AULA 12
CLASSIFICAÇÃO MULTICLASSE

1. Objetivos

- Apresentar o problema de classificação de múltiplas classes.
- Apresentar a função de ativação (camada) softmax.
- Apresentar a função de custo e a métrica usadas para problemas de classificação de múltiplas classes.
- Apresentar um problema de identificação de diferentes tipos de vestuários presentes em imagens.

2. Definição do problema

- Em um problema de classificação binária é feita uma escolha entre duas opções, tais como:
  - Uma mensagem de email é ou não spam;
  - Um tumor é ou não maligno;
  - Uma imagem mostra ou não um determinado objeto.
- Em um problema de classificação de múltiplas classes deve-se escolher entre muitas opções, como por exemplo:
  - Um cachorro em uma imagem é um beagle, um pastor alemão, um labrador, ou um doberman?
  - Uma flor em uma imagem é uma rosa, um cravo, uma tulipa, ou uma margarida?
  - Um avião é um Boeing 747, um Airbus 320, um Boeing 777, ou um Embraer 190?
  - Uma fruta em uma imagem é uma maçã, uma laranja, um mamão, ou um limão?
- Alguns problemas reais de classificação de múltiplas classes envolvem escolher uma opção entre milhões de classes. Por exemplo, em um problema de identificar qualquer tipo de objeto em uma imagem qualquer, têm-se milhões de objetos possíveis.

Dados de treinamento

- **Um elemento do conjunto de treinamento** de um problema de classificação de múltiplas classes em geral consiste de:

- Um vetor com os dados de entrada, $\mathbf{x}^{(i)}$, de dimensão (n_x, 1);
- Um vetor com a saída correspondente, $\mathbf{y}^{(i)}$, de dimensão (N_C, 1), onde N_C é o número de classes que se deseja identificar.

➢ Por exemplo, em um problema onde se deseja reconhecer cinco animais diferentes em uma imagem, tais como: cachorro, urso, gato, cabra e cavalo.

- O vetor com os dados de entrada de cada exemplo é composto pela matriz de intensidades luminosas dos pixels da imagem, redimensionada em um vetor linha (padrão do keras), da mesma forma como foi visto no problema de determinar se uma imagem tem ou não um gato (ver Aula 7).
- O vetor de saída de cada exemplo é um vetor linha com cinco elementos, sendo que cada elemento representa um tipo de animal que se deseja indentificar. Assim, exsitem as seguintes possibilidades para o vetor de saída de cada exemplo:

  Se imagem mostra um cahorro $\Rightarrow \mathbf{y}^{(i)} = [1, 0, 0, 0, 0]$;

  Se imagem mostra um urso $\Rightarrow \mathbf{y}^{(i)} = [0, 1, 0, 0, 0]$;

  Se imagem mostra um gato $\Rightarrow \mathbf{y}^{(i)} = [0, 0, 1, 0, 0]$;

  Se imagem mostra uma cabra $\Rightarrow \mathbf{y}^{(i)} = [0, 0, 0, 1, 0]$;

  Se imagem mostra um cavalo $\Rightarrow \mathbf{y}^{(i)} = [0, 0, 0, 0, 1]$.

➢ **Conjunto de exemplos de treinamento:**

- Cada exemplo $\Rightarrow (\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, $\mathbf{x}^{(i)} \in R^{nx}$ e $\mathbf{y}^{(i)} \in R^{NC}$;
- Os m exemplos $\Rightarrow \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\}$;

Codificação das classes (“One-Hot-Encoding”)

➢ Em muitos problemas de classificação de múltiplas classes, as classes são identificadas por números inteiros que variam de 1 até N_C, onde N_C é o número de classes que se deseja identificar.

➢ Nesse caso, no exemplo de identificar os cinco animais (cachorro, urso, gato, cabra e cavalo), a saída de cada exemplo é composta por um único número inteiro. Assim, tem-se, por exemplo:

- Se imagem mostra um cahorro $\Rightarrow y^{(i)} = 0$;
- Se imagem mostra um urso $\Rightarrow y^{(i)} = 1$;
- Se imagem mostra um gato $\Rightarrow y^{(i)} = 2$;
- Se imagem mostra uma cabra $\Rightarrow y^{(i)} = 3$;
- Se imagem mostra um cavalo $\Rightarrow y^{(i)} = 4$.

➢ A melhor forma de tratar esse tipo de dado de saída é transformá-la em um vetor de N_C elementos, sendo todos zeros com exceção do elemento correspondente à classe.

- Para esse exemplo de identificar animais, essa transformação realiza a seguinte operação:
  - Se imagem mostra um cahorro $\Rightarrow y^{(i)} = 0 \rightarrow \mathbf{y}^{(i)} = [1, 0, 0, 0, 0]$;
  - Se imagem mostra um urso: $y^{(i)} = 1 \rightarrow \mathbf{y}^{(i)} = [0, 1, 0, 0, 0]$;
  - Se imagem mostra um gato $\Rightarrow y^{(i)} = 2 \rightarrow \mathbf{y}^{(i)} = [0, 0, 1, 0, 0]$;
  - Se imagem mostra uma cabra $\Rightarrow y^{(i)} = 3 \rightarrow \mathbf{y}^{(i)} = [0, 0, 0, 1, 0]$;
  - Se imagem mostra um cavalo $\Rightarrow y^{(i)} = 4 \rightarrow \mathbf{y}^{(i)} = [0, 0, 0, 0, 1]$.

- **A função que realiza essa transformação é conhecida na literatura por "one-hot-encoding".**

- O Keras possui a função `to_categorical` que realizada essa transformação. Essa função pertence à classe de utilidades do Keras e deve ser importada antes de poder ser utilizada.

- O seguinte código apresenta um exemplo de como realizar essa transformação com o Keras para um vetor numpy, **y**, que representa a classe de 7 exemplos, para um problema de classificação de 5 possíveis classes.

```
# Importa bibliotecas
import numpy as np
from tensorflow.python.keras.utils import to_categorical

# Define vetor com saídas de 7 exemplos
y = np.array([0, 1, 2, 3, 4, 0, 2])

# Transforma saídas inteiras em vetores de 5 elementos cada
y_hot = to_categorical(y)
print(y_hot.shape)
print(y_hot)
```

```
(7, 5)
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0.]]
```

  - Observa-se que a bilbioteca numpy é necessária somente para definir o vetor com as classes dos exemplos.

  - O resultado é um tensor de saída com 7 linhas e 5 colunas. Onde cada linha correspondente a um exemplo e cada coluna corresponde a uma classe.

- Observa-se que essa função é também muito utilizada para codificar palavras de um texto, onde inicialmente diferentes palavras são identificadas por números inteiros.

Métodos de solução

- A solução de um problema de classificação de múltiplas classes pode ser realizada de duas formas diferentes:
  - Classificador um-contra-todos. Esse método consiste na solução de múltiplos problemas de classificação binária, um para cada classe que queremos identificar, essa é uma solução ingênua;
  - Classificador de múltiplas classes. Nesse método por meio da solução de um único problema é calculada a probabilidade de existir cada uma das classes nos dados de entrada, esse á a solução ótima.

3. Classificador um-contra-todos

- Nesse método, múltiplos sistemas de classificação binária são utilizados em paralelo para formar um problema de classificação de múltiplas classes.
- Dado um problema de classificação com N_C possíveis soluções, o método um-contra-todos consiste de N_C problemas separados de classificação binária, sendo um classificador binário para cada classe possível.
- Nesse método, durante o treinamento, o modelo executa uma sequência de classificadores binários, treinando cada um para responder a uma questão de classificação diferente.
- Em um problema de reconhecer cinco objetos diferentes em uma imagem, como por exemplo, no problema de reconhecer 5 animais diferentes: cachorro, urso, gato, cabra e cavalo. Com o classificador um-contra-todos, cinco classificadores diferentes devem ser treinados em sequência para resolver esse problema. Assim, se uma imagem de um gato é apresentada ao classificador, então, quatro dos classificadores binários devem prever essa imagem como sendo um exemplo negativo e um deles como sendo um exemplo positivo, ou seja:
  - Essa imagem contém um cachorro? Não;
  - Essa imagem contém um urso? Não;
  - Essa imagem contém um gato? Sim;
  - Essa imagem contém uma cabra? Não;
  - Essa imagem contém um cavalo? Não;
- Na Figura 1 é mostrado um classificador de múlitplas classe do tipo um-contra-todos para o exemplo anterior.
  - Nesse caso uma RNA é utilizada para cada classificador binário.
  - A função de ativação do neurônio da última camada de um classificador binário é uma função sigmóide, que produz na sua saída um número real entre 0 e 1. Assim, por exemplo, se a saída de um classificador binário para identificar se uma imagem apresenta um gato for

de 0,8, então, tem-se 80% de probabilidade da imagem mostrar um gato e 20% de probabilidade de não ser um gato.

- No classificador um-contra-todos o classificador binário que apresentar a maior probabilidade é o "vencedor".

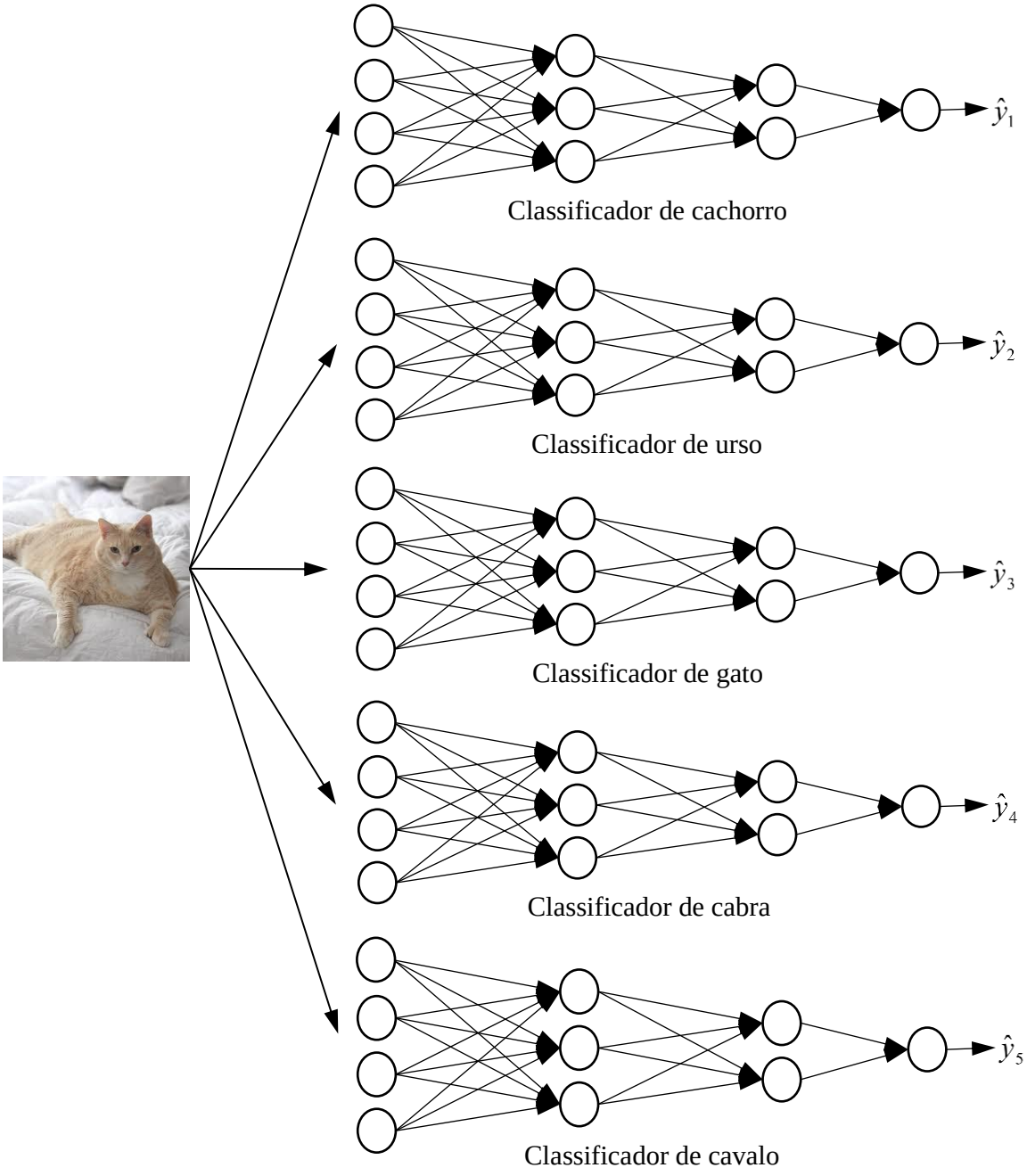

Figura 1. Classificador de múlitplas classe do tipo um-contra-todos para o exemplo de identificar cachorro, urso, gato, cabra e cavalo.

- Observa-se que essa abordagem é razoavelmente eficiente quando o número de classes é pequeno, mas se torna altamente ineficiente quando o número de classes aumenta.

4. Classificador de múltiplas classes

- Um classificador de múltiplas classes muito mais eficiente do que um classificador um-contra-todos pode ser criado usando uma rede neural deep learning, onde cada neurônio da camada de saída representa uma classe diferente.

- Na Figura 2 é apresentado uma solução possível para esse classificador, que usa a função de ativação sigmóide nos neurônios da última camada.

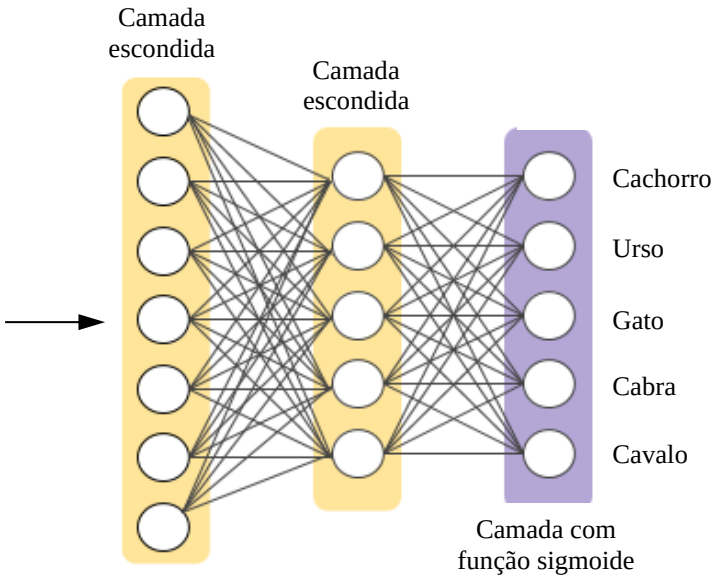

Figura 2. RNA para classificação de múltiplas classes usando função de ativação sigmoide na última camada.

- Esse classificador multi-classe é muito mais eficiente do que o do tipo um-contra-todos, mas ainda apresenta um problema.
  - Uma função sigmóide produz na sua saída um número real entre 0 e 1, assim, a soma das probabilidades de todos os neurônios da camada de saída será maior do que 1.
  - **Existe probabilidade maior do que 1?**

- A solução para essa aparente incongruência é alterar a função de ativação dos neurônios da camada de saída de forma que a soma das probabilidades calculadas por todos eles sejam sempre igual a um ⇒ **camada softmax**.

- A **camada softmax** calcula uma probabilidade para cada classe em um problema de classificação de múltiplas classes, cuja soma é sempre igual a um. Com essa restrição adicional o processo de treinamento da RNA é muito mais eficiente do que se for utilizada a função sigmoide na saída da RNA.

- A razão para que o treinamento da RNA se torna mais eficiente é que existe uma restrição imposta conjuntamente a todas as saídas da RNA e com isso qualquer divergência que pode ocorrer durante o treinamento tende a ser autocontrolada.

- Na Figura 3 é apresentado um classificador de múltiplas classes com a camada de saída sendo do tipo softmax. O número de neurônios de uma camada softmax é igual ao número de classes que se pretende identificar.

- Ressalta-se que camadas softmax somente são usadas para classificação de múltiplas classes e são sempre a última camada da RNA.

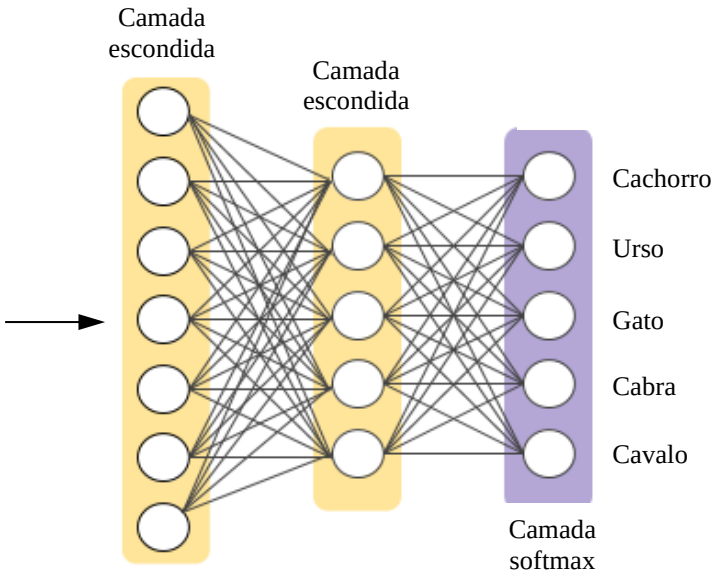

Figura 3. RNA para classificação de múltiplas classes com camada softmax.

- As equações que implementam a propagação para frente em uma camada do tipo softmax para cada exemplo são similares às de uma camada comum. Para um problema de N_C classes diferentes, temos:

  - **Entrada da camada** softmax $\Rightarrow \mathbf{a}^{[L-1]}$ = vetor de ativações dos neurônios da penúltima camada da RNA (L–1) $\rightarrow$ dimensão ($n^{[L-1]}$, 1).

  - **Saída da RNA (saída da camada softmax)** $\Rightarrow \mathbf{a}^{[L]}$ = vetor de ativações da última camada $\rightarrow$ dimensão (N, 1). Observe que o número de neurônios de uma camada softmax é igual ao número de classes que se deseja classificar, ou seja, $n^{[L]} = N_C$.

  - **Estados dos neurônios:**

$$\boxed{\mathbf{z}^{[L]} = \mathbf{W}^{[L]}\mathbf{a}^{[L-1]} + \mathbf{b}^{[L]}} \qquad (1)$$

onde:

$\mathbf{z}^{[L]}$ = vetor de estados da camada softmax $\rightarrow$ dimensão (N_C, 1);

$\mathbf{W}^{[L]}$ = matriz de pesos da camada softmax $\rightarrow$ dimensão (N_C, $n^{[L-1]}$);

$\mathbf{b}^{[l]}$ = vetor de vieses da camada l $\rightarrow$ dimensão (N_C, 1).

- **Saídas da RNA (saídas da camada softmax):**

$$\hat{y}_k = a_k^{[L]} = \frac{e^{z_k^{[L]}}}{\sum_{j=1}^{N_C} e^{z_j^{[L]}}}, \text{ para } k = 1,...,N_C \quad (2)$$

- Em palavras, na camada softmax aplica-se a função exponencial em cada estado da camada e depois normaliza esses valores dividindo pela soma de todas as exponenciais para garantir que a soma de todas as saídas da camada seja igual a um.

- Retornando ao exemplo de identificar animais em uma imagem, na Tabela 1 são apresentados exemplos de valores das saídas dos neurônios para o caso da camada de saída ter neurônios com função sigmoide e para o caso da camada de saída ser softmax.

Tabela 1. Exemplo de resultados de uma camada de saída com função de ativação sigmóide e de uma softmax.

Classe	Sigmóide	Softmax
Cachorro	0,05	0,015
Urso	0,10	0,032
Gato	0,08	0,025
Cabra	0,75	0,86
Cavalo	0,20	0,071
Soma das saídas	**1,18**	**1,00**

- Observa-se da Tabela 1 que a soma das saídas da camada com função de ativação sigmoide é diferente de um e da camada softmax é igual a um.

- Para determinar a classe que uma RNA identificou nos dados de entrada basta achar o valor máximo das saídas da camada softmax usando uma função que calcula o índice do valor máximo de um vetor de números reais.

Importante:

- Em problemas de classificação de múltiplas classes quando no mesmo exemplo têm-se simultaneamente objetos de classes diferentes, então, tem-se que:
  - nesse caso não se pode usar a softmax;
  - deve usar função sigmoide na camada de saída;
  - no exemplo de classificação de animais em uma imagem, se as imagens contiverem vários tipos de animais simultaneamente, então, devem-se usar vários problemas de classificação binária usando uma RNA do tipo da Figura 1 ou da Figura 2.

5. Função de custo (entropia cruzada de múltiplas classes)

- A função de custo utilizada para classificação de múltiplas classes é a função **entropia cruzada de múltiplas classes** (**"categorical cross entropy"**), que é a seguinte:

$$L(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}) = -\sum_{k=1}^{N_C} \left[y_k^{(i)} \log(\hat{y}_k^{(i)}) \right] \tag{4}$$

onde:

$y_k^{(i)}$ = *k*-ésimo elemento do vetor de saída com a classe do *i*-ésimo exemplo de treinamento;

$\hat{y}_k^{(i)}$ = *k*-ésimo elemento da camada softmax para o *i*-ésimo exemplo de treinamento;

N_C = número de classes a serem identificadas.

- Nota-se que somente um elemento do vetor de saída com a classe real, $\mathbf{y}^{(i)}$, é diferente de zero, portanto, todos os termos da somatória da equação (4) são iguais a zero a menos do termo que representa a classe real desse exemplo.

- Por exemplo, em um problema com cinco classes, a saída real de um exemplo de treinamento e a saída calculada pela RNA são as seguintes:

$$\mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{y}} = \begin{bmatrix} 0{,}05 \\ 0{,}75 \\ 0{,}10 \\ 0{,}08 \\ 0{,}20 \end{bmatrix} \tag{5}$$

Como todos os elementos do vetor $\mathbf{y}$ são iguais a zero a menos do 2º elemento, então, a aplicação da função de erro (eq. 4) para esse exemplo de treinamento resulta em:

$$L(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}) = -\log(\hat{y}_2) = -\log(0{,}75) = 0{,}29 \tag{6}$$

O objetivo do treinamento é minimizar a função de erro. Para isso, deve-se fazer com que $-\log(\hat{y}_2)$ diminua e, portanto, deve-se que aumentar $\hat{y}_2$ (k = 2 na equação 4).

- A função de custo usada com a função de entropia cruzada de múltiplas classes consiste da soma da função de erro para todos os exemplos de treinamento, ou seja:

$$J(\mathbf{W}, \mathbf{B}) = \frac{1}{m} \sum_{i=1}^{m} \left\{ -\sum_{k=1}^{N_C} \left[y_k^{(i)} \log(\hat{y}_k^{(i)}) \right] \right\} \tag{7}$$

onde

$\mathbf{W}$ e $\mathbf{B}$ representam genericamente os parâmetros da RNA;

m = número de exemplos de treinamento.

6. Métrica de desempenho

- Em um problema de classificação de múltiplas classes a métrica mais utilizada é a exatidão.
- **Para um problema de classificação de múltiplas classes a exatidão (*acc*) é calculada da seguinte forma:**

$$acc = 1 - \frac{1}{2m}\sum_{i=1}^{m}\left\|\mathbf{c}^{(i)} - \mathbf{y}^{(i)}\right\|_1 \tag{8}$$

onde:

m = número de exemplos;

$\mathbf{c}^{(i)}$ = vetor ("one-hot-encoded") com a classe prevista pela RNA para o i-ésimo exemplo;

$\mathbf{y}^{(i)}$ = vetor de saída ou classe real do i-ésimo exemplo.

- A saída de uma RNA em um problema de classificação de múltiplas classes é um vetor real com elementos entre 0 e 1 para cada exemplo analisado. Assim, dada a saída do i-ésimo exemplo, $\hat{\mathbf{y}}^{(i)}$, devemos decidir em qual classe esse exemplo pertence e para isso fazemos o seguinte;
  - Criamos o vetor com a classe calculada, $\mathbf{c}^{(i)}$, com a mesma dimensão do vetor $\hat{\mathbf{y}}^{(i)}$;
  - Identificamos o índice do elemento vetor $\hat{\mathbf{y}}^{(i)}$ cujo valor é o máximo (i_{max});
  - No vetor de classe calculada, $\mathbf{c}^{(i)}$, no elemento de índice i_{max} colocamos o valor 1 e em todos os outros elementos colocamos zero.
- Observe que a exatidão definida pela equação (8) representa a fração de exemplos classificada corretamente.

7. Como fazer no Keras

- No Keras uma camada softmax é definida como sendo um tipo de função de ativação.
- No código a seguir é mostrado um exemplo onde é configurada uma RNA de 3 camadas, sendo que a última camada é softmax, para um problema de classificação de múltiplas classes.

```
# Importa bibliotecas do TensorFlow
from tensorflow.keras import models
from tensorflow.keras import layers

# Configuração da RNA
rna = models.Sequential()
rna.add(layers.Dense(64, activation='relu', input_shape=(1024,)))
rna.add(layers.Dense(64,activation='relu'))
rna.add(layers.Dense(10, activation='softmax'))
rna.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 64)                65600
_________________________________________________________________
dense_1 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_2 (Dense)              (None, 10)                650
=================================================================
Total params: 70,410
Trainable params: 70,410
Non-trainable params: 0
```

- A RNA configurada nesse código tem as seguintes características
  - Número de camadas: $L = 3$;
  - Número de elementos no vetor de entrada: $n_x = 1024$;
  - Número de neurônios da 1ª e 2ª camadas: $n^{[1]} = n^{[2]} = 64$;
  - Função de ativação da 1ª e 2ª camadas: Relu;
  - Número de classes do problema: $N_C = 10$;
  - Função de ativação da camada de saída: softmax.

➢ A compilação dessa RNA deve ser realizada usando a **função de custo entropia cruzada de múltiplas classes ("categorical_crossentropy")** e pode ser realizada, por exemplo, usando o método de otimização RMSprop, como mostrado a seguir.

```
rna.compile(optimizer='rmsprop', loss='categorical_crossentropy',
            metrics=['accuracy'])
```

- A métrica mais utilizada nos problemas de classificação de múltiplas classes é a exatidão, que representa a fração do número de classes classificadas de forma correta.

➢ O treinamento da RNA é feita da mesma forma como é realizada para outros casos, como por exemplo, mostrado no código a seguir.

```
history = rna.fit(x_train, y_train, epochs=100,
                  validation_data=(x_val, y_val))
```

- A identificação das classes calculadas pela RNA é feita usando-se a função numpy `argmax`. Admitindo-se que as saídas de todos os exemplos estão na matriz **Y**, de dimensão (m, N_C), a matriz com as classes de todos os elementos pode ser criada de acordo com o código a seguir:

```
# Importa biblioteca numpy
import numpy as np

# Identifica índices dos máximos de cada coluna
columns = np.argmax(y,1)
# Cria vetor com índices das colunas
rows = np.arange(m)
# Cria matriz de classes com zeros
c = np.zeros((m,NC))
# Define a classe correta
c[rows,columns] = 1
```

  - Note que nesse exemplo a saída da RNA para cada exemplo é um vetor linha.
  - Nesse código *m* é o número de exemplos e *NC* é o número de classes.

8. Exemplo de problema de classificação de múltiplas classes

- Nesse problema queremos determinar o tipo de vestuário que é mostrado em uma imagem.
- O problema consiste em dada uma imagem, a RNA avalia a probabilidade de existirem determinados tipos de vestuário na imagem e determina qual o tipo mais provável entre dez possíveis.
- O objetivo desse problema é treinar uma RNA que recebe como entrada uma imagem e determina qual tipo de vestuário é mostrado na imagem.
- O banco de dados usado nesse exemplo é o Fashion-MNIST, que consiste de imagens de artigos de vestuário da Zalando. Esse banco de dados pode ser obtido no link https://github.com/zalandoresearch/fashion-mnist.
- Observa-se que Keras já possui esse banco de dados e para usá-lo basta importá-lo com um comando, como mostrado no código a seguir.

```
# Importa bancos de dados do Keras
from tensorflow.keras.datasets import fashion_mnist

# Carrega dados da Fashion-MNIST em tenores
(x_train, y_train),(x_test,y_test) = fashion_mnist.load_data()
```

- Observa-se que o Keras possui diversos bancos de dados, cujos exemplos já estão em tensores, facilitando o seu uso. No manual do TensorFlow é informado quais bancos de dados estão disponíveis e como fazer para utilizá-los (https://www.tensorflow.org/api_docs/python/tf/keras/datasets).

- O banco de dados Fashion MNIST possui 60.000 exemplos de treinamento e 10.000 exemplos de teste. Cada exemplo consiste de uma imagem em tons de cinza, de dimensão 28x28 pixels, associada a um rótulo de 10 classes. Algumas dessas imagens estão mostradas na Figura 4.

- O banco de dados original da MNIST consiste de imagens de caracteres escritos a mão. As imagens de vestuário da Zelando tentam aprimorar o banco de dados original do MNIST.

- Os dados de imagens da MNIST são muito utilizados na área de IA/ML como referência para validar sistemas de aprendizado de máquina. De fato os bancos de dados da MNIST são frequentemente os primeiros a serem testados com um novo sistema. Se não funcionar com esses dados, então, não vai funcionar com nada. Se funcionar com esses dados, pode ser que funcione ou não com outros dados.

- **Características dos dados:**
  - Cada imagem tem 28 pixels de altura e 28 pixels de largura, totalizando 784 pixels no total;
  - O valor da intensidade luminosa de cada pixel da imagem é um número inteiro entre 0 e 255;
  - Cada exemplo consiste de um vetor linha com 785 elementos;
  - A primeira coluna do vetor consiste nos rótulos da classe que representa o artigo de vestuário;
  - O restante das colunas contém os valores dos pixels da imagem associada.

- **As possíveis classes (rótulos) presentes nas imagens são as seguintes:**
  - 0 – camiseta;
  - 1 – calça;
  - 2 – pulôver;
  - 3 – vestido;
  - 4 – casaco;
  - 5 – sandália;
  - 6 – camisa;
  - 7 – tênis.
  - 8 – bolsa;
  - 9 – bota de cano curto.

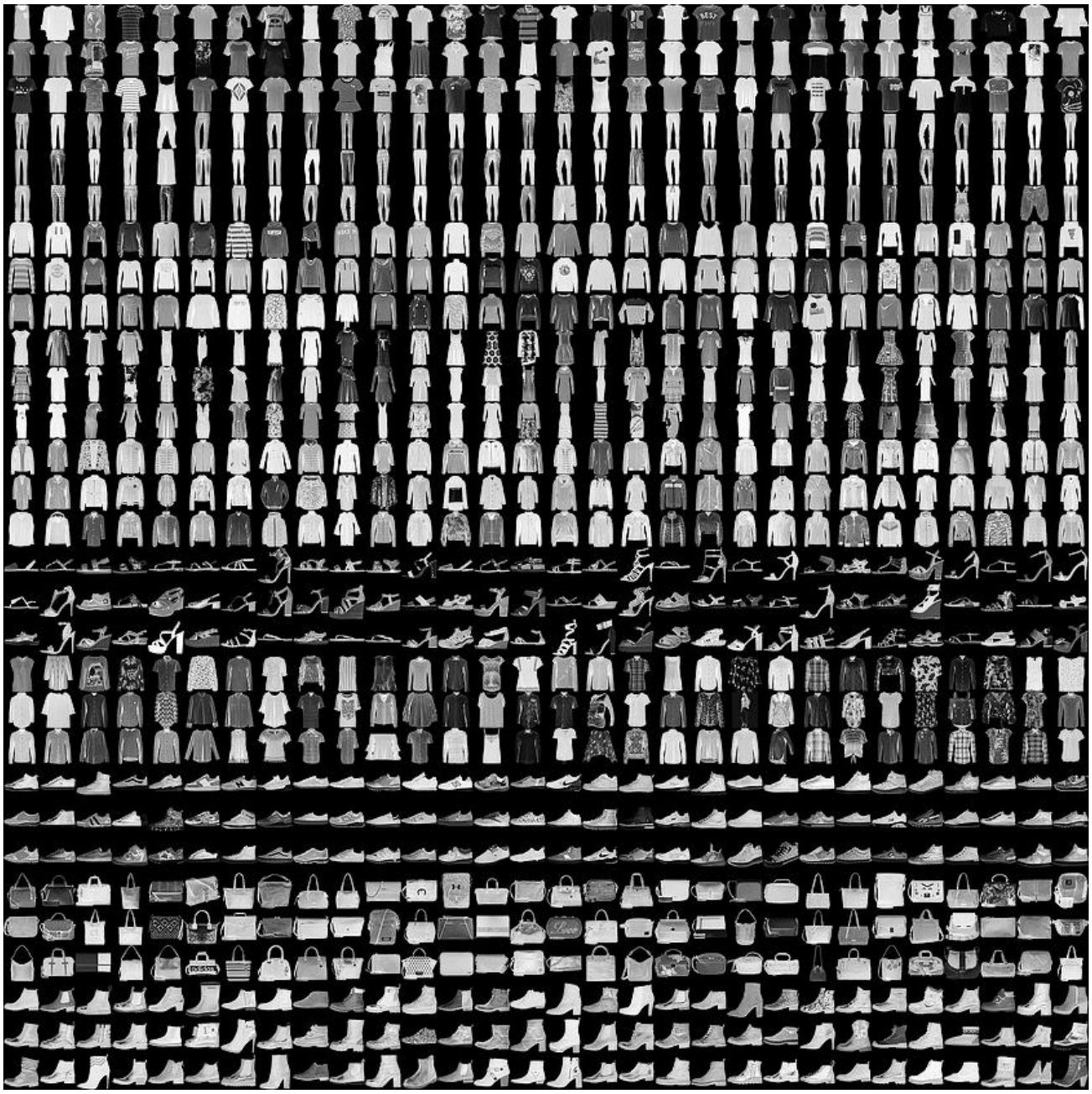

Figura 4. Exemplos de imagens do banco de dados Fashion-MNIST (https://github.com/zalandoresearch/fashion-mnist).