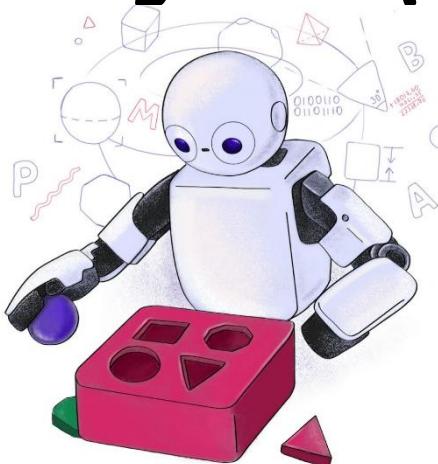


T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte II)*



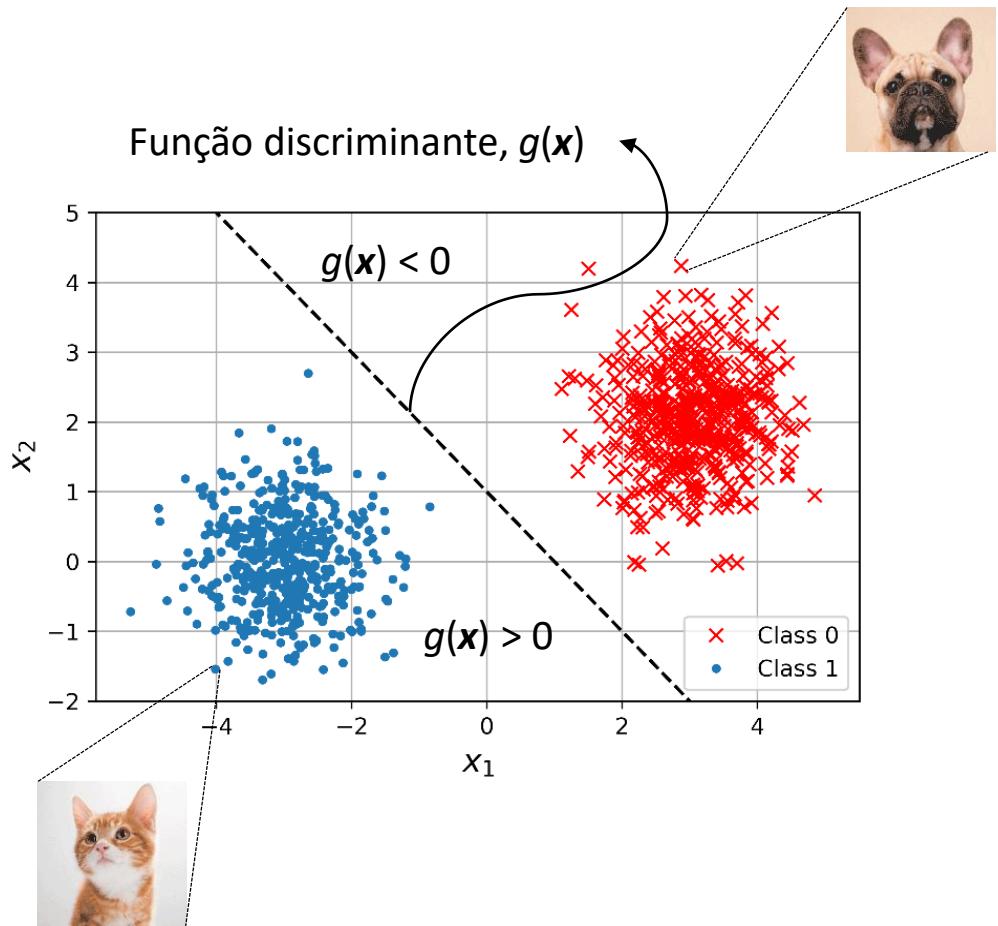
Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

Recapitulando

- Anteriormente, vimos alguns exemplos de aplicação de algoritmos de **classificação**:
 - Detecção de spam.
 - Análise de sentimentos.
 - Reconhecimento de objetos, faces, letras/dígitos.
- Definimos o problema da classificação e concluímos que ele também é um problema de **aprendizado supervisionado**.
- Aprendemos que as classes são separadas através de **funções discriminantes** e que o desafio é encontrar funções adequadas e seus respectivos pesos.
- A partir desta aula, começaremos a discutir como encontrar os pesos.

Classificação linear



- Como vimos, o objetivo da **classificação** é usar vetores de atributos, \mathbf{x} , de, por exemplo, um e-mail ou uma imagem, para **identificar** a qual classe ele pertence.
- Um **classificador linear** atinge esse objetivo **tomando uma decisão** (e.g., os **ifs** e **elses**) **com base** no valor de **combinações lineares dos atributos em relação aos pesos**, i.e., na saída de uma ou mais **funções discriminantes lineares**.

Classificação linear

- Portanto, a saída de um **classificador linear** é dada por

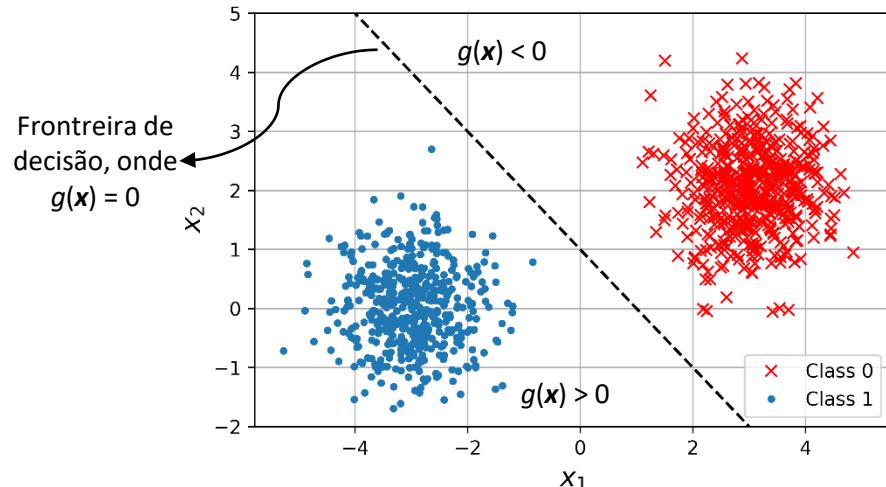
$$\hat{y} = h_a(x) = f(g(x)) = f\left(\sum_{k=0}^K a_k x_k\right) = f(a^T x),$$

onde $h_a(x)$ é conhecida como **função hipótese de classificação**, $x = [1, x_1, \dots, x_K]^T$ é o vetor de atributos com o primeiro elemento sendo o atributo de *bias*, $x_0 = 1$, e $f(\cdot)$ é uma **função de limiar de decisão**.

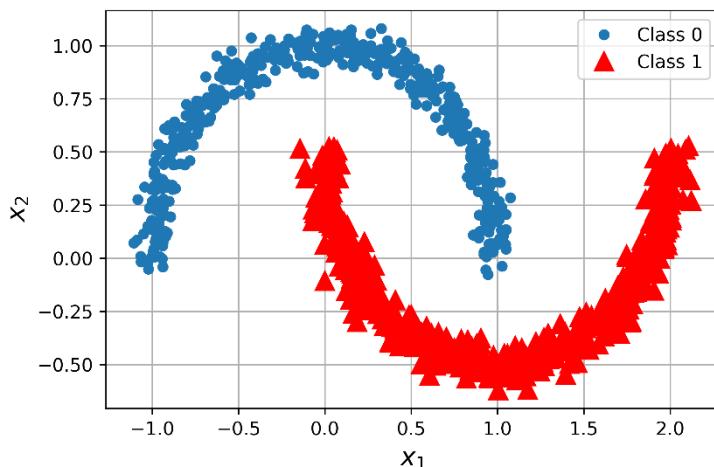
- **Função de limiar de decisão** é uma função que mapeia a saída da **função discriminante linear**, $g(x)$, na saída desejada, ou seja, na classe C_q , $q = 1, \dots, Q$, do objeto.
- Ela é apenas uma **formalização matemática** para os *ifs* e *elses* que usamos para decidir as classes dos atributos.
- Na **teoria original** dos classificadores lineares, as **funções discriminantes** seguiam equações de **hiperplanos**: $\sum_{k=0}^K a_k x_k$.

produto
escalar

Classes linearmente separáveis



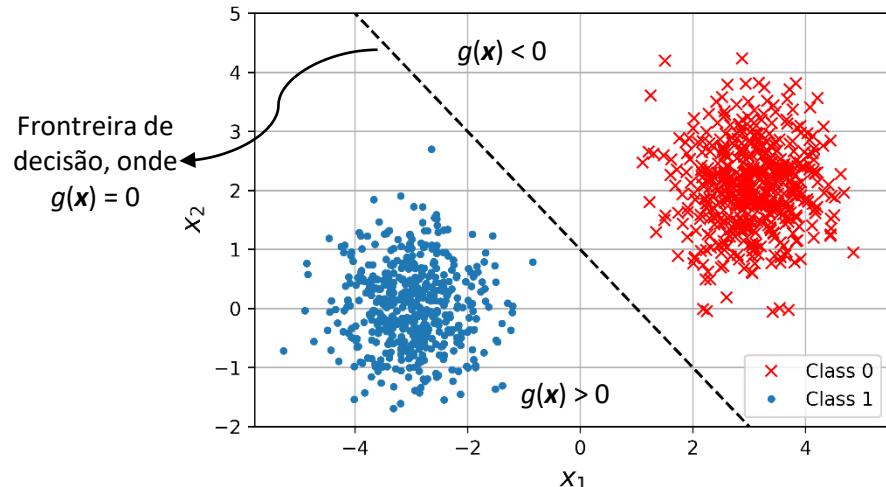
Classes linearmente separáveis.



Classes não-linearmente separáveis.

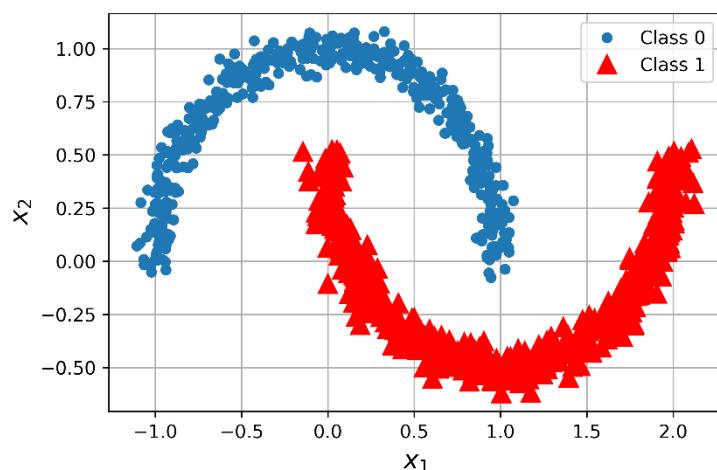
- Dado um **conjunto de treinamento**, a tarefa do **classificador** é a de **aprender** uma **função hipótese de classificação**, $h_a(x)$, que receba um exemplo de entrada (e.g., x_1 e x_2) e retorne a classe do exemplo.
- Para que um **classificador linear** aprenda uma boa separação, as classes devem ser **linearmente separáveis**.
- Isso significa que as classes devem ser **suficientemente separadas** umas das outras para garantir que a **superfície de decisão** seja um **hiperplano**.

Classes linearmente separáveis



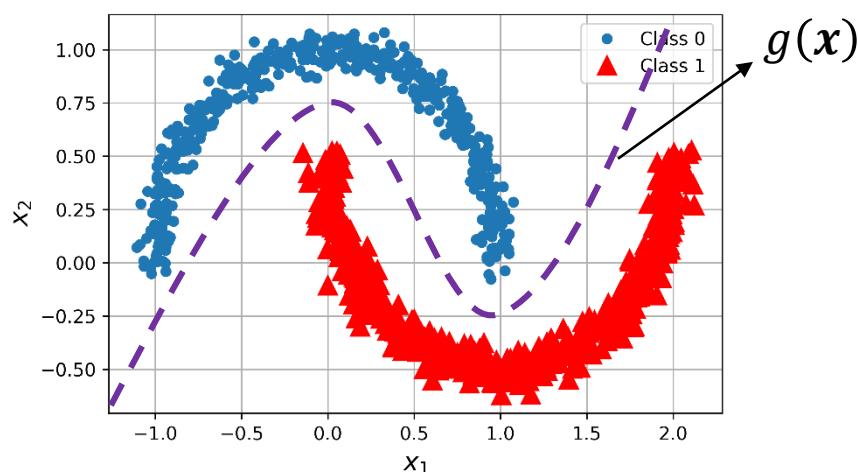
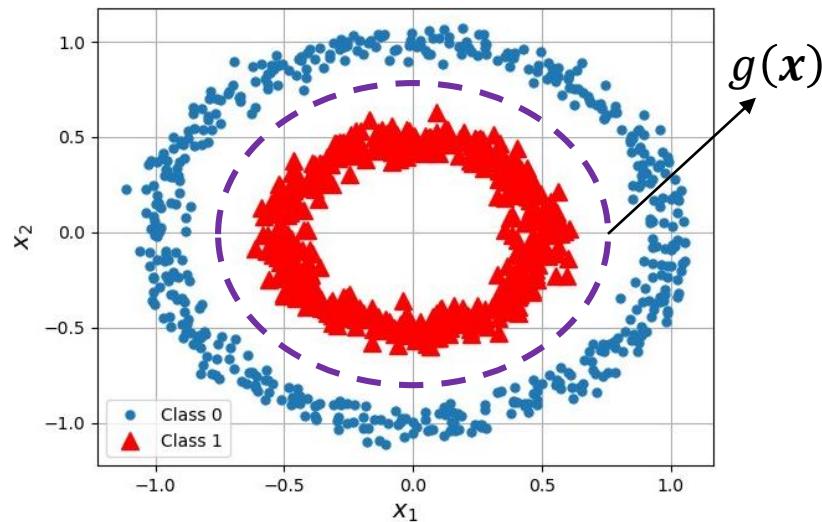
Classes linearmente separáveis.

- Classes que podem ser separadas por um **hiperplano** são chamadas de **linearmente separáveis**.
- Na primeira figura, a **fronteira de decisão** é definida por um **hiperplano**, i.e., uma reta.
- Na segunda figura, devido à disposição das classes, não existe um **hiperplano** que as separe.
- **Originalmente, classificação linear** é usada quando as classes podem ser separadas por **hiperplanos**: $\sum_{k=0}^K a_k x_k$.



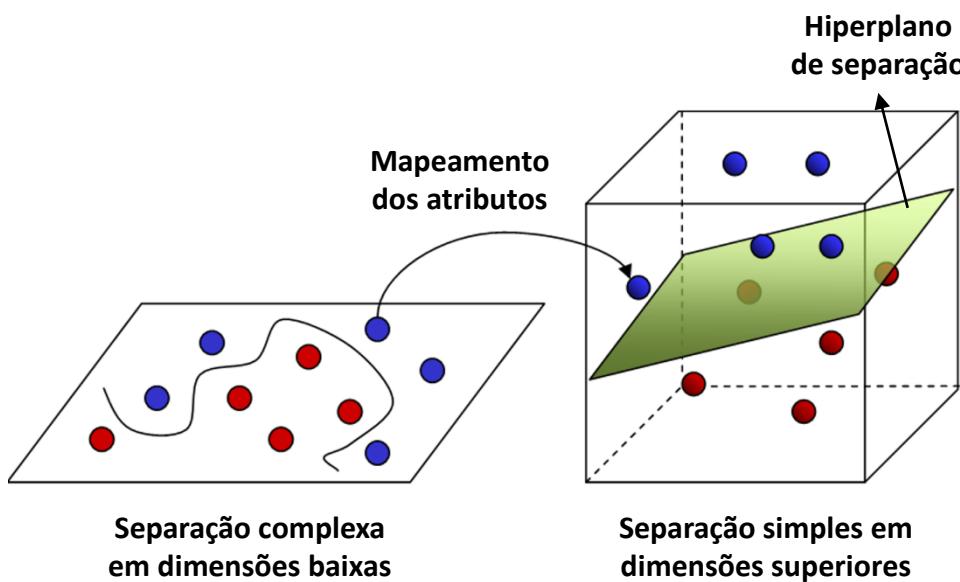
Classes não-linearmente separáveis.

Classificação não-linear



- Mas e se não pudermos separar as classes com um *hiperplano*, ou seja, se elas não forem **linearmente separáveis**?
- Nestes casos, usamos **polinômios**.
- Os polinômios podem fazer **mapeamentos não lineares** dos atributos no valor de saída:
 - $g(x) = (x_1 - a)^2 + (x_2 - b)^2 - r^2$, Círculo centrado em (a, b) e com raio r .
 - $g(x) = \frac{(x_1-a)^2}{c^2} + \frac{(x_2-b)^2}{d^2} - 1$, Elipse centrada em (a, b) , com largura $2c$ e altura $2d$.
 - $g(x) = (x_1 - a)(x_2 - b) - c$, Hipérbole retangular com eixos paralelos às suas assíntotas.

Classificação não-linear

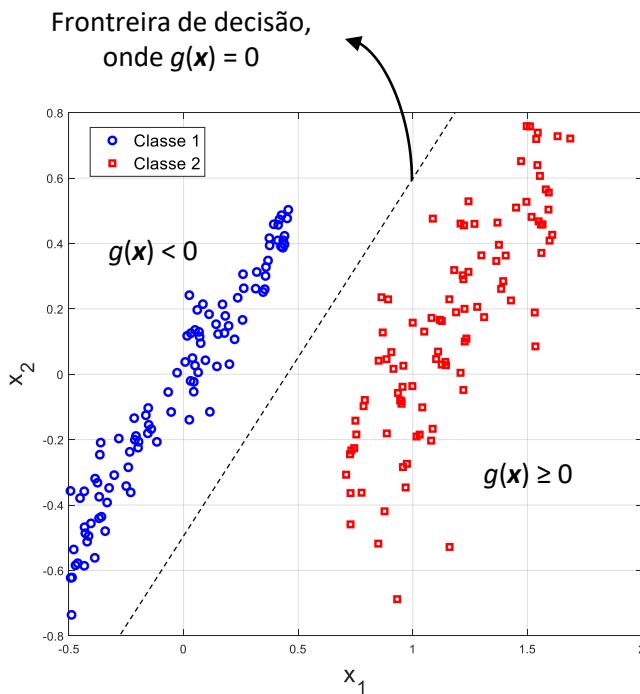


- Os polinômios podem aplicar **transformações não-lineares aos atributos originais**, levando ao **aumento das dimensões de entrada**, e.g.:

$$\begin{aligned}g(x) &= (x_1 - a)^2 + (x_2 - b)^2 - r^2 \\&= x_1^2 - 2ax_1 + x_2^2 - 2bx_2 + (a^2 + b^2 - r^2) \\&= a_1z_1 + a_2z_2 + a_3z_3 + a_4z_4 + a_0\end{aligned}$$

- O polinômio **cria novas coordenadas** no espaço de atributos.
- As transformações **mapeiam** os atributos para um **espaço de maior dimensão** onde a separação pode se tornar **mais fácil**, até linear.

Função de limiar de decisão

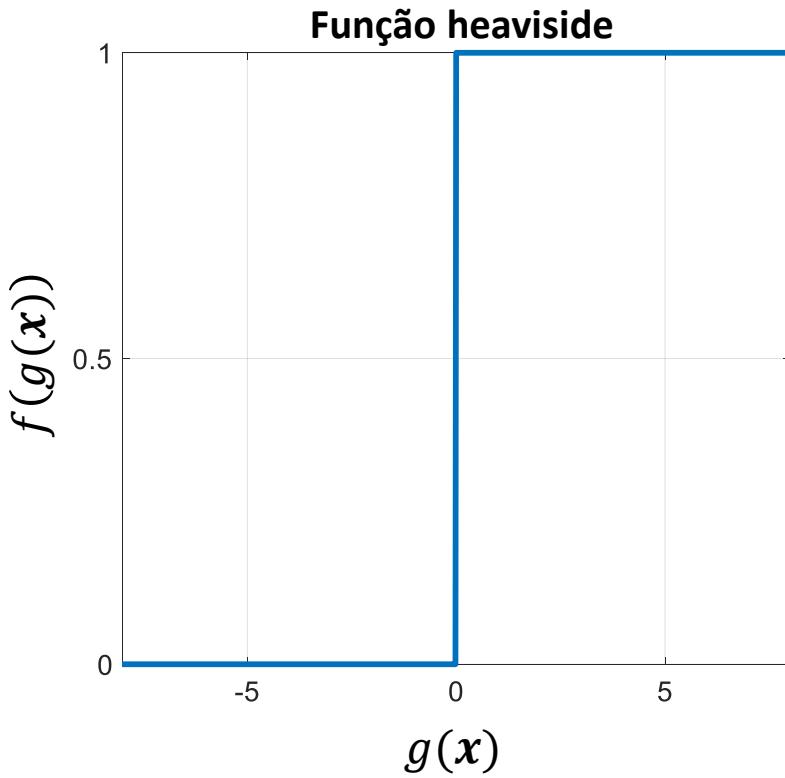


- Para o exemplo ao lado, podemos definir a **função hipótese de classificação** como duas **condições**:

$$\hat{y} = h_a(x) = \begin{cases} 0, & g(x) = x^T a < 0 \text{ (Classe -)} \\ 1, & g(x) = x^T a \geq 0 \text{ (Classe +)} \end{cases} \quad \left. \begin{array}{l} \text{if } e \\ \text{else} \end{array} \right.$$

- Percebam que a saída da **função** é **binária**.
- O mapeamento entre o valor da função discriminante, $g(x)$, e a saída 0 ou 1 é feito através da **função de limiar de decisão**, $f(g(x))$.
 - No caso acima ela é feita por uma **estrutura condicional**.
- Como implementar essas **condições** através de uma função matemática?

Função de limiar de decisão rígido

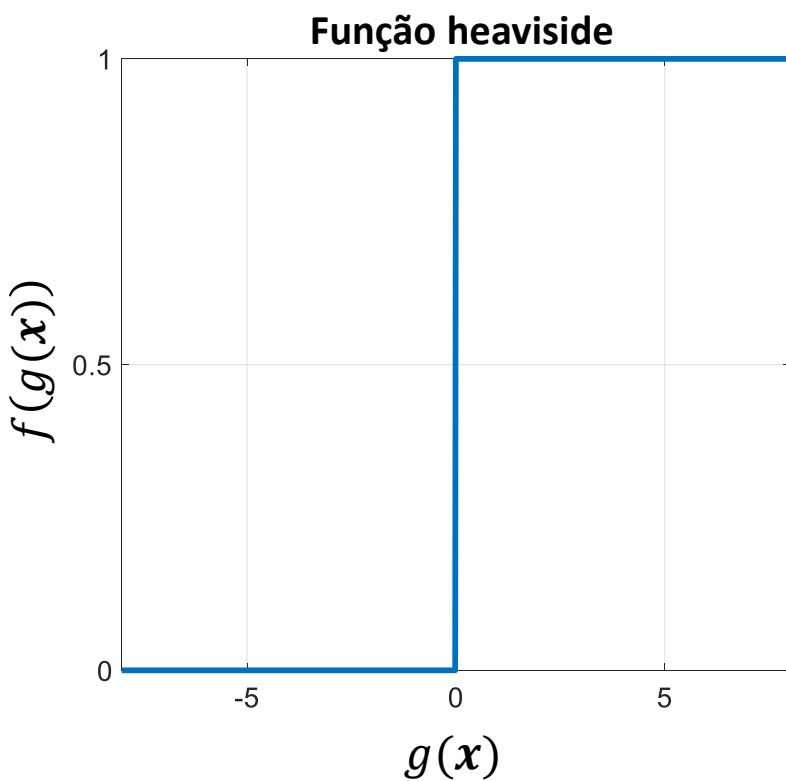


- Uma *função de limiar de decisão* simples que mapeia o valor de $g(x)$ em 2 valores de saída é a **função de limiar de decisão rígido**.
- Ela é mostrada na figura ao lado e é definida como

$$f(g(x)) = \begin{cases} 0, & g(x) < 0 \\ 1, & g(x) > 0 \\ \text{Indeterminado,} & g(x) = 0 \end{cases}$$

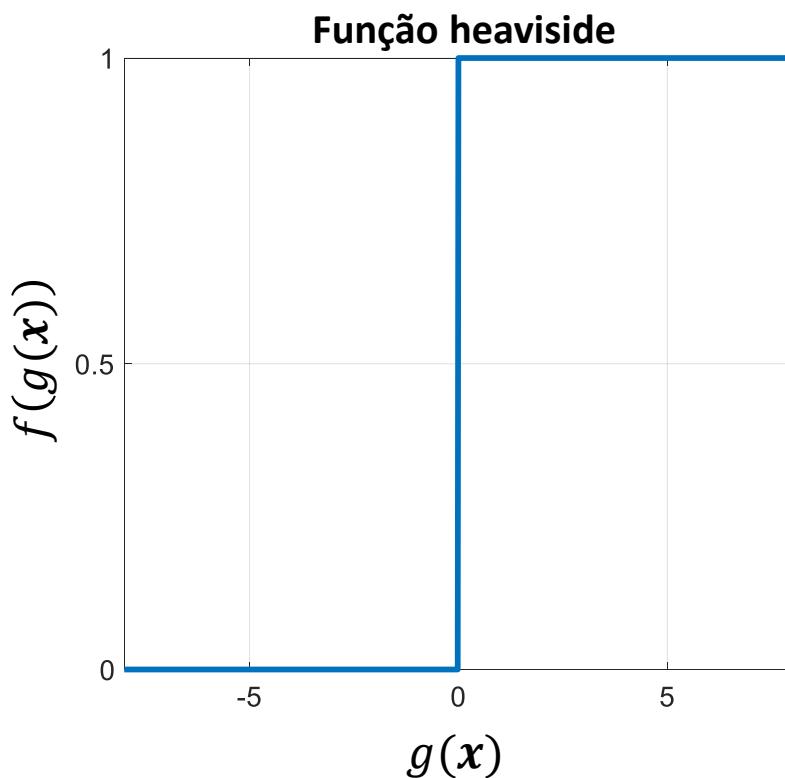
Conhecida também
como **função heaviside**
ou degrau unitário.

Classificação com limiar de decisão rígido



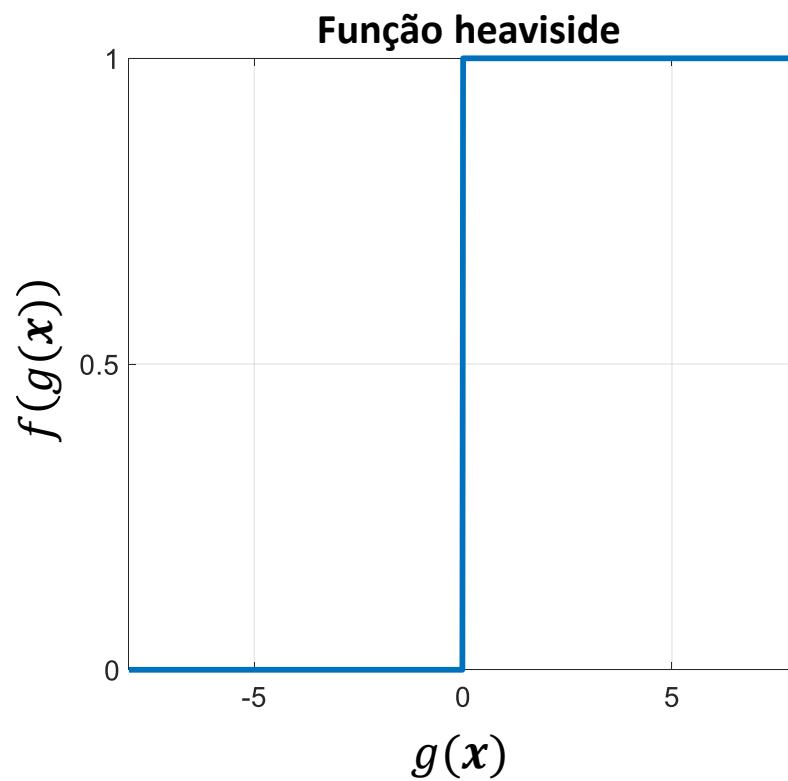
- Agora que a **função hipótese de classificação**, $h_a(x)$, tem uma **forma matemática bem** definida, precisamos pensar em **como encontrar os pesos**, a .

Classificação com limiar de decisão rígido



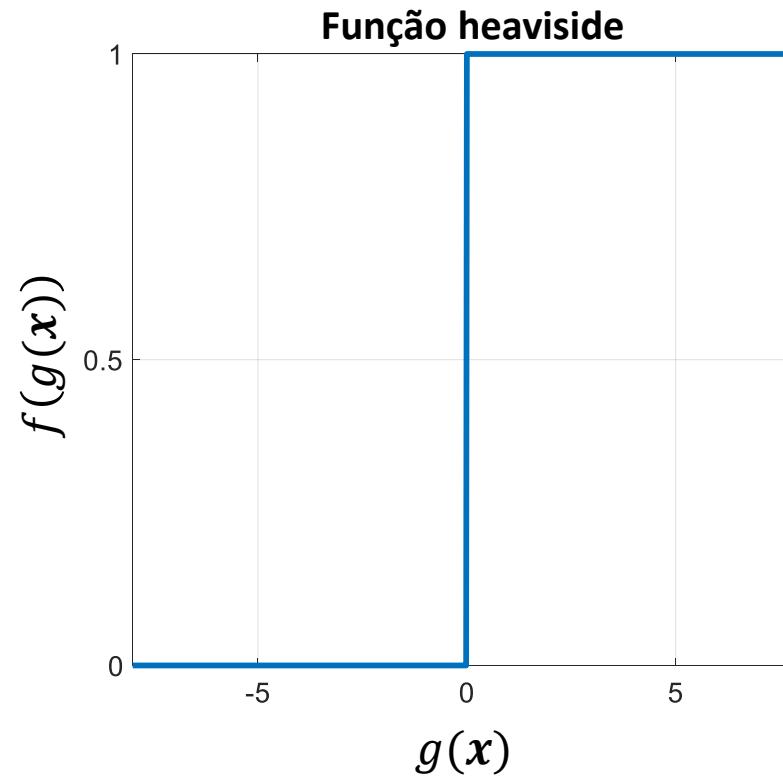
- Queremos encontrar os pesos de tal forma que ***o erro de classificação seja minimizado***, i.e., que os exemplos sejam classificados corretamente.
- No caso da ***regressão linear***, nós fizemos isso de duas maneiras:
 - i. de forma fechada (com a ***equação normal***) tomando a derivada parcial do erro em relação aos pesos, igualando a zero e resolvendo a equação em relação aos pesos;
 - ii. e através do algoritmo do ***gradiente descendente***, com as derivadas parciais do erro em função aos pesos.

Classificação com limiar de decisão rígido



- Entretanto, com a **função de limiar rígido**, **nenhuma das duas abordagens é possível** devido à **derivada** de $f(g(x))$ ser igual a zero em todos os pontos exceto em $g(x) = 0$, onde ela é indeterminada.

Classificação com limiar de decisão rígido



Como encontramos os pesos dada essa limitação?

Classificação com limiar de decisão rígido

- Uma possível solução é utilizar uma *regra intuitiva* de atualização dos *pesos* que *converge para uma solução dado que exista uma função discriminante adequada e que as classes não se sobreponham*.
- Essa *regra intuitiva de atualização dos pesos* é dada pela seguinte equação

$$\mathbf{a} = \mathbf{a} + \alpha \left(y(i) - h_{\mathbf{a}}(\mathbf{x}(i)) \right) \mathbf{x}(i), \forall i,$$

onde α é o passo de aprendizagem, o qual deve ser sempre maior do que zero.

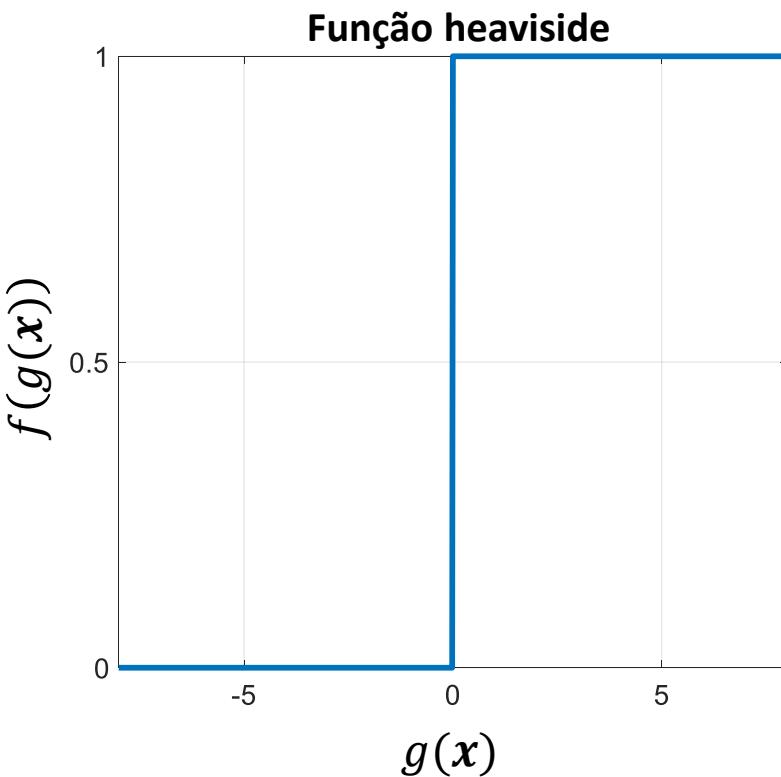
- A regra é idêntica à regra de atualização dos pesos para a *regressão linear* quando utilizamos o *gradiente descendente estocástico (GDE)*.

Classificação com limiar de decisão rígido

$$\mathbf{a} = \mathbf{a} + \alpha (y(i) - h_{\mathbf{a}}(\mathbf{x}(i))) \mathbf{x}(i), \forall i.$$

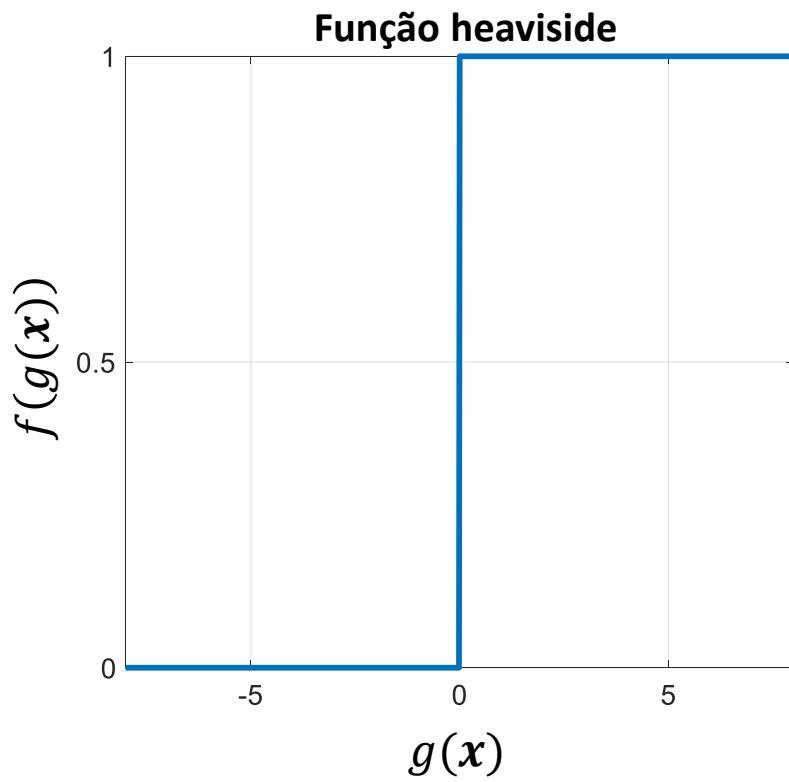
- Por razões que discutiremos mais adiante, esta regra é chamada de ***regra de aprendizagem do perceptron***.
- Os pesos são atualizados usando-se apenas ***um exemplo por vez***, escolhido de forma ***aleatória*** do conjunto de treinamento, assim como fizemos com o GDE.
- Como classificadores binários têm rótulos e valores de saída iguais a 0 ou 1, o comportamento da regra de atualização é diferente do comportamento do GDE.
- Como veremos a seguir, existem apenas 3 possibilidades para a regra.

Classificação com limiar de decisão rígido



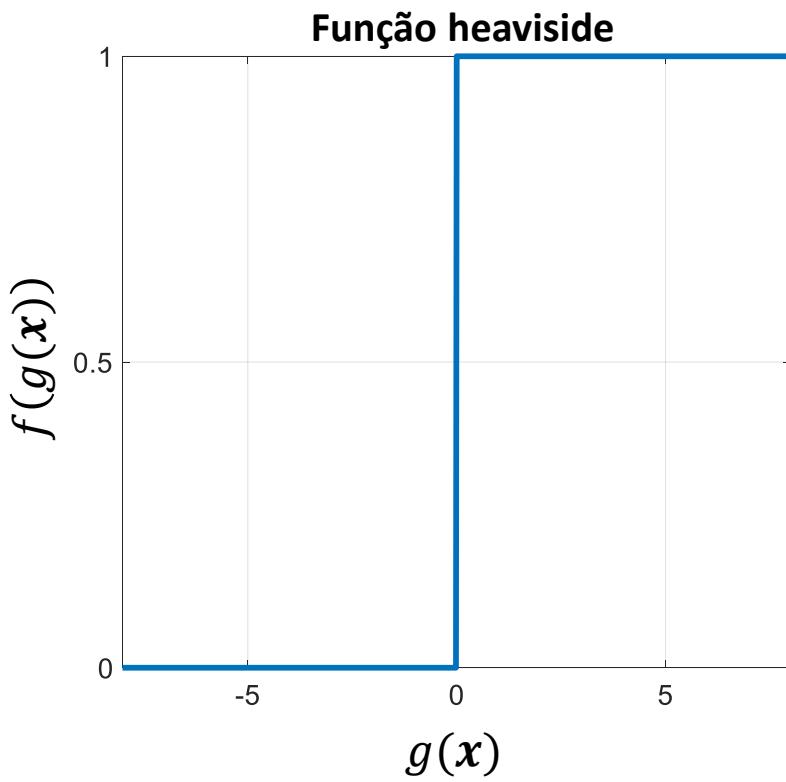
- Primeira possibilidade
 - Se o valor de saída do classificador for igual ao esperado, i.e., $h_a(x(i)) = y(i)$, então $y(i) - h_a(x(i)) = 0$.
- Observem que se na equação de atualização dos pesos
$$a = a + \alpha (y(i) - h_a(x(i))) x(i),$$
$$y(i) - h_a(x(i)) = 0,$$
 então, **os pesos não são atualizados.**

Classificação com limiar de decisão rígido



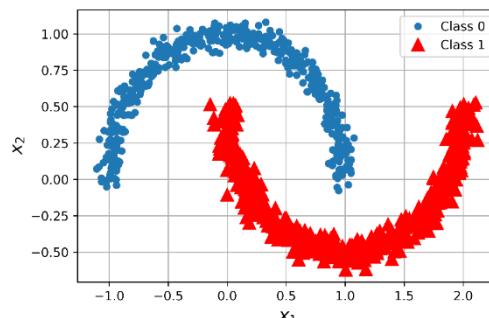
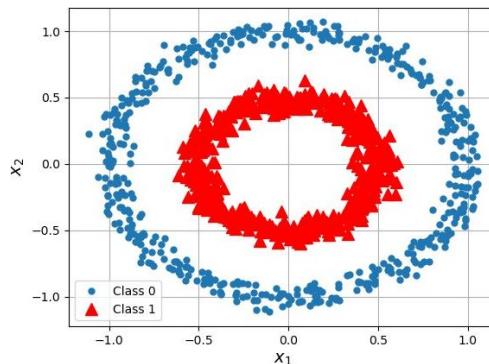
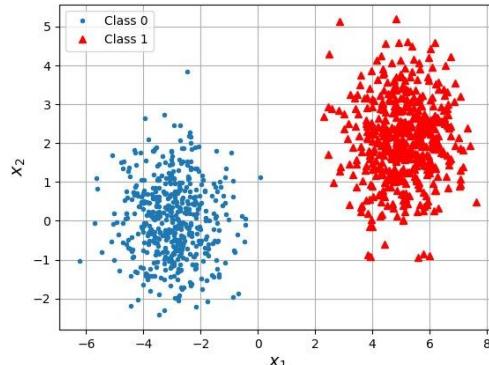
- Equação de atualização dos pesos
$$\mathbf{a} = \mathbf{a} + \alpha (y(i) - h_{\mathbf{a}}(\mathbf{x}(i))) \mathbf{x}(i).$$
- Segunda possibilidade
 - Se $y(i) = 1$, mas $h_{\mathbf{a}}(\mathbf{x}(i)) = 0$, então
$$\mathbf{a} = \mathbf{a} + \alpha \mathbf{x}(i).$$
 - Assim, o k -ésimo peso, a_k , tem seu valor **aumentado** quando o valor de $a_k \times x_k$ é positivo e **diminuído** quando o valor de $a_k \times x_k$ é negativo.
 - Isso faz sentido, pois nós queremos **aumentar** o valor de $g(\mathbf{x})$, de tal forma que $g(\mathbf{x}) > 0$ e, consequentemente, $h_{\mathbf{a}}(\mathbf{x})$ tenha como saída o valor 1.
 - Lembrando que $g(\mathbf{x}) = a_0 + a_1 x_1 + \dots + a_K x_K$.

Classificação com limiar de decisão rígido



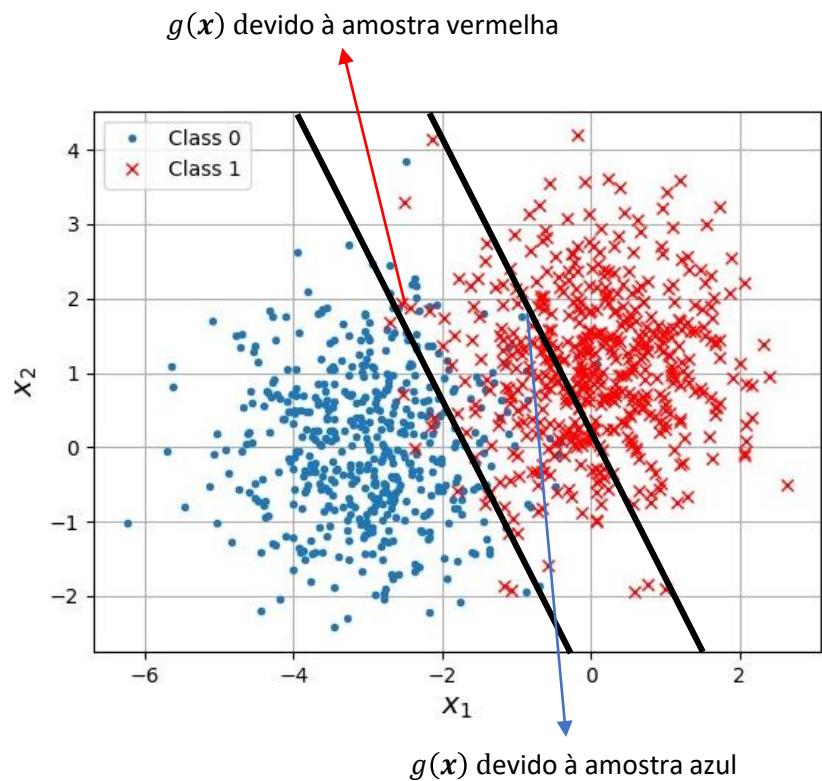
- Equação de atualização dos pesos
$$\mathbf{a} = \mathbf{a} + \alpha (y(i) - h_{\mathbf{a}}(\mathbf{x}(i))) \mathbf{x}(i).$$
- Terceira possibilidade
 - Se $y(i) = 0$, mas $h_{\mathbf{a}}(\mathbf{x}(i)) = 1$, então
$$\mathbf{a} = \mathbf{a} - \alpha \mathbf{x}(i).$$
 - Assim, o k -ésimo peso, a_k , tem seu valor **diminuído** quando o valor de $a_k \times x_k$ é positivo e **aumentado** quando o valor de $a_k \times x_k$ é negativo.
 - Isso faz sentido, pois nós queremos **diminuir** o valor de $g(\mathbf{x})$, de tal forma que $g(\mathbf{x}) < 0$ e, consequentemente, $h_{\mathbf{a}}(\mathbf{x})$ tenha como saída o valor 0.
 - Lembrando que $g(\mathbf{x}) = a_0 + a_1 x_1 + \dots + a_K x_K$.

Classificação com limiar de decisão rígido



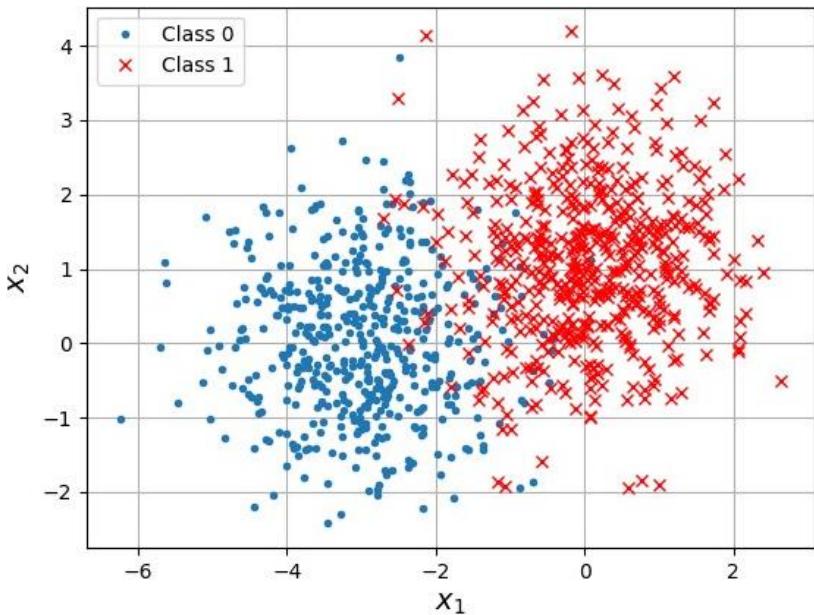
- A *regra de aprendizagem do perceptron* converge para um *separador perfeito* quando:
 - As classes são *suficientemente separadas* umas das outras, ou seja, não se sobrepõem.
 - E existe uma *função discriminante adequada para o problema*, mesmo que não seja um *hiperplano*.
 - ✓ Ou seja, não precisa ser um problema linearmente separável.
- *Separador perfeito*: com erro de classificação igual a zero, ou seja, todos os exemplos são perfeitamente classificados.
- Porém, na prática essa situação não é muito comum.

Classificação com limiar de decisão rígido



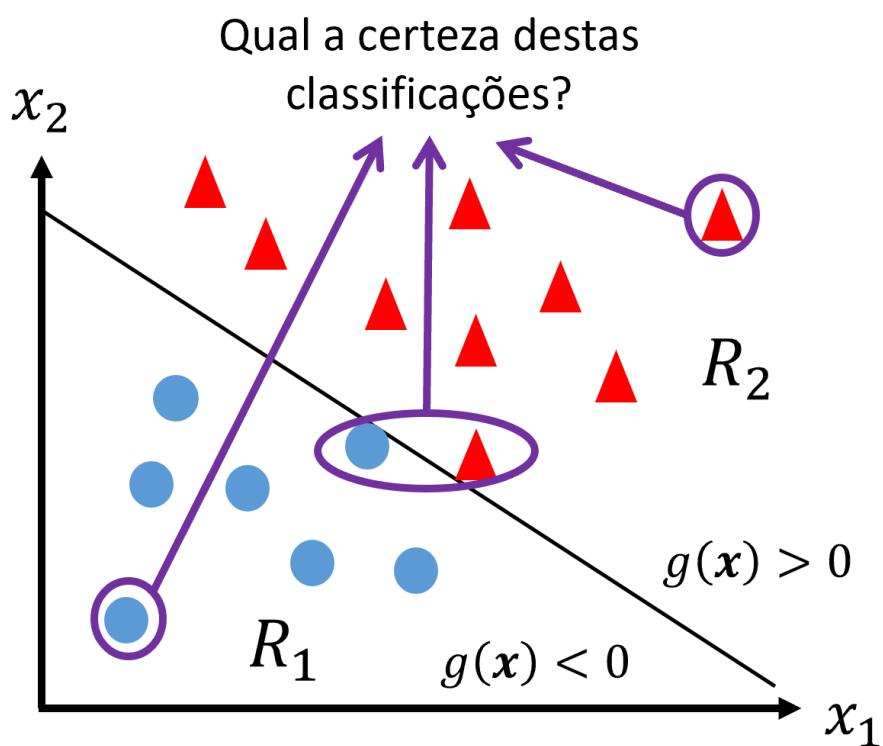
- Quando as classes se sobrepõem, a **regra de aprendizagem do perceptron** falha em convergir para uma solução perfeita.
- Nesse caso, a regra não converge para uma solução **estável** para **valores constantes do passo de aprendizagem**, α , assim como acontece com o GDE.
- Não há convergência, pois o objetivo é encontrar um erro de igual a 0.
- Além disso, os pesos são ajustados para uma única amostra por vez.

Classificação com limiar de decisão rígido



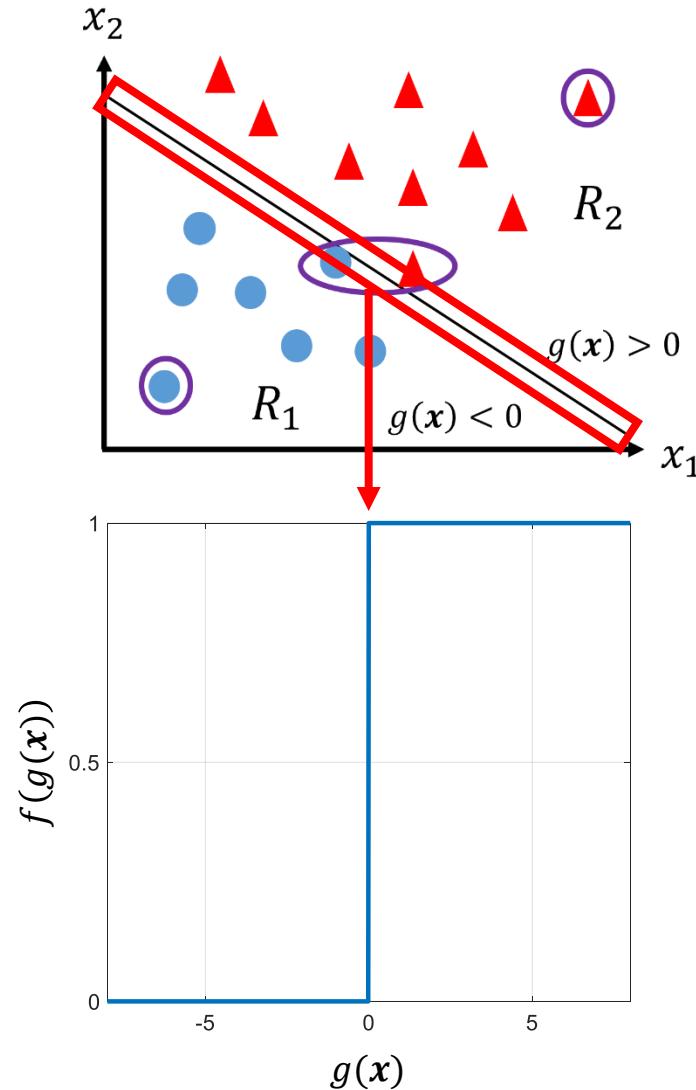
- Porém, se *α decrescer de acordo com as iterações de treinamento*, então a regra tem uma chance de convergir para uma solução de **erro mínimo** quando os exemplos são apresentados de forma aleatória.
 - Abordagem similar a que usamos com o GDE.
- Podemos também usar o **early-stop** e *guardar* os **pesos** que resultaram no **menor erro de validação**.

Classificação com limiar de decisão rígido



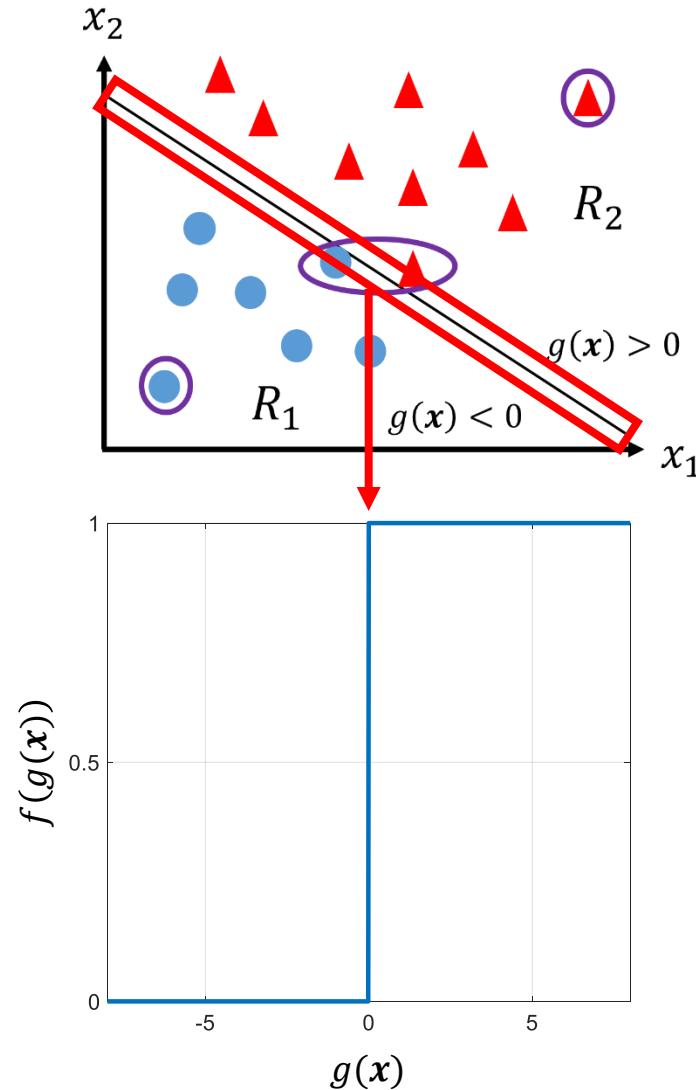
- Outro problema com classificadores que usam **limiar de decisão rígido** é a **falta de informação sobre a confiança** quanto a uma classificação.
- Na figura ao lado, dois exemplos estão bem próximos da **fronteira de decisão** enquanto outros dois estão bem distantes dela.
- Como o classificador com **limiar de decisão rígido** classificaria esses exemplos?

Classificação com limiar de decisão rígido



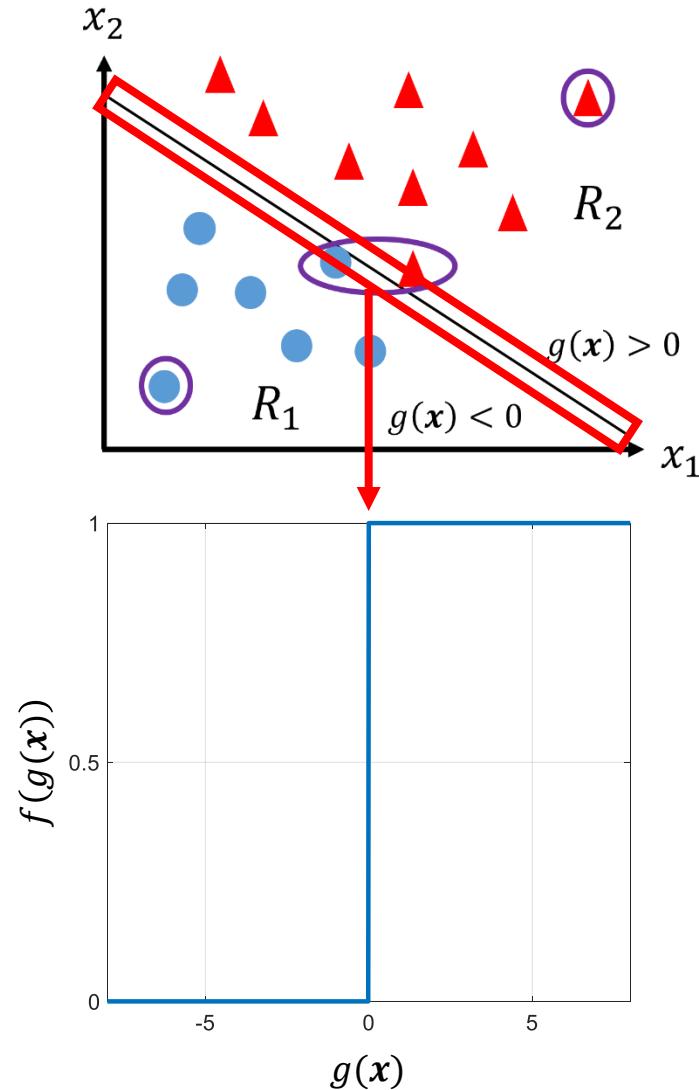
- Olhando para a função de **limiar de decisão rígido**, percebemos que o classificador faz previsões **muito confiantes** sempre iguais 0 para $g(x) < 0$ e iguais a 1 quando $g(x) > 0$, **independente se o exemplo está distante ou próximo da fronteira de decisão**.

Classificação com limiar de decisão rígido



- Exemplos *mais distantes da fronteira* têm uma *probabilidade maior* de *realmente pertencerem à classe da região onde se encontram e não serem outliers*.
 - Quanto maior o valor absoluto de $g(x)$, mais distante da fronteira está o exemplo.
- Por outro lado, exemplos próximos da fronteira têm probabilidade menor de pertencerem à classe onde estão.
 - Quanto menor o valor absoluto de $g(x)$, mais próximo da fronteira está o exemplo.

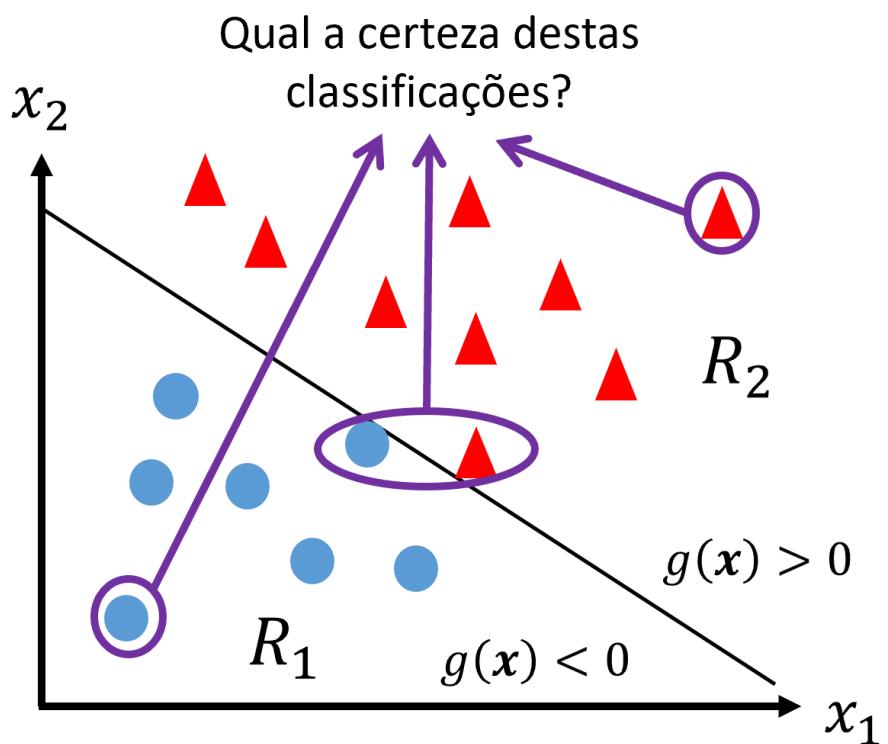
Classificação com limiar de decisão rígido



- Assim, usando **limiar de decisão rígido**, os dois **pontos azuis** são classificados como pertencentes à **classe negativa** (valor 0) e os dois **triângulos vermelhos** classificados como pertencentes à **classe positiva** (valor 1), mesmo tendo valores **absolutos de $g(x)$ bem diferentes**.

- Pontos muito próximos da fronteira de decisão têm valor absoluto de $g(x)$ próximo de zero.
- Já pontos muito distantes têm valor absoluto de $g(x)$ muito maior do que zero.

Classificação com limiar de decisão rígido



- Em resumo, *pontos distantes da fronteira* de decisão *deveriam ter uma confiança* (ou probabilidade) de *pertencerem a uma determinada classe bem maior do que pontos próximos*.
- Porém, isso não é refletido na saída do classificador com limiar de decisão rígido.
- Entretanto, em muitas situações, nós precisamos de previsões mais graduadas, que indiquem incertezas quanto à previsão.

Tarefas

- **Quiz:** “*T320 - Quiz - Classificação (Parte II)*” que se encontra no MS Teams.
- **Exercício Prático:** Laboratório #2.
 - Pode ser acessado através do link acima (Google Colab) ou no GitHub.
 - Se atentem aos prazos de entrega.
 - Instruções para resolução e entrega dos laboratórios.

Obrigado!

