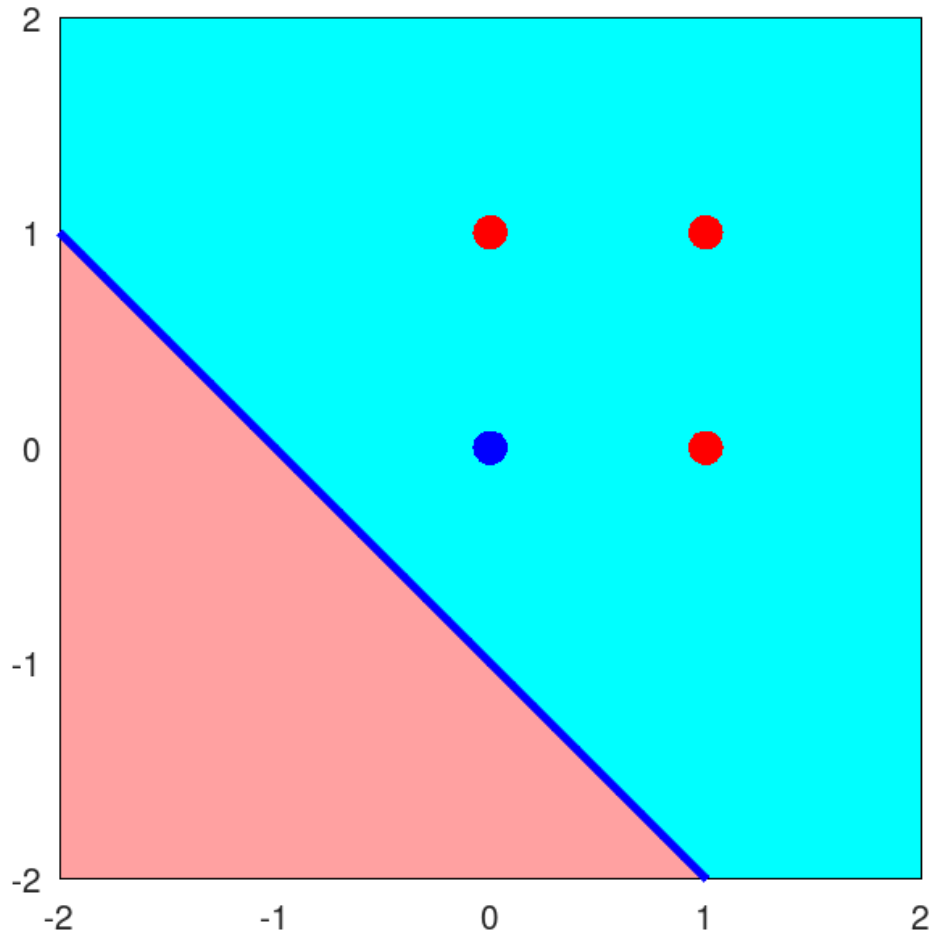
$x^1 = (0,0)^\top \quad y^1 = -1$ (F) $\qquad x^2 = (0,1)^\top \quad y^2 = +1$ (V)

$x^3 = (1,0)^\top \quad y^3 = +1$ (V) $\qquad x^4 = (1,1)^\top \quad y^4 = +1$ (V)



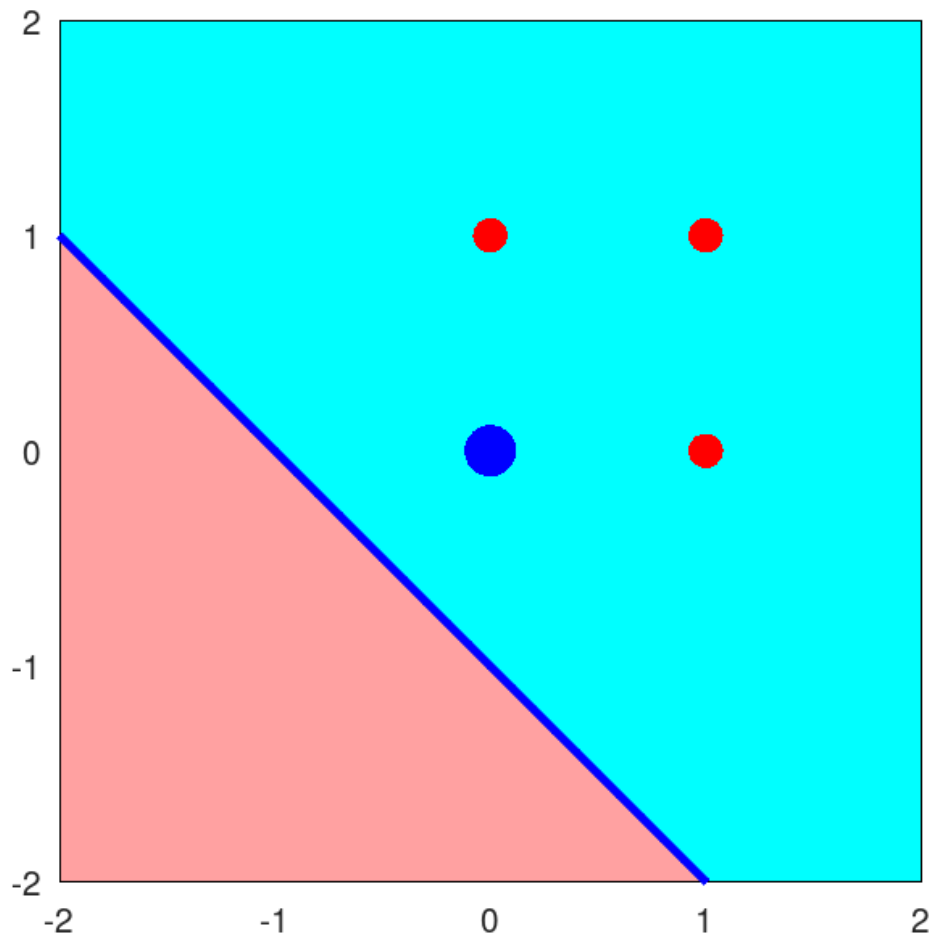
Ex2 (OR) | Perc–v1 | $t = 0$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top$, $E_{(0)} = 0.75$



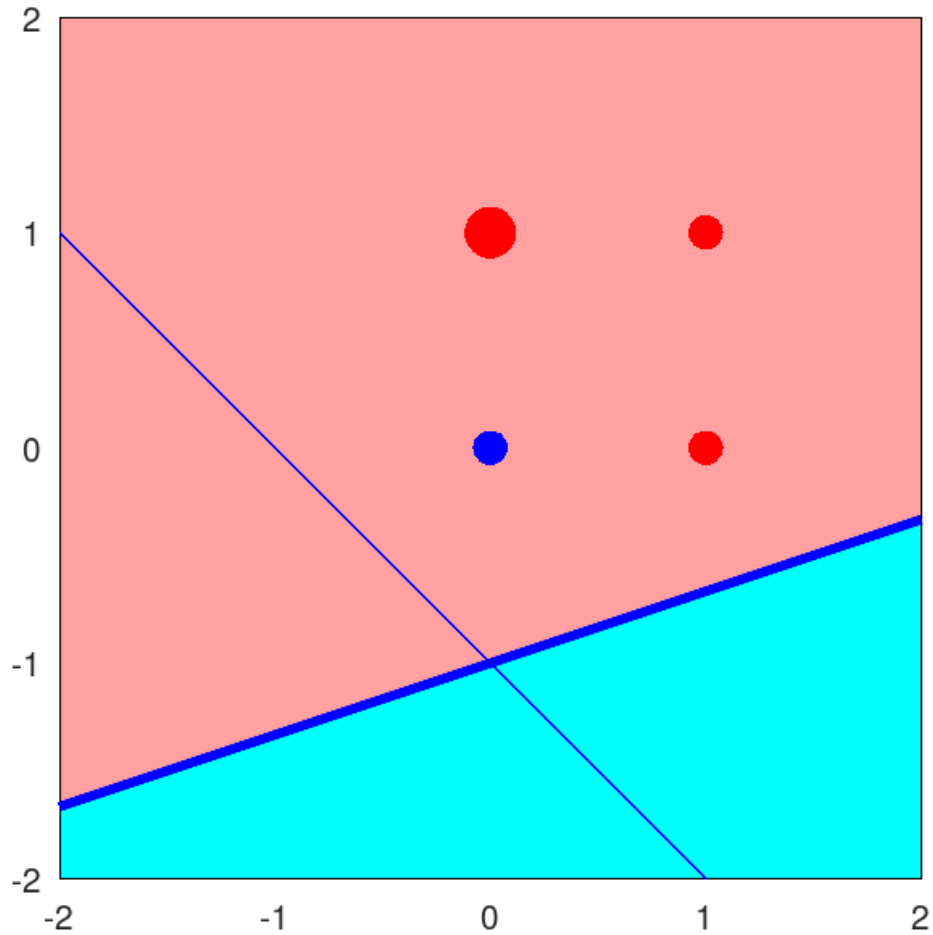
Ex2 (OR) | Perc–v1 | $t = 1$

- $n = 1, \tilde{w}_{(1)} = (-0.25, -0.25, -0.25)^\top, E_{(1)} = 0.75$



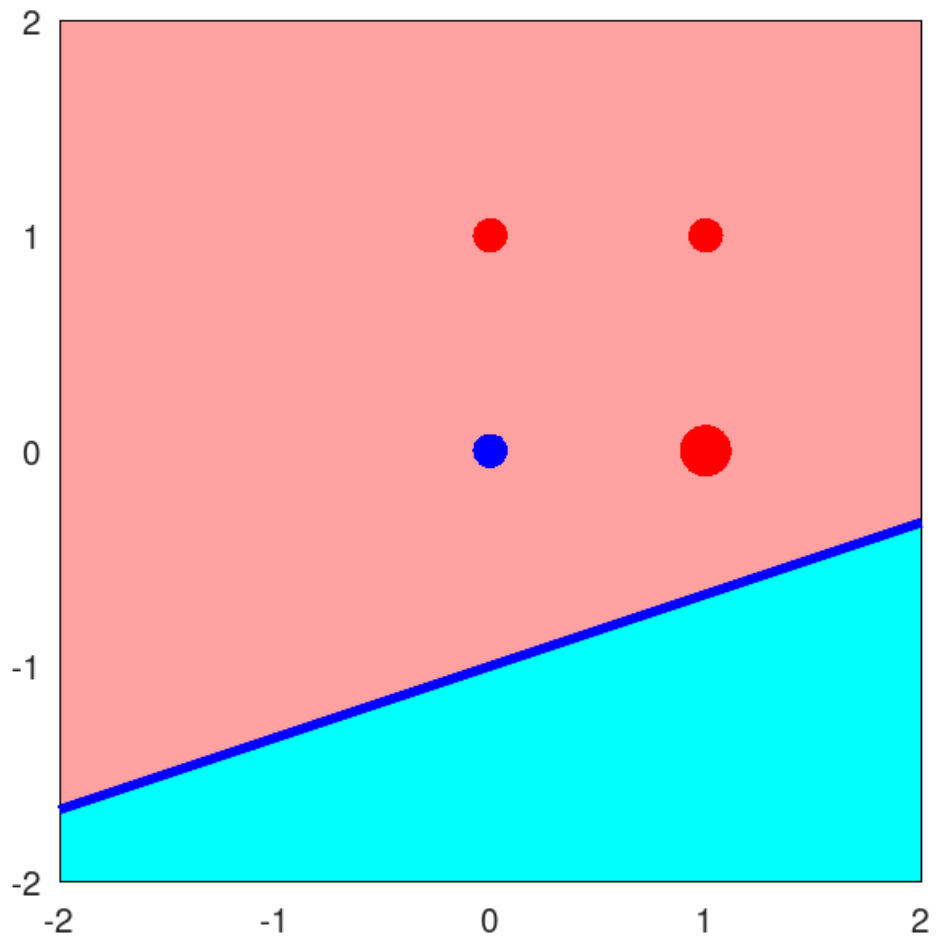
Ex2 (OR) | Perc–v1 | $t = 2$

- $n = 2, \tilde{w}_{(2)} = (0.75, -0.25, 0.75)^\top, E_{(2)} = 0.25$



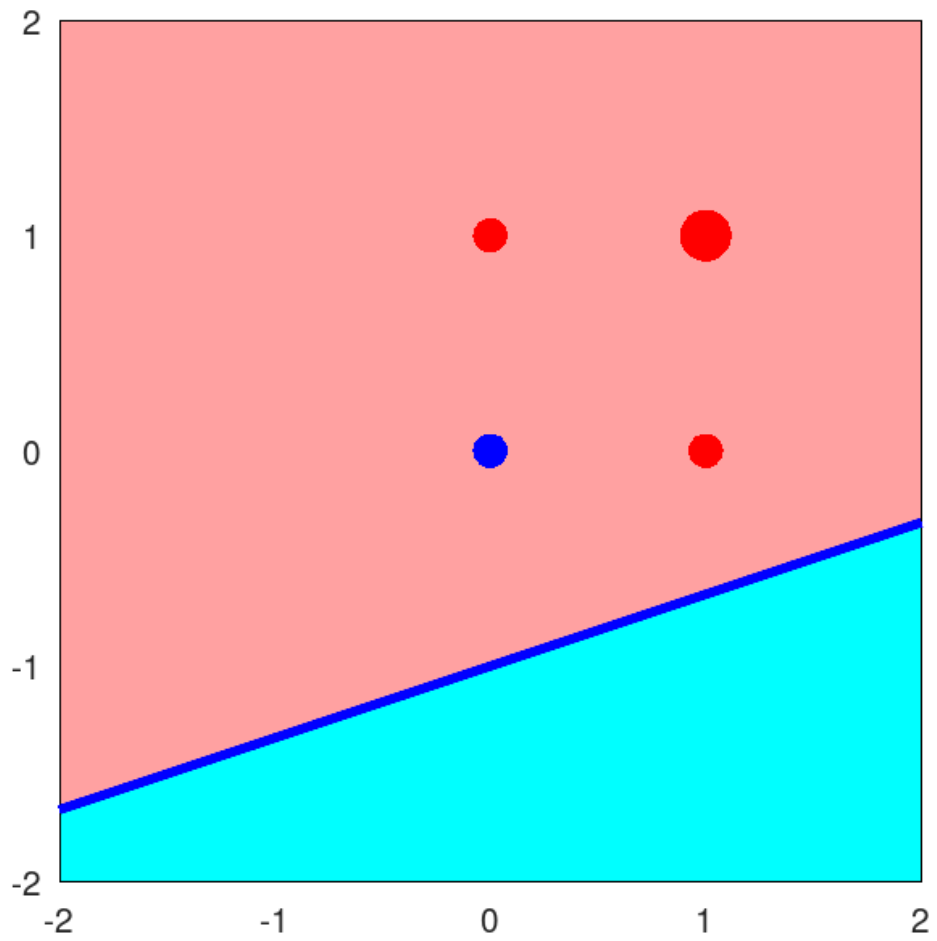
Ex2 (OR) | Perc–v1 | $t = 3$

- $n = 3, \tilde{w}_{(3)} = (0.75, -0.25, 0.75)^\top, E_{(3)} = 0.25$



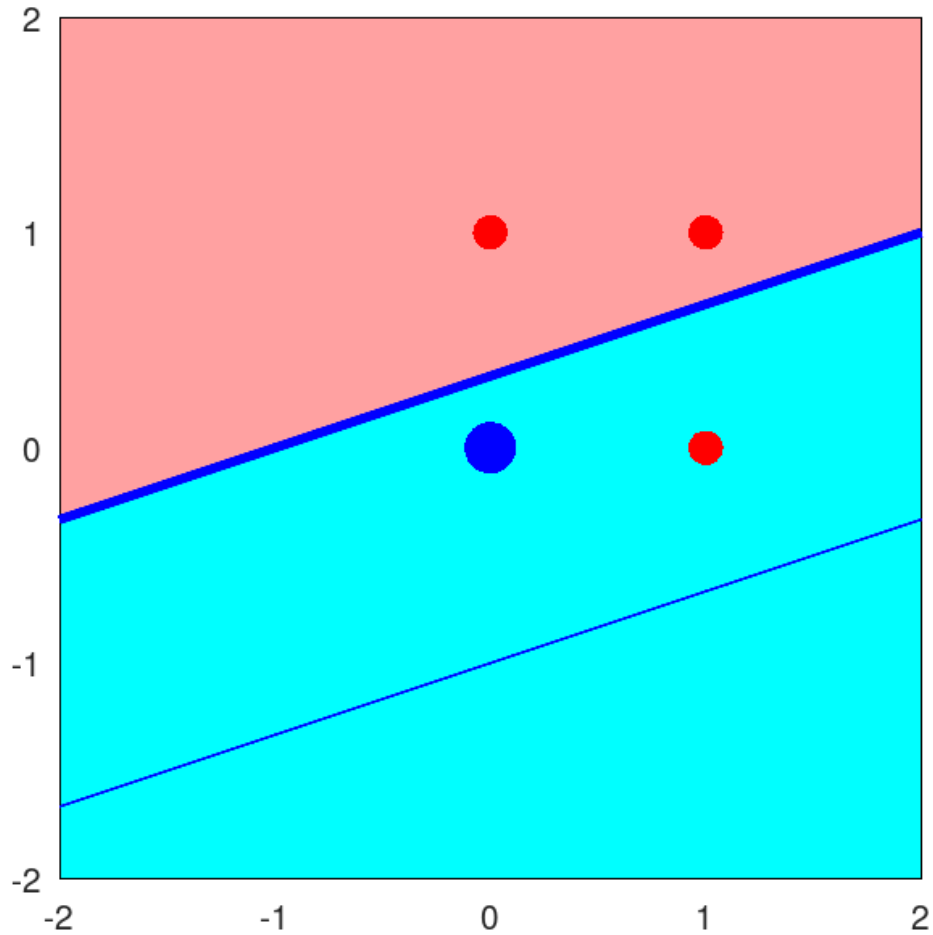
Ex2 (OR) | Perc–v1 | $t = 4$

- $n = 4, \tilde{w}_{(4)} = (0.75, -0.25, 0.75)^\top, E_{(4)} = 0.25$



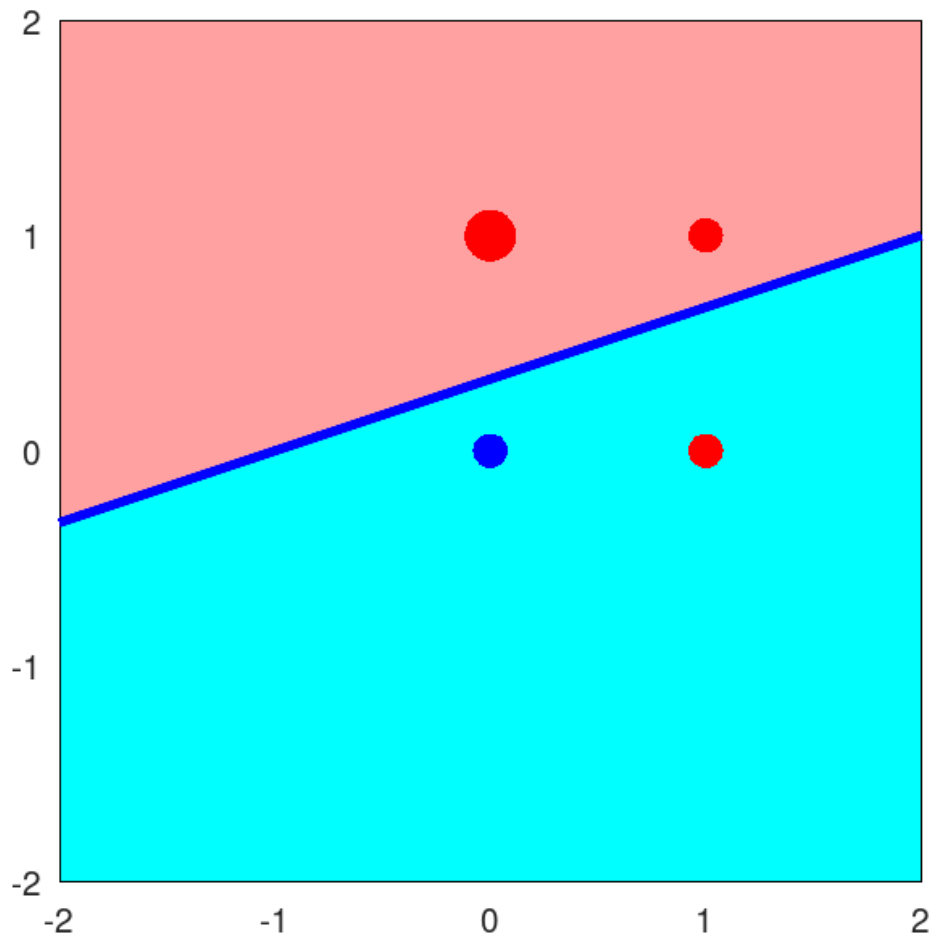
Ex2 (OR) | Perc–v1 | $t = 5$

- $n = 1, \tilde{w}_{(5)} = (-0.25, -0.25, 0.75)^\top, E_{(5)} = 0.25$



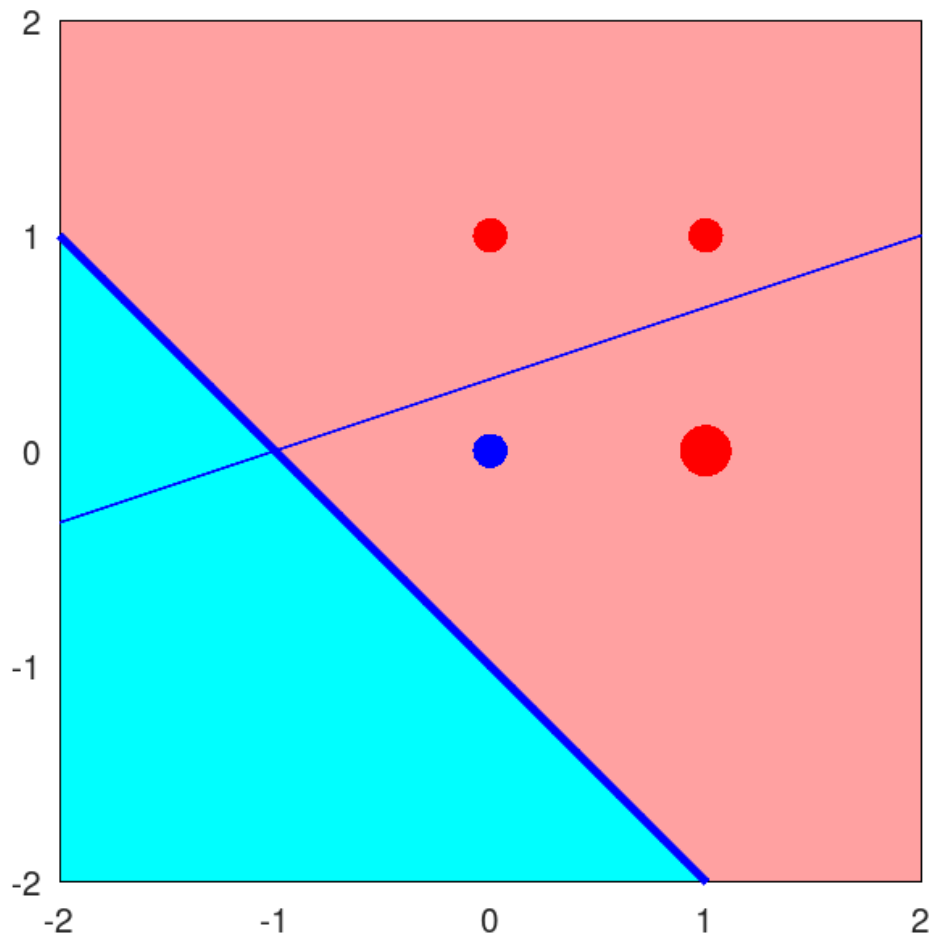
Ex2 (OR) | Perc–v1 | $t = 6$

- $n = 2, \tilde{w}_{(6)} = (-0.25, -0.25, 0.75)^\top, E_{(6)} = 0.25$



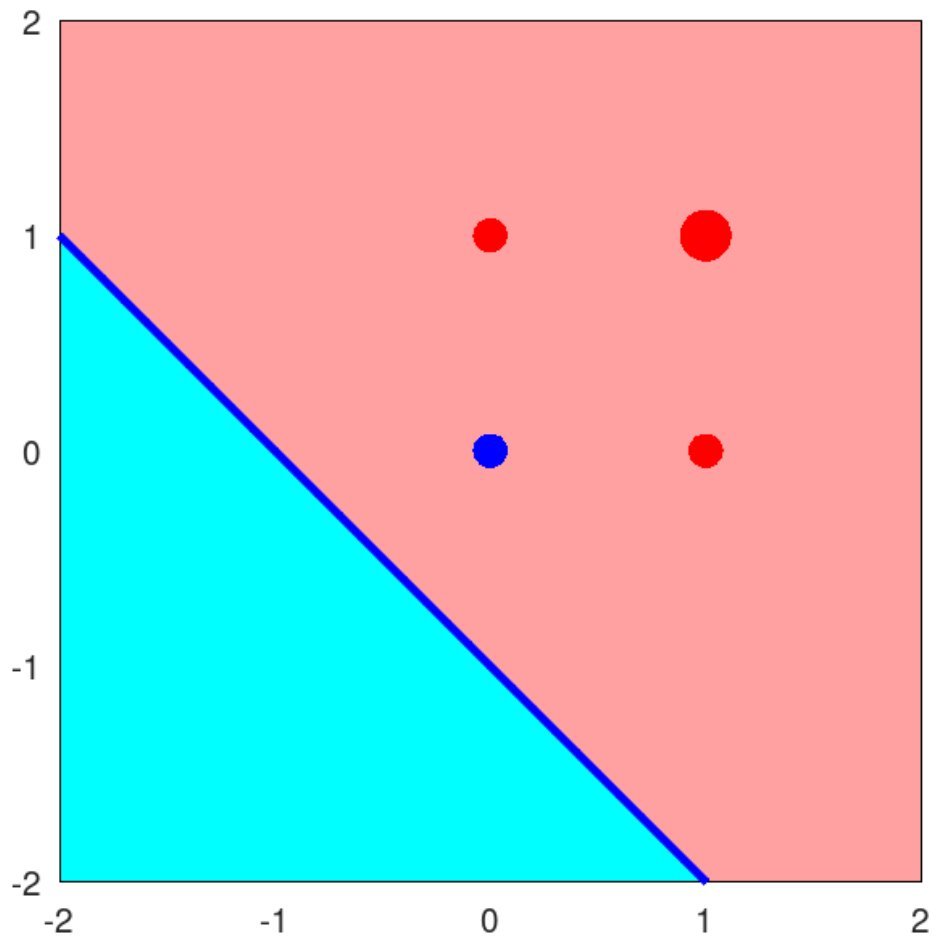
Ex2 (OR) | Perc–v1 | $t = 7$

- $n = 3, \tilde{w}_{(7)} = (0.75, 0.75, 0.75)^\top, E_{(7)} = 0.25$



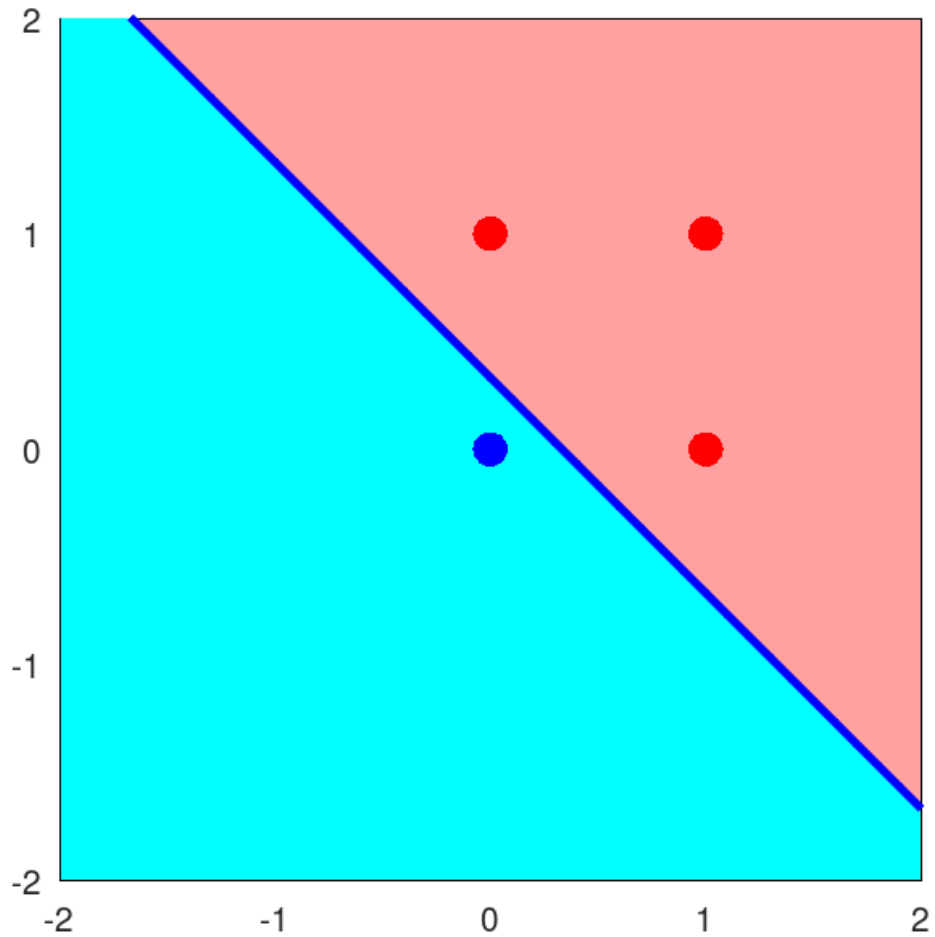
Ex2 (OR) | Perc–v1 | $t = 8$

- $n = 4, \tilde{w}_{(8)} = (0.75, 0.75, 0.75)^\top, E_{(8)} = 0.25$



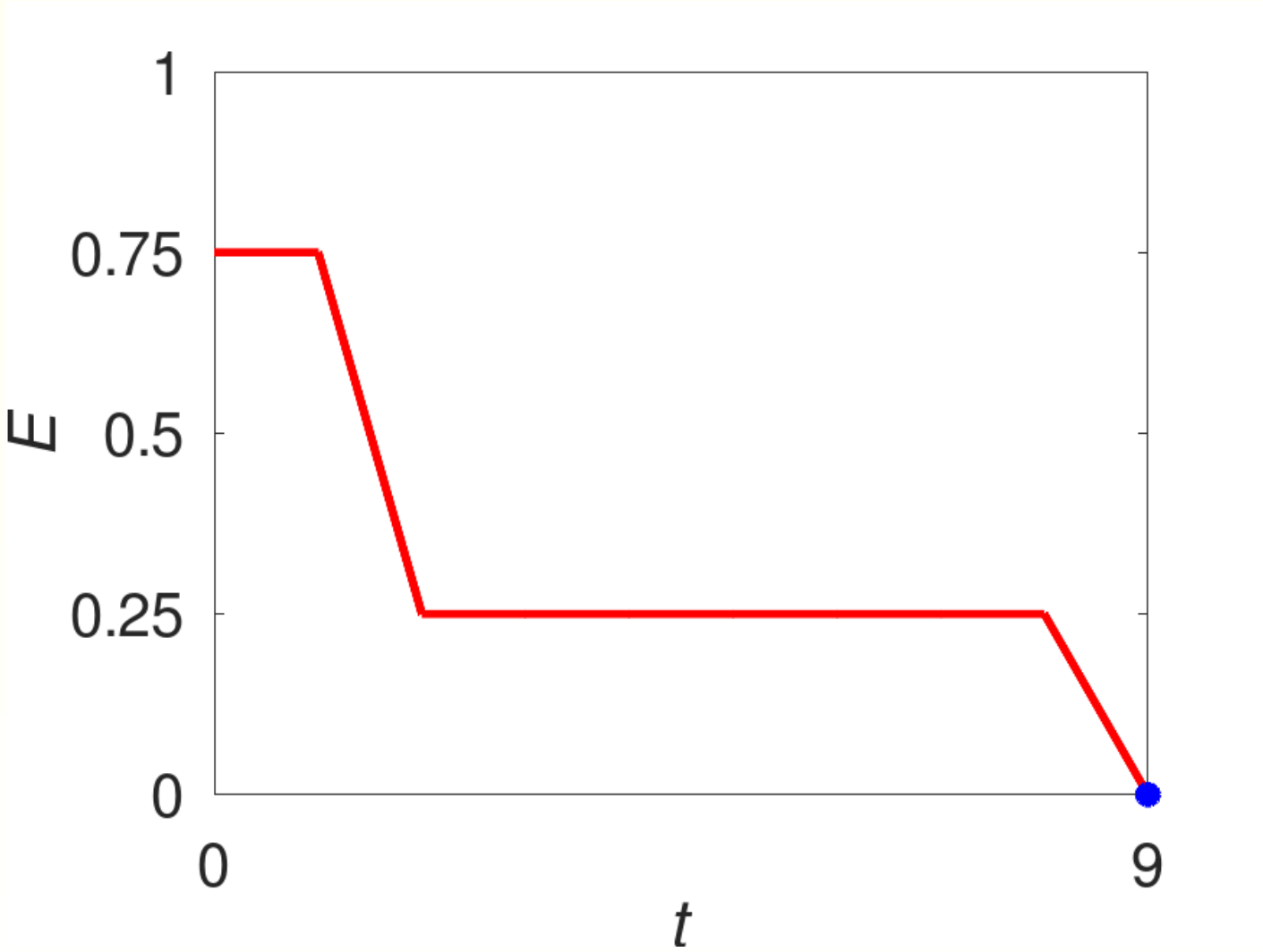
Ex2 (OR) | Perc–v1 | $t = 9$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top, E_{(0)} = 0.75$
- $\tilde{w}^* = (-0.25, 0.75, 0.75)^\top, E_{(9)} = 0$



Ex2 (OR) | Perc–v1 | E

Ex2 (OR) | Perc–v1 | E



Ex3 (XOR) | D

- Base de dados “XOR”

Ex3 (XOR) | D

- Base de dados "XOR" : $D = (x^n, y^n)_{n=1}^4$ com

Ex3 (XOR) | D

- Base de dados "XOR" : $D = (x^n, y^n)_{n=1}^4$ com

$x^1 = (0,0)^\top \quad y^1 = -1$ (F) $\qquad x^2 = (0,1)^\top \quad y^2 = +1$ (V)

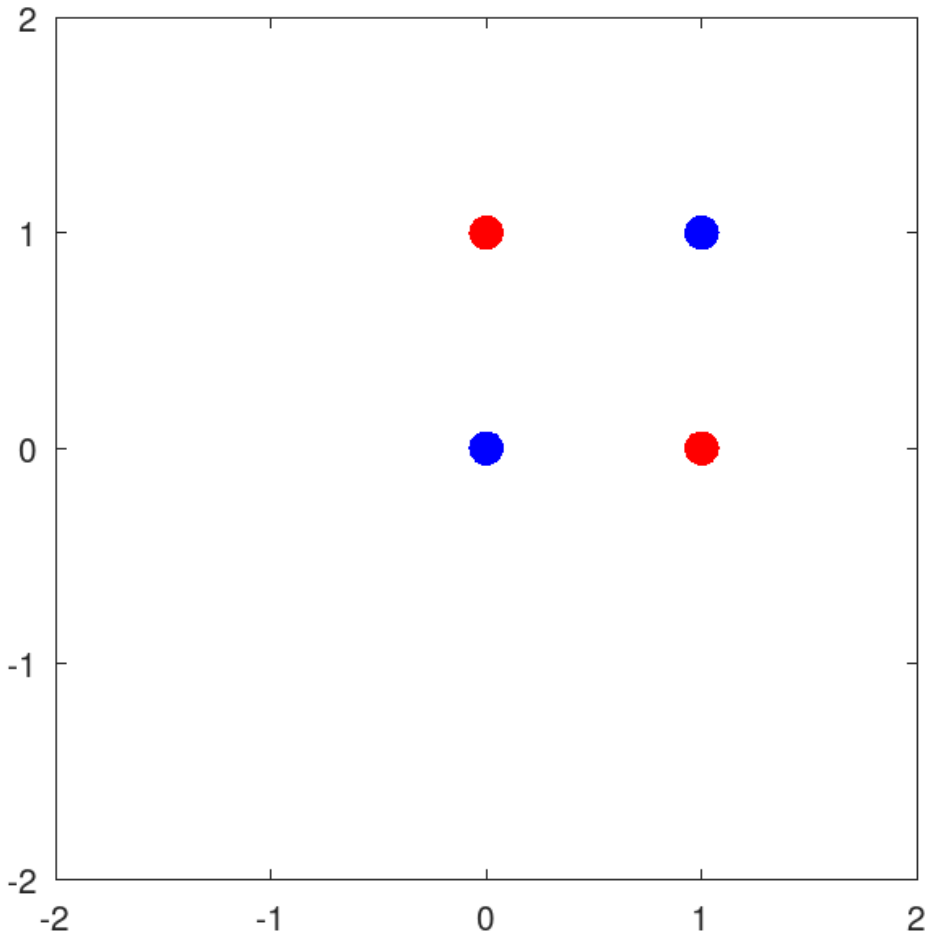
$x^3 = (1,0)^\top \quad y^3 = +1$ (V) $\qquad x^4 = (1,1)^\top \quad y^4 = -1$ (F)

Ex3 (XOR) | D

- Base de dados "XOR" : $D = (x^n, y^n)_{n=1}^4$ com

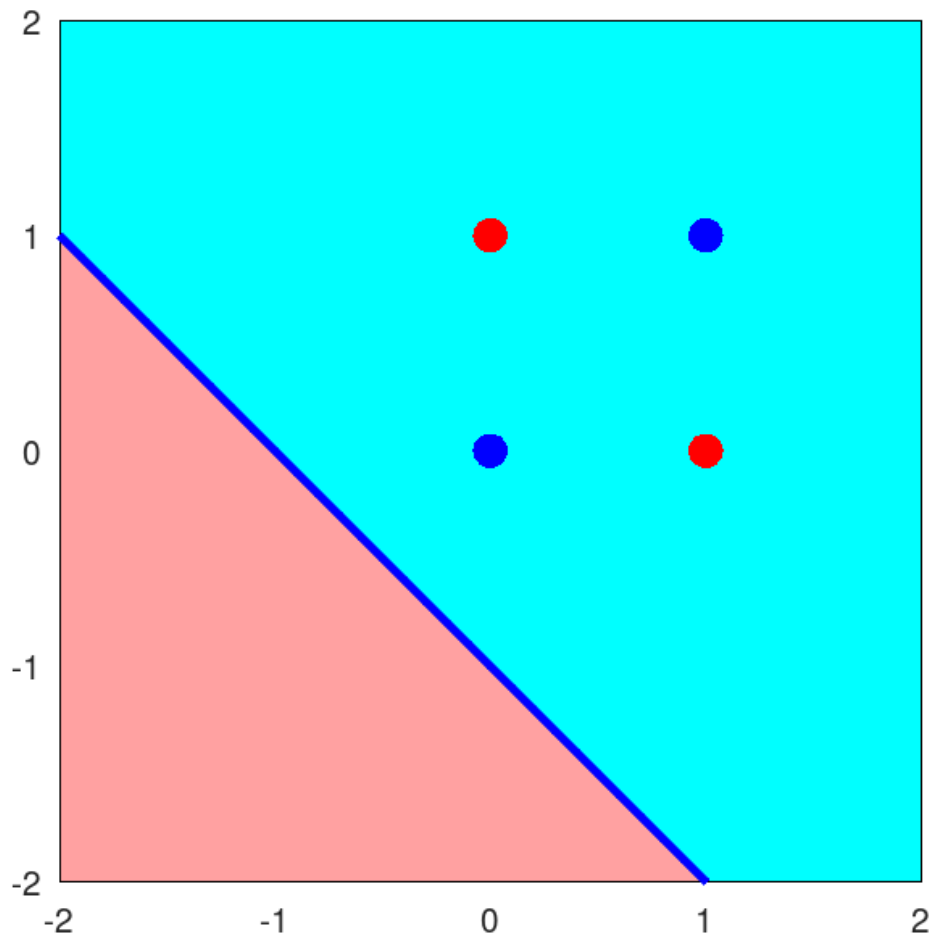
$x^1 = (0,0)^\top \quad y^1 = -1$ (F) $\qquad x^2 = (0,1)^\top \quad y^2 = +1$ (V)

$x^3 = (1,0)^\top \quad y^3 = +1$ (V) $\qquad x^4 = (1,1)^\top \quad y^4 = -1$ (F)



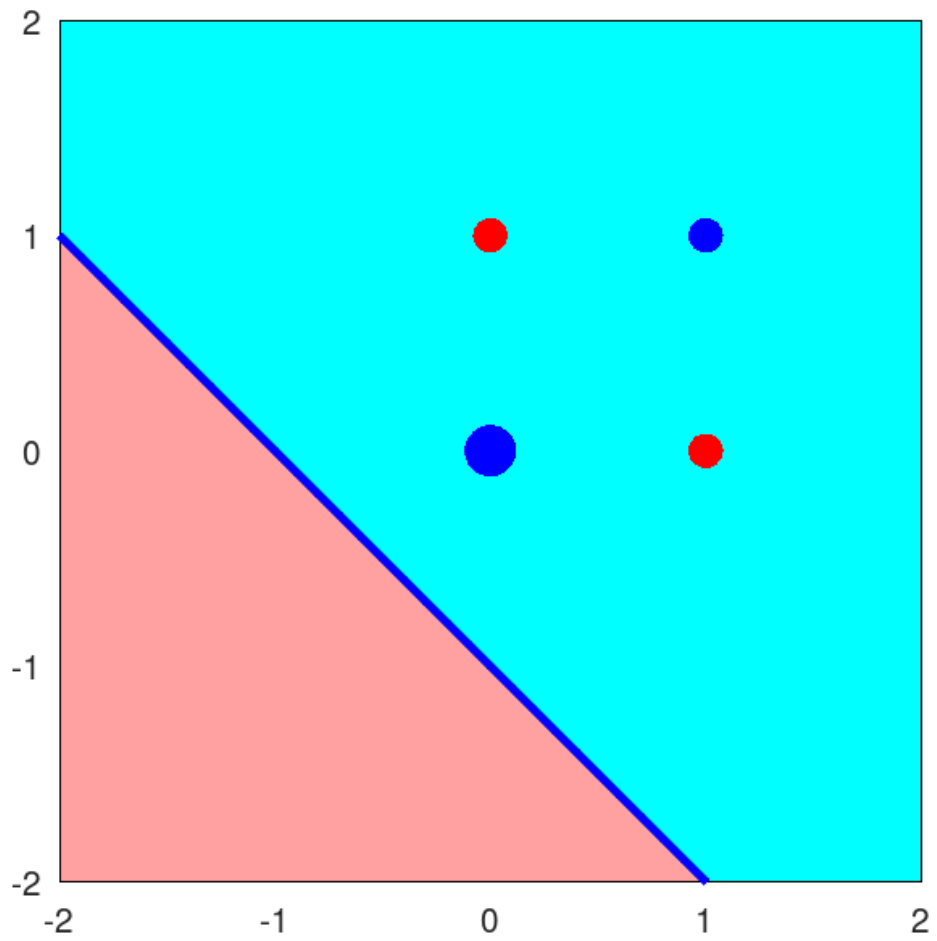
Ex3 (XOR) | Perc–v1 | $t = 0$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top, E_{(0)} = 0.5$



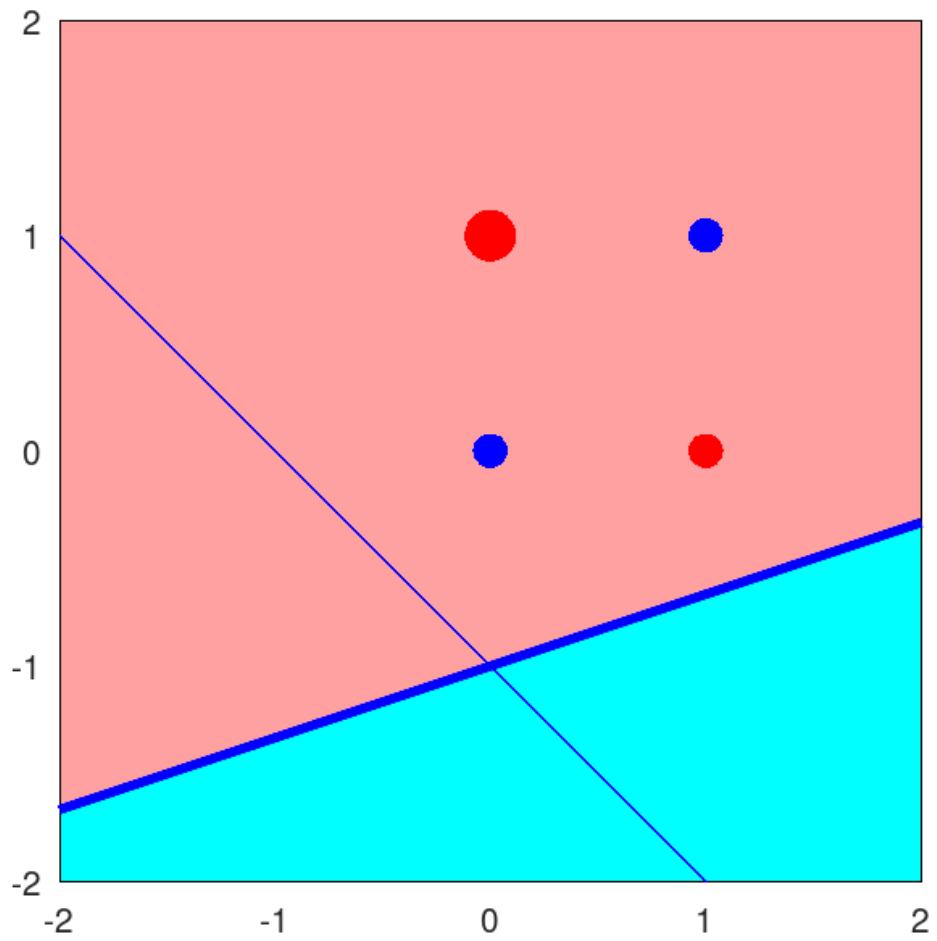
Ex3 (XOR) | Perc–v1 | $t = 1$

- $n = 1, \tilde{w}_{(1)} = (-0.25, -0.25, -0.25)^\top, E_{(1)} = 0.5$



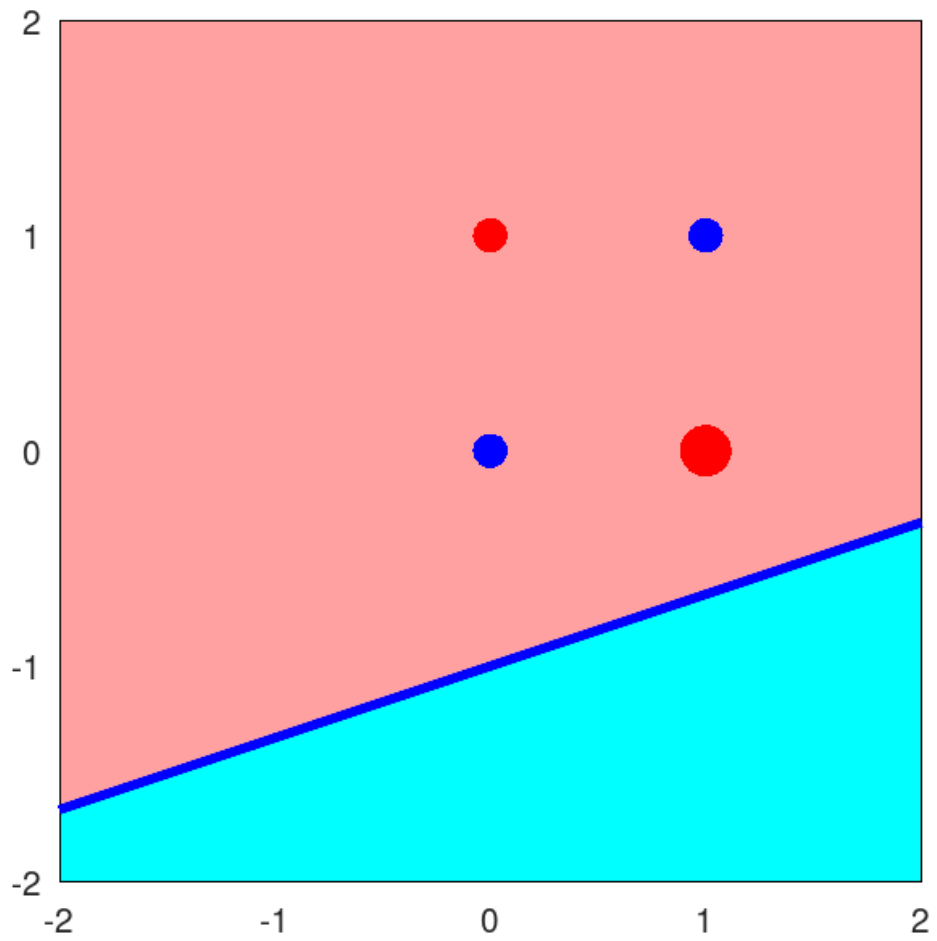
Ex3 (XOR) | Perc–v1 | $t = 2$

- $n = 2, \tilde{w}_{(2)} = (0.75, -0.25, 0.75)^\top, E_{(2)} = 0.5$



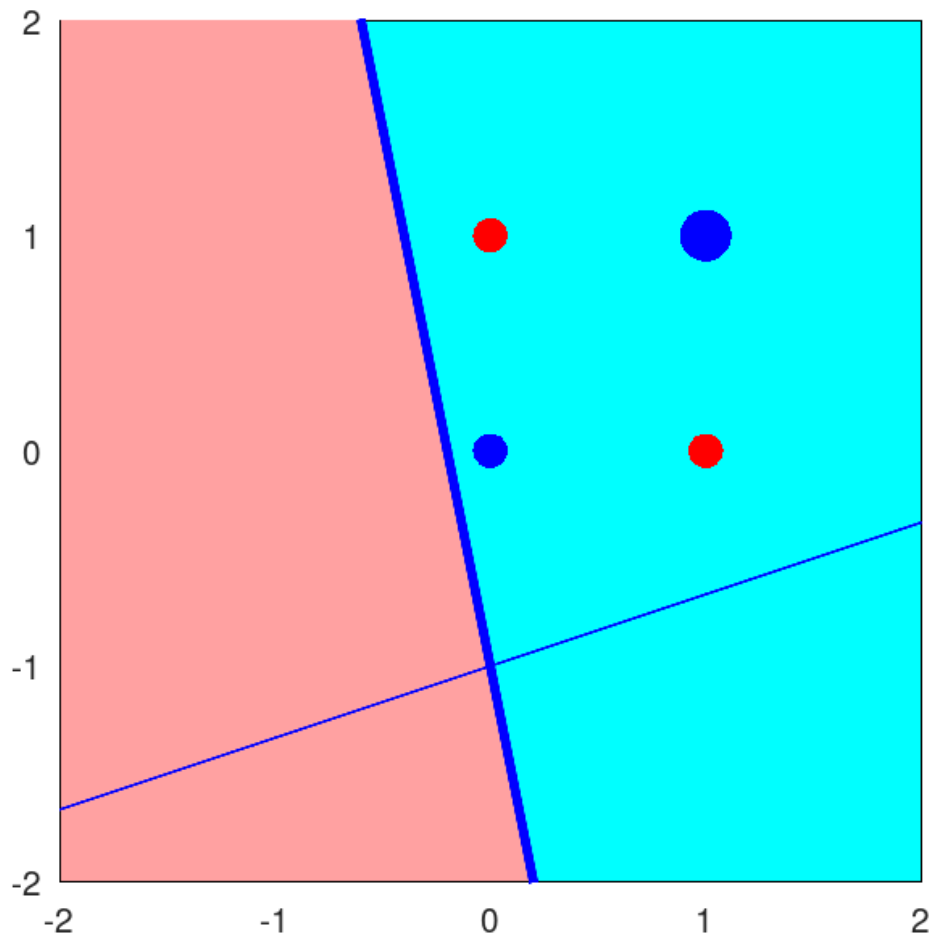
Ex3 (XOR) | Perc–v1 | $t = 3$

- $n = 3, \tilde{w}_{(3)} = (0.75, -0.25, 0.75)^\top, E_{(3)} = 0.5$



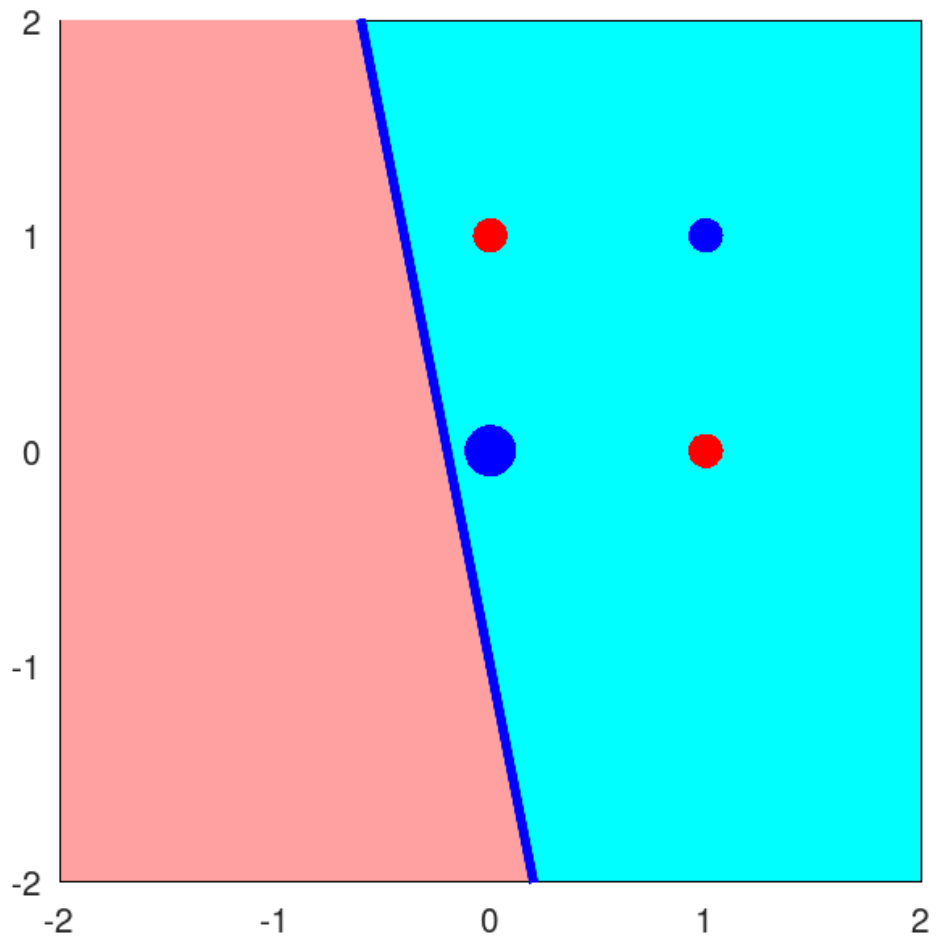
Ex3 (XOR) | Perc–v1 | $t = 4$

- $n = 4, \tilde{w}_{(4)} = (-0.25, -1.25, -0.25)^\top, E_{(4)} = 0.5$



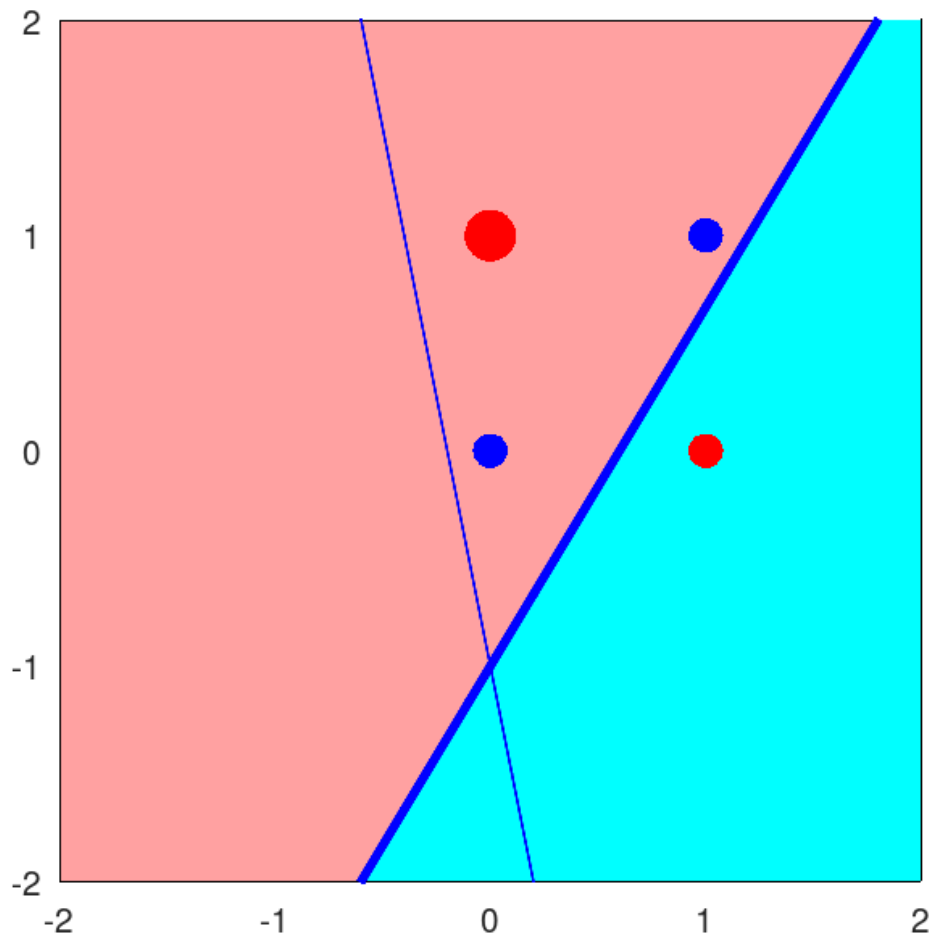
Ex3 (XOR) | Perc–v1 | $t = 5$

- $n = 1, \tilde{w}_{(5)} = (-0.25, -1.25, -0.25)^\top, E_{(5)} = 0.5$



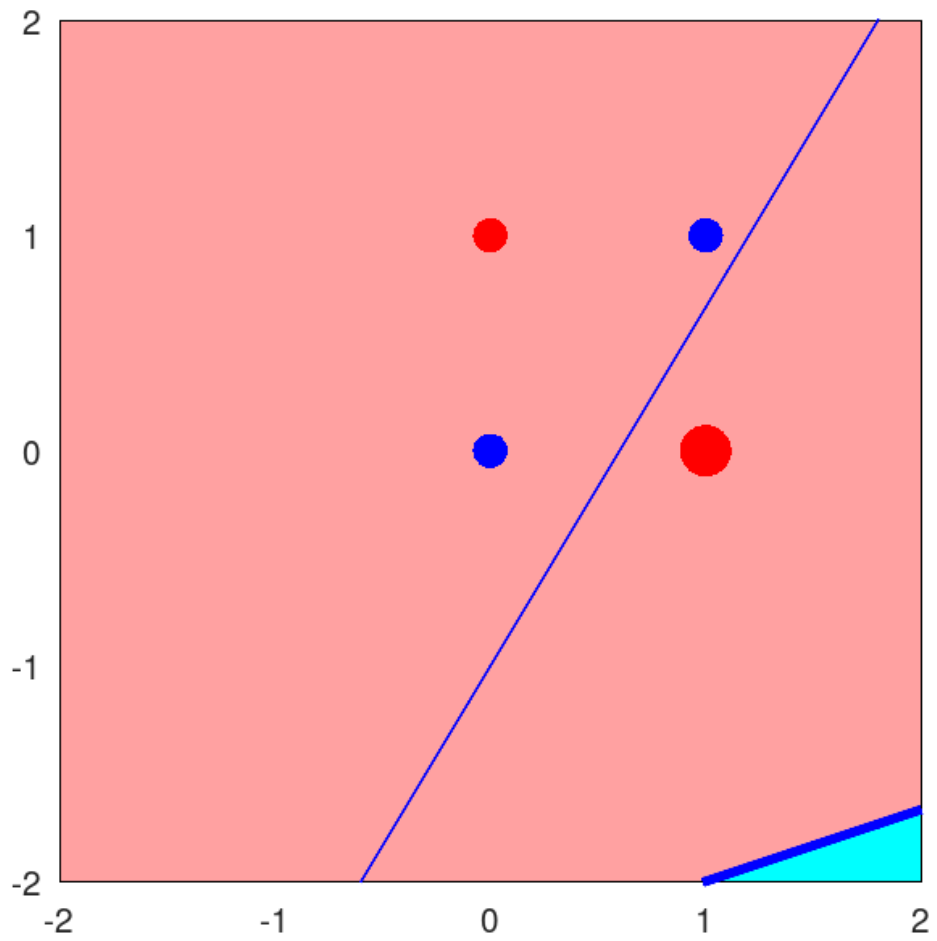
Ex3 (XOR) | Perc–v1 | $t = 6$

- $n = 2, \tilde{w}_{(6)} = (0.75, -1.25, 0.75)^\top, E_{(6)} = 0.75$



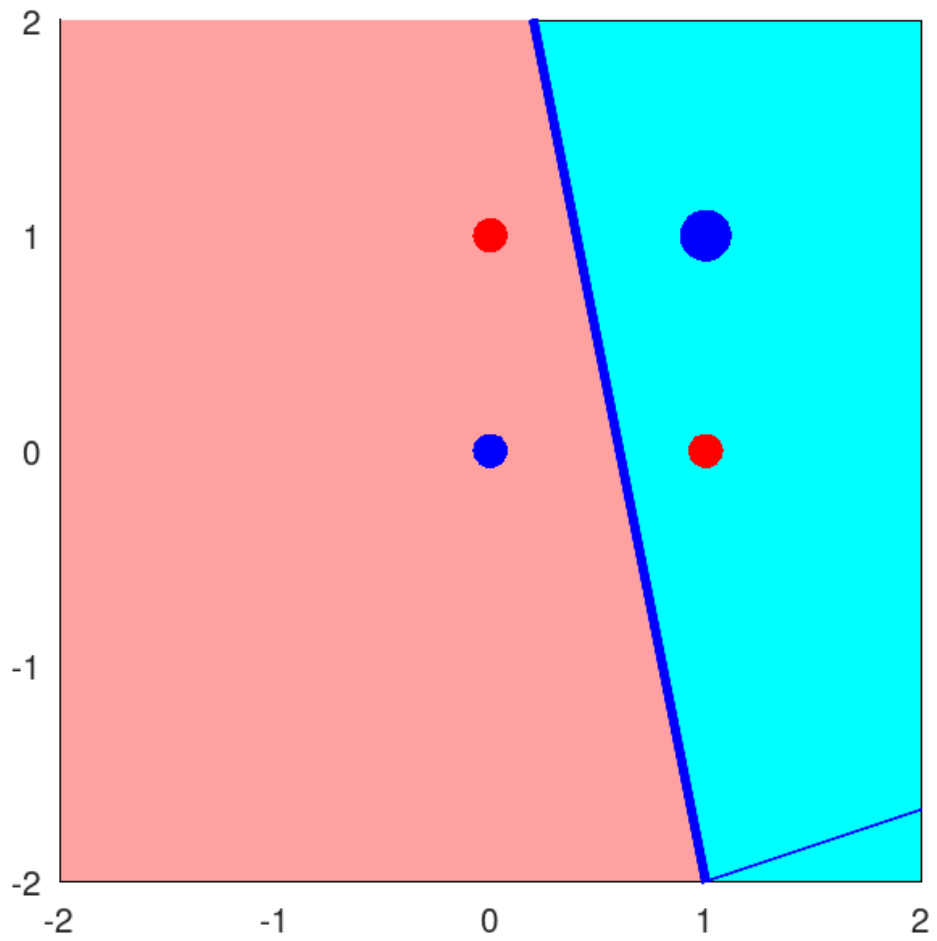
Ex3 (XOR) | Perc–v1 | $t = 7$

- $n = 3, \tilde{w}_{(7)} = (1.75, -0.25, 0.75)^\top, E_{(7)} = 0.5$



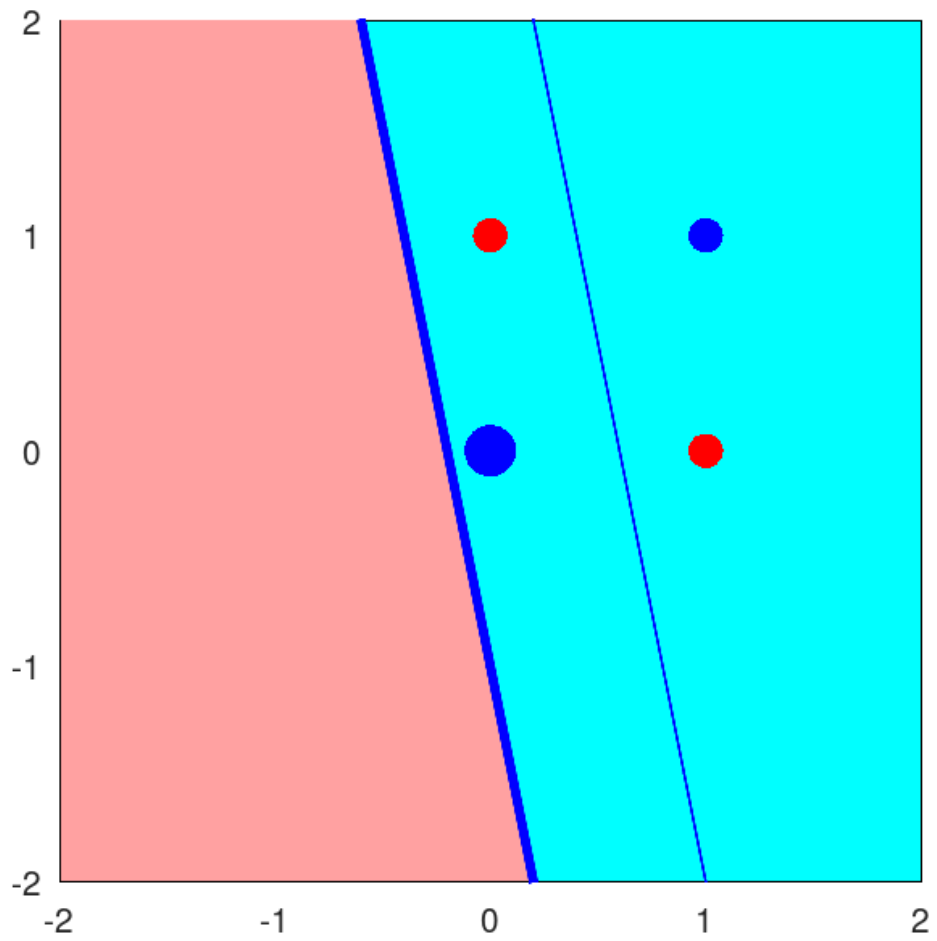
Ex3 (XOR) | Perc–v1 | $t = 8$

- $n = 4, \tilde{w}_{(8)} = (0.75, -1.25, -0.25)^\top, E_{(8)} = 0.5$



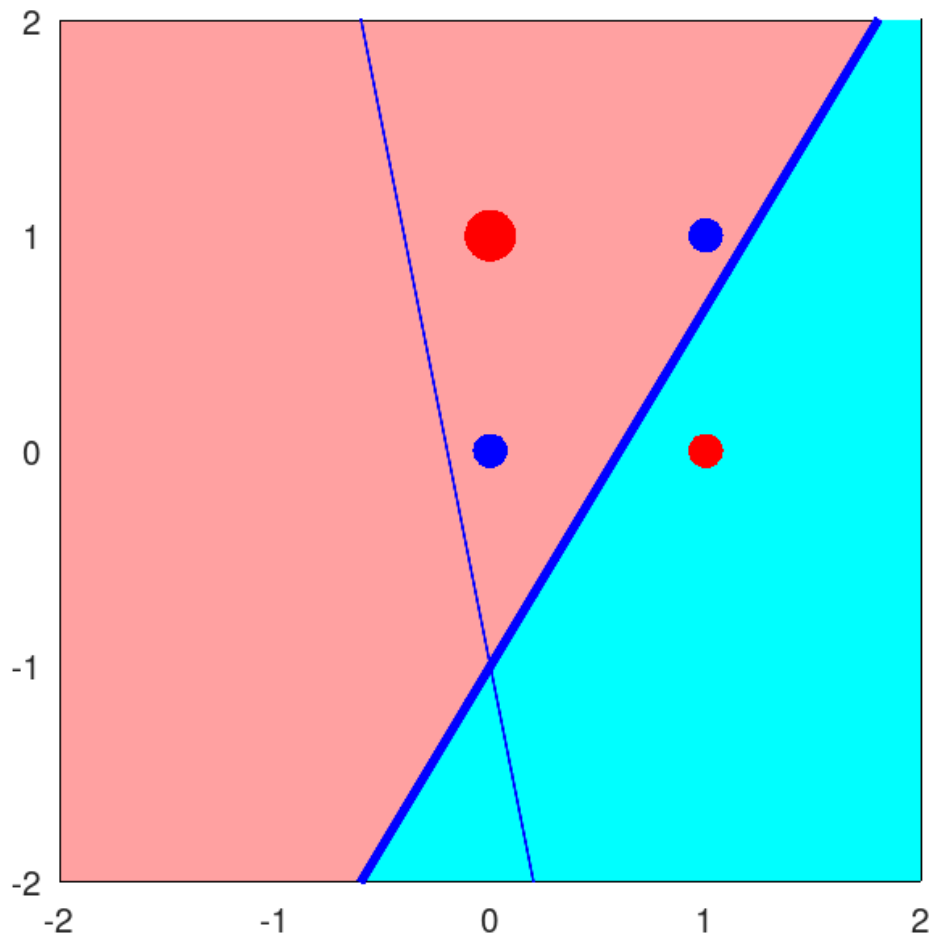
Ex3 (XOR) | Perc–v1 | $t = 9$

- $n = 1, \tilde{w}_{(9)} = (-0.25, -1.25, -0.25)^\top, E_{(9)} = 0.5$



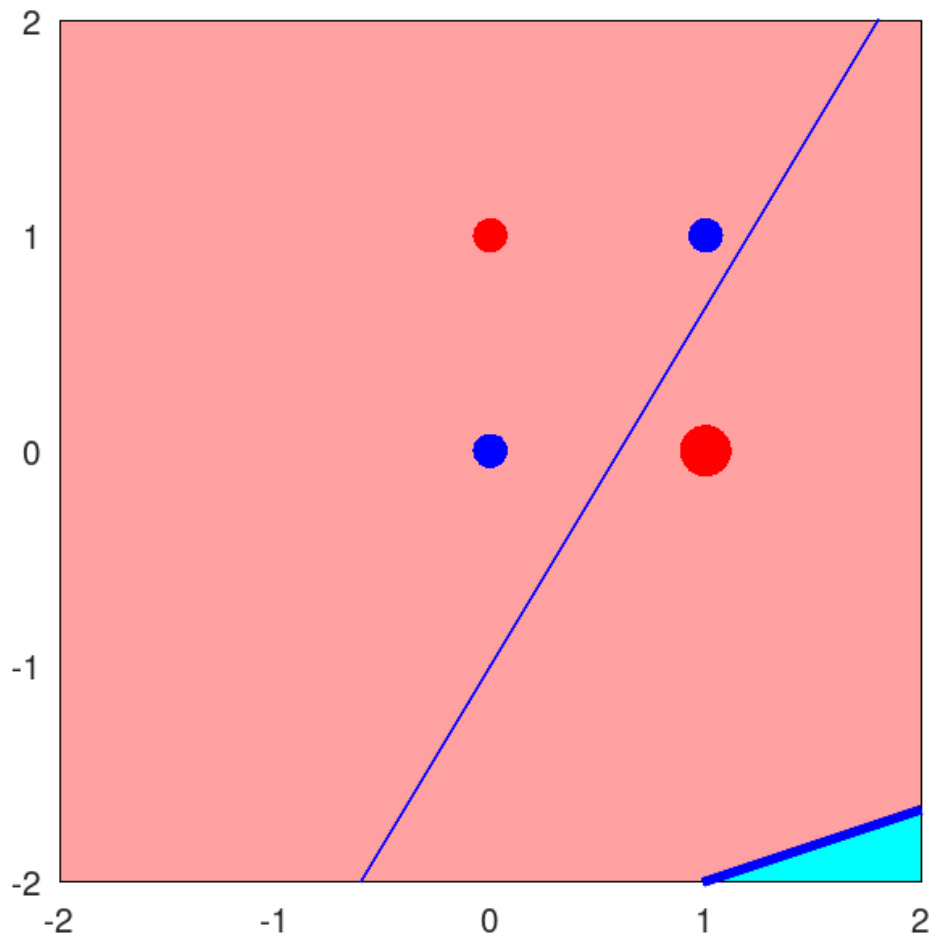
Ex3 (XOR) | Perc–v1 | $t = 10$

- $n = 2, \tilde{w}_{(10)} = (0.75, -1.25, 0.75)^\top, E_{(10)} = 0.75$



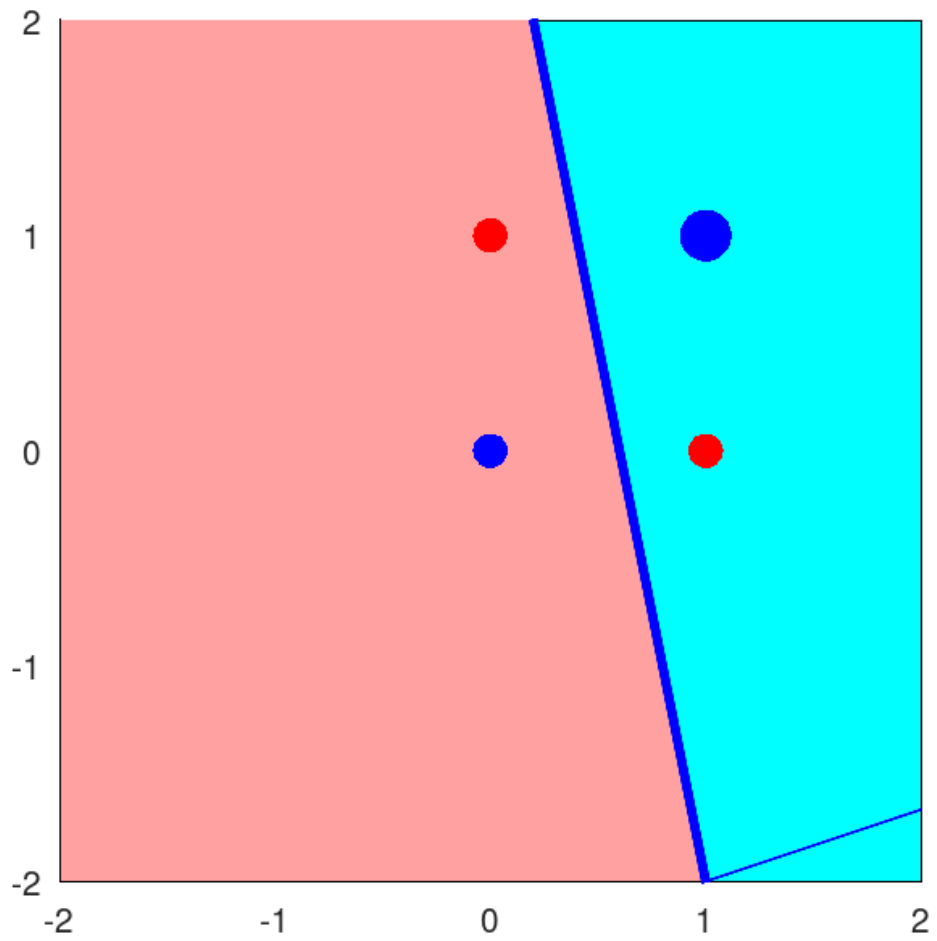
Ex3 (XOR) | Perc–v1 | $t = 11$

- $n = 3, \tilde{w}_{(11)} = (1.75, -0.25, 0.75)^\top, E_{(11)} = 0.5$



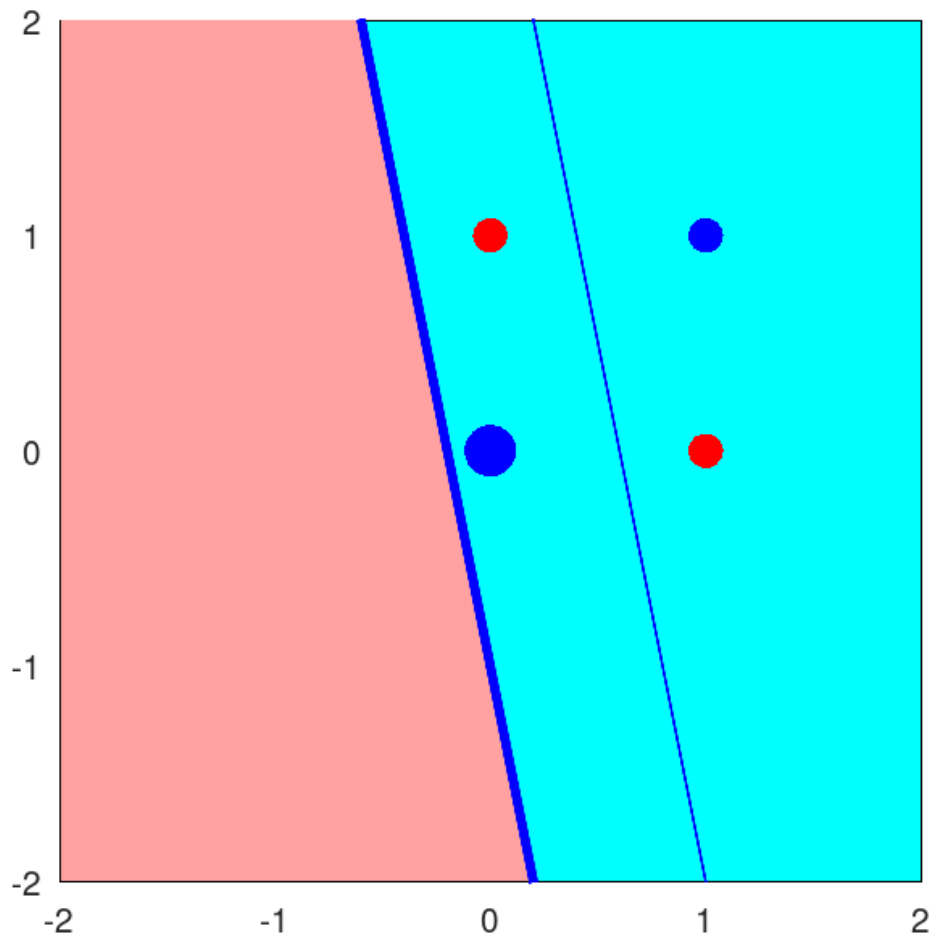
Ex3 (XOR) | Perc–v1 | $t = 12$

- $n = 4, \tilde{w}_{(12)} = (0.75, -1.25, -0.25)^\top, E_{(12)} = 0.5$



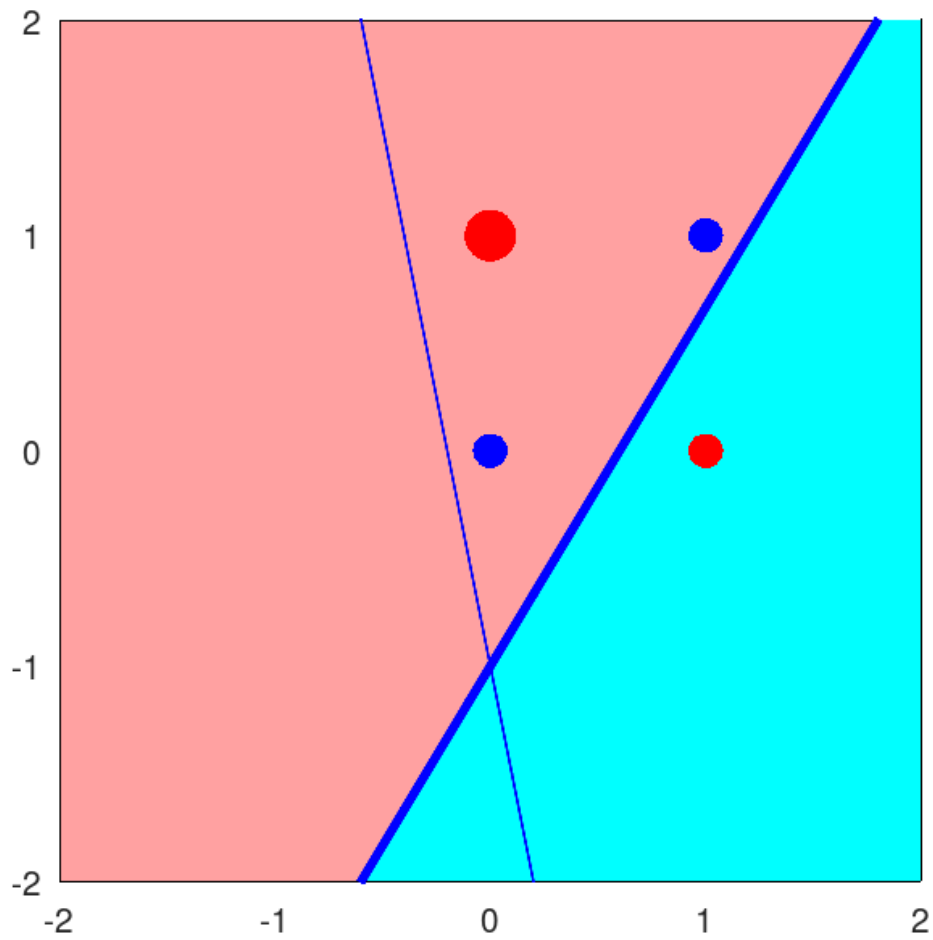
Ex3 (XOR) | Perc–v1 | $t = 13$

- $n = 1, \tilde{w}_{(13)} = (-0.25, -1.25, -0.25)^\top, E_{(13)} = 0.5$



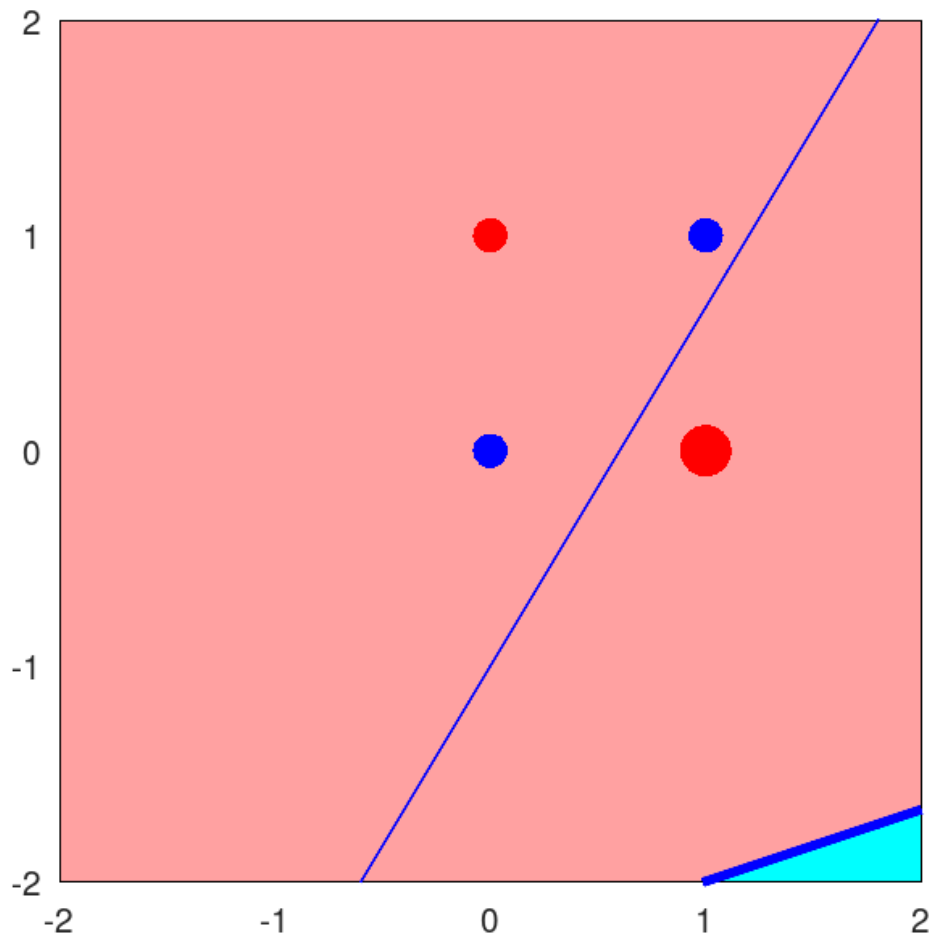
Ex3 (XOR) | Perc–v1 | $t = 14$

- $n = 2, \tilde{w}_{(14)} = (0.75, -1.25, 0.75)^\top, E_{(14)} = 0.75$



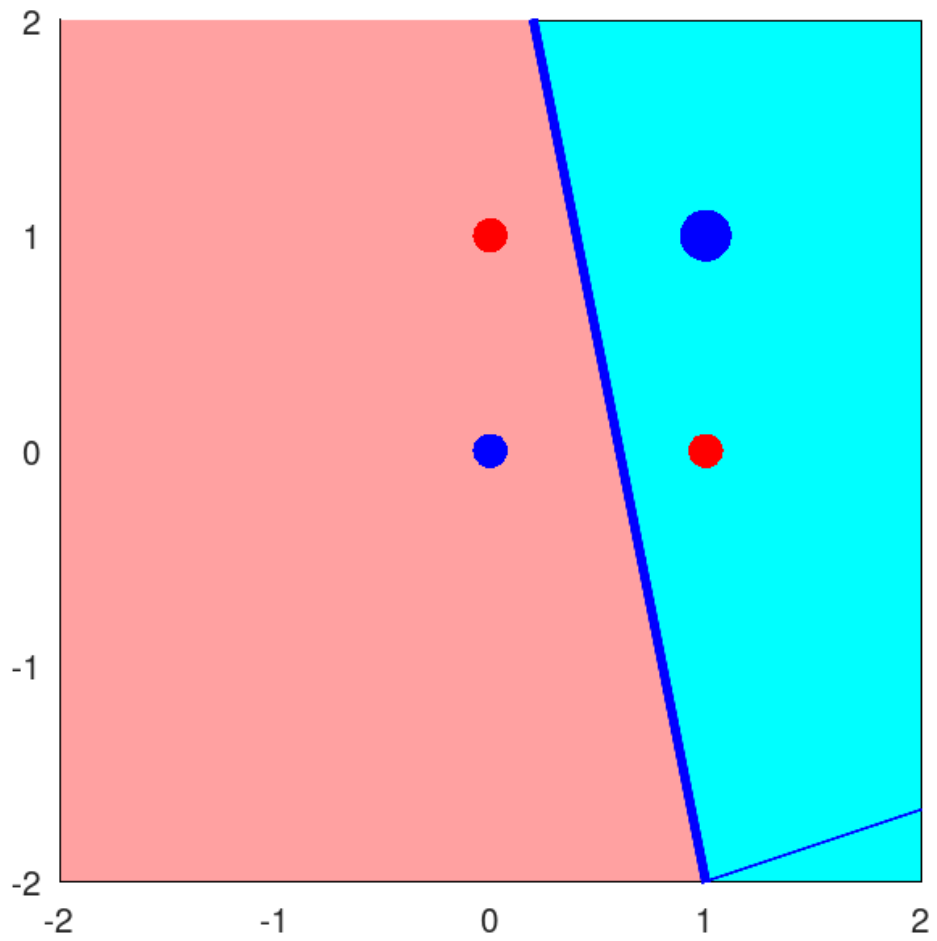
Ex3 (XOR) | Perc–v1 | $t = 15$

- $n = 3, \tilde{w}_{(15)} = (1.75, -0.25, 0.75)^\top, E_{(15)} = 0.5$



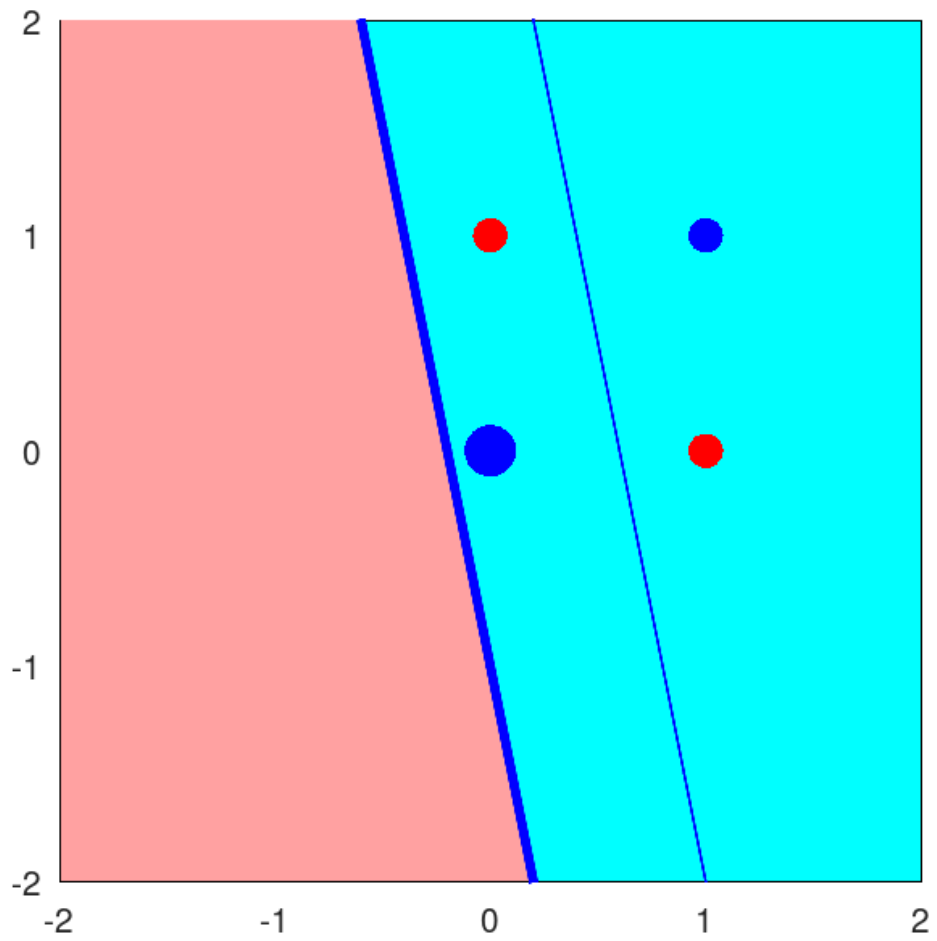
Ex3 (XOR) | Perc–v1 | $t = 16$

- $n = 4, \tilde{w}_{(16)} = (0.75, -1.25, -0.25)^\top, E_{(16)} = 0.5$



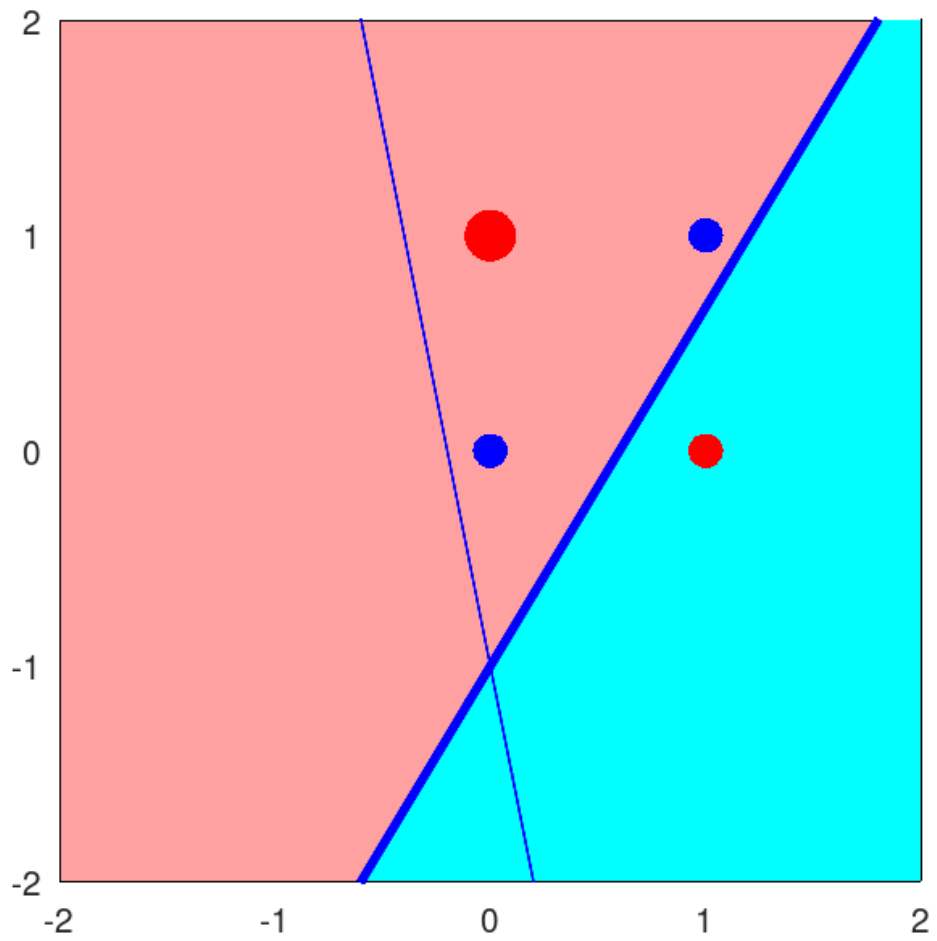
Ex3 (XOR) | Perc–v1 | $t = 17$

- $n = 1, \tilde{w}_{(17)} = (-0.25, -1.25, -0.25)^\top, E_{(17)} = 0.5$



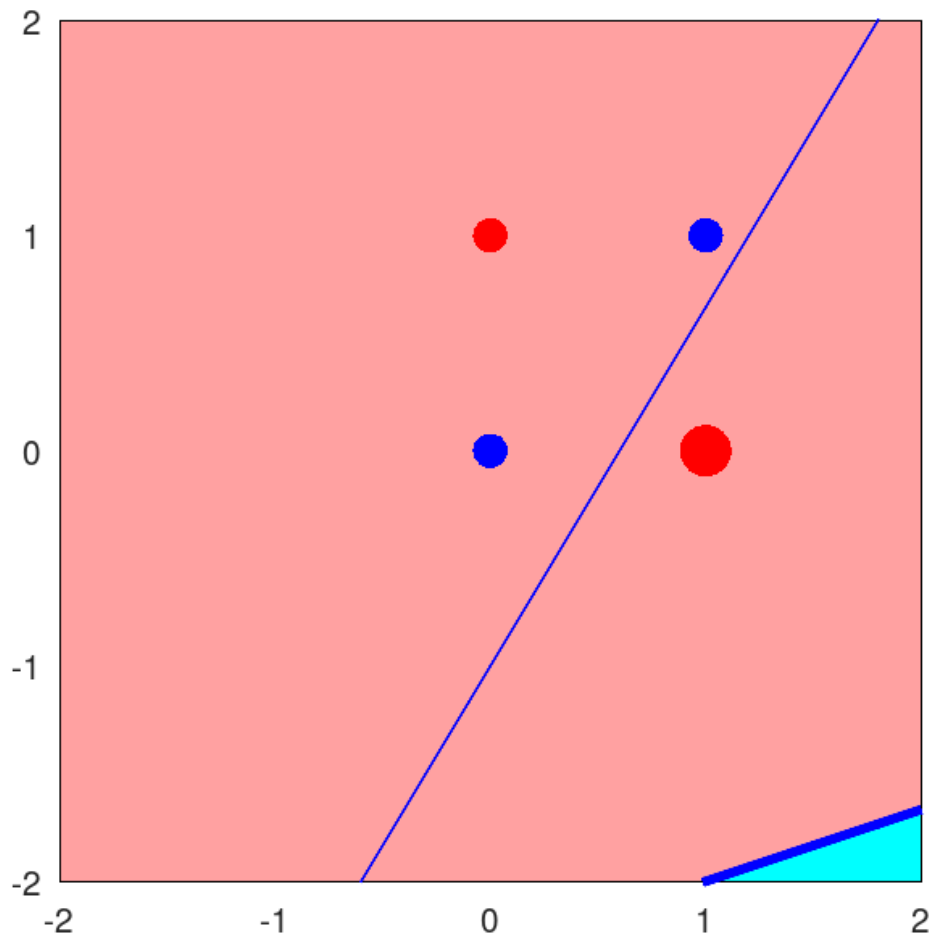
Ex3 (XOR) | Perc–v1 | $t = 18$

- $n = 2, \tilde{w}_{(18)} = (0.75, -1.25, 0.75)^\top, E_{(18)} = 0.75$



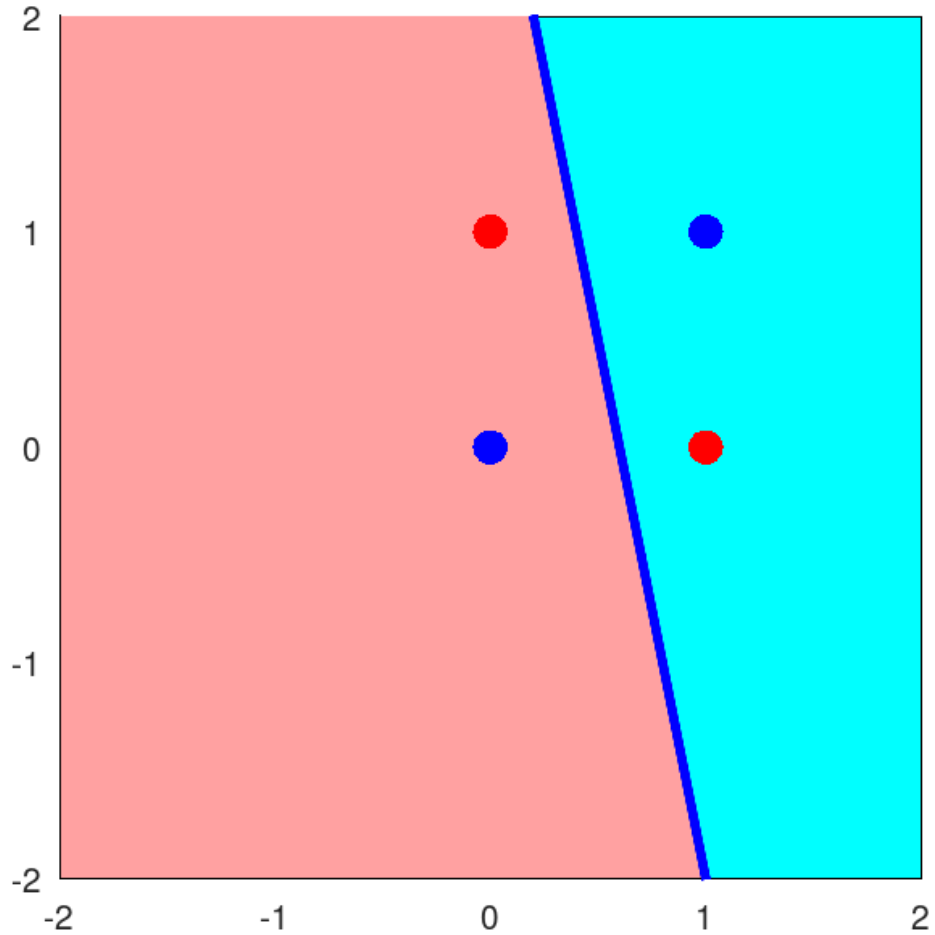
Ex3 (XOR) | Perc–v1 | $t = 19$

- $n = 3, \tilde{w}_{(19)} = (1.75, -0.25, 0.75)^\top, E_{(19)} = 0.5$



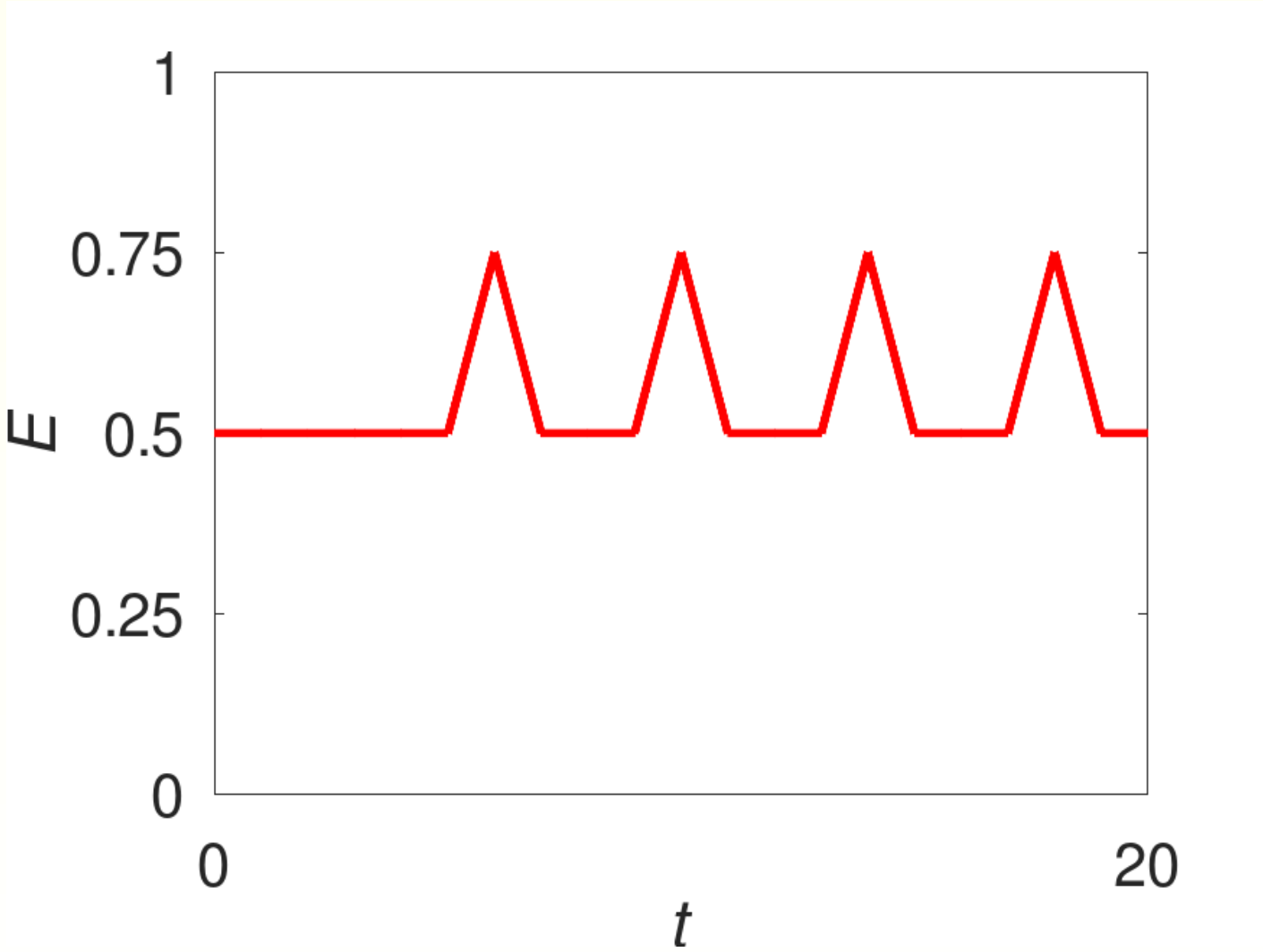
Ex3 (XOR) | Perc–v1 | $t = 20$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top, E_{(0)} = 0.5$
- $n = 4, \tilde{w}_{(20)} = (0.75, -1.25, -0.25)^\top, E_{(20)} = 0.5$

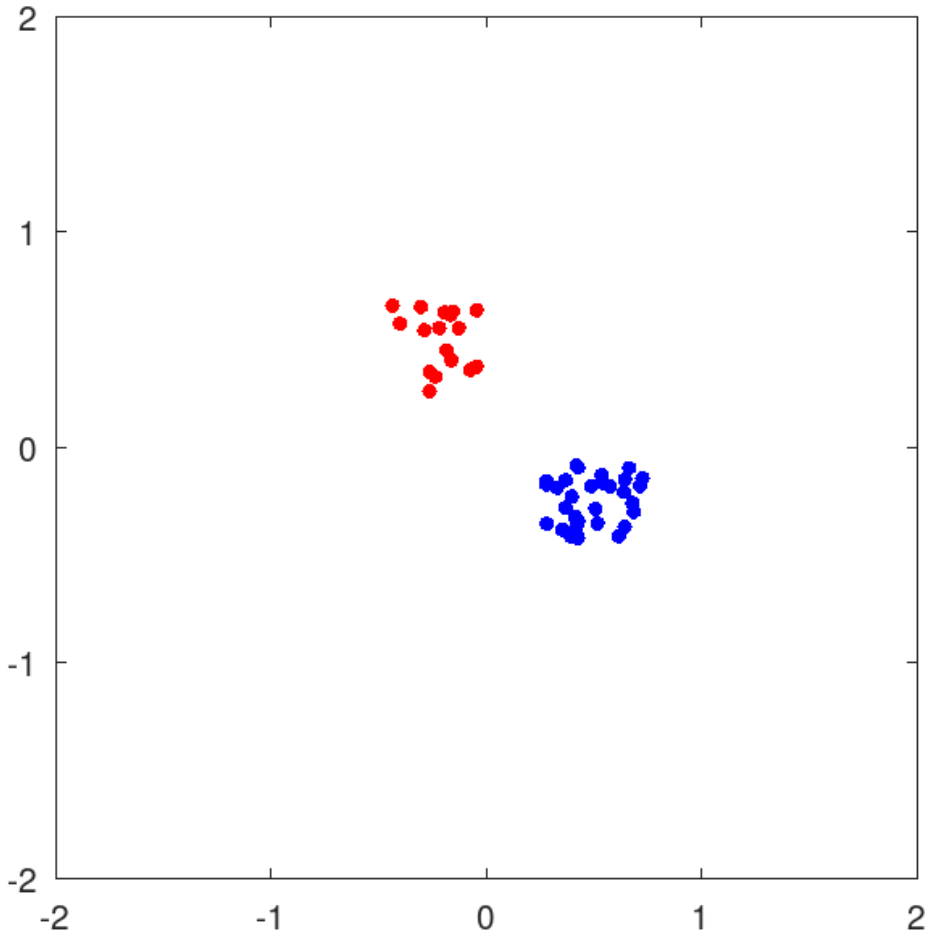


Ex3 (XOR) | Perc–v1 | E

Ex3 (XOR) | Perc–v1 | E

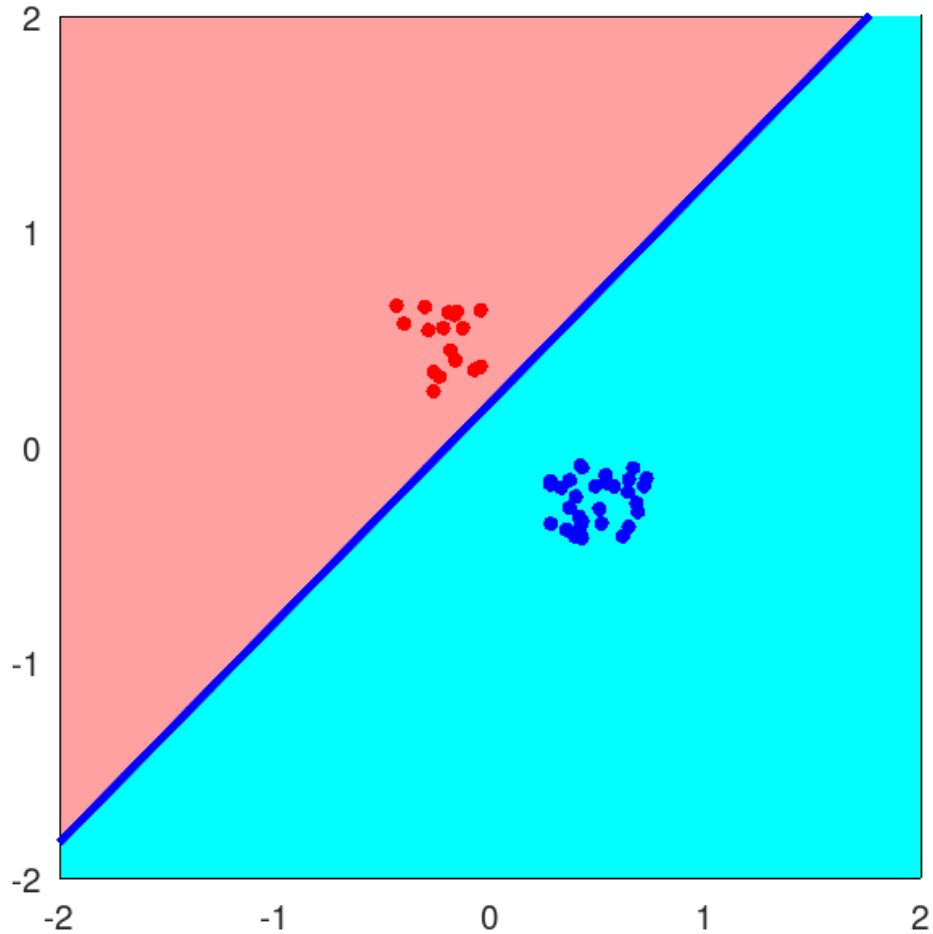


Ex4 | D



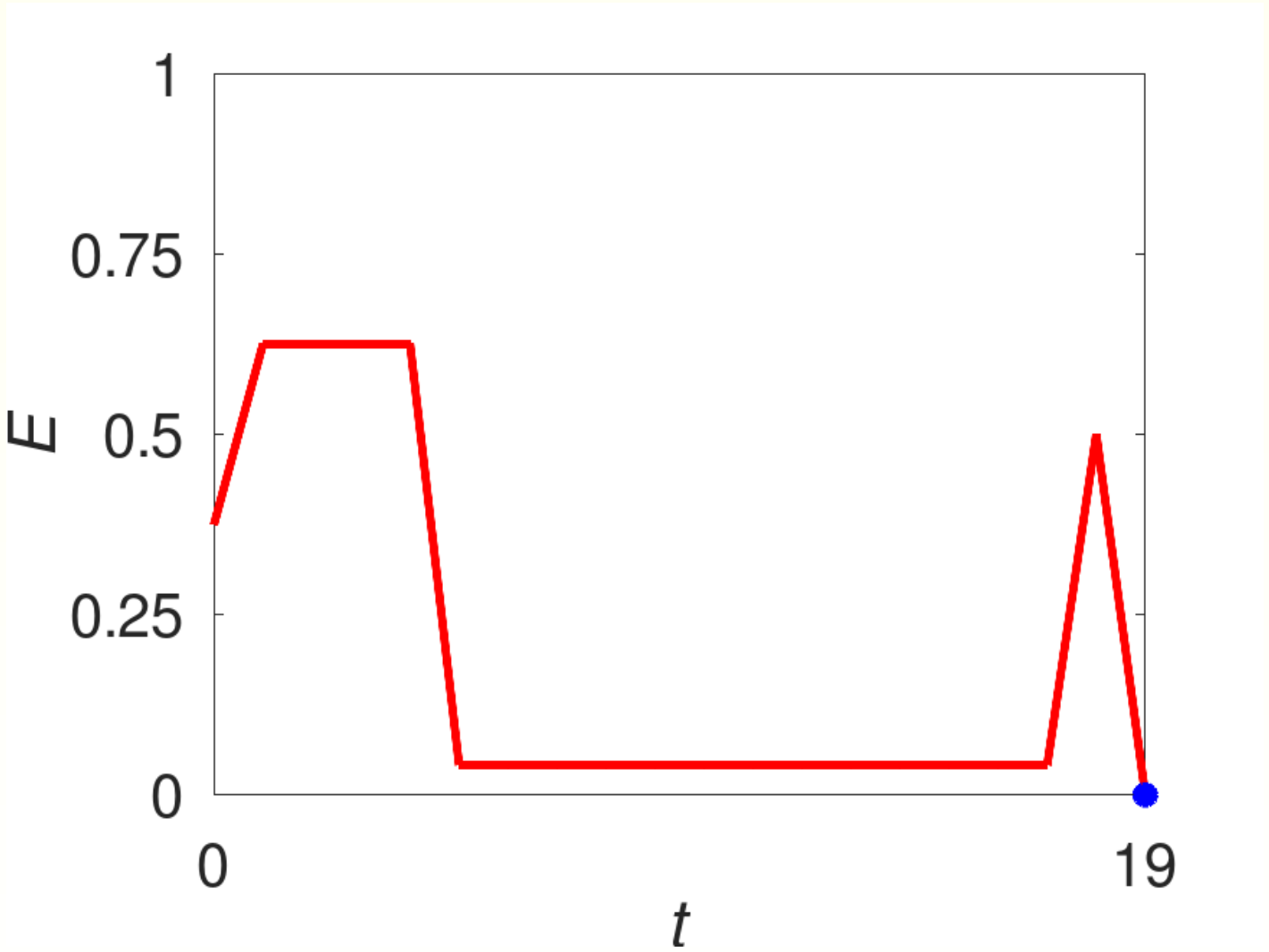
Ex4 | Perc–v1 | $t = 19$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top, E_{(0)} = 0.375$
- $\tilde{w}^* = (-0.25, -1.2322, 1.2076)^\top, E_{(19)} = 0$

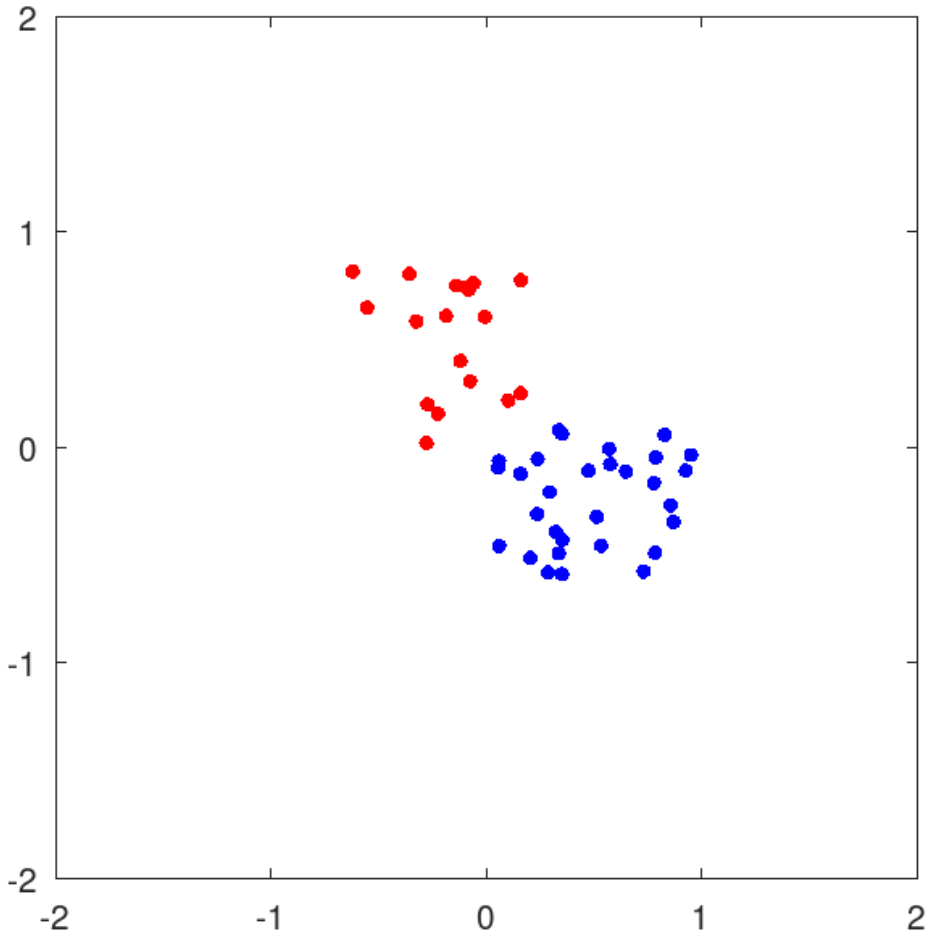


Ex4 | Perc–v1 | E

Ex4 | Perc–v1 | E

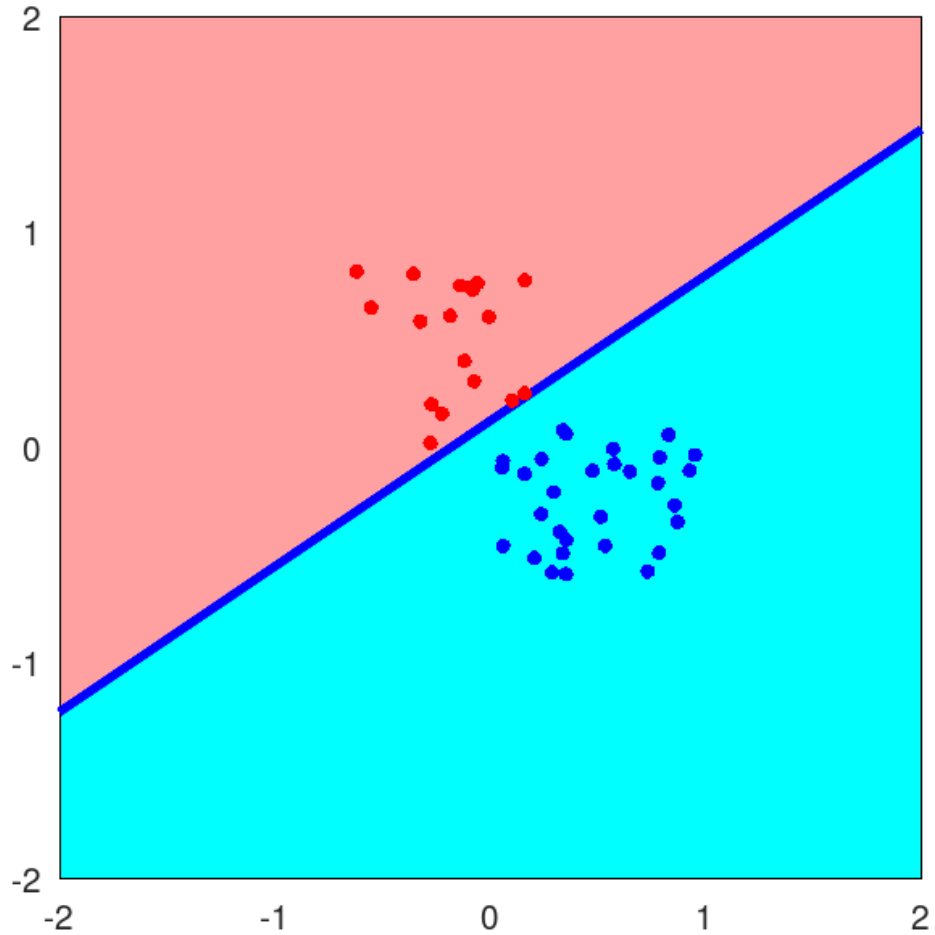


Ex5 | D



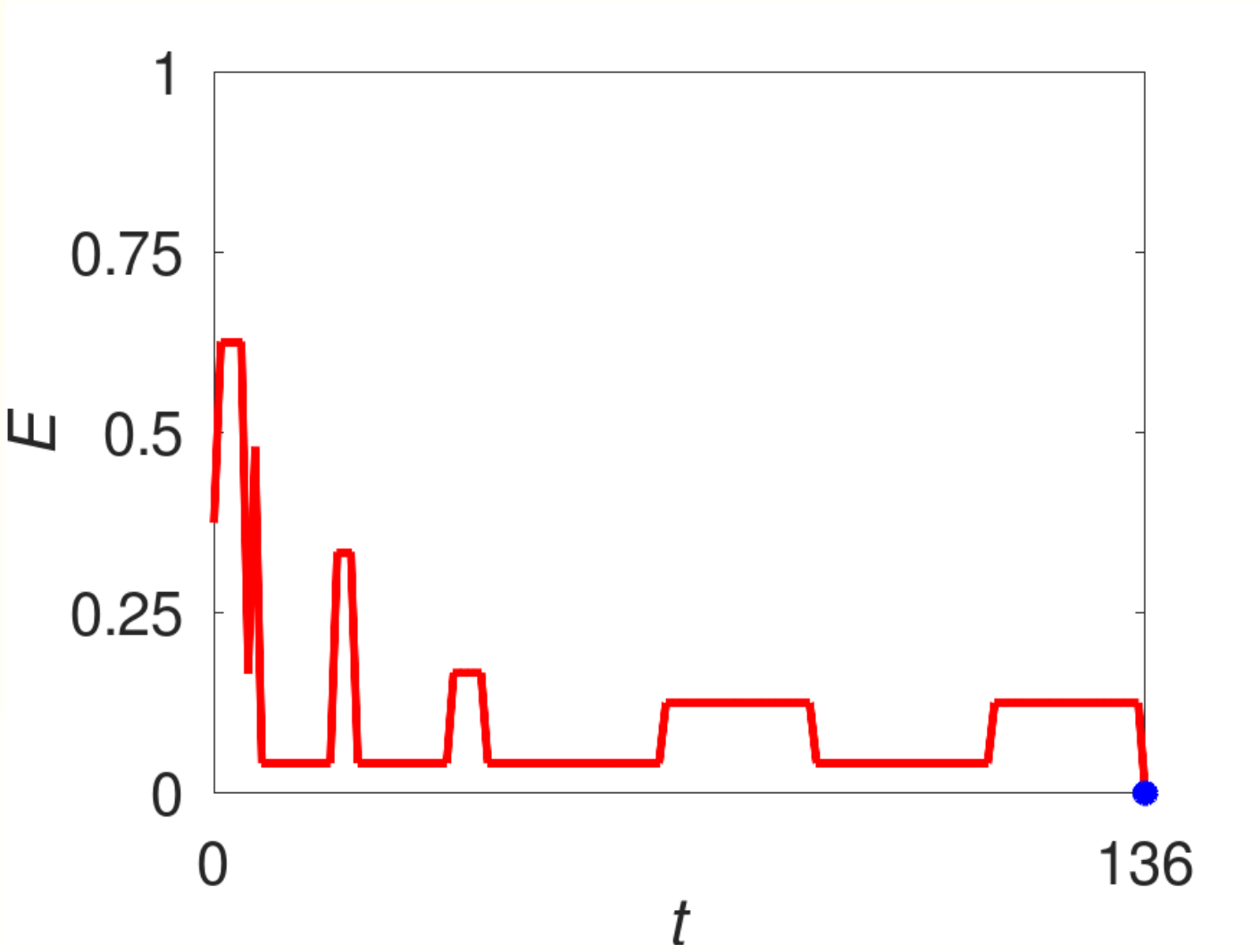
Ex5 | Perc–v1 | $t = 136$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top, E_{(0)} = 0.375$
- $\tilde{w}^* = (-0.25, -1.3712, 2.0301)^\top, E_{(136)} = 0$

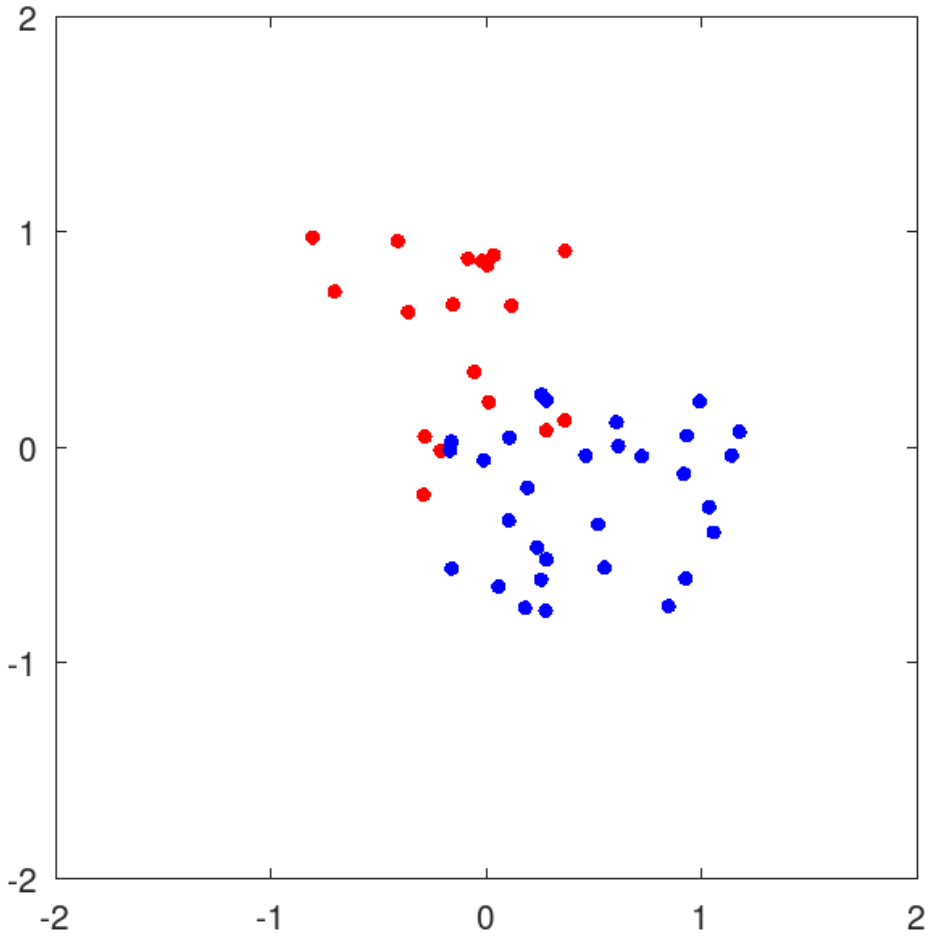


Ex5 | Perc–v1 | E

Ex5 | Perc–v1 | E

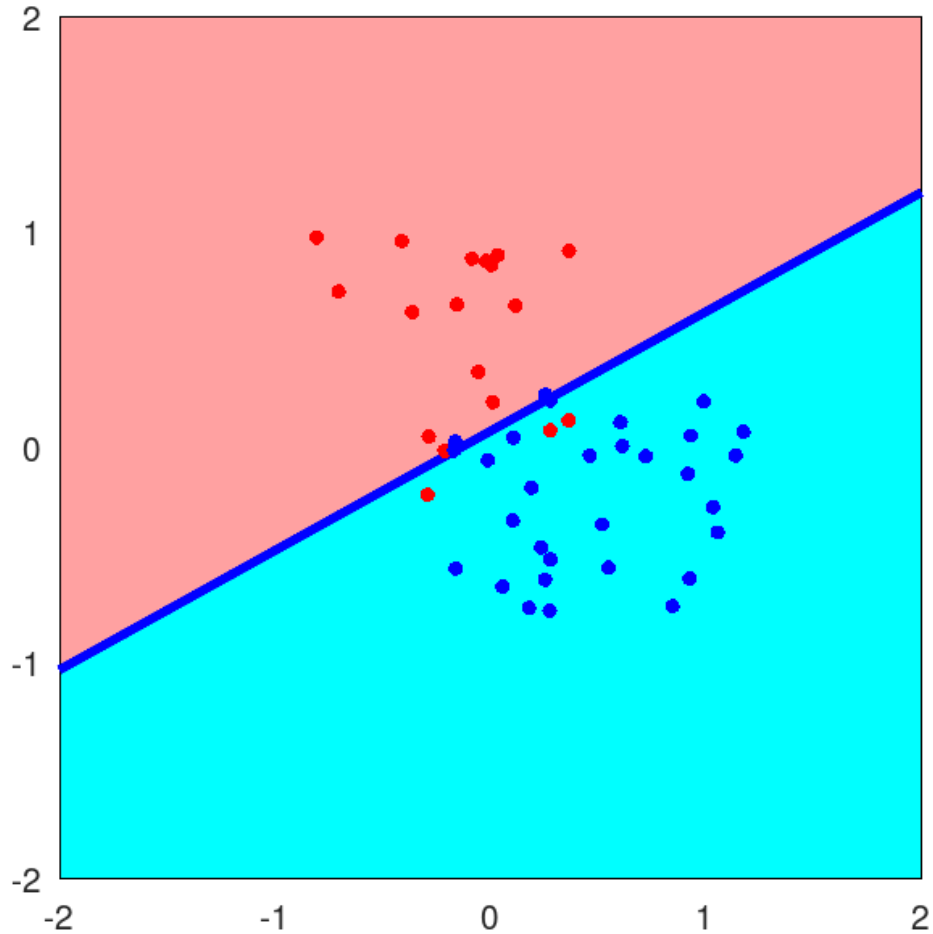


Ex6 | D



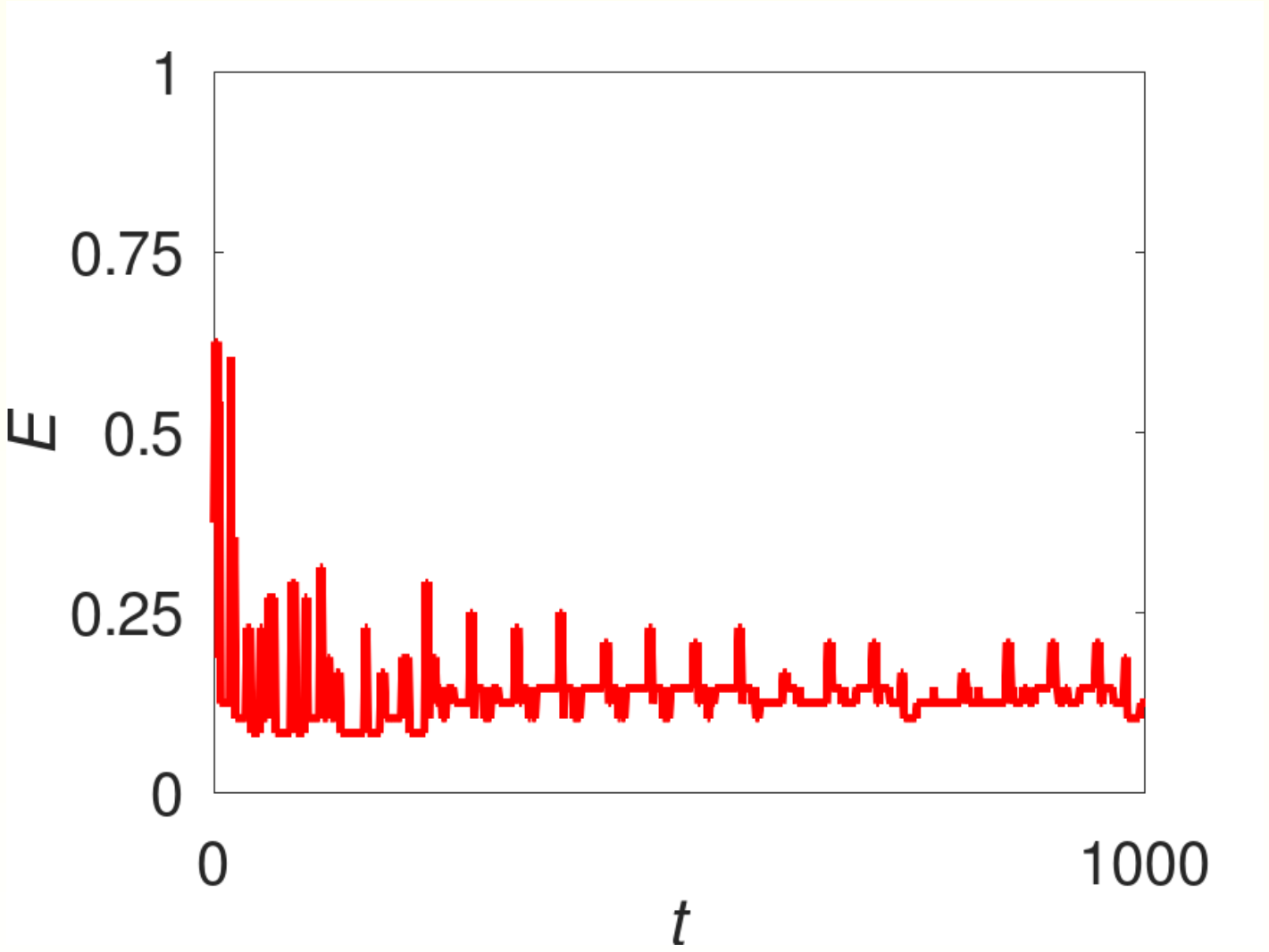
Ex6 | Perc–v1 | $t = 1000$

- $\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top, E_{(0)} = 0.375$
- $n = 40, \tilde{w}_{(1000)} = (-0.25, -1.8483, 3.3385)^\top, E_{(1000)} = 0.125$



Ex6 | Perc–v1 | E

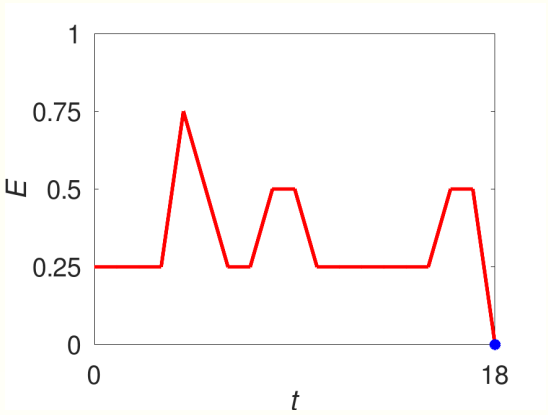
Ex6 | Perc–v1 | E



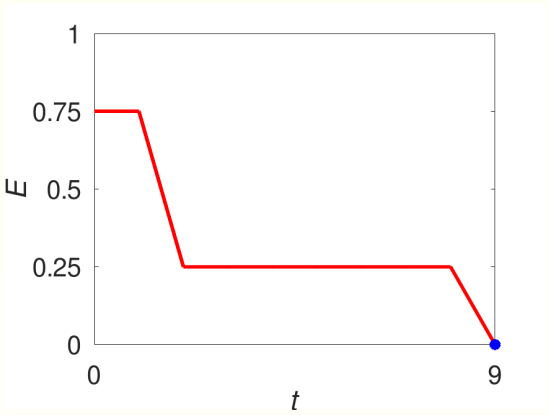
Perc–v1 | comentários (i)

- Se D é um conjunto linearmente separável, o Perc–v1 termina.

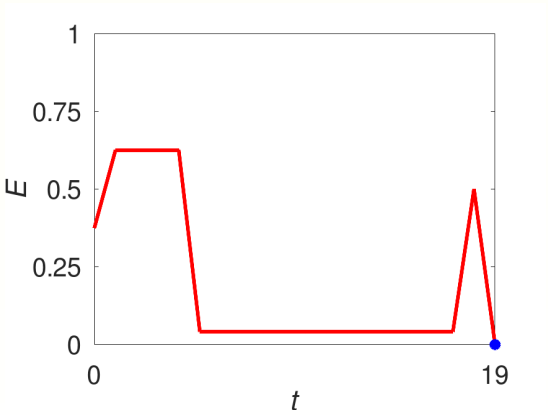
exemplo 1



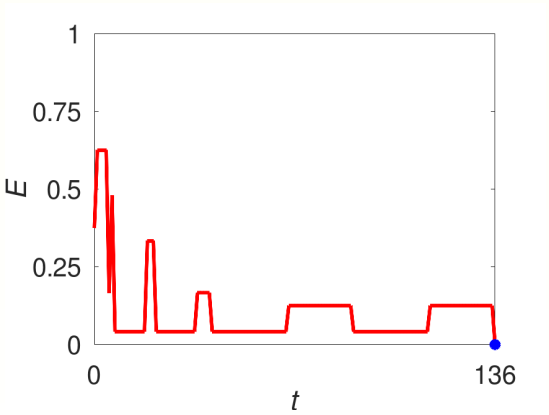
exemplo 2



exemplo 4

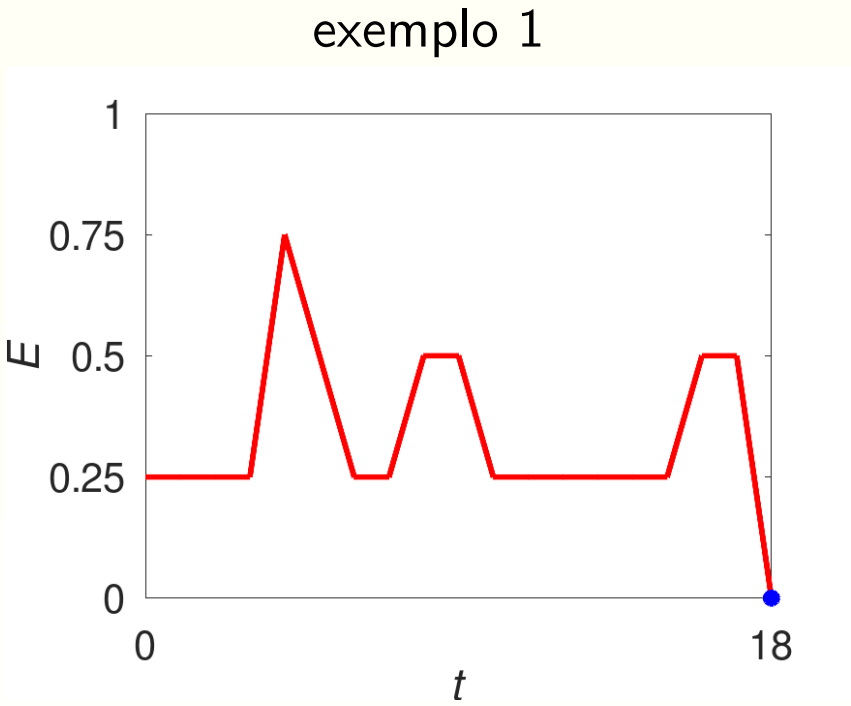


exemplo 5



Perc–v1 | comentários (ii)

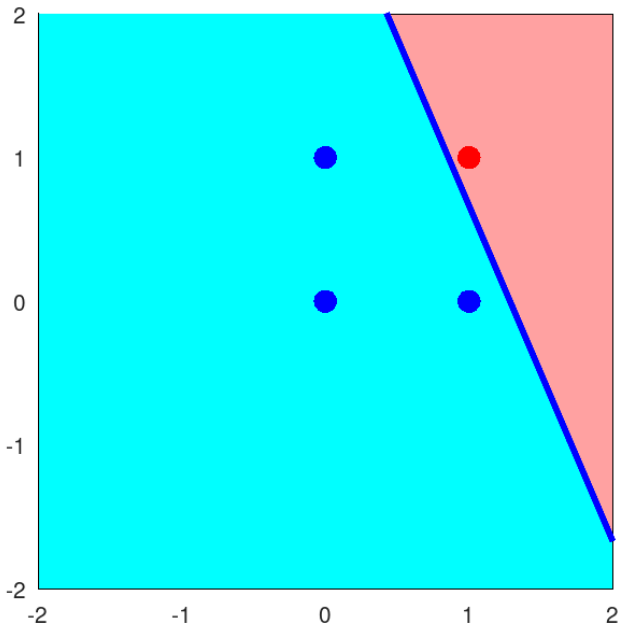
- A função custo E não é necessariamente uma função monótona decrescente.



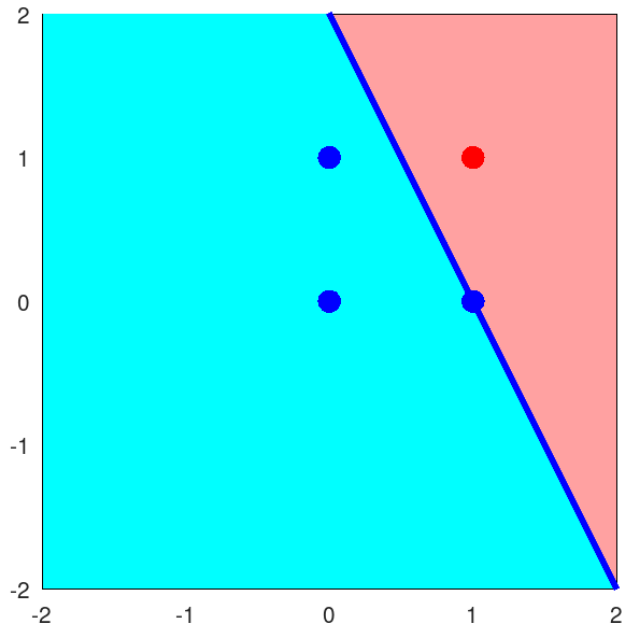
Perc–v1 | comentários (iii)

- O algoritmo pode convergir para soluções diferentes dependendo do valor de $\tilde{w}_{(0)}$.

$\tilde{w}_{(0)} = (-0.25, -0.25, -0.25)^\top$
$\tilde{w}^* = (-2.25, 1.75, 0.75)^\top$
$t = 18$



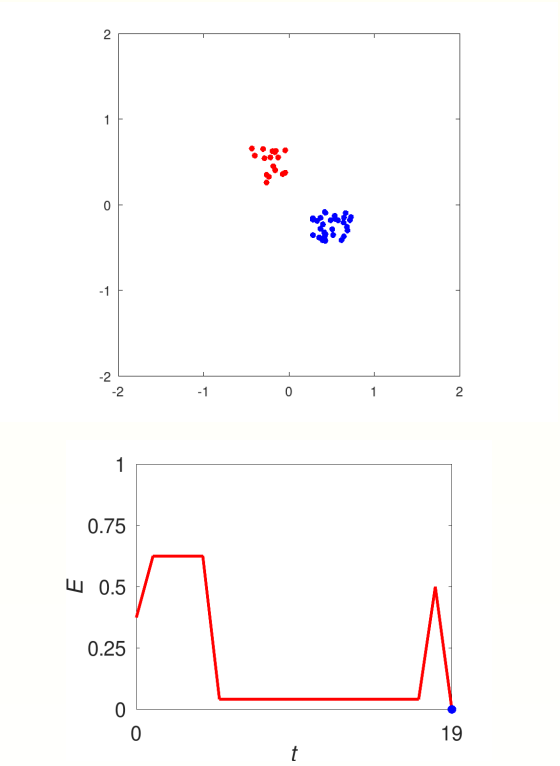
$\tilde{w}_{(0)} = (1, 1, 1)^\top$
$\tilde{w}^* = (-2, 2, 1)^\top$
$t = 14$



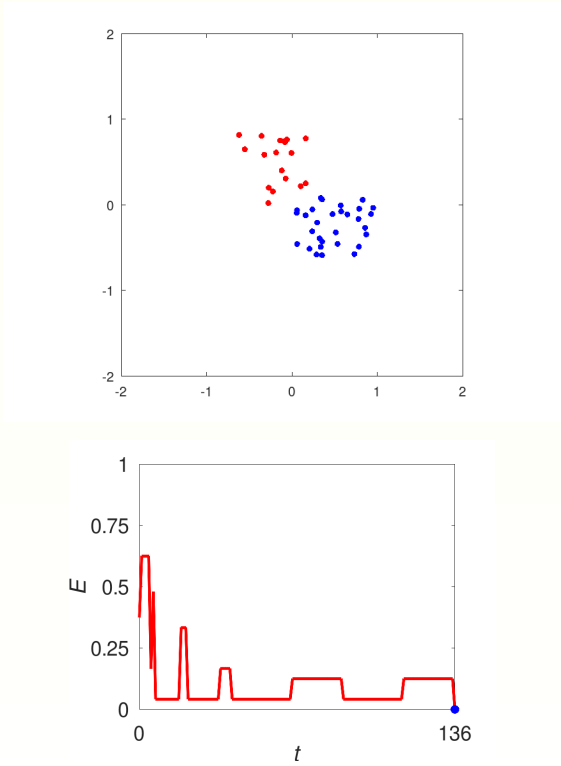
Perc–v1 | comentários (iv)

- Se D é um conjunto linearmente separável, quão mais separados estiverem os conjuntos rotulados com as *labels* $+1$ e -1, regra geral mais rapidamente o Perc–v1 converge.

exemplo 4

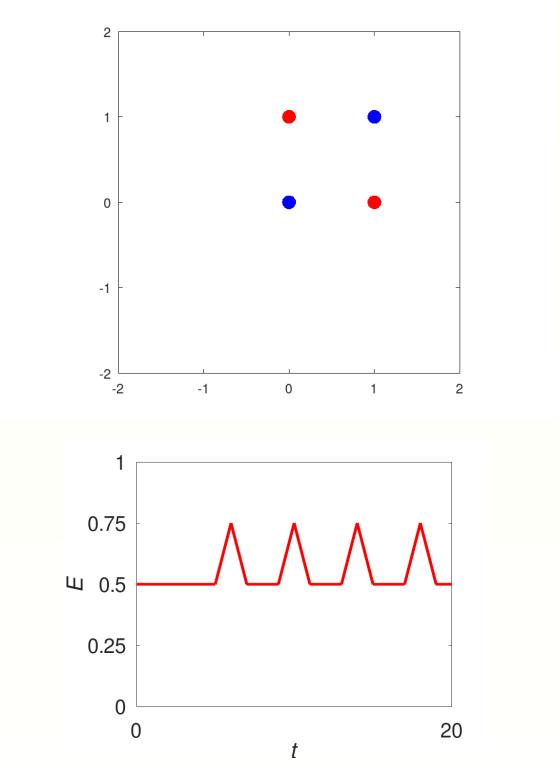


exemplo 5

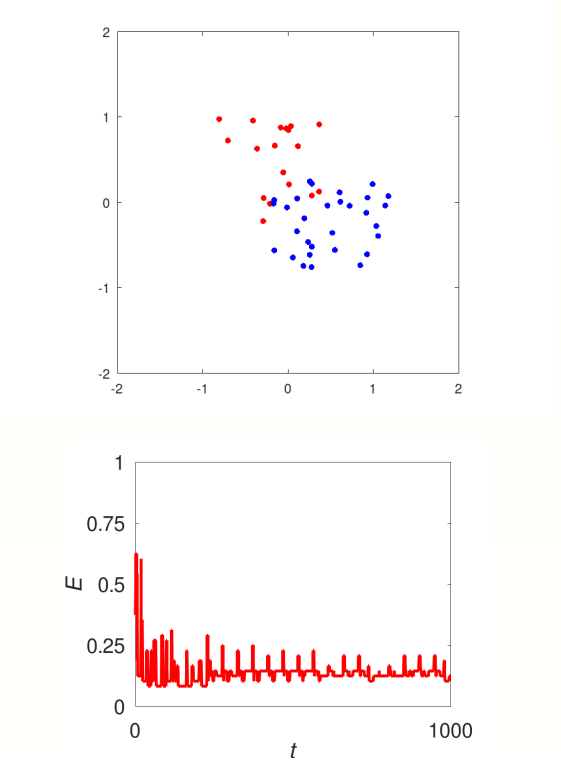


- Se D não é um conjunto linearmente separável, o Perc–v1 não converge.

exemplo 3



exemplo 6



Perc–v1 | comentários (vi)

- Atendedo a que

Perc–v1 | comentários (vi)

- Atendedo a que
  - calcular o valor da função custo em cada iteração pode ser computacionalmente pesado e

Perc–v1 | comentários (vi)

- Atendedo a que
  - calcular o valor da função custo em cada iteração pode ser computacionalmente pesado e
  - num caso real o habitual é não se saber *a priori* se D é um conjunto linearmente separável,

Perc–v1 | comentários (vi)

- Atendedo a que
  - calcular o valor da função custo em cada iteração pode ser computacionalmente pesado e
  - num caso real o habitual é não se saber *a priori* se D é um conjunto linearmente separável,
- considare-se como critério de paragem atingir um número máximo de iterações pré-estabelecido.

Perc–v1 | comentários (vi)

- Atendedo a que
  - calcular o valor da função custo em cada iteração pode ser computacionalmente pesado e
  - num caso real o habitual é não se saber *a priori* se D é um conjunto linearmente separável,
- considare-se como critério de paragem atingir um número máximo de iterações pré-estabelecido.
- Chama-se *época* a percorrer todos os elementos de D, sendo um critério de paragem habitual percorrer a base de dados T (10, 100, 1000, ...) épocas.

Perc–v1 | comentários (vii)

- A regra de aprendizagem pode ser reescrita como

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n,$$

  funcionando assim em todos os casos, pois:

Perc–v1 | comentários (vii)

- A regra de aprendizagem pode ser reescrita como

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n,$$

funcionando assim em todos os casos, pois:

- caso 1: $\hat{y}^n = y^n$ — pretende-se que $\tilde{w}_{(t+1)} = \tilde{w}_{(t)}$

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n = \tilde{w}_{(t)} + \frac{1}{2} \times 0 \times \tilde{x}^n = \tilde{w}_{(t)}.$$

Perc–v1 | comentários (vii)

- A regra de aprendizagem pode ser reescrita como

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n,$$

  funcionando assim em todos os casos, pois:
  - caso 1: $\hat{y}^n = y^n$ — pretende-se que $\tilde{w}_{(t+1)} = \tilde{w}_{(t)}$

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n = \tilde{w}_{(t)} + \frac{1}{2} \times 0 \times \tilde{x}^n = \tilde{w}_{(t)}.$$

  - caso 2: $\hat{y}^n \neq y^n$ — pretende-se que $\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + y^n\tilde{x}^n$

Perc–v1 | comentários (vii)

- A regra de aprendizagem pode ser reescrita como

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n,$$

  funcionando assim em todos os casos, pois:
  - caso 1: $\hat{y}^n = y^n$ — pretende-se que $\tilde{w}_{(t+1)} = \tilde{w}_{(t)}$

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n = \tilde{w}_{(t)} + \frac{1}{2} \times 0 \times \tilde{x}^n = \tilde{w}_{(t)}.$$

  - caso 2: $\hat{y}^n \neq y^n$ — pretende-se que $\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + y^n\tilde{x}^n$
    - caso 2a: $\hat{y}^n = -1 \neq y^n = +1$

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n = \tilde{w}_{(t)} + \frac{1}{2}(1 - (-1))\tilde{x}^n =$$
$$= \tilde{w}_{(t)} + \tilde{x}^n = \tilde{w}_{(t)} + y^n\tilde{x}^n.$$

Perc–v1 | comentários (vii)

- A regra de aprendizagem pode ser reescrita como

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n,$$

  funcionando assim em todos os casos, pois:
  - caso 1: $\hat{y}^n = y^n$ — pretende-se que $\tilde{w}_{(t+1)} = \tilde{w}_{(t)}$

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n = \tilde{w}_{(t)} + \frac{1}{2} \times 0 \times \tilde{x}^n = \tilde{w}_{(t)}.$$

  - caso 2: $\hat{y}^n \neq y^n$ — pretende-se que $\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + y^n\tilde{x}^n$
    - caso 2a: $\hat{y}^n = -1 \neq y^n = +1$

$$\begin{aligned}\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n = \tilde{w}_{(t)} + \frac{1}{2}(1 - (-1))\tilde{x}^n = \\ = \tilde{w}_{(t)} + \tilde{x}^n = \tilde{w}_{(t)} + y^n\tilde{x}^n.\end{aligned}$$

    - caso 2b: $\hat{y}^n = +1 \neq y^n = -1$

$$\begin{aligned}\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{1}{2}(y^n - \hat{y}^n)\tilde{x}^n = \tilde{w}_{(t)} + \frac{1}{2}(-1 - 1)\tilde{x}^n = \\ = \tilde{w}_{(t)} - \tilde{x}^n = \tilde{w}_{(t)} + y^n\tilde{x}^n.\end{aligned}$$

Perc–v1 | comentários (viii)

- Por forma a tentar acelerar a convergência do algoritmo, introduz-se um hiperparâmetro real $\eta \in]0, 1]$, a que se chama *taxa de aprendizagem* (*learning rate*) na regra de aprendizagem, vindo (considerando também o comentário (vi))

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{\eta}{2}(y^n - \hat{y}^n)\tilde{x}^n.$$

Perc–v1 | comentários (viii)

- Por forma a tentar acelerar a convergência do algoritmo, introduz-se um hiperparâmetro real $\eta \in]0, 1]$, a que se chama *taxa de aprendizagem* (*learning rate*) na regra de aprendizagem, vindo (considerando também o comentário (vi))

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{\eta}{2}(y^n - \hat{y}^n)\tilde{x}^n.$$

- É prática ir reduzindo o valor de η no decorrer das épocas, fazendo, por exemplo, $\eta = 10^{-1}, 10^{-2}, 10^{-3}, \ldots$.

Perc–v1 | comentários (viii)

- Por forma a tentar acelerar a convergência do algoritmo, introduz-se um hiperparâmetro real $\eta \in]0, 1]$, a que se chama *taxa de aprendizagem* (*learning rate*) na regra de aprendizagem, vindo (considerando também o comentário (vi))

$$\tilde{w}_{(t+1)} = \tilde{w}_{(t)} + \frac{\eta}{2}(y^n - \hat{y}^n)\tilde{x}^n.$$

- É prática ir reduzindo o valor de η no decorrer das épocas, fazendo, por exemplo, $\eta = 10^{-1}, 10^{-2}, 10^{-3}, \ldots$.
- É boa prática normalizar quer os vetores dos atributos da base de dados quer o vetor pos parãmetros $\tilde{w}$ (nos exemplos que se seguem tal não é feito).

Algoritmo Perc–v2

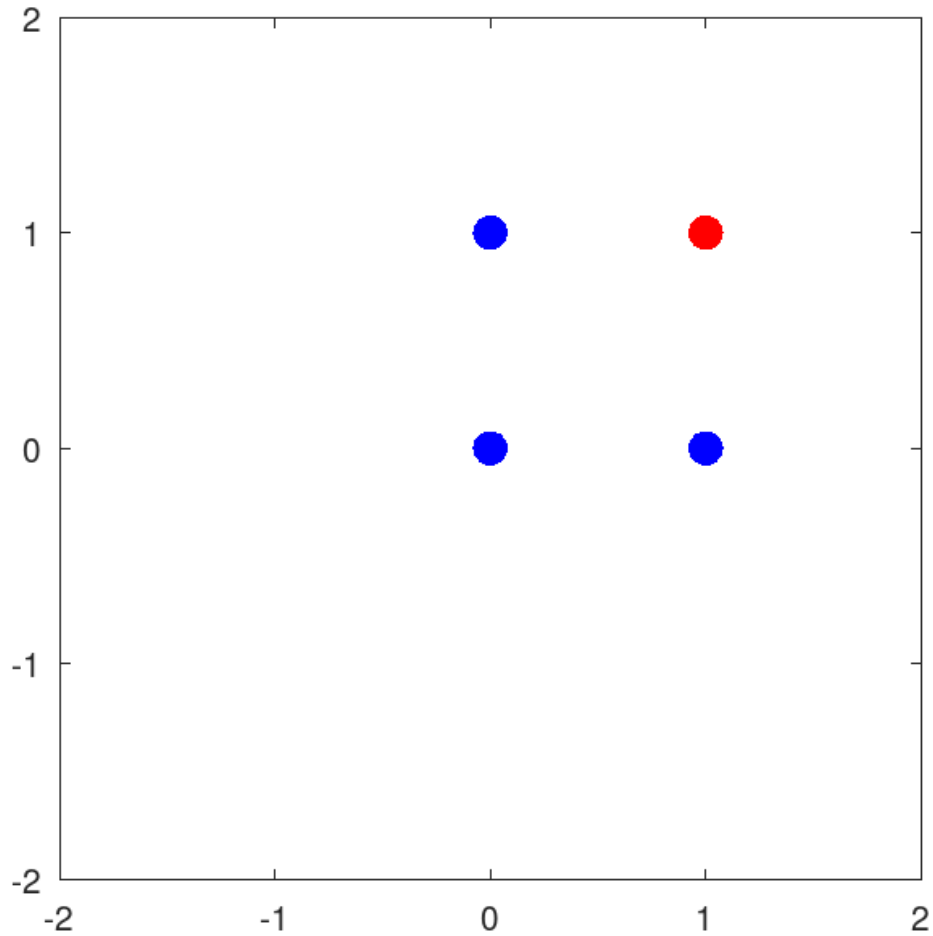
Input: $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^I$, $y^n \in \{-1, +1\}$, $\tilde{w}_{(0)} \in \mathbb{R}^{I+1}$, $T \in \mathbb{N}$, $\eta \in]0, 1]$

Output: $\tilde{w}^* \in \mathbb{R}^{I+1}$

$t \leftarrow 0$;
for $\tau \leftarrow 1$ **to** T **do**
    for $n \leftarrow 1$ **to** N **do**
        $\hat{y}^n \leftarrow \operatorname{sgn}(\tilde{w}_{(t)} \cdot \tilde{x}^n)$;
        $\tilde{w}_{(t+1)} \leftarrow \tilde{w}_{(t)} + \frac{\eta}{2}(y^n - \hat{y}^n)\tilde{x}^n$;
        $t \leftarrow t + 1$;
$\tilde{w}^* \leftarrow \tilde{w}_{(t)}$;
return $\tilde{w}^*$;

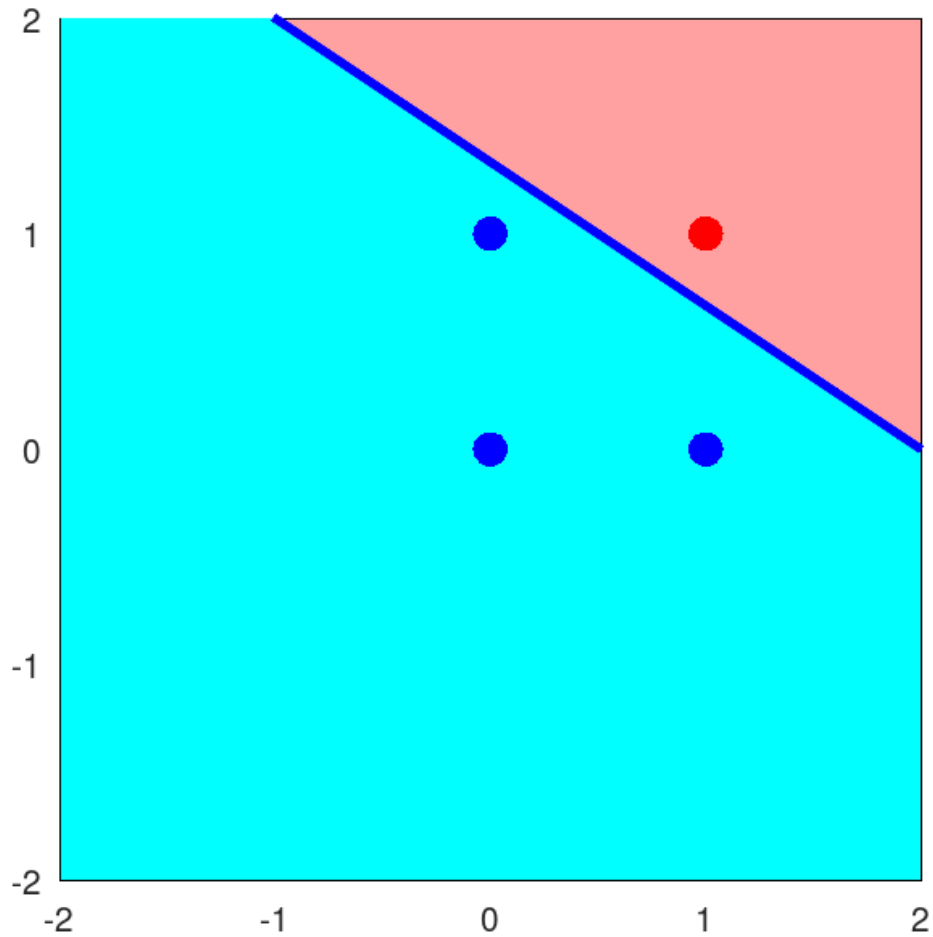
Ex1 (AND) | D

- Base de dados "AND"
- $\eta = 0.5,\ T = 5,\ \tilde{w}_{(0)} = (-0.4, -0.3, -0.2)^\top$



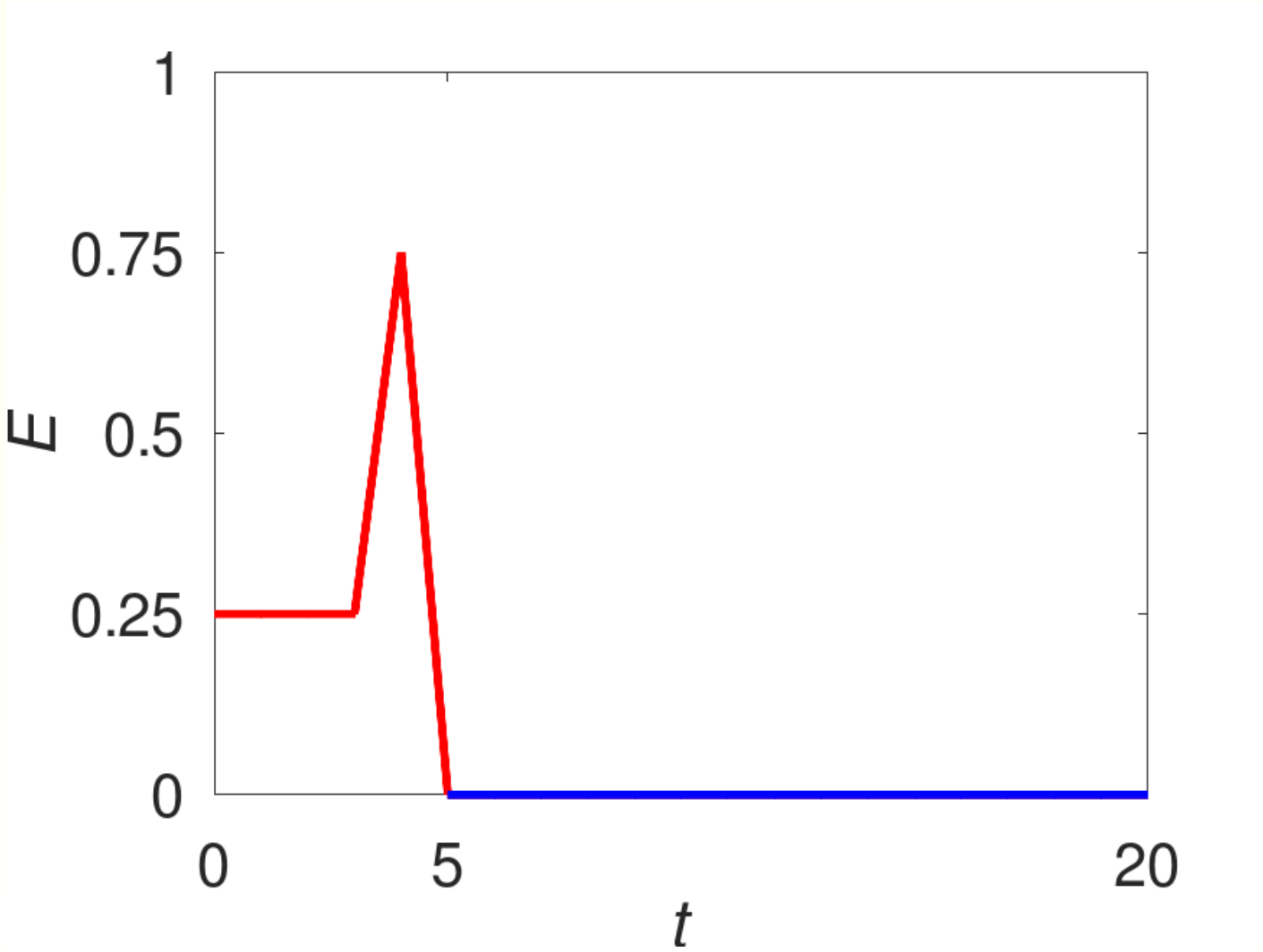
Ex1 (AND) | Perc–v2, $\eta = 0.5$, $NEp = 5$ | $t = 20$

- $\tilde{w}_{(0)} = (-0.4, -0.3, -0.2)^\top, E_{(0)} = 0.25$
- $\tilde{w}^* = (-0.4, 0.2, 0.3)^\top, E = 0$



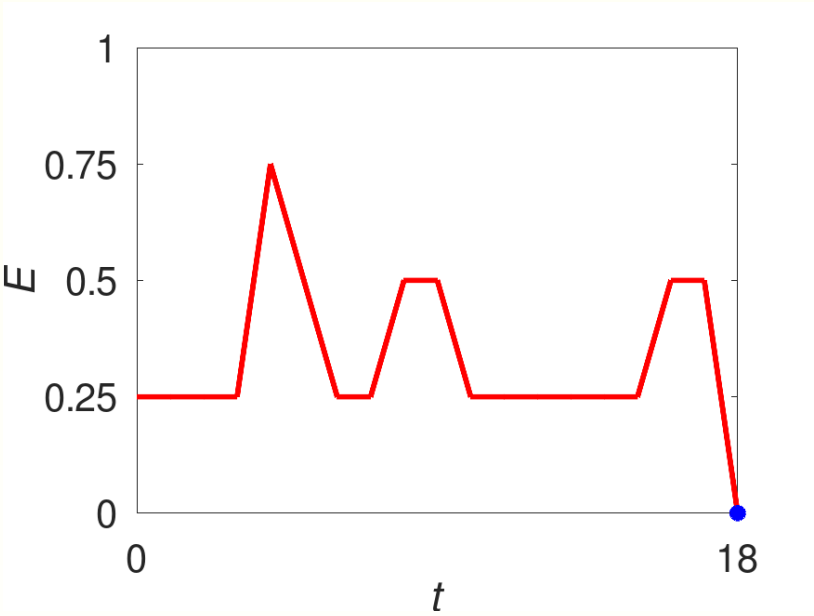
Ex1 (AND) | Perc–v2, $\eta = 0.5,\ T = 5$ | E

Ex1 (AND) | Perc–v2, $\eta = 0.5,\ T = 5$ | E

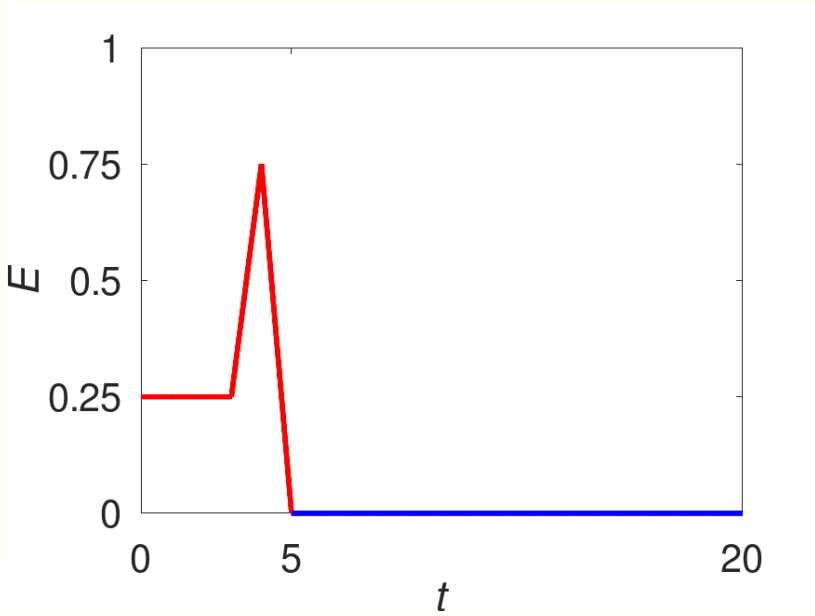


Ex1 (AND) | Perc–v1 vs Perc–v2

Perc–v1

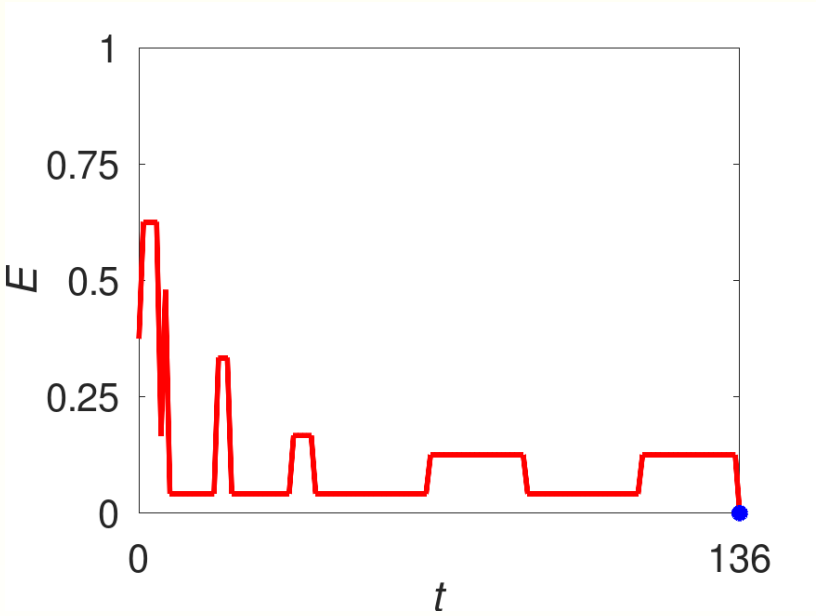


Perc–v2, $\eta = 0.5,\ T = 5$

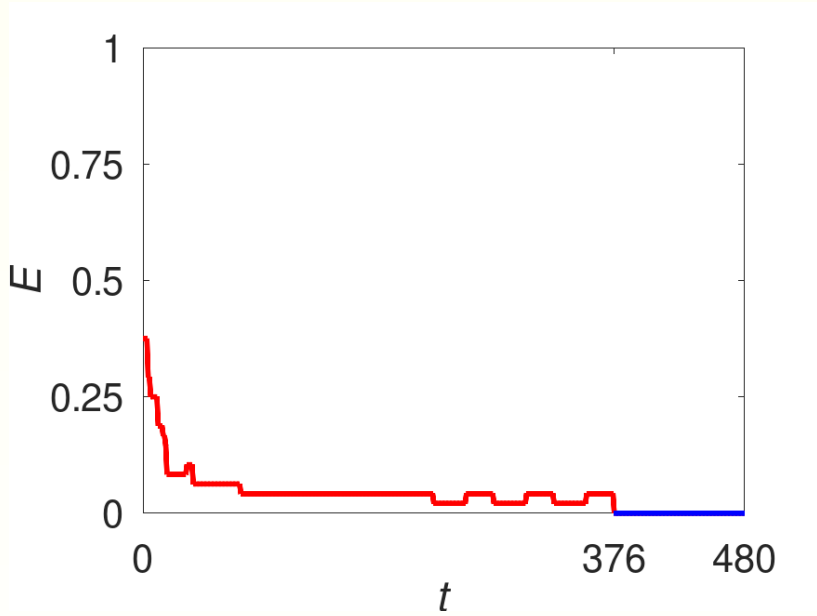


Ex5 | Perc–v1 vs Perc–v2

Perc–v1



Perc–v2, $\eta = 0.05,\ T = 10$



Exercícios

Exercício 1. Represente analiticamente e graficamente os pontos de $\mathbb{R}^2$ que são classificados como $+1$ e -1 pelo classificador $\tilde{w} = (1, 2, -1)^\top$. Represente também o vetor w na figura.

Exercício 2. Seja a base de dados $D = (x^n, y^n)_{n=1}^5$ com

$$
\begin{aligned}
x^1 &= (-0.5, 0.5)^\top & y^1 &= -1 \\
x^2 &= (0.5, 0.5)^\top & y^2 &= +1 \\
x^3 &= (1, 0.5)^\top & y^3 &= +1 \\
x^4 &= (-1, -0.5)^\top & y^4 &= -1 \\
x^5 &= (0.5, -0.5)^\top & y^5 &= +1
\end{aligned}
$$

(a) Represente graficamente D.
(b) Indique, justificando, um classificador linear cuja função custo é zero.
(c) Se os pesos forem os simétricos dos da alínea anterior, qual é o valor da função custo?

Exercício 3. Mostre que a base de dados XOR não é linearmente separável.

Exercícios

Exercício 4. Considere a base de dados do Exercício 1.

(a) Aplique o algoritmo Perc–v1 com $\tilde{w}_{(0)} = (0, 0.5, 1)^\top$.

(b) Aplique o algoritmo Perc–v2 com $\tilde{w}_{(0)} = (0, 0.5, 1)^\top$, $T = 1$ e $\eta = 0.5$.

Exercício 5. Considere a base de dados binária $D = (x^n, y^n)_{n=1}^3$ $(I = 4)$ com

$$\begin{aligned} x^1 &= (-1, 1, 0, 1)^\top \quad & y^1 &= -1 \\ x^2 &= (0, 0, 0, 2)^\top \quad & y^2 &= +1 \\ x^3 &= (1, 0, -1, 1)^\top \quad & y^3 &= +1 \end{aligned}$$

Indique, justificando, quais dos elementos de D são bem classificados pelo classificador $\tilde{w} = (1, 0, 1, 1, 2)^\top$.

Exercício 6. Implemente o algoritmo Perc–v1 e aplique-o aos exemplos 1 a 6 desta secção, testando diferentes valores para $\tilde{w}_{(0)}$ (dados no formato "$x_1^n \;\; x_2^n \;\; y^n$").

- Base de dados do exemplo 1 ($N = 4$, `Ex1_D.csv`):

0.00	0.00	−1	0.00	1.00	−1	1.00	0.00	−1	1.00	1.00	+1

- Base de dados do exemplo 2 ($N = 4$, `Ex2_D.csv`):

0.00	0.00	−1	0.00	1.00	+1	1.00	0.00	+1	1.00	1.00	+1

- Base de dados do exemplo 3 ($N = 4$, `Ex3_D.csv`):

0.00	0.00	−1	0.00	1.00	+1	1.00	0.00	+1	1.00	1.00	−1

- Base de dados do exemplo 4 ($N = 48$, `Ex4_D.csv`):

−0.17	0.62	+1	0.42	−0.37	−1	−0.17	0.62	+1	−0.13	0.55	+1
−0.24	0.33	+1	0.54	−0.16	−1	0.49	−0.18	−1	−0.26	0.26	+1
−0.22	0.55	+1	0.43	−0.42	−1	−0.16	0.40	+1	0.64	−0.15	−1
−0.30	0.65	+1	0.39	−0.42	−1	−0.40	0.57	+1	0.28	−0.16	−1
0.42	−0.09	−1	0.37	−0.15	−1	0.37	−0.28	−1	0.43	−0.34	−1
−0.19	0.45	+1	−0.05	0.37	+1	0.28	−0.35	−1	0.40	−0.23	−1
0.33	−0.19	−1	0.35	−0.38	−1	0.57	−0.18	−1	0.43	−0.09	−1
0.62	−0.41	−1	0.64	−0.21	−1	0.68	−0.26	−1	0.51	−0.29	−1
0.68	−0.30	−1	0.66	−0.10	−1	0.52	−0.35	−1	0.53	−0.13	−1
0.28	−0.17	−1	−0.26	0.35	+1	0.71	−0.18	−1	−0.29	0.54	+1
−0.44	0.66	+1	0.73	−0.14	−1	−0.07	0.36	+1	−0.04	0.64	+1
−0.20	0.63	+1	0.41	−0.32	−1	0.64	−0.37	−1	−0.15	0.63	+1

Exercícios

- Base de dados do exemplo 5 ($N = 48$, `Ex5_D.csv`):

−0.08	0.73	+1	0.34	−0.49	−1	−0.10	0.74	+1	−0.01	0.60	+1
−0.22	0.16	+1	0.58	−0.08	−1	0.47	−0.11	−1	−0.28	0.02	+1
−0.19	0.61	+1	0.35	−0.59	−1	−0.08	0.31	+1	0.79	−0.05	−1
−0.36	0.80	+1	0.29	−0.58	−1	−0.55	0.65	+1	0.06	−0.07	−1
0.34	0.08	−1	0.24	−0.05	−1	0.24	−0.31	−1	0.35	−0.43	−1
−0.12	0.40	+1	0.16	0.25	+1	0.06	−0.46	−1	0.29	−0.21	−1
0.16	−0.12	−1	0.20	−0.51	−1	0.65	−0.11	−1	0.35	0.06	−1
0.73	−0.58	−1	0.78	−0.17	−1	0.86	−0.27	−1	0.51	−0.32	−1
0.87	−0.35	−1	0.83	0.06	−1	0.53	−0.46	−1	0.57	−0.01	−1
0.05	−0.09	−1	−0.27	0.20	+1	0.93	−0.11	−1	−0.33	0.59	+1
−0.62	0.82	+1	0.95	−0.04	−1	0.10	0.22	+1	0.16	0.77	+1
−0.14	0.75	+1	0.32	−0.39	−1	0.78	−0.49	−1	−0.06	0.76	+1

- Base de dados do exemplo 6 ($N = 48$, `Ex6_D.csv`):

0.00	0.85	+1	0.25	−0.61	−1	−0.02	0.86	+1	0.12	0.66	+1
−0.21	−0.02	+1	0.61	0.01	−1	0.46	−0.04	−1	−0.29	−0.22	+1
−0.16	0.66	+1	0.28	−0.76	−1	0.01	0.21	+1	0.93	0.05	−1
−0.41	0.96	+1	0.18	−0.75	−1	−0.70	0.72	+1	−0.16	0.03	−1
0.26	0.24	−1	0.11	0.04	−1	0.10	−0.34	−1	0.28	−0.52	−1
−0.06	0.35	+1	0.36	0.12	+1	−0.16	−0.56	−1	0.19	−0.19	−1
−0.01	−0.06	−1	0.06	−0.65	−1	0.72	−0.04	−1	0.28	0.22	−1
0.85	−0.74	−1	0.92	−0.12	−1	1.03	−0.28	−1	0.52	−0.36	−1
1.05	−0.39	−1	0.99	0.21	−1	0.55	−0.56	−1	0.60	0.12	−1
−0.17	−0.01	−1	−0.29	0.05	+1	1.14	−0.04	−1	−0.36	0.63	+1
−0.81	0.97	+1	1.18	0.07	−1	0.28	0.08	+1	0.37	0.91	+1
−0.09	0.88	+1	0.24	−0.46	−1	0.92	−0.61	−1	0.04	0.89	+1

Exercícios

Exercício 7. Implemente o algoritmo Perc–v2 e aplique-o às bases de dados do **Exercício 5.**, testando diferentes valores para $\tilde{w}_{(0)}$, η e T.

Exercício 8. Seja o algoritmo Perc–v3 a variante do algoritmo Perc–v2 onde se cálcula o valor da função custo no final de cada época e memorizando o hiperplano com menor função custo.

(a) Escreva o pseudo-código do algoritmo Perc–v3.
(b) Implemente o algoritmo Perc–v3.
(c) Aplique o algoritmo Perc–v3 às bases de dados do **Exercício 5.**, testando diferentes valores para $\tilde{w}_{(0)}$, η e T.

Exercícios

Exercício 9. Seja o algoritmo Perc–v4 a variante do algoritmo Perc–v2 onde se conta o número de vezes que o hiperplano é atualizado em cada época, ou seja, o número de eventos mal classificados (*misclassified events*). Assim, nesta nova versão, o algoritmo deve terminar se se atingir o número de épocas pré-definido (o hiperparâmetro T) ou então se não houver nenhuma atualização do hiperplano numa época.

(a) Escreva o pseudo-código do algoritmo Perc–v4.
(b) Implemente o algoritmo Perc–v4.
(c) Aplique o algoritmo Perc–v4 às bases de dados do **Exercício 5.**, testando diferentes valores para $\tilde{w}_{(0)}$, η e T.

Exercício 10.

(a) Construa uma base de dados D com $I = 3$ e com as seguintes características:
- 100 ocorrências com *label* $+1$ que seguem uma distribuição uniforme na região $(x_1 - c_1)^2 + (x_2 - c_2)^2 + (x_3 - c_3)^2 \leqslant r_+^2$, com $(c_1, c_2, c_3) = (1, 1, 1)$ e $r_+ = 1$;
- 150 ocorrências com *label* -1 que seguem uma distribuição uniforme na região $(x_1 - c_1)^2 + (x_2 - c_2)^2 + (x_3 - c_3)^2 \leqslant r_-^2$, com $(c_1, c_2, c_3) = (-2, -2, -2)$ e $r_- = 2$.

(b) Faça o *shuffle* da base de dados.

(c) Divida agora a base de dados em duas partes: D_1 — uma parte para *training* (80%) e D_2 — a restante parte para *testing* (20%).

(d) Treine a *Machine Learning* considerando o algoritmo Perc–v3 e calcule o valor da função custo (*training error*).

(e) Teste a *Machine Learning* a D_2 e calcule o valor da função custo (*testing error*).

(f) Repita o processo considerando agora $r_+ = 1.5$ e $r_- = 3$.

(g) Refaça as alíneas anteriores considerando agora o algoritmo Perc–v4.