

Tarefa 3 - 29/06/2021 - Data de entrega: 09/07/2021

- I. Defina uma rede neural artificial. Pesquise e descreva duas aplicações reais para as quais uma rede neural artificial se apresenta como um modelo adequado de solução.
 - Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Procuram reproduzir características humanas como aprendizado, associação, generalização e abstração. As redes neurais permitem que máquinas consigam entender e aprender padrões, adaptar-se a mudanças e imitar processos de pensamento humano na busca por respostas e soluções.
 - **Funciona da seguinte maneira:** As redes neurais usam inteligência artificial para processar e entender informações. Uma rede neural pode ter dezenas de milhões de neurônios artificiais, que ficam dispostos em camadas. Essas camadas são compostas por vários nós, como nossos neurônios, que se ligam em diversas ramificações. Cada nó contém várias informações. Cada vez que uma relação lógica é estabelecida entre esses nós, surge uma nova conexão que os liga e assim por diante. As informações do mundo exterior chegam na primeira camada, conhecida como camada de entrada. Essas informações vão passando para uma ou mais camadas conectadas entre si. Cada camada procura transformar a informação recebida em algo que a unidade final, chamada de unidade de saída, possa compreender e utilizar.

Aplicações:

1. Sistemas de Apoio na Tomada de Decisão para mercados financeiros: grupos de investimento conhecidos utilizam redes neurais para analisar pelo menos uma parte do mercado financeiro e fazerem suas decisões. Um exemplo pode ser visto no trabalho de Pedronette (2004), com uma aplicação que permite a configuração dos parâmetros da Rede Neural de forma gráfica e simplificada e retorna os dados processados em forma de tabelas e gráficos.

2. Sistema de reconhecimento de imagens: detecção de padrões e reconhecimento de objetos em imagens. Como trabalho acadêmico, Castro (2003) desenvolveu um sistema de controle de um veículo autônomo terrestre que identifica as faixas laterais de uma pista detectando as bordas, por redes neurais artificiais (RNAs). A informação é enviada ao controle do veículo que, através de regras *fuzzy* e de RNAs, toma decisões sobre a trajetória do veículo, assim permanecendo dentro da pista. Outro exemplo de aplicação: um banco americano chamado Mellon Bank instalou um sistema de detecção de fraudes de cartão de crédito implementado com técnicas de redes neurais e os prejuízos evitados pelo novo sistema conseguiram cobrir os gastos de instalação em seis meses (Marangoni, 2010).

2. Considerando uma rede neural **supervisionada do tipo Perceptron** responda:

- a. Explique o modelo do neurônio artificial que compõe esta rede (explique cada componente do neurônio).
- Frank Rosenblatt (1958, 1962) concebeu um algoritmo de aprendizado para um tipo de rede de camada única chamada *Perceptron*. O *Perceptron* é um modelo matemático que recebe várias entradas, x_1, x_2, \dots, x_n e produz uma única saída binária. O modelo do neurônio de *Perceptron* conforme figura 1 consiste em:

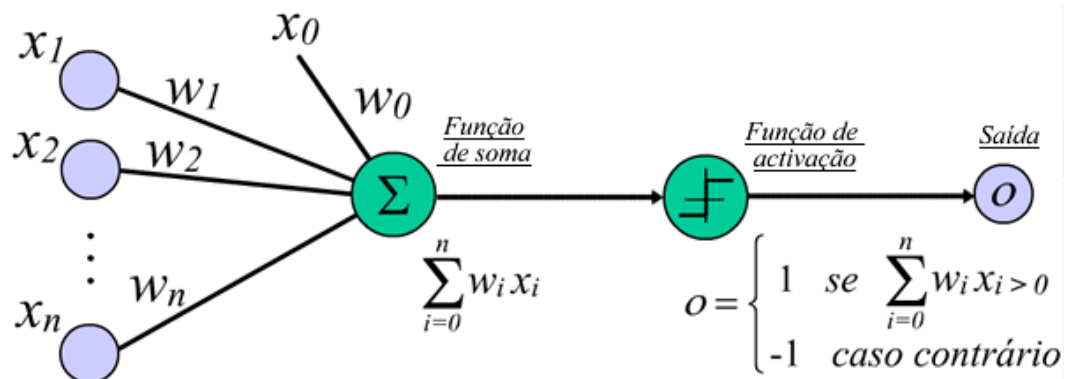


Figura 1: composição de um neurônio Perceptron

- ✓ **Entradas:** Dados provenientes do ambiente ou da ativação do neurônio com valores reais no intervalo $\{0,1\}$ ou $\{-1,1\}$.
- ✓ **Pesos:** Com efeito de inibição ou de excitação sobre a ativação dos sinais de entradas. Os pesos que representam a importância de cada entrada associada ao respectivo peso. Descrevem as forças da conexão e representam o conhecimento da rede. Cada peso é um valor numérico real e o treinamento do neurônio é justamente a busca do melhor valor de cada peso.

- ✓ **Nível de Ativação:** Considera o conjunto total das entradas e dos seus pesos associados. A Ativação que soma todos produtos obtidos entre a entrada e o peso correspondente, ou seja, é determinado pela soma das entradas ponderadas pelos pesos.
 - ✓ **Função limiar:** Consiste na regulação da saída da rede. Calcula o estado final, a saída do neurônio, verificando se o nível de ativação está acima ou abaixo de um valor limiar.
 - ✓ **Saída:** normalmente possui conexões com os outros neurônios da rede.
 - ✓ **Bias ou viés:** é um parâmetro adicional na Rede Neural que é usado para ajustar a saída junto da soma ponderada das entradas para o **neurônio**. Ou seja, **Bias** é uma constante que ajuda o modelo de uma maneira que ele possa se adaptar melhor aos dados fornecidos para gerar saída buscada.
 - **Características importantes do Perceptron:**
 - ✓ Os **valores de entrada** e os **níveis de ativação** do *perceptron* são sempre -1 ou 1.
 - ✓ Os **pesos** têm valores reais
 - ✓ O **nível de ativação** do *perceptron* é dado pela soma dos valores ponderados das entradas, $\sum x_i w_i$.
 - ✓ Os *perceptrons* usam uma **função limiar simples** em que uma ativação acima de um limiar resulta em um valor de saída de 1, ou, no caso contrário, de -1.
 - **Funcionamento básico do Perceptron**
 - ✓ No modelo do neurônio artificial *Perceptron* cada neurônio (ou nó) em uma rede neural recebe uma série de **entradas** ($x_1, x_2, x_2, \dots, x_n$) e um **peso** (w_1, w_2, \dots, w_n) associado, onde por ativação cada entrada é multiplicada por um peso e o resultado da multiplicação é somado ($\sum x_i w_i$) e então passado para uma **função limiar** que, por sua vez, resulta na saída 1 ou -1 do neurônio.
- b. Cite 3 funções que podem ser empregadas como função de ativação do neurônio.

A função pode ser: degrau, degrau bipolar, rampa, sigmoide, linear, semi-linear, tangente hiperbólica, entre outras. A função de ativação pode ter diferentes representações. Os três tipos básicos de função de ativação são: **limiar, linear e sigmoide**. A escolha do tipo varia de acordo com o objetivo do projeto ou problema a ser resolvido. As funções de ativação são escolhidas em função da necessidade do problema em que a rede esteja trabalhando. Abaixo seguem as características de algumas funções:

- ✓ **Função linear e degrau** possuem a saída do neurônio com o valor 1, se o nível de atividade interna total do neurônio for não-negativo, caso contrário, a saída do neurônio assumirá 0.

- ✓ **Função degrau bipolar** a saída do neurônio assumirá o valor 1 se o nível de atividade interna total do neurônio for não-negativo; caso contrário, a saída do neurônio assumirá -1.
- ✓ **Função rampa:** a saída do neurônio pode assumir valores positivos e negativos no domínio de -1 a 1, e no intervalo $\{-a, a\}$, a saída assume o valor da função $g(u)=u$
- ✓ **Função sigmoide logística:** β é o parâmetro que define a inclinação (ganho) da função sigmoide. Nesse tipo de função, a saída do neurônio assumirá valores reais entre 0 e 1.
- ✓ **Função sigmoide tangente hiperbólica:** a saída do neurônio pode assumir valores reais negativos e positivos no domínio de -1 a 1.

c. Explique o algoritmo de aprendizado por correção do erro.

- A *perceptron* multicamadas (MLP) é uma rede neural semelhante à *perceptron*, mas com mais de uma camada de neurônios em alimentação direta. Tal tipo de rede é composto por camadas de neurônios ligadas entre si por sinapses com pesos. A nesse tipo de rede é geralmente feito através do algoritmo de retro propagação do erro.
- O algoritmo de retro propagação (*Backpropagation*) de erro é um algoritmo utilizado no treinamento de redes neurais multicamadas, e consiste em dois passos: o processamento direto e o processamento reverso. No processamento direto, uma entrada é aplicada à rede neural e seu efeito é propagado pela rede, camada a camada. Durante o processamento direto, os pesos da rede permanecem fixos. Já no processamento reverso, um sinal de erro calculado na saída da rede é propagado no sentido reverso, camada a camada, e ao final deste processo os pesos são ajustados de acordo com uma regra de correção de erro.
- Basicamente, a rede aprende um conjunto pré-definido de pares de exemplos de entrada/saída em ciclos de propagação/adaptação. Depois que um padrão de entrada foi aplicado como um estímulo aos elementos da primeira camada da rede, ele é propagado por cada uma das outras camadas até que a saída seja gerada. Este padrão de saída é então comparado com a saída desejada e um sinal de erro é calculado para cada elemento de saída. O sinal de erro é então retro propagado da camada de saída para cada elemento da camada intermediária anterior que contribui diretamente para a formação da saída. Cada elemento da camada intermediária recebe apenas uma porção do sinal de erro total, proporcional apenas à contribuição relativa de cada elemento na formação da saída original. Este processo se repete, camada por camada, até que cada elemento da rede receba um sinal de erro que descreva sua contribuição relativa para o erro total. Baseado no sinal de erro recebido, os pesos das conexões são então atualizados para cada elemento de modo a fazer a rede convergir para um estado que permita a codificação de todos os padrões do conjunto de treinamento.
- O algoritmo de retro propagação segue os seguintes passos:
 1. Inicialização. Inicialize os pesos da rede aleatoriamente ou segundo algum método.
 2. Processamento direto. Apresente um padrão à rede. Compute as

- ativações de todos os neurônios da rede e então calcule o erro.
3. Passo reverso. Calcule os novos pesos para cada neurônio da rede, no sentido retroativo (isto é, da saída para a entrada), camada a camada.
 4. Teste de parada. Teste o critério de parada adotado. Se satisfeito, termine o algoritmo;
 5. Se não volte ao passo 2.
- O algoritmo *Backpropagation* é hoje em dia a técnica de aprendizado supervisionado mais utilizada para redes neurais unidirecionais multicamadas com neurônios estáticos.
- d. Explique o parâmetro taxa de aprendizado empregado. Como este parâmetro afeta a convergência da rede neural?
- A taxa de aprendizado empregado corresponde diretamente à velocidade do aprendizado. Uma taxa de aprendizado baixa torna o aprendizado da rede muito lento ao passo que uma taxa de aprendizado alta provoca oscilações no treinamento e impede a convergência do processo de aprendizado. O parâmetro da taxa auxilia no aprendizado da rede neural artificial através da adaptação dos pesos em resposta às entradas. Tem grande influência no processo de treinamento da rede neural. Tendo assim como função principal ajustar o conjunto de pesos da rede a fim de minimizar o erro médio do conjunto de treinamento. A taxa de aprendizagem exprime o quão rápido o processo de treinamento da rede estará sendo conduzido rumo à sua convergência (estabilização). A escolha da taxa deve ser realizada com cautela para evitar instabilidades no processo de treinamento, sendo que normalmente se adotam valores pertencentes ao intervalo compreendido entre 0 e 1.
 - A taxa de aprendizagem representa, em termos práticos, o tamanho do passo que a rede toma em busca do ajuste ideal dos pesos sinápticos. Passos muito pequenos podem fazer com que o treinamento fique lento. Passos muito longos podem conduzir o ajuste a direções incorretas em relação à direção ideal do treinamento, o que significa que o próximo passo irá requerer correção do erro anterior. Além desse problema, a convergência pode passar por longos períodos de estagnação quando o gradiente cai em mínimos locais. Há indicações de procedimentos para se evitarem esses problemas. Uma delas é a de que a taxa de aprendizagem deve ser menor nas últimas camadas. Isso é indicado pelo fato de que as últimas camadas possuem gradientes locais maiores do que as camadas anteriores (Haykin, 2008). Outra indicação é a de que neurônios com muitas entradas tenham taxa de aprendizagem menor (Haykin, 2008). LeCun (LeCun, 1993) sugere que, para um determinado neurônio, a taxa de aprendizagem seja inversamente proporcional à raiz quadrada das conexões sinápticas feitas com aquele neurônio. Também há indicações de que a taxa de aprendizagem deva ser ajustada ao longo do treinamento. Jacobs (Jacobs, 1988) sugere quatro heurísticas para a taxa de aprendizagem.
 1. Cada parâmetro ajustável da função de custo da rede deve ter seu parâmetro individual da taxa de aprendizagem. Isso significa que uma taxa

de aprendizagem eficiente para determinado peso nem sempre é eficiente para outro peso. Cada região da superfície do erro pode se adaptar melhor a determinado valor de taxa de aprendizagem.

2. A taxa de aprendizagem deve poder variar de um ciclo para outro. Isso é especialmente importante porque a superfície do erro tem comportamento diferente em 49 ciclos diferentes do treinamento.
3. A taxa de aprendizagem deve ser aumentada sempre que a derivada da função custo em relação a um peso sináptico tiver o mesmo sinal algébrico para ciclos consecutivos do treinamento. A principal motivação para essa heurística, reside no fato de que, em porções mais planas da superfície do erro ao longo da dimensão de um peso particular, o sinal algébrico se mantenha igual em vários ciclos do treinamento. O aumento na taxa de aprendizagem pode reduzir o número de ciclos para atravessar essa porção plana.
4. A taxa de aprendizagem deve ser reduzida sempre quando a derivada da função custo em relação a um peso sináptico alternar de sinal por vários ciclos. Essa situação ocorre em regiões da superfície do erro com grande quantidade de picos e vales. Reduzindo-se o valor da taxa de aprendizagem, evita-se que o vetor gradiente mude de sinal por muitas vezes. Isso acelera a convergência do treinamento.

3. Diferencie os paradigmas de aprendizado supervisionado e não supervisionado.

- **Aprendizado supervisionado:** são os modelos que realizam as tarefas preditivas, ou seja, visam encontrar uma função ou modelo a partir dos dados de treinamento que possa utilizada para prever um rótulo (classe) que caracterize um novo exemplo, de acordo com os atributos de entrada. Para tanto, o conjunto de treinamento deve possuir atributos de entrada e saída. O modelo aprende a executar uma tarefa a partir de exemplos rotulados, ou seja, a partir das respostas corretas, que de alguma forma devem ser passadas para o modelo existindo assim, portanto, um treinamento conforme a entrada e o conhecimento da saída desejada. Dentre as tarefas de previsão temos a classificação e a regressão.
- **Aprendizado não supervisionado:** são os modelos que realizam as tarefas descritivas, cuja meta é explorar ou descrever um conjunto de dados. Os conjuntos de dados não apresentam o atributo de saída e não existe nenhum agente externo indicando a resposta desejada para os padrões de entrada. Não existe, portanto, o processo de treinamento. O aprendizado é feito por meio dos próprios dados. Este modelo aprende a executar uma tarefa a partir de dados não-rotulados (sem um resultado conhecido), apenas com base em suas características e padrões semelhantes, ou seja, o modelo deduz estruturas a partir de uma amostra do problema. É aplicada nos casos em que, apesar de existir o objetivo da tarefa desejada, os resultados finais não são conhecidos e, portanto, não se tem os rótulos para passar ao modelo. Dentre as tarefas descritivas temos a clusterização que consiste no agrupamento de dados, onde o modelo busca por grupos de elementos similares e a técnica de associação: permite o descobrimento de regras e correlações, identificando conjuntos de itens

que frequentemente ocorrem juntos.

4. Explique o paradigma de aprendizado por reforço.

- Aprendizado por reforço é um tipo de aprendizado supervisionado que visa reforçar ou recompensar uma ação considerada positiva e punir uma ação considerada negativa. Portanto, Aprendizado por reforço é um método de programação de agentes através do oferecimento de recompensas e punições, sem a necessidade de especificar como uma tarefa deve ser realizada. Assim, a máquina tenta aprender qual é a melhor ação a ser tomada, dependendo das circunstâncias na qual essa ação será executada. Espera-se que o agente consiga associar as ações que geram maior recompensa para cada situação que o ambiente apresenta, e passe a evitar as ações que geram punição ou recompensa menor.
- **Funciona da seguinte maneira:** O agente realiza uma ação num dado ambiente, alterando seu estado inicial, o que gera uma recompensa ao agente. De forma cíclica, o agente avalia esta recompensa (que pode ser positiva ou negativa) e age novamente no ambiente, gerando o aprendizado. O processo se repete até que a máquina seja capaz de escolher a melhor ação a se tomar para cada um dos cenários possíveis a serem observados no futuro.
- É aplicada quando se conhece as regras, mas não se sabe a melhor sequência de ações que devem ser executadas, elas são iterativamente aprendidas, como num jogo de xadrez ou num videogame, onde o fundamental são as ações tomadas pelo jogador (ou pela máquina).
- Segundo **Teixeira**, o aprendizado por reforço é um campo do aprendizado de máquinas com foco na criação de agentes capazes de tomar decisões acertadas em um ambiente sem que se tenha qualquer conhecimento prévio sobre o tal ambiente. Para que ocorra aprendizado é preciso que o agente após perceber o ambiente tome ações e observe a recompensa imediata proveniente da ação e a mudança no ambiente decorrente da ação tomada. O objetivo do agente é maximizar o total de recompensas imediatas recebidas durante a interação com o ambiente. Desta forma, o agente não está sempre interessado em ações que geram a maior recompensa imediata, mas sim em ações que o leve ao maior acúmulo de recompensa a longo prazo.

5. Apresente a porta lógica OU como um problema de classificação que pode ser solucionado por um neurônio artificial, indicando:

Sendo : 0 = falso e 1 = verdade

a. As entradas do problema

- **Entradas: X1 e X2**

Entrada X1	Entrada X2
0	0
0	0
1	1
1	1

b. Número de entradas e a dimensão de cada uma delas

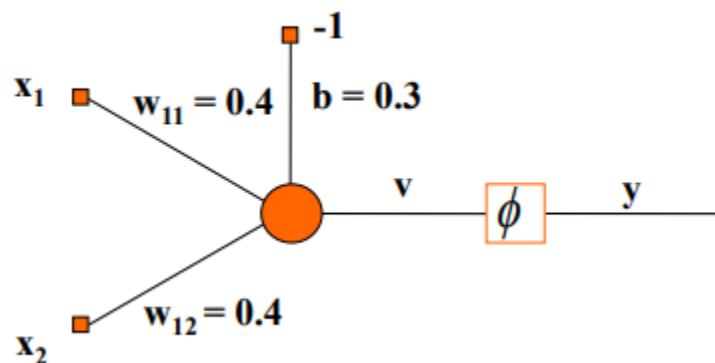
- **Tamanho da entrada** = 2
- **Numero de entradas** = 4 (Indica o nº de padrões diferentes de aprendizagem)
- **Peso W_{x1}** = 0.4
- **Peso W_{x2}** = 0.4

Obs: Os pesos W_{x1} e W_{x2} podem ser diferentes, mas optou-se para exemplificar a porta or/ou com valores de peso iguais a 0,4

c. Tamanho (dimensão) do vetor de pesos

Tamanho da Dimensões = 2 (Indica o nº de pesos e sera sempre igual ao tamanho da entrada do problema)

d. Representação gráfica da entrada



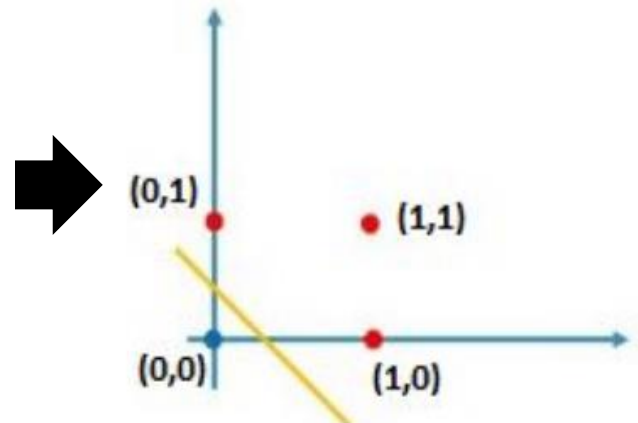
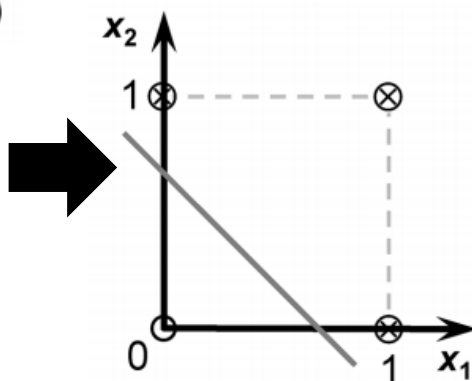
Para as saídas, a função de ativação considerará verdade (1) somente aqueles cujo resultado seja maior que zero ($Wx_1 X_1 + Wx_2 X_2 - b > 0$) e aquele que for menor que zero a saída será 0 (falsa). Assim temos:

Entrada X1	Peso Wx1	Entrada X2	Peso Wx2	Bias (b = -1.0,3)	Ativação		Saídas	
					Wx1 X1 + Wx2 X2 + (b)			
						Função Limiar		
0	0,4	0	0,4	-0,3	0,4.0 + 0,4.0 – 0,3 = - 0,3	< 0	0	Falso
0		0			0.4.0 + 0,4.1 – 0.3 = 0,1	> 0	1	Verdade
1		1			0.4.1 + 0,4.0 – 0.3 = 0,1	> 0	1	Verdade
1		1			0.4.1 + 0,4.1 – 0.3 = 0,5	> 0	1	Verdade

Graficamente então temos:

$$y \leftarrow (x_1) \text{ OR } (x_2)$$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



e. Porque o problema é linear

Porque é possível separar as duas classes: verdadeira (1) e falsa (0), através de uma única reta conforme mostrado no gráficos anteriores.

f. Como os pesos são atualizados no processo de aprendizagem

Após processar uma entrada, é feita a verificação da saída, de acordo com a

saída desejada (Lei do aprendizado). Se existe erro (se não for à saída desejada), então o modelo modifica os pesos buscando reduzir o erro. Assim se a saída produzida pelo *Perceptron* não coincidir com a saída desejada, os pesos sinápticos e limiares da rede serão então incrementados proporcionalmente aos valores de seus sinais de entrada; caso contrário, a saída produzida pela rede é igual ao valor desejado, os pesos sinápticos e limiar permanecerão então inalterados. Este processo é repetido, sequencialmente para todas as amostras de treinamento, até que a saída produzida pelo *Perceptron* seja similar à saída desejada de cada amostra. Ou seja, após apresentar as entradas e multiplica-las por seus respectivos pesos é gerado então um **valor** de ativação, que através da **função** de ativação é averiguado se o valor gerado representa a saída buscada, se o valor gerado não corresponder ao valor desejado é então gerado um erro e o peso precisará ser atualizado conforme a regra abaixo:

$$\begin{aligned}\text{peso novo} &= \text{peso antigo} + (\text{taxa de aprendizado} \cdot \text{erro} \cdot \text{entrada}) \\ w_{\text{novo}} &= w_{\text{antigo}} + c(d - \text{sinal}(\sum x_i w_i)) x_i \\ \text{onde } c &\text{ é a taxa de aprendizado e } d \text{ é a saída desejada.}\end{aligned}$$

Webgrafia: acesso em 04/07/2021

Questao 1:

<https://sites.icmc.usp.br/andre/research/neural/>

CARVALHO, A. P. L. F. Redes Neurais Artificiais. São Carlos, Brasil. Disponível em: <https://sites.icmc.usp.br/andre/research/neural/>. Acesso: 04 jul. 2021.

CASTRO, A. P. A.. Detecção de Bordas com Redes Neurais BAM e ART. Dissertação (Mestrado) — INPE, São José dos Campos, Brasil, 2003. Disponível em: http://mtc-m16c.sid.inpe.br/col/lac.inpe.br/lucio/2002/11.13.00.50/doc/worcap2_apaula-versaofinal.PDF. Acesso: 05 jul. 2021.

MARANGONI, P. H. Redes Neurais Artificiais para Previsão de Séries Temporais no Mercado Acionário. UFSC, Florianópolis, Brasil, 2010. Disponível em: <http://tcc.bu.ufsc.br/Economia292751>. Acesso: 05 jul. 2021.

PEDRONETTE, D. C. G. Apoio à tomada de decisões no Mercado de Financeiro utilizando Redes Neurais. 2004. USP, São Paulo, Brasil, 2004. 12º Simpósio Internacional de Iniciação Científica da USP. Disponível em: <https://www.ic.unicamp.br/~dcarlos/pubs/sup-material/SIICUsp2004/relatorio-iniciacao-cientifica.pdf>. Acesso: 05 jul. 2021.

Questao 2 – a:

ZAMBIASI, S. P. Introdução as Redes Neurais Artificiais. Florianópolis, Brasil, 2002. Disponível em: https://www.gsigma.ufsc.br/~popov/aulas/rna/uteis/RNA_material_apoio.pdf. Acesso: 04 jul. 2021.

FREITAS, A. T. Percepção. Lisboa, Portugal. Disponível em: <http://web.tecnico.ulisboa.pt/ana.freitas/bioinformatics.ath.cx/bioinformatics.ath.cx/index79ff.html?id=112>. Acesso: 04 jul. 2021.

Questao 2 – b:

CINTRA, R. S. C. Introdução à Neurocomputação. INPE, São José dos Campos, Brasil, 2018. Disponível em: http://www.inpe.br/elac2018/arquivos/ELAC2018_MC3_apostila.pdf. Acesso: 04 jul. 2021.

Questao 2 – c:

LUGER, G. F. Inteligência Artificial. Pearson Universidades - 6ª Edição. 2013. Págs. 386-390.

ZAMBIASI, S. P. Aprendizado de Redes Neurais Artificiais. Disponível em: <https://www.gsigma.ufsc.br/~popov/aulas/rna/aprendizado/>. Acesso: 04 jul. 2021.

Questao 2 – d:

SILVA, I. N. *Et al.* Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas: Fundamentos Teóricos e Aspectos Práticos. Artliber - 2ª edição. 2016.

Questao 3:

BIANCHI, A. L. As classificações dos algoritmos de Machine Learning. Viceri. 2020. Disponível em: <https://www.viceri.com.br/insights/as-classificacoes-dos-algoritmos-de->

machine-learning. Acesso: 04 jul. 2021.

Questao 4:

HONDA, H. *Et al.* Os Três Tipos de Aprendizado de Máquina. LAMFO. 2017. Disponível em: <https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/>. Acesso: 04 jul. 2021.

JÚNIOR, E. P. F. D. Aprendizado por Reforço Sobre o Problema de Revisitação de Páginas Web. Disponível em: http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0912826_2012_cap_3.pdf. Acesso: 04 jul. 2021.

TEIXEIRA, L. A. Métodos de Regressão para Aprendizado por Reforço. UFJF, Juíz de Fora, Brasil. 2016. Disponível em: <https://www.ufjf.br/getcomp/files/2013/03/MÉTODOS-DE-REGRESSÃO-PARA-APRENDIZADO-POR-REFORÇO.pdf>. Acesso: 04 jul. 2021.

