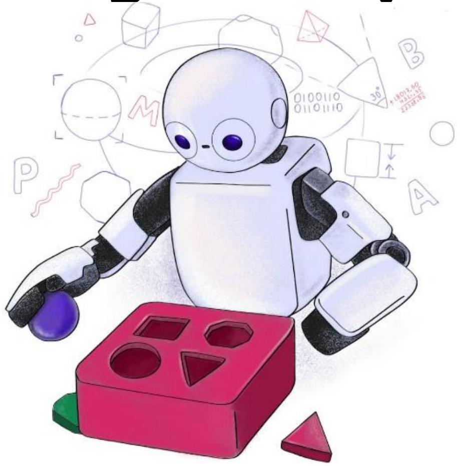
T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte IV)*



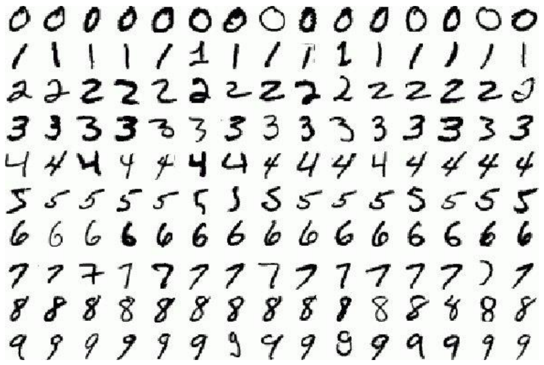
Inatel

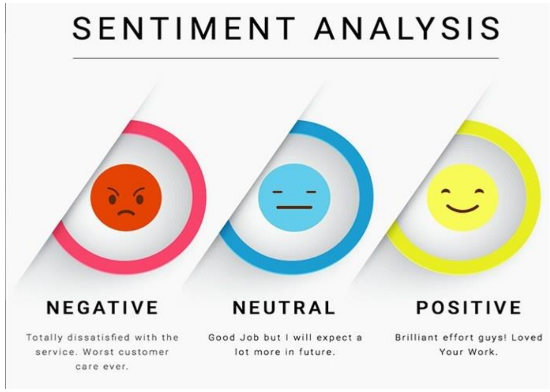
Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

Recapitulando

- Anteriormente, aprendemos uma nova ***função de limiar***, chamada de ***função logística***, com a qual foi possível encontrar uma solução para o problema de classificação usando 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 apenas 2 classes, 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***.

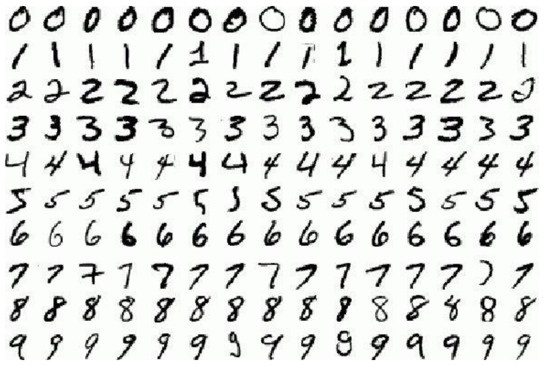
Casos multi-classe

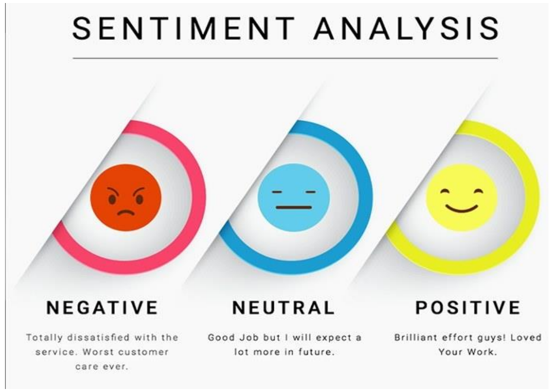




- Até agora, nós vimos como classificar utilizando a ***regressão logística***.
- Nesse caso, os dados pertencem a apenas 2 classes (i.e., $Q = 2$).
- Porém, e quando o problema possuir mais de 2 classes (i.e., $Q > 2$)?
- Por exemplo
  - Reconhecimento de dígitos escritos à mão: 10 dígitos.
  - Classificação de texto: Esportes, Economia, Política, Entretenimento, etc.
  - Classificação de sentimentos: Neutro, Positivo, Negativo.

Casos multi-classe





- Quando $Q > 2$, chamamos o problema de classificação ***multi-classes***.
- Existem algumas abordagens para a ***classificação multi-classe***:
  - Um-Contra-o-Resto
  - Um-Contra-Um
  - Regressão Softmax
- As duas primeiras são usadas com ***classificadores binários***, como o ***regressor logístico***.
- A terceira abordagem é uma generalização do ***classificador logístico*** para problemas multi-classe.

Um-Contra-o-Resto

- Nesta abordagem, treina-se ***um classificador binário*** para cada classe q, para predizer a probabilidade $P\left(C_q \middle| \boldsymbol{x}(i); \boldsymbol{a}\right), \forall q \in \{1, \dots, Q\}$, onde C_q é a classe positiva e as demais formam a classe negativa.
- Em outras palavras, treina-se Q ***classificadores binários***, onde para cada classificador, a classe positiva é a q-ésima classe e a classe negativa é a junção de todas as outras, $Q-1$, classes.
- Fazendo isso, nós ***transformamos um problema com Q classes em Q problemas binários***.
- Nesta abordagem, cada um dos Q classificadores binários é representado pela função hipótese $h_{\boldsymbol{a}}^{q}\big(\boldsymbol{x}(i)\big)$.

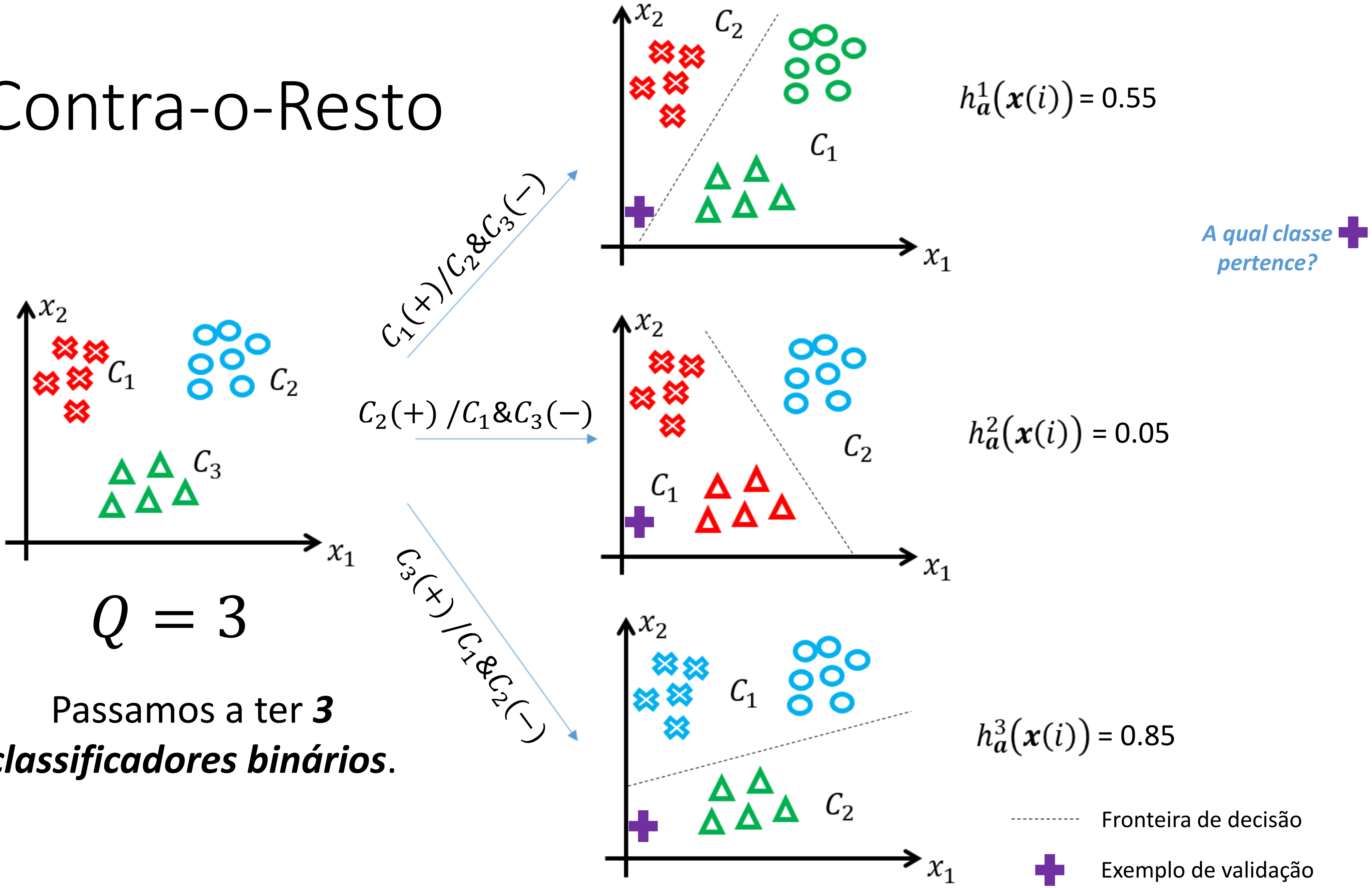
Um-Contra-o-Resto

- Portanto, o q-ésimo *classificador* deve indicar a classe positiva caso o exemplo pertença à q-ésima classe, ou à classe negativa caso o exemplo pertença a qualquer uma das outras $Q-1$ classes.
- Após o treinamento, ***para cada exemplo*** de entrada, $\boldsymbol{x}(i)$, ***realiza-se Q predições*** e escolhe-se a classe que maximiza $h_{\boldsymbol{a}}^{q}(\boldsymbol{x}(i))$

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

- A ***vantagem*** desta abordagem é que treina-se apenas Q ***classificadores***.
- Uma ***desvantagem*** é que cada ***classificador binário*** precisa ser treinado com um conjunto negativo que é Q-1 vezes maior, o que pode ***tornar o treinamento lento*** e ***aumentar a possibilidade de classes desbalanceadas***.

Um-Contra-o-Resto



$$Q = 3$$

Passamos a ter ***3 classificadores binários***.

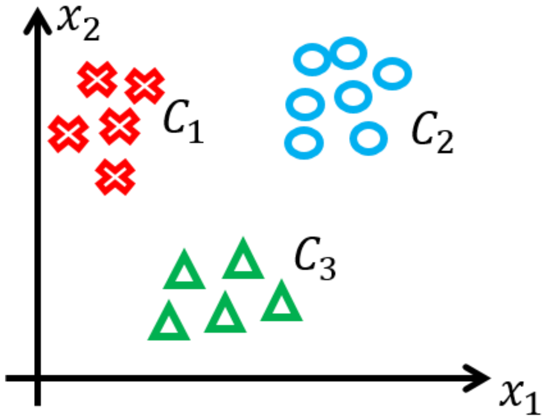
Um-Contra-Um

- Nesta abordagem, treina-se $Q(Q-1)/2$ ***classificadores binários***.
- Cada ***classificador*** é treinado para classificar os exemplos pertencentes a ***cada um dos possíveis pares de classes***.
  - Por exemplo, 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.
- ***Transformamos um problema com Q classes em $Q(Q-1)/2$ problemas binários.***
- No final, cada exemplo é classificado conforme o ***voto majoritário*** entre os ***classificadores***.
- Ou seja, a classe que receber mais votos é a classe atribuída ao exemplo.

Um-Contra-Um

- A principal ***vantagem*** dessa abordagem é que cada classificador precisa ser ***treinado apenas com as duas classes*** que ele deve distinguir, portanto, a ***chance de desbalanceamento é menor***.
- Além disso, o ***tempo de treinamento de cada classificador também é menor***, pois treina-se cada um deles com pares de classes.
- A ***desvantagem*** é que, por exemplo, se $Q = 10$, temos que treinar 45 classificadores.
- Consequentemente, o ***tempo total de treinamento pode ser alto***.

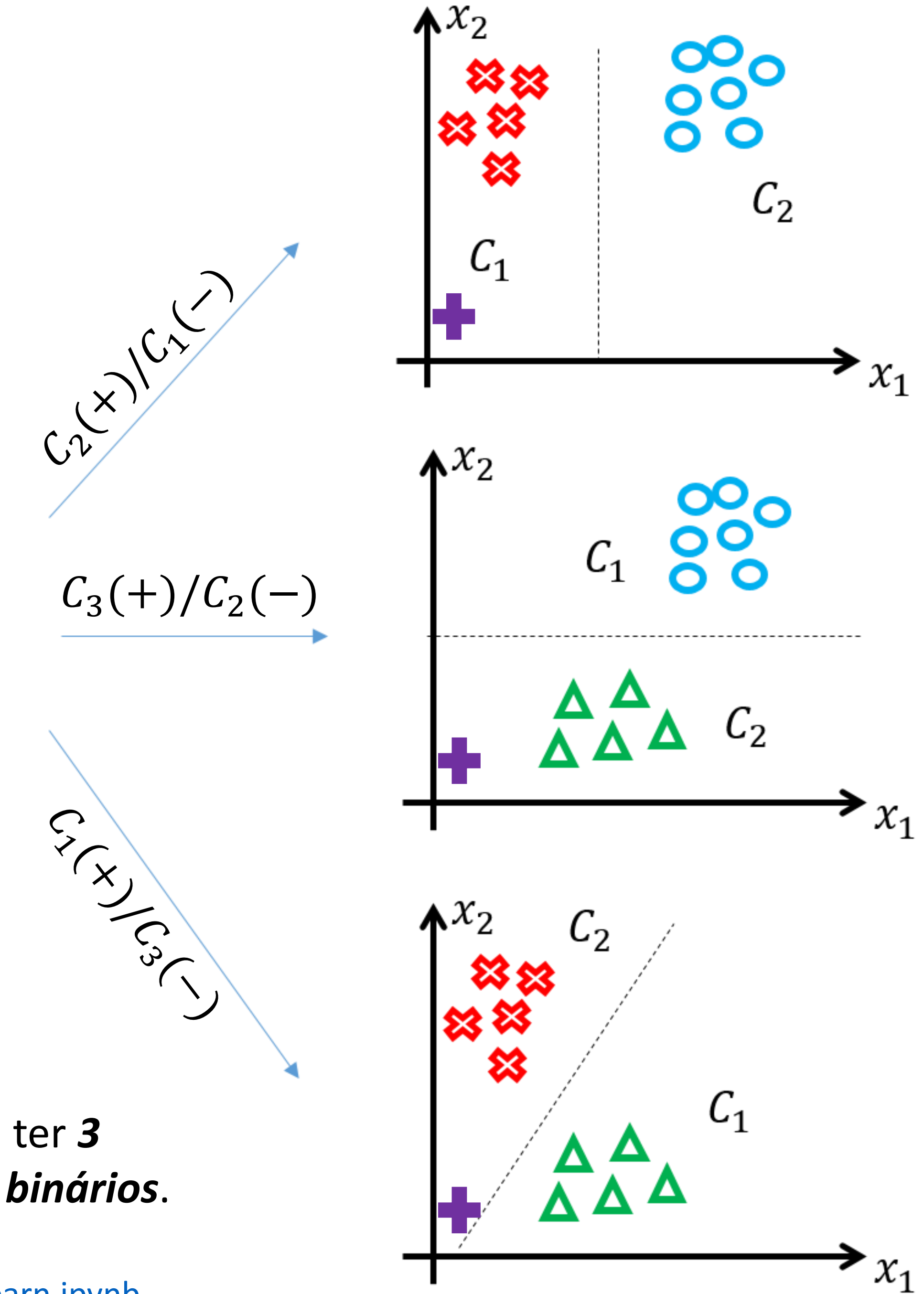
Um-Contra-Um



$$Q = 3$$

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

Passamos a ter ***3 classificadores binários***.



$h_{\boldsymbol{a}}^{1}(\boldsymbol{x}(i))$ = 0.05 (voto em C_1)

A qual classe ✚ *pertence?*

$h_{\boldsymbol{a}}^{2}(\boldsymbol{x}(i))$ = 0.85 (voto em C_3)

$h_{\boldsymbol{a}}^{3}(\boldsymbol{x}(i))$ = 0.55 (voto em C_1)

Fronteira de decisão

✚ Exemplo de validação

[Exemplo: ClassificationOfFourClassesWithOvAandOvO-SciKitLearn.ipynb]

Regressão softmax

- Também conhecida como ***regressão logística multinomial***.
  - Pois as saídas deste regressor podem ser interpretadas como as ***probabilidades de uma variável categoricamente distribuída*** (as classes) ***dado um conjunto de variáveis*** (atributos e pesos).
  - Em outras palavras, a regressão ***softmax estima a probabilidade de um exemplo de entrada pertencer a cada uma das Q possíveis classes***.
- É uma generalização do regressor logístico para problemas com ***múltiplas classes***, em geral, $Q > 2$.
- A ideia é treinar um ***único*** classificador com Q saídas, onde cada saída representa a ***probabilidade*** de um exemplo pertencer a uma das Q classes.
  - Por exemplo, para um problema com 4 classes, teremos um único classificador, mas com 4 saídas.

Regressão softmax

- Prediz ***apenas uma classe por classificação***, portanto, ele deve ser usado apenas com ***classes mutuamente exclusivas*** como por exemplo diferentes tipos de plantas, dígitos, carros, etc.
  - ***Classes mutuamente exclusivas***: exemplos pertencem a apenas uma das Q classes.
  - Já notícias, músicas e imagens, por exemplo, podem pertencer a várias ou conter várias classes ao mesmo tempo.
- Para termos um ***único*** classificador, o ***regressor softmax*** ***possui uma função hipótese de classificação***, $h_{\boldsymbol{a}}^{q}\left(\boldsymbol{x}(i)\right)$, e, consequentemente, ***uma função discriminante***, $g_q(\boldsymbol{x}(i))$, ***para cada classe*** q.

Regressão softmax

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

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

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 (coluna) de pesos*** da q-ésima ***função discriminante***, $g_q(\boldsymbol{x}(i))$, e i indica o número da amostra.

- O somatório de termos exponenciais no denominador ***normaliza*** o valor da q-ésima saída de tal forma que o somatório das Q saídas seja igual a 1.
- Cada função discriminante tem seu próprio vetor de pesos, $\boldsymbol{a}_q$.

Regressão softmax

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

$$P(C_0 \mid \boldsymbol{x}(i), \boldsymbol{a}_0) + P(C_1 \mid \boldsymbol{x}(i), \boldsymbol{a}_1) + \cdots + P\left(C_{Q-1} \mid \boldsymbol{x}(i), \boldsymbol{a}_{Q-1}\right) = 1.$$

- 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***.

Regressão softmax

- 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.

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

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*** das Q ***funções hipótese*** 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***.
- Essa função também é conhecida como ***função da entropia cruzada***.

Regressão softmax

- Usando-se a codificação ***one-hot***, a equação de erro médio pode ser reescrita como

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

onde $\boldsymbol{y}(i) = [1\{y(i) == 0\}, \quad \cdots, \quad 1\{y(i) == Q-1\}]^T \in \mathbb{R}^{Q\times 1}$ é um vetor coluna utilizando a codificação ***one-hot*** e $\boldsymbol{h_a}\big(\boldsymbol{x}(i)\big) \in \mathbb{R}^{Q\times 1}$ é um vetor coluna com as saídas das Q ***funções hipóteses de classificação***

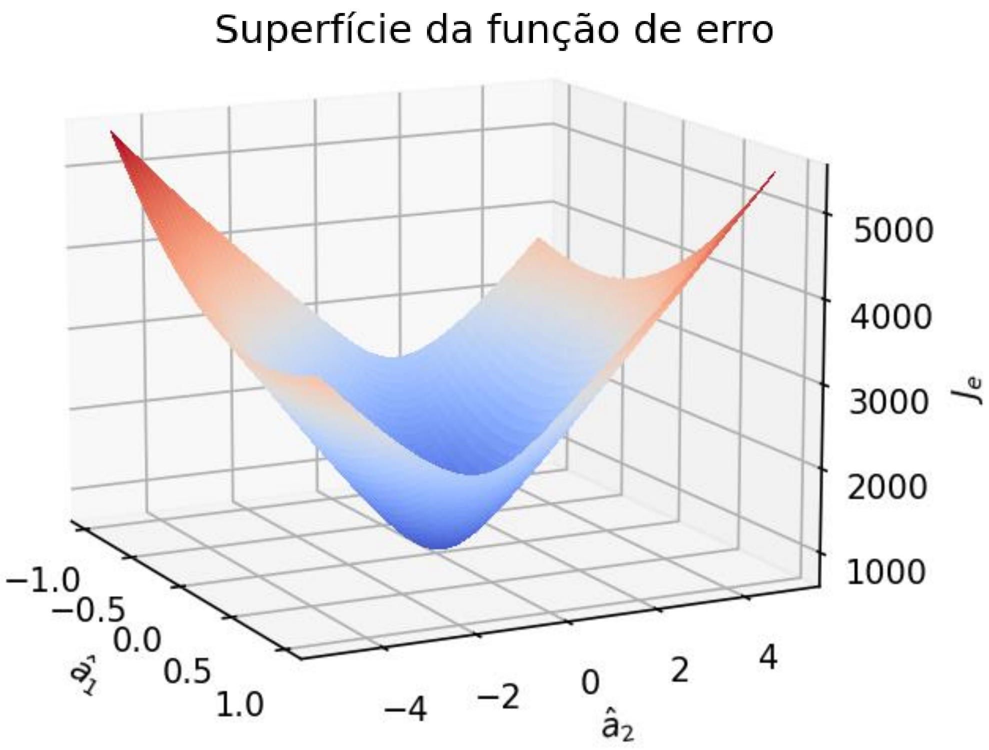
$$\begin{aligned}\boldsymbol{h_a}\big(\boldsymbol{x}(i)\big) &= \left[h_{\boldsymbol{a}}^{0}\big(\boldsymbol{x}(i)\big) \quad \cdots \quad h_{\boldsymbol{a}}^{Q-1}\big(\boldsymbol{x}(i)\big)\right]^T \\ &= \left[P(C_0 \mid \boldsymbol{x}(i), \boldsymbol{a}_0) \quad \cdots \quad P\big(C_{Q-1} \mid \boldsymbol{x}(i), \boldsymbol{a}_{Q-1})\right]^T.\end{aligned}$$

Regressão softmax

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

- Notem que quando existem apenas duas classes ($Q = 2$), a ***função de erro médio*** é equivalente à ***função de erro médio*** do ***regressor logístico***.
- Ou seja, mesmo tendo sido pensado para o caso onde $Q > 2$, o regressor softmax pode ser usado quando $Q = 2$.
- Entretanto, existe um problema com essa função de erro médio.
- A ***função de erro médio*** ***não é linear*** e, portanto, ***não existe uma forma fechada*** para encontramos os pesos.

Regressão softmax



- Porém, ela é ***convexa*** e ***contínua***, portanto, é ***garantido que o algoritmo do gradiente descendente encontre o mínimo global***.
- Sendo assim, usaremos o algoritmo do ***gradiente descendente*** para encontrar os ***pesos*** das Q funções discriminantes que ***minimizam*** a ***função de erro médio***.

Regressão softmax

- A atualização dos ***pesos*** da q-ésima função discriminante, $g_q(\boldsymbol{x}(i))$, é dada por

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

- Considerando o ***hiperplano*** como ***função discriminante***, a derivada da ***função de erro médio***, $J_e(\boldsymbol{A})$, com respeito ao vetor de pesos, $\boldsymbol{a}_q$, é dada por

Forma matricial

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

onde $\boldsymbol{y}_q$ e $\widehat{\boldsymbol{y}}_q$ são vetores coluna ambos com dimensão $N \times 1$ contendo os q-ésimos elementos dos vetores *one-hot* e de saída do regressor softmax para cada um dos N vetores de atributo, respectivamente.

Regressão softmax

$$\frac{\partial J_e(\boldsymbol{A})}{\partial \boldsymbol{a}_q} = -\frac{1}{N}\boldsymbol{X}^T\left(\boldsymbol{y}_q - \widehat{\boldsymbol{y}}_q\right).$$

- Notem que essa expressão do vetor gradiente é ***idêntica*** àquela obtida para o caso da ***regressão logística.***
- O vetor gradiente pode ser reescrito levando em consideração todos os Q vetores de peso:

$$\frac{\partial J_e(\boldsymbol{A})}{\partial \boldsymbol{A}} = -\frac{1}{N}\boldsymbol{X}^T(\boldsymbol{Y} - \boldsymbol{H}_{\boldsymbol{A}}),$$

onde $\boldsymbol{Y}$ e $\boldsymbol{H}_{\boldsymbol{A}}$ são matrizes com dimensão $N \times Q$ contendendo em cada linha os vetores $\boldsymbol{y}(i)^T$e $\boldsymbol{h}_{\boldsymbol{a}}\left(\boldsymbol{x}(i)\right)^T$ para todos os N exemplos, respectivamente.

Regressão softmax

$$\frac{\partial J_e(\boldsymbol{A})}{\partial \boldsymbol{A}} = -\frac{1}{N}\boldsymbol{X}^T(\boldsymbol{Y} - \boldsymbol{H}_{\boldsymbol{A}})$$

- Como aconteceu com a regressão logística, notem que o ***vetor gradiente*** da ***função de erro*** ***depende do formato da função discriminante*** adotada.
- Entretanto, como vimos antes, esta ***dependência afeta apenas a matriz de atributos***, $\boldsymbol{X}$.
- Portanto, ***para formatos*** de ***função discriminante*** diferentes de um hiperplano, basta ***criar a matriz de atributos***, $\boldsymbol{X}$, ***seguindo o formato da função***.

Regressão softmax

- O regressor softmax apresenta duas propriedades:
  - $0 \leq h_{\boldsymbol{a}}^{q}(\boldsymbol{x}(i)) \leq 1$, ou seja, a saída da q-ésima função hipótese sempre será um valor dentro do intervalo [0, 1];
  - $\sum_{q=0}^{Q-1} h_{\boldsymbol{a}}^{q}(\boldsymbol{x}(i)) = \sum_{q=0}^{Q-1} P(C_q \mid \boldsymbol{x}(i); \boldsymbol{a}_q) = 1$, ou seja, o somatório das ***probabilidades condicionais*** de todas as Q classes é igual a 1.
- Estas duas propriedades fazem com que o vetor

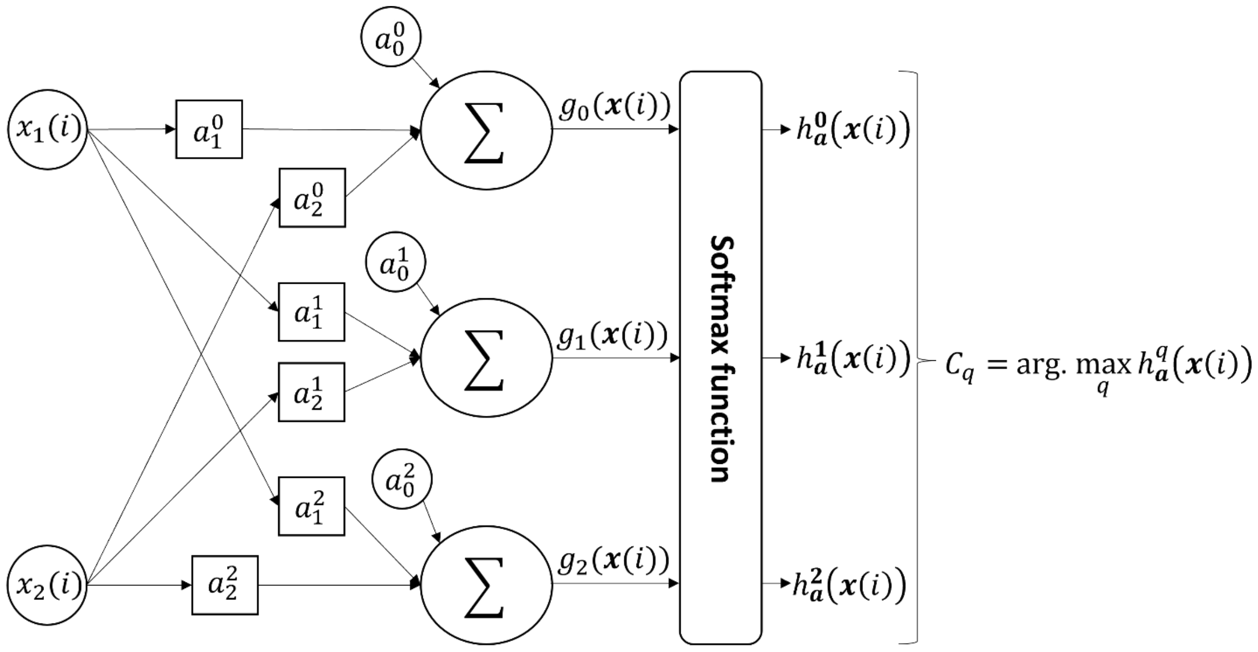
$$\boldsymbol{h}_{\boldsymbol{a}}(\boldsymbol{x}(i)) = \left[h_{\boldsymbol{a}}^{0}(\boldsymbol{x}(i)) \quad \cdots \quad h_{\boldsymbol{a}}^{Q-1}(\boldsymbol{x}(i))\right]^T \in \mathbb{R}^{Q\times 1},$$

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

Regressão softmax

- Após o treinamento, durante a ***fase de predição***, o classificador atribui ao exemplo de entrada, $\boldsymbol{x}(i)$, a classe, q, com a ***maior probabilidade estimada***, ou seja, atribui a classe, q, da função hipótese com maior valor:
$$C_q = \text{arg. } \max_q h_{\boldsymbol{a}}^q\big(\boldsymbol{x}(i)\big) = \text{arg. } \max_q P\big(C_q \mid \boldsymbol{x}(i); \boldsymbol{a}_q\big) .$$
- Uma outra forma de reescrevermos o problema de maximização acima é lembrarmos que a ***classe com maior probabilidade*** é simplesmente a ***classe com maior valor de função discriminante***, $g_q\big(\boldsymbol{x}(i)\big) = \boldsymbol{x}(i)^T \boldsymbol{a}_q$
$$C_q = \text{arg. } \max_q g_q\big(\boldsymbol{x}(i)\big) = \text{arg. } \max_q \boldsymbol{x}(i)^T \boldsymbol{a}_q .$$
- **OBS.**: A ***normalização*** é fundamental para garantir que as saídas do modelo sejam ***interpretáveis como probabilidades*** e para ***garantir um treinamento estável***.

Regressão softmax



- A arquitetura de um ***regressor softmax*** para um problema com três classes (i.e., $Q = 3$) e dois atributos (x_1 e x_2) é mostrada ao lado.
- A ideia por trás da ***regressão softmax*** é bastante simples:
  - dado um exemplo de entrada, $\boldsymbol{x}(i)$,
  - o regressor softmax calcula uma "***pontuação***", para cada classe q, i.e., as saídas das Q funções discriminantes $g_q(\boldsymbol{x}(i)) = \boldsymbol{x}(i)^T \boldsymbol{a}_q, \forall q$.
  - e, 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.

Obrigado!