

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

AYLEN RAMOS FREITAS

**DESENVOLVIMENTO DO SOFTWARE DE REDES NEURAIS
“NEURO AGROCLIMATE”**

VIÇOSA
MINAS GERAIS - BRASIL
NOVEMBRO/2022

AYLEN RAMOS FREITAS

**DESENVOLVIMENTO DO SOFTWARE DE REDES NEURAIIS
“NEURO AGROCLIMATE”**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia II – e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Alexandre Santos Brandão (DEL/UFV)

Coorientadora: Dra. Mariana Rodrigues Magalhães Romeiro

VIÇOSA
MINAS GERAIS - BRASIL
NOVEMBRO/2022

AYLEN RAMOS FREITAS

**DESENVOLVIMENTO DO SOFTWARE DE REDES NEURAIIS
“NEURO AGROCLIMATE”**

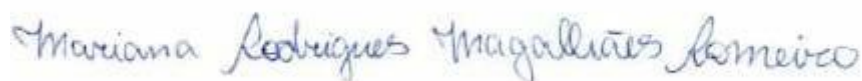
Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia II – e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 24 de novembro de 2022.

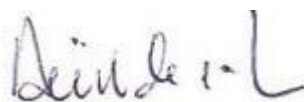
COMISSÃO EXAMINADORA



Prof. Dr. Alexandre Santos Brandão – Orientador
Universidade Federal de Viçosa – UFV



Dra. Mariana Rodrigues Magalhães Romeiro – Coorientadora



Prof. Dr. Aristides Ribeiro - Membro
Universidade Federal de Viçosa – UFV



Prof. Dr. Rodolpho Vilela Alves Neves - Membro
Universidade Federal de Viçosa – UFV

“Todas as coisas cooperam para aqueles que amam a Deus”

Romanos 8:28

*Aos meus pais e avós por sempre
me amarem e apoiarem em tudo!*

Agradecimentos

Em primeiro lugar, agradeço a Deus, minha fonte de vida, rocha inabalável, que permitiu que eu chegasse até aqui. Com Ele superei todos os obstáculos encontrados ao longo desses anos.

Aos meus pais, Cláudia e Nivaldo, e às minhas avós, Sebastiana e Perpétua, por acreditarem no meu sonho, proporcionando total apoio e incentivos durante a graduação. Sem vocês eu não teria conseguido chegar até aqui.

Ao professor Aristides Ribeiro pela oportunidade, confiança e amizade. Sou grata pela imensa contribuição na minha formação profissional, por acreditar nas minhas ideias e apoiar meus sonhos.

À Mariana Magalhães por ter acreditado no meu potencial e me dado a oportunidade de trabalhar ao seu lado. Ao longo do seu doutorado pude aprender muito com você, obrigada por todo o apoio e oportunidade ao longo dos anos.

Ao meu orientador Alexandre Santos Brandão pela oportunidade de atuar no BDP (Believe, Do and Play) e no NERO (Núcleo de Especialização em Robótica), contribuindo grandemente para minha formação.

Às amigas Alessandra Marto, Ana Paula Costa, Débora Ribeiro, Rita de Cássia e Kamila Castro, por estarem comigo e tornarem a trajetória muito mais leve e divertida.

À toda minha família, em especial, à Cristina, Sinha e Cleonice.

À minha célula e à Comunidade Batista da Paz (CBPaz) pelas orações e apoio.

Por fim, agradeço a Universidade Federal de Viçosa que através de suas diversas iniciativas, proporcionaram uma caminhada repleta de aprendizado e oportunidades.

Resumo

Diversas ferramentas computacionais de Inteligência Computacional e de modelagem matemática têm sido utilizadas com êxito em situações de tomada de decisão de problemas complexos, simulando a atitude e o pensamento humano, como a lógica Fuzzy (nebulosa), algoritmos genéticos, destacando-se entre elas as Redes Neurais Artificiais (RNA's). Diante da crescente demanda, o trabalho visa atender às seguintes questões relacionadas: i) Desenvolvimento de um software que possibilite ao usuário: treinar redes neurais artificiais de forma simplificada, modelar sua arquitetura e seus parâmetros de construção; ii) Implementar ao software de redes neurais artificiais, três arquiteturas do tipo Adaline, Multi Layer Perceptron (MLP) e Nonlinear Autoregressive Exogenous (NARX); iii) Implementar a importação e simulação de redes neurais artificiais já treinadas, aplicando-as em outros bancos de dados de entrada; iv) Aplicar metodologias com finalidade de reduzir o número de variáveis de entrada por meio de ordenação e correlação entre as variáveis. Desta forma, implementou-se ao software as correlações de Pearson e Spearman, além do método de Garson. As aplicações permitem otimizar a construção dos modelos de redes neurais, permitindo ao usuário uma ferramenta robusta e dinâmica para o trabalho com redes neurais. De posse do software, o usuário conseguirá treinar redes de uma forma muito mais fácil e intuitiva.

Palavras-chave: Inteligência computacional, redes neurais artificiais, adaline, MLP, NARX.

Abstract

Several computational tools of Computational Intelligence and mathematical modeling have been successfully used in decision-making situations of complex problems, simulating human attitude and thinking, such as fuzzy logic, genetic algorithms, highlighting among them the Artificial Neural Networks (ANN's). In view of the growing demand, the work aims to address the following related issues: i) Development of software that allows the user to: train artificial neural networks in a simplified way, model their architecture and construction parameters; ii) Implement three architectures such as Adaline, Multi Layer Perceptron (MLP) and Nonlinear Autoregressive Exogenous (NARX) to artificial neural networks software; iii) Implement the import and simulation of already trained artificial neural networks, applying them to other input databases; iv) Apply methodologies in order to reduce the number of input variables through ordering and correlation between them. In this way, the Pearson and Spearman correlations were implemented in the software, in addition to the Garson method. The applications allow optimizing the construction of neural network models, providing the user with a robust and dynamic tool for working with neural networks. With the software, the user will be able to train networks in a much easier and more intuitive way.

Keywords: Computational intelligence, artificial neural network, adaline, MLP, NARX.

Sumário

Lista de Figuras	10
Abreviações	11
1 Introdução	12
2 Revisão Bibliográfica	13
2.1 Definição RNA's	13
2.2 Classificação das RNA's	15
2.3 Rede Adaline	17
2.4 Rede MLP	18
2.5 Rede NARX	20
3 Objetivos	21
3.1 Objetivo Geral	21
3.2 Objetivos Específicos	21
4 Materiais e Métodos	21
5 Resultados e Discussão	26
6 Conclusões	29
7 Referência Bibliográficas	30
Apêndice A – Exemplo de aplicação: O caso da radiação	31

Lista de Figuras

Figura 1: Representação simplificada de um neurônio.

Figura 2: Neurônio de McCulloch e Pitts.

Figura 3: Classificação de RNA's em relação à sua estrutura.

(a) RNA de única camada intermediária.

(b) RNA de múltiplas camadas intermediárias.

Figura 4: Classificação de RNA's quanto às suas conexões.

(a) RNA feed forward.

(b) RNA feedback.

Figura 5: Rede neural artificial Adaline.

Figura 6: Arquitetura das Redes MLP

Figura 7: Arquitetura da rede Nonlinear Autoregressive Exogenous (NARX).

Figura 8: Interface para configurar parâmetros da rede e seu treinamento.

Figura 9: Funções de ativação das camadas.

Figura 10: Exemplo de equação de uma rede neural do tipo MLP.

Figura 11: Fluxograma de uso do software.

Figura 12: Configuração da estrutura da rede neural.

Figura 13: Configurações do treinamento.

Figura 14: Critérios de parada do treinamento.

Figura 15: Sensor radiômetro.

Figura 16: Definição de variáveis.

Figura 17: Valor mínimo e máximo das variáveis.

Figura 18: Erros no treinamento, validação e teste.

Figura 19: Aplicação do método de Garson na rede MLP 10.

Figura 20: Análise gráfica da rede MLP 10.

Figura 21: Equação da rede MLP 10.

Abreviações

MLP	Multi Layer Perceptron
MSE	Mean Square Error
NARX	Nonlinear Autoregressive Exogenous
RNA	Rede Neural Artificial
BP	Backpropagation
GD	Gradient descent backpropagation
GDM	Gradient descent backpropagation com momentum
GDA	Gradient descent backpropagation com taxa adaptativa
GDX	Gradient descent backpropagation com momentum e taxa adaptativa
LM	Levenberg-Marquardt backpropagation
RP	Resilient backpropagation
BFG	BFGS quasi-Newton backpropagation
LM	Levenberg-Marquardt
BR	Bayesian Regularization
SCG	Scaled Conjugate Gradient

1 Introdução

As Redes Neurais Artificiais (RNA's) são estruturas matemáticas capazes de aprender, memorizar e generalizar determinadas situações e problemas a elas apresentadas. Elas são sistemas compostos por unidades elementares, denominadas de neurônios, que calculam determinadas funções matemáticas, geralmente não-lineares, cujo funcionamento é inspirado no próprio cérebro humano. O procedimento pelo qual uma RNA encontra as soluções, passa por um processo de aprendizado, onde uma série de dados de entrada e saída são apresentados às suas unidades elementares, que por si só encontram as características necessárias para representar a informação fornecida e definir o modelo resultante (BRANDÃO, 2005).

A partir dos dados de entrada que são fornecidos para as redes neurais, é possível que as mesmas aprendam e generalizem os dados aprendidos, permitindo que após o treinamento novos dados desconhecidos sejam aplicados, desde que os mesmos se assemelhem com os dados já utilizados na etapa de treinamento, ou seja, desde que esses dados estejam dentro dos limites inferiores e superiores.

As redes neurais artificiais são capazes de extrair características que não estejam explicitamente aparentes. As características mais atrativas das RNA's consistem de suas habilidades em modelar sistemas lineares e não-lineares, aprendendo comportamentos envolvidos a partir de informações fornecidas.

Uma das vantagens das RNA's sobre os métodos convencionais de modelagem é que elas não requerem informações detalhadas sobre os processos físicos, químicos ou biológicos do sistema a ser modelado, mas apenas um banco de dados representativo do mesmo (SUDHEER, 2003).

Diversas ferramentas computacionais e de modelagem matemática têm sido utilizadas com êxito em situações de tomada de decisão, como a lógica Fuzzy, algoritmos genéticos, destacando-se entre elas as Redes Neurais Artificiais. Softwares existentes no mercado até o momento, como o Neuroforest e o Statistica são de grande utilização no setor florestal, mas devido ao grande volume de dados apresentam problemas de limitações em seu uso, tais como: limitação no número de linhas e colunas dos dados de input, relatórios simples e sem opção de exportação das análises realizadas, falta da opção de salvamento e utilização posterior da rede neural artificial treinada, falta de uma rede neural mais simples (como a Adaline), assim como

a falta de uma rede neural mais complexa (como a de série temporal). Além disso, tais programas trabalham com limitação significativa na quantidade de variáveis do input.

Outro ponto importante é a implementação de métodos para a redução do número de variáveis de entrada, por meio da aplicação de métodos de ordenação de importância, a fim de poder otimizar a construção de modelos de redes neurais para modelagem e previsões. A seleção das variáveis dependentes será capaz de otimizar o tempo de processamento das redes e o custo para coleta das variáveis de entrada. Para reduzir a quantidade de variáveis de entrada, será implementado no software 3 metodologias: modelos construídos com a ordenação da Correlação de Pearson, Correlação de Spearman e modelos construídos com a ordenação do Método de Garson.

A idealização do software foi feita pela pesquisadora Mariana Rodrigues Magalhães Romeiro, já o desenvolvimento do software foi feito em conjunto. O trabalho em questão fez parte de sua tese de doutorado, apresentada à Universidade Federal de Viçosa em 2017, intitulada como “Prognose do volume de madeira em florestas equiâneas por meio de modelos agrometeorológicos de redes neurais artificiais”. Somente a idealização do software proporcionou 2 premiações em congressos, a primeira no XVIII Congresso Brasileiro de Meteorologia e a segunda no XVIII Congresso Brasileiro de Agrometeorologia.

Com isso, o trabalho consistiu no desenvolvimento do software Neuro AgroClimate, software este capaz de ampliar o processamento das redes neurais artificiais, e suprir as limitações dos programas disponíveis (Neuroforest e Statistica) utilizados atualmente na área florestal no Brasil.

2 Revisão Bibliográfica

2.1 Definição RNA's

O cérebro humano é formado por neurônios que enviam e recebem informações para milhares de outros neurônios conectados a eles por meio de sinapses. O cérebro é projetado para gerenciar nossos corpos em termos de emoções, raciocínio e funções motoras. As RNA's, por outro lado, têm como objetivo simular esse mundo de atividade cerebral, implementando comportamentos básicos e dinâmicas funcionais. Texto adaptado de [1].

O advento das RNA's ocorreu utilizando o modelo matemático de neurônios biológicos proposto por McCulloch e Pitts (1943). O modelo em si era uma simplificação do então conhecido neurônio biológico. Para representar os dendritos, o modelo consistiu em n variáveis de entrada x_1, x_2, \dots, x_n e um terminal de saída y representando o axônio. Cada entrada tem um peso ponderador com a finalidade de simular as sinapses do cérebro humano. Na Figura 1 observamos a representação simplificada das partes que compõem o neurônio.

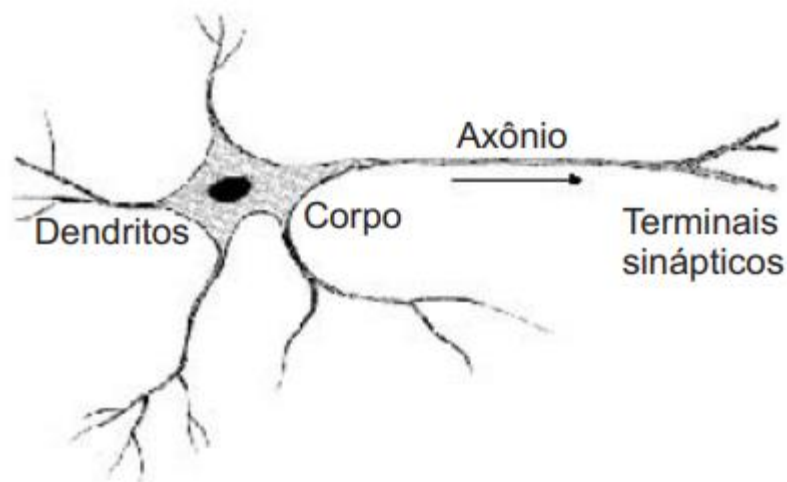


Fig. 1: Representação simplificada de um neurônio [9].

Pela sua construção, a semelhança das redes neurais com o cérebro humano dá-se pelas características de poder adquirir conhecimento por meio de processos de aprendizagem e pelas forças de conexão que existem entre os neurônios artificiais (chamadas de pesos sinápticos) (HAYKIN, 2001).

São nos pesos sinápticos que o conhecimento é armazenado. Semelhante aos neurônios biológicos, as sinapses ocorrem apenas quando a soma ponderada dos sinais de entrada excede um limite predefinido. No modelo proposto, os limites são definidos como booleanos, acionados ou não acionados como resultado da função de ativação, conforme mostrado na Figura 2.

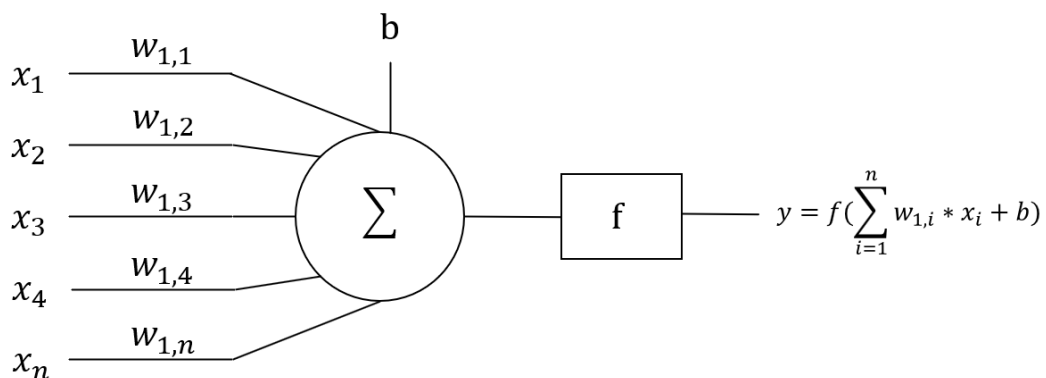


Fig. 2: Neurônio de McCulloch e Pitts.

A saída y do neurônio de McCulloch e Pitts (1943) pode ser equacionada por:

$$y = f\left(\sum_{i=1}^n x_i * w_i + b\right)$$

onde n é o número de variáveis de entradas do neurônio, w_i é o peso associado à cada entrada x_i , b é a constante bias e f é a função de ativação aplicada ao modelo.

Todavia, uma restrição existente no modelo criado, é que as redes desenvolvidas só conseguem implementar funções que sejam linearmente separáveis, ou seja, aquelas que se podem separar os padrões por meio de uma reta.

2.2 Classificação das RNA's

As RNA's podem ser classificadas quanto a sua estrutura, conexões e tipo de treinamento.

a) Quanto à sua estrutura:

- Redes de única camada intermediária: existe apenas uma camada de neurônio entre as variáveis de entrada e a variável de saída (Figura 3a).
- Redes de múltiplas camadas intermediárias: existem mais de uma camada de neurônios entre as variáveis de entrada e a variável de saída (Figura 3b).

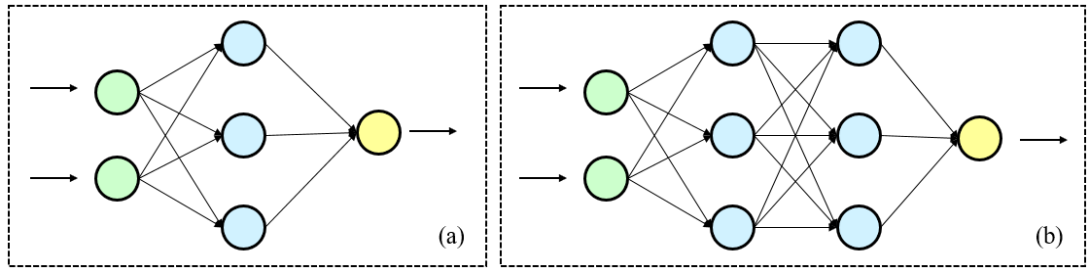


Fig. 3: Classificação de RNA's quanto à sua estrutura.

(a) RNA de única camada intermediária. (b) RNA de múltiplas camadas intermediárias.

b) Quanto às suas conexões:

- Redes feed forward: são estruturas neurais, onde a saída de um neurônio de uma dada camada realiza conexões com os neurônios das camadas seguintes e não com os das camadas antecessoras (Figura 4a).
- Redes feedback: são aquelas cuja saída de uma camada também pode atuar tanto na entrada das camadas antecessoras como nas posteriores (Figura 4b).

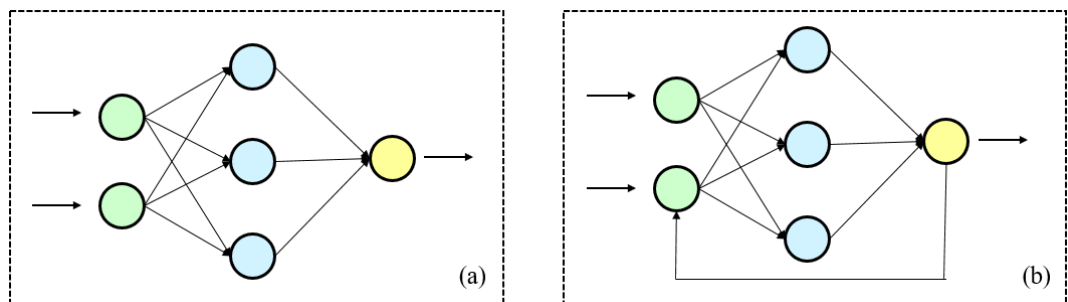


Fig. 4: Classificação de RNA's quanto às suas conexões.

(a) RNA feed forward. (b) RNA feedback.

c) Quanto ao treinamento:

- Supervisionado: caracteriza-se pela existência de um supervisor que monitora a saída da rede, e compara a mesma com a resposta desejada. Posteriormente, calcula-se o erro existente entre estas respostas e realiza-se o ajuste dos pesos sinápticos até

que um erro mínimo pré-estabelecido seja alcançado. No software foi definido que o treinamento será concluído caso o erro mínimo seja atingido ou o número de iterações (ciclos) pré-estabelecido seja superado.

- Não supervisionado: caracteriza-se pela não existência de saídas desejadas, sendo o conjunto de treinamento estabelecido somente pelo vetor de entrada. A atualização dos pesos sinápticos é obtida com base nos próprios valores de entrada. Este tipo de treinamento é aplicado a problemas de categorização de dados.
- Reforço: é um treinamento que mescla as características do treinamento supervisionado e do não supervisionado. O conjunto de treinamento é formado simplesmente por dados de entrada, todavia há um crítico, similar ao supervisor, que reforça ou penaliza a saída da rede.

2.3 Rede Adaline

A rede neural Adaline, do inglês Adaptive Linear Neuron, foi proposta por Widrow e Hoff (1960), e possui a mesma estrutura que o Perceptron. Em comparação com a rede Perceptron, a rede Adaline se diferencia pelo algoritmo de treinamento, conhecido como Regra Delta, e pela variável de saída. Na rede Perceptron o resultado do neurônio de saída é uma variável booleana, já na rede Adaline o resultado é uma função linear.

A Adaline é uma rede neural que possui uma camada de entrada com n unidades, uma camada de saída com apenas uma unidade e não apresenta camadas escondidas (Figura 5). As atividades dos seus neurônios fornecem uma representação do padrão de entrada, sendo apresentado à rede neural e os seus valores são números reais quaisquer.

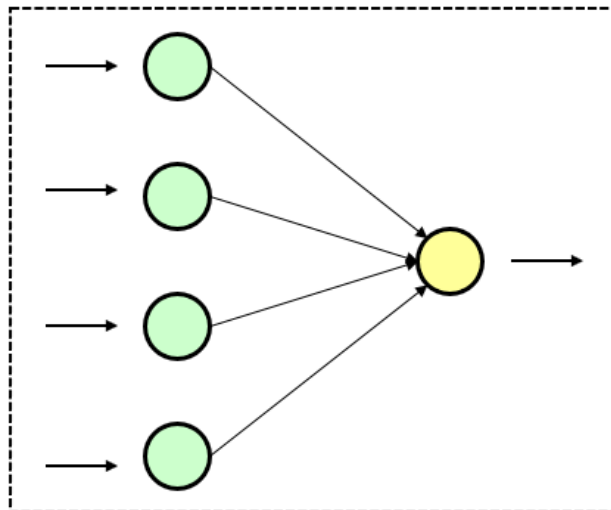


Fig. 5: Rede neural artificial Adaline.

O treinamento da rede neural Adaline consiste no treinamento por um algoritmo de aprendizado supervisionado. Durante o período de treinamento, para cada padrão de entrada apresentado $x = (x_1, x_2, \dots, x_n)$ deve existir um padrão de saída desejado “d” e o objetivo do treinamento é fazer com que a saída “S” da rede para esse padrão seja igual a “d”.

Em 1960, quando a rede Adaline foi idealizada por Widrow e Hoff,, sua principal aplicação foi nos sistemas de chaveamento de circuito telefônicos, sendo esta as primeiras aplicações em ambiente industriais que utilizavam redes neurais artificiais na época (SILVA, SPATTI, FLAUZINO, 2010). De forma geral, podemos dizer que suas aplicações se estriguem a problemas do tipo lineares.

2.4 Rede MLP

As RNA's com uma camada de neurônio são capazes de resolver problemas linearmente separáveis, contudo apesar de resolver vários problemas, existe uma outra vasta coleção de problemas não linearmente separáveis. Este problema foi proposto por Minsky e Pappert na década de 70, quando, em suas publicações, depreciaram a habilidade das RNA de encontrar soluções para problemas simples. A solução encontrada para contornar este problema e como consequência retomar as pesquisas sobre RNA, até então

desacreditas por Minsky e Pappert, foram as estruturas neurais de múltiplas camadas, também conhecida como redes MLP (Multi Layer Perceptron), na década de 80.

As redes MLP apresentam a arquitetura mostrada na Figura 6, onde se encontram a camada de entrada, as camadas intermediárias (ocultas) e a camada de saída. O número de variáveis da camada de entrada depende diretamente do número de características informadas para a rede neural. O número de neurônios das camadas intermediárias depende da complexidade do problema. E a camada de saída contém o número de neurônios necessários para modelar a variável de saída em função das variáveis de entrada.

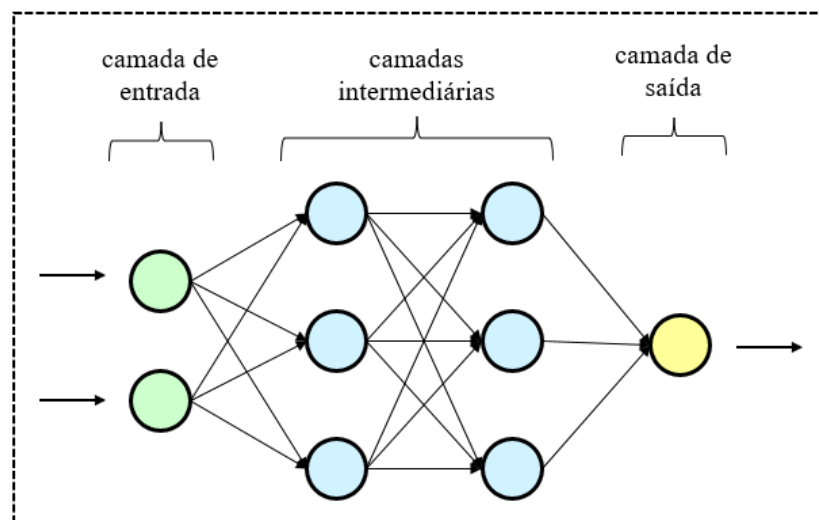


Fig. 6: Arquitetura das Redes MLP

O número de neurônios das camadas intermediárias é determinado de forma empírica, atentando para o caso de overfitting (ou superajuste), onde existe uma grande quantidade de neurônios e a estrutura ao invés de generalizar as informações, acabar por memorizar os padrões apresentados tornando-se incapaz de classificar padrões semelhantes. Outro efeito do superajuste é que a RNA além de armazenar as características relevantes extraídas das amostras, esta passa a guardar em seus pesos informações de ruídos que a princípio não revelam interesse. Por outro lado, caso o número de neurônios seja inferior ao desejado, pode ocorrer um underfitting e a RNA não convergir para uma resposta devido a uma sobrecarga de informações a serem armazenadas em poucos pesos.

Basicamente, o processo de treinamento da rede MLP tem como objetivo, durante o treinamento, a minimização da função de erro quadrático médio (mse – mean square error). O treinamento da RNA deve ocorrer até que se complete um número predeterminado de iterações (ciclos) para atualização dos pesos ou quando o erro quadrático médio se encontrar abaixo de um valor pré-estabelecido.

As redes MLP são destinadas a resolver problemas diversos, tanto de punho linear, como o não linear. Isso deve-se ao fato de sua arquitetura de rede possuir camadas intermediárias. Logo, diversas aplicações que não são linearmente presente no mundo real, são possíveis de solução com as redes MLP.

2.5 Rede NARX

A rede neural artificial NARX (Nonlinear Autoregressive Exogenous) é uma rede comumente utilizada para sistemas dinâmicos não lineares. Ela é uma RNA não-linear auto regressiva com entradas exógenas, isto é, a camada de entrada recebe tanto dados da própria realimentação com atraso de tempo, quanto dados exteriores com atrasos. Sendo assim, o resultado se torna dinâmico e varia em função do tempo. Dessa forma, valores já obtidos influenciam nos resultados subsequentes. A Figura 7 ilustra uma topologia para a rede NARX.

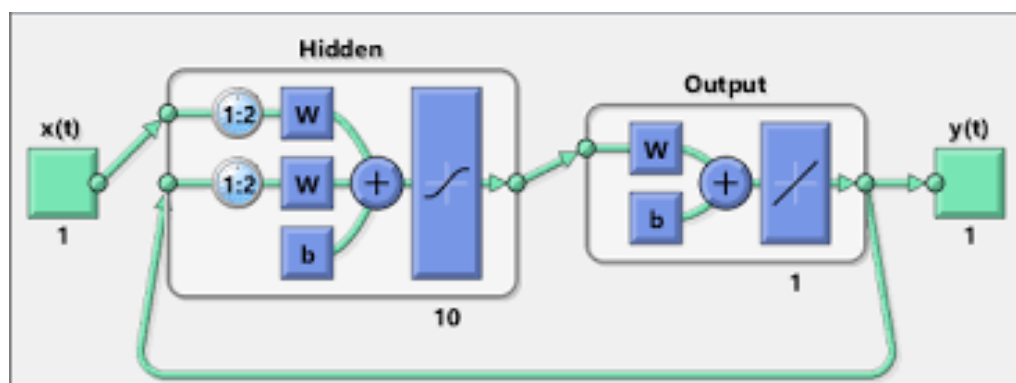


Fig. 7: Arquitetura da rede Nonlinear Autoregressive Exogenous (NARX) [10].

As redes NARX também são aplicadas para problemas lineares e não lineares, mas agora temos que os dados de retroalimentação da saída e os de delay na entrada são importantes para a modelação do sistema. Exemplos de aplicação podem ser vistos em uma gama de problemas de previsões de séries temporais.

3 Objetivos

3.1 Objetivo Geral

Este trabalho tem por objetivo apresentar o software criado, denominado “Neuro AgroClimate”, que permite ao usuário realizar a criação de RNA’s do tipo Adaline, MLP e NARX, realizar o treinamento e aplicação dessas redes neurais artificiais.

3.2 Objetivos Específicos

Diante do exposto, tem-se por objetivos específicos:

- i) Mostrar o processo de desenvolvimento do Neuro AgroClimate, que pode ser utilizado como sistema de apoio de tomada de decisões por meio da Redes Neurais Artificiais (RNA’s);
- ii) Trabalhar com variáveis quantitativas e qualitativas (entrada binária do input);
- iii) Aplicação de métodos estatísticos que determinarão quais variáveis de entrada apresentarão maior sensibilidade à variável de saída;
- iv) Calibrar, validar e simular modelos agrometeorológicos, avaliando o desempenho;
- v) Implementar sistema de acessibilidade para navegação pelo software por meio de teclas de atalho.

4 Materiais e Métodos

O software Neuro AgroClimate foi desenvolvido por completo na linguagem de programação MATLAB, que é uma linguagem de programação de alto nível e de fácil implementação computacional.

Foram implementadas 3 arquiteturas de redes neurais, sendo elas: Adaline (Figura 5), Perceptron Multicamadas – MLP (Figura 6) e Nonlinear Autoregressive Exogenous – NARX (Figura 7).

Considera como algoritmo de treinamento um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais, estes algoritmos diferem principalmente entre si principalmente pelo modo como os pesos são modificados.

Para as redes do tipo Adaline e MLP foram disponibilizados para o usuário a escolha dos seguintes algoritmos de treinamento:

- Gradient descent backpropagation (GD);
- Gradient descent backpropagation com momentum (GDM);
- Gradient descent backpropagation com taxa adaptativa (GDA);
- Gradient descent backpropagation com momentum e taxa adaptativa (GDX);
- Levenberg-Marquardt backpropagation (LM);
- Resilient backpropagation (RP);
- BFGS quasi-Newton backpropagation (BFG).

Já para a rede NARX foram disponibilizados os seguintes algoritmos:

- Levenberg-Marquardt (LM);
- Bayesian Regularization (BR);
- Scaled Conjugate Gradient (SCG).

No primeiro ciclo de treinamento a rede produz uma saída, que ao ser subtraída do valor desejado, determina o erro. O objetivo do processo de treinamento é buscar diminuir o valor do erro. Para isso, o valor dos pesos deve ser ajustado a cada novo passo do treinamento. A regra backpropagation (BP) proporciona que os pesos da camada de saída sejam os primeiros a serem ajustados e, posteriormente, os pesos das demais camadas, de trás para frente. (Brondino, 1999). Existem diversas variações do algoritmo BP, são elas: Gradient descent backpropagation, Gradient descent backpropagation com momentum, Gradient descent backpropagation com taxa adaptativa, Gradient descent backpropagation com momentum e taxa adaptativa, Levenberg-Marquardt backpropagation, Resilient backpropagation, entre outros.

Na Figura 8 temos a interface do programa responsável por escolher e configurar a arquitetura da rede. Nela informamos o tipo de rede (Adaline, MLP e NARX), o número de neurônios da camada intermediária, as funções de ativação da camada intermediária e de saída, os parâmetros de treinamento como o algoritmo, a taxa de aprendizagem, a quantidade de rede a ser treinada, as proporções do banco de dados utilizado para treinamento, validação e teste, e os critérios de parada como a quantidade de ciclos (iterações), o erro médio quadrático (mse) e a validation checks. Nessa interface, é possível ver os valores mínimos e máximos de cada variável de entrada e variável de saída, cuja informação é importante para simulação e aplicação da rede em outro input.

The screenshot shows the 'Neuro AgroClimate' application window. At the top, there is a menu bar with options: Arquivo, Dados, Correlações, Redes Neurais, Relatórios, Gráficos, Equação de Garson, and Sobre. The main interface is divided into several sections:

- Dados - Variáveis Contínuas:** A table showing input variable statistics.

	u2	Tmax	Tmin	RH
Valor Máximo	1.7000	36.6000	21.1000	94.2000
Valor Mínimo	0.4000	16.4000	5.2000	49.3000
- Dados - Variável Saída:** A table showing output variable statistics.

	Rs
Valor Máximo	30.4000
Valor Mínimo	3.1000
- Parâmetros da Rede Neural:**
 - Tipo de Rede:** Three options are shown: Curve Fitting - MLP, Dynamic Time Series - NARX, and Adaline.
 - Arquitetura da Rede:** Input layer: 4, Hidden layer: 5, Output layer: 1.
 - Função de Ativação:** Hidden layer: Tangente Hiperbólica, Output layer: Identidade.
 - Treinamento:** Algorithm: Resilient backpropagation, Learning rate: 0.5, Number of repetitions: 2, % Training data: 70, % Validation data: 15, % Test data: 15.
 - Critério de Parada:** Number of iterations: 1000, Mean Squared Error: 0.0001, Validation Checks: 1000.
- Buttons:** Gravar Parâmetros, Treinar Rede, Salvar Rede.

Fig. 8: Interface para configurar parâmetros da rede e seu treinamento.

Em se tratando de funções de ativação, temos 4 principais opções de funções disponíveis para escolha do usuário, tanto para a camada intermediária, quanto para a camada de saída. São elas: a função identidade, degrau, tangente hiperbólica e sigmoide, conforme a Figura 9.

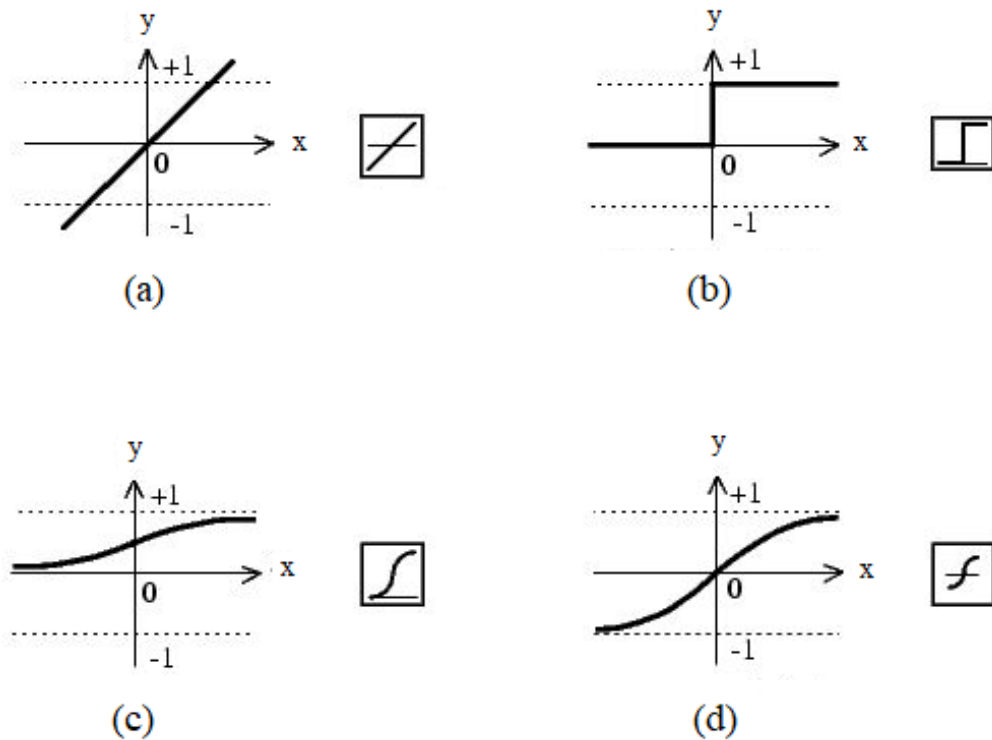


Fig. 9: Funções de ativação das camadas.

(a) Identidade. (b) Degrau. (c) Sigmoide. (d) Tangente hiperbólica.

O Neuro AgroClimate contará com a opção “Correlação de Pearson e Spearman”, que podem ser usados para ordenar variáveis dependentes que serão utilizadas no processamento e treinamento da rede, classificando a importância relativa das variáveis de entrada ajuste igual ou melhor ao modelo com todas as variáveis, usando um número reduzido de variáveis (MAGALHÃES, 2017). Essas correlações foram utilizadas para classificação da importância relativa das variáveis de entrada do modelo. Com os valores finais da correlação, um ranking poderá ser gerado com a classificação da importância relativa das variáveis.

Outro método importante idealizado por MAGALHÃES e implementado no Neuro AgroClimate foi o Método de Garson. Esse método também visa a classificação da importância relativa das variáveis de entrada (variáveis dependentes). Este método envolve essencialmente a divisão dos pesos das conexões de saída de cada neurônio escondido nos componentes associados com cada entrada da rede neural. Como contraste ao método de pesos de conexão geral, este método usa os valores absolutos dos pesos de conexão (KAUR, 2005).

Detalhes desse método proposto por GARSON (1991) são vistos a seguir.

- 1) Para cada neurônio i da camada intermediária, o valor absoluto do peso da conexão entre este neurônio e um da camada de saída é multiplicado pelo valor absoluto do peso da conexão entre o mesmo neurônio da camada intermediária e um neurônio da camada de entrada. Este cálculo deve ser feito para todos os j -ésimos neurônios da camada de entrada (P_{ij});
- 2) Para cada neurônio da camada intermediária, dividiu-se cada P_{ij} obtido na etapa 1) pela soma de todos os P_{ijs} para cada neurônio da camada da entrada para obter Q_{ijs} ;
- 3) Para cada neurônio da camada de entrada, os valores de Q_{ijs} foram somados obtendo-se o valor de S_j ;
- 4) Dividiu-se cada valor de S_j pela soma de todos os valores de S_j . Dessa forma foi obtida a importância relativa ou contribuição de cada um dos valores.

Os passos acima podem ser representados com as seguintes equações a seguir:

$$P_{i,j} = w_{i,j} * w_{i,o}$$

$$Q_{i,j} = P_{i,j} / \sum_{i=1}^n \sum_{j=1}^n P_{i,j}$$

$$S_j = \sum_{i=1}^n Q_{i,j}$$

$$R_j = \left(\frac{S_j}{\sum_{j=1}^n S_j} \right) * 100$$

Outra funcionalidade que o software Neuro AgroClimate apresenta é a opção de transformar a rede neural em uma única equação que a represente. Nesta opção temos a conversão do modelo e todas as conexões da rede de forma explícita em equação. Essa funcionalidade pode ser útil quando se deseja eliminar a rede neural e trabalhar diretamente com a equação. Um exemplo dessa equação que represente uma rede neural artificial do tipo MLP é mostrada na Figura 10.

```

1 Saída Rede = 3.1 + ((purelin(tansig( (2*(u2 - (0.4))/(1.7 - (0.4)) - 1)*(0.21558) +
2 (2*(Tmax - (16.4))/(36.6 - (16.4)) - 1)*(0.79352) +
3 (2*(Tmin - (5.2))/(21.1 - (5.2)) - 1)*(1.8795) +
4 (2*(RH - (49.3))/(94.2 - (49.3)) - 1)*(0.39351) + (-2.1658))*(0.46898) +
5 tansig( (2*(u2 - (0.4))/(1.7 - (0.4)) - 1)*(0.2582) +
6 (2*(Tmax - (16.4))/(36.6 - (16.4)) - 1)*(0.02527) +
7 (2*(Tmin - (5.2))/(21.1 - (5.2)) - 1)*(0.975) +
8 (2*(RH - (49.3))/(94.2 - (49.3)) - 1)*(1.329) + (-0.66215))*(-0.93831) +
9 tansig( (2*(u2 - (0.4))/(1.7 - (0.4)) - 1)*(0.48057) +
10 (2*(Tmax - (16.4))/(36.6 - (16.4)) - 1)*(1.5935) +
11 (2*(Tmin - (5.2))/(21.1 - (5.2)) - 1)*(3.4092) +
12 (2*(RH - (49.3))/(94.2 - (49.3)) - 1)*(3.0341) + (-1.4181))*(0.37527) +
13 tansig( (2*(u2 - (0.4))/(1.7 - (0.4)) - 1)*(1.2568) +
14 (2*(Tmax - (16.4))/(36.6 - (16.4)) - 1)*(2.4358) +
15 (2*(Tmin - (5.2))/(21.1 - (5.2)) - 1)*(0.48772) +
16 (2*(RH - (49.3))/(94.2 - (49.3)) - 1)*(-0.30095) + (0.44974))*(0.61549) +
17 tansig( (2*(u2 - (0.4))/(1.7 - (0.4)) - 1)*(2.2934) +
18 (2*(Tmax - (16.4))/(36.6 - (16.4)) - 1)*(4.1872) +
19 (2*(Tmin - (5.2))/(21.1 - (5.2)) - 1)*(3.1339) +
20 (2*(RH - (49.3))/(94.2 - (49.3)) - 1)*(-3.8382) +
21 (1.5892))*(-0.31021) + (0.17078) + 1)*(30.4 - (3.1))/2
22

```

length : 1.232 lines : 22 Ln : 21 Col : 57 Pos : 1.232 Unix (LF) UTF-8 INS

Fig. 10: Exemplo de equação de uma rede neural do tipo MLP.

Além disso, o software importa dados de qualquer versão do Excel, trabalha com variáveis quantitativas e qualitativas (converte-as automaticamente para dados binários), salva redes treinadas e exporta para o Excel diversos relatórios. O próprio programa verifica outliers no banco de dados usado pelas redes neurais, permite treinar várias redes no mesmo processo e salva quantas redes neurais o usuário deseja, entre outras funcionalidades.

O fluxograma de uso do software na fase de treinamento pode ser visto na Figura 11, onde a cada etapa de redução de variáveis, seja pelas correlações estatísticas (Pearson e Spearman), ou pelo método de Garson, o usuário deve reiniciar o programa, juntamente com seu processamento. Ao final do processo, as redes neurais escolhidas poderão ser salvas e utilizadas posteriormente.

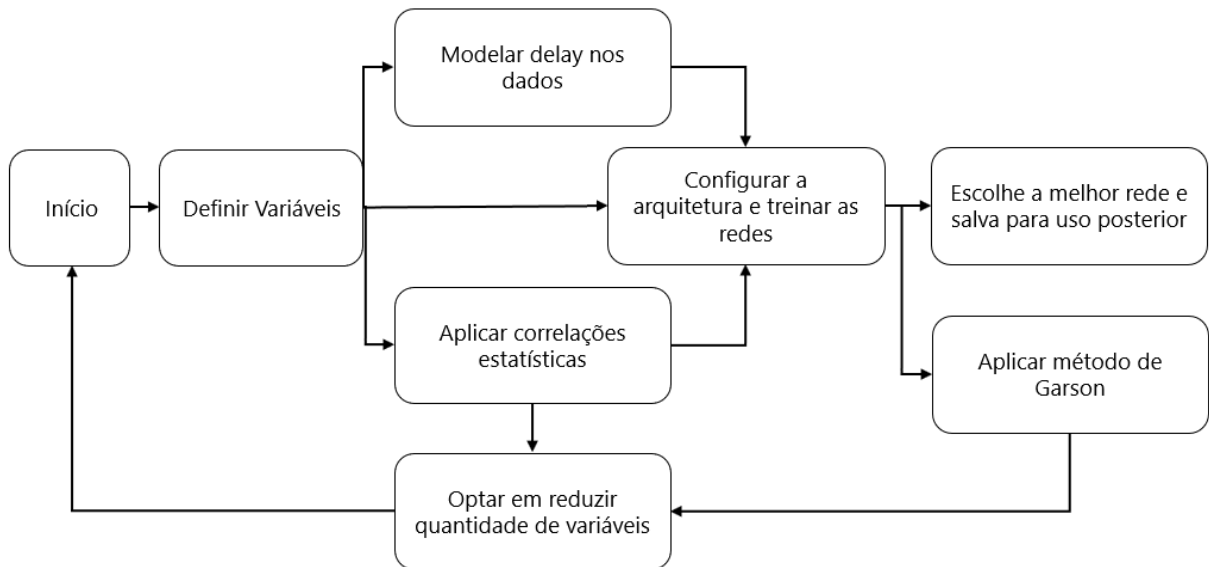




Fig. 11: Fluxograma de uso do software.

5 Resultados e Discussão

O Neuro AgroClimate é capaz de realizar previsões matemáticas por meio de aproximação de função ou modelagem dinâmica usando redes neurais artificiais. O software foi idealizado e implementado, tendo como o objetivo ser aplicado nos mais diversos bancos de dados, sejam eles quantitativos e qualitativos.

O programa possui uma interface de fácil manuseio, permitindo ao usuário vasta liberdade para escolher a estrutura da rede: tipo, camadas intermediárias, funções de ativação da camada intermediária e de saída (Figura 12). Ainda, algoritmos de treinamento, taxa de aprendizagem, porcentagem de particionamento do banco de dados, dividindo-o em 3 banco de dados, sendo deles de treinamento, teste e validação (Figura 13). Ainda, quanto a critério de paradas, temos as configurações do número de ciclos, erro quadrático médio e validation checks (Figura 14). O Neuro AgroClimate permite salvar as redes treinadas para posterior validação e sua utilização.

Tipo de Rede

-  **Curve Fitting - MLP**
-  **Dynamic Time Series - NARX**
-  **Adaline**

Arquitetura da Rede

Camada Entrada:

Camada Intermediária:

Camada Saída:

Função de Ativação

Camada Intermediária:

Camada Saída:

Fig. 12: Configuração da estrutura da rede neural.

Treinamento

Algoritmo:

Taxa de Aprendizado: Número de Repetições:

% Dados Treinamento: % Dados Validação:

% Dados Teste:

Fig. 13: Configurações do treinamento.

Critério de Parada	
Número de Iterações:	1000
Erro Quadrático Médio:	0.0001
Validation Checks:	1000

Fig. 14: Critérios de parada do treinamento.

A primeira versão do Neuro AgroClimate oferece:

- a) Importação e exportação dos dados no formato ".xls" e ".xlsx" com opção de escolha da planilha desejada;
- b) Opções de configurações da arquitetura da rede entre Adaline, MLP ou NARX (Figura 11);
- c) Exportação de diversos resultados da etapa de treinamento, separados por planilhas (sumário, predição e pesos);
- d) Opção de editar os dados dentro da interface gráfica e, se desejado, salvar as correções no formato ".xls" e ".xlsx";
- e) Possibilidade de treinar várias redes no mesmo processamento e salvar as redes que o usuário desejar;
- f) Ao utilizar a rede treinada em outro banco de dados, o software apresenta a funcionalidade de analisar se as variáveis estão dentro da amplitude dos dados da etapa de treinamento;
- g) Plotagem de gráficos da saída estimada versus a saída observada;
- h) Cálculo da “Correlação de Pearson” e “Correlação de Spearman”;
- i) Cálculo do “Método de Garson”.

A primeira versão do software Neuro AgroClimate levou 24 meses para ser concluída e foi desenvolvida com uma bolsa de iniciação científica via empresa. As principais atividades estão descritas abaixo.

- 1) Estudos e revisão da literatura sobre as diversas arquiteturas de redes neurais artificiais;
- 2) Desenvolvimento do escopo principal do software, via linguagem de programação Matlab;
- 3) Implementação de métodos e testes estatísticos;
- 4) Construção de relatórios padronizados conforme a necessidade do usuário;
- 5) Registro do software no INPI (Instituto Nacional da Propriedade Industrial).

6 Conclusões

A premissa básica da criação do Neuro AgroClimate foi criar um software eficiente e com uma interface de fácil manipulação para atender à necessidade dos processamentos propostos e que pudesse ser utilizada posteriormente por outros usuários em outros bancos de dados. Todas as interfaces possuem teclas de acessibilidade, o que permite que pessoas com deficiências motoras naveguem e usufruam do software de forma plena.

O principal desafio no desenvolvimento do software foi compreender a teoria e os conceitos acerca das redes neurais artificiais. O Neuro AgroClimate foi registrado no INPI sob a numeração BR512016000503-3. Para futuros trabalhos, com os conhecimentos adquiridos, pretende-se expandir o software para outros tipos de arquiteturas de redes, abrangendo uma maior quantidade de problemas a serem solucionados com o Neuro AgroClimate.

7 Referência Bibliográfica

- [1] Brandão, A. S., Redes Neurais Artificiais aplicadas ao reconhecimento de comandos de voz, 2005.
- [2] MAGALHÃES, Mariana Rodrigues. Prognose do volume de madeira em florestas equiâneas por meio de modelos agrometeorológicos de redes neurais artificiais. 2017. Tese (Doutorado em Meteorologia Agrícola) – Engenharia Agrícola, Universidade Federal de Viçosa, Viçosa, 2017.
- [3] Braga, A. P., Carvalho, A. C. P. L. F., Ludermir, T. B. Redes Neurais Artificiais, Teoria e Aplicações. Ed LTC, 2000.
- [4] Haykin, S. Redes Neurais, Princípios e Prática. Ed. Bookman, 2001.
- [5] Sudheer, K. P., Gosain, A. K., Ramasastri, K. S. Estimating actual evapotranspiration from limited climatic data using neural computing technique. Journal of Irrigation and Drainage Engineering, 2003.
- [6] McCulloch, W. S., Pitts, W. H. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 1943.
- [7] Widrow, B., Hoff Jr., M. E. Adaptive switching circuits, IRE WESCON Convention Record, 1960.
- [8] Minsky, M. L., Pappert, S. Perceptron: an introduction to computational geometry. Cambridge: MIT Press, 1969.
- [9] Ferneda, Edberto. Redes neurais e sua aplicação em sistemas de recuperação de informação. Ribeirão Preto, USP, 2006. Disponível em: <<https://www.scielo.br/j/ci/a/SQ9myjZWLxnyXfstXMgCdcH/?format=pdf&lang=pt>>. Acesso em: 05 de novembro de 2022.
- [10] The structure of a non-linear autoregressive neural network (NARX) with external input from the MatLab neural network simulation module environment. Disponível em: <https://www.researchgate.net/figure/The-structure-of-a-non-linear-autoregressive-neural-network-NARX-with-external-input_fig1_264083064>. Acesso em: 05 de novembro de 2022.

- [11] Kaur, K., Kaur, A., Malhotra, R., Alternative methods to rank the impact of object oriented metrics in fault prediction modeling using neural networks. *International Journal of Engineering and Applied Sciences*, 2005.
- [12] Garson, D. G. Interpreting neural network connection weights. *Artificial Intelligence Expert*, n.6, v.4, 1991.
- [13] HAYKIN, S. *Redes Neurais - Princípios e prática*. 2. ed. Porto Alegre: Bookman, 2001.
- [14] SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. *Redes neurais artificiais para engenharia e ciências aplicadas*. São Paulo: Artliber, 2010.
- [15] Brondino, M. C. (1999). *Estudo da influência da acessibilidade no valor de lotes urbanos através de redes neurais*. Tese (Doutorado em Engenharia Civil - Transportes), Escola de Engenharia de São Carlos da Universidade de São Paulo, São Paulo.
- [16] Xavier, A. C., Scanlon, B. R., King, C. W., & Alves, A. I. (2022). New improved Brazilian daily weather gridded data (1961–2020). *International Journal of Climatology*.

Apêndice A – Exemplo de aplicação: O caso da radiação

Para exemplificar o uso do software Neuro AgroClimate foi escolhido um exemplo de aplicação para estimativa da radiação solar (MJ/m^2) a partir de dados diários de temperatura mínima ($^{\circ}\text{C}$), temperatura máxima ($^{\circ}\text{C}$), umidade relativa do ar (%), velocidade do vento (m/s) e precipitação (mm).

Tal exemplo se aplica na realidade, por exemplo, quando temos na estação meteorológica sensores de temperatura, umidade relativa do ar, pluviômetro, velocidade e direção do vento, mas não temos o sensor radiômetro (Figura 15), uma vez que seu custo é bem elevado comparado aos outros sensores. Sendo assim, em vez de medir a radiação, podemos utilizar das redes neurais artificiais para estimar seu valor.



Fig. 15: Sensor radiômetro.

Nesta simulação, foram utilizados dados do ano de 2000 para a cidade de Viçosa/MG, da série histórica [16], cuja latitude é $-20,7546$ e longitude $-42,8825$. Foram definidas as variáveis de entrada e saída conforme a Figura 16.

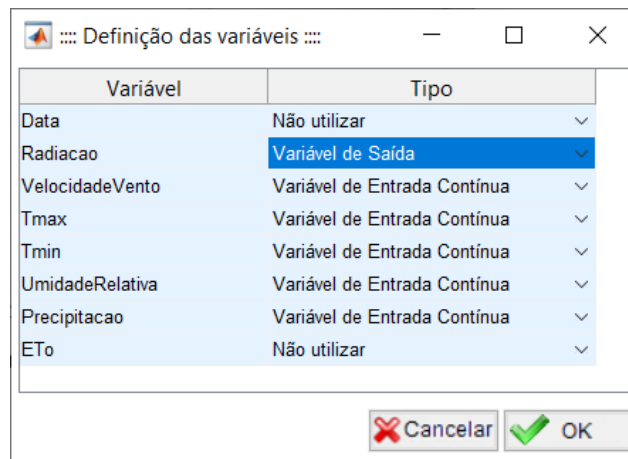


Fig. 16: Definição de variáveis.

Na sequência do processamento, o sistema informa qual os valores mínimos e máximos de cada variável (Figura 17). Essas informações são de extrema importância na fase de utilização da rede neural, uma vez que tais limites devem ser obedecidos.

Dados - Variáveis Contínuas					
	VelocidadeVento	Tmax	Tmin	UmidadeRelativa	Precipitacao
Valor Máximo	2.2000	34.4000	21.2000	95.3000	69.9000
Valor Mínimo	0.2000	15	2.5000	57	0

Dados - Variável Saída	
	Radiacao
Valor Máximo	29.1000
Valor Mínimo	6.4000

Fig. 17: Valor mínimo e máximo das variáveis.

Para treinamento, foram escolhidas redes neurais do tipo MLP, com os seguintes parâmetros de treinamento:

- Quantidade de neurônios da camada intermediária = 7
- Função de ativação da camada intermediária = tangente hiperbólica
- Função de ativação da camada de saída = identidade
- Taxa de aprendizado = 0,7
- Porcentagem de dados para treinamento = 70
- Porcentagem de dados para validação = 15
- Porcentagem de dados para teste = 15
- Número de ciclos = 5000
- Erro médio quadrático = 0,0001

Com essas configurações, foram treinadas 10 redes neurais com algoritmo de treinamento “Resilient backpropagation” e mais 10 redes neurais mudando apenas o algoritmo de treinamento para “Levenberg-Marquardt backpropagation”, totalizando assim 20 redes treinadas.

A partir do relatório sumário, obtemos informações quanto aos erros associados a cada banco de dados definidos aleatoriamente (treinamento, teste e erro), e qual critério de parada cada rede obteve. Uma parte desse relatório pode ser vista na Figura 18.

	Index	Net. name	Training error	Test error	Validation error	Training algorithm
1	RNA 1	MLP 5-7-1	4.3804	7.0310	7.1064	Resilient backpropagation
2	RNA 2	MLP 5-7-1	4.3762	4.4832	16.3794	Resilient backpropagation
3	RNA 3	MLP 5-7-1	3.7494	6.2386	9.7251	Resilient backpropagation
4	RNA 4	MLP 5-7-1	3.8658	7.3057	7.9337	Resilient backpropagation
5	RNA 5	MLP 5-7-1	4.1221	4.3602	9.4990	Resilient backpropagation
6	RNA 6	MLP 5-7-1	4.6247	7.5907	7.4715	Resilient backpropagation
7	RNA 7	MLP 5-7-1	4.0203	9.1588	6.2848	Resilient backpropagation
8	RNA 8	MLP 5-7-1	4.5057	7.1073	6.0034	Resilient backpropagation
9	RNA 9	MLP 5-7-1	4.4941	12.3034	7.3310	Resilient backpropagation
10	RNA 10	MLP 5-7-1	3.8926	7.1939	6.7730	Resilient backpropagation
11	RNA 11	MLP 5-7-1	3.2448	8.3535	11.6724	Levenberg-Marquardt backpropagation
12	RNA 12	MLP 5-7-1	3.1511	7.3155	9.3150	Levenberg-Marquardt backpropagation
13	RNA 13	MLP 5-7-1	4.1299	444.9094	270.9001	Levenberg-Marquardt backpropagation
14	RNA 14	MLP 5-7-1	2.8666	13.3218	7.6057	Levenberg-Marquardt backpropagation
15	RNA 15	MLP 5-7-1	3.3866	8.2015	9.6778	Levenberg-Marquardt backpropagation
16	RNA 16	MLP 5-7-1	3.9017	7.1596	7.8277	Levenberg-Marquardt backpropagation
17	RNA 17	MLP 5-7-1	4.3477	17.2396	6.4507	Levenberg-Marquardt backpropagation
18	RNA 18	MLP 5-7-1	3.2569	47.7903	8.2829	Levenberg-Marquardt backpropagation
19	RNA 19	MLP 5-7-1	3.2074	7.3340	7.7787	Levenberg-Marquardt backpropagation
20	RNA 20	MLP 5-7-1	3.6377	7.0646	7.3645	Levenberg-Marquardt backpropagation

Fig. 18: Erros no treinamento, validação e teste.

Analisando os erros no treinamento, validação e teste, para esta simulação escolhemos a rede neural de número 10 (MLP 10), por apresentar os menores erros nas 3 etapas.

A equação de Garson para essa rede pode ser vista na Figura 19, onde as variáveis de maior valor apresentam maior correlação com a variável de saída, como é o caso da temperatura máxima e da umidade relativa do ar.

Variável	Importância - RNA 10
VelocidadeVento	13.0878
Tmax	32.7434
Tmin	16.8793
UmidadeRelativa	20.9251
Precipitacao	16.3644

Fig. 19: Aplicação do método de Garson na rede MLP 10.

Analisando agora o recurso gráfico que o software apresenta, temos a interface onde o usuário seleciona a rede neural treinada e na sequência é plotado o gráfico onde podemos visualizar a saída desejada e a saída real da rede. Na Figura 20, tal análise é mostrada para a rede MLP 10.

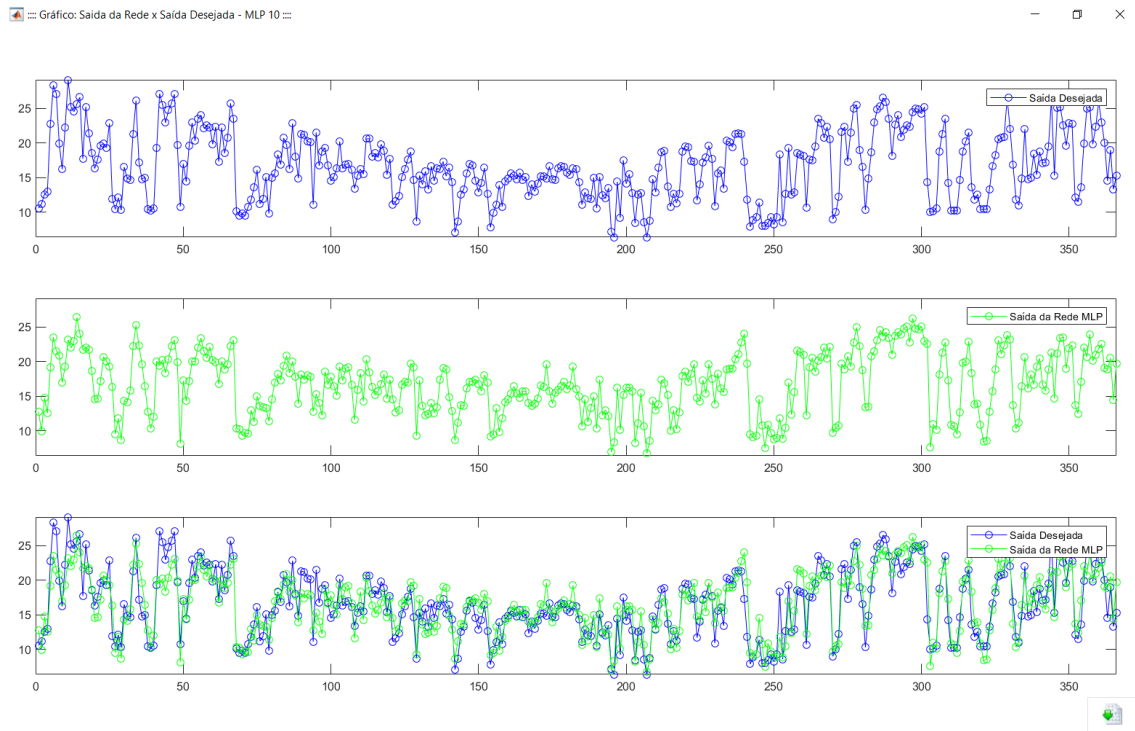


Fig. 20: Análise gráfica da rede MLP 10.

Outro recurso importante é a geração da equação que resolve a chamada “caixa preta”, ou seja, transforma a rede neural em uma única equação, obtendo o modelo matemático que descreve a rede. Na Figura 21, temos a equação para a rede treinada MLP 10.

```

Eqbt x
1 Saida_Rede = 6.4 + ((purelin(
2 tansig( (2*(VelocidadeVento - (0.2))/(2.2 - (0.2)) - 1)*(-0.24932) +
3 (2*(Tmax - (15))/(34.4 - (15)) - 1)*(1.1179) +
4 (2*(Tmin - (2.5))/(21.2 - (2.5)) - 1)*(-1.1275) +
5 (2*(UmidadeRelativa - (57))/(95.3 - (57)) - 1)*(-1.454) +
6 (2*(Precipitacao - (0))/(69.9 - (0)) - 1)*(1.2948)+ (2.0734))* (0.58853) +
7 tansig( (2*(VelocidadeVento - (0.2))/(2.2 - (0.2)) - 1)*(-0.0083389) +
8 (2*(Tmax - (15))/(34.4 - (15)) - 1)*(2.6685) +
9 (2*(Tmin - (2.5))/(21.2 - (2.5)) - 1)*(-0.97765) +
10 (2*(UmidadeRelativa - (57))/(95.3 - (57)) - 1)*(0.27693) +
11 (2*(Precipitacao - (0))/(69.9 - (0)) - 1)*(-0.42405)+ (1.0979))* (-0.56045) +
12 tansig( (2*(VelocidadeVento - (0.2))/(2.2 - (0.2)) - 1)*(-0.021583) +
13 (2*(Tmax - (15))/(34.4 - (15)) - 1)*(-1.6387) +
14 (2*(Tmin - (2.5))/(21.2 - (2.5)) - 1)*(-0.5151) +
15 (2*(UmidadeRelativa - (57))/(95.3 - (57)) - 1)*(0.17241) +
16 (2*(Precipitacao - (0))/(69.9 - (0)) - 1)*(0.8204)+ (-1.0335))* (-0.60667) +
17 tansig( (2*(VelocidadeVento - (0.2))/(2.2 - (0.2)) - 1)*(-1.0328) +
18 (2*(Tmax - (15))/(34.4 - (15)) - 1)*(4.9067) +
19 (2*(Tmin - (2.5))/(21.2 - (2.5)) - 1)*(-1.9607) +
20 (2*(UmidadeRelativa - (57))/(95.3 - (57)) - 1)*(-1.2844) +
21 (2*(Precipitacao - (0))/(69.9 - (0)) - 1)*(-0.52409)+ (0.00020842))* (0.1535) +
22 tansig( (2*(VelocidadeVento - (0.2))/(2.2 - (0.2)) - 1)*(-6.0138) +
23 (2*(Tmax - (15))/(34.4 - (15)) - 1)*(6.145) +
24 (2*(Tmin - (2.5))/(21.2 - (2.5)) - 1)*(1.8133) +
25 (2*(UmidadeRelativa - (57))/(95.3 - (57)) - 1)*(11.1413) +
26 (2*(Precipitacao - (0))/(69.9 - (0)) - 1)*(-2.4064)+ (-0.77847))* (-0.12545) +
27 tansig( (2*(VelocidadeVento - (0.2))/(2.2 - (0.2)) - 1)*(8.0402) +
28 (2*(Tmax - (15))/(34.4 - (15)) - 1)*(-1.602) +
29 (2*(Tmin - (2.5))/(21.2 - (2.5)) - 1)*(-1.1931) +
30 (2*(UmidadeRelativa - (57))/(95.3 - (57)) - 1)*(-2.7255) +
31 (2*(Precipitacao - (0))/(69.9 - (0)) - 1)*(-2.1174)+ (1.4205))* (0.12365) +
32 tansig( (2*(VelocidadeVento - (0.2))/(2.2 - (0.2)) - 1)*(-0.16969) +
33 (2*(Tmax - (15))/(34.4 - (15)) - 1)*(1.0005) +
34 (2*(Tmin - (2.5))/(21.2 - (2.5)) - 1)*(1.087) +
35 (2*(UmidadeRelativa - (57))/(95.3 - (57)) - 1)*(0.75854) +
36 (2*(Precipitacao - (0))/(69.9 - (0)) - 1)*(-1.0378)+ (-2.0514))* (0.54343) +
37 (-0.19462)) + 1)*(29.1 - (6.4))/2

```

Fig. 21: Equação da rede MLP 10.