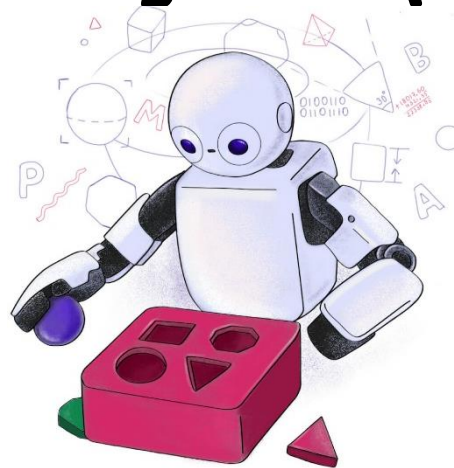


T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte II)*



Inatel

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

Recapitulando

- Anteriormente, apresentei 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 concluimos 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 da aula de hoje, começamos a discutir como encontrar os pesos.

Classificação linear

- Como vimos, o objetivo da **classificação** é usar as características (i.e., vetores de atributos, \mathbf{x}) de, por exemplo, um objeto para identificar a qual classe ele pertence.
- Um **classificador linear** atinge esse objetivo **tomando uma decisão de classificação** com base no valor de uma **combinação linear** dos **atributos**, ou seja, na saída de uma **função discriminante linear**.

- A saída de um **classificador linear** é dada por

$$y = h_a(\mathbf{x}) = f(g(\mathbf{x})) = f(a_0 + \dots + a_K x_K) = f\left(\sum_{k=0}^K a_k x_k\right) = f(\mathbf{a}^T \mathbf{x}),$$

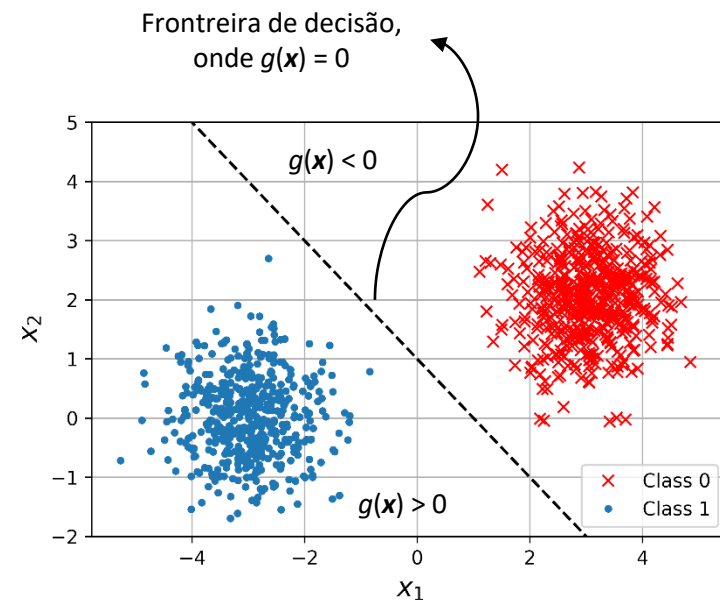
onde $h_a(\mathbf{x})$ é conhecida como **função hipótese de classificação**, $\mathbf{x} = [1, x_1, \dots, x_K]^T$ e $f(\cdot)$ é uma **função de limiar de decisão**.

- **Função de limiar de decisão** é uma função que mapeia a saída da **função discriminante linear**, $\mathbf{a}^T \mathbf{x}$ (produto escalar), 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 definir as classes.
- Originalmente, as **funções discriminantes** seguem equações de hiperplanos:
 $\sum_{k=0}^K a_k x_k.$

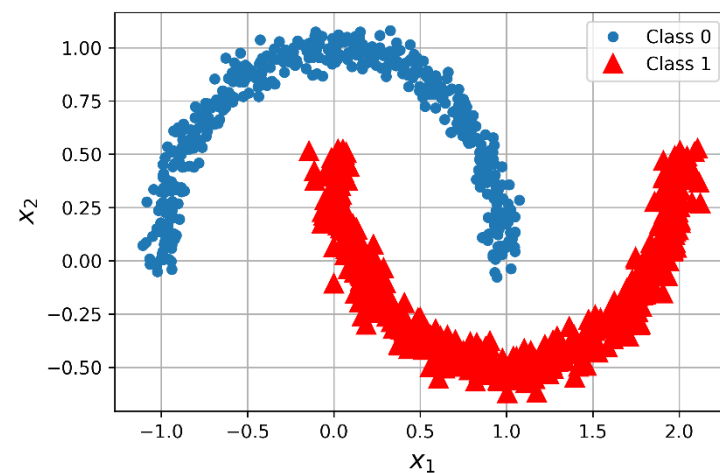
produto
escalar

Classificação linear

- Dado um **conjunto de treinamento**, a tarefa do **classificador** é a de **aprender** uma **função hipótese de classificação**, $h_a(\mathbf{x})$, que recebe um exemplo (e.g., x_1 e x_2) e retorna a classe do exemplo.
- Para que um **classificador linear** funcione corretamente, 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** consista de um **hiperplano**.
- Classes que podem ser separadas por um **hiperplano** são chamadas de **linearmente separáveis**.
- Na primeira figura, a **fronteira de decisão** é definida por uma **função discriminante** que é uma **reta**: $g(\mathbf{x}) = 1 - x_1 - x_2$.
- Na segunda figura, devido à proximidade das classes, não existe um **hiperplano** que as separe.



Classes linearmente separáveis.



Classes não-linearmente separáveis.

Classificação não-linear

- Originalmente, **classificação linear** é usada quando as classes podem ser separadas por **superfícies de decisão lineares**.
- Ou seja, as **funções discriminantes** são **hiperplanos**: $\sum_{k=0}^K a_k x_k$.
- Mas e se não pudermos separar as classes com um **hiperplano**, ou seja, se elas não forem **linearmente separáveis**?
- Nestes casos, usamos **funções discriminantes não-lineares**, como, por exemplo, **polinômios**:
 - $g(x) = (x_1 - a)^2 + (x_2 - b)^2 - r^2$, 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.
- Portanto, quando usamos uma **função discriminante não-linear**, convertemos **classificadores lineares** em **classificadores não-lineares** através de uma **transformação dos atributos**.
 - $g(x) = (x_1 - a)^2 + (x_2 - b)^2 - r^2 = x_1^2 - 2ax_1 + x_2^2 - 2bx_2 + (a^2 + b^2 - r^2)$
- Esta transformação pode ser vista também como uma mudança do **espaço de entrada**, o que normalmente leva ao **aumento das dimensões de entrada** ou **mudança dos eixos**.

Função de limiar de decisão rígido

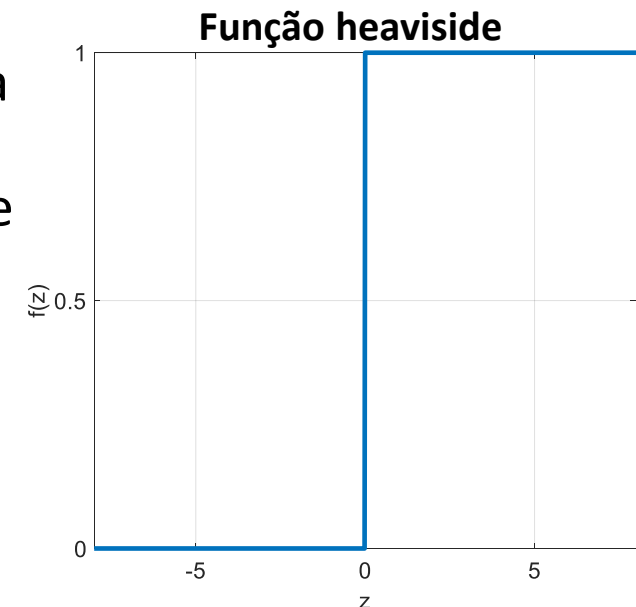
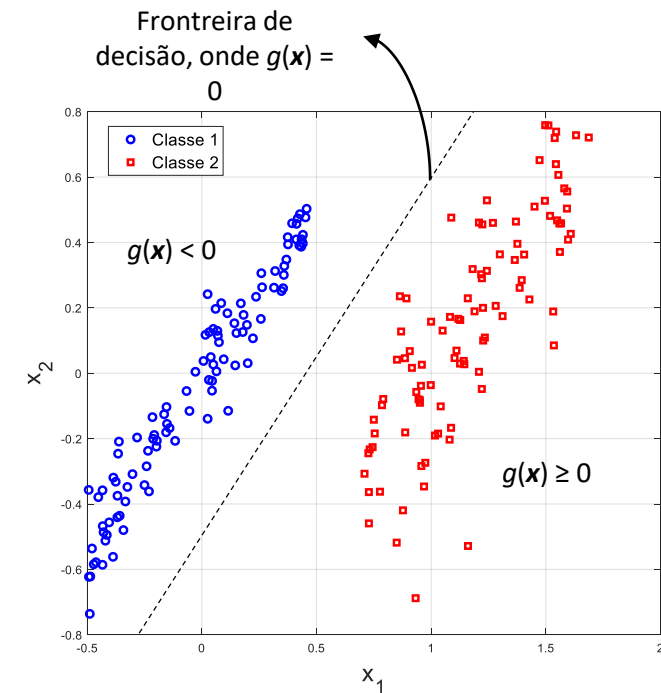
- Para o exemplo ao lado, podemos definir a **função hipótese de classificação** como

$$y = h_a(\mathbf{x}) = \begin{cases} 0, & g(\mathbf{x}) = \mathbf{x}^T \mathbf{a} < 0 \text{ (Classe 1)} \\ 1, & g(\mathbf{x}) = \mathbf{x}^T \mathbf{a} \geq 0 \text{ (Classe 2)} \end{cases} \quad \begin{matrix} \text{if e} \\ \text{else} \end{matrix}$$

- Percebam que a saída da **função hipótese** é **binária**, ou seja, temos apenas 2 possíveis valores, 0 ou 1.
 - Como implementar essas **condições** através de uma função matemática?
- O mapeamento entre o valor da função discriminante, $g(\mathbf{x})$, e a saída 0 ou 1 é feito através da **função de limiar de decisão**, $f(g(\mathbf{x}))$.
- Uma **função de limiar de decisão** que faça o mapeamento do valor de $g(\mathbf{x})$ em apenas 2 valores é chamada de **função de limiar de decisão rígido**.
- A **função de limiar de decisão rígido** é mostrada na figura ao lado e é definida como

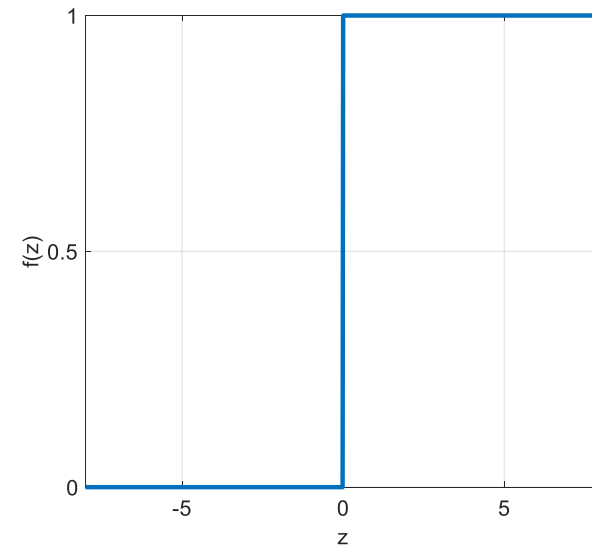
Conhecida também como **função heaviside**. \longrightarrow

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



Classificação com limiar de decisão rígido

- Agora que a **função hipótese**, $h_a(x)$, tem uma forma matemática bem definida, precisamos pensar em como encontrar os pesos, a .
- Nós queremos encontrá-los de tal forma que **o erro de classificação seja minimizado**, ou seja, que os exemplos sejam atribuídos às suas respectivas classes.
- No caso da **regressão linear**, nós fizemos isso de duas maneiras:
 - i. de forma fechada (através da **equação normal**) fazendo a derivada parcial com relação aos pesos igual a zero e resolvendo a equação para os pesos;
 - ii. e através do algoritmo do **gradiente descendente**.
- Entretanto, com a **função de limiar rígido**, nenhuma das duas abordagens é possível devido a **derivada** de $f(g(x))$ ser igual a zero em todos os pontos exceto em $g(x) = x^T a = 0$, onde ela é indeterminada.
- **Portanto, o que podemos fazer?**



Classificação com limiar de decisão rígido

- Uma possível abordagem para o problema da aprendizagem quando utilizamos o **limiar de decisão rígido** 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**.

- A **regra de atualização** dos **pesos** é dada pela seguinte equação

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

Os pesos são atualizados a cada novo exemplo, ou seja, amostra a amostra.

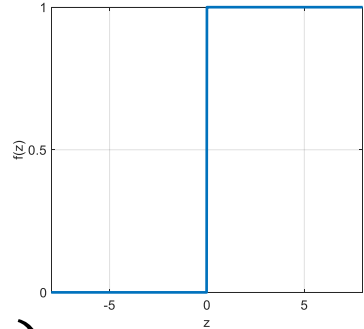
a qual é essencialmente idêntica à regra de atualização para a **regressão linear** quando utilizamos o **gradiente descendente estocástico**, onde α é o passo de aprendizagem.

- Por razões que discutiremos mais adiante, esta regra é chamada de **regra de aprendizagem do perceptron**.
- Essa regra de aprendizagem é **aplicada a um exemplo por vez**, escolhendo exemplos **aleatoriamente**, assim como fizemos com o **gradiente descendente estocástico**.
- Como estamos considerando classificadores com valores de saída iguais a 0 ou 1, o comportamento da regra de atualização será diferente do comportamento para a regressão linear, como veremos a seguir.

Classificação com limiar de decisão rígido

- Observem a equação de atualização dos pesos

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



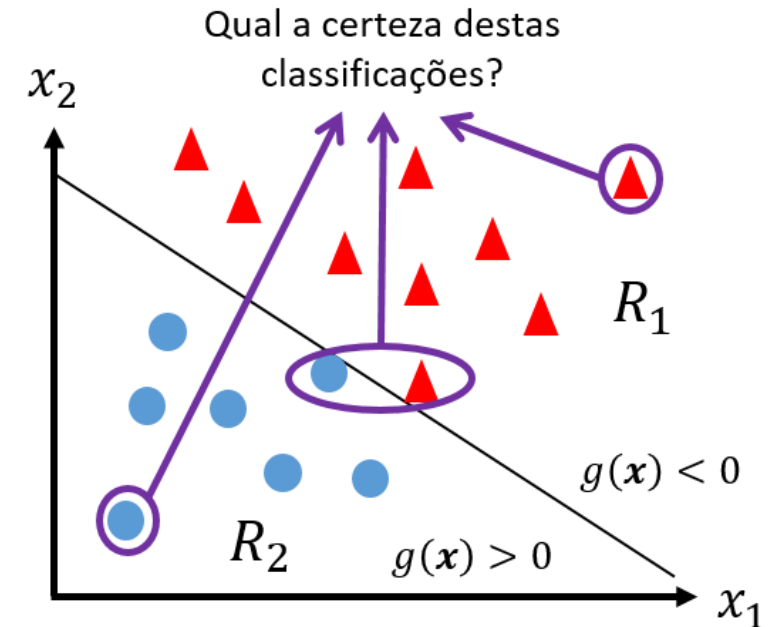
- Ambos, o valor esperado, y , e a saída da **função hipótese**, $h_{\mathbf{a}}(\mathbf{x})$, assumem os valores 0 ou 1, portanto, existem 3 possibilidades:
 - Se a saída estiver correta, i.e., $y = h_{\mathbf{a}}(\mathbf{x})$, então os pesos não são atualizados.
 - Se $y = 1$, mas $h_{\mathbf{a}}(\mathbf{x}) = 0$, então o peso a_k tem seu valor **aumentado** quando o valor de x_k é positivo e **diminuído** quando o valor de x_k é negativo.
 - Isso faz sentido pois nós queremos aumentar o valor do produto escalar $\mathbf{x}^T \mathbf{a}$, ou seja, $g(\mathbf{x})$, de tal forma que $g(\mathbf{x}) > 0$ e, consequentemente, $h_{\mathbf{a}}(\mathbf{x})$ tenha como saída o valor 1.
 - Se $y = 0$, mas $h_{\mathbf{a}}(\mathbf{x}) = 1$, então o peso a_k tem seu valor **diminuído** quando o valor de x_k é positivo e **aumentado** quando o valor de x_k é negativo.
 - Isso faz sentido pois nós queremos diminuir o valor do produto escalar $\mathbf{x}^T \mathbf{a}$, ou seja, $g(\mathbf{x})$, de tal forma que $g(\mathbf{x}) < 0$ e, consequentemente, $h_{\mathbf{a}}(\mathbf{x})$ tenha como saída o valor 0.

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.
 - Existe uma **função discriminante adequada para o problema**, mesmo que não seja um **hiperplano**.
- **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.
- 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 fixos do **passo de aprendizagem**, α , assim como acontece com o GDE.
- Porém, se α decresce 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.
- Podemos também usar o **early-stop** e utilizar 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** do classificador quanto a um resultado.
- Na figura ao lado, dois exemplos estão bem próximos da **fronteira de decisão** enquanto outros dois estão bem distantes dela.
- O classificador com **limiar rígido**, faria uma previsão **completamente confiante** pelo valor 1 para os dois pontos azuis e 0 para os dois triângulos vermelhos, mesmo eles tendo valores bem diferentes de $g(x)$.
- Em muitas situações, nós precisamos de previsões mais graduadas, que indiquem incertezas quanto à classificação.



- Os pontos distantes da **fronteira de decisão** têm valores **absolutos** de $g(x)$ bem maiores do que os dos pontos próximos, os quais têm valores de $g(x)$ muito próximos de 0.
- Ou seja, a confiança deveria ser maior para pontos distantes da fronteira.
- Porém, isso não é refletido na saída do classificador com limiar rígido.

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.](#)
 - **Laboratórios podem ser resolvidos em grupo, mas as entregas devem ser individuais.**

Obrigado!

