AULA 6
RNA PROFUNDA (DEEP-LEARNING)
GRADIENTE DESCENDENTE

1. Objetivos

- Motivação para usar RNAs profundas.
- Apresentar com detalhes as RNAs profundas (deep learning).
- Apresentar os blocos construtores das RNAs.
- Apresentar a vetorização da Retro-Propagação para RNAs deep-learning.

2. Motivação para RNAs profundas (deep-learning)

- Como já visto uma RNA profunda (deep-leanring) possui muitas camadas intermediárias.
- **Para que utilizar múltiplas camadas em uma RNA?**

  RNA de múltiplas camadas é capaz de representar funções complexas.
- Intuitivamente podemos pensar que cada camada da RNA extrai (aprende) alguma informação nova sobre o problema $\Rightarrow$ assim múltiplas camadas conseguem aprender mais atributos do problema e, assim, é capaz de associar mais informações para obter melhores resultados.

- **Processamento de imagens:**
  - Tudo ocorre como se cada camada aprendesse a extrair alguma característica da imagem.
  - Primeiras camadas extraem características simples (bordas, cores, cantos etc).
  - Camadas finais combinam características obtidas nas primeiras camadas para extrair atributos mais complexo (olho, boca, nariz etc).
  - Na Figura 1 é apresentado um exemplo de processamento de uma imagem por uma RNA deep-learning para identificar pessoas pela face.

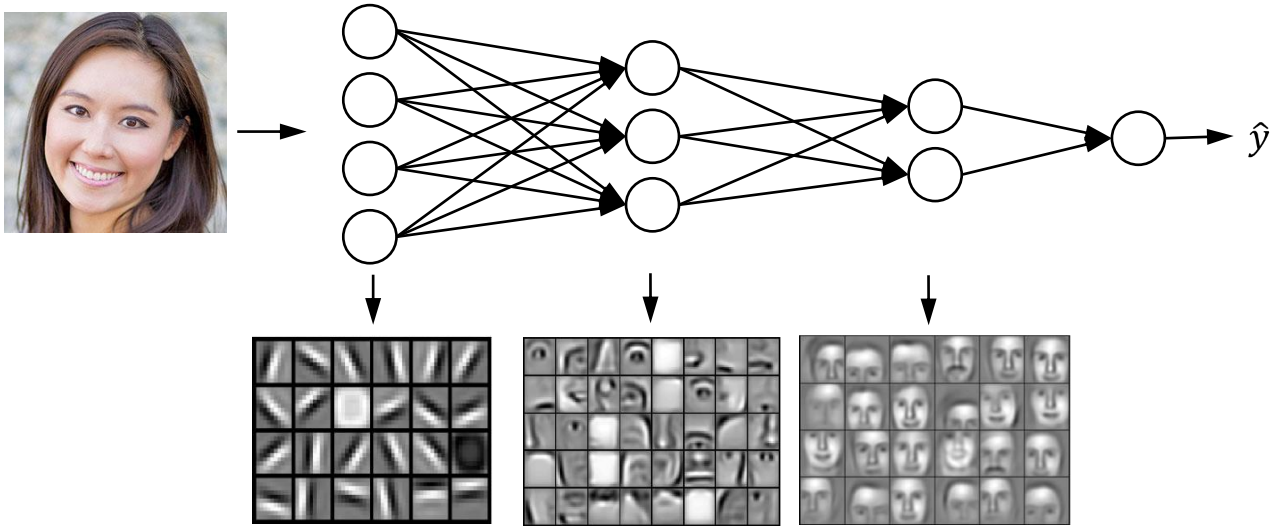

Figura 1. Exemplo de processamento de uma imagem por uma RNA profunda (adaptado de Andrew Ng, deeplearning.ai).

- **Processamento de áudio:**

  - Da mesma forma que realizado para imagens cada camada aprende a extrair alguma característica diferente do som.
  - Primeiras camadas extraem características simples (formas de onda, fonemas etc).
  - Camadas finais combinam características obtidas nas primeiras camadas para extrair atributos mais complexos (palavras, frases etc).
  - Na Figura 2 é apresentado um exemplo de processamento de um sinal de áudio por uma RNA profunda para reconhecer frases.

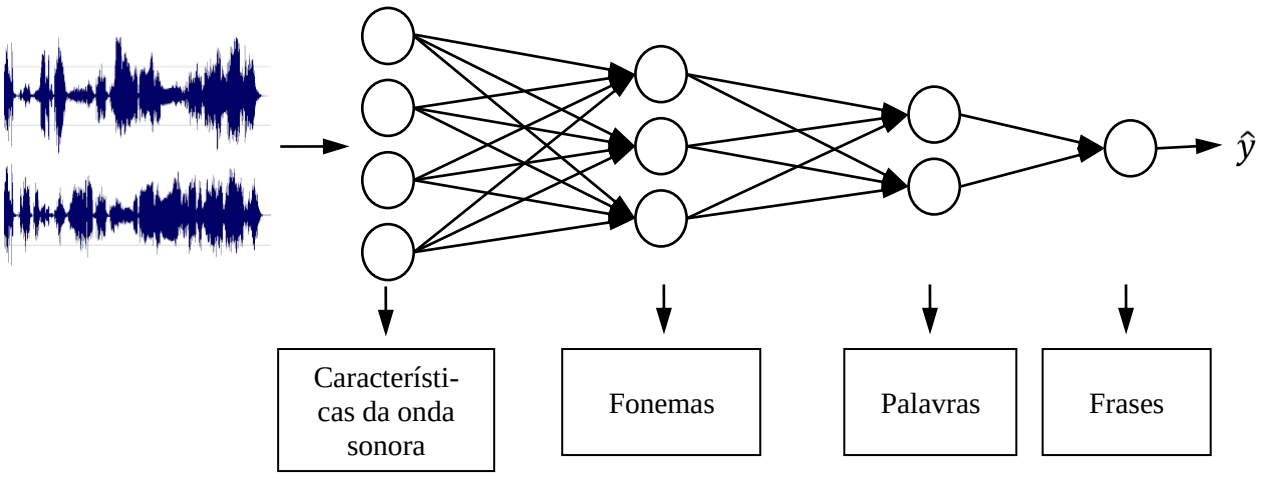

Figura 2. Exemplo de processamento de um áudio por RNA profunda.

- **Cálculo de funções lógicas:**

  - Existem funções que são possíveis de calcular com uma RNA profunda com poucos neurônios. Essas mesmas funções podem exigir um número exponencialmente maior se fosse utilizada uma RNA de uma única camada intermediária.

  - A função da equação (1) calcula um XOR de *n* números:

    $$y = x \text{ XOR } x_2 \text{ XOR } x_3 \text{ XOR } .... \text{ XOR } x_n \quad (1)$$

  - A sua implementação pode ser feita com poucos neurônios em uma RNA de $\log_2(n)$ camadas intermediárias, enquanto que exigiria 2^{n-1} neurônios se for utilizada uma RNA de uma única camada intermediária, como mostrado na Figura 3 para 8 números.

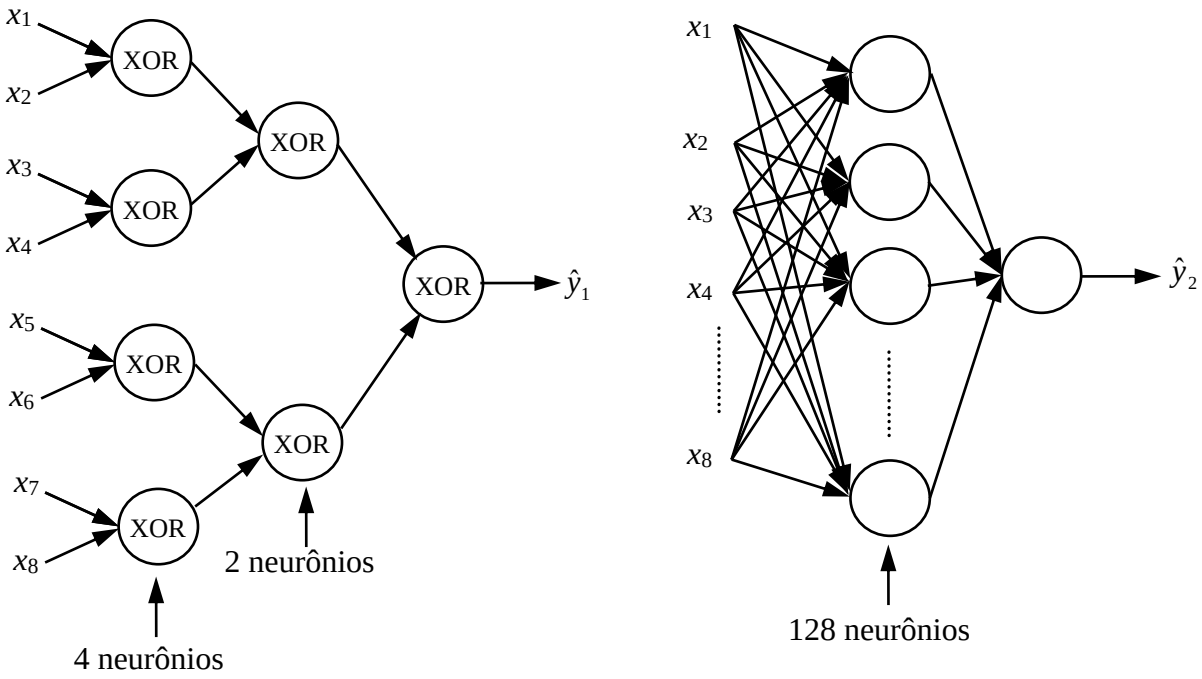

Figura 3. Implementação da função XOR de 8 números por uma RNA. A RNA de várias camadas precisa somente de 6 neurônios e a RNA rasa de 128 neurônios.

3. Blocos construtores de uma RNA

- Como visto uma RNA profunda (deep-learning) possui múltiplas camadas como mostrado na Figura 4.

- As RNAs consistem de um conjunto de camadas dispostas uma após a outra.

- Cada camada tem as suas características ⇒ tipo, número de neurônios e função de ativação.

- Além das camadas totalmente conectadas (que chamamos de densa) existem outros tipos de camadas ⇒ as camadas convolucionais e recorrentes que podem ser misturadas em uma única RNA.

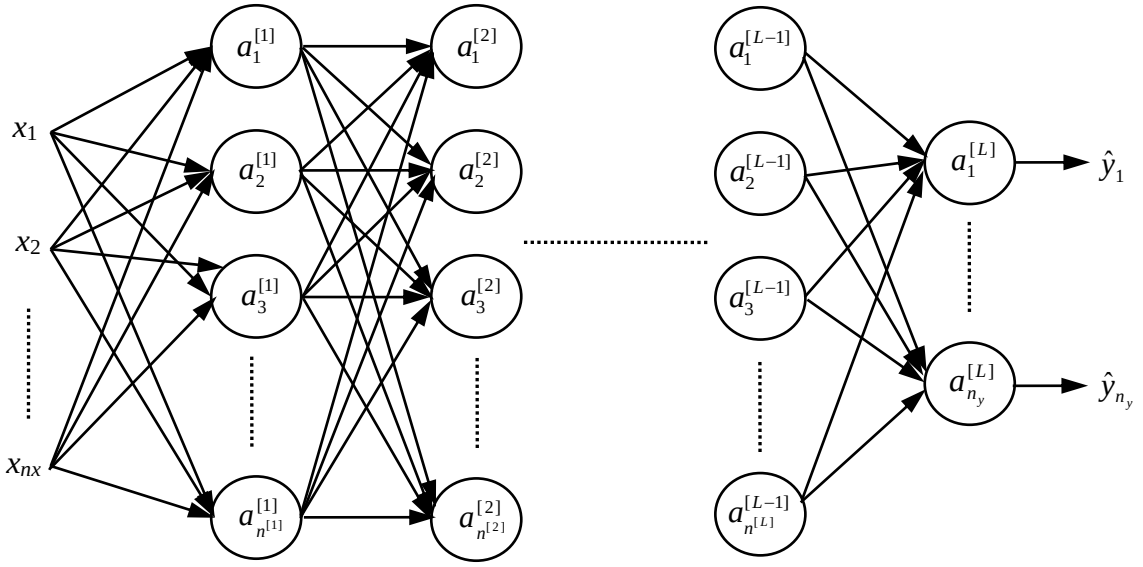

Figura 4. RNA profunda (deep-learning) de L camadas com n_x entradas e n_y saídas.

- As camadas podem ser consideradas como sendo blocos construtores das RNAs:
  - Cálculos realizados nas camadas, independentemente das suas características, são todos similares tanto para a propagação para frente como para a retro-propagação.
  - Softwares de desenvolvimento de RNAs trabalham com as camadas.
- Uma RNA pode ser representada por um diagrama de blocos como mostrado na Figura 5, onde cada camada é um bloco de cálculos.

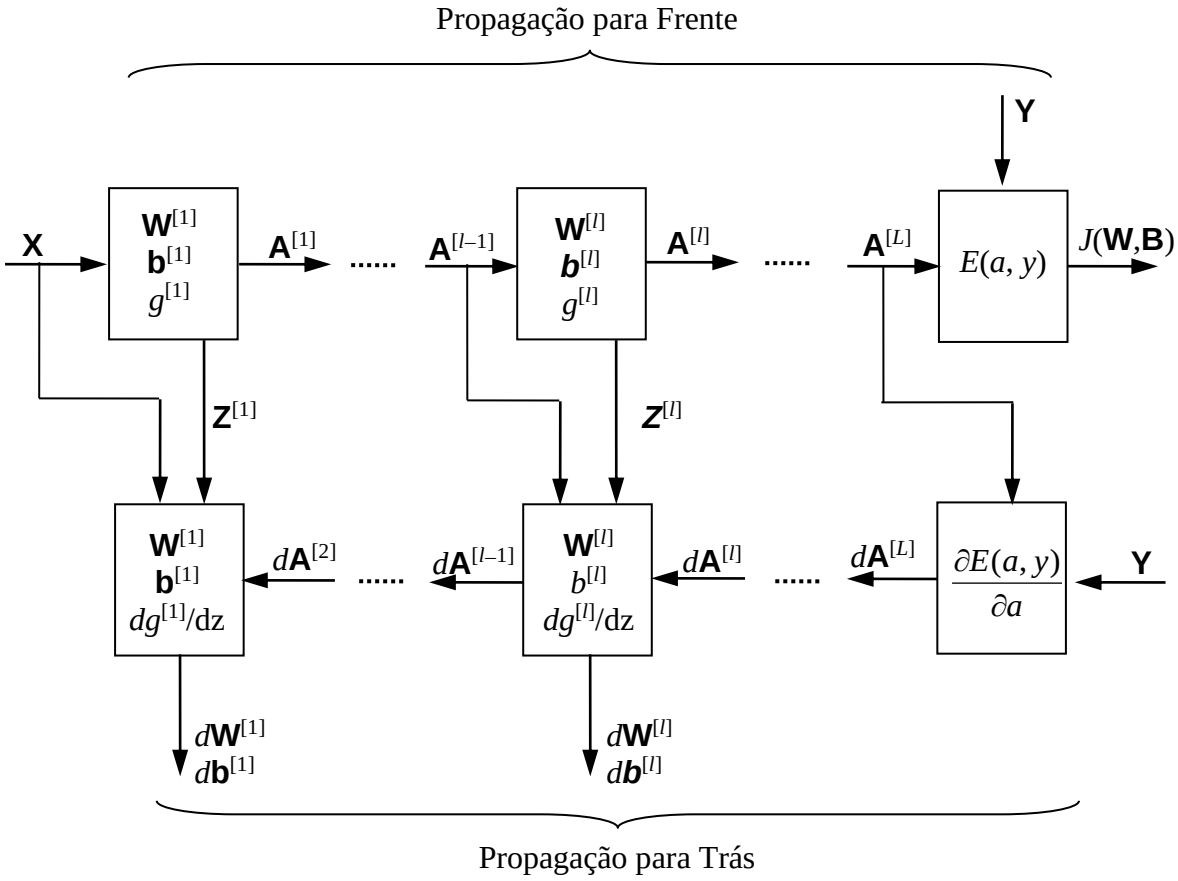

Figura 5. Diagrama de blocos de uma RNA de *L* camadas.

4. Cálculo não vetorizado em uma camada

- **Principais parâmetros da RNA:**
  - Número de entradas: n_x;
  - Número de saídas: n_y;
  - Número de exemplos de treinamento: *m;*
  - Número de camadas: *L*;
  - Número de neurônios da camada *l*: $n^{[l]}$.

- **Propagação para frente na camada *l* para cada exemplo de treinamento:**
  - **Entrada da camada** $\Rightarrow \mathbf{a}^{[l-1](i)}$ = vetor de ativações dos neurônios da camada anterior (*l*–1) para o *i*-ésimo exemplo $\rightarrow$ dimensão ($n^{[l-1]}$, 1).

- **Saída da camada** $\Rightarrow \mathbf{a}^{[l](i)}$ = vetor de ativações da camada l para o i-ésimo exemplo $\rightarrow$ dimensão $(n^{[l]}, 1)$.

- **Cálculos realizados para cada exemplo de treinamento:**

$$\mathbf{z}^{[l](i)} = \mathbf{W}^{[l]}\mathbf{a}^{[l-1](i)} + \mathbf{b}^{[l]} \quad (2)$$

$$\mathbf{a}^{[l](i)} = g^{[l]}(\mathbf{z}^{[l](i)}) \quad (3)$$

onde:

  - $\mathbf{z}^{[l](i)}$ = vetor de estados da camada l para o i-ésimo exemplo $\rightarrow$ dimensão $(n^{[l]}, 1)$;
  - $g^{[l]}$ = função de ativação da camada l;
  - $\mathbf{W}^{[l]}$ = matriz de pesos da camada $l \rightarrow$ dimensão $(n^{[l]}, n^{[l-1]})$;
  - $\mathbf{b}^{[l]}$ = vetor de vieses da camada $l \rightarrow$ dimensão $(n^{[l]}, 1)$.

- No caso da primeira camada temos que: $\mathbf{a}^{[0](i)} = \mathbf{x}^{(i)}$ (vetor de entrada do i-ésimo exemplo).

- No caso da última camada temos que $\mathbf{a}^{[L](i)} = \hat{\mathbf{y}}^{(i)}$ (vetor de saída do i-ésimo exemplo).

➢ **Retro-propagação na camada *l*:**

- **Entradas da camada:**
  - $d\mathbf{a}^{[l](i)}$ = vetor de derivadas parciais da função de custo em relação às ativações da camada l para o i-ésimo exemplo $\rightarrow$ dimensão $(n^{[l]}, 1)$;
  - $\mathbf{z}^{[l](i)}$ = vetor de estados da camada l para o i-ésimo exemplo $\rightarrow$ dimensão $(n^{[l]}, 1)$;
  - $\mathbf{a}^{[l-1](i)}$ = vetor de ativações dos neurônios da camada anterior $(l-1)$ para o i-ésimo exemplo $\rightarrow$ dimensão $(n^{[l-1]}, 1)$.

- **Saídas da camada:**
  - $d\mathbf{a}^{[l-1](i)}$ = vetor de derivadas parciais da função de custo em relação às ativações da camada $l-1 \rightarrow$ dimensão $(n^{[l-1]}, 1)$;
  - $d\mathbf{W}^{[l]}$ = matriz de derivadas parciais da função de custo em relação aos pesos da camada $l \rightarrow$ dimensão $(n^{[l]}, n^{[l-1]})$;
  - $d\mathbf{b}^{[l]}$ = vetor de derivadas parciais da função de custo em relação aos vieses da camada $l \rightarrow$ dimensão $(n^{[l]}, 1)$.

- **Cálculos realizados para cada exemplo de treinamento:**

$$d\mathbf{z}^{[l](i)} = d\mathbf{a}^{[l](i)} * \frac{dg^{[l]}(\mathbf{z}^{[l](i)})}{dz} \quad (4)$$

$$d\mathbf{W}^{[l]} += d\mathbf{z}^{[l](i)}\mathbf{a}^{[l-1](i)T} \quad (5)$$

Multiplicação elemento por elemento

$$d\mathbf{b}^{[l]} += d\mathbf{z}^{[l](i)} \tag{6}$$

$$d\mathbf{a}^{[l-1](i)} = \mathbf{W}^{[l]T} d\mathbf{z}^{[l](i)} \tag{7}$$

onde:

- $d\mathbf{z}^{[l](i)}$ = vetor de derivadas parciais da função de custo em relação aos estados da camada l para o i-ésimo exemplo → dimensão ($n^{[l]}$, 1);
- $dg^{[l]}$/dz = derivada da função de ativação da camada l.

- No caso da primeira camada não se calcula $d\mathbf{a}^{[0](i)}$.

- No caso da última camada, temos que $d\mathbf{a}^{[L](i)}$ é a derivada parcial da função de custo em relação às saídas da RNA para o i-ésimo exemplo, ou seja, em relação às ativações da última camada, dado por:

$$d\mathbf{a}^{[L](i)} = \frac{\partial J(\mathbf{W},\mathbf{B})}{\partial \hat{\mathbf{y}}^{(i)}} \rightarrow \text{dimensão } (n^{[L]} = n_y, 1) \tag{8}$$

5. Cálculo vetorizado em uma camada

- No cálculo vetorizado de uma RNA todos os exemplos são calculados de uma única vez sem a necessidade de um comando de repetição para passar por cada exemplo.

- **Principais parâmetros da RNA:**
  - Número de entradas: n_x;
  - Número de saídas: n_y;
  - Número de exemplos de treinamento: m;
  - Número de camadas: L;
  - Número de neurônios da camada l: $n^{[l]}$.

- **Propagação para frente na camada *l*:**
  - **Entrada da camada** ⇒ $\mathbf{A}^{[l-1]}$ = matriz de ativações dos neurônios da camada anterior (l–1) → dimensão ($n^{[l-1]}$, m), cada coluna da matriz é referente à um exemplo de treinamento.
  - **Saída da camada** ⇒ $\mathbf{A}^{[l]}$ = matriz de ativações da camada l → dimensão ($n^{[l]}$, m), cada coluna da matriz é referente à um exemplo de treinamento.
  - **Cálculo realizado de uma única vez para todos os exemplos de treinamento:**

$$\mathbf{Z}^{[l]} = \mathbf{W}^{[l]}\mathbf{A}^{[l-1]} + \mathbf{b}^{[l]} \tag{9}$$

$$\boxed{\mathbf{A}^{[l]} = g^{[l]}(\mathbf{Z}^{[l]})} \tag{10}$$

onde $\mathbf{Z}^{[l]}$ = matriz de estados da camada $l \rightarrow$ dimensão $(n^{[l]}, m)$, cada coluna da matriz é referente à um exemplo de treinamento.

- No caso da primeira camada temos que: $\mathbf{A}^{[0]} = \mathbf{X} \Rightarrow$ matriz com as entradas de todos os exemplos, onde cada coluna da matriz $\mathbf{X}$ é o vetor de entradas de um exemplo de treinamento:

$$\mathbf{X} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(m)} \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{(n_x, m)}$$

- No caso da última camada temos que $\mathbf{A}^{[L]} = \hat{\mathbf{Y}}$ (matriz de saída). As m saídas são agrupadas em uma matriz de saídas, onde cada coluna é o vetor de saída da RNA para um exemplo de treinamento:

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{\mathbf{y}}^{(1)} & \hat{\mathbf{y}}^{(2)} & \cdots & \hat{\mathbf{y}}^{(m)} \end{bmatrix}_{(n_y, m)}$$

➢ **Retro-propagação na camada *l*:**

- **Entradas da camada:**
  - $d\mathbf{A}^{[l]}$ = matriz de derivadas parciais da função de custo em relação às ativações da camada $l \rightarrow$ dimensão $(n^{[l]}, m)$, cada coluna da matriz é referente à um exemplo de treinamento;
  - $\mathbf{Z}^{[l]}$ = matriz de estados da camada $l \rightarrow$ dimensão $(n^{[l]}, m)$, cada coluna da matriz é referente à um exemplo de treinamento;
  - $\mathbf{A}^{[l-1]}$ = matriz de ativações dos neurônios da camada anterior $(l-1) \rightarrow$ dimensão $(n^{[l-1]}, m)$, cada coluna da matriz é referente à um exemplo de treinamento.
- **Saída da camada:**
  - $d\mathbf{A}^{[l-1]}$ = matriz de derivadas parciais da função de custo em relação às ativações da camada $l-1 \rightarrow$ dimensão $(n^{[l-1]}, m)$, cada coluna da matriz é referente à um exemplo de treinamento;
  - $d\mathbf{W}^{[l]}$ = matriz de derivadas parciais da função de custo em relação aos pesos da camada $l \rightarrow$ dimensão $(n^{[l]}, n^{[l-1]})$;
  - $d\mathbf{b}^{[l]}$ = vetor de derivadas parciais da função de custo em relação aos vieses da camada $l \rightarrow$ dimensão $(n^{[l]}, 1)$.
- **Cálculo realizado de uma única vez para todos os exemplos de treinamento:**

$$\boxed{d\mathbf{Z}^{[l]} = d\mathbf{A}^{[l]} * \frac{dg^{[l]}(\mathbf{Z}^{[l]})}{dz}} \tag{11}$$

Multiplicação elemento por elemento

$$d\mathbf{W}^{[l]} = \frac{1}{m} d\mathbf{Z}^{[l]} \mathbf{A}^{[l-1]T} \tag{12}$$

$$d\mathbf{b}^{[l]} = \frac{1}{m} \sum_{i=1}^{m} d\mathbf{z}^{[l](i)} \tag{13}$$

$$d\mathbf{A}^{[l-1]} = \mathbf{W}^{[l]T} d\mathbf{Z}^{[l]} \tag{14}$$

onde:

- $d\mathbf{Z}^{[l]}$ = matriz de derivadas parciais da função de custo em relação aos estados da camada $l \rightarrow$ dimensão ($n^{[l]}$, m), cada coluna da matriz é referente à um exemplo de treinamento;
- Na equação (13) a somatória é realizada nas linhas da matriz $\mathbf{Z}^{[l]}$, ou seja, o elemento k do vetor $\mathbf{b}^{[l]}$ é igual à soma de todos os m elementos da linha k da matriz $\mathbf{Z}^{[l]}$.

- No caso da primeira camada não se calcula $d\mathbf{A}^{[0]}$.
- No caso da última camada, temos que $d\mathbf{A}^{[L]}$ é a derivada parcial da função de custo em relação às saídas da RNA (ativações da última camada) para todos os exemplos, ou seja:

$$d\mathbf{A}^{[L]} = \frac{\partial J(\mathbf{W},\mathbf{B})}{\partial \hat{\mathbf{Y}}} \rightarrow \text{dimensão } (n^{[L]} = n_y,\ m) \tag{15}$$

Note que cada coluna da matriz $d\mathbf{A}^{[L]}$ é referente à um exemplo de treinamento.

6. Atualização dos parâmetros da RNA

- Após o cálculo dos gradientes deve ser realizada a atualização de cada parâmetro da RNA.
- A atualização dos parâmetros é feita de forma vetorizada para cada camada individualmente.
- Para uma RNA de L camadas, a equações de atualização são as seguintes:

A atualização dos parâmetros da RNA também pode ser implementada vetorizadamente para todos os exemplos por uma única operação matricial em cada época do treinamento, ou seja:

$$\mathbf{W}^{[l]} = \mathbf{W}^{[l]} - \alpha\, d\mathbf{W}^{[l]},\ \text{para } l = 1,...,L \tag{16}$$

$$\mathbf{b}^{[l]} = \mathbf{b}^{[l]} - \alpha\, d\mathbf{b}^{[l]},\ \text{para } l = 1,...,L \tag{17}$$

onde α é a taxa de aprendizagem, como já visto.

7. Processo de treinamento de uma RNA

- Como já visto, o treinamento de um RNA é um processo iterativo no qual se deseja calcular os parâmetros da RNA para que ela aprenda os padrões contidos nos exemplos fornecidos.

- **Processo de treinamento:**

  1) Inicializar os parâmetros da RNA;

  2) Executar a RNA com os exemplos e obter as saídas previstas;

  3) Calcular os erros entre as saídas desejadas e as previstas pela RNA computando a função de custo;

  4) Calcular o gradiente da função de custo em relação à todos os parâmetros da RNA.

  5) Atualizar os parâmetros da RNA.

  6) Repetir os passos 2 a 5 quantas vezes for necessário para os parâmetros convergirem, ou a função e custo atingir um valor mínimo desejado, ou até o número máximo de épocas for alcançado.

- Esse processo de treinamento deve ser repetido inúmeras vezes até ser obtida uma RNA que apresente o desempenho desejado.

- O desenvolvimento de uma RNA que apresenta o desempenho desejado é um processo iterativo de solução de um problema.

- **O processo de solução de um problema usando uma RNA consiste de um ciclo que envolve as seguintes etapas:**

  1. Escolha inicial dos hiperperâmetros (número de camadas, número de neurônios em cada camada, tipos de funções de ayivação etc);

  2. Configuração da RNA;

  3. Processo de treinamento;

  4. Avaliação do desempenho da RNA;

  5. Ajuste dos hiperparâmetros;

  6. Repetir os passos 2 a 5 quantas vezes for necessário para obter uma RNA com o desempenho deejado.

- Veremos com detalhes esse processo de solução de problemas usando RNAs e os diversos hiperparâmetros de uma RNA que podem ser ajustados para melhorar o seu desempenho.