

UNIVERSIDADE ESTADUAL DE GOIÁS  
CÂMPUS ANÁPOLIS DE CIÊNCIAS EXATAS E TECNOLÓGICAS -  
HENRIQUE SANTILLO  
MESTRADO *STRICTO SENSU* EM ENGENHARIA AGRÍCOLA

**ESTIMATIVA DA PRODUTIVIDADE DA SOJA  
COM REDES NEURAS ARTIFICIAIS**

Guiliano Rangel Alves

ANÁPOLIS – GO  
ABRIL 2016

# **ESTIMATIVA DA PRODUTIVIDADE DA SOJA COM REDES NEURAIAS ARTIFICIAIS**

GUILIANO RANGEL ALVES

Orientador: PROFESSOR D.SC. ITAMAR ROSA TEIXEIRA

Dissertação apresentada à Universidade Estadual de Goiás – UEG, Câmpus Anápolis de Ciências Exatas e Tecnológicas - Henrique Santillo como parte das exigências do Programa de Pós- Graduação *Stricto sensu* em Engenharia Agrícola – Engenharia de Sistemas Agroindustriais, para obtenção do título de MESTRE.

**Anápolis**  
Goiás  
2016

Alves, Guiliano Rangel

Estimativa da produtividade da soja com redes neurais artificiais – 2016.

76 f.il.

Orientador: Prof. D.Sc. Itamar Rosa Teixeira.

Dissertação (Mestrado) – Universidade Estadual de Goiás. Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo. Programa de Pós-Graduação stricto sensu em Engenharia Agrícola, 2016.

Bibliografia.

1. Glycine max L., 2. Características agronômicas, 3. MLP, 4. Perceptron. I. Título.

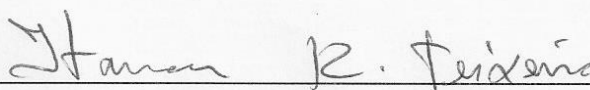
# ESTIMATIVA DA PRODUTIVIDADE DA SOJA COM REDES NEURAIS ARTIFICIAIS

Por

Guiliano Rangel Alves

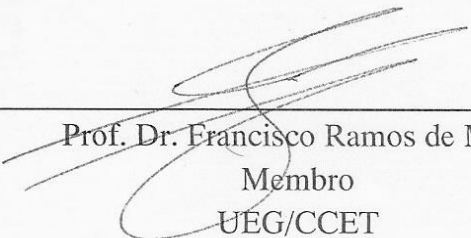
Dissertação apresentada como parte das exigências para obtenção do título de  
MESTRE EM ENGENHARIA AGRÍCOLA

Aprovado em 29/04/2016



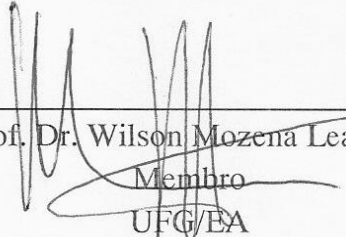
---

Prof. Dr. Itamar Rosa Teixeira  
Orientador  
UEG/CCET



---

Prof. Dr. Francisco Ramos de Melo  
Membro  
UEG/CCET



---

Prof. Dr. Wilson Mozena Leandro  
Membro  
UEG/EA

A minha esposa, Hellen Corrêa da Silva.  
Aos meus filhos, Gustavo e Guilherme.  
Pelo amor, compreensão,  
incentivo e apoio em todos os momentos.

## AGRADECIMENTOS

A Deus, criador e conselheiro, que nunca me deixou cair em desespero.

A Universidade Estadual de Goiás e ao Programa de Pós-Graduação *Stricto sensu* em Engenharia Agrícola, pelo ambiente criativo.

Ao Professor D.Sc. Itamar Rosa Teixeira pela orientação, compreensão.

Ao Professor D.Sc Francisco Ramos de Melo pela amizade e prontidão em me mostrar “a luz no fim do túnel”

Aos professores do Programa de Pós-Graduação *Stricto sensu* em Engenharia Agrícola, em especial aos professores D. Sc. Elton Filho Reis pela atenção, consideração e incentivo.

Ao funcionário Valdeir por auxiliar nas tarefas diárias.

Aos amigos e combatentes, Roberto Bittar, Rogério, Beethoven, por enfrentarem as dificuldades das disciplinas e projetos com companheirismo e altruísmo.

Ao agora mestre Raniele Tadeu Guimarães de Souza por fornecer os dados de sua pesquisa que foram base para o desenvolvimento deste trabalho.

A Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG) pela bolsa de estudo.

## SUMÁRIO

PÁGINA

<b>LISTA DE TABELAS.....</b>	<b>vii</b>
<b>LISTA DE QUADROS .....</b>	<b>viii</b>
<b>LISTA DE FIGURAS .....</b>	<b>ix</b>
<b>RESUMO .....</b>	<b>xi</b>
<b>ABSTRACT .....</b>	<b>xii</b>
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
<b>2. REVISÃO DE LITERATURA .....</b>	<b>3</b>
2.1. A cultura da soja .....	3
2.2. Estimativas de produtividade na soja .....	5
2.3. Aplicabilidade de Redes Neurais Artificiais (RNAs).....	6
2.4. Funcionalidade das Redes Neurais Artificiais (RNA).....	8
2.4.1. Neurônio Artificial .....	8
2.4.2. Funções de ativação do Neurônio Artificial.....	9
2.4.3. Arquitetura de RNA .....	12
2.4.4. Treinamento e aprendizagem .....	15
2.4.5. Tipos de redes.....	16
2.5. Normalização de variáveis.....	22
<b>3. MATERIAL E MÉTODOS .....</b>	<b>24</b>
3.1. Informações gerais.....	24
3.2. Redes Neurais Artificias .....	26
3.2.1. Recursos de informática utilizados .....	26
3.2.2. Treinamento das RNAs .....	27
3.2.3. Categorização e normalização das variáveis .....	27
3.2.4. Desenvolvimento da Rede Neural Artificial (RNA) .....	29
3.2.5. Seleção da rede.....	31
<b>4. RESULTADOS E DISCUSSÃO .....</b>	<b>34</b>
4.1. Análise da estatística descritiva das variáveis de entrada.....	34
4.2. Exploração de configurações para treinamento.....	35

4.3.	Treinamento e seleção da RNA .....	36
4.4.	Rede selecionada .....	44
<b>5.</b>	<b>CONCLUSÕES .....</b>	<b>46</b>
<b>6.</b>	<b>REFERÊNCIAS BIBLIGRÁFICAS .....</b>	<b>47</b>
<b>7.</b>	<b>APÊNDICES .....</b>	<b>52</b>
7.1.	Apêndice A – Programa para importar dados de treinamento.....	52
7.2.	Apêndice B – Programa para treinamento do MLP. ....	54
7.3.	Apêndice C – programa para geração de gráficos .....	57
7.4.	Apêndice D – Tabelas para auxiliar na discussão dos dados. ....	59



**LISTA DE TABELAS**

TABELA 1 - Resultado da análise químico-física do solo da área experimental, na camada de 0,0 - 0,30 cm .....	24
TABELA 2 - Principais características das cultivares testadas no experimento.....	25
TABELA 3 - Valores para identificar habito de crescimento da cultivar .....	27
TABELA 4 - Valores para identificar a densidade populacional para a cultivar.....	28
TABELA 5 - Valores utilizados para Xmin e Xmax por variável .....	28
TABELA 6 - Variáveis apresentadas a camada de neurônio de entrada. ....	29
TABELA 7 - Estatística Descritiva das variáveis utilizadas no treinamento da RNAs .....	34
TABELA 8 - Dados treinamento RNAs .....	37
TABELA 9 - Análise dos erros da rede com 9 neurônio repetição 963.....	44
TABELA 10 - Dados utilizados para treinamento, validação e teste.....	59
TABELA 11 - Erros de treinamento, dados normalizados e em kg Ha <sup>-1</sup> .....	61
TABELA 12 - Erros de validação, dados normalizados e em kg Ha <sup>-1</sup> .....	62
TABELA 13 - Erros de teste, dados normalizados e em kg Ha <sup>-1</sup> .....	62

**LISTA DE QUADROS**

QUADRO 1 - Dados para treinamento em diagnóstico de doenças .....	16
QUADRO 2 - Descrição das informações das colunas do arquivo contendo os dados do treinamento das redes.....	32
QUADRO 3 - Parâmetros utilizado para realizar treinamento da rede MLP.....	36

## LISTA DE FIGURAS

FIGURA 1 – Representação do neurônio de McCulloch e Pitts. Fonte: Braga et al. (2012).....	8
FIGURA 2 - Representação gráfica função de ativação degrau .....	10
FIGURA 3 - Representação gráfica função de ativação degrau bipolar .....	10
FIGURA 4 - Representação gráfica função de ativação rampa .....	11
FIGURA 5 - Representação gráfica função de ativação sigmoide.....	11
FIGURA 6 - Representação gráfica função de ativação tangente hiperbólica.....	12
FIGURA 7 - Representação das camadas de uma rede neural artificial. Fonte: Silva et al. (2010) .....	12
FIGURA 8 – Representação de uma rede <i>Feedforward</i> (camada única). Fonte: Silva et al. (2010).....	13
FIGURA 9 – Representação de uma rede <i>Feedforward</i> (multicamadas). Fonte: Silva et al. (2010).....	14
FIGURA 10 - Representação de uma rede recorrente. Fonte: Silva et al. (2010).....	15
FIGURA 11 - Representação gráfica da fronteira de separação entre as duas classes de saída. Fonte: Silva et al. (2010).....	18
FIGURA 12 - Representação de uma rede do tipo MLP com camadas intermediárias ou ocultas. Fonte: Silva et al. (2010) (modificada) .....	19
FIGURA 13 - Representação gráfica de um neurônio (j) da camada (L) de uma rede Multilayer Perceptron. Fonte: Braga et al. (2012) (modificado) .....	20
FIGURA 14 - Representação geométrica da obtenção do erro. Fonte: Silva et al. (2010) .....	21
FIGURA 15 – Dados transformados no Microsoft Excel. ....	30
FIGURA 16 – Representação gráfica do treinamento da rede com 19 neurônios repetição 806 .....	38
FIGURA 17– Representação gráfica do treinamento da rede com 2 neurônios repetição 212	39
FIGURA 18 – Representação gráfica do comparativo dos valores estimados e observados do conjunto de validação, rede com 1 neurônio repetição 340 .....	40
FIGURA 19 – Representação gráfica do comparativo dos valores estimados e observados do conjunto de validação, rede com 9 neurônio repetição 963 .....	41
FIGURA 20 – Representação gráfica do R linear de treinamento, validação, teste e geral da rede com 9 neurônio treinamento 963.....	42

FIGURA 21 – Representação gráfica do comparativo dos valores observado e estimados de treinamento, rede com 9 neurônios repetição 963 .....	42
FIGURA 22 – Representação gráfica do comparativo dos valores observados e estimados de teste, rede com 9 neurônio repetição 963.....	43
FIGURA 23 – Representação gráfica do comparativo dos valores observado e estimados de teste, rede com 9 neurônio repetição 963 após removidas as amostras 14,29 e 32. ....	43
FIGURA 24 – Histograma de erro em sacas no treinamento (a) e validação (b) .....	45

# ESTIMATIVA DA PRODUTIVIDADE DA SOJA COM REDES NEURAIIS ARTIFICIAIS

## RESUMO

Atualmente, para estimar a produtividade da soja são utilizados modelos estatísticos complexos, que torna restrito o acesso a essa prática. Uma alternativa a estes modelos é a utilização de sistemas computacionais empregando Inteligência Artificial (IA). Nesta linha de sistemas, um caminho é o emprego de Redes Neurais Artificiais (RNA), que possui a capacidade de aprendizagem por meio de exemplos de problemas apresentados. Este trabalho teve por objetivo: avaliar a possibilidade da utilização de RNA do tipo *Multilayer Perceptron* (MLP) para estimar a produtividade da soja baseada nos hábitos de crescimento, densidade de semeadura e características agronômicas; definir os parâmetros relevantes durante o desenvolvimento da RNA para avaliação das características agronômicas e sua relação com a produtividade da soja; desenvolver e selecionar uma arquitetura de RNA para solução do problema proposto. Para realizar o trabalho foram utilizados dados agronômicos da cultura da soja obtidos em experimento conduzido na safra 2013/2014 em Anápolis-GO, cujos os dados foram normalizados em intervalo compatível para trabalho com RNA e em seguida feito o treinamento de várias RNAs para a escolha da RNA com melhor performance. Após o treinamento das redes, foi realizada a análise de performance de cada uma para seleção da RNA com a performance mais adequada ao problema. A RNA selecionada apresentou um índice de acerto de 98% com os dados do treinamento e um acerto de 72% com dados de validação, este nível de acerto pode ser considerado alto, principalmente se for considerada a complexidade de fatores envolvidos na estimativa da produtividade da soja. A aplicação das RNAs do tipo MLP nos dados do experimento conduzido demonstram que é possível estimar a produtividade da soja baseando-se nas características agronômicas, hábito de crescimento e densidade populacional por meio da IA.

**Palavras-chave:** *Glycine max*, características agronômicas, MLP, perceptron.

## ESTIMATE OF PRODUCTIVITY OF SOYBEAN YIELD WITH ARTIFICIAL NEURAL NETWORKS

### ABSTRACT

Nowadays, to estimate soybeans productivity are used complex statistical models, which turns limited the access to this practice. An alternative to these models is use of computer systems applying Artificial Intelligence (AI). On this line of system, an option is apply Artificial Neural Networks (ANN), which has the capacity of learning through examples of problems presented. This work aimed then: evaluate the possibility of using the Multilayer Perception (MLP) ANN to estimate the productivity of soybean based on the growing habits, seeding density and agronomical characteristics; define the relevant parameters during the ANN developing to evaluate the agronomical characteristics and its relation with soybean productivity; develop and choose an ANN architecture to solve the proposed problem. To realize the work were used agronomical data of the soybean culture obtained on experiments leaded on 2013/2014 harvest at Anapolis-GO, which data were normalized on compatible range to work with the ANN and then done the training of the several ANNs, to choose the ANN with best performance. After the network training were realized a performance analysis of each one to select the ANN with the most appropriate answer to the problem. The chosen ANN indicated a success range of 98% with the training data and 72% with the validation data. This can be considered a high level of achievement, mostly if considered the complexity of factors involved on the estimative of the soy productivity. The ANN application of the kind MLP on the conducted experiment data shows that is possible to estimate the soy productivity based on agronomical characteristics, growing habits and seeding density through AI.

**Keywords** *Glycine max*, agronomic characteristics, MLP, Perceptron.

## 1. INTRODUÇÃO

Uma rede neural biológica consiste de neurônios interconectados cuja a ativação é definida por um caminho linear reconhecível. Os neurônios interagem com seus vizinhos por meio dos terminais dos axônios conectados via sinapses para dendritos em outros neurônios (BRAGA *et al.*, 2012; HAYKIN, 2005). O termo redes neurais é usado tanto para indicar a estrutura biológica utilizada pelo cérebro para realizar diversas atividades (reconhecimento de padrões, percepção, controle motor, entre outros), quanto para as Redes Neurais Artificiais (RNAs) utilizadas em atividades computacionais.

As RNAs são uma representação matemática da arquitetura neural do cérebro humano, refletindo suas habilidades de aprendizagem e generalização, por esta razão estão classificadas no campo da ciência que estuda Inteligência Artificial (IA). As RNAs são amplamente aplicadas em pesquisas, por suas habilidades em modelar sistemas altamente não-lineares em que a relação entre as variáveis é desconhecida ou muito complexa (ALEKSANDER; MORTON, 1995; RUSSELL; NORVIG, 2010).

A habilidade de encontrar relação complexa entre variáveis levou a utilização das RNAs em vários campos da ciência, a na área das Ciências Agrárias existem vários trabalhos empregando o uso de RNAs com as mais diversas finalidades a exemplo a engenharia agrícola cuja metodologia foi usada para determinar a resistividade elétrica do solo por meio dos fatores resistividade térmica, tipo do solo e grau de saturação (ERZIN *et al.*, 2010), na zootecnia para avaliar a produção de leite (KOMINAKIS *et al.*, 2002; SHARMA, A. K. *et al.*, 2007; TORRES *et al.*, 2005) e na parte de produção vegetal na qual as RNAs foram utilizadas para prever o rendimento de diversas culturas como milho (KAUL *et al.*, 2005; UNO *et al.*, 2005), trigo (ALVAREZ, 2009; SAFA *et al.*, 2015) e banana (SOARES, J. D. R. *et al.*, 2013). Para a soja não foram encontrados relatos na literatura nacional e internacional sobre estimativa de rendimento por meio de RNAs utilizando características agronômicas.

A soja é tida atualmente como a principal commodity agrícola brasileira, sendo o país o segundo maior produtor mundial desta oleaginosa. Na safra 2014/2015 foram produzidas em torno de 96 milhões de toneladas de grãos no Brasil, sendo a região Centro-Oeste responsável por quase a metade deste montante (CONAB, 2015), com destaque para os estados de Mato Grosso e Goiás.

A diferença de produtividade de uma região em relação a outra advém de vários fatores que influenciam durante o desenvolvimento da cultivar no campo. O potencial de rendimento da soja é determinado geneticamente e quanto deste potencial vai ser atingido depende do efeito

de fatores limitantes que estarão atuando em alguma fase durante o ciclo (COSTA, 1996). O efeito desses fatores pode ser minimizado pela adoção de um conjunto de práticas de manejo que faz com que a comunidade de plantas tenha o melhor aproveitamento possível dos recursos ambientais disponíveis (PIRES *et al.*, 2000).

Dentre os fatores que influenciam o rendimento da soja pode-se citar: espaçamento entre as linhas e a densidade de planta (HEIFFIG *et al.*, 2006; HOLTZ *et al.*, 2014; TOURINO *et al.*, 2002; VENTIMIGLIA *et al.*, 1999), elementos agrometeorológicos como radiação solar, temperatura do ar e precipitação pluvial (BERLATO *et al.*, 1992; FONTANA *et al.*, 2001). A estimativa da produtividade da soja foi realizada com sucesso por Monteiro e Sentelhas (2014) utilizando modelo agrometeorológico, bem como em estudos conduzidos por Martin *et al.* (2012) utilizando modelo de previsão e por Araújo *et al.* (2008) utilizando Modelo de Regressão Espacial Autorregressivo Misto e Modelo do Erro Espacial. Estes trabalhos apresentam uma característica em comum, todos necessitam de informações que devem ser coletadas durante todo o período de cultivo, o que torna o uso dessas metodologias complexas para serem utilizadas pelos produtores (MARTIN, 2007). Mas as características inerentes a própria planta de soja, e que estão relacionadas diretamente ao potencial genético dos materiais, podem afetar a sua adaptação aos mais diversos ambientes, bem como também afetar o rendimento, a exemplo da área foliar produzida (DORAISWAMY *et al.*, 2004) e a sua morfofisiologia responsável pela plasticidade fenotípica da planta (HEIFFIG *et al.*, 2006). Dentre as principais características agronômicas influenciadas pelo comportamento diferenciado entre cada cultivar cita-se: produção de ramos por planta; número de vagens e sementes por planta; número de entrenós; inserção da primeira vagem; diâmetro do caule, altura da planta e obviamente a produtividade de grãos.

O uso das características agronômicas da soja na fase R6 ou posterior torna possível a estimativa da produtividade sem a necessidade de dados anteriores (LEE, C.; HERBEK, 2005), pois nessa fase os grãos já preencheram completamente a cavidade das vagens estando próximo ao que será encontrado na colheita, sofrendo pouco ou nenhuma influência após essa fase.

Assim, este trabalho teve como objetivo geral avaliar a possibilidade do uso de Redes Neurais Artificiais como ferramenta de avaliação das principais características agronômicas de cultivares de soja com distintos hábitos de crescimento, submetidas à diferentes densidades de semeadura, visando obter estimativa de produtividade.



## 2. REVISÃO DE LITERATURA

### 2.1. A cultura da soja

O cultivo da soja (*Glycine max(L.) Merrill*) ocorreu inicialmente no continente asiático, principalmente na região do rio Yangtze, na China (LACLC, 2003). As domesticações da soja remontam entre 3000 a 5000 anos passados (CHUNG *et al.*, 2014; LEE, G.-A. *et al.*, 2011), sendo muito dos relatos escritos em línguas arcaicas. Para a antiga civilização chinesa a importância da soja assim como do trigo, arroz, centeio e o milho tinham tamanha relevância em sua cultura que eram consideradas sagradas, com cerimoniais ritualísticos na época da semeadura e da colheita.

Os EUA, maior produtor de soja na atualidade (USDA, 2016), antes da Segunda Guerra mundial importava mais de 40% de óleos e gorduras comestíveis. A interrupção das rotas comerciais durante a guerra resultou na rápida expansão da área plantada de soja nos EUA, pois o país procurava uma alternativa para as importações. Por volta de 1882 a soja chegou ao Brasil, proveniente do sul dos EUA. Os primeiros estudos com a cultura no país tiveram como responsável o professor Gustavo Dutra, da Escola de Agronomia da Bahia. No Estado de São Paulo o Instituto Agrônomo de Campinas (IAC) iniciou em 1892 os estudos para obtenção de cultivares aptos para a região. O interesse pela cultura, no entanto, não tinha como objetivo a produção de grãos, mas a utilização como forragem durante a rotação de culturas. Os grãos eram dados aos animais e não existia utilização pela indústria (NUNES, 2013).

Mais tarde, no início do século XX, o IAC iniciou a distribuição de sementes. Nesse período o estado do Rio Grande do Sul iniciou o cultivo da soja, que devido à semelhança do clima da região com a do sul dos EUA permitiu a região apresentar as melhores condições para o desenvolvimento da cultura. O registro do primeiro cultivo com a finalidade comercial ocorreu no início dos anos 40, quando o cultivo adquiriu importância econômica, merecendo assim o primeiro registro estatístico nacional. No final da década de 40 a produção já estava em 25.000 toneladas, levando o país a figurar pela primeira vez como produtor de soja nas estatísticas internacionais.

A partir da década de 1960 a soja se estabeleceu como cultura economicamente importante para o Brasil multiplicando por cinco (de 206 mil de toneladas, em 1960, para 1,056 milhões de toneladas em 1969). A região Sul foi responsável por 98% da produção nacional, utilizando principalmente o rodízio entre o trigo, no inverno, e a soja no verão. Mas foi na década seguinte que a soja se consolidou como a principal cultura do agronegócio brasileiro, passando de 1,056 milhões de toneladas (1970) para mais de 15 milhões de toneladas (1979).

Crescimento resultado não apenas do aumento da área cultivada (1,3 para 8,8 milhões de hectares), mas também pela utilização de novas tecnologias, permitindo o aumento expressivo na produtividade (1,14 ton ha<sup>-1</sup> para 1,73 ton ha<sup>-1</sup>). Sendo que 80% do volume produzido ainda se concentrava na Região Sul (EMBRAPA, 2012).

A região do cerrado, na década de 70, era responsável por menos de 2% da produção nacional de soja. Porém, nas décadas de 80 e 90, ocorreu uma grande expansão na região conhecida como polígono dos solos ácidos (Triângulo Mineiro, Mato Grosso do Sul, Mato Grosso, Goiás, Tocantins, sul do Maranhão, sul do Piauí e oeste da Bahia). A partir do ano de 1980 houve um grande crescimento da participação na produção nacional, passando para 20% em 1980, 40% em 1990 e chegando a 60% da produção nacional em 2003. Com esse crescimento, a região do cerrado tornou-se a maior região produtora do país e o estado do Mato Grosso passou de produtor despercebido para líder nacional de produtividade de soja. Tamanho expansão só foi possível devido aos estudos de fertilização dos solos do cerrado, à sua topografia plana e favorável à mecanização, e o desenvolvimento de cultivares aptas à região.

A soja é considerada atualmente como a oleaginosa mais importante comercialmente no mundo. Para a safra 2015/16 a projeção da produção mundial é de 320 milhões de toneladas (USDA, 2016), sendo os EUA os responsáveis por aproximadamente 33% (108,35 milhões de toneladas) desta produção, o que torna o país o maior produtor mundial. O Brasil ocupa o segundo lugar, sendo que na projeção para safra 2015/16 está estimada a obtenção de 100.933,0 milhões toneladas do grão em 33.234,0 milhões de ha, o que representa um incremento de 4,9% e 3,6% respectivamente em relação à safra passada (CONAB, 2016). A produtividade média nacional é de 3.037,0 kg ha<sup>-1</sup>, sendo o estado do Mato Grosso o principal produtor, enquanto o estado de Goiás aparece em quarto lugar no ranque nacional.

Em 2015 o país exportou 56,7 milhões de toneladas de grãos (CONAB, 2016), um montante US\$ 27,96 bilhões deste total, US\$ 20,98 bilhões (75,1%) referem-se às exportações de soja em grão, US\$ 5,82 bilhões (20,8%) às de farelo de soja e US\$ 1,2 bilhão (4,1%) às de óleo de soja. Somados, os três produtos responderam por 14,6% do total das exportações brasileiras realizadas em 2015 (ABIOVE, 2015).

O Brasil destaca-se pela competitividade na produção de soja, que se deve em parte ao investimento em tecnologia realizado pelo país, a exemplos do uso de materiais genéticos mais produtivos, uso de inoculantes como fonte de nitrogênio, cultivo em sistema conservacionistas como plantio direto e integração lavoura-pecuária, adubação equilibrada, população adequada

de plantas, melhorias das táticas de manejo fitossanitário da lavoura, e que contribuiu para elevar os níveis de produtividade.

Apesar do segundo lugar na produção mundial da soja o Brasil precisa melhorar a forma como escoar a produção. O transporte interno, em especial as rodovias apresentam qualidade muito variadas, podendo percorrer distâncias de 2400 quilômetros para atingir pontos de exportação, o que dificulta e encarece a produção da soja, diminuindo a competitividade do País no mercado externo.

## 2.2. Estimativas de produtividade na soja

Estimar a produtividade da soja, enquanto a cultura ainda está de pé no campo pode ser um processo desafiador. É necessária cautela, pois a variabilidade nos componentes do rendimento, tais como população de plantas, sementes por vagem, tamanho da semente, espaçamento entre plantas, pressões causadas por pragas entre outros fatores podem afetar drasticamente a produtividade final. Suposições de número final de vagens, sementes por vagem, e tamanho das sementes podem não refletir com precisão os valores na época da colheita. A melhor estimativa pode ser alcançada na fase R6 (vagens verdes com sementes que enchem a cavidade da vagem) ou mais tarde (LEE, C.; HERBEK, 2005).

As estimativas de componentes de produção devem ser feitas em cinco a dez locais aleatórios em todo o campo para obter uma média melhor do rendimento. Cada um dos locais deve ser típico das áreas circunvizinhas. Amostragem de vários locais no campo ajuda a melhorar a estimativa global de rendimento. Lee e Herbek (2005) consideram que a estimativa da produtividade do campo de soja é tão boa quanto os números colocados na equação (plantas por acre) x (vagens por planta) x (sementes por vagem) ÷ (sementes por linha).

Para a estimativa da produtividade da soja, Martin (2007) considera que se deva primeiro definir o nível de estimação a ser obtido: produtividade genética que é produtividade teórica da cultivar não levando em consideração as perdas devido a interação da planta com o ambiente; produtividade potencial que é a produção de uma variedade cultivada em ambiente ao qual está adaptada, sem limitações edafoclimáticas e nutricionais, livre da ação de pragas e doenças e com os demais estresses efetivamente controlados; produtividade deplecionada, ou ainda, estimativa da produção de grãos, que leva em consideração questões relacionadas com a disponibilidade hídrica e/ou o suprimento de nutrientes, que muitas vezes causa estresse nas plantas de soja e reduz a produtividade total; altas produtividades a campo diz que em algumas

condições especiais, é possível obter-se produções por área bem elevadas, chegando até 8000 kg ha<sup>-1</sup>. Contudo, para que este patamar de produtividade possa ser atingido deve-se maximizar os recursos do meio, utilizando o genótipo mais adaptado para um determinado local.

A produtividade deplecionada é o nível de estimativa que mais se aproxima do rendimento de grãos que o produtor deseja conhecer antes da colheita. Poder determinar com antecedência e precisão a produtividade de uma safra agrícola é uma questão estratégica para produtores (ZACHARIAS *et al.*, 2008). Alguns trabalhos foram conduzidos para estimar a produtividade de grãos de soja. Gomes *et al.* (2014) utilizando um modelo de simulação concluiu ser possível a previsão da produtividade em condições irrigadas, evidenciando as melhores estratégias de irrigação que resultem em elevadas produções de grãos. Monteiro e Sentelhas (2014) utilizando modelo agrometeorológico também comprovaram ser possível estimar a produtividade da soja em diferentes condições climáticas.

Vários outros trabalhos (BERKA *et al.*, 2003; KAUL *et al.*, 2005; MARTIN, 2007; MARTIN *et al.*, 2012; MERCANTE *et al.*, 2010; PRASAD *et al.*, 2006) foram conduzidos para estimar a produtividade da soja, alcançando resultados estatisticamente satisfatórios, mas para alcançar esses resultados muitas vezes é necessário realizar ajustes estatísticos dos coeficientes ou no modelo (ARAÚJO, M. A. *et al.*, 2011) o que torna a utilização destes modelos complexa. Essa dificuldade pode ser superada utilizando-se de Redes Neurais Artificiais (RNAs) que possuem a capacidade de encontrar a relação entre as variáveis apresentadas e o resultado esperado (HAYKIN, 2005), não necessitando assim de conhecimentos de estatística da pessoa que utiliza a RNA.

### 2.3. Aplicabilidade de Redes Neurais Artificiais (RNAs).

As RNAs são amplamente aplicadas nos mais diversos campos da ciência, em grande parte, isso se deve a sua capacidade de aprendizagem e generalização permitindo a modelagem de sistemas altamente não-lineares em que a relação entre as variáveis é desconhecida ou muito complexa (ALEKSANDER; MORTON, 1995; RUSSELL; NORVIG, 2010).

Muitos pesquisadores tem aplicado RNAs em vários cenários para resolver diferentes tipos de problemas, onde não é possível a formulação explícita para o problema (FANG *et al.*, 2000). A maior vantagem da utilização das RNAs está na sua capacidade de aprender com exemplos (dados históricos já conhecidos) para desenvolver uma representação precisa da relação entre os dados apresentados e o resultado esperado.

Entre os benefícios da utilização das RNAs podemos citar a simplicidade da aplicação

e a robustez dos resultados apresentados. Permitindo assim uma abordagem poderosa para aproximar qualquer função de mapeamento de entrada-saída não-linear para qualquer grau de precisão de forma iterativa. O que permite modelar sistemas complexos, apresentando propriedades como: capacidade de aproximação de funções universais, tolerância a dados com ruídos (*outliers*) ou incompletos, suporte a múltiplas variáveis não-lineares com interações desconhecidas e boa capacidade de generalização (HAGAN, M T *et al.*, 2014).

Essas características levaram a utilização das RNAs em vários campos da ciência, por exemplo na medicina onde um experimento foi conduzido durante quatro anos para avaliar os fatores que influenciam na qualidade do atendimento de uma ala médica (SCHIAVO, LO *et al.*, 2015), esse estudo concluiu que as RNAs podem ser utilizadas como uma ferramenta descritiva e preditiva para ajudar a determinar quais os parâmetros influenciam mais a qualidade da ala médica. Ainda na área médica Amato et al. (2013) utilizou RNAs para diagnósticos médicos onde as RNAs se mostram uma ferramenta poderosa com destaque para a capacidade em processar grande quantidade de dados, redução da probabilidade de ignorar informações relevantes e redução do tempo de diagnóstico.

Na área das Ciências Agrárias vários trabalhos foram propostos com as mais diversas finalidades, a exemplo na engenharia agrícola a metodologia foi utilizada para simular a precipitação de um aspersor utilizado para irrigação (MENEZES *et al.*, 2015) onde a RNA apresentou bons resultados na simulação da precipitação com correlação de Spearman entre os dados obtidos em ensaio de campo e os dados simulados variando entre 0,92 e 0,97, sendo que o autor destaca que esse resultado só foi possível devido à alta capacidade de adaptação do algoritmo de treinamento *backpropagation*.

Em outro exemplo de uso de RNAs Erzin et al. (2010) em seu experimento comprovou que as RNAs em especial a do tipo MLP pode ser utilizada para prever a resistividade elétrica do solo por meio da resistividade térmica, tipo do solo e grau de saturação. Concluindo que é possível substituir testes de laboratório pelo uso da rede neural do tipo MLP de forma simples e eficiente.

Ainda na Engenharia Agrícola na parte de produção vegetal as RNAs foram utilizadas para prever o rendimento de diversas culturas como milho (KAUL *et al.*, 2005) onde o autor concluiu que essa metodologia poderia ser utilizada por especialistas em gestão de nutrientes de Maryland para desenvolver, revisar ou atualizar os planos de gestão de nutrientes. Na cultura do trigo Safa et al.(2015) com seu experimento para estimativa da produtividade do trigo comprovou que a capacidade das redes neurais em prever a produção de trigo utilizado dados

heterogêneos é superior a capacidade dos modelos de regressão múltiplas (comumente utilizado em estudos agrícolas).

## 2.4. Funcionalidade das Redes Neurais Artificiais (RNA)

As Redes Neurais Artificiais (RNA), também conhecidas como *conexionismo* ou sistemas de processamento paralelo e distribuído, são estruturas baseadas em modelos que lembram a estrutura do cérebro humano que utilizam unidades de processamento simples (neurônios artificiais) que calculam determinadas funções matemáticas para criar conexões que armazenam o conhecimento das RNAs (BRAGA *et al.*, 2012).

Uma RNA possui a capacidade de adquirir, armazenar e administrar o conhecimento e é definida por Silva *et al.* (2010) como sendo “um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligadas por sinapses artificiais”. Este sistema de interconexões é representado por vetores e matrizes de pesos sinápticos. Estes autores enumeram ainda algumas características das redes neurais tais como: a adaptação por experiência; capacidade de aprendizado; habilidade de generalização; organização dos dados; tolerância a falhas (devido ao elevado número de interconexões), armazenamento distribuído.

### 2.4.1. Neurônio Artificial

A descrição matemática de neurônio artificial, proposto por McCulloch e Pitts (1943), segue um modelo composto por  $n$  entradas (terminais, equivalente aos dendritos) que são responsáveis por receber os valores  $x_1, x_2, \dots, x_n$  e apenas um terminal de saída  $y$  (equivalente ao axônio). O comportamento das sinapses é simulado pela aplicação de pesos ( $w_1, w_2, \dots, w_n$ ) na qual as entradas que podem assumir valores positivos ou negativos, dependendo das sinapses correspondentes serem inibitórias ou excitatórias. A saída só é acionada quando a soma dos impulsos recebidos ( $\Sigma$ ) ultrapassa o limiar de excitação ( $g(u)$ ) (Figura 1).

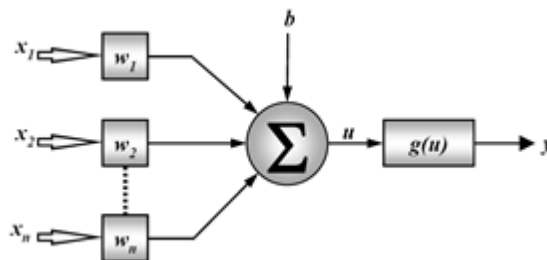


FIGURA 1 – Representação do neurônio de McCulloch e Pitts. Fonte: Braga *et al.* (2012)

O modelo matemático que descreve o comportamento do neurônio artificial é expresso pela Equação 1:

$$y = g \left( \sum_{i=1}^n w_i \cdot x_i + b \right) \quad \text{Eq. ( 1 )}$$

Em que:

- $n$  é o número de entradas do neurônio;
- $w_i$  é o peso associado com a  $i$ -ésima entrada;
- $b$  é o limiar associado ao neurônio;
- $x_i$  é a  $i$ -ésima entrada do neurônio;
- $y$  é a saída do neurônio;

Descrita a equação, um neurônio artificial opera da seguinte forma: a) os sinais são apresentados à entrada; b) cada sinal é multiplicado por um peso que representa o seu nível de relevância na saída da unidade; c) é feita uma soma ponderada dos sinais, resultando assim em um nível de atividade; d) caso este nível de atividade exceda certo limite (*threshold*), então o neurônio produz uma saída.

#### 2.4.2. Funções de ativação do Neurônio Artificial

A função de ativação é responsável por gerar a saída  $y$  do neurônio a partir dos valores de peso  $\mathbf{w} = (w_1, w_2, \dots, w_n)$  e de entrada  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , podendo assumir valores do tipo (HAYKIN, 2005): binários (0 ou 1); bipolares (-1 ou 1) e reais.

Na fase de treinamento das redes neurais artificiais, após o final de cada ciclo de treinamento, pode-se alterar os principais parâmetros da rede de modo a buscar uma configuração de rede eficiente. A função de ativação é um desses parâmetros que pode ser alterado. As principais funções de ativação utilizadas são (BRAGA *et al.*, 2012):

##### 2.4.2.1. Função Degrau (Binária).

A função degrau (binária) (Equação 2) e sua representação gráfica (Figura 2), diz que se o nível de atividade interna do neurônio for um valor positivo ou igual a zero, a saída assumirá o valor 1; caso contrário, assumirá o valor 0.

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases} \quad \text{Eq. ( 2 )}$$

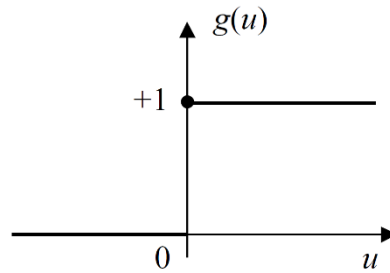


FIGURA 2 - Representação gráfica função de ativação degrau

#### 2.4.2.2. Função Degrau (Bipolar)

A função degrau (bipolar) (Equação 3) e sua representação gráfica (Figura 3) da mesma forma que a função de degrau binária, se o nível de atividade interna do neurônio for positivo ou igual a zero, sua saída assumirá o valor 1; contudo, neste caso, se o nível de atividade do neurônio for um valor negativo, a saída do neurônio assumirá o valor -1.

$$g(u) = \begin{cases} +1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases} \quad \text{Eq. ( 3 )}$$

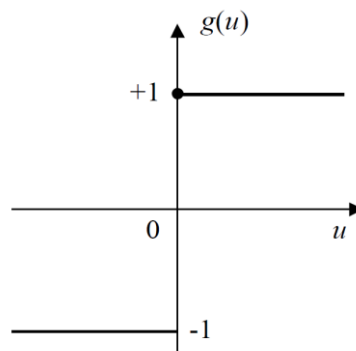


FIGURA 3 - Representação gráfica função de ativação degrau bipolar

#### 2.4.2.3. Função Rampa

A função rampa (Equação 4) e sua representação gráfica (Figura 4), o valor máximo e mínimo da saída do neurônio é +1 e -1. Os valores de saída poderão variar de acordo com a função  $g(u)$  no intervalo definido em  $(-a, a)$ , sendo  $a \neq 0$ .

$$g(u) = \begin{cases} +1, \text{ se } & u \geq a \\ u, \text{ se } & -a < u < a \\ -1, \text{ se } & u \leq -a \end{cases} \quad \text{Eq. ( 4 )}$$



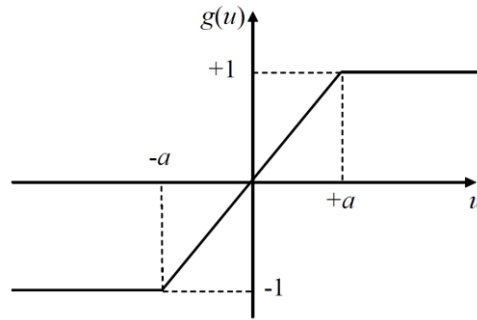


FIGURA 4 - Representação gráfica função de ativação rampa

#### 2.4.2.4. Função Sigmoide

A função sigmoide (Equação 5) e sua representação gráfica (Figura 5), onde o parâmetro  $\beta$  define a suavidade ou grau de inclinação da curva da função sigmoide. A saída no neurônio assumirá valores entre 0 e 1.

$$g(u) = \frac{1}{1 + e^{(-\beta u)}} \quad \text{Eq. ( 5 )}$$

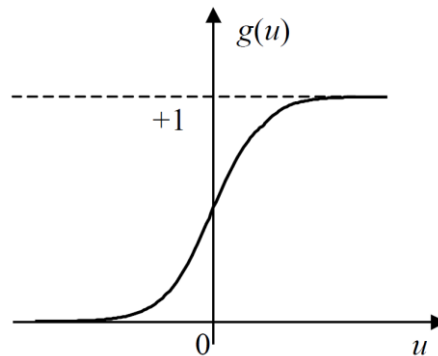


FIGURA 5 - Representação gráfica função de ativação sigmoide

#### 2.4.2.5. Função Tangente Hiperbólica

A função tangente hiperbólica (Equação 6) e sua representação gráfica (Figura 6), utiliza a função tangente hiperbólica, a saída do neurônio assumirá valores positivos e negativos, oscilando no domínio de -1 a 1.

$$g(u) = \tanh(u) = \frac{1 - e^{-u}}{1 + e^{-u}} \quad \text{Eq. ( 6 )}$$

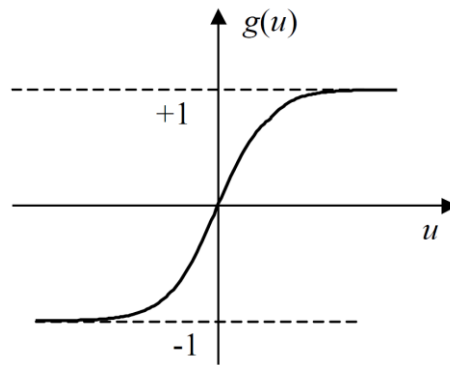


FIGURA 6 - Representação gráfica função de ativação tangente hiperbólica

### 2.4.3. Arquitetura de RNA

A arquitetura de uma rede neural artificial é definida pela forma como os neurônios estão estruturadas. Esta estrutura está intimamente ligada ao algoritmo de aprendizagem que será utilizado para realizar seu treinamento (HAYKIN, 2005). A estrutura de uma rede neural, basicamente, se divide em três partes (Figura 7): camada de entrada - responsável pelo recebimento das informações de entrada, isto é, onde os padrões são inicialmente apresentados à rede; camadas intermediárias ou ocultas - responsáveis pela extração das características do processo ou sistema ao qual se está inferindo, é camada onde praticamente todo o processamento da rede é realizado; camada de saída - responsável pela apresentação do resultado alcançado, o qual foi processado através de todas as camadas da rede.

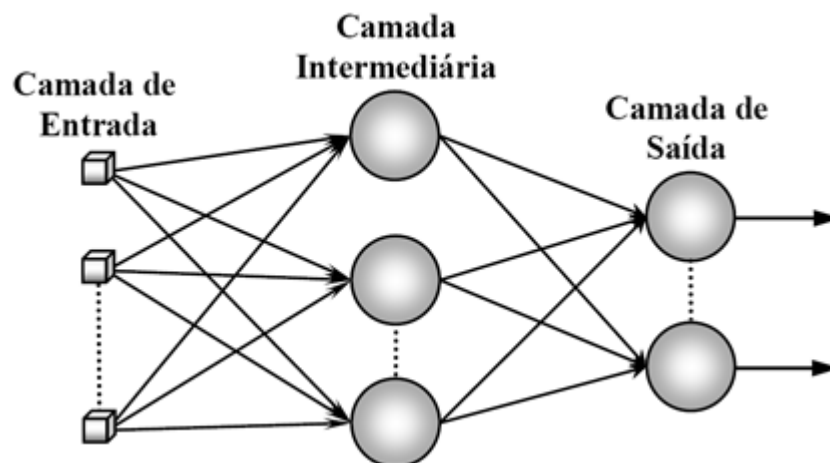


FIGURA 7 - Representação das camadas de uma rede neural artificial. Fonte: Silva et al. (2010)

Em geral pode-se classificar em três tipos de arquiteturas conforme descrito a seguir

### 2.4.3.1. Redes *Feedforward* (Camada Única)

Esta Arquitetura é caracterizada por possuir uma camada de entrada e uma única camada de neurônios que é a própria camada de saída. Esta é a arquitetura utilizada pelo *Perceptron* e ADALINE (HAYKIN, 2005; SILVA, I. N. Da *et al.*, 2010). Ela é aplicada, principalmente, em reconhecimentos de padrões e em memórias associativas sendo composta por  $n$  entradas e  $m$  saídas (Figura 8).

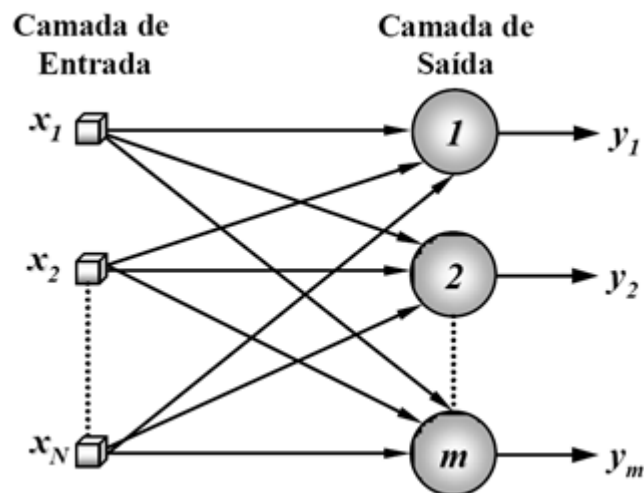


FIGURA 8 – Representação de uma rede *Feedforward* (camada única). Fonte: Silva et al. (2010)

### 2.4.3.2. Redes *Feedforward* (Multicamadas)

Esta arquitetura possui uma ou mais camadas ocultas. As informações são recebidas pela camada de entrada, processadas pela camada oculta, e apresentadas pela camada de saída (HAYKIN, 2005) (Figura 9). Geralmente é aplicada em reconhecimento de padrões e como aproximador universal de funções, pois é capaz de aproximar funções não lineares.

A principal rede que utiliza esta arquitetura é o *Multilayer Perceptron* (MLP).

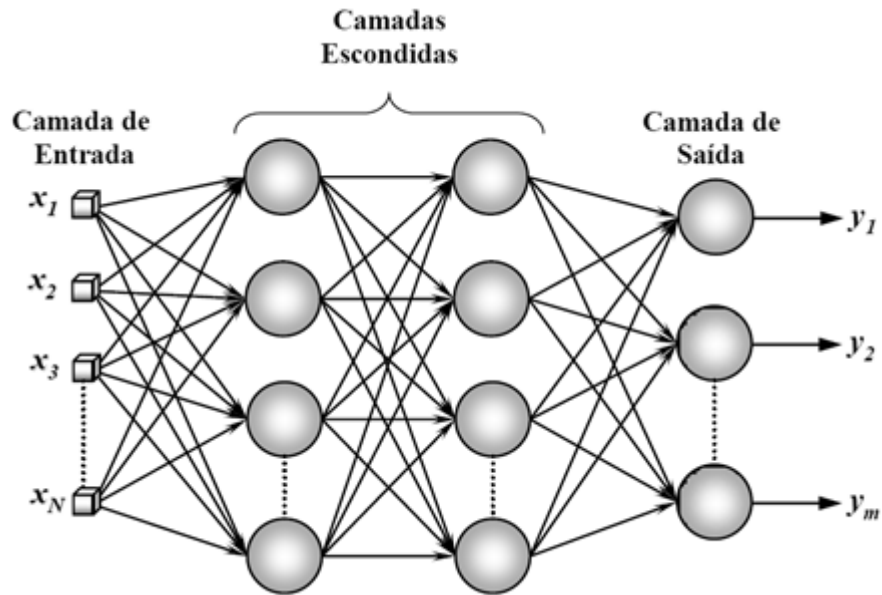


FIGURA 9 – Representação de uma rede *Feedforward*(multicamadas). Fonte: Silva et al. (2010)

### 2.4.3.3. Redes Recorrentes

As redes recorrentes, possuem pelo menos uma realimentação durante o processo de execução da rede. A realimentação gera um impacto significativo na capacidade de aprendizagem da rede e na sua performance (MATSUNAGA, 2012). Uma rede recorrente tem uma camada de entrada de neurônios que alimenta sua saída de volta a entrada de todos os outros neurônios, podendo ou não ter auto-realimentação (HAYKIN, 2005). As conexões de realimentação se originam dos neurônios da camada de saída e possuem uma memória de atraso que operam sobre todas as entradas  $x(n)$  produzindo uma versão atrasada (Figura 10).

Esta arquitetura é utilizada principalmente pelas redes *Perceptron* com realimentação e redes de *Hopfield*, sendo aplicadas em sistemas dinâmicos, séries temporais, previsões, identificação e controle

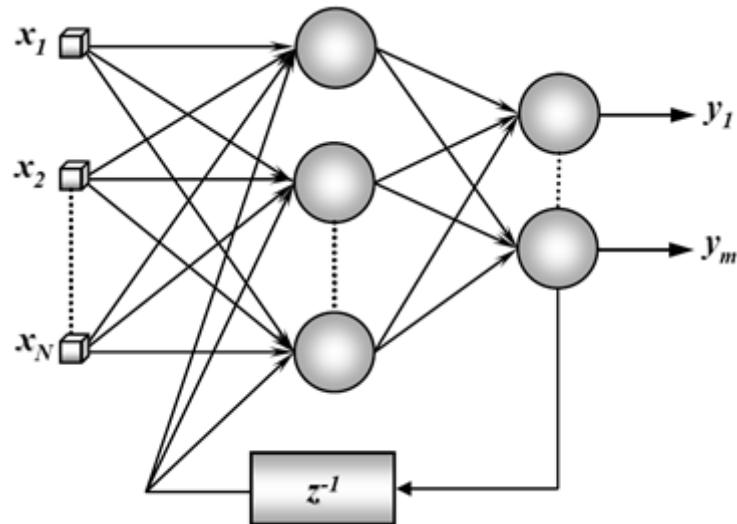


FIGURA 10 - Representação de uma rede recorrente. Fonte: Silva et al. (2010)

#### 2.4.4. Treinamento e aprendizagem

Haykin (2005) apresenta uma RNA como um artefato computacional que possui a capacidade de aprender a partir de exemplos melhorando seu desempenho com o treinamento. A aprendizagem ocorre pelo processo de ajuste dos parâmetros livres da rede neural, que ocorre por meio das estimulações criadas pelos exemplos que são apresentados a rede. O tipo de aprendizagem é determinado pela maneira pela qual os parâmetros são modificados (HAYKIN, 2005).

Silva et al. (2010) explica que após a rede ter encontrado o relacionamento entre as entradas e saídas, ela adquire a capacidade de generalizar essa relação produzindo assim saídas próximas a esperada quando novos dados são apresentados a suas entradas.

Por exemplo, uma rede neural pode ser utilizada para auxiliar no diagnóstico de doenças, utilizando conhecimentos adquiridos previamente. Supondo que uma rede fosse treinada utilizando os dados mostrados no Quadro 1. Estes dados foram obtidos por meio de pesquisa com pacientes com confirmação de diagnóstico por exame. Os dados possuem informação de 7 sintomas (dor de cabeça, náuseas, febre, entre outros) que foram informados pelos pacientes utilizando SIM ou NÃO, a fim de determinar se eles apresentavam estado de gripe ou pneumonia. Após a realização do treinamento da rede com esses dados estará preparada para presumir qual o diagnóstico para pacientes que possuem os sintomas (como exemplo):

- Paciente 1: S, N, S, N, S, N, S;
- Paciente 2: S, N, N, N, N, S, S.

Desta forma, após o treinamento, a rede neural assimila a relação entre os dados apresentados e a resposta esperada, e a partir deste momento passa a produzir uma saída satisfatória.

QUADRO 1 - Dados para treinamento em diagnóstico de doenças

Questionário							Diagnóstico
1	2	3	4	5	6	7	
S	S	S	A	S	S	N	Pneumonia
S	N	S	N	S	S	S	Pneumonia
S	N	N	N	S	N	S	Gripe
S	N	S	S	N	N	N	Gripe

O conjunto de passos para o treinamento da rede é denominado de algoritmo de aprendizagem, segundo Haykin (2005) não existe um algoritmo único para realizar o treinamento. Eles se diferenciam um dos outros pela forma como ocorre o ajuste dos pesos de cada neurônio. O Meio mais simples de ajuste de um peso sináptico  $w(k)$ , no instante de tempo  $k$ , o ajuste  $\Delta w(k)$  é realizado utilizando o valor corrigido  $w(k + 1)$ , na forma  $w(k + 1) = w(k) + \Delta w(k)$ .

O treinamento das redes neurais artificiais pode ser dividido em dois modelos, as que utilizam a aprendizagem supervisionada e as que utilizam aprendizagem não-supervisionada, sendo selecionado o modelo conforme a disponibilidade de resposta esperada ou não (RUSSELL; NORVIG, 2010).

A aprendizagem supervisionada baseia-se no erro entre a resposta esperada e a calculada pela rede, que é utilizado para ajustar os pesos. Se o resultado gerado pela rede for diferente da saída esperada, o algoritmo de treinamento ajusta os pesos, utilizando alguns critérios e procura reduzir o erro a um valor aceitável (BRAGA *et al.*, 2012).

Na aprendizagem não supervisionada não existe um resultado esperado. A rede se organiza buscando encontrar características similares nos subconjuntos que lhe são apresentados, de forma que os pesos da rede são ajustados pelo algoritmo de treinamento para criar sua própria representação das entradas (HAYKIN, 2005). Portanto, quando mais evidente forem as características marcantes dos dados apresentados, mais fácil será a aprendizagem.

#### 2.4.5. Tipos de redes

Na literatura é possível encontrar diversos modelos de rede, cada um proveniente de uma linha de pesquisa diferente e que é mais adequado para a solução de determinados tipos de problema. Na sequência serão detalhadas as duas redes neurais mais conhecidas: *Perceptron*,

que é o precursor no desenvolvimento das redes neurais, e o *Multilayer Perceptron* (MLP) o mais amplamente utilizado. Mas vale ressaltar que existem outros tipos de redes, tais como: Redes de função de base radial, *Adaline*, redes recorrentes de *Hopfield*, redes auto organizáveis de Kohonen (SOM), Redes LVQ (*Learning Vector Quantization*), *counter-propagation* e redes ART (*Adaptive Resonance Theory*) (RUSSELL; NORVIG, 2010).

#### 2.4.5.1. *Perceptron*

Uma rede *Perceptron* é a configuração mais simples de uma rede neural artificial, constituída por apenas uma camada neural, que por sua vez possui somente um neurônio. Apesar da sua simplicidade, esta rede atraiu vários pesquisadores que estudavam sobre inteligência artificial na época (SILVA, I. N. Da *et al.*, 2010).

*Perceptron* é representada pela arquitetura *feedforward*, pois as informações fluem da entrada para a saída de forma unidirecional (HAYKIN, 2005).

Seguindo seu princípio de funcionamento, cada uma das entradas ( $X_i$ ) é ponderada pelos pesos das sinapses ( $w_i$ ) para determinar a importância atribuída a entrada do neurônio em questão. O valor resultante é adicionado ao limiar de ativação ( $\theta$ ) que é repassado como argumento para função de ativação ( $g(u)$ ) cujo resultado é a saída ( $y$ ) (Figura 1).

Em geral a função de ativação utilizada na rede *Perceptron* é do tipo degrau ou degrau bipolar. Podendo assumir as saídas 0 ou 1 para função degrau e 1 ou -1 para função degrau bipolar.

Este tipo de rede é geralmente utilizado em problemas de reconhecimento de padrões com o objetivo de dividir as classes utilizadas na entrada, que sejam linearmente separáveis. Um *Perceptron* constituído por duas entradas a separação das classes pode ser realizada com uma reta, se a função de ativação utilizada for do tipo degrau bipolar, a reta irá separar os elementos apresentados da classe cuja saída for -1 (que estarão abaixo da reta) da classe cuja saída for 1 (que estarão acima da reta) (Figura 11).

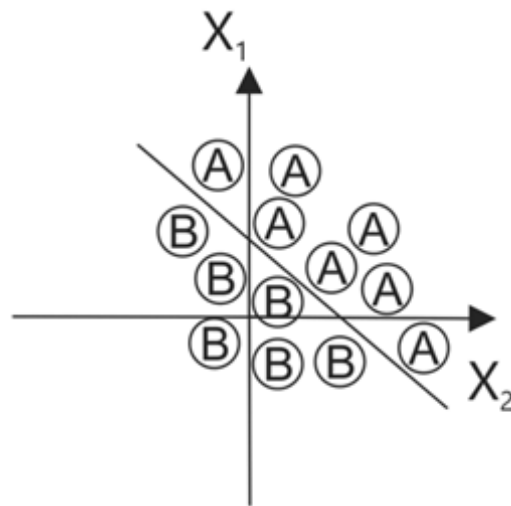


FIGURA 11 - Representação gráfica da fronteira de separação entre as duas classes de saída.  
Fonte: Silva et al. (2010)

#### 2.4.5.2. *Multilayer Perceptron*

As redes *Multilayer Perceptron* ou de retropropagação do erro possuem como característica principal ter no mínimo uma camada oculta de neurônios. Este tipo de rede se destaca por ser amplamente utilizada em diversas áreas do conhecimento, sendo considerada uma das mais versáteis quanto à aplicabilidade (BRAGA *et al.*, 2012).

Este tipo de rede implementa a arquitetura *feedforward* onde os dados fluem da camada de entrada para camada de saída. Os estímulos (dados) são apresentados à rede na camada de entrada. A camada intermediária (oculta) por sua vez extrai as informações e as codificam por meio dos pesos sinápticos, formando assim sua representação interna do ambiente externo. Por fim a camada de saída recebe os estímulos da camada intermediária, mapeando o padrão de resposta que será a saída da rede (Figura 12).



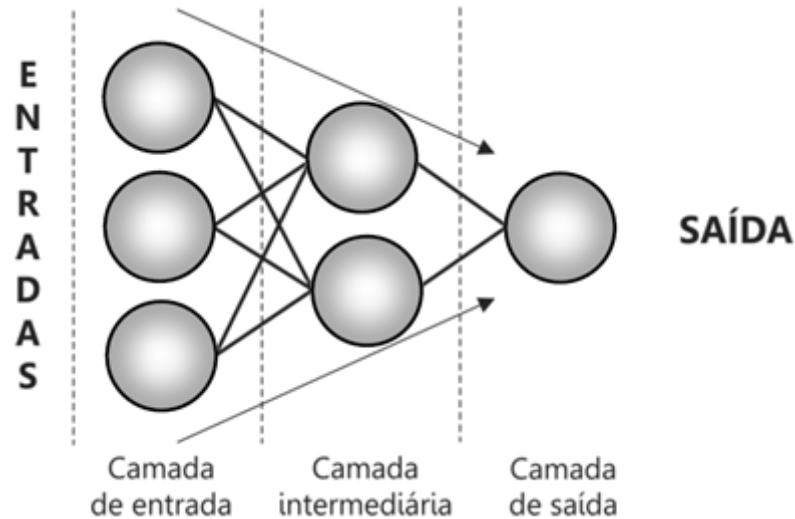


FIGURA 12 - Representação de uma rede do tipo MLP com camadas intermediárias ou ocultas. Fonte: Silva et al. (2010) (modificada)

O treinamento deste tipo de rede é realizado utilizando-se a regra Delta Generalizada, sendo o algoritmo mais empregado conhecido por *backpropagation* ou algoritmo de retropropagação do erro por meio do treinamento supervisionado (HAYKIN, 2005). O processo de treinamento é realizado em duas fases bem definidas de aprendizagem (BRAGA *et al.*, 2012; HAYKIN, 2005; SILVA, I. N. Da *et al.*, 2010).

A primeira fase é denominada propagação adiante ou *forward*. Nesta fase os pesos sinápticos e os limiares dos seus neurônios não sofrem alteração, pois o objetivo é obter o resultado da rede levando em consideração os valores dos parâmetros atuais da rede, que posteriormente serão ajustados pelo processo. Assim um conjunto de amostras de treinamento é apresentado a camada de entrada e seus efeitos se propagam até a camada de saída.

Utilizando o treinamento supervisionado, as respostas da rede são comparadas com as respostas esperadas e os erros gerados pela diferença das saídas são calculados e na fase seguinte utilizados para ajustar os pesos e limiares de todos os neurônios. Desta forma, os pesos das sinapses são alterados de acordo com a regra de correção do erro (HAYKIN, 2005), ou seja, a resposta real é subtraída da resposta esperada para produzir um sinal de erro.

A segunda fase é denominada propagação reversa, retropropagação ou *backward*, é onde o sinal de erro é propagado para trás através da rede. Assim os pesos são ajustados utilizando os valores de erros produzidos na etapa anterior. As execuções sucessivas das duas fases, faz com que os pesos sinápticos sejam ajustados automaticamente em cada iteração. Como consequência ocorre a diminuição dos erros produzidos pelas respostas calculados em relação

as esperadas. Estatisticamente, os pesos sinápticos são ajustados de forma que a resposta da rede se mova para mais próximo da resposta esperada (HAYKIN, 2005).

Vale destacar que o sistema de interconexões utilizados na rede é representado por vetores e matrizes de dados de entrada e pesos sinápticos e, geralmente as funções de ativação utilizadas no método *backpropagation* são a logística ou a tangente hiperbólica.

De forma mais detalhada, tem-se um neurônio ( $j$ ) (Figura 13), pertencente à camada ( $L$ ), e cuja função de ativação ( $g$ ) deve ser contínua e diferenciável em torno do seu domínio. Os sinais de entrada representados por ( $X_1, X_2, \dots X_n$ ) e os pesos sinápticos por ( $W_1, W_2, \dots W_n$ ), e a soma ponderada das entradas ( $\Sigma$ ) gera a saída do corpo celular artificial ( $u$ ) após a introdução do limiar de ativação ( $\theta$ ).

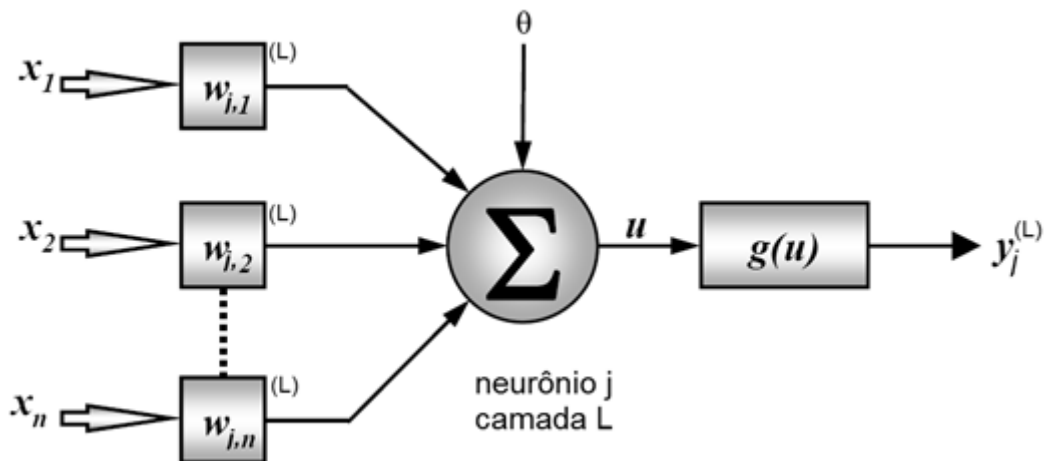


FIGURA 13 - Representação gráfica de um neurônio ( $j$ ) da camada ( $L$ ) de uma rede Multilayer Perceptron. Fonte: Braga et al. (2012) (modificado)

Analisando apenas um neurônio ( $j$ ) (Figura 13), temos a Equação 7 para determinar o erro, cuja atribuição é determinar o desvio entre a saída calculada e o valor esperado.

$$e_j(n) = d_j(n) - y_i(n) \quad \text{Eq. (7)}$$

Em que:

$d_j(n)$  - é o respectivo valor desejado e  $n$  é a  $n$ -ésima interação.

$e_j(n)$  - é o respectivo erro e  $n$  é a  $n$ -ésima interação.

$y_j(n)$  - é o respectivo a saída calculada e  $n$  é a  $n$ -ésima interação

Para determinar o erro total de todos os neurônios da camada de saída aplica-se a Equação 8.

$$E(n) = \frac{1}{2} \sum_{j \in C}^{n_3} (d_j(n) - y_j(n))^2 \quad \text{Eq. (8)}$$

Em que:

$E(n)$  - é o respectivo valor do erro total da e  $n$  é a  $n$ -ésima interação.

$d_j(n)$  - é o respectivo valor desejado e  $n$  é a  $n$ -ésima interação.

$e_j(n)$  - é o respectivo erro e  $n$  é a  $n$ -ésima interação.

$y_j(n)$  - é o respectivo a saída calculada e  $n$  é a  $n$ -ésima interação

O ajuste  $\Delta w_j(n)$  a ser realizado no peso sináptico  $w_j(n)$ , a partir do valor da saída da interação, deve ser proporcional à derivada parcial entre o erro total e o peso sináptico utilizando anteriormente (Equação 9), e  $\eta$  é o parâmetro da taxa de aprendizagem do algoritmo e o sinal negativo indica a descida do gradiente no espaço de pesos em procura do menor valor possível para o erro quadrático.

$$\Delta w_j(n) = -\eta \frac{\partial E(n)}{\partial w_j(n)} \quad \text{Eq. (9)}$$

Em que:

$E(n)$  - é o respectivo valor do erro total da e  $n$  é a  $n$ -ésima interação.

$w_j(n)$  - é o respectivo valor do peso e  $n$  é a  $n$ -ésima interação.

$\eta$  - é o valor da taxa de aprendizagem do algoritmo

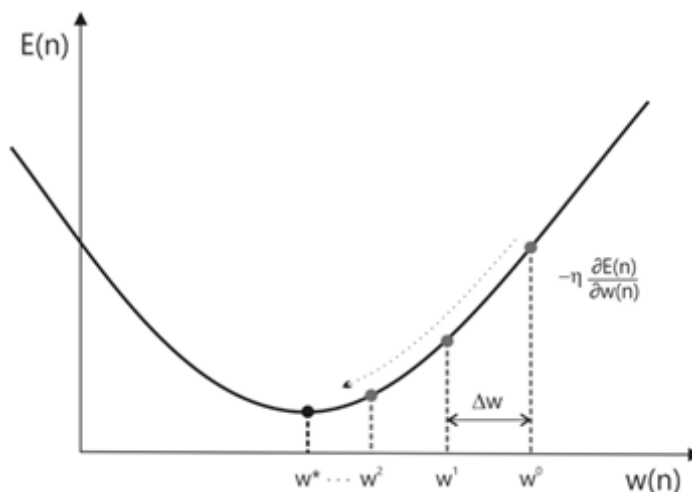


FIGURA 14 - Representação geométrica da obtenção do erro. Fonte: Silva et al. (2010)

Silva et al. (2010) explica por meio do gráfico (Figura 14) que a convergência para o menor erro possível ( $w^*$ ) que corresponde a representação geométrica das etapas para ajustar os pesos rumo a otimização. Iniciando com  $w^0$ , o valor de  $w^1$  é obtido a partir da direção oposta do vetor de gradiente em relação ao valor da etapa anterior, neste caso,  $w^0$ . Quando o processo converge para  $w^*$ , o valor de  $E$  é o menor possível em relação a qualquer outro calculado nas etapas anteriores, em um total de  $n$  iterações.

Para o ajuste dos pesos das camadas intermediárias (ocultas) não existe uma forma de determinar diretamente o valor esperado da saída dos neurônios destas camadas. Todavia, os ajustes são realizados por meio dos erros da saída produzida pelos neurônios da camada imediatamente posterior, que já foram ajustados. Desta forma, se faz necessário realizar o ajuste da camada de saída que servirá como estimativa do erro para camada anterior, e assim por diante.

Este processo de ajustar os pesos de cada neurônio é realizado a cada iteração o que leva a rede a convergir muito lentamente, demandando assim, um grande esforço computacional. Buscando contornar esse inconveniente, várias técnicas têm sido incorporadas ao algoritmo *backpropagation* na busca por melhorar o seu tempo de convergência e diminuir o custo computacional requerido. Entre as técnicas de otimização mais utilizadas destacam-se: método de inserção do termo *momentum*, método *resilient-propagation* e o método Levenberg-Marquardt (HAGAN, Martin T.; MENHAJ, 1994), técnica que foi utilizada na condução do treinamento neste trabalho.

## 2.5. Normalização de variáveis

As redes neurais trabalham ponderando os valores das entradas, assim uma variável que tenha valores mais elevados pode ser considerada, no início do treinamento, como mais relevante que outra variável de menor grandeza. No entanto com treinamento a rede ajustará os pesos para que a variáveis tenha o valor mais adequado conforme sua importância no problema considerado. O tratamento de variáveis com diferentes grandezas pode ser simplificado submetendo-se as variáveis à normalização para uma escala comum com valores entre -1 e 1 (ZARIFNESHAT, S. *et al.*, 2012).

As variáveis podem ter grandezas diferentes, o que pode dificultar ou mesmo impedir que a RNA venha a convergir (encontrar a relação entre as variáveis de entrada e a saída esperada); além da questão das grandezas diferentes também deve ser tratado o caso de variáveis que não são valores numéricos, mas categorias.

Essas variáveis de categoria poderiam simplesmente serem convertidas em valores numéricos, considere que a variável habito de crescimento fosse tratada da seguinte forma: atribuído -1 para crescimento determinado, 0 para crescimento indeterminado e 1 para crescimento indeterminado. Ao se fazer isso, pressupõe-se para a RNA que o habito de crescimento semi-determinado tem maior influência na produtividade da soja, o que não é necessariamente verdade. Assim para as variáveis que definem categoria/tipo é recomendado seguir o mesmo tratamento dado a variáveis tipo *dummy* da Análise de Regressão Múltipla (BOHL *et al.*, 2015; SHARMA, A. K. *et al.*, 2007), que diz que deve ser criado n-1 variáveis, onde “n” é número de categorias que as variáveis podem assumir.

### 3. MATERIAL E MÉTODOS

#### 3.1. Informações gerais

O desenvolvimento de uma Rede Neural Artificial (RNA) do tipo *Multilayer Perceptron* (MLP) necessita de treinamento supervisionado para ajustar os pesos das sinapses. Dessa forma na criação do MLP utilizado neste trabalho para estimativa da produtividade da soja foi utilizado os dados da dissertação do aluno Raniele Tadeu Guimarães de Souza<sup>1</sup> defendida junto ao Programa de Mestrado em Engenharia Agrícola da Universidade Estadual de Goiás. O referido experimento foi conduzido na safra 2013/2014 em área experimental pertencente à Emater-GO, unidade Anápolis-GO (48°18'23'' W 16°19'44'' S, com altitude de 1.017 m). O clima predominante é tipo AW tropical úmido, segundo a classificação de Köppen, caracterizado pelo inverno seco e verão chuvoso.

O solo da área experimental classificado como Latossolo Vermelho distrófico foi amostrado na camada de 0,0 – 0,30 cm, cujos resultados das análises química e físicas são apresentados abaixo na Tabela 1.

TABELA 1 - Resultado da análise químico-física do solo da área experimental, na camada de 0,0 - 0,30 cm

pH	M.O.	P	K	Ca	Mg	Al	H+Al	SB	T
(CaCl <sub>2</sub> )	(g dm <sup>-3</sup> )	(mg dm <sup>-3</sup> )				cmol c dm <sup>-3</sup>			
5,5	31	4,3	0,09	2,7	0,8	00	2,7	57,21	6,32

Capacidade de troca cátions: 6,32. P, K, Fe, Zn, Mn, Cu: extrator Mehlich<sup>1</sup>. . Ca<sup>2+</sup>, Mg<sup>2+</sup>, Al<sup>3+</sup>: KCl 1 mol L<sup>-1</sup>. H + AL: método Ca (OAc)<sup>2</sup> 0,5 mol L<sup>-1</sup>, pH 7. Matéria orgânica: C orgânico x 1,724 – Walkley Black.

O experimento foi implantado seguindo o delineamento inteiramente casualizado em esquema fatorial 3 x 3, com oito repetições. Os tratamentos foram constituídos de três cultivares de soja com diferentes hábitos e tipos de crescimento (BRS Valiosa RR, BMX Potencia RR e NA 7337RR) e três densidades de semeadura (D1: 245.000 plantas ha<sup>-1</sup>; D2: 350.000 plantas ha<sup>-1</sup>; e D3: 455.000 plantas ha<sup>-1</sup>).

Foram utilizadas três cultivares distintas em relação ao hábito de crescimento, todas recomendadas e bastante utilizadas na região Centro-Oeste e no Estado de Goiás (Tabela 2), conforme área inscrita para produção de sementes no estado de Goiás (MAPA, 2009):

<sup>1</sup> Dissertação: SOUZA, R. T. G. De. Espectro de gotas e deposição de solução aquosa em diferentes sistemas de semeadura da soja. 2014. 67p. Universidade Estadual de Goiás, 2014.

TABELA 2 - Principais características das cultivares testadas no experimento.

Características	BRS Valiosa RR	BMX Potência RR	NA 7337 RR
Ciclo (dias)	115 a 140	106 a 114	109 a 128
Ciclo (classificação)	8.1	6.7	7.3
Altura de planta	0,71 m	Alto	0,75-0,97m
Hábito de crescimento	Determinado	Indeterminado	Semi-determinado
Acamamento	Resistente	Resistente	Resistente
Cor do hipocótilo	Roxa	Verde	-
Cor da flor	Roxa	Branca	Branca
Cor da pubescência	Marrom	Cinza	Cinza
Cor do tegumento	Marrom claro	Marrom claro	-
Cor do Hilo	Preta	Marrom clara	Marrom claro
Peso médio (100 sementes)	15,2 g	16,8 g	-
Época de semeadura	15/10 a 10/12	15/10 a 10/12	20/ a 20/11
Reação cancro da haste	Resistente	Resistente	Resistente
Reação mancha "olho-de-rã"	Resistente	M. resistente	Resistente
Oídio	Moderadamente resistente	-	-
<i>Meloidogyne incognita</i>	Moderadamente tolerante	Susceptível	-
<i>Meloidogyne javanica</i>	Tolerante	Moderadamente resistente	-
<i>Heterodera glycines</i>	Susceptível	-	Susceptível

Fonte: Embrapa (2015), Brasmax (2015), Nidera Sementes (2015).

As parcelas foram constituídas por cinco fileiras, de 5 m de comprimento, espaçadas de 0,45 m. A área útil foi constituída pelas três linhas centrais, desconsiderando 1 m de cada extremidade.

O solo foi preparado convencionalmente, utilizando-se arado de aiveca e uma gradagem leve com grade destorroadora-niveladora em toda área para incorporação de restos culturais de plantas milho. Na área das parcelas inicialmente foi realizada sulcagem manual, com profundidade de aproximadamente 0,10 m, e em seguida foi feita a adubação e distribuição da semente na linha a 0,05 m de profundidade. Entre o adubo e a semente foi depositado camada de solo de 0,05 m, evitando com isso o contato direto entre semente e adubo.

Antes da distribuição das sementes foi realizado tratamento de sementes com fungicida carbendazim (15%) + tiram (35%) na dose de 200 mL para 100 kg de sementes. Posteriormente as sementes foram inoculadas com bactéria do tipo *Bradyrhizobium elkanii* (semia 587) e *Bradyrhizobium japonicum* (semia 5079), com concentração bacteriana de  $5,0 \times 10^9$  UFC por mL, na dose de 150 mL do produto comercial para 50 kg de sementes. A distribuição das sementes foi realizada manualmente no sulco, com utilização de fita de poliéster, devidamente marcada para delimitar a distância entre sementes.

A adubação de semeadura foi realizada de acordo com a recomendação para cultura da soja para obtenção da produtividade de 3.000 kg ha<sup>-1</sup> (EMBRAPA, 2015). Aplicou-se a quantidade de 400 kg ha<sup>-1</sup> da formula NPK 04-30-16, correspondendo a 16 kg de nitrogênio, 120 kg de P<sub>2</sub>O<sub>5</sub> e 64 kg de K<sub>2</sub>O, tendo como fontes a ureia, superfosfato simples e cloreto de potássio, respectivamente. Aplicou-se ainda, aos 30 dias após semeadura 24 kg ha<sup>-1</sup> de K<sub>2</sub>O, tendo como fonte o cloreto de potássio

Por ocasião da colheita foram avaliadas as seguintes características agronômicas (variáveis), empregando 10 plantas na área útil de cada parcela: altura de planta - através da utilização de régua graduada foi determinada a altura da planta, obedecendo a distância entre a superfície do solo e a extremidade apical da haste principal, sendo obtido o valor médio da altura das 10 plantas analisadas; número de ramos por planta - resultado da contagem do número de ramos inseridos na haste principal em cada uma das 10 plantas e obtenção da média do número de ramificações por planta; número de vagens por planta - contagem direta do total de vagens formadas em cada uma das 10 plantas e obtenção da média de vagens por planta; número de grãos por vagem - na ocasião da contagem de vagens das 10 plantas na área útil foi determinada a contagem de grãos em razão do número total de vagens, sendo definido assim o número médio de grãos por vagem; massa de 1000 sementes (PMS) - foi determinada segundo metodologia de (MAPA, 2009), onde foram contados oito subamostras de 100 sementes por parcela, cuja massa foi determinada em balança analítica de 0,001 gramas de sensibilidade; rendimento de grãos - a produtividade foi determinada em kg ha<sup>-1</sup> obtido pela trilhagem total de todas as plantas da parcela. O teor de água da massa de sementes foi ajustado para 13% b.u.

## 3.2. Redes Neurais Artificiais

### 3.2.1. Recursos de informática utilizados

Foi utilizado um computador com processador Intel® Core™ i7-4510U 2GHz, 8GB de RAM e HD SSD de 256 GB e Sistema Operacional Windows 10 64bits. Foi criado um programa para realizar o treinamento e seleção das RNAs.

Para auxiliar na preparação dos dados foi utilizado o *Microsoft Excel* versão 2013 para fazer o pré-processamento dos dados de entrada (Normalização dos dados).



### 3.2.2. Treinamento das RNAs

Para realizar o treinamento das RNAs foram selecionadas as variáveis independentes (cultivares com distintos hábitos de crescimento e densidade populacional), que são propriamente os tratamentos testados, e as variáveis dependentes a exemplo das características agrônômicas: altura da planta(A), número de ramos por planta(R), número de vargens por planta(V), número de grãos por vagem(G), massa de 1000 sementes(PMS) e a produtividade de grãos (Prod kg ha<sup>-1</sup>) obtidas conforme descrito no Tópico 3.1. Essas variáveis foram normalizadas com o objetivo de equalizar os dados de entrada da RNA (JOSÉ *et al.*, 2015), buscando-se assim equalizar o peso inicial que a variável possa assumir no início do treinamento.

### 3.2.3. Categorização e normalização das variáveis

Como exposto anteriormente as variáveis podem ter grandezas diferentes, o que pode dificultar ou mesmo impedir que a RNA venha a convergir (quando a RNA atingir a performance definida como aceitável), além da questão das grandezas diferentes também deve ser tratado o caso de variáveis que não são valores numéricos mais categorias, como as variáveis de hábito de crescimento e densidade populacional, classificadas como independentes. Para estas duas últimas variáveis é recomendado seguir o mesmo tratamento dado a variáveis tipo *dummy* da Análise de Regressão Múltipla (BOHL *et al.*, 2015; SHARMA, A. K. *et al.*, 2007). Desta maneira foi definido a variável C2 e C3 para o hábito de crescimento, D2 e D3 para densidade populacional, sendo que C2, C3, D2 e D3 podem assumir os valores 0 ou 1.

Na Tabela 3 observa-se como identificar o habito de crescimento utilizando as variáveis C2 e C3, em que as duas variáveis assumem o valor 0 quando o habito de crescimento considerado é o Determinado. Na Tabela 4 estão as variáveis utilizadas para representar a densidade populacional, em que as variáveis D2 e D3 assumirem o valor 0 quando a densidade populacional considerada é a Recomendada.

TABELA 3 - Valores para identificar habito de crescimento da cultivar

Hábitos de Crescimento	C2	C3
Determinado	0	0
Indeterminado	1	0
Semi-determinado	0	1

TABELA 4 - Valores para identificar a densidade populacional para a cultivar.

Densidade Populacional	D2	D3
Densidade recomendada	0	0
Densidade 50% abaixo da recomendada	1	0
Densidade 50% acima da recomendada	0	1

Para as demais variáveis de entrada, ou seja, as dependentes, que estão compreendidas dentro dos números reais, foi utilizada a transformação linear descrita pela Equação 10. (SRINIVASAN *et al.*, 1994)

$$scaledX = \frac{(X - X_{min})(d2 - d1)}{X_{max} - X_{min}} + d1 \quad \text{Eq. ( 10 )}$$

Em que:

scaledX - valor transformado

X - valor a ser transformado

$X_{max}$  - valor máximo possível para a variável

$X_{min}$  - valor mínimo possível para a variável

d1 - valor do limite inferior do valor convertido (no estudo -1)

d2 - valor do limite superior do valor convertido (no estudo 1)

Os valores de  $X_{max}$  e  $X_{min}$  utilizados na transformação das variáveis altura (A), número de ramos por planta (R), número de vagens por planta (V), número de grãos por vagem (G), massa de 1000 sementes (PMS) e produtividade de grãos (Prod) são apresentados na Tabela 5.

TABELA 5 - Valores utilizados para Xmin e Xmax por variável

Variável	Unidade	$X_{min}$	$X_{max}$
Altura ( A )	cm	20	200
Número de ramos por planta ( R )	un	1	15
Número de vagens por planta ( V )	un	10	150
Número de grãos por vagem ( G )	un	1	4
Massa de 1000 sementes ( PMS )	kg	5	40
Produtividade ( Prod )	kg ha <sup>-1</sup>	1000	6000

Os valores utilizados como entrada e resultado esperado foram trabalhados com valores normalizados entre -1 e 1, no entanto após o treinamento e validação das redes o valor resultante da rede foi retornado a sua grandeza original, para realizar a transformação do valor normalizado para a sua grandeza original foi utilizado a Equação 11 utilizando os valores mínimos e máximo para cada variável conforme a Tabela 5. A equação foi obtida isolando-se a variável “X” da Equação 10.

$$X = \frac{(scaledX - d2)(X_{max} - X_{min})}{d2 - d1} + X_{min} \quad \text{Eq. ( 11 )}$$

Em que:

- X - valor na grandeza original
- scaledX - valor transformado
- $X_{max}$  - valor máximo possível para a variável
- $X_{min}$  - valor mínimo possível para a variável
- d1 - valor do limite inferior do valor convertido (no estudo -1)
- d2 - valor do limite superior do valor convertido (no estudo 1)

### 3.2.4. Desenvolvimento da Rede Neural Artificial (RNA)

Para o desenvolvimento da RNA inicialmente os dados obtidos a partir do experimento foram tabulados utilizando o Microsoft Excel, e posteriormente os dados das características foram transformados seguindo o exposto no Tópico 3.2.3 obtendo as variáveis que foram utilizadas como entrada para o treinamento.

TABELA 6 - Variáveis apresentadas a camada de neurônio de entrada.

Variável	Transformação	Descrição
C2	Categoria/tipo	Hábito de Crescimento
C3		
D2	Categoria/tipo	Densidade
D3		
A	Transformação linear	Altura da planta
R	Transformação linear	Número de ramos por planta
V	Transformação linear	Número de vagens por planta
G	Transformação linear	Número de grãos por vagem
PMS	Transformação linear	Massa de 1000 sementes

C2, C3: identificação do hábito de crescimento conforme Tabela 3. D2, D3: identificação da densidade conforme Tabela 4. PMS: Massa de 1000 sementes.

Para o início do desenvolvimento da RNA do tipo *Multilayer Perceptron* (MLP) foi definido a camada de entrada, que utilizou um neurônio para cada uma das variáveis de entrada conforme Tabela 6, e a camada de saída contendo um neurônio, representando a produtividade. A camada de entrada e saída foram os primeiros parâmetros definidos para o treinamento do MLP (HAYKIN, 2005), pois sem a definição desses parâmetros não seria possível a criação do arquivo com os dados que foram submetidos ao programa visando realizar o treinamento da rede.

Na estrutura do arquivo gerado pelo Microsoft Excel nota-se que no arquivo contém uma coluna para cada neurônio da camada de entrada (9 colunas) e mais uma coluna para a camada de saída (produtividade de grãos) (Figura 15). Este arquivo foi exportado para o formato *Common Separated Values* – CSV, e para adequação do formato esperado pelo software programa de treinamento foi editado no Notepad++ removendo-se a primeira linha.

	A	B	C	D	E	F	G	H	I	J
1	C2	C3	D2	D3	A	R	V	G	PMS	Produtividade
2	0	0	0	0	-0,01044444	0,02857143	-0,30571429	-0,29351536	-0,28870286	0,43344772
3	0	0	0	0	-0,12155556	0,47142857	0,23714286	-0,25051760	0,33554857	0,57929865
4	0	0	0	0	-0,07844444	0,00000000	-0,24142857	-0,32699419	-0,18180571	0,24086991
5	0	0	1	0	-0,18955556	-0,10000000	-0,03000000	-0,36157467	0,04282857	0,17437394
6	0	0	1	0	-0,11240329	0,00899471	-0,12783069	-0,31481095	-0,03643280	-0,05727004
7	0	0	1	0	-0,09592593	0,00000000	-0,16349206	-0,31172339	-0,07404444	0,03819886
8	0	0	1	0	-0,05172840	0,12698413	-0,19000000	-0,26336832	-0,07808254	0,20278091
9	0	0	0	1	-0,13488889	-0,24285714	-0,20285714	-0,40628166	-0,25700571	0,49858569
10	0	0	0	1	-0,02500000	-0,41428571	-0,57000000	-0,32502078	-0,61761143	-0,21651252
11	0	0	0	1	-0,01866667	-0,10000000	-0,30142857	-0,39219015	-0,38569143	0,27204812

FIGURA 15 – Dados transformados no Microsoft Excel.

Após a importação dos dados que foram separados em dois vetores: uma matriz de entrada contendo as variáveis independentes (C2, C3, D2, D3 A, R, V, G e PMS) e outra matriz com a variáveis dependentes (produtividades de grãos) que é resultado esperado da rede quando um item correspondente da matriz de entrada for apresentado ao MLP. A listagem do programa utilizado para realizar a importação do arquivo CSV contendo os dados de treinamento é apresentada no Apêndice A.

Para construção do programa para o treinamento do MLP foram consideradas as seguintes definições: a) os dados devem ser divididos de forma aleatória em três conjuntos sendo eles treinamento (65%), validação (25%) e teste (10%); b) utilização do algoritmo de treinamento Levenberg-Marquardt por ser um dos mais eficientes algoritmos do tipo *Backpropagation* (SCHIAVO, LO *et al.*, 2015); c) utilização da função de performance o Erro

Quadrático Médio (EQM) do inglês *Mean Squared Error* (MSE) (Equação 12), definindo a condição de parada quando o erro (EQM) atingir valor menor ou igual a 0,0005; d) número máximo de falhas de validação igual a 6; e) possibilitar especificar o número máximo de épocas (ciclos de treinamento), número de redes para treinar, números de neurônios iniciais na camada oculta, número final de neurônio na camada oculta, incremento de neurônios na camada oculta; f) armazenar as informações de cada treinamento (performance, performance de treinamento, performance de validação, performance de teste, R linear de Treinamento, R linear de Validação, R linear de Teste, R linear Geral, correlação de Pearson de Treinamento, correlação de Pearson de validação, correlação de Person de Teste, Época da convergência ou parada, Número de Neurônio do treinamento, Número do treinamento). O Apêndice B contém a listagem desse programa

$$MSE = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - \alpha_i)^2 \quad \text{Eq. ( 12 )}$$

Em que:

- N - número de dados apresentados para treinamento;
- e - diferença entre o valor esperado e o valor estimado pela rede;
- t - valor estimado pela rede;
- $\alpha$  - valor esperado

Após a elaboração do programa descrito foi realizado o treinamento de 20.000 (vinte mil) redes do tipo MLP sendo treinado mil redes em cada arquitetura, variando-se o número de neurônio na camada oculta entre 1 e 20, seguindo a recomendação de Hecht-Nielsen (1989) que diz que com  $2i+1$  neurônios na camada oculta é o necessário para mapear qualquer função continua com  $i$  entradas.

### 3.2.5. Seleção da rede

Após concluído o treinamento das redes neurais, foi gerado para cada treinamento um arquivo contendo os dados do treinamento (parâmetros utilizados no treinamento, índice dos dados de treinamento, teste, validação e performance) e um arquivo contendo os dados de todos os treinamentos consolidados.

Esse arquivo foi salvo no formato CSV que foi tratado utilizando em primeiro momento o software Notepad++ para deixar o arquivo compatível com o Microsoft Excel sendo que para

isso primeiro foram substituídos as “,” (vírgulas) por “;” (ponto e vírgula) e em seguida os “.” (pontos) por “,” (vírgulas), concluído esses ajustes o arquivo foi aberto no Microsoft Excel e salvo no formato nativo do XLS.

O arquivo resultante continha uma linha para cada rede treinada (20.000 redes), tendo suas colunas descritas no Quadro 2. As colunas de 1 a 65 são os valores estimados pela rede, sendo que algumas colunas foram adicionadas (colunas adicionadas possuem a coluna fórmula preenchida).

QUADRO 2 - Descrição das informações das colunas do arquivo contendo os dados do treinamento das redes.

Índice Coluna	Nome Coluna	Fórmula	Informação
66	BN	-	Performance geral da rede
67	BO	-	Performance no conjunto treinamento
68	BP	-	Performance no conjunto de validação
69	BQ	-	Performance no conjunto de teste
70	BR	-	R linear do conjunto de treinamento
71	BS	$=(BR_{nn})^2$	$R^2$ do conjunto de treinamento
72	BT	-	R linear do conjunto de validação
73	BU	$=(BT_{nn})^2$	$R^2$ do conjunto de validação
74	BV	-	R linear do conjunto de teste
75	BW	$=(BV_{nn})^2$	$R^2$ do conjunto de teste
76	BX	-	R linear do conjunto geral
77	BY	$=(BX_{nn})^2$	$R^2$ do conjunto geral
78	BZ	-	Correlação de Pearson do conjunto de treinamento
79	CA	-	Correlação de Pearson do conjunto de validação
80	CB	-	Correlação de Pearson do conjunto de teste
81	CC	-	Época que ocorreu a parada do treinamento (convergência)
82	CD	-	Número de neurônios na camada oculta do treinamento
83	CE	-	Número do treinamento realizada na arquitetura

nn: é o número da linha

Para determinar a rede com melhor desempenho foram eleitas algumas redes utilizando os critérios descritos a seguir.

A planilha foi ordenada pela coluna BN (Performance geral do conjunto) em ordem crescente, tendo em vista que para avaliação da performance foi definida a função de Erro Quadrático Médio, que representa o erro da rede. Assim quanto menor esse valor melhor será a performance da rede. Foi selecionada e marcada a linha com menor valor.

Selecionada a rede de melhor performance geral (BN) a próxima etapa foi repetir o processo para a coluna BO (Performance do conjunto de treinamento), BP (Performance do

conjunto de validação), em seguida BQ (Performance do conjunto de teste) seguindo o mesmo procedimento descrito para coluna BN.

Após a seleção das redes por performance prosseguiu-se para as colunas referentes ao  $R^2$  (BS –  $R^2$  de Treinamento, BU –  $R^2$  de Validação, BW –  $R^2$  de Teste e BY –  $R^2$  Geral), repetido o mesmo procedimento descrito anteriormente. Com a diferença que ordenação foi realizada de forma decrescente, tendo-se em vista que o  $R^2$  indica correlação entre os dados observados e os estimados pela RNA treinada.

Para concluir a seleção das RNAs foi repetido o esse mesmo processo para as colunas referentes a correlação de Pearson (BZ – para Treinamento, CA – para Validação e CB – para Testes) utilizando a ordenação decrescente, pois quanto maior o valor da correlação de Pearson mais próximo o valor estimado está do valor observado.

Para cada rede selecionada foi executado o programa para geração de imagens (Programa disponível no Apêndice C), gerando dois arquivos de imagens para cada rede, um com o gráfico de treinamento e outro com os gráficos do R linear dos conjuntos de treinamento, validação, teste e geral.

## 4. RESULTADOS E DISCUSSÃO

### 4.1. Análise da estatística descritiva das variáveis de entrada

A estatística descritiva das variáveis apresentadas para o treinamento teste e validação da rede MLP estão apresentados na Tabela 7, onde é possível observar o pequeno número de amostras (65 amostras) que foram utilizadas como entrada para o treinamento, validação e teste. Fato que dificultou o treinamento e validação da rede tendo em vista que RNAs do tipo MLP aprendem com exemplos.

TABLEA 7 - Estatística Descritiva das variáveis utilizadas no treinamento da RNAs

Característica	N	Média	Mín	Máx	Desvio padrão	C.V.
Altura	65	92,37361	78,80	112,18	8,87538	10%
Ramos/planta	65	4,81704	2,00	11,30	2,09585	44%
Massa de grãos/planta	65	18,18143	10,00	30,57	5,32825	29%
nº de vagens/planta	65	59,63328	27,60	119,80	19,28325	32%
grao/vagens	65	2,18204	1,82	2,67	0,27201	12%
Produz/há (kg)	65	4347,56436	2958,72	5248,17	477,41305	11%

N: número de amostras; Mín: valor mínimo; Máx: valor máximo; Fonte: autor

O coeficiente de variação em todas as características está igual ou acima dos 10% o que contribui para o treinamento, pois a variação dos valores permite o melhor ajuste dos pesos das sinapses das conexões. O alto coeficiente de variação, no entanto, o alto coeficiente de variação também pode representar *outliers* nos dados, no entanto a utilização de redes do tipo MLP utilizando validação cruzada conseguiu lidar com esses valores evitando a influência desses no ajuste dos pesos das sinapses garantindo que a rede não modelasse os ruídos presentes nas amostras (REED, 1993).

As 65 amostras obtidas foram divididas de forma aleatória em três subconjuntos: treinamento (42 amostras – 65%), validação (16 amostras – 25%) e teste (7 amostras – 10%), e para cada novo treinamento foi realizado um novo sorteio. Essa estratégia de divisão dificultou o treinamento, pois ao realizar a divisão aleatória sem considerar os tratamentos utilizados para obtenção das amostras, os conjuntos foram formados com amostras com pouca representatividade, o que teve como consequência várias redes com performances de treinamento elevadas, mas validação baixa na rede com 19 neurônio treinamento 806 (linha 3 da Tabela 8), onde a performance de treinamento observada da rede obteve  $R^2$  de 0,999 e validação com  $R^2$  de 0,145 (Tabela 8).



#### 4.2. Exploração de configurações para treinamento

Neste experimento foram desenvolvidas aproximadamente cem mil redes neurais do tipo MLP. Durante o treinamento foi criado um programa para automatizar os treinamentos, o que permitiu realizar vários testes de configuração de parâmetros para treinamento, possibilitando assim chegar ao resultado obtido no trabalho. Em cada uma das configurações discutidas abaixo foi variado o número de neurônios da camada oculta entre 1 e 20, com mil treinamentos distintos por variação totalizando vinte mil redes por configuração.

No início do experimento a primeira configuração da rede foi feita com uso dos valores das variáveis de entrada não normalizados, pois as redes treinadas não atingiram performance aceitável ( $R^2$  abaixo de 65%). Uma segunda configuração com a normalização dos valores de entrada, utilizando uma variável binária (que assume um ou zero) para cada uma das variáveis de categoria (hábito de crescimento e densidade, totalizando 6 variáveis de categoria), a configuração melhorou a sensibilidade da rede aos valores o que corrobora com Lafeté (2012), no entanto a performance continuou baixa.

Na tentativa de melhorar a performance fez-se a alteração da normalização da variável dependente (produtividade) que estava sendo normalizada para valores entre -1 e 1 para classes de intervalo (entre -1 e 1 variando em 0,2 com intervalos de 500 kg em cada intervalo, criando-se assim 11 classes), o que resultou em redução da performance geral das redes não atingindo valores de erros aceitáveis, o que é justificável pois ao reduzir os valores possíveis a aproximação da função de ativação do neurônio da camada de saída pode gerar valores mais distantes dos esperados (JOSÉ *et al.*, 2015).

Prosseguindo foi testada a configuração com 11 neurônios na camada de saída, na qual cada neurônio representa uma classe (classes de 500 kg em 500 kg em cada classe) de produtividade. Essa configuração foi testada na tentativa de reduzir o erro da rede, uma vez que cada neurônio da camada de saída foi tratado com valores de um ou zero. No entanto essa variação da quantidade de neurônios da camada de saída não contribuiu para melhorar a performance geral da rede, pelo contrário dificultou a convergência da rede e piorou a performance.

O treinamento que melhor apresentou resultados ( $R^2$  de treinamento de 0,98), e que serão discutidos em maiores detalhes, foi a configuração com duas variáveis para representar os três tipos de hábitos de crescimento (Tabela 3) e duas variáveis para representar as três densidades populacionais (Tabela 4) e a produtividade normalizada para valores entre -1 e 1.

### 4.3. Treinamento e seleção da RNA

O treinamento da rede RNA do tipo MLP foi realizado utilizando o programa descrito anteriormente e que pode ser acompanhado no Apêndice B chamando-se a função TREINAR\_V6 (Quadro 3), utilizando o computador descrito acima. Foram necessárias 2h54min para realizar o treinamento de vinte mil redes variando o número de neurônios na camada oculta de um a vinte.

QUADRO 3 - Parâmetros utilizado para realizar treinamento da rede MLP

Parâmetro	Valor	Descrição
NrCiclos	1000	Número de ciclos – número máximo de vezes que os dados de treinamento serão apresentados para cada treinamento
QteTreina	1000	Número de treinamentos – número de treinamentos que devem ser realizados em cada arquitetura (número de neurônio na camada oculta)
QteIniO	1	Quantidade de neurônios iniciais na camada oculta
QteIniO	20	Quantidade final de neurônios na camada oculta
QtePasO	1	Passos - é o incremento que será realizado na quantidade de neurônios na camada oculta cada vez que o número de treinamentos especificado for atingido

Na primeira versão da função que gerava os gráficos (performance, R linear de treinamento, teste, validação e geral) para todas as vinte mil redes foi necessário 36h42min para o treinamento, fato devido ao grande número de arquivo gerados e o tempo gasto para gravar os arquivos em disco. A função foi modificada para remover os gráficos o que agilizou consideravelmente o tempo necessário para o treinamento das redes.

O desenvolvimento do programa foi imprescindível para a determinação da arquitetura que conseguisse performance adequada, pois segundo Braga et al. (2012) o problema de dimensionamento envolve o ajuste entre a complexidade do modelo neural e a complexidade do problema a ser resolvido. Com a utilização do programa foi possível de forma simples variar a complexidade da arquitetura (variando o número de neurônio na camada oculta) permitindo assim a avaliação da performance em diferentes arquiteturas e configurações.

Outro artifício utilizado foi a repetição do treinamento em cada arquitetura mil vezes. Essa abordagem que a cada novo treinamento realizava novo sorteio dos conjuntos de treinamento, validação e teste assim como a inicialização dos pesos das sinapses, permitiu superar um problema comum no treinamento de redes MLP que utilizam algoritmo *backpropagation* devido a propensão de ficar preso em mínimos locais (ZWEIRI et al., 2005). A importância das repetições realizadas foi observada pela rede com melhor performance ( $R^2$

0,987 no treinamento e 0,727 na validação) e que foi encontrado após 963 repetições no treinamento da arquitetura com 9 neurônios na camada oculta, linha 2 da (Tabela 8).

TABELA 8 - Dados treinamento RNAs

Treinamento			Performance				R2				Correlação de Pearson			
NN	NT	C	Geral	Tre	Val	Test	Geral	Tre	Val	Test	Geral	Tre	Val	Test
14	124	18	0,0099	0,0008	0,0267	0,0264	0,743	0,980	0,240	0,203	0,862	0,990	0,490	0,451
9	963	19	0,0252	0,0005	0,0334	0,1550	0,611	0,987	0,727	0,033	0,782	0,993	0,852	-0,181
19	806	48	0,0439	0,00001	0,1383	0,0919	0,357	0,999	0,145	0,045	0,598	0,999	0,381	0,212
2	212	926	0,0275	0,0216	0,0455	0,0213	0,239	0,423	0,0001	0,991	0,489	0,650	0,013	0,996
6	336	42	0,0291	0,0335	0,0096	0,0475	0,195	0,283	0,264	0,151	0,441	0,532	0,513	-0,389
1	340	119	0,0325	0,0379	0,0315	0,0025	0,107	0,244	0,041	0,090	0,328	0,494	-0,203	0,300

C: Ciclos ou época que o treinamento da arquitetura foi finalizado; NN: número de neurônios na camada oculta; NT: número do treinamento realizado na arquitetura de NN; Tre: treinamento; Val: validação; Test: Teste; Performance: Erro Quadrático Médio (EQM); Fonte: autor

Durante o treinamento foi observado que a estratégia de um novo sorteio dos conjuntos de treinamento, validação e teste ocasionou problemas para a convergência e generalização das redes, pois a cada novo sorteio os tratamentos não foram levados em consideração, resultando no agrupamento de amostras no conjunto de treinamento, teste e validação com pouca representatividade, e que corrobora com os resultados obtidos por Lisboa e Taktak (2006) e comprovando que a rede passou a se especializar no conjunto de treinamento. Isto pode ser comprovado verificando o excelente  $R^2$  de 0,999 obtido durante o treinamento da rede com 19 neurônio repetição 806 (Figura 16), onde próximo à época 48 foi observado que a linha do conjunto de treinamento caiu drasticamente, indicativo de que a rede memorizou o conjunto de treinamento. Ao mesmo tempo observou-se que as linhas do conjunto de validação e teste não acompanharam essa queda apresentando baixo  $R^2$  de 0,145 e 0,045 respectivamente.

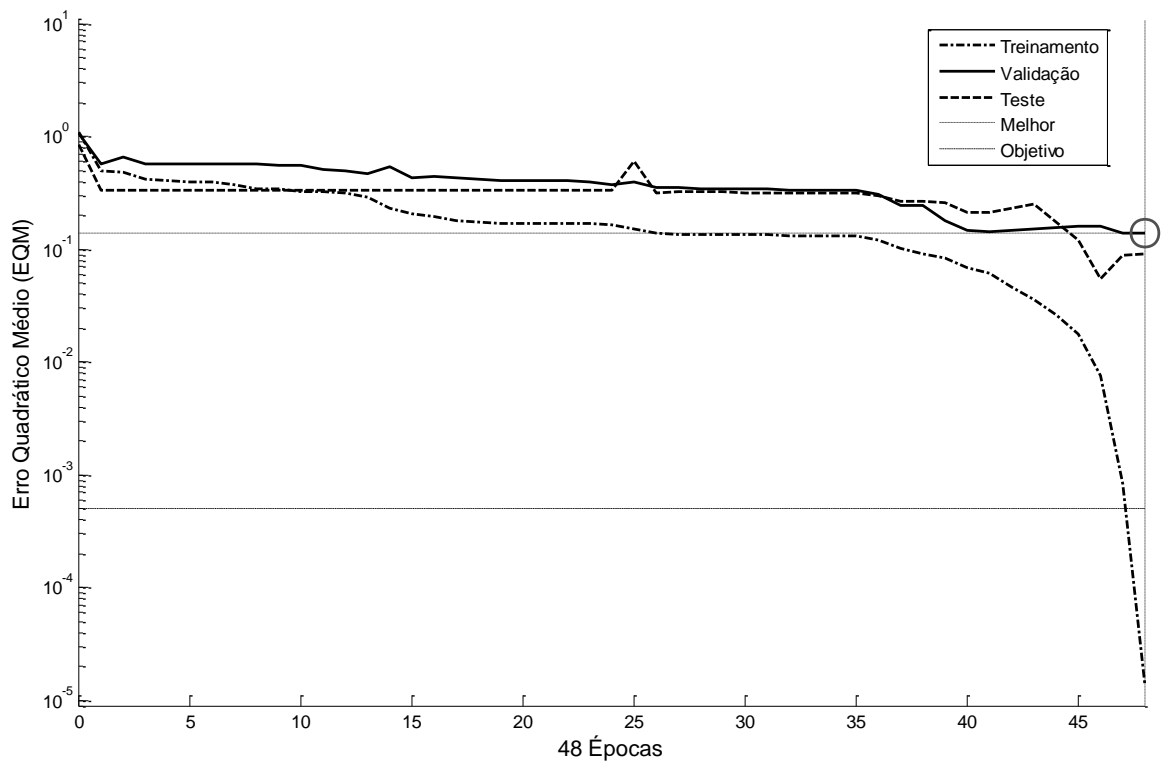


FIGURA 16 – Representação gráfica do treinamento da rede com 19 neurônios repetição 806

A rede com 2 neurônios repetição 212 (Figura 17) apresentou o mesmo problema referente ao sorteio dos conjuntos, diferindo apenas no fato de que o conjunto que demonstrou melhor correlação de Pearson foi o de teste com 99,55%. No entanto o conjunto de validação apresentou apenas 13,45% de correlação.

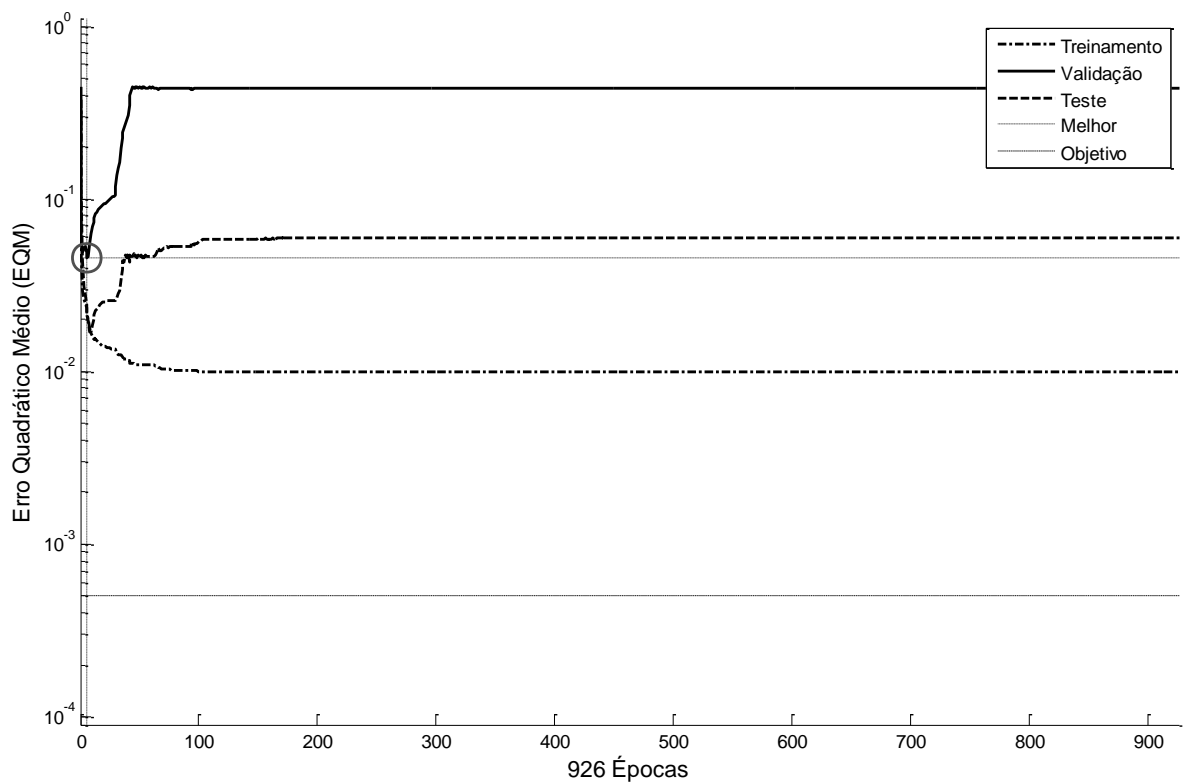


FIGURA 17– Representação gráfica do treinamento da rede com 2 neurônios repetição 212

Para selecionar a rede que melhor conseguiu generalizar o problema foram utilizados vários critérios para garantir que fosse selecionada a rede com maior capacidade de generalização. Redes com menor número de neurônios na camada oculta tem maior capacidade de generalização, no entanto a escolha de uma rede com poucos neurônios na camada oculta para um problema com alto grau de complexidade pode não conseguir modelar o problema incorrendo em *underfitting* (BRAGA *et al.*, 2012). Isto foi observado nos valores estimados e observados para a rede com 1 neurônio repetição 340 (Figura 18) que teve Erro Quadrático Médio (EQM) de 0,0315 melhor que o da rede com 9 neurônio repetição 963 (rede escolhida como a mais adequada) que apresentou EQM de 0,0334, percebeu-se que apesar do baixo EQM o  $R^2$  e a correlação de Pearson não foram bons, 0,041 e -0,203 respectivamente (Tabela 8), não conseguindo modelar a relação que existe entre as variáveis de entrada e o resultado esperado.

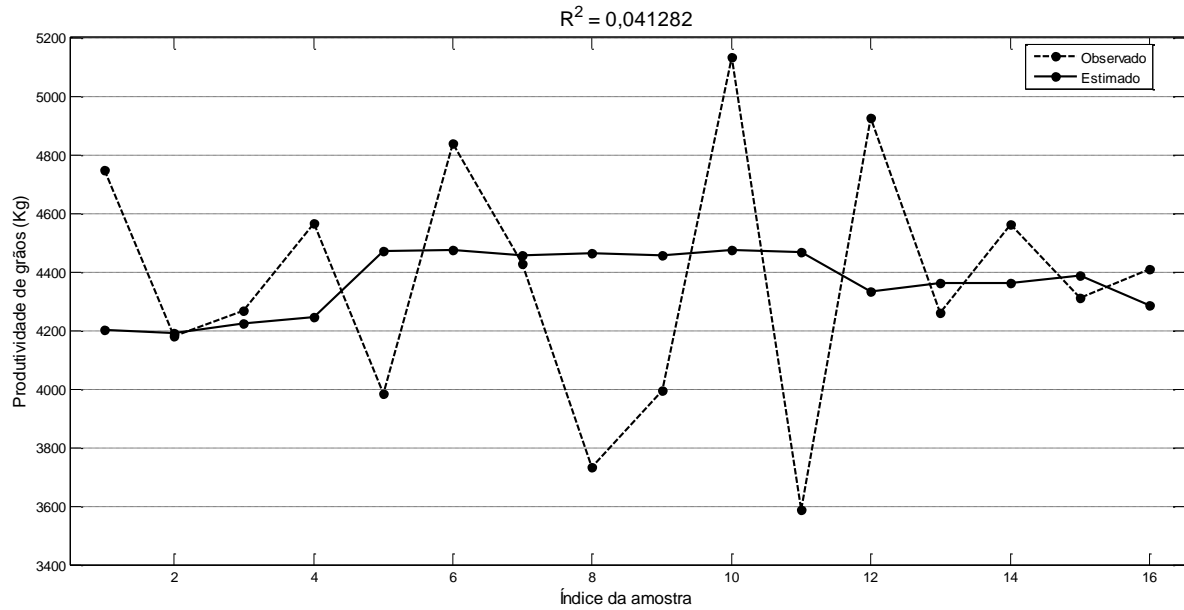


FIGURA 18 – Representação gráfica do comparativo dos valores estimados e observados do conjunto de validação, rede com 1 neurônio repetição 340

Na validação da rede com 9 neurônio repetição 963 foi possível verificar a tendência do valor observado acompanhar o estimado (Figura 19) o que é confirmada pelo  $R^2$  de 0,726 e correlação de Pearson de 85,20%. Caso fossem removidas as amostras 7 e 11 (Figura 19 índice 3 e 4) dessa validação, o  $R^2$  subiria para 0,811 e a correlação de Pearson para 90,08%, essa simulação foi realizada ao se observar que a amostra 11 e 2 (valores podem ser encontrados no Apêndice D – Tabela 10) possuem característica agronômicas com valores muito próximos e produtividade muito diferentes (quase 800 kg) o que levanta a hipótese que eventos não registrados no experimento (por exemplo ocorrência de pragas), que deu origem aos dados terem influenciado a produtividade nessas amostras.

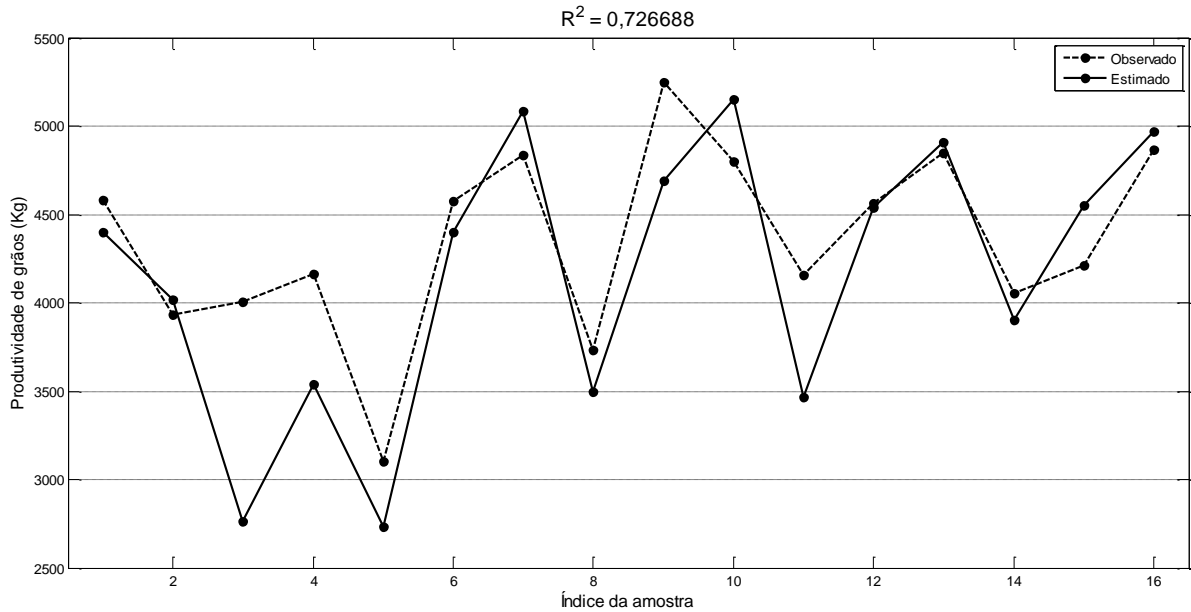


FIGURA 19 – Representação gráfica do comparativo dos valores estimados e observados do conjunto de validação, rede com 9 neurônio repetição 963

A rede com 9 neurônios repetição 963 selecionada como melhor solução para o problema não apresentou o melhor  $R^2$  no treinamento entre todas as redes treinadas, mas foi a selecionada por apresentar a melhor validação com  $R^2$  de 0,727 e correlação de Pearson de 85,2%. No treinamento foi observado que os valores estimados e observados ficaram muito próximo o que demonstra a capacidade de aprendizagem da RNA do tipo MLP (Figura 20 e 21).

Na validação, apesar de alcançar performance considerável ( $R^2$  de 0,727 e correlação de Pearson de 85,20%), apresentou alguns valores distantes da reta, sendo que isto pode ser atribuído a influência da produtividade das amostras selecionadas no conjunto de validação, que não foram considerados no treinamento. Também foi verificado que a performance do conjunto de teste ficou muito baixa ( $R^2$  de 0,033 e correlação de Pearson -18,10%), reforçando a questão de que a seleção dos conjuntos terem sido realizadas totalmente ao acaso agrupar amostras com baixa representatividade e a hipótese dos fatores não registrados também deve ser considerada.

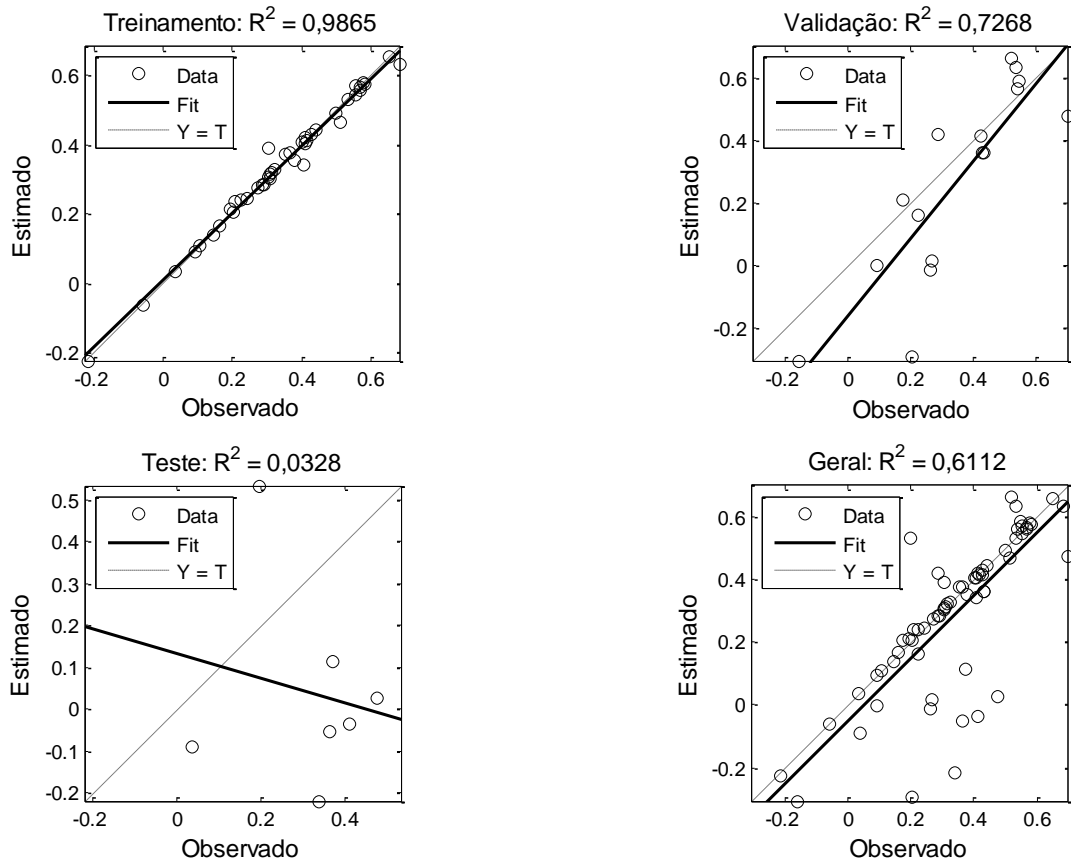


FIGURA 20 – Representação gráfica do R linear de treinamento, validação, teste e geral da rede com 9 neurônio treinamento 963

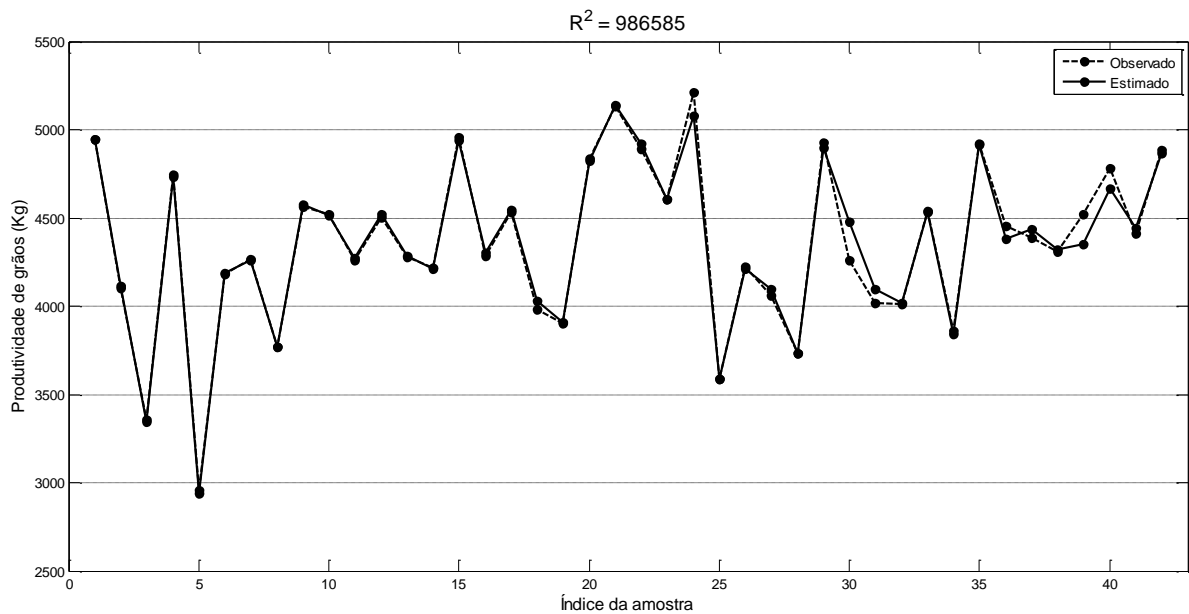


FIGURA 21 – Representação gráfica do comparativo dos valores observado e estimados de treinamento, rede com 9 neurônios repetição 963



Em geral, pode-se constatar que o conjunto teste não apresentou boa correlação de Pearson, no entanto, desconsiderando as amostras 14,29 e 32 (índice 2,3 e 4 Figura 22) pôde-se perceber que os valores observados e estimados apresentaram, apesar do erro, uma relação, o que é confirmado pelo aumento do  $R^2$  para 0,730 (Figura 23).

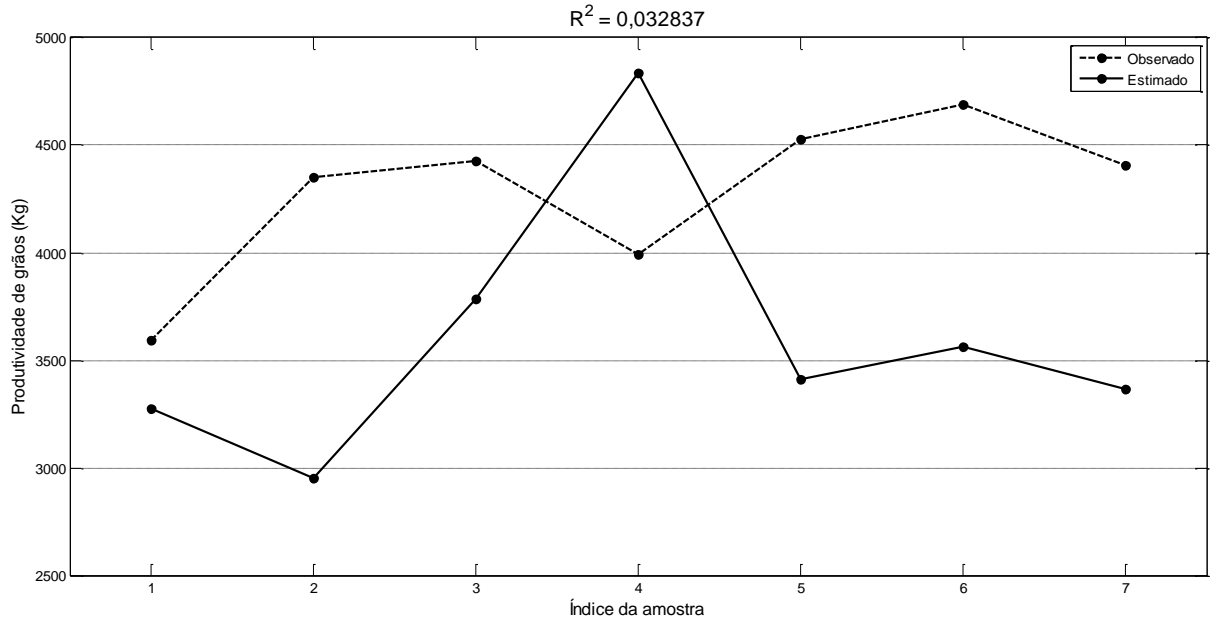


FIGURA 22 – Representação gráfica do comparativo dos valores observados e estimados de teste, rede com 9 neurônio repetição 963

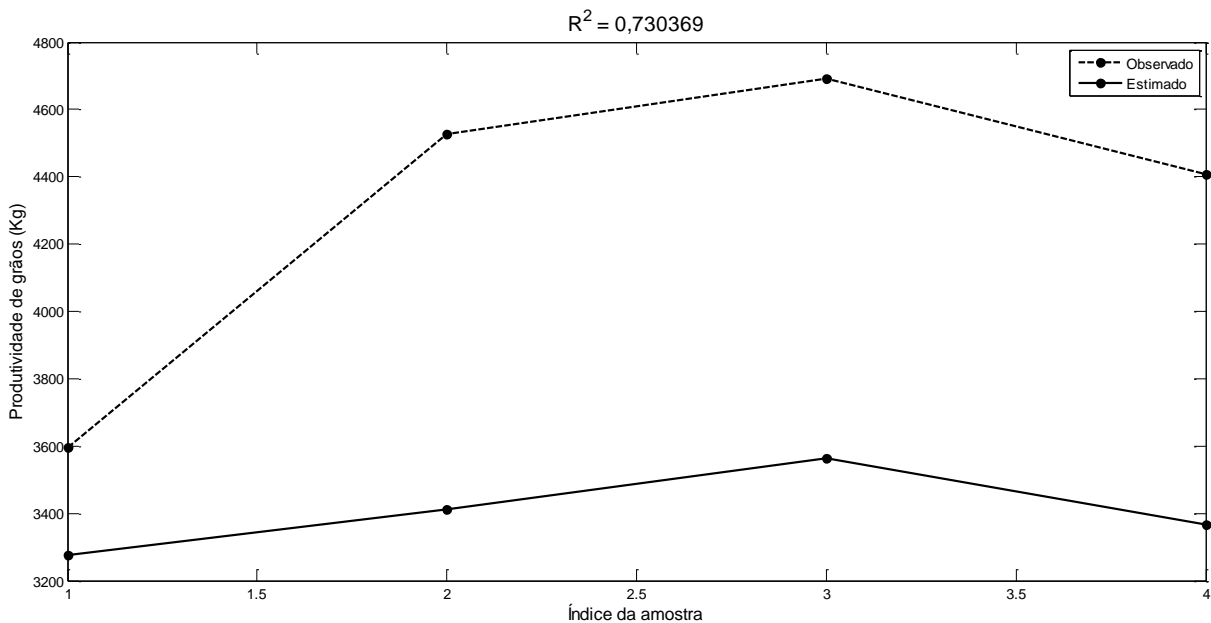


FIGURA 23 – Representação gráfica do comparativo dos valores observado e estimados de teste, rede com 9 neurônio repetição 963 após removidas as amostras 14,29 e 32.

#### 4.4. Rede selecionada

A rede selecionada conforme descrita na seção anterior foi a rede com 9 neurônios repetição 963 do treinamento. Esta rede teve Erro Quadrático Médio (EQM) geral de 0,0252 e treinamento de 0,0005 (condição para convergência estipulada no programa de treinamento). A avaliação para seleção da rede levou em consideração não apenas o EQM mas também o  $R^2$  e a correlação de Pearson dos conjuntos de treinamento, validação e teste.

O treinamento da rede neural ajustou os valores dos nove neurônios da camada oculta e os valores dos *bias* de cada um, assim como do neurônio de saída.

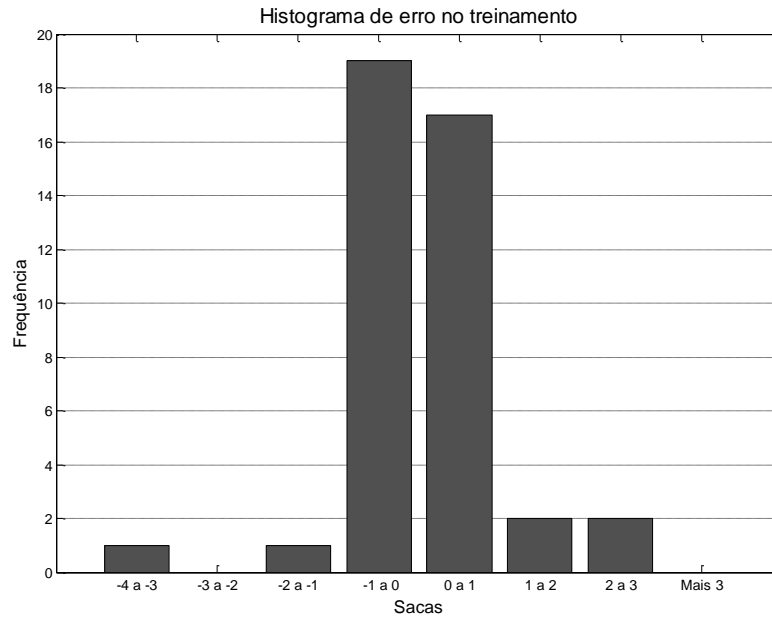
TABELA 9 - Análise dos erros da rede com 9 neurônio repetição 963

	Amostras	Mín	Máx	Média	Erro Médio Absoluto	Desvio
Geral	65	-841,23	1398,14	122,28	203,78	380,75
Treinamento	42	-216,42	167,81	1,17	30,42	55,85
Validação	16	-351,78	1246,33	193,32	340,56	427,76
Teste	7	-841,23	1398,14	686,57	926,92	761,56

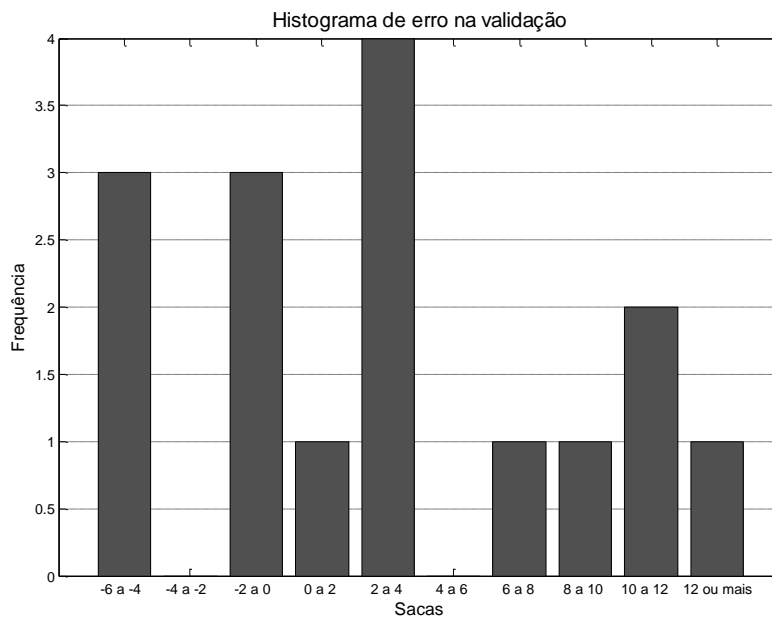
Mín: Mínimo; Máx: Máximo; Erro Médio Absoluto: Média calculada a partir dos valores absoluto do erro; Valores Mínimo abaixo de zero indicam que a rede estimou acima do valor observado. Valores expressos em kg.

A diferença entre a produtividade de grãos observada e a estimada (erro) da rede com o conjunto de treinamento foi de 30,42 kg, o que confirma a alta correlação de Pearson de 99,30%. O erro médio absoluto foi avaliado no lugar do erro médio (1,17 kg) para não mascarar a distância entre os valores observados e os estimados, pois os valores estimados podem ser negativos indicando que a rede estimou acima do valor observado, o que diminui o erro médio.

Na Figura 24 vemos a distribuição dos erros em sacas, observou-se que no treinamento a rede apresentou 85,7% das amostras com um erro de uma saca (60 kg) para menos ou para mais, considerando que o maior erro no treinamento foi de 3,6 sacas e que a média de sacas por hectare da região centro-sul foi de cinquenta sacas (CONAB, 2015) temos um erro de 7% entre o valor observado e o estimado. No conjunto de validação devido a correlação de Pearson menor (85,2%) o erro ficou mais distribuído, mas desconsiderarmos as amostras 7 e 11 conforme descrito na seção anterior, teríamos 78,5% do erro entre seis sacas para mais ou para menos o que leva a um erro de 12% na estimativa da produtividade em sacas.



(a)



(b)

FIGURA 24 – Histograma de erro em sacas no treinamento (a) e validação (b)

O desempenho apresentado pela RNA do tipo MLP foi considerado bom comparado com outros trabalhos de estimativa de produtividade da soja como o apresentado por Fontana (2001) que obteve correlação entre valores observados e estimados de 0,85 utilizando o modelo de Jensen modificado para a estimativa do rendimento da cultura de soja no Estado do Rio Grande do Sul. Monteiro e Sentelhas (2014) utilizando modelo agrometeorológicos obtiveram índice de correlação  $R^2$  de 0,64 para estimar a produtividade da soja em diferentes regiões brasileiras.

## 5. CONCLUSÕES

O emprego de sistemas computacionais inteligentes na estimação da produtividade da soja é viável.

A RNA do tipo *Multilayer Perceptron* (MLP) com treinamento supervisionado conseguiu estimar com assertividade considerável a produtividade da soja utilizando as características agronômicas, hábito de crescimento e densidade populacional.

A arquitetura que melhor estimou a produtividade da soja foi de 11 neurônios na camada de entrada, 9 na camada oculta e 1 na camada de saída (11-9-1).

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

ABIOVE, Associação Brasileira das Indústrias de Óleos Vegetais, 2015. **Brasil - exportações do complexo soja**. 2015

ALEKSANDER, I.; MORTON, H. **An introduction to neural computing**. Lodon: Int Thomson Comput Press, 1995.

ALVAREZ, R. Predicting average regional yield and production of wheat in the argentine pampas by an artificial neural network approach. **European Journal of Agronomy**, . v. 30, n. 2, p. 70–77. 2009.

AMATO, F.; LÓPEZ, A.; PEÑA-MÉNDEZ, E. M.; VAÑHARA, P.; HAMPL, A.; HAVEL, J. Artificial neural networks in medical diagnosis. **Journal of Applied Biomedicine**, . v. 11, p. 47–58. 2013.

ARAÚJO, E. C. De; URIBE-OPAZO, M. A.; JOHANN, J. A. Modelo de regressao espacial para estimativa da produtividade da soja associada a variáveis agrometeorológicas na região oeste do estado do paraná. Jaboticabal: **Engenharia Agrícola**, . v. 34, n. 2, p. 286–299. 2008.

ARAÚJO, M. A.; SOUZA, J. L. M. De; TSUKAHARA, R. Y. Modelos agro-meteorológicos na estimativa da produtividade da cultura da soja na região de ponta grossa, estado do paraná. **Acta Scientiarum. Agronomy**, . v. 33, n. 1, p. 23–31. 2011.

BERKA, L. M. S.; RUDORFF, B. F. T.; SHIMABUKURO, Y. E. Soybean yield estimation by an agrometeorological model in a gis. Piracicaba: **Scientia Agricola**, . v. 60, n. 3, p. 433–440. 2003.

BERLATO, M. A.; FONTANA, D. C.; GONÇALVES, H. M. Relação entre o rendimento de grãos da soja e variáveis meteorológicas. Brasília: **Pesquisa Agropecuária Brasileira**, . v. 27, n. 5, p. 695–702. 1992.

BOHL, M. T.; DIESTELDORF, J.; SALM, C. A.; WILFLING, B. Spot market volatility and futures trading: the pitfalls of using a dummy variable approach. **Journal of Futures Markets**, . v. XX, n. X, p. n/a–n/a. 2015.

BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDEMIR, T. B. **Redes neurais artificiais: Teoria e Aplicações**. 2. ed. Rio de Janeiro: LTC, 2012.

BRASMAX, 2015. **Cultivares de soja para região do cerrado**. 2015

CHUNG, W. H.; JEONG, N.; KIM, J.; LEE, W. K.; LEE, Y. G.; LEE, S. H.; YOON, W.; KIM, J. H.; CHOI, I. Y.; CHOI, H. K.; MOON, J. K.; KIM, N.; JEONG, S. C. Population structure and domestication revealed by high-depth resequencing of korean cultivated and wild soybean genomes. **DNA Research**, . v. 21, n. 2, p. 153–167. 2014.

CONAB, Companhia Nacional de Abastecimento, 2015. **Acompanhamento da safra brasileira: grãos**. 2015

CONAB, Companhia Nacional de Abastecimento, 2016. **Acompanhamento da safra brasileira: grãos**. 2016

COSTA, J. A. **Cultura da soja**. Porto Alegre: I. Manica & J. A. Costa, 1996.

DORAISWAMY, P. C.; HATFIELD, J. L.; JACKSON, T. J.; AKHMEDOV, B.; PRUEGER, J.; STERN, A. Crop condition and yield simulations using landsat and modis. **Remote Sensing of Environment**, . v. 92, n. 4, p. 548–559. set. 2004.

EMBRAPA. Soja em números: a evolução da soja no brasil. [S.l.], 2012.

EMBRAPA, Empresa Brasileira de Pesquisa Agropecuária, 2015. **Cultivares de soja**. 2015

ERZIN, Y.; RAO, B. H.; PATEL, A.; GUMASTE, S. D.; SINGH, D. N. Artificial neural network models for predicting electrical resistivity of soils from their thermal resistivity. **International Journal of Thermal Sciences**, . v. 49, n. 1, p. 118–130. 2010.

FANG, Q.; HANNA, M. A.; HAQUE, E.; SPILLMAN, C. K. Neural network modeling of energy requirements for size reduction of wheat. **Transactions of the ASAE**, . v. 43, n. 4, p. 947–952. 2000.

FONTANA, D. C.; BERLATO, M. A.; LAUSCHNER, M. H. Modelo de estimativa de rendimento de soja no estado do rio grande do sul. Brasília: **Pesquisa Agropecuária Brasileira**, . v. 36, n. 3, p. 399–403. 2001.

GOMES, A. C. Dos S.; ROBAINA, A. D.; PEITER, M. X.; SOARES, F. C.; PARIZI, A. R. C. Modelo para estimativa da produtividade para a cultura da soja. (portuguese). **Model to estimate the yield for soybean. (English)**, . v. 44, n. 1, p. 43–49. 2014.

HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. H. **Neural network design**. [S.l.]: Martin Hagan, 2014.

HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. **IEEE Transactions on Neural Networks**, . v. 5, n. 6, p. 989–993. 1994.

HAYKIN, S. **Neural network: A Comprehensive Foundation**. 2. ed. Ontario: Pearson Prentice Hall, 2005.

HECHT-NIELSEN, R. **Neurocomputing**. San Diego: Addison-Wesley, 1989.

HEIFFIG, L. S.; CÂMARA, G. M. D. S.; MARQUES, L. A.; PEDROSO, D. B.; PIEDADE, S. M. D. S. Fechamento e índice de área foliar da cultura da soja. Campinas: **Bragantia**, . v. 65, n. 2, p. 285–295. 2006.

HOLTZ, V.; COUTO, R. F.; OLIVEIRA, D. G. De. Deposição de calda de pulverização e produtividade da soja cultivada em diferentes arranjos espaciais. Santa Maria: . p. 1371–1376. 2014.

JOSÉ, A.; LEAL, F.; MIGUEL, E. P.; HENRIQUE, F.; BAIO, R.; NEVES, D. D. C.; ALVES, U.; LEAL, S. Redes neurais artificiais na predição da produtividade de milho e definição de sítios de manejo diferenciado por meio de atributos do solo. **Bragantia**, . v. 74, p. 436 – 444. 2015.

KAUL, M.; HILL, R. L.; WALTHALL, C. Artificial neural networks for corn and soybean yield prediction. Beltsville: **Agricultural Systems**, . v. 85, n. 1, p. 1–18. 2005.

KOMINAKIS, A. P.; ABAS, Z.; MALTARIS, I.; ROGDAKIS, E. A preliminary study of the application of artificial neural networks to prediction of milk yield in dairy sheep. **Computers and Electronics in Agriculture**, . v. 35, n. 1, p. 35–48. 2002.

LACLC, Los Angeles Chinese Learning Center, 2003. **History of soybeans**. 2003

LAFETÁ, B. O. **Eficiência nutricional, área foliar e produtividade de plantações de eucalipto em diferentes espaçamentos estimados com redes neurais artificiais**. 2012. Dissertação (Mestrado em Ciência Florestal) - Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina, 2012.

LEE, C.; HERBEK, J. Estimating soybean yield. **University of Kentucky**, . v. agr-188, n. 12-2005, p. 7–8. 2005.

LEE, G.-A.; CRAWFORD, G. W.; LIU, L.; SASAKI, Y.; CHEN, X. Archaeological soybean (glycine max) in east asia: does size matter? **PLoS ONE**, . v. 6, n. 11, p. e26720. 2011.

LISBOA, P. J.; TAKTAK, A. F. G. The use of artificial neural networks in decision support in cancer: a systematic review. Liverpool: **Neural networks: the official journal of the International Neural Network Society**, . v. 19, n. 4, p. 408–415. 2006.

MAPA, Ministério da Agricultura, Pecuária e Abastecimento, 2009. **Regras para análise de sementes**. 2009

MARTIN, T. N. **Modelo estocástico para estimação da produtividade de soja no estado de são paulo utilizando simulação normal bivariada**. 2007. Tese (Doutorado em Fitotecnia) - Escola Superior de Agricultura Luiz de Queiroz: Universidade de São Paulo, Piracicaba, 2007.

MARTIN, T. N.; DOURADO NETO, D.; VIEIRA JUNIOR, P. A.; PEREIRA, A. R.; MANFRON, P. A.; CHRISTOFFOLETI, P. J. Modified crop model estimation of depleted and potential soybean yield. **Acta Scientiarum. Agronomy**, . v. 34, n. 4, p. 369–378. 6 set. 2012.

MATSUNAGA, V. Y. Curso de redes neurais utilizando o matlab. 2012.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. Illinois: **The Bulletin of Mathematical Biophysics**, . v. 5, n. 4, p. 115–133. 1943.

MENEZES, P. L. De; AZEVEDO, C. A. V. De; EYNG, E.; NETO, J. D.; LIMA, V. L. A. De. Artificial neural network model for simulation of water distribution in sprinkle irrigation. **Revista Brasileira de Engenharia Agrícola Ambiental**, . v. 19, n. 9, p. 817–822. 2015.

MERCANTE, E.; LAMPARELLI, R. A. C.; URIBE-OPAZO, M. A.; ROCHA, J. V. **Modelos de regressão lineares para estimativa de produtividade da soja no oeste do paraná, utilizando dados espectrais**. 2010. 2010.

MONTEIRO, L. A.; SENTELHAS, P. C. <b>calibration and testing of an agrometeorological model for the estimation of soybean yields in different brazilian regions. **Acta Scientiarum. Agronomy**, . v. 36, n. 3, p. 265. 2014.

NUNES, J. L. Da S. Histórico da soja. [S.l.], 2013.

PIRES, J. L. F.; COSTA, J. A.; THOMAS, A. L.; MAEHLER, A. R. Efeito de populações e espaçamentos sobre o potencial de rendimento da soja durante a ontogenia. Brasília: **Pesquisa Agropecuária Brasileira**, . v. 35, n. 8, p. 1541–1547. 2000.

PRASAD, A. K.; CHAI, L.; SINGH, R. P.; KAFATOS, M. Crop yield estimation model for iowa using remote sensing and surface parameters. **International Journal of Applied Earth Observation and Geoinformation**, . v. 8, n. 1, p. 26–33. 2006.

REED, R. Pruning algorithms - a survey. **IEEE Transactions on Neural Networks**, . v. 4, n. 5, p. 740–747. 1993.

RUSSELL, S.; NORVIG, P. **Artificial intelligence: A modern Approach**. New Jersey: Pearson Education, Inc., 2010.

SAFA, M.; SAMARASINGHE, S.; NEJAT, M. Prediction of wheat production using artificial neural networks and investigating indirect factors affecting it: case study in canterbury province , new zealand. **Journal Of Agricultural Science And Technology**, . v. 17, n. 4, p. 791–803. 2015.

SCHIAVO, M. LO; PRINARI, B.; GRONSKI, J. A.; SERIO, A. V. An artificial neural network approach for modeling the ward atmosphere in a medical unit. **Mathematics and Computers in Simulation**, . v. 116, p. 44–58. 2015.

NIDERA Sementes, 2015. **Cultivares de soja para o centro**. 2015

SHARMA, A. K.; SHARMA, R. K.; KASANA, H. S. Prediction of first lactation 305-day milk yield in karan fries dairy cattle using ann modeling. **Applied Soft Computing Journal**, . v. 7, n. 3, p. 1112–1120. 2007.



SILVA, I. N. Da; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas – curso prático**. São Paulo: Artliber, 2010.

SOARES, J. D. R.; PASQUAL, M.; LACERDA, W. S.; SILVA, S. O.; DONATO, S. L. R. Utilization of artificial neural networks in the prediction of the bunches' weight in banana plants. Lavras: **Scientia Horticulturae**, . v. 155, p. 24–29. 2013.

SRINIVASAN, D.; LIEW, A. C.; CHANG, C. S. A neural network short-term load forecaster. **Electric Power Systems Research**, . v. 28, n. 3, p. 227–234. 1994.

TORRES, M.; HERVÁS, C.; AMADOR, F. Approximating the sheep milk production curve through the use of artificial neural networks and genetic algorithms. **Computers & Operations Research**, . v. 32, n. 10, p. 2653–2670. 2005.

TOURINO, M. C. C.; REZENDE, P. M. De; SALVADOR, N. Espaçamento , densidade e uniformidade de semeadura na produtividade e características agronômicas da soja. Brasília: **Pesquisa Agropecuária Brasileira**, . v. 37, n. 8, p. 1071–1077. 2002.

UNO, Y.; PRASHER, S. O.; LACROIX, R.; GOEL, P. K.; KARIMI, Y.; VIAU, A.; PATEL, R. M. Artificial neural networks to predict corn yield from compact airborne spectrographic imager data. **Computers and Electronics in Agriculture**, . v. 47, n. 2, p. 149–161. 2005.

USDA, United States Department of Agriculture, 2016. **World agricultural supply and demand estimates**. 2016

VENTIMIGLIA, L. A.; COSTA, J. A.; THOMAS, A. L.; PIRES, J. L. F. Potencial de rendimento da soja em razão da disponibilidade de fósforo no solo e dos espaçamentos. Brasília: **Pesquisa Agropecuária Brasileira**, . v. 34, n. 2, p. 195–199. 1999.

ZACHARIAS, A. O.; CAMARGO, M. B. P. DE; FAZUOLI, L. C. Modelo agrometeorológico de estimativa do início da florada plena do cafeeiro. **Bragantia**, . v. 67, n. 1, p. 249–256. 2008.

ZARIFNESHAT, S.; ROHANI, A.; GHASSEMZADEH, H. R.; SADEGHI, M.; AHMADI, E.; ZARIFNESHAT, M. Predictions of apple bruise volume using artificial neural network. Mashhad: **Computers and Electronics in Agriculture**, . v. 82, p. 75–86. 2012.

ZWEIRI, Y. H.; SENEVIRATNE, L. D.; ALTHOEFER, K. Stability analysis of a three-term backpropagation algorithm. **Neural Networks**, . v. 18, p. 1341–1347. 2005.

## 7. APÊNDICES

### 7.1. Apêndice A – Programa para importar dados de treinamento.

```
%PROGRAMA PARA IMPORTAR DADOS DO ARQUIVO DE CSV EXPORTADO
% NOME: import_data_completo_v6
%CONTENDO 10 COLUNAS, SENDO AS 9 PRIMEIRA OS NEURÔNIOS DE ENTRADA E A 10ª O
VALOR DE SAÍDA DA REDE (OUTPUT)
%
% Copyright: Guiliano Rangel Alves
%
%DATA: 12/2015

% ARQUIVO A SER IMPORTADO
filename = 'C:\Dropbox\UEG\2015\mestrado\Projeto\matlab\dados _v10.txt';
delimiter = ',';

%% Read columns of data as strings:
% For more information, see the TEXTSCAN documentation.
formatSpec = '%s%s%s%s%s%s%s%s%s%s%[\n\r]';

%% Open the text file.
fileID = fopen(filename,'r');

%% Read columns of data according to format string.
% This call is based on the structure of the file used to generate this
% code. If an error occurs for a different file, try regenerating the code
% from the Import Tool.
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,
'ReturnOnError', false);

%% Close the text file.
fclose(fileID);

%% Convert the contents of columns containing numeric strings to numbers.
% Replace non-numeric strings with NaN.
raw = [dataArray{:},1:end-1];
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=[1,2,3,4,5,6,7,8,9,10]
    % Converts strings in the input cell array to numbers. Replaced non-
    numeric
    % strings with NaN.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Create a regular expression to detect and remove non-numeric
        prefixes and
        % suffixes.
        regexstr = '(?<prefix>.*?) (?<numbers>([-
]*(\d+(\.)*+[\,]{0,1}\d*[eEdD]{0,1}[-+]*\d*[i]{0,1})|([-
]*(\d+(\.)*+[\,]{1,1}\d+[eEdD]{0,1}[-+]*\d*[i]{0,1})) (?<suffix>.*?));
        try
            result = regexp(rawData{row}, regexstr, 'names');
            numbers = result.numbers;

            % Detected commas in non-thousand locations.
            invalidThousandsSeparator = false;
            if numbers=='.';

```

```

        thousandsRegExp = '^\\d+(\\.\\d{3})*\\.{0,1}\\d*$';
        if isempty(regexpi(thousandsRegExp, '.', 'once'));
            numbers = NaN;
            invalidThousandsSeparator = true;
        end
    end
    % Convert numeric strings to numbers.
    if ~invalidThousandsSeparator;
        numbers = strrep(numbers, '.', '');
        numbers = strrep(numbers, ',', '.');
        numbers = textscan(numbers, '%f');
        numericData(row, col) = numbers{1};
        raw{row, col} = numbers{1};
    end
    catch me
    end
end
end

%% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x), raw); % Find non-numeric
cells
raw(R) = {NaN}; % Replace non-numeric cells

%% Allocate imported array to column variable names
%% resultado (valor esperado)
CPROD = cell2mat(raw(:, 10));

%% matriz de dados(dados de treinamento)
cmat = cell2mat(raw);

%% matriz para ser utilizado como resultado esperado no treinamento
CPROD_FINAL = CPROD.';
%% matriz com os dados de teste(dados para ser utilizado no treinamento)
cmat_final = cmat.';
%% remove first column
%cmat_final(:,1)=[];
%% remover last row
cmat_final(end,:) = [];

%% remove first column from prod
%CPROD_FINAL(:,1)=[];

%% Clear temporary variables
clearvars filename delimiter formatSpec R fileID raw dataArray ans
numericData col rawData row regexstr result numbers
invalidThousandsSeparator thousandsRegExp me;

```

## 7.2. Apêndice B – Programa para treinamento do MLP.

```

%FUNCAO PARA TREINAR UMA RNA COM ALGORITMO BACKPROPAGATION
% NOME: TREINAR_V6
%
% Copyright: Guiliano Rangel Alves

%
%SINTAXE: prede = treina_v6(NrCiclos,QteTreina,QteIniO,QteFimO,QtePasO);
%
%PARAMETROS DE ENTRADA:
% NrCiclos: Numero de ciclos de cada treinamento
% QteTreina: Quantidade de treinamentos
% QteIniO: Quantidade inicial de neuronios da camada oculta
% QteFimO: Quantidade final de neuronios da camada oculta
% QtePasO: Quantidade de passos de neuronios da camada oculta
%
%PARAMETROS DE SAIDA:
% prede: Objeto contendo uma RNA treinada com os dados de treinamento
%
%DATA: 12/2015

function prede = TREINAR_V6(NrCiclos,QteTreina,QteIniO,QteFimO,QtePasO)

%programa que importa os dados do arquivo csv exportado para o treinamento
import_data_completo_v6;

inputs = cmat_final;
targets = CPRÓD_FINAL;

% Nr de neuronios da camada de saida - nao variavel
pcamadas(1,2) = 1

numEntradas = size(cmat_final,1);
numAmostras = size(cmat_final,2);

limites = zeros(numEntradas,2);
limites(1:numEntradas,1)=-1;
limites(1:numEntradas,2)=1;

matResults = zeros(1,numAmostras+14);%estimations(N),performance
trainPerformance valPerformance testPerformance rTra rVal rTst rAll
personTra personVal personTst ContNeuro ContTreina
matErrors = zeros(1,numAmostras+6);%errors(N), performance,
trainPerformance,valPerformance, testPerformance, nneuro, treinum

for ContNeuro = QteIniO:QtePasO:QteFimO
    %*** Nome do arquivo de exportação txt => Co=Qte de neuronios da
    Camada oculta; Tr=Treinamento
    NomArqRede = strcat('Net',int2str(ContNeuro));
    NomArqRede = strcat(NomArqRede,'Tr');

    % Nr de neuronios da camada oculta - variavel
    pcamadas(1,1) = ContNeuro

    for ContTreina = 1:QteTreina

```

```

    %*** Nome do arquivo de exportação txt => Sequencia do
    Treinamento
    ArqRedeTre = strcat(NomArqRede,int2str(ContTreina));

    %*** Criando e salvando a estrutura da RNA Feed Forward =>
    VariavelRede = newff(limites,camadas)
    prede = newff(limites,pcamadas);
    save(strcat('Redes\ ',ArqRedeTre,'In'),'prede');

prede.divideFcn = 'dividerand'; % Divide data randomly
prede.divideMode = 'sample'; % Divide up every sample
prede.divideParam.trainRatio = 65/100;
prede.divideParam.valRatio = 25/100;
prede.divideParam.testRatio = 10/100;

% For help on training function 'trainlm' type: help trainlm
% For a list of all training functions type: help nntrain
prede.trainFcn = 'trainlm'; % Levenberg-Marquardt

% Choose a Performance Function
% For a list of all performance functions type: help
nntperformance
prede.performFcn = 'mse'; % Mean squared error

prede.trainParam.show = 200;
prede.trainParam.goal = 0.0005;
prede.trainParam.mu = 0.0001;
prede.trainParam.max_fail = 6000;
prede.trainParam.epochs = NrCiclos;
%*** Treinando a RNA com algoritmo Backpropagation
[prede,tr] = train(prede,inputs,targets);

epochs = tr.epoch(end);

% Test the Network
outputs = prede(inputs);
errors = gsubtract(targets,outputs);
performance = perform(prede,targets,outputs);

% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets .* tr.valMask{1};
testTargets = targets .* tr.testMask{1};
trainPerformance = perform(prede,trainTargets,outputs);
valPerformance = perform(prede,valTargets,outputs);
testPerformance = perform(prede,testTargets,outputs);

trOut = outputs(tr.trainInd);
vOut = outputs(tr.valInd);
tsOut = outputs(tr.testInd);
trTarg = targets(tr.trainInd);
vTarg = targets(tr.valInd);
tsTarg = targets(tr.testInd);

NomFase = 'Treinamento\';

%Para pegar o valor do R
% rTra = Training

```

```

% rVal = Validating
% rTst = Testing
% rAll = All

[rTra,mTra,bTra] = regression(trTarg,trOut);
[rVal,mVal,bVal] = regression(vTarg,vOut);
[rTst,mTst,bTst] = regression(tsTarg,tsOut);
[rAll,mAll,bAll] = regression(targets,outputs);

personTra = corr (trTarg(:),trOut(:));
personVal = corr (vTarg(:),vOut(:));
personTst = corr (tsTarg(:),tsOut(:));

%
matResults = [matResults;[outputs performance trainPerformance
                        valPerformance testPerformance rTra rVal rTst rAll
                        personTra personVal personTst epochs ContNeuro
                        ContTreina]];
matErrors = [matErrors;[errors performance trainPerformance
                        valPerformance testPerformance ContNeuro
                        ContTreina]];

*** Salva pesos (arquivo MAT) da rede Final treinada
save (strcat (NomFase,ArqRedeTre,'F'),'prede','tr');

%informação do andamento do treinamento
strcat (int2str (ContNeuro),'-',int2str (ContTreina))
end %ContTreina

end %ContNeuro

csvwrite (strcat (NomFase,'out.csv'), matResults);

headerRow = ['performance' 'trainPerformance' 'valPerformance'
             'testPerformance' 'nneuro' 'treinum'];
csvwrite (strcat (NomFase,'err.csv'), headerRow);
csvwrite (strcat (NomFase,'err.csv'), matErrors);
end

```

### 7.3. Apêndice C – programa para geração de gráficos

```

cd C:\Dropbox\UEG\2015\mestrado\Projeto\matlab\treino;
%importa os dados utilizados para treinamento, validação e teste
import_data_completo_v6;

inputs = cmat_final;
targets = CPROD_FINAL;

NomFase = 'Treinamento\';
NomeFaseSave = 'Treinamento\imagens\';
ContNeuro = 19;% número de neurônio da rede que se deseja plotar
ContTreina = 40;% número do treinamento da rede que se deseja plotar

%*** Nome do arquivo de exportação txt => Co=Qte de neuronios da Camada
oculta; Tr=Treinamento
NomArqRede = strcat('Net',int2str(ContNeuro));
NomArqRede = strcat(NomArqRede,'Tr');
%*** Nome do arquivo de exportação txt => Sequencia do Treinamento
ArqRedeTre = strcat(NomArqRede,int2str(ContTreina));

load(strcat(NomFase,ArqRedeTre,'F'),'prede','tr');

% Test the Network
outputs = prede(inputs);
errors = gsubtract(targets,outputs);
performance = perform(prede,targets,outputs);

% Test the Network
outputs = prede(inputs);
errors = gsubtract(targets,outputs);
performance = perform(prede,targets,outputs);

% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets .* tr.valMask{1};
testTargets = targets .* tr.testMask{1};
trainPerformance = perform(prede,trainTargets,outputs);
valPerformance = perform(prede,valTargets,outputs);
testPerformance = perform(prede,testTargets,outputs);

trOut = outputs(tr.trainInd);
vOut = outputs(tr.valInd);
tsOut = outputs(tr.testInd);
trTarg = targets(tr.trainInd);
vTarg = targets(tr.valInd);
tsTarg = targets(tr.testInd);

plotperform(tr);

Titulo = strcat('Neuronios:',int2str(ContNeuro), '
Net:',int2str(ContTreina), ' .');
h=get(gca,'Title');
t=get(h,'String'); %t is now 'Sin(x)'

title([strcat('\newline \fontname{Arial}\fontsize{14}',t),strcat('\newline
\fontname{Arial}\fontsize{14}',Titulo)]);

```

```
%xlabel('Ciclos');
%ylabel('Performance');
%legend('Treinamento');
%text(ciclobom,tr.perf(TamF),'X');
%Imprimindo graficos
print('-djpeg75', strcat(NomeFaseSave,ArqRedeTre));

        saveas (plotregression( trTarg,trOut,'Training',...
                                vTarg,vOut,'Validation',...
                                tsTarg,tsOut,'Test', ...
                                targets,outputs, 'All'), ...

strcat (NomeFaseSave,ArqRedeTre, 'Reg'), 'png');
```



## 7.4. Apêndice D – Tabelas para auxiliar na discussão dos dados.

TABELA 10 - Dados utilizados para treinamento, validação e teste

Amostras	C2	C3	D2	D3	A	R	V	G	PMS	Prod
1	0	0	0	0	109,0600	8,2000	58,6000	2,0597	17,4477	4583,6193
2	0	0	0	0	99,0600	11,3000	96,6000	2,1242	28,3721	4948,2466
3	0	0	0	0	102,9400	8,0000	63,1000	2,0095	19,3184	4102,1748
4	0	0	1	0	92,9400	7,3000	77,9000	1,9576	23,2495	3935,9349
5	0	0	1	0	99,8837	8,0630	71,0519	2,0278	21,8624	3356,8249
6	0	0	1	0	101,3667	8,0000	68,5556	2,0324	21,2042	3595,4972
7	0	0	1	0	105,3444	8,8889	66,7000	2,1049	21,1336	4006,9523
8	0	0	0	1	97,8600	6,3000	65,8000	1,8906	18,0024	4746,4642
9	0	0	0	1	107,7500	5,1000	40,1000	2,0125	11,6918	2958,7187
10	0	0	0	1	108,3200	7,3000	58,9000	1,9117	15,7504	4180,1203
11	0	0	0	0	100,8300	9,3000	102,6000	2,0965	28,1552	4165,6956
12	0	0	0	0	104,4000	7,9000	82,5000	1,8170	25,2512	4267,8572
13	0	0	0	0	108,4100	9,9000	97,1000	2,1256	29,1814	3769,5364
14	0	0	1	0	104,8000	7,1000	45,9000	2,0675	13,4266	4352,3912
15	0	0	1	0	112,1800	6,2000	59,8000	1,9866	17,9633	4565,5695
16	0	0	0	1	110,9400	6,4000	48,6000	1,9691	13,4051	4520,4690
17	0	0	0	1	106,8300	4,7000	45,3000	1,9890	10,0030	4259,0900
18	0	0	0	1	100,3800	7,9000	55,4000	2,0397	15,6752	3102,7822
19	0	0	0	1	107,6300	6,6000	57,1000	2,0140	17,3336	4502,9672
20	1	0	0	0	85,3000	3,9000	48,8000	2,4918	16,5258	4276,6082
21	1	0	0	0	78,8000	4,5000	73,2000	2,5669	25,5224	4216,4524
22	1	0	0	0	90,3000	4,4000	50,7000	2,6686	20,0224	4956,1004
23	1	0	0	0	85,3000	4,3000	59,6000	2,6443	22,1864	4282,9714
24	1	0	1	0	85,9000	3,7000	47,8000	2,4163	17,9658	4530,3036
25	1	0	1	0	88,1000	2,7000	28,9000	2,5190	11,2124	3983,5999
26	1	0	1	0	84,1000	3,0000	33,6000	2,5357	12,8683	4575,8761
27	1	0	1	0	88,1000	3,3000	36,6000	2,5191	13,4682	4838,6236
28	1	0	0	1	86,5000	3,3000	35,6000	2,5169	12,4760	3904,4942
29	1	0	0	1	85,3000	3,4444	41,1111	2,4514	14,0049	4428,6200
30	1	0	0	1	91,2000	2,9000	37,2000	2,5430	13,4639	3734,1406
31	1	0	0	0	79,8000	4,3000	53,0000	2,5830	19,9368	4839,0613
32	1	0	0	0	82,4000	4,7000	82,5000	2,0133	23,0307	3993,4726
33	1	0	0	0	85,2000	5,2000	81,2000	2,6305	29,6196	5248,1746
34	1	0	0	0	85,0000	4,5000	56,3000	2,6536	20,4464	4800,3731
35	1	0	1	0	85,6000	3,5000	50,8000	2,5492	16,8703	5132,5783
36	1	0	1	0	88,4000	2,7778	39,4444	2,4761	13,2592	4888,0124
37	1	0	1	0	86,3000	3,7000	50,3000	2,6103	18,5827	4603,3998
38	1	0	1	0	90,4000	3,1000	34,2000	2,4912	12,1517	5209,3852
39	1	0	0	1	92,1000	2,0000	27,6000	2,6486	10,7926	3587,4764
40	1	0	0	1	79,9000	2,6000	34,9000	2,4928	12,0289	4227,0303

TABELA 10 Cont.

41	1	0	0	1	90,3000	3,0000	35,3000	2,5439	13,1354	4527,2972
42	0	1	0	0	84,4000	4,8000	78,6000	1,9962	20,5371	4160,4512
43	0	1	0	0	89,7000	3,5000	61,5000	2,0000	20,4250	4060,9330
44	0	1	0	0	86,4000	5,4000	119,8000	1,9382	29,9177	3735,7576
45	0	1	0	0	88,0000	3,5000	69,6000	2,0402	18,6507	4925,4497
46	0	1	1	0	83,4000	3,8000	80,0000	1,9738	23,1118	4259,8288
47	0	1	1	0	87,6000	3,8000	62,1000	1,9775	17,2947	4020,9135
48	0	1	1	0	88,2000	3,3333	61,9000	2,0598	18,9816	4563,6039
49	0	1	1	0	93,4300	3,2000	56,4000	1,8972	13,8921	4847,7778
50	0	1	0	1	85,3000	3,7000	51,7000	1,9961	15,5092	4009,5365
51	0	1	0	1	96,7000	3,0000	39,4000	2,0000	11,9762	4540,5298
52	0	1	0	1	89,3000	3,2000	57,4000	1,9216	13,8882	4052,9568
53	0	1	0	1	93,7000	2,8000	46,2000	1,9048	11,2136	3862,9820
54	0	1	0	0	82,3000	5,0000	90,7000	1,9735	30,5683	4919,7433
55	0	1	0	0	84,5000	3,6000	71,7000	1,9902	21,1502	4451,8542
56	0	1	0	0	88,4000	4,5000	87,8000	1,9544	22,3609	4385,3803
57	0	1	0	0	89,1000	4,5000	65,9000	2,5175	22,5710	4310,2460
58	0	1	1	0	87,2000	3,7000	69,4000	1,9971	21,5377	4518,7312
59	0	1	1	0	89,5000	2,6000	50,8000	1,9291	13,1772	4214,2334
60	0	1	1	0	95,1300	4,1000	66,7000	1,9835	17,3689	4866,6205
61	0	1	1	0	86,5000	4,4000	84,6000	1,9976	22,1533	4782,4265
62	0	1	0	1	85,6000	4,2000	58,7000	1,9813	15,7802	4690,1931
63	0	1	0	1	82,9000	3,3000	40,0000	1,9900	12,1093	4407,8400
64	0	1	0	1	94,0000	2,5000	47,7000	1,9937	13,4623	4414,5133
65	0	1	0	1	97,8000	3,4000	57,3000	2,0175	16,1579	4884,2180

C2, C3: identificação do hábito de crescimento conforme Tabela 3; D2, D3: identificação da densidade conforme Tabela 4; A: Altura em cm; R: ramos por planta; V: vagens por planta; G: grãos por vagem; PMS: Massa de 1000 sementes. Prod: Produtividade em kg ha<sup>-1</sup>

TABELA 11 - Erros de treinamento, dados normalizados e em kg Ha<sup>-1</sup>

Amostra	Dados Normalizados			Dados em kg Ha <sup>-1</sup>		
	Observado	Estimado	Erro	Observado	Estimado	Erro
2	0,57930	0,57868	0,00062	4948,24664	4946,70000	<b>1,54664</b>
3	0,24087	0,24627	-0,00540	4102,17478	4115,67500	-13,50022
5	-0,05727	-0,06234	0,00507	3356,82489	3344,16000	12,66489
8	0,49859	0,49360	0,00499	4746,46424	4734,00000	12,46424
9	-0,21651	-0,22458	0,00806	2958,71870	2938,55000	20,16870
10	0,27205	0,27430	-0,00226	4180,12030	4185,75000	-5,62970
12	0,30714	0,30346	0,00369	4267,85719	4258,65000	9,20719
13	0,10781	0,10679	0,00103	3769,53636	3766,97500	2,56136
15	0,42623	0,42949	-0,00326	4565,56949	4573,72500	-8,15551
16	0,40819	0,40564	0,00254	4520,46904	4514,10000	6,36904
17	0,30364	0,30807	-0,00443	4259,09000	4270,17500	-11,08500
19	0,40119	0,40743	-0,00624	4502,96723	4518,57500	-15,60777
20	0,31064	0,31423	-0,00358	4276,60821	4285,57500	-8,96679
21	0,28658	0,28408	0,00250	4216,45236	4210,20000	6,25236
22	0,58244	0,57476	0,00768	4956,10039	4936,90000	19,20039
23	0,31319	0,32188	-0,00869	4282,97137	4304,70000	-21,72863
24	0,41212	0,41921	-0,00709	4530,30356	4548,02500	-17,72144
25	0,19344	0,21284	-0,01940	3983,59990	4032,10000	-48,50010
28	0,16180	0,16483	-0,00304	3904,49422	3912,07500	-7,58078
31	0,53562	0,52990	0,00572	4839,06133	4824,75000	14,31133
35	0,65303	0,65642	-0,00339	5132,57834	5141,05000	-8,47166
36	0,55520	0,56930	-0,01410	4888,01241	4923,25000	-35,23759
37	0,44136	0,44199	-0,00063	4603,39978	4604,97500	-1,57522
38	0,68375	0,63178	0,05197	5209,38522	5079,45000	129,93522
39	0,03499	0,03546	-0,00046	3587,47636	3588,63750	-1,16114
40	0,29081	0,28538	0,00544	4227,03028	4213,45000	13,58028
43	0,22437	0,23910	-0,01473	4060,93296	4097,75000	-36,81704
44	0,09430	0,09213	0,00217	3735,75756	3730,32250	5,43506
45	0,57018	0,55925	0,01093	4925,44973	4898,12500	27,32473
46	0,30393	0,39050	-0,08657	4259,82880	4476,25000	<b>-216,42120</b>
47	0,20837	0,23786	-0,02949	4020,91354	4094,65000	-73,73646
50	0,20381	0,20711	-0,00330	4009,53647	4017,77500	-8,23853
51	0,41621	0,41435	0,00187	4540,52981	4535,87500	4,65481
53	0,14519	0,13756	0,00763	3862,98204	3843,90000	19,08204
54	0,56790	0,56506	0,00284	4919,74332	4912,65000	7,09332
55	0,38074	0,35359	0,02716	4451,85422	4383,97500	67,87922
56	0,35415	0,37397	-0,01982	4385,38029	4434,92500	-49,54471
57	0,32410	0,32924	-0,00515	4310,24604	4323,10000	-12,85396
58	0,40749	0,34037	0,06712	4518,73119	4350,92500	167,80619
61	0,51297	0,46766	0,04531	4782,42649	4669,15000	113,27649
64	0,36581	0,37662	-0,01081	4414,51330	4441,55000	-27,03670
65	0,55369	0,54661	0,00708	4884,21801	4866,52500	17,69301

TABELA 12 - Erros de validação, dados normalizados e em kg Ha<sup>-1</sup>

Amostra	Dados Normalizados			Dados em kg Ha <sup>-1</sup>		
	Observado	Erro	Estimado	Observado	Estimado	Erro
1	0,43345	0,35957	0,07388	4583,61931	4398,92500	184,69431
4	0,17437	0,20721	-0,03284	3935,93485	4018,02500	-82,09015
7	0,20278	-0,29575	0,49853	4006,95227	2760,62500	<b>1246,32727</b>
11	0,26628	0,01544	0,25083	4165,69564	3538,61000	627,08564
18	-0,15889	-0,30653	0,14764	3102,78216	2733,67500	369,10716
26	0,43035	0,36069	0,06966	4575,87611	4401,72500	174,15111
27	0,53545	0,63367	-0,09822	4838,62360	5084,17500	-245,55140
30	0,09366	-0,00095	0,09461	3734,14060	3497,61598	236,52462
33	0,69927	0,47568	0,22359	5248,17463	4689,20000	558,97463
34	0,52015	0,66086	-0,14071	4800,37313	5152,15000	-351,77687
42	0,26418	-0,01433	0,27851	4160,45117	3464,16750	696,28367
48	0,42544	0,41527	0,01017	4563,60391	4538,17500	<b>25,42891</b>
49	0,53911	0,56320	-0,02409	4847,77780	4908,00000	-60,22220
52	0,22118	0,16020	0,06098	4052,95676	3900,50000	152,45676
59	0,28569	0,42053	-0,13483	4214,23340	4551,32500	-337,09160
60	0,54665	0,58711	-0,04046	4866,62050	4967,77500	-101,15450

TABELA 13 - Erros de teste, dados normalizados e em kg Ha<sup>-1</sup>

Amostra	Dados Normalizados			Dados em kg Ha <sup>-1</sup>		
	Observado	Estimado	Erro	Observado	Estimado	Erro
6	0,03820	-0,08978	0,12798	3595,49716	3275,55750	319,93966
14	0,34096	-0,21830	0,55926	4352,39120	2954,25000	1398,14120
29	0,37145	0,11315	0,25830	4428,62004	3782,87500	645,74504
32	0,19739	0,53388	-0,33649	3993,47257	4834,70000	-841,22743
41	0,41092	-0,03521	0,44612	4527,29724	3411,98750	1115,30974
62	0,47608	0,02567	0,45041	4690,19307	3564,16500	1126,02807
63	0,36314	-0,05367	0,41681	4407,83997	3365,81750	1042,02247