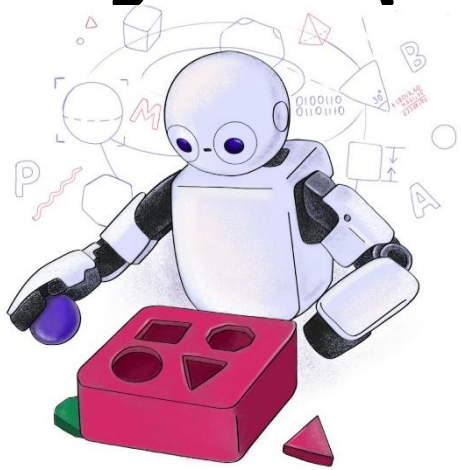
T320 - Introdução ao Aprendizado de Máquina II:

Classificação (Parte IV)



Inatel

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

Recapitulando

- Anteriormente, aprendemos uma nova ***função de limiar***, chamada de ***função logística***, com a qual foi possível se encontrar uma solução com o algoritmo do ***gradiente descendente***.
- Classificadores que utilizam a ***função logística*** como ***função de limiar*** são conhecidos como ***regressores logísticos*** e são utilizados em problemas de ***classificação binária***, ou seja, problemas com 2 classes apenas, após a discretização do valor de saída.
- Na sequência, veremos como lidar com problemas de classificação que envolvem mais de 2 classes, também chamados de ***classificação multi-classes***.

Casos multi-classe

- Até agora, nós vimos como classificar utilizando ***regressão logística*** quando os dados pertencem a apenas 2 classes (i.e., $Q = 2$), mas e quando o problema possui mais de 2 classes (i.e., $Q > 2$)? Por exemplo
  - Reconhecimento de dígitos escritos à mão: 10 dígitos.
  - Classificação de texto: Esportes, Economia, Política, Entretenimento, etc.
  - Classificação de sentimentos: Neutro, Positivo, Negativo.
- Existem algumas abordagens para a ***classificação multi-classe***:
  - Um-Contra-o-Resto
  - Um-Contra-Um
  - Regressão Softmax
- As duas primeiras podem ser aplicadas a qualquer tipo de ***classificador binário*** e não apenas ao ***regressor logístico***.
- A terceira abordagem é uma generalização do ***classificador logístico*** para problemas multi-classe.

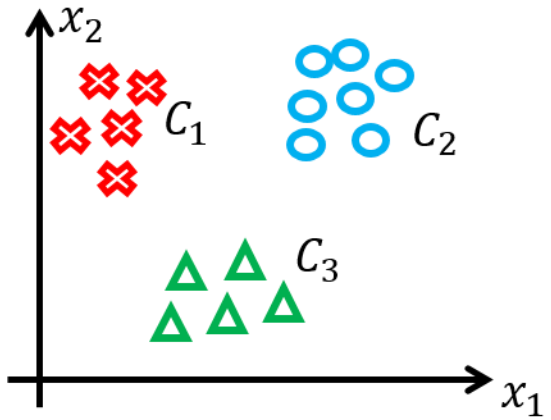
Um-Contra-o-Resto

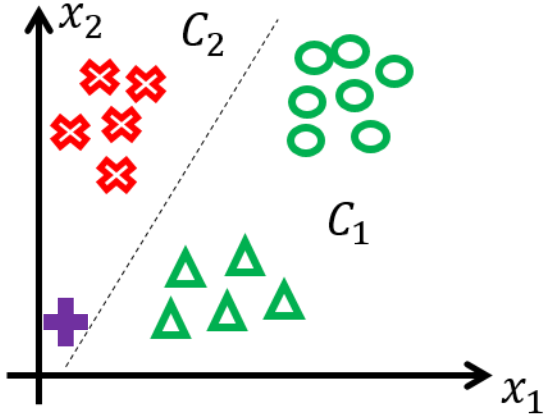
- Nesta abordagem, nós treinamos um ***classificador binário*** (e.g., ***regressor logístico***), representado pela função hipótese, $h_{\boldsymbol{a}}^{q}\big(\boldsymbol{x}(i)\big)$, para cada classe q para predizer a probabilidade de $\hat{y} = q$, ou seja, $P(\hat{y} = q|\boldsymbol{x}(i); \boldsymbol{a})$.
- Em outras palavras, cria-se Q ***classificadores binários***, onde para cada classificador, a classe positiva $C_2 = q$ e a classe negativa C_1 é a junção de todas as outras $Q - 1$ classes.
- Portanto, o ***classificador*** deve indicar a classe positiva caso o exemplo pertença à classe q, e a classe negativa caso o exemplo pertença a qualquer outra classe.
- Para cada novo exemplo de entrada, $\boldsymbol{x}(i)$, realiza-se as predições e escolhe-se a classe que maximize

$$C_q = \arg\max_{q} h_{\boldsymbol{a}}^{q}\big(\boldsymbol{x}(i)\big).$$

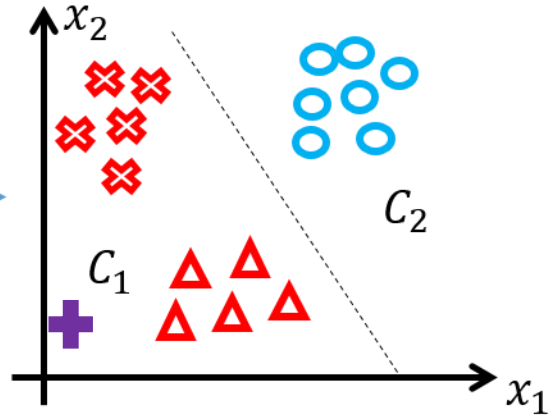
- A vantagem desta abordagem é que se treina apenas Q ***classificadores***.
- A desvantagem é que cada ***classificador binário*** precisa ser treinado com um conjunto negativo que é Q-1 vezes maior, o que pode aumentar o tempo de treinamento.

Um-Contra-o-Resto

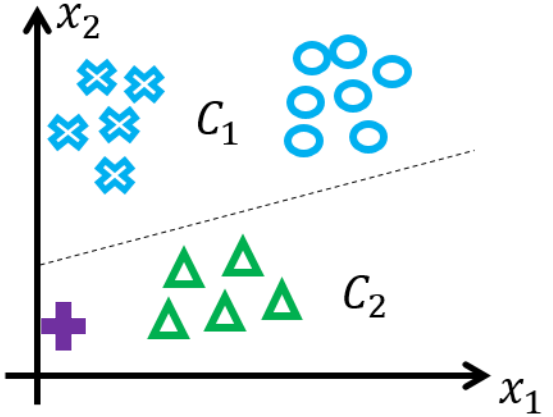




$h_a^1(\boldsymbol{x}(i))$



$h_a^2(\boldsymbol{x}(i))$



$h_a^3(\boldsymbol{x}(i))$

A qual classe pertence ✚ ?

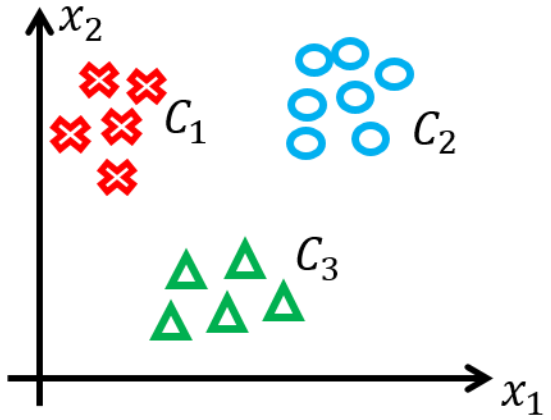
Fronteira de decisão

Exemplo de validação

Um-Contra-Um

- Nesta abordagem, treina-se $Q(Q-1)/2$ ***classificadores binários***.
- Cada ***classificador*** é construído para fazer a distinção entre exemplos pertencentes a cada um dos possíveis ***pares*** de classes.
  - Se $Q = 4$, então treina-se 6 ***classificadores*** para classificar entre C_1/C_2, C_1/C_3, C_1/C_4, C_2/C_3, C_2/C_4, e C_3/C_4.
- No final, cada exemplo é classificado conforme o ***voto majoritário*** entre os ***classificadores***.
- A principal vantagem da abordagem ***Um-Contra-Um*** é que cada ***classificador*** precisa ser treinado apenas com as duas classes que ele deve distinguir.
- A desvantagem é que, por exemplo, se $Q = 10$, temos que treinar 45 ***classificadores***.

Um-Contra-Um



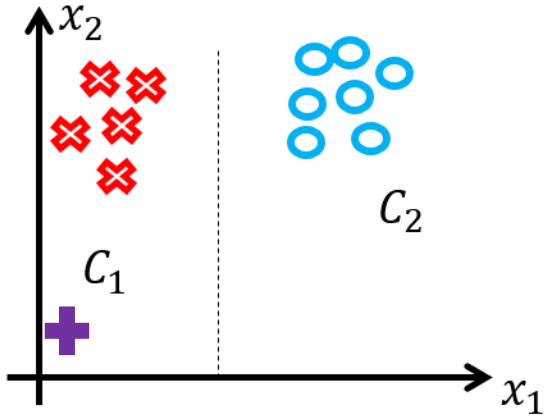
$$Q = 3$$

$$\frac{Q(Q-1)}{2} = 3$$

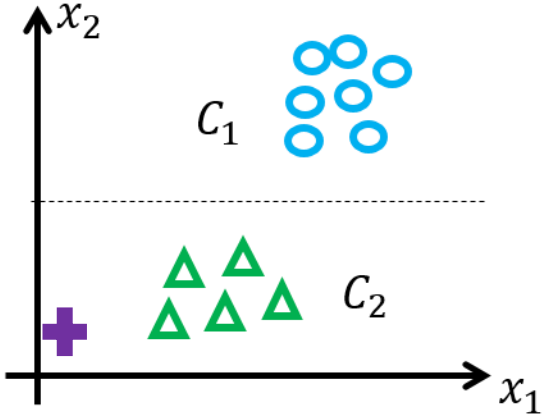
C_2/C_1

C_3/C_2

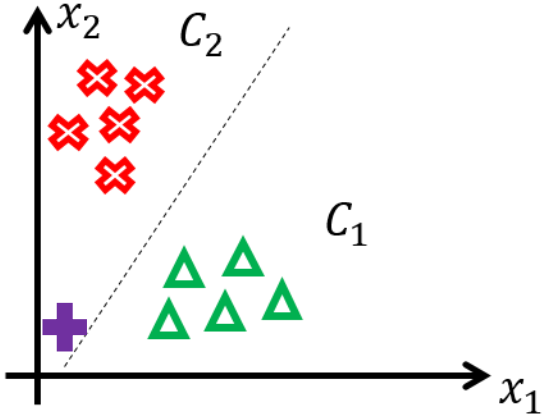
C_1/C_3



$h_a^1(\boldsymbol{x}(i))$



$h_a^2(\boldsymbol{x}(i))$



$h_a^3(\boldsymbol{x}(i))$

A qual classe pertence ✚ ?

Fronteira de decisão

✚ Exemplo de validação

[Exemplo: ClassificationOfFourClassesWithOvAandOvO.ipynb]

Regressão Softmax

- Também conhecida como ***regressão logística multinomial***.
  - Pois as saídas do regressor podem ser interpretadas como as probabilidades de uma variável categoricamente distribuída (as classes) dado um conjunto de variáveis (atributos e pesos).
- É uma generalização do regressor logístico para problemas com ***múltiplas classes***.
- A ideia é treinar um ***único*** classificador com $Q > 2$ saídas, onde cada saída representa a ***probabilidade*** de um exemplo pertencer a uma das Q classes.
  - Por exemplo, para um problema com 4 classes, teríamos um único classificador, mas com 4 saídas.
- Prediz ***apenas uma classe por classificação***, portanto, ele deve ser usado apenas com ***classes mutuamente exclusivas*** como por exemplo diferentes tipos de plantas, dígitos, etc.
  - ***Classes mutuamente exclusivas***: exemplos pertencem a apenas uma das Q classes.
  - Já notícias e animais, por exemplo, podem pertencer a várias a várias classes.
- Para termos um ***único*** classificador, o ***regressor softmax*** possui uma ***função hipótese de classificação***, $h_{\boldsymbol{a}}^{q}\left(\boldsymbol{x}(i)\right)$, e uma ***função discriminante***, $g_q(\boldsymbol{x}(i))$, para cada classe q.

Regressão Softmax

- A ***função hipótese de classificação*** associada à q-ésima classe, $h_{\boldsymbol{a}}^{q}\big(\boldsymbol{x}(i)\big)$, é obtida passando-se a ***função discriminante*** da q-ésima classe, $g_q(\boldsymbol{x}(i))$, através da ***função softmax***,

Cada função discriminante tem seu próprio vetor de pesos.

$$P\big(C_q \mid \boldsymbol{x}(i), \boldsymbol{a}_q\big) = h_{\boldsymbol{a}}^{q}\big(\boldsymbol{x}(i)\big) = \frac{e^{g_q(\boldsymbol{x}(i))}}{\sum_{j=1}^{Q} e^{g_j(\boldsymbol{x}(i))}} = \frac{e^{\boldsymbol{x}(i)^T \boldsymbol{a}_q}}{\sum_{j=1}^{Q} e^{\boldsymbol{x}(i)^T \boldsymbol{a}_j}} \in \mathbb{R}\ [0,1],$$

O somatório de termos exponenciais normaliza o valor da q-ésima saída de tal forma que o somatório das Q saídas seja igual a 1.

onde $\boldsymbol{a}_q = \begin{bmatrix} a_0^q & a_1^q & \cdots & a_K^q \end{bmatrix}^{\boldsymbol{T}} \in \mathbb{R}^{K+1\times 1}$ é o ***vetor de pesos*** da q-ésima ***função discriminante***, $g_q(\boldsymbol{x}(i))$, e i indica o número da amostra.

- Assim como com o regressor logístico, podemos usar equações de ***hiperplanos*** ou ***polinomiais*** como ***funções discriminantes***.
- Assim, a ***função softmax*** estende a ideia do ***regressor logístico*** ao mundo multi-classes.
- Ou seja, a ***função softmax*** atribui uma ***probabilidade condicional***, $P\big(C_q \mid \boldsymbol{x}(i), \boldsymbol{a}_q\big)$, a cada classe, q, em um problema com múltiplas classes ($Q > 2$), onde a soma destas Q probabilidades deve ser igual a 1

$$P(C_1 \mid \boldsymbol{x}(i), \boldsymbol{a}_1) + P(C_2 \mid \boldsymbol{x}(i), \boldsymbol{a}_2) + \cdots + P\big(C_Q \mid \boldsymbol{x}(i), \boldsymbol{a}_Q\big) = 1.$$

Regressão Softmax

- Portanto, o objetivo é encontrar um ***modelo*** (i.e., os ***pesos*** das Q funções hipótese) que atribua uma alta probabilidade para a classe alvo e consequentemente uma baixa probabilidade para as demais classes.
- Assim como fizemos anteriormente, precisamos definir uma ***função de erro*** e ***minimizá-la*** para encontrarmos os ***pesos*** das Q ***funções hipótese*** do classificador softmax.
- A ***função de erro médio*** para a ***regressão softmax*** é dada por

O erro tende a 0 quando $h_{\boldsymbol{a}}^{q}(\boldsymbol{x})$ tende a 1, caso contrário, o erro aumenta.

$$J_e(\boldsymbol{A}) = -\frac{1}{N}\sum_{i=0}^{N-1}\sum_{q=1}^{Q} 1\{y(i)+1 == q\}\log\left(h_{\boldsymbol{a}}^{q}\left(\boldsymbol{x}(i)\right)\right),$$

onde $1\{\cdot\}$ é a ***função indicadora***, de modo que 1{uma condição verdadeira} = 1 e 1{uma condição falsa} = 0, $\boldsymbol{A} \in \mathbb{R}^{K+1\times Q}$ é a matriz com os ***pesos*** para todas as ***funções hipótese*** das Q classes e $y(i)$ é o i-ésimo valor esperado.

- A matriz $\boldsymbol{A}$ contém em suas colunas os vetores de pesos, $\boldsymbol{a}_q$, de cada umas das Q ***funções discriminantes***.

Regressão Softmax

- Usando-se a codificação ***one-hot***, a equação anterior pode ser re-escrita como

$$J_e(\boldsymbol{A}) = -\frac{1}{N}\sum_{i=0}^{N-1} \boldsymbol{y}(i)^T \log\left(\boldsymbol{h_a}\left(\boldsymbol{x}(i)\right)\right),$$

onde $\boldsymbol{y}(i) = [1\{y(i)+1 == 1\}, \quad \cdots, \quad 1\{y(i)+1 == Q\}]^T \in \mathbb{R}^{Q\times 1}$ é o vetor utilizando a codificação ***one-hot*** e $\boldsymbol{h_a}\left(\boldsymbol{x}(i)\right) \in \mathbb{R}^{Q\times 1}$ é o vetor com as saídas das Q ***funções hipóteses***

$$\begin{aligned}\boldsymbol{h_a}\left(\boldsymbol{x}(i)\right) &= \left[h_{\boldsymbol{a}}^1\left(\boldsymbol{x}(i)\right), \quad \cdots, \quad h_{\boldsymbol{a}}^Q\left(\boldsymbol{x}(i)\right)\right]^T \\ &= \left[P(C_1 \mid \boldsymbol{x}(i), \boldsymbol{a}_1), \quad \cdots, \quad P\left(C_Q \mid \boldsymbol{x}(i), \boldsymbol{a}_Q\right)\right]^T.\end{aligned}$$

- Notem que quando existem apenas duas classes ($Q = 2$), a ***função de erro*** acima é equivalente à ***função de erro*** do ***regressor logístico***.
- Ou seja, mesmo tendo sido pensado para caso onde $Q > 2$, o regressor softmax pode ser usado quando $Q = 2$.
- A ***função de erro médio não é linear*** e, portanto, ***não existe uma forma fechada*** para encontramos os pesos. Porém, ela é ***convexa*** e, portanto, é garantido que o algoritmo do ***gradiente descendente*** encontre o mínimo global.

Regressão Softmax

- Sendo assim, usamos o algoritmo do ***gradiente descendente*** para encontrar os ***pesos*** das Q funções discriminantes que ***minimizam*** a ***função de erro médio***.
- A atualização iterativa dos ***pesos*** da q-ésima classe, C_q, é dada por

$$\boldsymbol{a}_q = \boldsymbol{a}_q - \alpha \frac{\partial J_e(\boldsymbol{A})}{\partial \boldsymbol{a}_q}, \forall q$$

- Considerando uma ***função discriminante linear*** (i.e., ***hiperplano***), a derivada da ***função de erro médio***, $J_e(\boldsymbol{A})$, com respeito a cada vetor de pesos, $\boldsymbol{a}_q$, tem uma expressão idêntica àquela obtida para a ***regressão logística***:

Para outros formatos de função discriminante, basta alteramos o formato da matriz $\boldsymbol{X}$.

Forma matricial

$$\frac{\partial J_e(\boldsymbol{A})}{\partial \boldsymbol{a}_q} = -\frac{1}{N}\sum_{i=0}^{N-1}\left[1\{y(i)+1 == q\} - h_{\boldsymbol{a}}^{q}\left(\boldsymbol{x}(i)\right)\right]\boldsymbol{x}(i)^T = -\frac{1}{N}\boldsymbol{X}^T\left(\boldsymbol{y}_q - \widehat{\boldsymbol{y}}_q\right).$$

Regressão Softmax

Observações

- O ***vetor gradiente*** da ***função de erro*** depende da ***função discriminante*** adotada.
- Entretanto, como vimos antes, esta dependência afeta apenas a ***matriz de atributos***, $\boldsymbol{X}$.
- O regressor softmax apresenta duas propriedades:
  - $0 \leq h_{\boldsymbol{a}}^{q}\big(\boldsymbol{x}(i)\big) \leq 1$, ou seja, a saída da q-ésima função hipótese de classificação sempre será um valor dentro do intervalo [0, 1];
  - $\sum_{q=1}^{Q} h_{\boldsymbol{a}}^{q}\big(\boldsymbol{x}(i)\big) = \sum_{q=1}^{Q} P\big(C_q \,\big|\boldsymbol{x}(i), \boldsymbol{a}_q\big) = 1$, ou seja, o somatório das ***probabilidades condicionais*** de todas as Q classes é igual a 1.
- Estas duas propriedades fazem com que o vetor

$$\boldsymbol{h_a}\big(\boldsymbol{x}(i)\big) = \Big[h_{\boldsymbol{a}}^{1}\big(\boldsymbol{x}(i)\big), \quad \cdots, \quad h_{\boldsymbol{a}}^{Q}\big(\boldsymbol{x}(i)\big)\Big]^{T} \in \mathbb{R}^{Q\times 1},$$

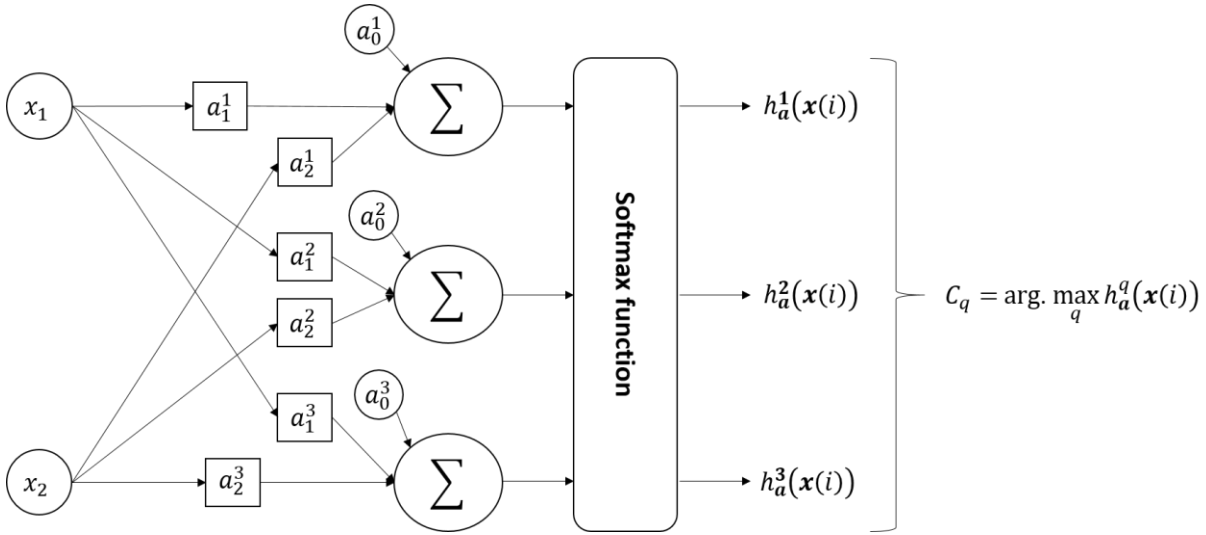
contendo todas as saídas do regressor softmax atenda os requisitos de uma ***função massa de probabilidade multinomial.***

Regressão Softmax

- Após o treinamento, o classificador atribui ao exemplo de entrada, $\boldsymbol{x}(i)$, a classe, q, com a ***maior probabilidade estimada***, que é simplesmente a classe com maior valor para $g_q\big(\boldsymbol{x}(i)\big) = \boldsymbol{x}(i)^T \boldsymbol{a}_q$

$$C_q = \arg.\max_q h_{\boldsymbol{a}}^q\big(\boldsymbol{x}(i)\big) = \arg.\max_q P\big(C_q \,\big|\, \boldsymbol{x}(i), \boldsymbol{a}_q\big) = \arg.\max_q \boldsymbol{x}(i)^T \boldsymbol{a}_q\,.$$

- A arquitetura de um ***regressor softmax*** para três classes (i.e., $Q = 3$) e dois atributos (x_1 e x_2) é mostrada abaixo.



A ideia por trás da ***regressão softmax*** é bastante simples: dado um exemplo de entrada $\boldsymbol{x}$, o regressor softmax primeiro calcula uma "***pontuação***", $g_q(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{a}_q$, para cada classe q, em seguida, estima a probabilidade de cada classe aplicando a função softmax às "***pontuações***".

Exemplo: softmax_regressor_with_scikit_learn.ipynb

Tarefas

- **Quiz**: “*T320 - Quiz - Classificação (Parte IV)*” que se encontra no MS Teams.
- **Exercício Prático**: **Laboratório #4**.
  - 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!