



Redes Neurais Artificiais aplicadas a classificação e reconhecimento de padrões

Artificial Neural Networks applied to classification and pattern recognition

Gabriel Roberto Lamoglia*, Hugo Valadares Siqueira†,

RESUMO

O presente trabalho trata sobre redes neurais artificiais (RNAs) e algumas de suas arquiteturas. O objetivo é demonstrar uso das RNAs em diferentes bases de dados, criando uma arquitetura para detecção de câncer de mama simples; efetuando um comparativo entre duas arquiteturas para uma mesma base de dados e propondo um programa para ilustrar a aplicabilidade de uma estrutura ao mundo real. As estruturas abordadas nas seções seguintes foram escolhidas de acordo com as bases utilizadas: “*Breast Cancer (Diagnostic)*” com 569 registros e 32 atributos previsores, “*Iris*” com 150 registros e 4 atributos previsores e “*MNIST*” com 70.000 registros e atributos variados (64, 128). As arquiteturas aplicadas foram MLP, ELM e CNN. Estas foram construídas com códigos em Python 3 e auxílio da IDE Spyder 4.3.2. Os resultados mostraram-se eficientes, tendo as RNAs treinadas bons resultados. A partir dos resultados apresentados, conclui-se que quando bem estruturadas e com os parâmetros adequados, as redes podem ter abundantes aplicações.

Palavras-chave: Rede-Neural-Artificial, Python, MLP, CNN.

ABSTRACT

The present study is about artificial neural networks (RNAs) and some of their architectures. The purpose of this is to demonstrate the use of RNAs in different databases; creating a simple breast cancer detection architecture, making a comparison between two architectures for the same database and proposing a program to illustrate the applicability of a structure to the real world. The structures discussed in the following sections were chosen according to the bases used: “*Breast Cancer (Diagnostic)*” with 569 records and 32 predictive attributes, “*Iris*” with 150 records and 4 predictive attributes and “*MNIST*” with 70,000 records and varied attributes (64, 128). The architectures applied to the described databases were MLP, ELM and CNN. They were built with code in Python 3 and the help of IDE Spyder 4.3.2. The results were efficient, with the trained ANNs presenting good results. From the results presented, it is concluded that when well-structured and with the appropriate parameters, networks can be used in plenty applications.

Keywords: Artificial-Neural-Network, Python, MLP, CNN.

1. INTRODUÇÃO

O início das redes neurais artificiais (RNA's), surge da modelagem matemática dos neurônios biológicos. A referência mais antiga encontrada quando se aborda o tema é de 1943 e fora proposta por Warren McCulloch e Walter Pitts no *Bulletin of Mathematical Biophysics*, um artigo denominado “*A Logical Calculus of the Ideas Immanent in Nervous Activity*” (KOVACS, 2006, p.13-32). McCulloch propôs um modelo de neurônio com uma lógica simples. McCulloch e Pitts foram desbravadores ao proporem um modelo computacional para o neurônio biológico, porém nos dias atuais esta proposta é de pouco uso. Já ao final da década de 1950, Frank Rosenblatt propôs o ‘Perceptron’, uma proposta similar, entretanto com diferenças básicas na elaboração do ajuste dos pesos (ROSENBLATT, 1958, p.386-408).

Após anos de aperfeiçoamento as redes neurais passaram a ser implementadas com facilidade devido ao uso das linguagens de programação. A rede neural típica é constituída por um conjunto de neurônios artificiais interligados que influenciam uns aos outros, essa rede armazena o conhecimento que adquire por meio de exemplos apresentados e assim pode agir sob outros exemplos ainda não conhecidos.

Após anos de aperfeiçoamento as redes neurais passaram a ser implementadas com facilidade devido ao uso das linguagens de programação e interfaces de desenvolvimento que facilitam a visualização de códigos e resoluções de problemas por usuários. A rede neural típica é constituída por um conjunto de neurônios artificiais

* Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Ponta Grossa, Paraná, Brasil; lamoglia@alunos.utfpr.edu.br

† Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa; hugosiqueira@utfpr.edu.br



interligados que influenciam uns aos outros, essa rede armazena o conhecimento que adquire por meio de exemplos apresentados e assim pode agir sob outros exemplos ainda não conhecidos.

Trazendo à tona problemas enfrentados regularmente no mundo real, exemplos podem ser citados quanto ao uso de RNA's descritas no documento, como o para o auxílio na identificação do câncer (GEBRIM; QUADROS, 2006). O comparativo ocorre entre duas arquiteturas, MLP e ELM. Além disso, utiliza-se tais metodologias para identificação de números escritos a mão (LECUN, 1997), Mas afinal, como as RNA's podem ser cabíveis ao mundo real?

Nas seções seguintes, discutem-se os aspectos construtivos para elaboração de uma rede neural artificial. Este estudo também vai analisar as RNA's quanto a resultados obtidos em suas rodagens, comparar estruturas e exemplificar como esses modelos podem ser utilizados em problemas de mundo real.

2. CLASSIFICAÇÃO E APRENDIZAGEM DAS REDES NEURAIS ARTIFICIAIS

As redes neurais no geral, são orientadas por camadas de modo que os neurônios artificiais são dispostos em conjuntos distintos e ordenados, o conjunto dessas sequencias forma uma camada. O aprendizado ocorre por meio da atualização constante dos pesos sinápticos por meio de processos iterativos ativando condições de neurônios específicos durante esta etapa. Para Simon Haykin (HAYKIN, 1999) existem quatro tipos de regras quanto ao aprendizado: Correção e erro, Teoria Hebbiana (HEBB, 1949), aprendizagem de Boltzman (ACKLEY, HINTON, SEJNOWSKIM, 1985, p.147-169) e aprendizagem do tipo competitiva.

A implementação de técnicas por meio de bibliotecas auxilia na criação das redes. As bibliotecas utilizadas têm base em Python 3 (Auxílio de IDE Spyder Anaconda) e foram escolhidas as seguintes:

- Keras: biblioteca de código aberto que tem seu código baseado no tensorflow e é responsável por permitir o trabalho com redes convolucionais e redes profundas devido à grande quantidade de ferramentas como otimizadores, funções de perda, etc;
- Pandas: biblioteca utilizada para a leitura, manipulação, alinhamento e manipulação de dados ausentes, tal como limpeza e filtro de amostras. O Pandas permite exportar e importar vários formatos de arquivo, o que torna a biblioteca útil para quando trabalhamos com *dataframes* e arquivos “.csv”.
- TKINTER: oferece ferramentas de janela que podem fazer com que o usuário crie interfaces com comandos simples criados em Python.

Técnicas utilizadas na criação das redes foram:

- *Dropout*: Responsável por eliminar neurônios de uma camada com determinada percentagem. Durante o aprendizado de determinado registro, a rede elimina aleatoriamente n% dos neurônios da camada oculta, evitando *overfitting*;
- *K-Fold Cross-Validation*: as bases de dados trabalham são divididas em três subgrupos: treinamento, validação e teste. O método *K-Fold Cross-validation*, ou validação cruzada com k-grupos, embaralha a base de dados de treinamento de forma aleatória e divide os dados em k-grupos, de modo que um destes é utilizado para validação cruzada com o intuito de maximizar a capacidade de generalização da rede, enquanto os k-1 restantes são utilizados para treinamento. O procedimento é repetido alternando o grupo de validação entre os subgrupos formados. O resultado final é uma média das percentagens encontradas em cada teste pela quantia de grupos utilizados ‘k’ (média simples).
- *Tunning*: Está técnica apesar de ser poderosa, é muito pesada e não é utilizada em muitos casos, pois testa n-parâmetros passados para a rede realizando combinações.



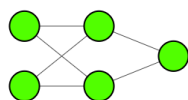
De maneira padrão para todas as redes criadas ao decorrer deste documento, foram adotadas as bibliotecas supramencionadas, criando arquiteturas através de codificação e submetendo os dados das bases as arquiteturas, criando desta maneira as inteligências artificiais.

3. ARQUITETURA DAS REDES

3.1. Perceptron de múltiplas camadas (MLP)

O treinamento envolve ajuste de pesos com técnicas como *tuning*. As redes funcionam como uma via de mão-dupla, ida e volta. A rede então executa processo iterativo até que responda a determinado requisito de parada, ajustando os pesos. Tal requisito pode ser o de não haverem mais possíveis diminuições de erro, número de iterações ou até mesmo resultados aproximados. A Figura 1 apresenta uma forma de distribuição de neurônios de uma MLP genérica.

Figura 1: MLP na arquitetura mais simples.



FONTE 1: Autoria própria.

3.2. Máquina de Aprendizado Extremo (ELM)

As ELM são arquiteturas de rede neural com conexões aleatórias entre as camadas (tradicionalmente, utiliza-se apenas de uma camada oculta) (GUANG-BINHUANG, QIN-YUZHU; CHEE-KHEONG, 2006). São similares as tradicionais MLP's, mas não são treinadas pela mecânica convencional de *backpropagation*. Neste caso, apenas a camada de saída é ajustada por meio da solução por fórmula fechada, utilizando a pseudoinversão de Moore-Penrose.

3.3. Redes Neurais Convolucionais (CNN)

As Redes Neurais Convolucionais (CNN) São arquiteturas profundas que podem ser usadas para classificar imagens, agrupar imagens por padrões de características e realizar reconhecimento de objetos dentro de imagens ou cenas. São frequentemente colocadas em carros autônomos, detecção de pedestres e objetos. Toda imagem é formada por uma matriz de pixels ($L \times C$) e cada pixel contém a combinação de cores RGB disposto em certo padrão, gerando a cor desejada. Dentro de uma CNN cada atributo há ser comparado em uma imagem. Por exemplo, imagine a existência de uma imagem de $32 \times 32 = 1024$ pixels, sendo uma imagem RGB (3 cores). Uma CNN pode não utilizar todas as entradas (pixels), mas apenas as características importantes. Ademais a rede escolhe por si só quais são as características mais importantes, transformando os dados na camada de entrada da rede. Seguindo as etapas: Operador de convolução; Pooling; Flattening; Conexão profunda.

4. RESULTADOS

Nesta seção, apresenta-se os resultados computacionais da aplicação de modelos neurais citados nas seções anteriores em problemas conhecidos da literatura. Sendo esses problemas a criação de redes para as bases *Breast Cancer*, *Iris* e *NMIST*, resolvidos respectivamente com MLP, MLP e ELM e CNN.

4.1 Breast Cancer submetido a MLP

Uma MLP com modelo sequencial totalmente conectado. Parâmetros escolhidos de acordo com a necessidade e definidos através da técnica de *tunning*. Os parâmetros como o número de neurônios na camada oculta, seguem conceitos antigos como “número de neurônios na entrada + número de neurônios na saída/2” (HAYKIN, S., 1999). Enquanto que os parâmetros definidos por *tunning* foram feitos conforme apresentados na Tabela 1. A validação cruzada aumentou o desempenho da rede quanto a precisão em 5%. O gráfico da evolução da função de perda que tal como o gráfico da precisão, são mostrados nas Figuras 3a e 3b.

Tabela 1: Parâmetros pré-selecionados; em verde selecionados com *tunning*.

Parâmetro	Parâmetro 1	Parâmetro 2
Batch Size	10	30
Épocas	100	120
Otimizador	Adam	Adamax
Ativação f(x)	Relu	Tangh
Inicial. Pesos	Random	Normal
Unidades	16	12
Função perda	Binária (CE)	Hinge
Dropout	10%	20%

FONTE 2: Autoria própria (Software: Microsoft Excel).

Figura 3a: Gráfico BC, precisão do treinamento

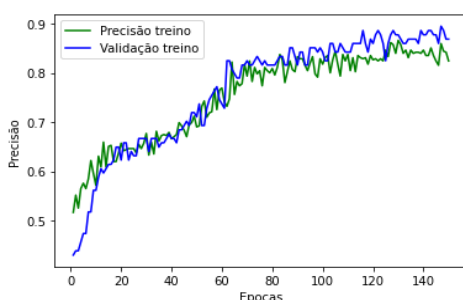
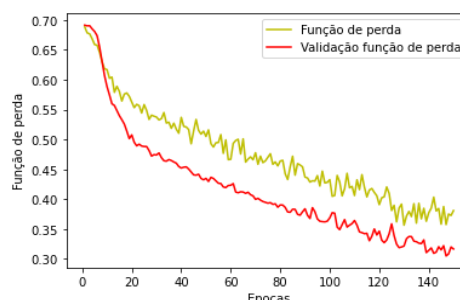


Figura 2b: Gráfico BC, função de perda



FONTE 3: Autoria própria (Software: Spyder 4.3.2 - Matplot)

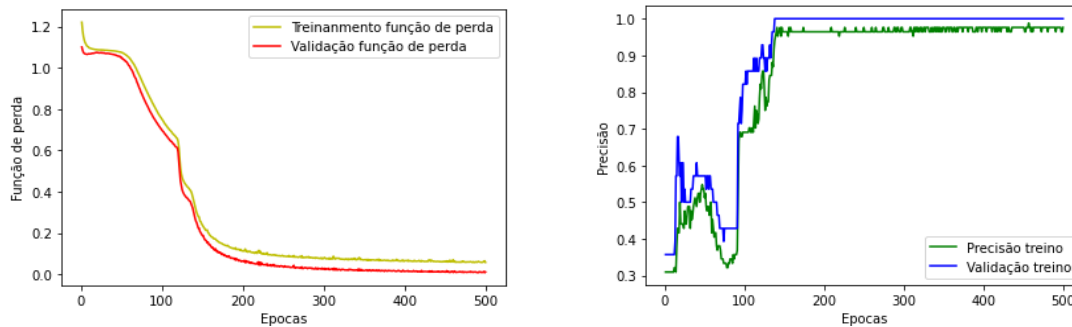
O otimizador da MLP foi o Adamax. A rede utilizou 30 neurônios em sua camada de entrada, correspondente aos seus 30 atributos dos registros previsores e 1 neurônio na camada de saída, contém duas camadas ocultas que trabalham com 12 e 8 neurônios, respectivamente. A função Relu é utilizada nos neurônios das camadas ocultas pois está descarta a possibilidade de os pesos obterem valores menores ou iguais a zero. A função sigmoide é usada na saída, pois sempre retorna um valor entre 0 e 1, cabendo apenas a escolha de um valor liquido para determinar se o tumor deve ser considerado maligno.

4.2 Iris submetido a MLP e ELM

A estrutura da MLP é muito semelhante ao trabalhado no tópico “*Breast Cancer*”. Entretanto, esta base requer utilização de técnicas diferenciadas quanto ao tratamento dos dados. (HAYKIN, S., 1999). O diferencial desta abordagem, ocorre justamente na multiclassificação que passa a requerer uma função de ativação não-linear dos neurônios da última camada, definida como sendo a função softmax, que retorna probabilidades para cada registro se enquadrar em uma classe. As Figuras 5 e 6 apresentam gráficos da melhor execução.

Figura 3a: Gráfico Iris, função de perda MLP.

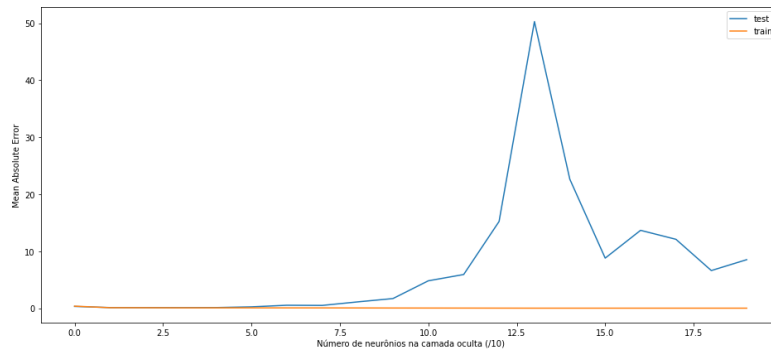
Figura 4b: Gráfico Iris, precisão de treinamento MLP



FONTE 4: Autoria própria (Software: Spyder 4.3.2 - Matplot)

A ELM variou número de neurônios na camada oculta, tendo em vista que seu treinamento é efetuado apenas uma vez, não apresentando processos iterativos. A rede apresentou bons resultados, exceto ao trabalhar com uma quantidade muito grande de neurônios. A Figura 6 mostra a evolução da precisão em termos do MAE para diversas quantidades de neurônios.

Figura 4: Execução da ELM comparando o erro (Desvio surgindo com mais de 75 neurônios)

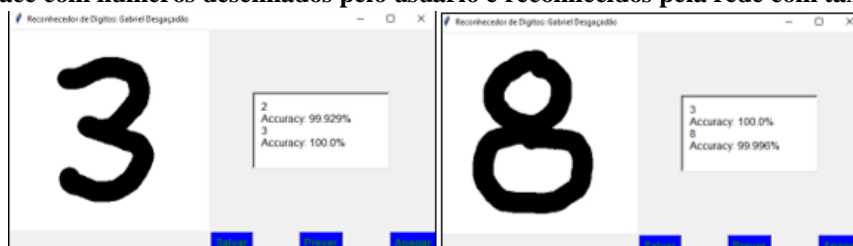


FONTE 5: Autoria própria (Software: Spyder 4.3.2 - Matplot)

4.3 MNIST submetido a CNN

A estrutura da CNN utilizada para aplicação da base MNIST foi elaborada da seguinte maneira: primeiro as imagens foram redimensionadas para seu tamanho e postas em uma variável; depois seu módulo de cor foi retirado fazendo deste modo a imagem tenha apenas um canal (RGB para cinza) presente; por fim pode ser tratada nos passos descritos da criação de uma rede neural convolucional e submetida a rede de aprendizado. Foi criada com 32 kernels de tamanho proporcional aos registros e um *maxpooling* de 2x2. Para a camada de saída com 10 neurônios, a função adotada foi a *softmax* (pois precisa determinar uma probabilidade dos dígitos de 0 a 9). Para exemplificar tal rede, um pequeno software do tipo interface homem máquina (IHM), foi criado. A Figura 7 apresenta o funcionamento da interface.

Figura 5: Interface com números desenhados pelo usuário e reconhecidos pela rede com taxa de certeza



FONTE 6: Autoria própria (Software: Spyder 4.3.2 - Tkinter)



5. CONCLUSÃO

Redes neurais são ferramentas essenciais no ramo de aprendizado de máquinas. São métodos poderosos, pois exigem um número de processamento relativamente pequeno em comparação ao processamento bruto. Estão sendo cada vez mais utilizadas, devido sua aplicabilidade e a possibilidade de elaboração por meio de pacotes e toolboxes de livre acesso na internet. Neste sentido, este trabalho apresentou as etapas de construção de modelos de redes neurais artificiais tradicionais como a MLP; ELM que demonstra a diferença clara da MLP quanto ao uso computacional, sendo rodada por muito menos e obtendo erros baixos; e uma proposta de rede neural convolucional que da noção ao usuário com sua interface, quanto ao como uma RNA pode ser útil

Os resultados computacionais obtidos em bases de dados benchmark mostram a viabilidade da proposta e permitem atingir os objetivos didáticos primordiais. Trabalhos futuros serão desenvolvidos no sentido de aprimorar o conhecimento na área, bem como elaborar novas arquiteturas de redes e procurar por aplicações de mundo real.

AGRADECIMENTOS

Os autores agradecem a Universidade Tecnológica Federal do Paraná pela bolsa concedida pelo edital 22/2019 – Programa de Apoio a Grupos de Pesquisa.

REFERÊNCIAS

- KOVACS L SOLTS. **Redes Neurais Artificiais**, ed. 4, p.13-32, 2006;
- GEHRM L HENRIQUE, QUADROS L GERK. **Rastreamento do câncer de mama no Brasil. Revista Brasileira de Ginecologia e Obstetrícia**, 20 out de 2006;
- ROSENBLATT F. **The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review**, 65(6), 386–408, 1958;
- WOLBERG H WILLIAM, STREET W NICK, MANGASARIAN O. **Breast Cancer Diagnostic Database**, 1993;
- LECUN Y, BURGES J C CHRISTIAN. **MNIST Database of handwritten digits**, 1997;
- HAYKIN S. **Redes Neurais, Princípios e prática**. ed. 2 [S.l.]: Bookman, 1999;
- DONALD H. **The Organization of Behavior**. [S.l.: s.n.], 1949;
- ACKLEY, HINTON, SEJNOWSKIM, **A learning algorithm for Boltzmann Machines**. Redação: Cognitive Science, v. 9, I.1, p.147-169, 1985;
- GUANG-BINHUANG; QIN-YUZHU; CHEE-KHEONG SIEW; **Extreme learning machine: Theory and applications**; P. 489-501, 2006.