



Módulo 9 – Sessão 1

INTRODUCTION TO NETWORKS

TUTORA

Carla Cardoso

Freelancer AI Manager

09 de abril 2025



1

O QUE ESPERAR DESTE MÓDULO



Quando terminarmos esta sessão vocês vão:

- Ter **conceitos teóricos** sobre Neural Networks (NN)
- Conseguir identificar as **diferentes componentes** de uma arquitetura *feedforward backpropagation*
- Compreender o **processo de aprendizagem** de um NN
- Saber o que são modelos de **Deep Learning** (DL) e como se diferenciam de modelos de NN
- Conhecer algumas **arquiteturas** de DL
- Ver alguns **exemplos** que aplicações atuais de DL
- Conhecer os **riscos e limitações** destas soluções

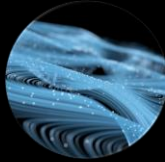
2

AGENDA

EDIT.



NEURAL NETWORKS



DEEP LEARNING



EXEMPLO APLICAÇÕES



RECOMENDAÇÕES

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

3

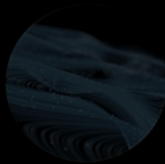
3

AGENDA

EDIT.



NEURAL NETWORKS



DEEP LEARNING



EXEMPLO APLICAÇÕES



RECOMENDAÇÕES

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

4

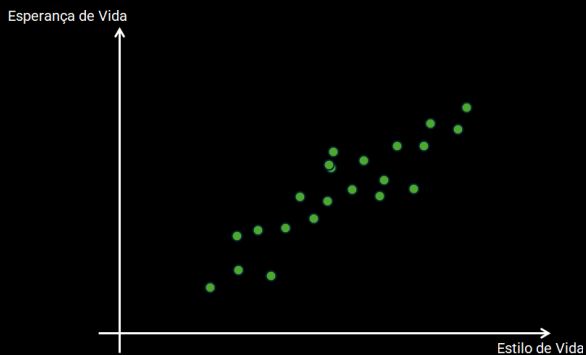
4

MOTIVAÇÃO

5

MOTIVAÇÃO: INTRODUÇÃO À REGRESSÃO LINEAR

Imaginem o cenário onde queremos prever a **Esperança de Vida** (*output*) com base na qualidade do **Estilo de Vida** (*input*) dos indivíduos. Recolhemos um conjunto de dados e obtivemos a seguinte distribuição:



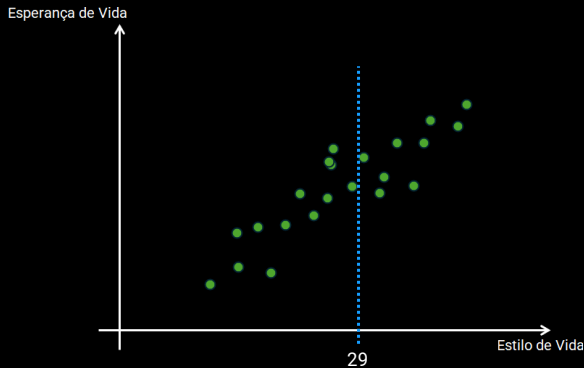
Estilo de Vida	Esperança de Vida
10	18
15	20
23	25
35	33
47	42
52	53
...	...

6

MOTIVAÇÃO: INTRODUÇÃO À REGRESSÃO LINEAR

EDIT.

Imaginem o cenário onde queremos prever a **Esperança de Vida** (*output*) com base na qualidade do **Estilo de Vida** (*input*) dos indivíduos. Recolhemos um conjunto de dados e obtivemos a seguinte distribuição:



Estilo de Vida	Esperança de Vida
29	?

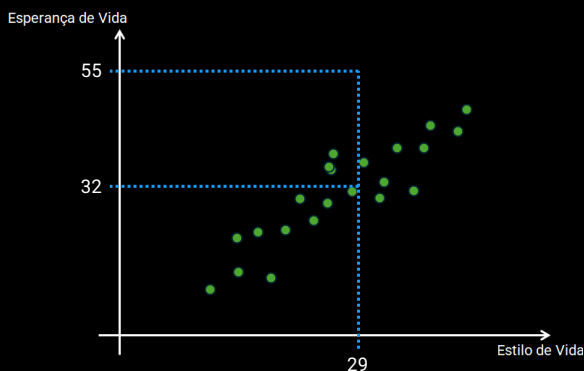
Dado que sabemos que o **Estilo de Vida** do novo indivíduo é **29**, como podemos saber qual a sua **Esperança de Vida**?

7

MOTIVAÇÃO: INTRODUÇÃO À REGRESSÃO LINEAR

EDIT.

Imaginem o cenário onde queremos prever a **Esperança de Vida** (*output*) com base na qualidade do **Estilo de Vida** (*input*) dos indivíduos. Recolhemos um conjunto de dados e obtivemos a seguinte distribuição:



Estilo de Vida	Esperança de Vida
29	?

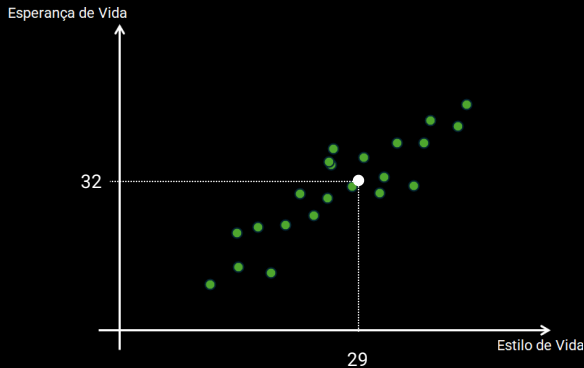
Qual dos valores vos parece mais provável para a **Esperança de Vida**?

8

MOTIVAÇÃO: INTRODUÇÃO À REGRESSÃO LINEAR

EDIT.

Imaginem o cenário onde queremos prever a **Esperança de Vida** (*output*) com base na qualidade do **Estilo de Vida** (*input*) dos indivíduos. Recolhemos um conjunto de dados e obtivemos a seguinte distribuição:



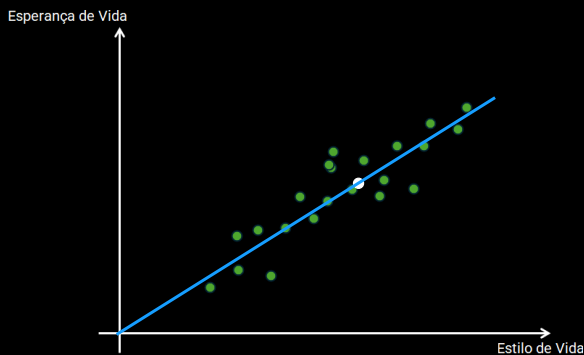
Estilo de Vida	Esperança de Vida
29	32

9

MOTIVAÇÃO: INTRODUÇÃO À REGRESSÃO LINEAR

EDIT.

Imaginem o cenário onde queremos prever a **Esperança de Vida** (*output*) com base na qualidade do **Estilo de Vida** (*input*) dos indivíduos. Recolhemos um conjunto de dados e obtivemos a seguinte distribuição:



O que fizemos mentalmente foi ajustar uma **solução linear** ao problema, que pode ser especificada da seguinte forma:

$$\text{Esperança de Vida} = a + b * \text{Estilo de Vida}$$

a : valor da **Esperança de Vida** quando Estilo de Vida = 0

b : incremento na **Esperança de Vida** quando Estilo de Vida sobe 1 unidade

A equação geral do modelo linear é dada por:

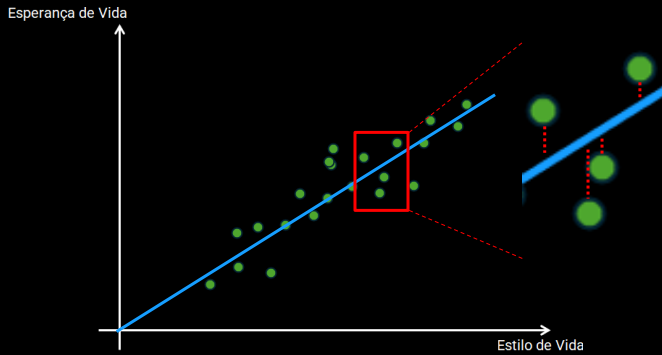
$$Y = a + b * X$$

10

MOTIVAÇÃO: INTRODUÇÃO À REGRESSÃO LINEAR



Ainda assim o modelo que criamos não é perfeito, existem **erros** associados a esta “simplificação” da realidade.



As distancias medidas pelas linhas vermelhas (---) representam o **erro** de estimação do nosso modelo em cada observação.

No caso de um modelo linear, como o do nosso exemplo, o erro do modelo pode ser dado por:

$$\text{Erro} = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

Onde:

m = # casos \Leftrightarrow dimensão da amostra

y_i = valor real de y para o indivíduo i

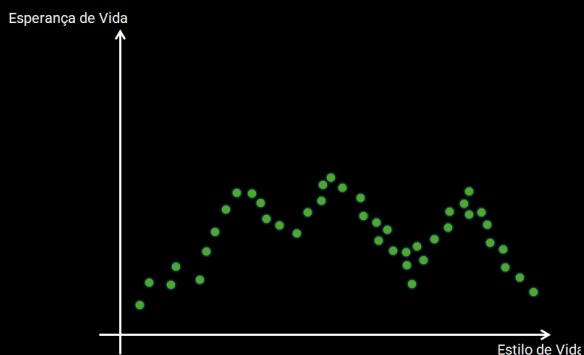
\hat{y}_i = valor estimado de y para o indivíduo i

11

MOTIVAÇÃO: MODELOS NÃO LINEARES



E se a distribuição dos nossos casos **não seguir uma distribuição Linear**?

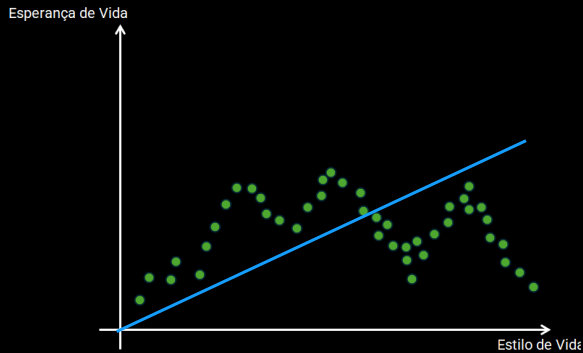


12

MOTIVAÇÃO: MODELOS NÃO LINEARES



E se a distribuição dos nossos casos **não seguir uma distribuição Linear**?



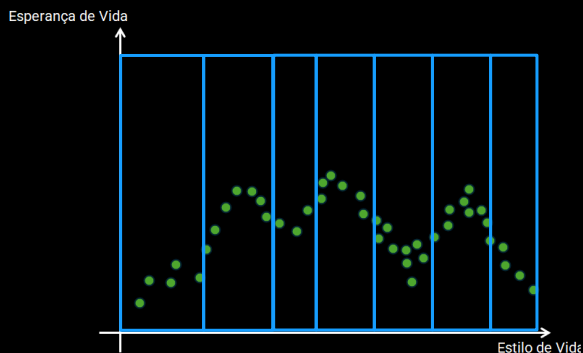
A aplicação de uma solução **linear** não parece ajustada neste caso.

13

MOTIVAÇÃO: MODELOS NÃO LINEARES



E se a distribuição dos nossos casos **não seguir uma distribuição Linear**?



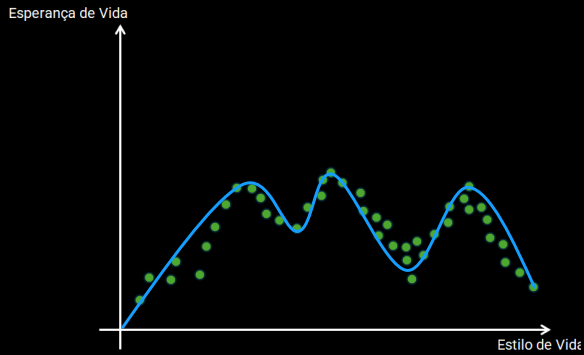
A aplicação de uma solução baseada em **árvores de decisão** também não parece ajustada neste caso.

14

MOTIVAÇÃO: **MODELOS NÃO LINEARES**

EDIT.

E se a distribuição dos nossos casos **não seguir uma distribuição Linear?**



Algo assim seria o ideal para este caso?

15

NEURAL NETWORKS

EDIT.

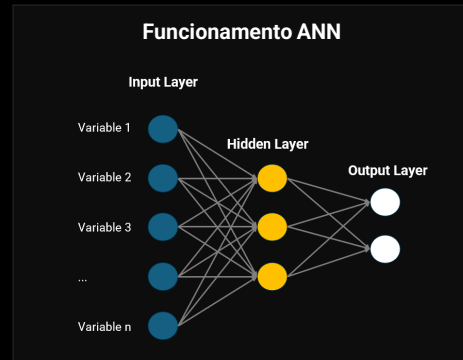
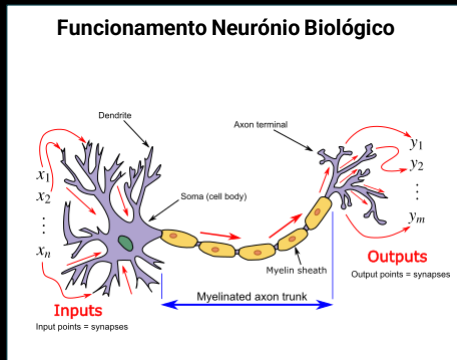
CONCEITOS BASE

16

CONCEITOS BASE: DEFINIÇÃO



As **Artificial Neural Networks** (ANN) ou, em Machine Learning, simplesmente **Neural Networks**, são modelos matemáticos inspirados na organização neuronal encontrada nas redes neurais biológicas em cérebros de animais.



DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

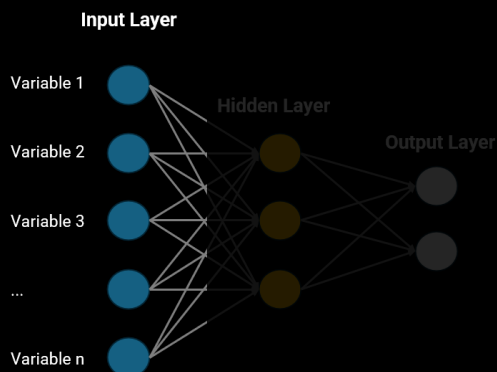
17

17

CONCEITOS BASE: COMPONENTES



Uma **Neural Network** é composta por nós (neurónios), conectados entre si em diferentes camadas.



A **Input Layer** (camada de entrada) é constituída por nós que correspondem às variáveis de *input*, um nó por cada variável.

Por **cada observação** lida do nosso *dataframe*, corresponde à atribuição de um **valor numérico** a cada um dos nós da *input layer*.

Cada nó da *input layer* está ligado a **TODOS** os nós da camada seguinte. Dizemos que a rede é do tipo **Fully Connected**.

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

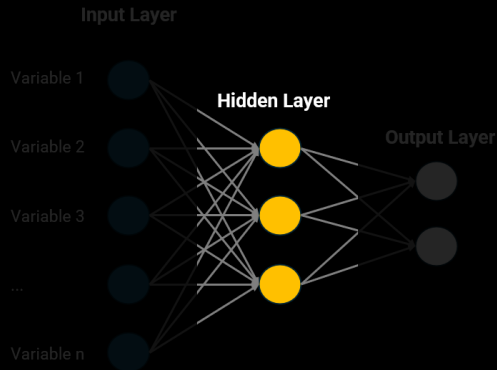
18

18

CONCEITOS BASE: COMPONENTES



Uma **Neural Network** é composta por nós conectados entre si em diferentes camadas.



Cada nó da **Hidden Layer** (camada escondida) recebe informação de todos os nós da camada anterior, que pode ser a **Input Layer** ou outra **Hidden Layer**.

Os valores de cada um dos nós da camada anterior é combinada através da **Transfer Function**, uma **transformação linear** equivalente à que vimos inicialmente:

Exemplo:

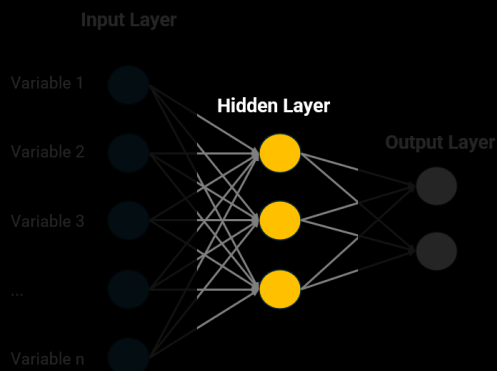
$$Y_1 = a_1 + b_{11} * Var_1 + b_{12} * Var_2 + ... + b_{1n} * Var_n$$

19

CONCEITOS BASE: COMPONENTES

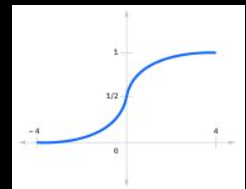


Uma **Neural Network** é composta por nós conectados entre si em diferentes camadas.



Sobre cada valor de Y é posteriormente aplicada uma transformação **não linear**, como por exemplo a função **logit**, a qual é designada por **Activation Function** (função de ativação):

$$Y_1' = \frac{1}{1 + e^{-Y_1}}$$



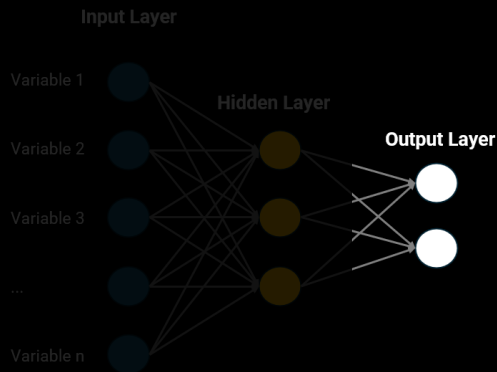
E é esse resultado não linear que é passado à **camada seguinte**.

20

CONCEITOS BASE: COMPONENTES



Uma **Neural Network** é composta por nós conectados entre si em diferentes camadas.



A **Output Layer** (camada de saída) é a última camada e, tal como as camadas anteriores, realiza uma combinação linear sobre os valores recebidos: Exemplo:

$$O_1 = a_1 + b_{11} * Y_1' + b_{12} * Y_2' + b_{13} * Y_3'$$

No caso do modelo ser de classificação (ex.: Fraude / não-Fraude) é ainda aplicada a **Activation Function** e ficamos com:

$$O_1' = \frac{1}{1 + e^{-O_1}}$$

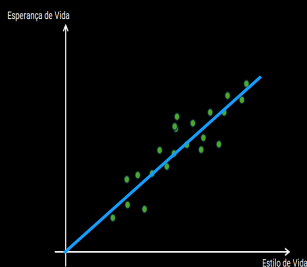
O_1' é o valor que o modelo estima para O_1 .

21

CONCEITOS BASE: EXEMPLO

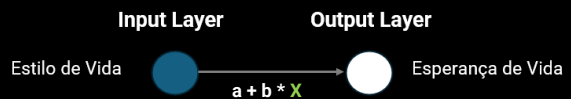


Voltando ao exemplo da previsão da **Esperança de Vida** podemos aplicar uma rede neuronal para simular uma função linear, da seguinte forma:



$$\text{Esperança de Vida} = a + b * \text{Estilo de Vida}$$

Não vai haver necessidade de camadas escondidas nem de funções de ativação, dado que a **solução é linear**.



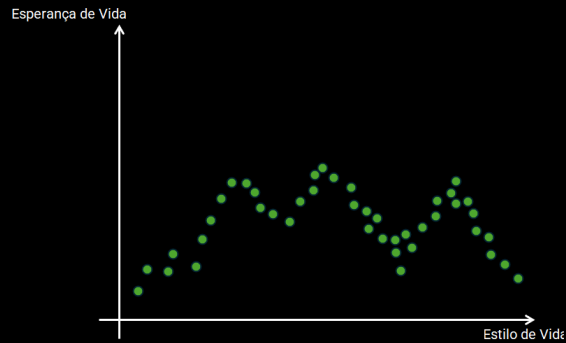
Uma consideração importante é que as redes neurais **apenas aceitam valores entre no intervalo** [-1 : 1] ou no intervalo [0 : 1] e por isso pode haver necessidade de **transformação de variáveis**. Esta transformação procura impedir que variáveis em variáveis com valores mais elevados (ex.: saldo médio da conta a ordem) se tornem mais relevantes do que variáveis de valores inferiores (ex.: dimensão agregado familiar)

22

CONCEITOS BASE: **EXEMPLO**

EDIT.

Mas quando imaginamos um cenário **não linear** é quando as redes acrescentam real valor.



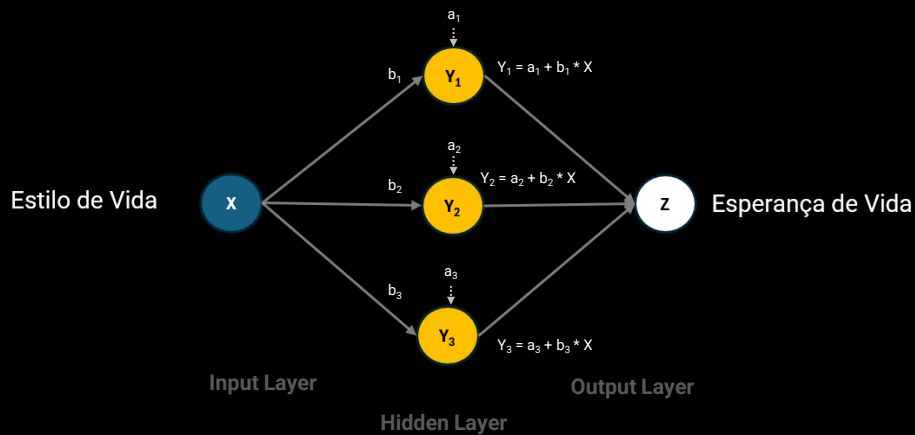
23

CONCEITOS BASE: **EXEMPLO**

EDIT.

Vamos assumir uma arquitetura com **1 Hidden Layer** e **3 Hidden Nodes**.

Passo 1:



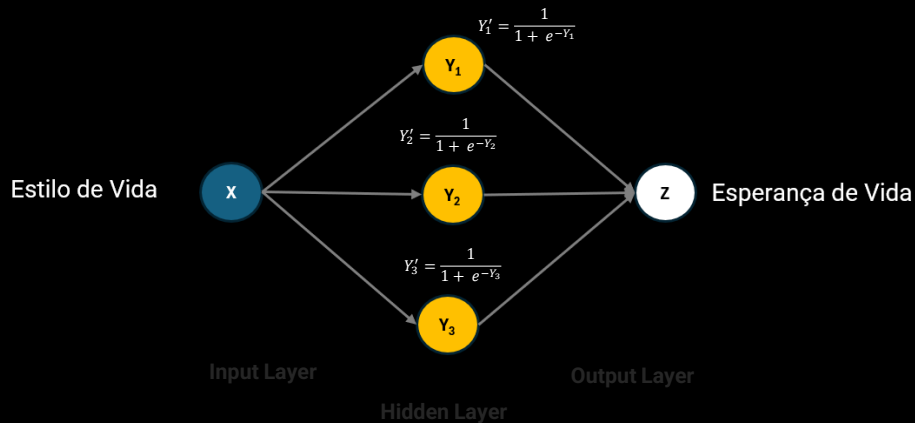
24

CONCEITOS BASE: EXEMPLO

EDIT.

Vamos assumir uma arquitetura com **1 Hidden Layer** e **3 Hidden Nodes**.

Passo 2:



DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

25

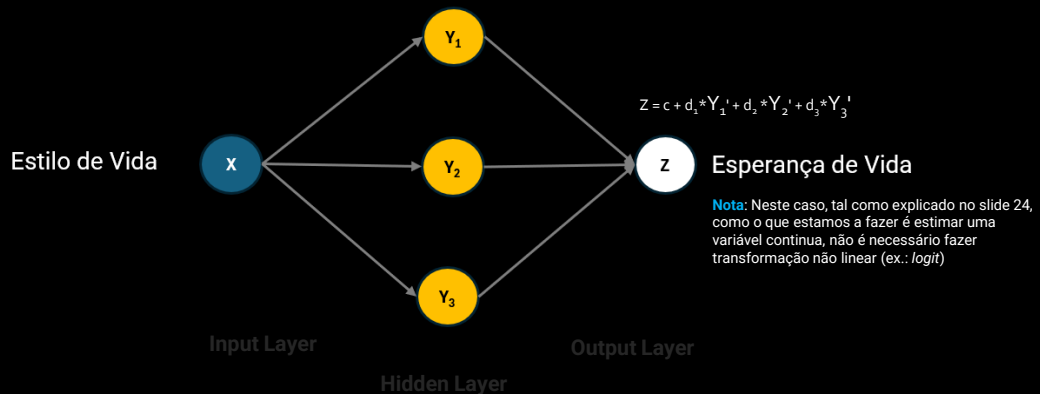
25

CONCEITOS BASE: EXEMPLO

EDIT.

Vamos assumir uma arquitetura com **1 Hidden Layer** e **3 Hidden Nodes**.

Passo 3:



DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

26

26

CONCEITOS BASE: EXEMPLO

EDIT.

Vamos simular um modelo que **explica** a **Esperança de Vida (Z)** com base na informação da **variável explicativa** o **Estilo de Vida (X)** **através** de uma **Neural Network (NN)**.

Tomemos os seguintes valores como pesos e bias:

- $a_1 = 0.1$
- $b_1 = 0.11$
- $a_2 = 0.2$
- $b_2 = 0.22$
- $a_3 = 0.3$
- $b_3 = 0.33$
- $c = 0.4$
- $d_1 = 5$
- $d_2 = 10$
- $d_3 = 20$

Estilo de Vida	Esperança de Vida
29	?

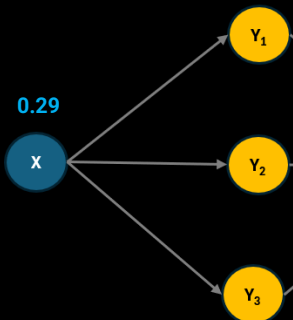
X normalizado = **0.29**

28

CONCEITOS BASE: EXEMPLO

EDIT.

- $a_1 = 0.1$
- $b_1 = 0.11$
- $a_2 = 0.2$
- $b_2 = 0.22$
- $a_3 = 0.3$
- $b_3 = 0.33$
- $c = 0.4$
- $d_1 = 5$
- $d_2 = 10$
- $d_3 = 20$

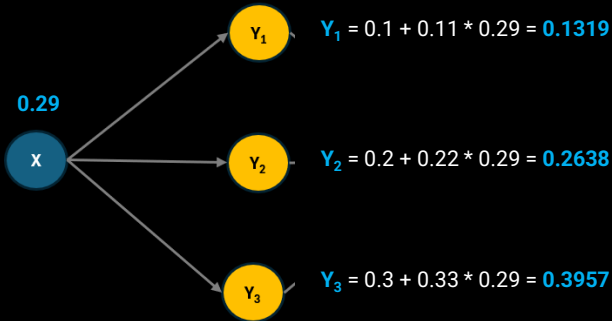


29

CONCEITOS BASE: EXEMPLO

EDIT.

- $a_1 = 0.1$
- $a_2 = 0.2$
- $a_3 = 0.3$
- $c = 0.4$
- $d_2 = 10$
- $b_1 = 0.11$
- $b_2 = 0.22$
- $b_3 = 0.33$
- $d_1 = 5$
- $d_3 = 20$

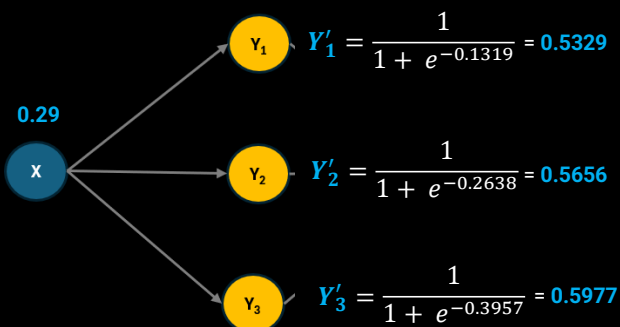


31

CONCEITOS BASE: EXEMPLO

EDIT.

- $a_1 = 0.1$
- $a_2 = 0.2$
- $a_3 = 0.3$
- $c = 0.4$
- $d_2 = 10$
- $b_1 = 0.11$
- $b_2 = 0.22$
- $b_3 = 0.33$
- $d_1 = 5$
- $d_3 = 20$



33

CONCEITOS BASE: **EXEMPLO**

EDIT.

- $a_1 = 0.1$
- $a_2 = 0.2$
- $a_3 = 0.3$
- $c = 0.4$
- $d_2 = 10$

- $b_1 = 0.11$
- $b_2 = 0.22$
- $b_3 = 0.33$
- $d_1 = 5$
- $d_3 = 20$

$Z = 0.4 + 5 * 0.5329 + 10 * 0.5656 + 20 * 0.5977$

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

34

34

CONCEITOS BASE: **EXEMPLO**

EDIT.

- $a_1 = 0.1$
- $a_2 = 0.2$
- $a_3 = 0.3$
- $c = 0.4$
- $d_2 = 10$

- $b_1 = 0.11$
- $b_2 = 0.22$
- $b_3 = 0.33$
- $d_1 = 5$
- $d_3 = 20$

20.66

Com base no nosso modelo, estimamos que alguém com um **Estilo de Vida** de **29 anos**, tem uma **Esperança de Vida ~21 ano de vida**.

Estilo de Vida	Esperança de Vida
29	20.67

Parece-vos bem? Como podemos melhorar o modelo?

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

36

36

NEURAL NETWORKS

EDIT.

Exemplo Prático

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

37

37

NEURAL NETWORKS

EDIT.

BORA LÁ POR A MÃO NA MASSA

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

38

38

EXERCICIO 1

EDIT.



Imaginem que têm de desenvolver um **Modelo Não Linear** para classificar se uma compra com cartão de credito é **Fraude**.

Para isso vocês têm acesso a ~500 mil transações e, para cada transação, têm a seguinte informação:

- Hora da transação
- Tipo de Transação (ex.: online)
- CAE do comerciante
- Valor da transação
- Fraude (1/0)

OBJETIVO: Desenhar a **arquitetura de uma rede neuronal** (incluindo formulação matemática de cada nó) para modelar o problema acima descrito tomando as seguintes restrições:

Hidden Layers = 1

Hidden Nodes = 2

Activation Function = **Logit**

EXERCICIO 1

EDIT.



O trabalho é para entregar até ao **final do dia de hoje**, e deve incluir a seguinte informação (1 slide por tópico):

- Descrição do vosso **dataframe** e caracterização do vosso **objetivo de negocio**
- **Tratamento de dados** que vocês vão fazer por forma a ter os dados prontos para ser usados numa rede neuronal (ex.: como tratar variáveis categóricas)
- Informação sobre o **data partition** que vão aplicar, e com que objetivo
- Desenho da **arquitetura da rede neuronal** com informação das formulas em cada nó

Posteriormente será elegido **um dos grupos para apresentar** o seu trabalho, e para os restantes grupos comentarem / complementarem.

EXERCICIO 1

EDIT.



Equipa 1

Carolina M.
Susana

Equipa 2

Tamara
Rui

Equipa 3

Nuno
José P.M.

Equipa 4

Gonçalo
João B.

Equipa 5

Alexandre
Stéfane

Equipa 6

Sara
Andreia

Equipa 7

Carolina L.
Ana

Equipa 8

Joana
Filipa
José F.

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

41

41

NEURAL NETWORKS

EDIT.

APRENDIZAGEM

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

42

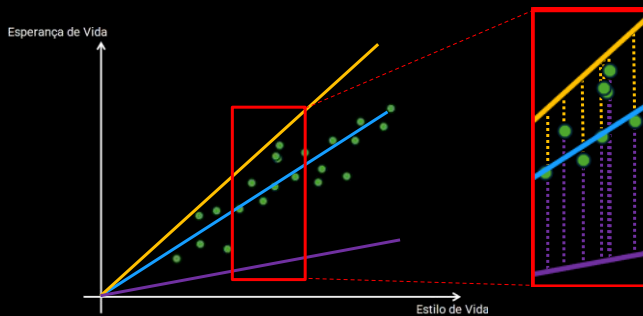
42

APRENDIZAGEM: GRADIENT DESCENT



Como já vimos, os modelos procuram **explicar os dados** observados através da criação de uma **função** que transforma os dados de **input** (variáveis explicativas) nos **outputs** (variáveis resultado).

Podemos ter diferentes modelos para o mesmo conjunto de dados, mas escolhemos o melhor com base na sua **capacidade de ajustamento aos dados**. Por exemplo, para os nossos dados sobre a Esperança de Vida, qual dos modelos iremos preferir?



Para responder a essa pergunta, temos de calcular o erro associado a cada um dos modelos.

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

44

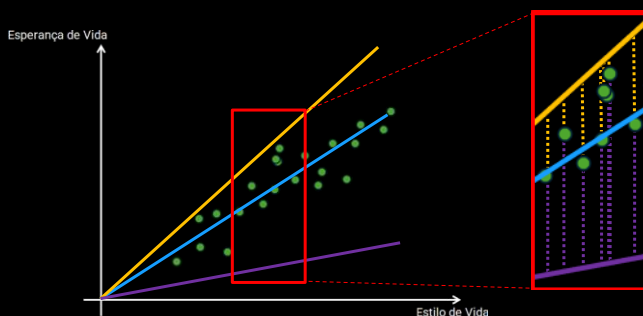
44

APRENDIZAGEM: GRADIENT DESCENT



Como já vimos, os modelos procuram **explicar os dados** observados através da criação de uma **função** que transforma os dados de **input** (variáveis explicativas) nos **outputs** (variáveis resultado).

Podemos ter diferentes modelos para o mesmo conjunto de dados, mas escolhemos o melhor com base na sua **capacidade de ajustamento aos dados**. Por exemplo, para os nossos dados sobre a Esperança de Vida, qual dos modelos iremos preferir?



Com base nos erros calculados para os 3 modelos, **modelo azul**, **modelo amarelo** e **modelos roxo** facilmente verificamos que o modelo que minimiza o "custo" (soma dos erros):

$$\text{Erro} = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

é o **modelo azul**.

DATA SCIENCE & BUSINESS ANALYTICS

Módulo 9 – Introduction to Networks – Sessão 1

45

45

APRENDIZAGEM: GRADIENT DESCENT



É definida a **Cost Function** (função de custo) que será **minimizada** recorrendo um **processo de otimização**. Essa função é calculada a cada iteração do modelo.

Por exemplo, a nossa função de custo pode ser dado por:

$$e = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

Dado que \hat{y}_i corresponde ao valor que o modelo estima para o individuo i , então e depende dos pesos que o modelo está a usar.

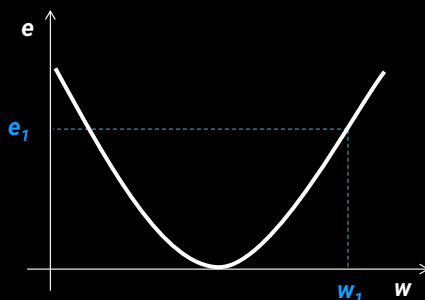
Vamos chamar w ao vetor de pesos que o modelo está a usar. Então podemos dizer que e é uma função de w .

APRENDIZAGEM: GRADIENT DESCENT



Nós não conhecemos a distribuição do erro, pois caso a soubéssemos conseguimos encontrar diretamente o vetor w que a minimizava e com isso encontrávamos o melhor modelo que explica os dados.

Mas vamos imaginar que essa função tem a seguinte forma:



Fazemos uma iteração do modelo e chegamos ao vetor de pesos w_1 que tem um erro associado igual a e_1 .

Qual seria o próximo passo?

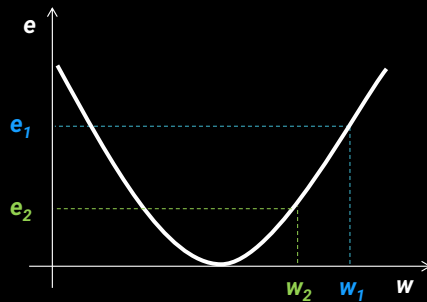
1. "Aumentar" w_1
2. "Diminuir" w_1
3. Nada, atingimos o ponto ótimo

APRENDIZAGEM: GRADIENT DESCENT



Nós não conhecemos a distribuição do erro, pois caso a soubéssemos conseguimos encontrar diretamente o vetor w que a minimizava e com isso encontrávamos o melhor modelo que explica os dados.

Mas vamos imaginar que essa função tem a seguinte forma:



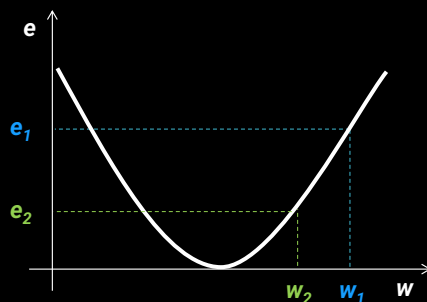
Ao fazer w_2 "menor" que w_1 vamos obter e_2 , que é menor e_1 .

APRENDIZAGEM: GRADIENT DESCENT



Nós não conhecemos a distribuição do erro, pois caso a soubéssemos conseguimos encontrar diretamente o vetor w que a minimizava e com isso encontrávamos o melhor modelo que explica os dados.

Mas vamos imaginar que essa função tem a seguinte forma:



Ao fazer w_2 "menor" que w_1 vamos obter e_2 , que é menor e_1 .

A conclusão, que neste caso foi intuitiva, é formulada matematicamente com recurso a **derivadas**.

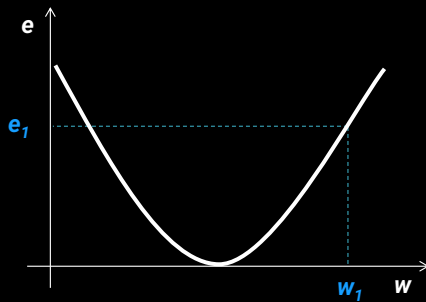
Neste caso, no ponto w_1 a função de erro é **crescente**, ou seja, tem a sua derivada positiva, e por isso devemos diminuir w_1 para minimizar o erro. Se a função fosse **decrecente** (como é o caso na zona à esquerda da curva), teríamos de incrementar o valor de w_1 .

APRENDIZAGEM: LEARNING RATE



Tipicamente a cada iteração do algoritmo, que corresponde à passagem de uma observação dos dados de *input* pela rede, é feito um ajuste à **Learning Rate** (rácio de aprendizagem).

Já concluímos que w_1 tem de “diminuir”, mas quanto?



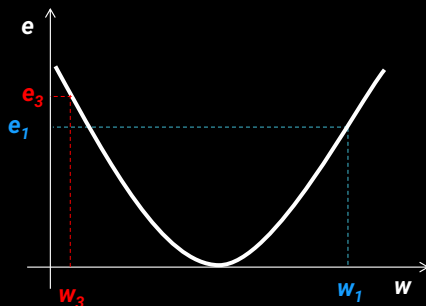
51

APRENDIZAGEM: LEARNING RATE



Tipicamente a cada iteração do algoritmo, que corresponde à passagem de uma observação dos dados de *input* pela rede, é feito um ajuste à **Learning Rate** (rácio de aprendizagem).

Já concluímos que w_1 tem de “diminuir”, mas quanto?



Se “diminuirmos” os pesos de forma muito “rápida” podemos ir parar a w_3 , onde o erro é superior comparado ao erro associado a w_1 .

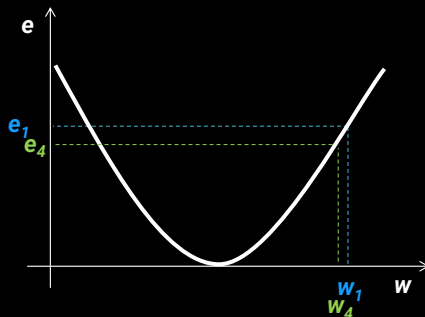
52

APRENDIZAGEM: LEARNING RATE



Tipicamente a cada iteração do algoritmo, que corresponde à passagem de uma observação dos dados de *input* pela rede, é feito um ajuste à **Learning Rate** (rácio de aprendizagem).

Já concluímos que w_i tem de “diminuir”, mas quanto?



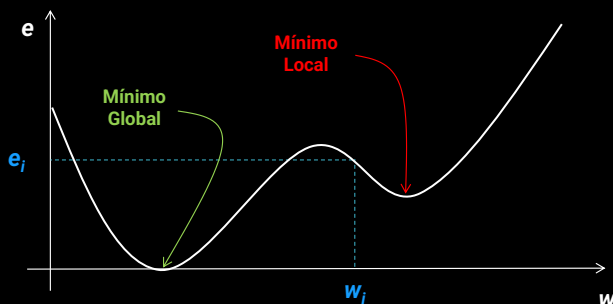
Mas se formos “diminuindo” os pesos de forma muito “lenta” podemos ficar apenas em w_3 , onde o impacto no erro não é significativo.

Tipicamente a **Learning Rate** é ajustada ao momento da aprendizagem, sendo superior em iterações iniciais, e vai diminuindo em iterações mais avançadas por forma a **convergir** para um ponto onde o erro é mínimo.

APRENDIZAGEM: MÍNIMOS LOCAIS



Um dos problemas que podemos encontrar com o uso do **Gradient Decent** e a **Learning Rate** tem a ver com os mínimos locais. Imaginemos a seguinte função de erro e o valor atual de w é w_i :



Dado o que aprendemos, a derivada é negativa em w_i e por isso devemos “aumentar” w_i .

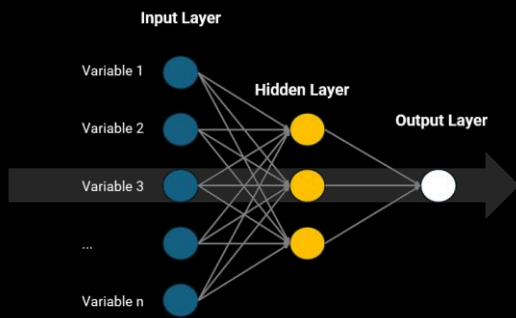
A modelo irá convergir para um **Mínimo Local**, não para o **Mínimo Global**.

A melhor solução para este problema é **testar várias iterações** do modelo, com **diferentes pesos iniciais**.

APRENDIZAGEM: BACKPROPAGATION



Agora que já sabemos que w tem de “aumentar” ou “diminuir” como fazemos isso?



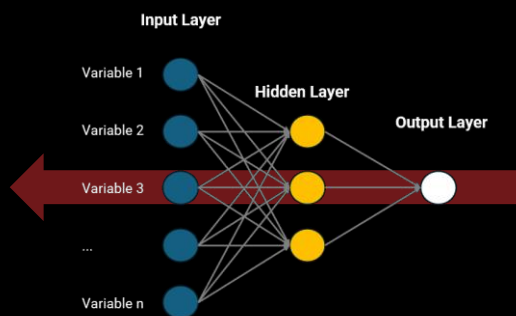
Quando estamos a calcular as estimativas para os valores da camada de output, a direção da fluxo de informação é **Forward** (para a frente), daí o nome deste tipo de rede, **Feedforward**.

57

APRENDIZAGEM: BACKPROPAGATION



Agora que já sabemos que w tem de “aumentar” ou “diminuir” como fazemos isso?



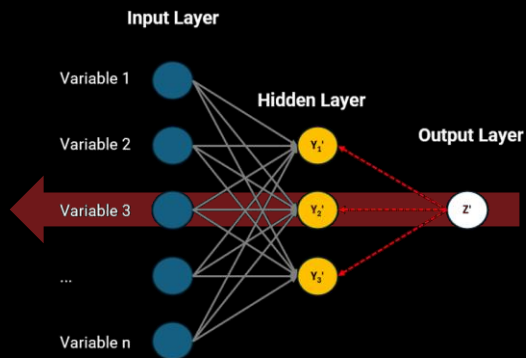
Quando estamos a “propagar” o erro para ajustar os pesos de cada nós, o sentido é inverso, para trás, daí a denominação **Backpropagation**.

58

APRENDIZAGEM: BACKPROPAGATION



Para propagar o ajuste de w pela rede vamos novamente recorrer ao uso das derivadas.



Vamos começar por **propagar o erro desde o nó da camada de output, Z**, para os nós da camada imediatamente anterior.

Cada peso, será ajustada na mesma medida que a sua **"contribuição" para o erro**, dado pela derivada da função de erro em base ao nó:

$$w_{li} := w_{li} - \alpha \frac{\partial e}{\partial l_i}$$

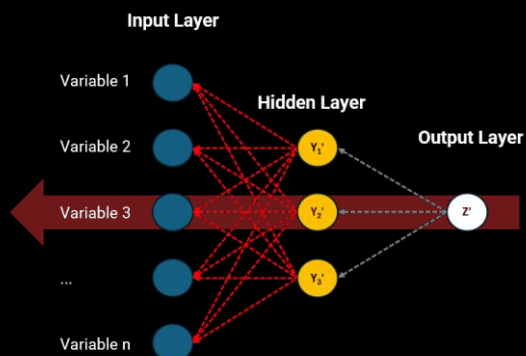
Onde α é a **Learning Rate**.

59

APRENDIZAGEM: BACKPROPAGATION



Para propagar o ajuste de w pela rede vamos novamente recorrer ao uso das derivadas.



Fazemos o **mesmo processo para cada nó** da camada anterior, para propagar o erro pelos diferentes nós nas diferentes camadas.

Após termos o **novo vetor w** , passamos para a **nova iteração** do modelo, obtemos um novo resultado \hat{y} e com ele novos erros para propagar.

O **modelo pára** quando a **condição de paragem** é atingida, pode ser # iterações, erro mínimo, etc.

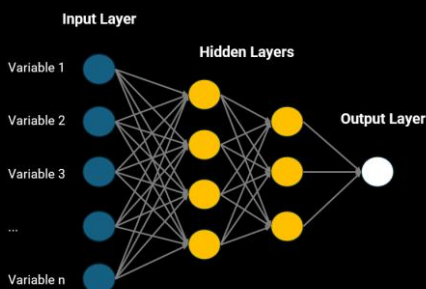
60

MULTILAYER

61

MULTILAYER

Até agora vimos arquiteturas com **apenas 1 camada escondida**, mas nem sempre 1 camada é suficiente. As redes **multicamada têm maior flexibilidade** para se ajustar aos dados com comportamentos não lineares dado incorporar mais níveis de abstração:



O funcionamento das redes **Multilayer** é equivalente ao funcionamento das redes **Single Layer**, sendo as últimas um caso particular das primeiras.

A escolha do **# de camadas escondidas** e o **# de nós** em cada camada vai depender da **complexidade** do problema que temos para resolver, ou seja, dos dados. É preferível **fazer alguns testes** com diferentes arquiteturas para escolher.

Nem sempre mais complexidade na rede leva a melhores resultados.

62

PASSO A PASSO

63

PASSO A PASSO

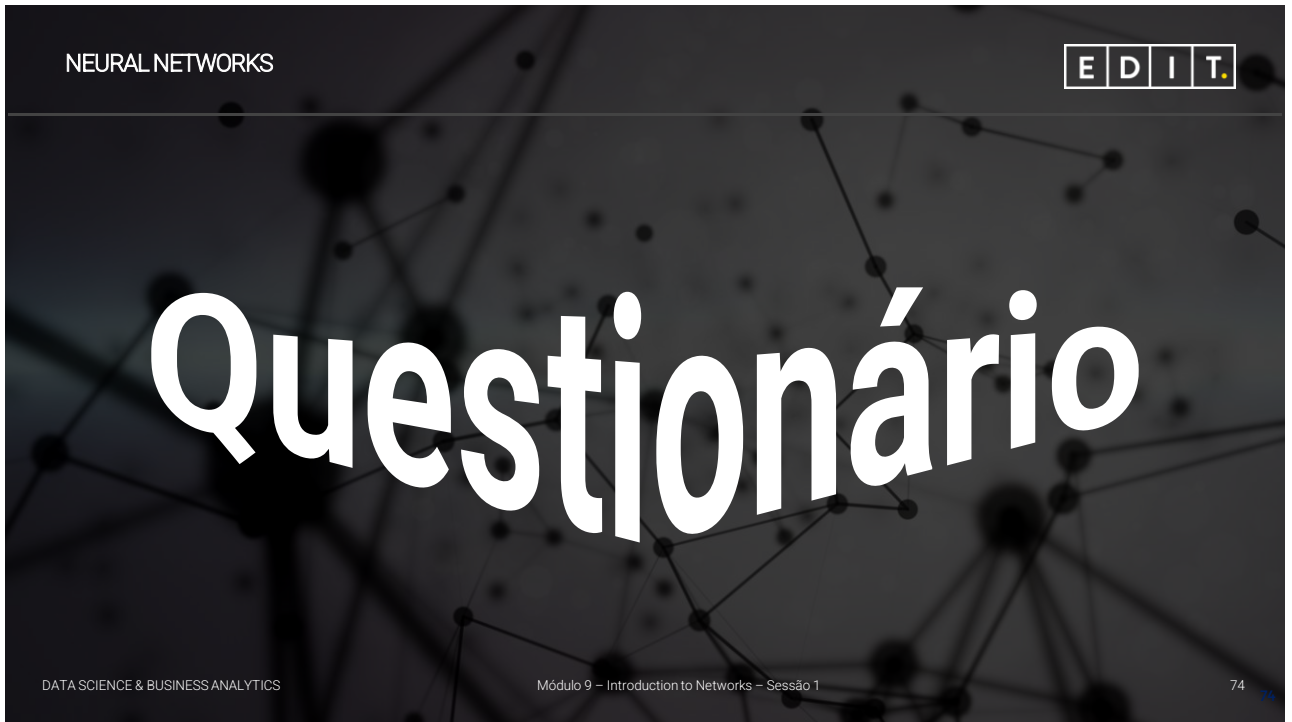
Imaginem que chegaram à conclusão que a melhor maneira de resolver o problema que vocês têm em mãos, é através de uso de algoritmos de Redes Neurais.

E agora? ☺

Existem várias coisas que têm de fazer e de definir, apesar de que existem *defaults* nas ferramentas de desenvolvimento que são um excelente ponto de partida.

- | | |
|--------------------------------|---|
| 1 Normalização Variáveis Input | 5 Activation Function |
| 2 # Camadas / # Nós Escondidos | 6 Cost Function |
| 3 Inicialização dos pesos | 7 Learning Rate |
| 4 Transfer Function | 8 Critério de paragem: # iterações, tempo, custo... |

73

The slide features a dark background with a complex network diagram of interconnected nodes and lines. The title 'Questionário' is prominently displayed in the center in a large, white, sans-serif font. In the top left corner, the text 'NEURAL NETWORKS' is written in a smaller, white, sans-serif font. In the top right corner, the word 'EDIT' is enclosed in a white rectangular box with a small yellow dot at the end. At the bottom left, the text 'DATA SCIENCE & BUSINESS ANALYTICS' is written in a small, white, sans-serif font. At the bottom center, the text 'Módulo 9 - Introduction to Networks - Sessão 1' is written in a small, white, sans-serif font. At the bottom right, the number '74' is written in a small, white, sans-serif font, followed by a small blue logo.