

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

Luciana Gomes

**Modelagem de variáveis qualitativas por meio de redes neurais artificiais:
Avaliação do uso de Análise de Correspondência como técnica de codificação**

Juiz de Fora

2012

Luciana Gomes

**Modelagem de variáveis qualitativas por meio de redes neurais artificiais:
Avaliação do uso de Análise de Correspondência como técnica de codificação**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre.

Orientador: Prof. DSc. Henrique Steinherz Hippert

Juiz de Fora

2012

Gomes, Luciana.

Modelagem de variáveis qualitativas por meio de redes neurais artificiais: avaliação do uso de análise de correspondência como técnica de codificação / Luciana Gomes. – 2012.

77 f. : il.

Dissertação (Mestrado em Modelagem Computacional)–Universidade Federal de Juiz de Fora, Juiz de Fora, 2012.

1. Redes neurais (Computação). 2. Análise de correspondência. 3. I. Título.

CDU 004.032.26

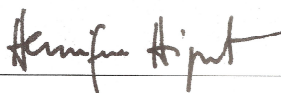
Luciana Gomes

**Modelagem de variáveis qualitativas por meio de redes neurais artificiais:
Avaliação do uso de Análise de Correspondência como técnica de codificação**

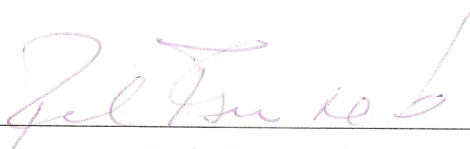
Dissertação apresentada ao Programa de Pós-
graduação em Modelagem Computacional da
Universidade Federal de Juiz de Fora como
requisito parcial à obtenção do grau de Mestre.

Aprovada em 15 de agosto de 2012.

BANCA EXAMINADORA



Prof. DSc. Henrique Steinherz Hippert – Orientador
Universidade Federal de Juiz de Fora



Prof. DSc. Raul Fonseca Neto
Universidade Federal de Juiz de Fora



Prof. DSc. Marcelo Costa Pinto e Santos
Instituto Federal do Sudeste de Minas Gerais

À minha mãe e a mim.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus por ter me dado discernimento quando mais precisei.

Aos colegas Carlos Henrique Flores; Erika Araújo; Rafael Ribeiro; Marcelo Ribeiro e Accacio Ferreira que tanto me ajudaram no decorrer do curso.

Aos amigos Shimenni Castro e Omar Souza pelo apoio quando mais precisei.

Ao meu namorado, Guilherme Teixeira, que apesar da distância sempre esteve presente em todos os momentos.

À minha mãe pelo apoio durante o curso.

Ao professor Ronaldo Rocha Bastos por sua ajuda no desenvolvimento do trabalho.

Ao meu orientador, professor Henrique Steinerz Hippert, por ter aceitado assumir uma orientação em um curto espaço de tempo, não ter em momento algum desistido e ter acreditado que eu conseguiria alcançar bons resultados apesar do tempo.

E a CAPES pelo financiamento deste trabalho.

“Mais importante que adquirir uma grande
sabedoria é a humildade na hora de transmiti-la.”

Anônimo

“Não sabendo que era
impossível, foi lá e fez.”

Jean Cocteau

“Se quiser por à prova o caráter
de um homem, dê-lhe poder.”

Abraham Lincoln

RESUMO

Em muitos problemas é necessária a descrição qualitativa dos dados (por exemplo, usando variáveis tais como sexo ou idade de um paciente). Para uso em redes neurais artificiais, contudo, estas variáveis têm que ser recodificadas quantitativamente. Neste trabalho, foram feitas simulações com seis técnicas bastante conhecidas de recodificação de variáveis qualitativas: Dummy 1-de-c, Dummy 1-de-(c-1), Termômetro, Numérica, Gray e Binária. O desempenho das seis técnicas foi comparado com o desempenho obtido utilizando-se os fatores de Análise de Correspondência (AC) ao invés das variáveis qualitativas originais. O uso destes fatores de AC como forma de codificar variáveis de entrada de uma rede neural ainda não foi relatado na literatura. As simulações foram feitas com três bases de dados. Duas delas envolvem problemas de classificação de padrões em duas classes (o desempenho foi medido por meio da proporção de classificações corretas); a terceira base envolve um problema de aproximação de funções (o desempenho foi medido por meio dos erros MAPE e MSE). Nas bases de dados Seguros e Consumo, os resultados obtidos para AC são equivalentes aos das demais técnicas aplicadas e na base Córneas, não foi satisfatório, não demonstrando, assim, vantagens sobre as demais técnicas.

Palavras-chave: Redes neurais artificiais. Análise de Correspondência. Variáveis qualitativas.

ABSTRACT

In many problems, a qualitative description of the data is needed (using for example variables such as age or sex of a patient). For use on artificial neural networks, however, these variables must be recoded quantitatively. In this study, simulations were made with six well-known techniques for recoding qualitative variables: Dummy 1-of-c, Dummy 1-of-(c-1), Thermometer, Numerical, Gray, and Binary. The performance of these techniques was compared to the performance obtained using the factors of Correspondence Analysis (CA) instead of the original qualitative variables. The use of these factors as inputs to the neural network has not been reported in the literature. Simulations were made with three dataset. Two of them involve classification problems, with two classes (performance was measured by the percentage of correct classifications); the third dataset involves a problem of function approximation (performance was measured by MAPE and MSE). For two of the datasets, the results for CA are equivalent to those of the other techniques applied; for the third, the performance was not satisfactory, do not showing any advantages over other techniques.

Keywords: Artificial neural networks. Correspondence Analysis. Qualitative variables.

SUMÁRIO

1	INTRODUÇÃO	4
2	BREVE INTRODUÇÃO SOBRE REDES NEURAIIS ARTIFICIAIS	5
2.1	MODELO BÁSICO DE RNA COM ARQUITETURA EM CAMADAS MÚLTIPLAS	5
2.2	TREINAMENTO DAS RNAS	8
2.3	PRINCIPAIS APLICAÇÕES DAS RNAS	10
3	MATERIAIS E MÉTODOS	11
3.1	PROGRAMAS E COMPUTADORES.....	11
3.2	BANCOS DE DADOS	11
3.2.1	<i>Banco de Dados I: Córneas</i>	11
3.2.2	<i>Banco de Dados II: Seguros</i>	13
3.2.3	<i>Banco de Dados III – Consumo</i>	14
3.3	REDES NEURAIIS ARTIFICIAIS	15
3.3.1	<i>Arquitetura: Multilayer Perceptron</i>	15
3.3.2	<i>Treinamento</i>	15
3.3.3	<i>Seleção de variáveis de entrada</i>	16
3.4	FORMAS DE CODIFICAÇÃO DE VARIÁVEIS DE ENTRADA	17
3.4.1	<i>Codificação Dummy</i>	17
3.4.2	<i>Codificação Termômetro</i>	18
3.4.3	<i>Codificação Binária</i>	19
3.4.4	<i>Codificação Gray</i>	19
3.4.5	<i>Codificação Numérica</i>	21
3.4.6	<i>Análise de Correspondência</i>	22
4	RESULTADOS	28
4.1	SIMULAÇÕES FEITAS PARA DEFINIÇÃO DAS ARQUITETURAS DAS REDES	28
4.1.1	<i>Banco de Dados Córneas</i>	28
4.1.2	<i>Banco de Dados Seguros</i>	37
4.1.3	<i>Banco de Dados Consumo</i>	49
4.2	COMPARAÇÃO FINAL DOS MODELOS E DISCUSSÃO	57
4.2.1	<i>Banco de Dados Córneas</i>	57
4.2.2	<i>Banco de dados Seguros</i>	59
4.2.3	<i>Banco de dados Consumo</i>	61
5	DISCUSSÃO E CONCLUSÃO	64
	REFERÊNCIAS	65

LISTA DE FIGURAS

FIGURA 2-1 – MODELO DO NEURÔNIO ARTIFICIAL.	5
FIGURA 2-2 – MODELO DO PERCEPTRON.....	6
FIGURA 2-3 - <i>MULTILAYER PERCEPTRON</i> COM DUAS CAMADAS.....	6
FIGURA 2-4 - FUNÇÕES DE ATIVAÇÃO TANGENTE HIPERBÓLICA, LOGARÍTMICA E LINEAR.	7
FIGURA 3-1 – (A) CODIFICAÇÃO BINÁRIO-GRAY; (B) GRAY-BINÁRIO (TOCCI ET AL., 2007).....	21
FIGURA 3-2 – VARIÁVEIS CATEGÓRICAS.	22
FIGURA 3-3 - ANÁLISE DE CORRESPONDÊNCIA MÚLTIPLA DE CARACTERÍSTICAS DOS DOMICÍLIOS	25
FIGURA 4-1 - NÚMEROS DE CLASSIFICAÇÕES CORRETAS, <i>DUMMY</i> 1-DE-C, UMA CAMADA OCULTA.	29
FIGURA 4-2 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, <i>DUMMY</i> 1-DE-C, DUAS CAMADAS OCULTAS.	30
FIGURA 4-3 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, <i>DUMMY</i> 1-DE-(C-1), UMA CAMADA.	31
FIGURA 4-4 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, TERMÔMETRO, UMA CAMADA OCULTA.	32
FIGURA 4-5 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO BINÁRIA, UMA CAMADA OCULTA.	33
FIGURA 4-6 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>GRAY</i> , UMA CAMADA OCULTA.	34
FIGURA 4-7 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO NUMÉRICA,	35
FIGURA 4-8 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>AC</i> , UMA CAMADA OCULTA.	36
FIGURA 4-9 – NÚMEROS DE CLASSIFICAÇÕES CORRETAS, <i>DUMMY</i> 1-DE-C, UMA CAMADA OCULTA.	38
FIGURA 4-10 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, <i>DUMMY</i> 1-DE-C, DUAS CAMADAS OCULTAS.	39
FIGURA 4-11 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, <i>DUMMY</i> 1-DE-(C-1), UMA CAMADA.	40
FIGURA 4-12 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, TERMÔMETRO, UMA CAMADA OCULTA.	41
FIGURA 4-13 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO BINÁRIA, UMA CAMADA OCULTA.	42
FIGURA 4-14 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>GRAY</i> , UMA CAMADA OCULTA.....	43
FIGURA 4-15 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO NUMÉRICA,	44
FIGURA 4-16 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>AC</i> , UMA CAMADA OCULTA.....	45
FIGURA 4-18 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>AC</i> , DUAS CAMADAS OCULTAS....	47
FIGURA 4-19 - NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>AC</i> , DUAS CAMADAS OCULTAS (ZOOM).	47
FIGURA 4-20 – NÚMEROS DE CLASSIFICAÇÕES CORRETAS, <i>DUMMY</i> 1-DE-C, UMA CAMADA OCULTA.	50
FIGURA 4-21 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, TERMÔMETRO, UMA CAMADA OCULTA.	51
FIGURA 4-22 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO BINÁRIA, UMA CAMADA OCULTA.	52
FIGURA 4-23 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>GRAY</i> , UMA CAMADA OCULTA.....	53
FIGURA 4-24 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO NUMÉRICA,	54
FIGURA 4-25 – NÚMERO DE CLASSIFICAÇÕES CORRETAS, CODIFICAÇÃO <i>AC</i> , UMA CAMADA OCULTA.....	55
FIGURA 4-26 – COMPARAÇÃO ENTRE AS CODIFICAÇÕES - <i>CÓRNEAS</i>	58
FIGURA 4-27 – COMPARAÇÃO ENTRE AS CODIFICAÇÕES - <i>SEGUROS</i>	60
FIGURA 4-28 – COMPARAÇÃO ENTRE AS CODIFICAÇÕES DO BD CONSUMO – ERRO MAPE.	62
FIGURA 4-29 – COMPARAÇÃO ENTRE CODIFICAÇÕES DO BD CONSUMO – ERRO MSE.	63

LISTA DE TABELAS

TABELA 3-1 – DESCRIÇÃO DO BANCO DE DADOS CÓRNEAS (CYRINO, 2011).....	12
TABELA 3-2 - DESCRIÇÃO DO BANCO DE DADOS SEGUROS (GREENACRE & BLASIU, 2006).	13
TABELA 3-3 – DESCRIÇÃO DO BANCO DE DADOS CONSUMO.	14
TABELA 3-4 - CODIFICAÇÃO <i>DUMMY</i> DAS VARIÁVEIS CATEGÓRICAS.	18
TABELA 3-5 - CODIFICAÇÃO TERMÔMETRO.	18
TABELA 3-6 - CODIFICAÇÃO BINÁRIA.	19
TABELA 3-7 – EXEMPLO DE DISTÂNCIA <i>HAMMING</i> NAS CODIFICAÇÕES BINÁRIO E <i>GRAY</i> (KAK, 2010).....	20
TABELA 3-8 – TABELA DE CONTINGÊNCIA GENÉRICA (MINGOTI, 2005).....	22
TABELA 4-1 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	29
TABELA 4-2 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	30
TABELA 4-3 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	31
TABELA 4-4 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	32
TABELA 4-5 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	33
TABELA 4-6 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	34
TABELA 4-7 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	35
TABELA 4-8 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	36
TABELA 4-9 - NÚMERO DE NEURÔNIOS ESCOLHIDOS PARA CADA CODIFICAÇÃO.	37
TABELA 4-10 – NÚMERO DE CLASSIFICAÇÕES OBTIDAS EM 100 SIMULAÇÕES DE CADA REDE.	38
TABELA 4-11 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	39
TABELA 4-12 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	40
TABELA 4-13 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	41
TABELA 4-14 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	42
TABELA 4-15 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	43
TABELA 4-16 – CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.....	44
TABELA 4-17 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	46
TABELA 4-18 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	48
TABELA 4-19 – NÚMERO DE NEURÔNIOS ESCOLHIDOS PARA CADA CODIFICAÇÃO.....	48
TABELA 4-20 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	50
TABELA 4-21 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	51
TABELA 4-22 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	52
TABELA 4-23 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	53
TABELA 4-24 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	54
TABELA 4-25 - CLASSIFICAÇÕES CORRETAS EM 100 SIMULAÇÕES DE CADA REDE.	55
TABELA 4-26 - NÚMERO DE NEURÔNIOS ESCOLHIDOS PARA	56
TABELA 4-27 - NÚMERO DE NEURÔNIOS ESCOLHIDOS.....	56
TABELA 4-28 – COMPARAÇÃO ENTRE AS CODIFICAÇÕES –.....	58
TABELA 4-29 – COMPARAÇÃO ENTRE AS CODIFICAÇÕES –.....	60
TABELA 4-30 – COMPARAÇÃO ENTRE AS CODIFICAÇÕES –.....	62
TABELA 4-31 – COMPARAÇÃO ENTRE AS CODIFICAÇÕES –.....	63

1 Introdução

Diversos métodos científicos utilizam informações advindas de dados quantitativos, como é o caso das Redes Neurais Artificiais (RNAs). Em muitas situações, porém, se faz necessária à utilização de variáveis qualitativas para a definição dos dados; um exemplo disso seria uma pesquisa de indicadores sócio-econômicos com o campo “classe social” que poderia ser preenchido por A, B, C, D ou E (Coutto Filho *et al.*, 2007). A entrada de uma RNA não conseguiria interpretar dados deste tipo, e é, portanto, necessário encontrar outra representação para essas variáveis. Em diversas áreas do conhecimento, como *marketing*, engenharia, ciências sociais, entre outras, foram criadas diferentes técnicas de recodificação dessas variáveis, cada uma das quais criadas para atender às necessidades que apareciam nas interpretações de dados em seus problemas específicos. Algumas das formas mais difundidas de codificações desenvolvidas para essas áreas foram: *dummy*, termômetro, numérica, *Gray* e binária. Mais recentemente, tem sido desenvolvida uma técnica chamada *Análise de Correspondência* (AC), a qual consegue representar uma matriz de dados qualitativos por meio de vetores de fatores quantitativos de menor dimensionalidade. Um dos criadores desta técnica, Greenacre, sugere que estes fatores podem ser usados como entrada em algoritmos que empregam variáveis quantitativas: “O uso de fatores coordenados também provém uma chave para a aplicação de metodologias designadas por preditores numéricos, tais como uma máquina de vetor suporte ou uma rede neural, para dados categóricos” (Greenacre *et al.*, 2006).

Este trabalho visa avaliar o desempenho preditivo de modelos que utilizam diferentes técnicas de codificação numérica das variáveis qualitativas, e compará-lo com modelos que usam os fatores obtidos por AC para a avaliação da técnica como forma de codificação de variáveis qualitativas. O desenvolvimento deste trabalho ocorreu através de testes e avaliação de desempenho, de cada codificação, em problemas de classificação de três bases de dados de origens distintas. Duas destas bases de dados estão relacionadas a problemas de classificação, e o desempenho foi medido em termos da porcentagem de acertos; a terceira base está relacionada a um problema de aproximação de funções, e o desempenho foi medido pelo erro médio quadrático (MSE) e pelo erro percentual absoluto médio (MAPE).

2 Breve Introdução Sobre Redes Neurais Artificiais

2.1 Modelo básico de RNA com arquitetura em camadas múltiplas

Apresentamos nesta seção uma breve introdução sobre as RNAs, baseada em Braga *et al* (2000). As RNAs são modelos cuja estrutura procurou (pelo menos, inicialmente) representar a organização de um cérebro humano. Há vários tipos de RNAs, mas todos eles compõem-se de unidades de processamento simples (chamadas de *neurônios*), organizadas em camadas e interligadas por conexões geralmente associadas a pesos que possuem a finalidade de ponderar cada variável de entrada.

A Figura 2-1 mostra um modelo de neurônio artificial. As variáveis de entrada (x_1 a x_4 , no exemplo) são combinadas linearmente, por meio dos pesos w_i , e uma constante θ é adicionada. O resultado desta combinação é usado como argumento de uma função, chamada de *função de ativação*, que pode ser linear ou não.

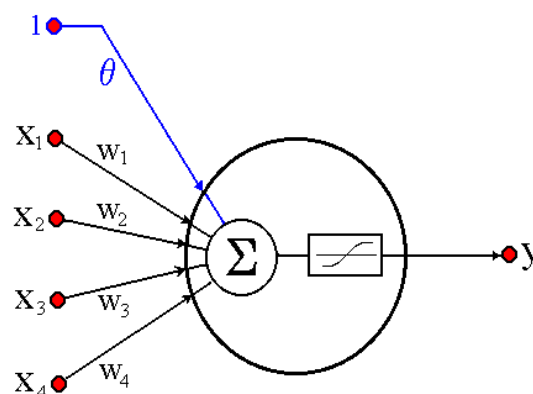


Figura 2-1 – Modelo do neurônio artificial.

O primeiro modelo de neurônio artificial foi desenvolvido por Warren McCulloch e Walter Pitts e publicado em 1943, num trabalho que discute redes lógicas de nós (MCP). Em 1949, Donald Hebb desenvolve a *Regra de Hebb* que explica o aprendizado em nós biológicos com base no esforço das ligações sinápticas entre nós excitados. Alguns anos depois, Widrow e Hoff desenvolvem a *Regra Delta* e em 1958, Frank Rosenblatt apresenta o modelo do *Perceptron*, onde os neurônios se encontravam organizados em uma única camada, conectados diretamente aos nós de entrada e aos de saída (Figura 2-2).

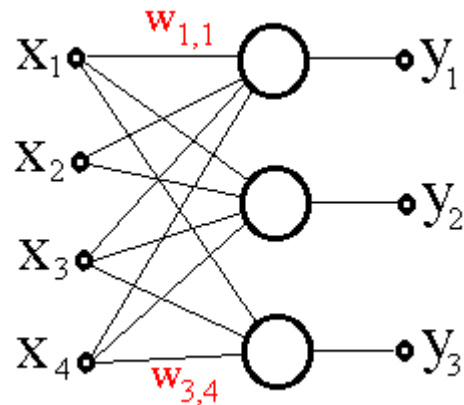


Figura 2-2 – Modelo do Perceptron.

O *Perceptron* de camada única desenvolvido por Rosenblatt se limitava a resolução de problemas lineares de classificação. O perceptron da figura acima, por exemplo, poderia ser usado para separar linearmente os padrões de entrada em 3 classes. Observou-se então que um *Perceptron* com multicamadas resolveria com eficiência problemas mais complexos, não-lineares (Gomes, 2006). A Figura 2-3 mostra um exemplo de um *multilayer perceptron* (MLP) com duas camadas.

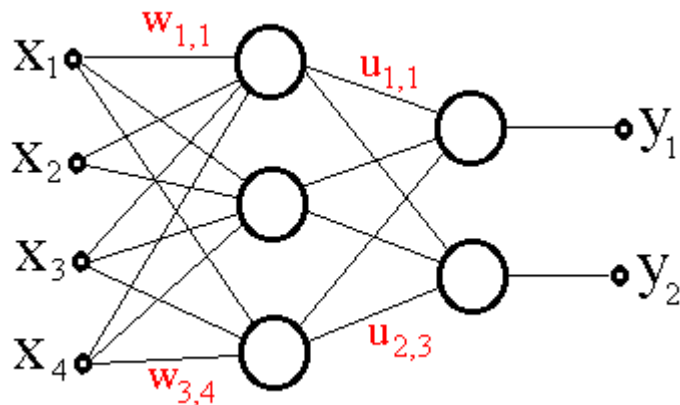
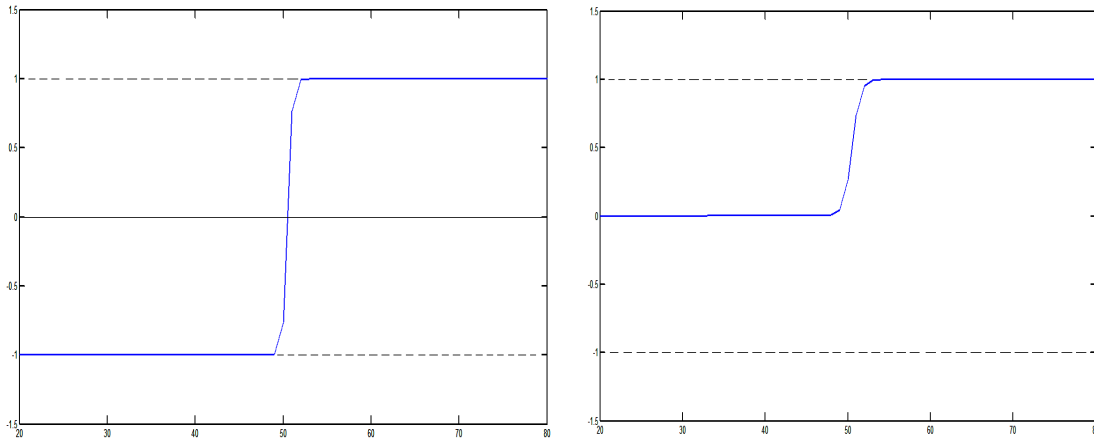


Figura 2-3 - *Multilayer Percetron* com duas camadas.

O MLP é baseado no arranjo dos neurônios em camadas e na retropropagação do sinal de entrada. Este tipo de arquitetura é classificado como *feed-forward*, pois a propagação de sinal ocorre apenas no sentido positivo, ou seja, a saída da camada anterior é a entrada da camada seguinte. Neste tipo de arquitetura, as camadas intermediárias entre a entrada e a

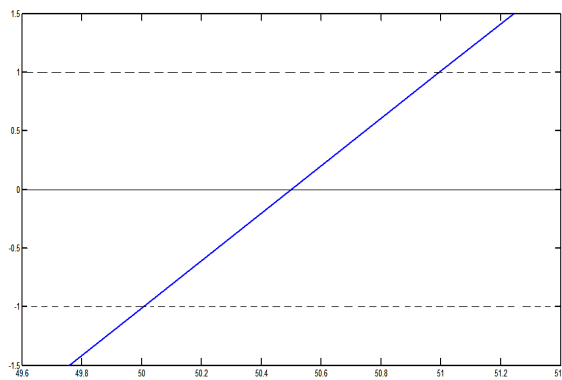
saída são denominadas ocultas, sendo todas idênticas umas as outras. Este tipo de arquitetura foi utilizado em todas as simulações que fizemos neste trabalho.

Funções de ativação sigmoidais podem ser usadas nos neurônios da camada de saída, em problemas de reconhecimento de padrões (onde a saída da rede é geralmente binária). Funções lineares são geralmente utilizadas em problemas de aproximação de funções (Demuth, 2008; Cerqueira et al, 2001).



$$f(net) = \frac{1 - e^{-net}}{1 + e^{-net}} \quad (2-1)$$

$$f(net) = \frac{1}{1 + e^{-(net)}} \quad (2-2)$$



$$f(net) = a(net) + b \quad a, b \in \mathbb{R}_+^* \quad (2-3)$$

Figura 2-4 - Funções de ativação tangente hiperbólica, logarítmica e linear.

2.2 Treinamento das RNAs

As RNA do tipo *perceptron* têm, inicialmente, uma fase de aprendizado, em que são apresentados exemplos à rede e são retiradas as características que serão utilizadas na fase de geração de resultados, para a estimação dos parâmetros da rede (os pesos que ligam os neurônios entre si e às variáveis de entrada). A dificuldade, contudo, foi a falta de um algoritmo de treinamento, que permitisse esta estimação (em termos estatísticos, da falta de um método para estimar os parâmetros da rede). Os estudos de RNAs ficam adormecidos durante alguns anos devido a publicações pessimistas sobre o assunto. Em 1982, porém, John Hopfield publica um artigo que mostra a relação entre sistemas físicos e redes recorrentes auto-associativas, o que contribui para a retomada dos estudos na área (Braga *et al.*, 2000). Anos depois, desenvolve-se o algoritmo de treinamento *back-propagation* (Braga *et al.*, 2000) e a partir daí é crescente o interesse por pesquisas na área de redes neurais artificiais.

Atualmente, porém, o algoritmo mais utilizado em redes *feedforward backpropagation*, é o de Levenberg-Marquardt (LM), por sua rapidez. Este método foi desenvolvido para que se pudesse solucionar, através do método dos mínimos quadrados, problemas de mínimo de funções não-lineares. A diferença entre o LM e o *backpropagation* tradicional é que no primeiro a aceleração do treinamento se dá pela resolução das derivadas segundas do erro quadrático em relação ao peso, enquanto que o segundo considera as de primeira ordem (Barbosa *et al.*, 2005; Winandy *et al.*, 2007).

Levenberg enunciou a regra de atualização dos pesos, vista em (2-4), que considera ao mesmo tempo os métodos do *Gradiente Descendente* e o de *Gauss-Newton* (Winandy *et al.*, 2007).

$$W_{(i+1)} = W_{(i)} - [H + \lambda I]^{-1} \cdot \nabla f(W_{(i)}) \quad (2-4)$$

Onde: I é a matriz identidade, H é a matriz Hessiana e λ é a constante do método de LM.

O método *Gradiente Descendente* tem como objetivo o valor do ponto de mínimo da função-objetivo. O método busca a direção oposta do gradiente, pois este aponta para a direção de maior crescimento da função. O Gauss-Newton converge mais rapidamente que o

método anterior e atua na minimização da função-objetivo através de aproximações quadráticas feitas por expansão em série de Taylor (Gonçalves, 2011).

A constante λ , também conhecida como *fator de ajuste* da função, determina, através de seu valor, qual método irá predominar. Para λ de valores grandes, teremos o Gradiente Descendente como o mais predominante no algoritmo LM, e para valores pequenos, o Gauss-Newton. Essa constante é controlada pelo algoritmo: primeiramente atribui-se um valor arbitrário a λ , então o erro atual é calculado e atualizado os pesos; um novo erro é calculado e se este aumentar, λ também aumenta e o erro volta ao valor anterior à atualização; caso contrário λ diminui e a atualização do erro permanece (Winandy *et al.*, 2007).

Há algumas dificuldades no algoritmo LM e a principal delas ocorre quando λ apresenta valor muito alto, levando ao cálculo da Hessiana ser desprezado pelo algoritmo. Devido a esta questão, Marquardt apresentou melhorias ao algoritmo fazendo com que a parcela do Gradiente Descendente presente no mesmo considerasse as atualizações dos pesos ainda que o gradiente fosse pequeno. A partir daí, surge uma nova regra de atualização dos pesos do algoritmo LM, vista em (2-5) (Winandy *et al.*, 2007).

$$W_{(i+1)} = W_{(i)} - [H + \lambda \cdot \text{diag}[H]]^{-1} \cdot \nabla f(W_{(i)}) \quad (2-5)$$

A matriz Hessiana H do método de Gauss-Newton, (2-6), é determinada através da matriz Jacobiana que contém em suas posições as derivadas primeiras dos pesos em função dos pesos sinápticos, que pode ser vista em (2-7) (Barbosa *et al.*, 2005).

$$H = \frac{\partial^2 E_R(W)}{\partial W^2} \quad (2-6)$$

$$J = \frac{\partial e(W)}{\partial W} \quad (2-7)$$

Onde $e(W)$ é dado por (2-8):

$$e(W) = \sum_{i=1}^n (y_i - y_{ei}) \quad (2-8)$$

Este método é o mais rápido para a utilização em redes neurais com volume médio de parâmetros. Apresenta convergência com menor número de iterações, apesar do grande número de cálculos realizados (Barbosa *et al.*, 2005).

2.3 Principais aplicações das RNAs

Duas das principais aplicações das RNAs, que iremos explorar neste trabalho, são em problemas de reconhecimento de padrões e de classificação, onde as redes se mostram ferramentas extremamente flexíveis, uma vez que são métodos multivariados, não-lineares, não-paramétricos e orientados a dados. A previsão para dados quantitativos se baseia na retirada de dados de observações passadas, sua modelagem, e sua posterior extrapolação para o futuro (Hippert, 2001).

3 Materiais e métodos

3.1 Programas e computadores

Neste trabalho foi utilizado o *software* MATLAB® (*Matrix Laboratory* - Licença nº 334543) e um computador HP modelo G42-240BR com processador Intel® core TM i3-350M, 2GB RAM e de sistema operacional *Windows® 7 Home Basic*.

O MATLAB é um *software* para desenvolvimento de algoritmos e computação numérica. É usado em diversas aplicações, tais como processamento de imagem e de sinal e análise e modelagem financeira. Possui interface com as linguagens C, C++ e Fortran. Para a resolução de nossos problemas utilizamos a *Neural Network Toolbox™*, biblioteca específica que nos permite desenvolver, implementar, visualizar e simular as redes. Esta biblioteca possui suporte para redes *feedforward*, redes de base radial, entre outras (*Mathworks*, n.d.).

3.2 Bancos de dados

3.2.1 Banco de Dados I: Córneas

Este banco de dados faz parte do projeto de pesquisa “Efeito da Idade do Doador na Utilização do Tecido Corneano” do Hospital Regional Dr. João Penido realizada pelo pesquisador Rafael Vidal Mérula em conjunto com o Prof. Ronaldo Rocha Bastos (Departamento de Estatística/UFJF), o qual gentilmente nos cedeu os dados para este trabalho. Estes dados foram obtidos a partir do Banco de Olhos do Hospital Regional Dr. João Penido que possui informações de prontuários de doadores que deram a autorização para a doação em vida.

A base de dados possui 264 casos, cada um representando um possível doador. Esta foi obtida a partir de uma base inicial com 622 linhas, representando uma córnea cada, refinada de acordo com a qualidade da córnea até que se chegasse a um banco com possíveis doadores. Com o refinamento chegou-se a oito variáveis: idade; local da coleta; sexo do doador; tempo do óbito à enucleação, em horas; tempo da enucleação à preservação, em horas; tempo do óbito à preservação, em horas; qualidade da córnea e causa do óbito.

Apenas as variáveis “local da coleta”, “sexo do doador”, “qualidade da córnea” e “causa do óbito” são originalmente qualitativas. As demais variáveis passaram por processo de categorização seguindo diferentes critérios (Cyrino, 2011). A descrição das variáveis pode ser vista na Tabela 3-1.

Tabela 3-1 – Descrição do banco de dados Córneas (Cyrino, 2011).

Descrição	Categorias
Idade	Até 40 anos 41 – 50 anos 51 – 60 anos 61 – 70 anos Mais de 70 anos
Local da Coleta	Juiz de Fora Zona da Mata
Sexo do doador	Feminino Masculino
Tempo do óbito à enucleação	Até 6h Mais de 6h
Tempo da enucleação à preservação	Até 3,79h De 3,80h a 6,50h De 6,51h a 10,54h Mais de 10,54h
Tempo do óbito à preservação	Até 7,25h De 7,26h a 10,25h De 10,26h a 15h Mais de 15h
Qualidade da córnea	Aceitável Inaceitável
Causa do óbito	Doenças do aparelho circulatório Doenças do aparelho respiratório Neoplasias Causas externas de morbidade e mortalidade Outras

3.2.2 Banco de Dados II: Seguros

O segundo banco de dados corresponde a um estudo de caso retirado da literatura de (Greenacre & Blasius, 2006). O estudo consiste da observação de 1106 donos de automóveis da Bélgica que fizeram seguros no ano de 1992.

Os donos dos automóveis foram classificados em dois grupos:

- “bons” – aqueles que não usaram o seguro ($n_1 = 556$);
- “maus” – aqueles que usaram o seguro uma ou mais vezes ($n_2 = 550$);

Foram usadas nove categorias de previsão, listadas na Tabela 3-2. As variáveis *Bónus* e *Malus* são, respectivamente, a diminuição e o incremento do prêmio a ser pago conforme o número de sinistros do segurado (Lamelas, 2012).

Tabela 3-2 - Descrição do banco de dados Seguros (Greenacre & Blasius, 2006).

Descrição	Categorias
Tipo de usuário	Profissional Privado
Tipo de segurado	Masculino Feminino Empresa
Língua	Francês Flamengo
Época de nascimento	1890 – 1949 1950 – 1973 Desconhecido
Região	Bruxelas Outra
Nível de <i>Bónus-Malus</i>	B-M+ B-M (-1)
Potência [cv]	10 – 39 40 – 349
Ano de assinatura do seguro	< 86 Outros
Ano do veículo	1933 – 1989 1990 - 1991

3.2.3 Banco de Dados III – Consumo

Este banco de dados se refere a uma pesquisa do Departamento de Estatística da UFJF em 2004-2005 realizada em 577 residências, escolhidas por amostragem, para responderem a um questionário de 72 perguntas referentes às características da residência, sócio-econômicas, de consumo elétrico e de atitudes de racionamento energético dos moradores. Após a conclusão do levantamento dos dados foi realizado o registro das coordenadas (em metros) das residências participantes da pesquisa através do instrumento GPS (*Global Positioning System*), e os dados obtidos foram incorporados à pesquisa.

Neste trabalho foram selecionadas apenas algumas variáveis desta pesquisa para a manipulação dos dados, vistas na Tabela 3-3, as quais são codificadas em diferentes tipos para posterior análise de desempenho das codificações medido pelo erro médio quadrático (MSE - *mean square error*) ou erro percentual absoluto médio (MAPE - *mean absolute percentage error*). Para a análise foram utilizados 444 casos.

Tabela 3-3 – Descrição do banco de dados Consumo.

Variável	Descrição
1º grau	Grau de instrução do chefe da família: Ensino fundamental completo ou não
2º grau	
3º grau	
Empregada	Possui empregada doméstica?
Carro_d	Possui carro?
Carros	Quantidade de carros no domicílio
Tipo de residência	Tipo de residência (casa/apartamento)
Área	Área da residência [m]
Cômodos	Quantidade de cômodos
Banheiro	Número de banheiros
Residentes	Quantidade de residentes
Relógio	O medidor de energia atende mais de um domicílio?
Coord_n	Coordenada norte [m]
Coord_e	Coordenada leste [m]
Consumo	Consumo [kWh]
Monofásico	Padrão de abastecimento: Monofásico
Bifásico	
Trifásico	

Observação: A variável “Carros” foi utilizada somente na codificação AC.

3.3 Redes Neurais Artificiais

Nesta seção apresentaremos as principais características das RNAs utilizadas neste trabalho como o modelo, o algoritmo de treinamento, o critério de parada, entre outros.

3.3.1 Arquitetura: Multilayer Perceptron

Todas as redes usadas neste trabalho foram do tipo MLP, a maioria delas com apenas uma camada oculta. Fizemos alguns experimentos com redes contendo duas camadas ocultas, cujos resultados estão mostrados na seção 4.1.1.2; estas redes, contudo, não mostraram nenhuma vantagem em relação a de apenas uma camada.

O MLP permite que modelos sejam desenvolvidos utilizando diversas funções de ativação diferentes (Braga *et al.*, 2000). As funções utilizadas neste trabalho foram a sigmóide tangente hiperbólica, que possui saída variando entre -1 e 1; a sigmóide logarítmica, com saída entre 0 e 1; e a linear, que possui valor de saída igual ao de entrada (Cazes, 2005; Sampaio, 2009). O número de neurônios de cada rede foi determinado por meio de validação cruzada (ver seção 3.3.2).

3.3.2 Treinamento

O treinamento foi feito por *batches* (ou ciclo), em que atualização dos pesos ocorre somente após todos os padrões de entrada terem sido apresentados. Apesar de considerada estável, a técnica pode se mostrar lenta para um conjunto de dados redundantes (NCE-UFRJ, n.d.).

O critério de parada utilizado neste trabalho foi a Validação Cruzada (*cross validation*) - técnica estatística indicada para se evitar *overtraining* e generalizar uma rede, com grande quantidade de dados, durante o treinamento (Silva, 2005).

Para a aplicação desta técnica, os dados devem ser divididos aleatoriamente em dois conjuntos: *treino* e *teste*. O conjunto de treino é também dividido em dois subconjuntos disjuntos: *estimação* e *validação*. O primeiro é usado para a estimação dos parâmetros

(treinamento) do modelo e o segundo para a validação do mesmo. O objetivo da técnica é validar o modelo com um grupo de dados diferente daquele utilizado para a estimação dos resultados, a fim de tentar evitar o *overfitting* - ajuste excessivo do modelo aos dados de treino, o que geralmente leva a um mal resultado preditivo (Haykin, 2001).

O número de neurônios das redes também foi determinado por validação cruzada, da seguinte forma: os dados disponíveis foram separados em três partes, uma para treino da rede (60% dos dados), outra para validação (20% dos dados), e outra para teste (20%). Simulamos uma rede de cinco neurônios 100 vezes, a partir de valores iniciais aleatórios, usando os dados o conjunto de treino, até que os algoritmos convergissem; depois, verificamos os erros obtidos por estas redes nos dados de validação. A seguir, aumentamos o número de neurônios para 10, fazemos novamente 100 simulações, e verificamos os erros nos dados de validação; e assim sucessivamente, acrescentando a cada vez mais cinco neurônios à rede, até obtermos o número de neurônios da camada oculta que nos dava o melhor desempenho da rede na amostra de validação, para os dados de entrada - este procedimento foi realizado para cada uma das diferentes codificações de variáveis. Iniciamos as simulações a partir de 5 neurônios, aumentando sempre 5 neurônios a cada nova simulação da rede.

Escolhido a melhor rede (em termos do número de neurônios) para cada conjunto de entradas, realizamos novas simulações com esta rede, e avaliamos seu erro nos dados de teste, a fim de verificar qual forma de codificação das variáveis de entrada conseguiria melhor desempenho (menor erro).

3.3.3 Seleção de variáveis de entrada

Para selecionar as variáveis de entrada de cada rede, fizemos testes usando a técnica de *backward elimination*. Esta técnica procura simplificar a arquitetura de uma rede através da eliminação das variáveis de entrada que se mostrarem menos relevantes em testes empíricos. Inicialmente eliminamos uma variável e treinamos a rede sem ela; em seguida, outra variável é retirada e a rede novamente é treinada, e assim sucessivamente até que não haja mais possibilidades de eliminação. Os desempenhos destas diversas redes são comparados. Aquela variável cuja eliminação menos afetou o desempenho da rede é considerada a menos relevante, e eliminada. A seguir o processo é repetido com as variáveis restantes, eliminando-se uma a uma e comparando os resultados, até que não haja mais

variável a ser eliminada (Larsen, 2008; Steinmetz, 2009). Neste trabalho esta técnica foi aplicada, porém não obtivemos sucesso, pois os resultados mostraram que praticamente todas as variáveis eram significativas. Isto já era em parte esperado, uma vez que os bancos de dados que utilizamos já tinham sido usados antes em diversos estudos de modelagem, e as variáveis que continham já tinham sido passadas antes por diversos testes.

3.4 Formas de codificação de variáveis de entrada

Utilizamos seis diferentes formas de codificação numérica de variáveis qualitativas nos dados dos problemas propostos, as quais são: *Dummy*, Termômetro, Gray, Binário, Numérico e Análise de Correspondência. A seguir discorreremos sobre o conceito de cada uma das codificações utilizadas neste trabalho.

3.4.1 Codificação Dummy

A codificação por meio de *dummies* é uma das formas de usar variáveis qualitativas (também chamadas categóricas) em estimação de modelos. Um exemplo de seu uso seria em regressão linear (UCLA, n.d.). Variáveis *dummy* indicam presença ou ausência de atributo; na codificação em *dummies*, as variáveis originais são transformadas em artificiais assumindo valores de 0 ou 1, onde zero indica ausência de atributo e um, sua presença. Podem ser utilizadas em modelos simples, em que a variável dá origem a apenas uma *dummy*; ou complexos, no qual a variável qualitativa origina um vetor com duas ou mais *dummies* (Missio & Jacobi, 2007). Para codificações com mais de duas categorias é recomendado utilizar sempre uma variável a menos; isto é, se a variável qualitativa pode assumir C valores, deverá ser representada por um vetor de $C-1$ *dummies*. Em regressão linear, violar esta recomendação faz com que o algoritmo de mínimos quadrados não funcione, devido à multicolinearidade entre os *dummies* (Baker, 2006).

A Tabela 3-4 ilustra, como exemplo, a codificação *dummy* para a variável “Faixa Etária” do banco de dados *Córneas*.

Tabela 3-4 - Codificação *Dummy* das variáveis categóricas.

Faixa Etária (anos)	Codificação
Até 40	10000
41-50	01000
51-60	00100
61-70	00010
70 ou mais	00001

3.4.2 Codificação Termômetro

A codificação *Termômetro* é aquela em que uma variável quantitativa é transformada proporcionalmente ao seu valor em uma sequência composta por zeros e uns. As unidades são ordenadas sequencialmente de acordo com o valor original do dado, tendo a cada acréscimo em seu valor a adição de uma unidade na sequência. Se a codificação for utilizada na variável de saída em problemas de aproximação de funções, a cada nó será atribuído um valor de *threshold* diferente e a saída será 1 se o valor do *target* for maior que seu *threshold* (Jeon & Choi, 1999; Baesens *et al.*, 2003). Possui ampla utilização na área de redes neurais.

Na Tabela 3-5 é apresentado um exemplo de codificação *Termômetro* para a mesma variável do exemplo da codificação *Dummy*.

Tabela 3-5 - Codificação *Termômetro*.

Faixa Etária (anos)	Codificação
Até 40	10000
41-50	11000
51-60	11100
61-70	11110
70 ou mais	11111

3.4.3 Codificação Binária

O sistema de numeração binário é considerado o sistema mais importante em sistemas digitais, utilizando exclusivamente os dígitos 1 (um) e 0 (zero) para representar qualquer número. É uma representação posicional, pois cada dígito binário (*bit*) possui um peso, de acordo com a posição relativa ao bit menos significativo (LSB - *least significant bit*), e expresso por uma potência de dois.

Muitas vezes, se faz necessária a conversão do binário para o sistema decimal, como, por exemplo, em calculadoras, em que o usuário entra com a numeração decimal, os cálculos são feitos internamente no sistema binário e o resultado é apresentado no sistema decimal, como forma de interface amigável com o usuário (Tocci *et al.*, 2007).

Uma maneira de se converter um número decimal para a base binária é expressá-lo como soma de potências de dois e a partir daí os dígitos 1 e 0 são alocados em suas posições corretas como mostra a Equação (3-1). Outros exemplos podem ser vistos na Tabela 3-6.

$$\begin{aligned}
 45_{10} &= 32 + 8 + 4 + 1 = 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 \\
 &= 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1_2
 \end{aligned}
 \tag{3-1}$$

Tabela 3-6 - Codificação Binária.

Decimal	Binário
0	000
1	001
2	010
3	011
4	100

3.4.4 Codificação Gray

Quando um número se encontra na codificação binária, percebemos que às vezes é necessário que vários de seus *bits* mudem de estado para representar seu sucessor e foi pensando nesta mudança que poderia vir a ser erroneamente interpretada pelo sistema digital, que se criou o Código *Gray* para a representação de sistemas numéricos (Tocci *et al.*, 2007). Segundo Tun (2007) “o Código *Gray* é não-ponderado e não é uma codificação aritmética: ou

seja, não há pesos específicos atribuídos para as posições dos bits”. A importância deste código é que há apenas uma única mudança de bit entre palavras consecutivas, por exemplo, para um caso de velocidade angular do eixo de um motor nos beneficiamos com a codificação *Gray* por haver uma mudança de um único bit por palavra consecutiva se compararmos à codificação binária na qual teríamos a troca de três bits, o que representaria grande erro de cálculo no resultado final.

Um ponto importante para este sistema é a Distância *Hamming*, medida mais simples do vetor binário, que vem a ser o número de bits que diferem na mesma posição em dois elementos consecutivos e de mesmo tamanho (Santos, 2000; Kak, 2010). Com relação a esta medida, a representação do código binário é não uniforme, pois pontos distantes podem ter pequena distância *Hamming* e pontos próximos podem ter grandes distâncias *Hamming* (Kak, 2010). A Tabela 3-7 mostra a sequência da mudança de bits do sistema Binário em relação ao *Gray*, como podemos observar a Distância *Hamming* do *Gray* é menor do que a do Binário para determinados elementos, pois há apenas uma troca de bit nos elementos consecutivos em toda sua sequência, o que não ocorre no Binário, como exemplo, para a mudança do elemento 3 para 4 há troca de 3 bits. A codificação *Gray* garante a distância *Hamming* de um elemento para seu adjacente, porém esta não é bem definida em elementos não adjacentes (Kak, 2010).

Tabela 3-7 – Exemplo de Distância *Hamming* nas codificações Binário e *Gray* (Kak, 2010).

Numérico	1	2	3	4	5	6	7
Binário	0001	0010	0011	0100	0101	0110	0111
Gray	0001	0011	0010	0110	0111	0101	0100

A conversão de um número binário de três bits para *Gray* ocorre da seguinte forma: o bit mais significativo é usado como *Gray* MSB (most significant bit), depois o binário MSB é comparado com o bit binário (B1) mais próximo e, se forem iguais, $G1=0$, caso contrário, $G1=1$. O último bit, $G0$, é encontrado pela comparação entre B1 e B0. Esta conversão pode ser vista na Figura 3-1.

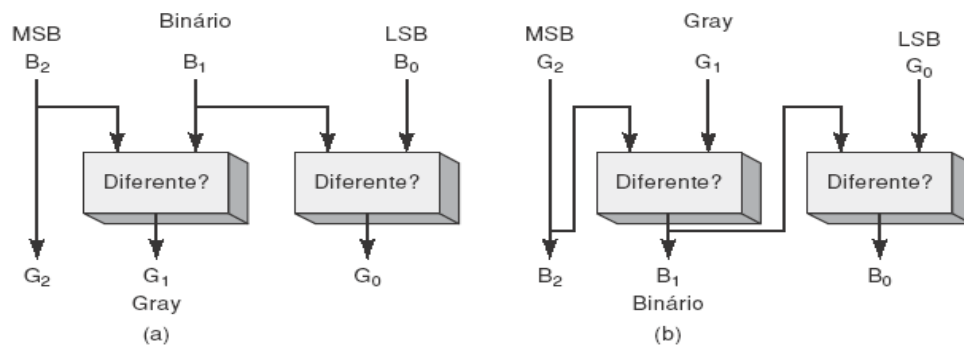


Figura 3-1 – (a) Codificação Binário-Gray; (b) Gray-Binário (Tocci et al., 2007).

O código *Gray* pode ser convertido em binário primeiramente atentando para o fato de que o MSB de ambos é o mesmo sempre; o próximo bit é obtido pela comparação do bit binário da esquerda com seu correspondente em *Gray* – bits iguais produzem zero e diferentes, um. Esta codificação é mais utilizada em codificadores de rotação de eixo (Tocci et al., 2007).

3.4.5 Codificação Numérica

A codificação numérica utiliza simplesmente o sistema decimal de numeração, composto por dez numerais ou símbolos que tem o valor definido por sua posição no número, por isso é dito um sistema posicional (Tocci et al., 2007).

É importante ressaltar que não se deve usar esse tipo de sistema em variáveis categóricas sem ordem. Um exemplo é o caso de termos três categorias de cores – vermelho, verde e azul – simbolizadas, respectivamente, pelos valores 1, 2 e 3 e treinarmos uma rede para encontrarmos uma categoria desconhecida X entre a 1 e a 3, como mostra a Figura 3-2. A saída correta seria termos 50% de chance de o resultado ser 1 ou 3, porém, utilizando o sistema numérico na variável de saída teremos como resposta a média das duas variáveis, neste caso 2, um resultado que não seria correto. Para variáveis categóricas o indicado é a conversão das mesmas para codificações compostas apenas pelos elementos 0 e 1 – *Dummy*, Termômetro, *Gray*, Binário, etc. (Sarle, 2002).

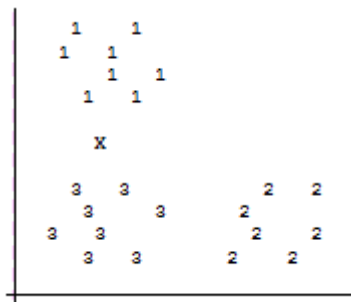


Figura 3-2 – Variáveis categóricas.

3.4.6 Análise de Correspondência

A AC é uma técnica de análise multivariada que converte uma matriz de dados não negativos em uma representação gráfica em que linhas e colunas são representadas em dimensões reduzidas, sendo muito utilizada, com eficiência, em matrizes com grande quantidade de dados (Czermainski, 2004; Greenacre & Hastie, 1987). Tem como objetivo o condensamento dos dados, sendo classificada como técnica de interdependência por analisar todas as variáveis ao mesmo tempo. Seu nome tem origem no estudo da correspondência entre variáveis (Alves *et al.*, 2007; Freitas & Janissek-Muniz, 2008). As primeiras publicações sobre Análise de Correspondência (AC) são datadas de 1933, mas foi em 1960 que Benzécri formalizou o estudo para tabelas de contingência (Mingoti, 2005). A Tabela 3-8 é um exemplo de tabela de contingência genérica.

Quando os dados estão dispostos na forma de *Tabelas de Contingência* (TC), as quais são tabelas de frequências absolutas, a técnica recebe o nome de *Análise de Correspondência Simples* (ACS) (Pamplona, 1998). Um exemplo de tabela de contingência genérica pode ser visto na Tabela 3-8.

Tabela 3-8 – Tabela de contingência genérica (Mingoti, 2005).

		Variável Y				Total
		1	2	.	q	
Variável X	1	n_{11}	n_{12}	.	n_{1q}	$n_{1.}$
	2	n_{21}	n_{22}	.	n_{2q}	$n_{2.}$

	p	n_{p1}	n_{p2}	.	n_{pq}	$n_{p.}$
Total		$n_{.1}$	$n_{.2}$.	$n_{.q}$	$n_{..} = n$

Onde:

- n_{ij} – número de elementos pertencentes à categoria i da variável X e à categoria j da variável Y ;
- $n_{i.}$ – frequência total observada na i -ésima categoria de X ;
- $n_{.j}$ – frequência total observada na j -ésima categoria de Y ;
- n – total geral de frequências observadas.

A tabela de contingência pode ser representada por uma matriz de dados na qual será aplicada a teoria matemática, como mostra a Equação (3-2) (Pamplona, 1998).

$$N = \begin{bmatrix} n_{11} & \cdots & n_{1j} \\ \vdots & \ddots & \vdots \\ n_{i1} & \cdots & n_{ij} \end{bmatrix} = [n_{ij}]_{I \times J}, \quad n_{ij} \geq 0 \quad (3-2)$$

Partindo-se da matriz de dados, \mathbf{N} , é construída a matriz de frequências absolutas, \mathbf{P} , também chamada de *Matriz de Correspondência* (MC), Tabela 3-9, na qual cada linha ou coluna pode ser vista como um vetor de proporções, Equação (3-3) (Pamplona, 1998; Czermainski, 2004).

$$P = \left(\frac{1}{n}\right) \cdot N = [p_{ij}]_{I \times J}, \quad 0 \leq p_{ij} \leq 1 \quad (3-3)$$

Tabela 3-9 – Matriz de Correspondência (Czermainski, 2004).

		Variável Y				Total
		1	2	.	j	
Variável X	1	p_{11}	p_{12}	.	p_{1j}	$p_{1.}$
	2	p_{21}	p_{22}	.	p_{2j}	$p_{2.}$

	i	p_{i1}	p_{i2}	.	p_{ij}	$p_{i.}$
Total		$p_{.1}$	$p_{.2}$.	$p_{.j}$	1

Onde:

- $p_{ij} = \frac{n_{ij}}{n}$;
- $p_{i.} = \frac{n_{i.}}{n}$;
- $p_{.j} = \frac{n_{.j}}{n}$.

As massas, vetores de frequências relativas marginais, são definidas através da MC ou através dos vetores de massa da TC dividido pelo total geral (n), sua representação para as linhas (\mathbf{r}_i) e para as colunas (\mathbf{c}_j) é apresentada, respectivamente, pelas Equações (3-4) e (3-5). Com isso, têm-se os vetores de massa das linhas e das colunas (ou *centróide dos perfis de coluna e de linha, respectivamente*) da matriz \mathbf{P} , Equações (3-6) e (3-7), que representam as frequências relativas das mesmas (Pamplona, 1998; Czermainski, 2004).

$$r_i = \frac{n_{i.}}{n} = \sum_{j=1}^J p_{ij}, i = 1, \dots, I; \quad (3-4)$$

$$c_j = \frac{n_{.j}}{n} = \sum_{i=1}^I p_{ij}, j = 1, \dots, J; \quad (3-5)$$

$$r = (r_1, \dots, r_I)'; \quad (3-6)$$

$$c = (c_1, \dots, c_J)'. \quad (3-7)$$

As variáveis utilizadas para a análise devem ser qualitativas ou quantitativas após serem tratadas por processo de categorização (Freitas & Janissek-Muniz, 2008). Uma questão a ser abordada é o tratamento estatístico de variáveis qualitativas antes de ser aplicada a técnica de AC o que pode nos revelar baixa variabilidade de variáveis em detrimento de outras mais importantes para o estudo em questão (Guedes *et al.*, 1999).

Para a conversão dos dados a matriz deve ser retangular com entradas não negativas, pois o método se baseia na decomposição do valor singular desta matriz e representa suas

linhas e colunas como pontos no espaço. A representação gráfica nos permite avaliar estes de acordo com sua posição no espaço vetorial, temos então o chamado “Mapa de Correspondência” (Czermainski, 2004; Souza *et al.*, 2010; Alves *et al.*, 2007). Um exemplo de mapa de correspondência pode ser visto na Figura 3-3.

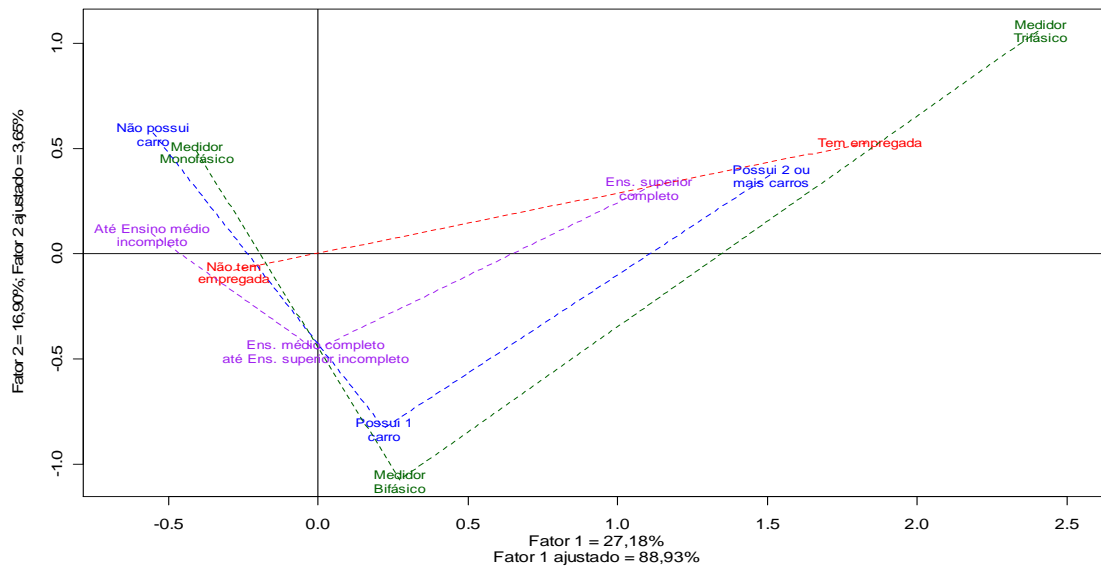


Figura 3-3 - Análise de Correspondência Múltipla de características dos domicílios e de consumo de energia, em solução bi-dimensional.

A distância entre dois pontos, ou duas categorias, é a “Distância Qui-Quadrado”. Se as categorias estiverem dispostas em colunas, calculamos a distância, Equação (3-8), através do perfil de coluna, Equação (3-9), considerando o “peso” de cada linha. O cálculo é feito da mesma forma para as linhas, Equações (3-10) e (3-11). Temos então a distância de todas as categorias em relação às outras formando uma “nuvem de pontos” (Carvalho & Struchiner, 1992). A média dos perfis nos resulta o centro de gravidade da “nuvem” conhecido como centróide, Equação (3-12), que é a representação das frequências marginais relativas (Mendes, 2009; Pamplona, 1998). De acordo com Mendes 2009, “a média das distâncias qui-quadrado de todos os objetos em relação ao centróide é uma medida de inércia da distribuição dos pontos e terá valor zero quando todos os pontos recaírem sobre a média (centróide)”.

Distância Qui-quadrado entre perfis de coluna:

$$d_r(b_j, b_j') = (b_j - b_j')' \cdot D_r^{-1} \cdot (b_j - b_j') \quad (3-8)$$

Perfil de coluna:

$$b_j = \left[\frac{n_{1j}}{n_{.j}}, \frac{n_{2j}}{n_{.j}} \dots \frac{n_{Ij}}{n_{.j}} \right]', \quad j = 1, \dots, J \quad (3-9)$$

Distância Qui-quadrado entre perfis de linha:

$$d_c(a_i, a_i') = (a_i - a_i')' \cdot D_c^{-1} \cdot (a_i - a_i') \quad (3-10)$$

Perfil de linha:

$$a_i = \left[\frac{n_{i1}}{n_{i.}}, \frac{n_{i2}}{n_{i.}} \dots \frac{n_{iJ}}{n_{i.}} \right]', \quad i = 1, \dots, I \quad (3-11)$$

Centróide dos perfis de linha:

$$\bar{R} = \frac{1}{\bar{r}} \cdot \sum_{i=1}^I (r_i \cdot R_i) \quad (3-12)$$

Massa do centróide dos perfis de linha:

$$\bar{r} = \sum_{i=1}^I r_i \quad (3-13)$$

Onde:

- D_r é a matriz diagonal de elementos $r_i = \frac{n_{i.}}{n}$, $i = 1, \dots, I$;
- D_c é a matriz diagonal de elementos $c_j = \frac{n_{.j}}{n}$, $j = 1, \dots, J$.

Os auto-vetores da matriz de dados nos retorna a solução do problema. Cada eixo corresponde a um auto-vetor e define o espaço de multi-dimensões. O eixo também corresponde a um percentual da variância da matriz; quanto maior seu valor, melhor é sua representatividade na “nuvem de pontos” e a relação entre variáveis (Carvalho & Struchiner, 1992).

O Mapa de Correspondência é resultado do que inicialmente é a “nuvem de pontos” e que posteriormente é projetada num plano capaz de representar as distâncias originais entre os pontos e a relação entre categorias. Os pontos próximos entre si possuem perfil semelhante; os mais distantes, perfil discrepante; e os pontos próximos ao centro do gráfico

estão associados a várias categorias (Gonçalves & Santos, 2009; Carvalho & Struchiner, 1992; Cunha Jr., 1997).

A AC é considerada um caso especial da Análise de Componentes Principais (ACP) voltada para dados categóricos organizados em tabelas de contingência de dupla entrada. O problema é análogo, porém com modificações devido à ponderação das observações e à métrica ponderada (Czermainski, 2004; Alves *et al.*, 2007).

A técnica tem vasta aplicação na área de ciências humanas, devido à grande utilização de variáveis categóricas. Um exemplo de seu uso é na área de marketing, em que pesquisadores buscam identificar organizações concorrentes de acordo com seu perfil de consumidor (Cunha Jr., 1997).

4 Resultados

4.1 Simulações feitas para definição das arquiteturas das redes

4.1.1 Banco de Dados Córneas

Nesta seção podem ser vistos os resultados das simulações feitas para as diferentes codificações utilizadas no banco de dados Córneas. Estes resultados são apresentados por meio de gráficos e tabelas; a métrica usada para avaliar o desempenho foi o número de classificações corretas obtidas em cada simulação na amostra de treino (que contém 158 casos).

As formas de codificação apresentadas abaixo foram as descritas na seção 3.4; foram usadas três variações de emprego da codificação *Dummy*: MLP de uma camada oculta com *dummies 1-de-c* e *1-de-(c-1)*, e MLP com duas camadas ocultas com *dummies 1-de-c*. O número de neurônios para cada rede foi escolhido por meio da melhor mediana alcançada entre várias simulações na amostra de treino (este número está destacado nas tabelas).

4.1.1.1 Codificação *Dummy* 1-de-c: uma camada oculta

O gráfico da Figura 4-1 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando como codificação dos dados de entrada *Dummy* 1-de-c. A Tabela 4-1 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

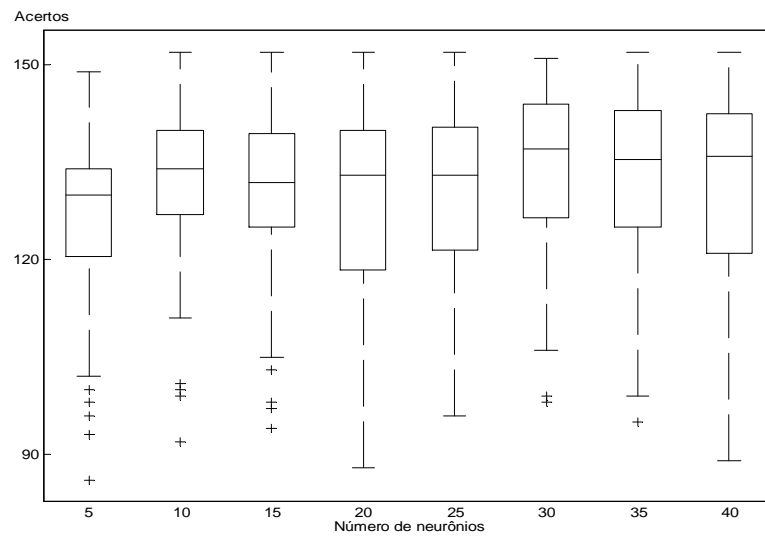


Figura 4-1 - Números de classificações corretas, *dummy* 1-de-c, uma camada oculta.

Tabela 4-1 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	125,70	130,0
10	131,82	134,0
15	130,71	132,0
20	129,18	133,0
25	129,62	133,0
30	134,09	137,0
35	132,14	135,5
40	131,46	136,0

4.1.1.2 Codificação *Dummy* 1-de-c: duas camadas ocultas

O gráfico da Figura 4-2 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de duas camadas ocultas, com diferentes números de neurônios, usando como codificação dos dados de entrada *Dummy* 1-de-c com duas camadas ocultas. A Tabela 4-2 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

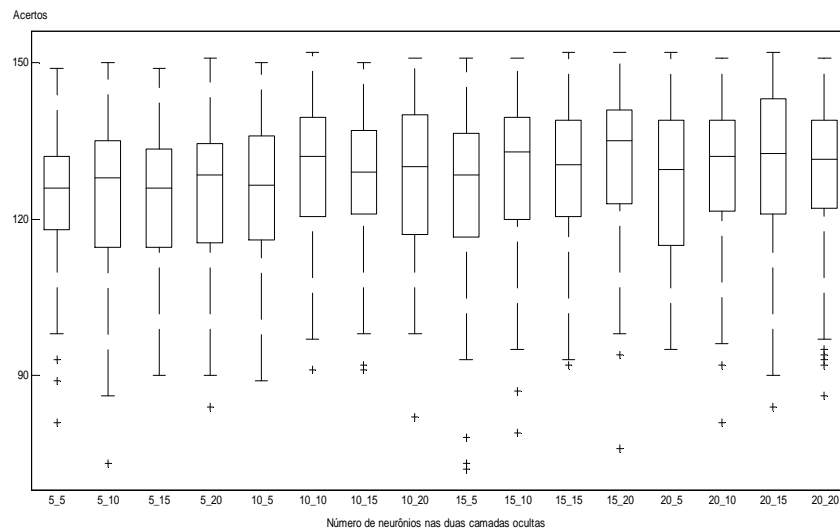


Figura 4-2 – Número de classificações corretas, *dummy* 1-de-c, duas camadas ocultas.

Tabela 4-2 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Média	Mediana
(5,5)	123,57	126,0
(5,10)	123,39	128,0
(5,15)	123,11	126,0
(5,20)	124,73	128,5
(10,5)	125,19	126,5
(10,10)	128,73	132,0
(10,15)	127,54	129,0
(10,20)	127,71	130,0
(15,5)	124,57	128,5
(15,10)	128,39	133,0
(15,15)	128,54	130,5
(15,20)	130,67	135,0
(20,5)	126,86	129,5
(20,10)	128,94	132,0
(20,15)	130,23	132,5
(20,20)	128,32	131,5

4.1.1.3 Codificação *Dummy* 1-de-(c-1)

O gráfico da Figura 4-3 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação dos dados de entrada por *Dummy* 1-de-(c-1). A Tabela 4-3 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

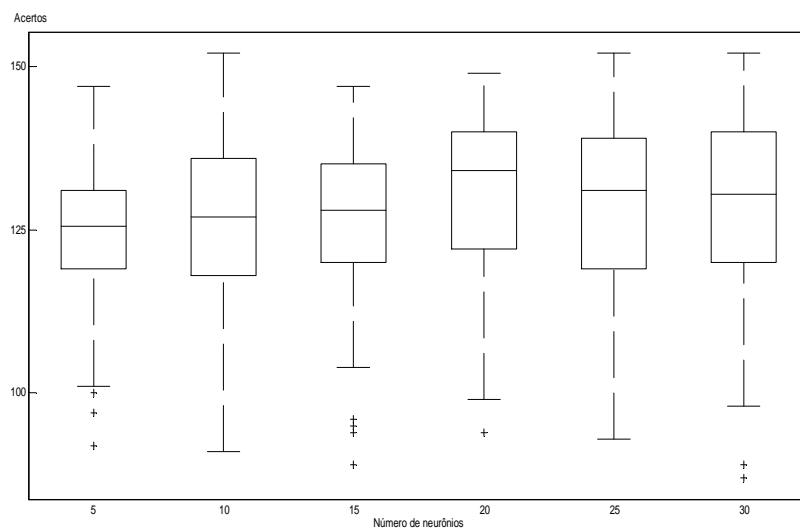


Figura 4-3 – Número de classificações corretas, *Dummy* 1-de-(c-1), uma camada.

Tabela 4-3 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	123,45	125,5
10	126,53	127,0
15	125,94	128,0
20	131,10	134,0
25	128,44	131,0
30	129,06	130,5

4.1.1.4 Codificação Termômetro

O gráfico da Figura 4-4 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Termômetro* dos dados de entrada. A Tabela 4-4 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

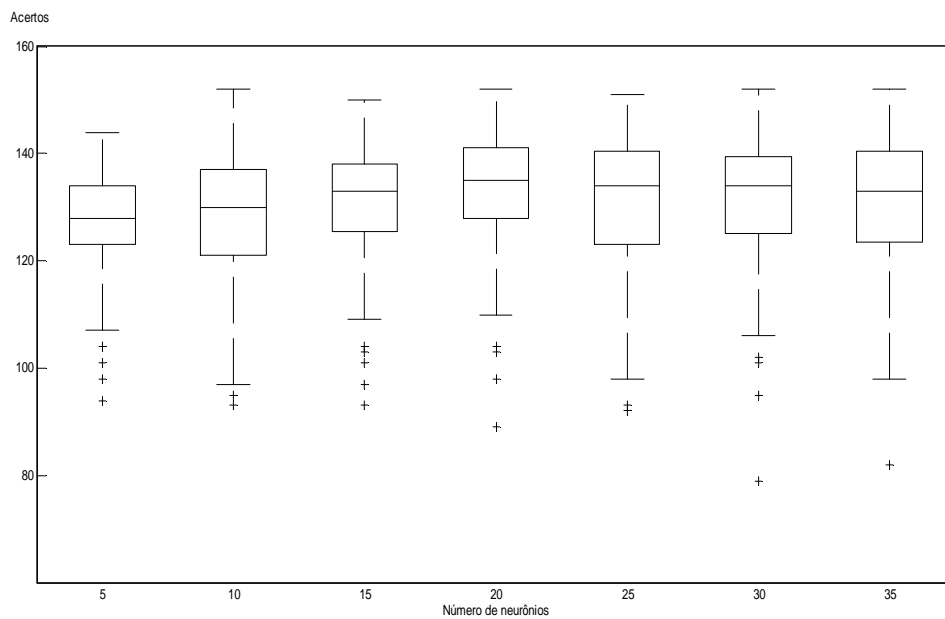


Figura 4-4 – Número de classificações corretas, Termômetro, uma camada oculta.

Tabela 4-4 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	127,17	128,0
10	127,26	130,0
15	130,24	133,0
20	132,89	135,0
25	130,52	134,0
30	131,00	134,0
35	130,56	133,0

4.1.1.5 Codificação Binária

O gráfico da Figura 4-5 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Binária* dos dados de entrada. A Tabela 4-5 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

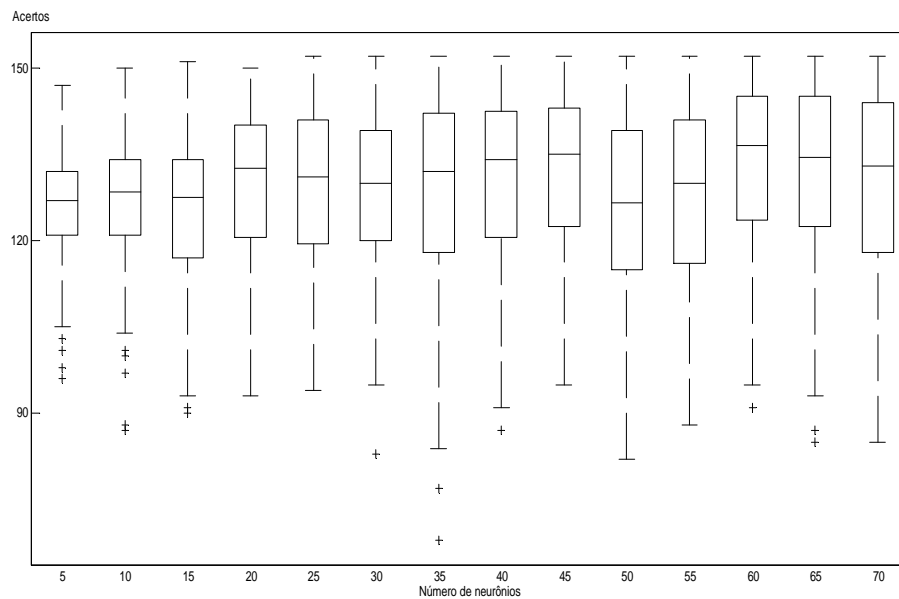


Figura 4-5 – Número de classificações corretas, codificação Binária, uma camada oculta.

Tabela 4-5 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	125,38	127,0
10	126,00	128,5
15	125,59	127,5
20	128,93	132,5
25	129,31	131,0
30	128,02	130,0
35	128,61	132,0
40	130,25	134,0
45	131,52	135,0
50	125,96	126,5
55	128,05	130,0
60	133,01	136,5
65	131,74	134,5
70	129,22	133,0

4.1.1.6 Codificação *Gray*

O gráfico da Figura 4-6 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Gray* dos dados de entrada. A Tabela 4-6 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

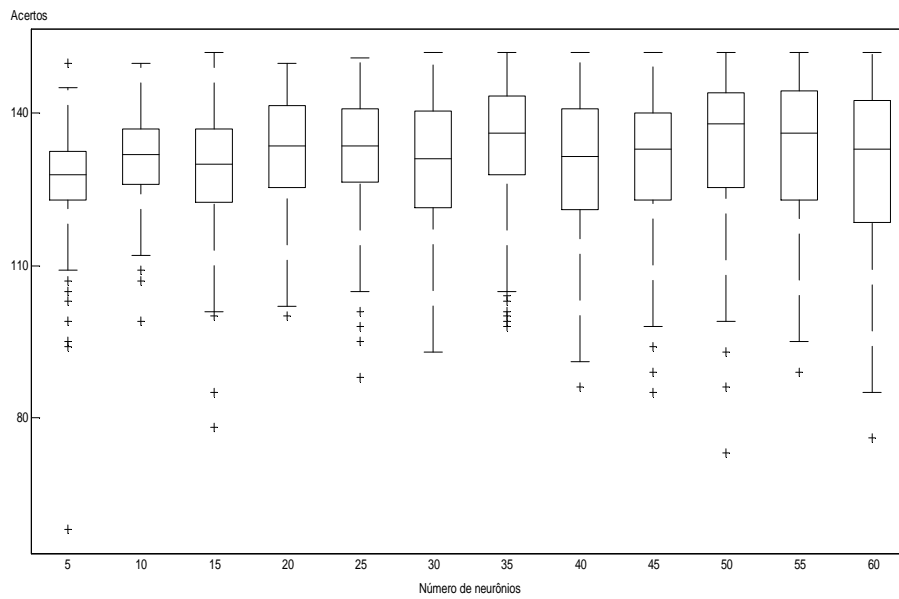


Figura 4-6 – Número de classificações corretas, codificação *Gray*, uma camada oculta.

Tabela 4-6 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	126,01	128,0
10	130,82	132,0
15	127,64	130,0
20	132,49	133,5
25	131,51	133,5
30	129,86	131,0
35	133,38	136,0
40	129,57	131,5
45	130,00	133,0
50	133,10	138,0
55	131,81	136,0
60	129,23	133,0

4.1.1.7 Codificação Numérica

O gráfico da Figura 4-7 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Numérica* dos dados de entrada. A Tabela 4-7 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

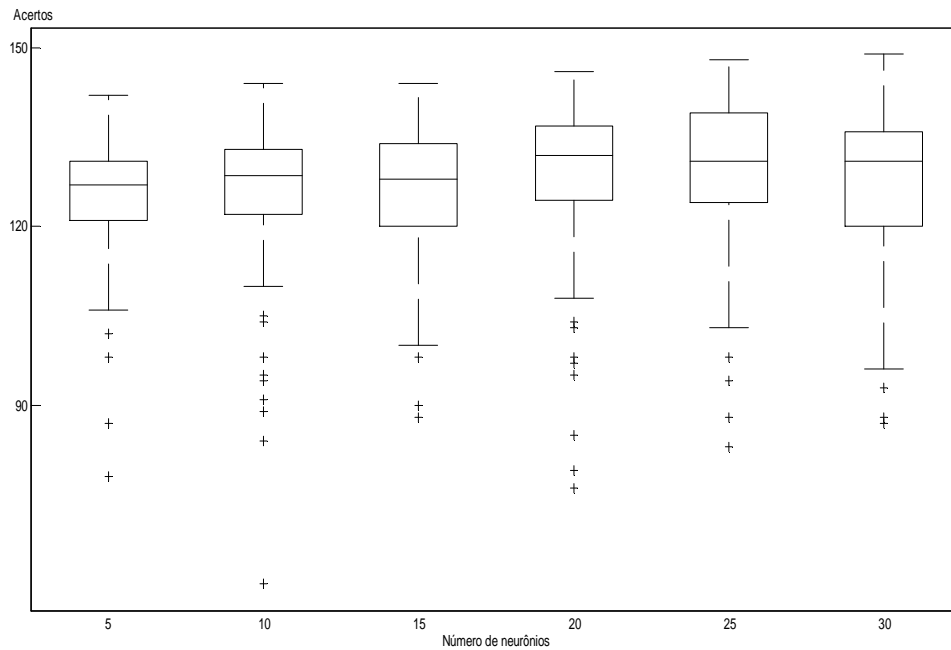


Figura 4-7 – Número de classificações corretas, codificação Numérica, uma camada oculta.

Tabela 4-7 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	124,62	127,0
10	124,26	128,5
15	125,75	128,0
20	127,88	132,0
25	129,62	131,0
30	127,35	131,0

4.1.1.8 Codificação AC

O gráfico da Figura 4-8 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação dos dados de entrada por meio dos fatores obtidos por *Análise de Correspondência*. A Tabela 4-8 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

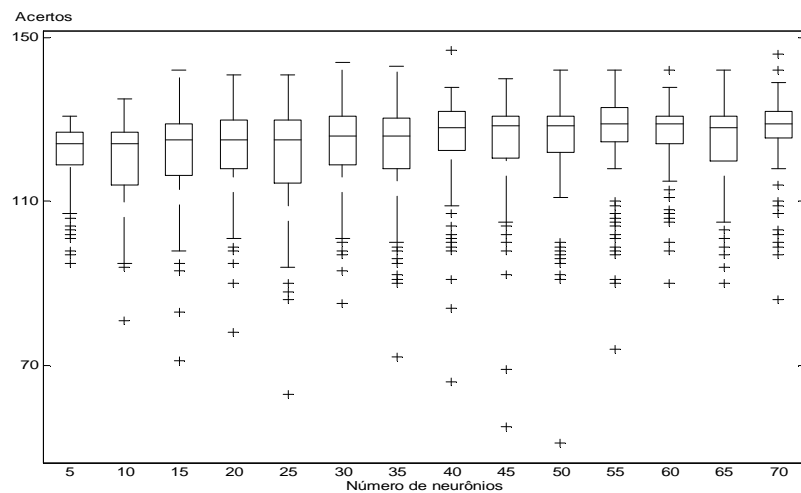


Figura 4-8 – Número de classificações corretas, codificação AC, uma camada oculta.

Tabela 4-8 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	120,49	124,0
10	120,51	124,0
15	121,05	125,0
20	121,65	125,0
25	120,24	125,0
30	123,16	126,0
35	122,32	126,0
40	124,27	128,0
45	123,61	128,5
50	123,82	128,5
55	124,16	129,0
60	125,81	129,0
65	123,79	128,0
70	126,14	129,0

4.1.1.9 Conclusão

A partir da análise das figuras e tabelas apresentadas e aplicando como critério de seleção dos neurônios o melhor desempenho na classificação, chegamos à conclusão que a melhor arquitetura para cada codificação, a ser empregada posteriormente na fase de teste, é a apresentada na Tabela 4-9.

Tabela 4-9 - Número de neurônios escolhidos para cada codificação.

Codificação	Neurônios
<i>Dummy</i>	30
<i>Dummy 1 - de - (c -1)</i>	20
<i>Dummy Dupla camada</i>	15-20
Binário	60
<i>Gray</i>	50
Numérico	20
Termômetro	20
AC	55

4.1.2 Banco de Dados Seguros

Nesta seção são vistos os resultados das simulações para as amostras de treino em diferentes codificações para o banco de dados *Seguros*. A forma de avaliação do desempenho foi a mesma utilizada para o banco de dados *Córneas*.

As codificações utilizadas para a análise deste banco de dados foram: *Dummy 1-de-c* com uma camada oculta; *Dummy 1-de-c* com duas camadas ocultas; *Dummy 1-de-(c-1)* com uma camada oculta; *Dummy + AC* com uma camada oculta; além das demais codificações apresentadas na seção 3.4.

4.1.2.1 Codificação *Dummy 1-de-c*: uma camada oculta

O gráfico da Figura 4-9 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação dos dados de entrada por *Dummy 1-de-c*. A Tabela 4-10 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

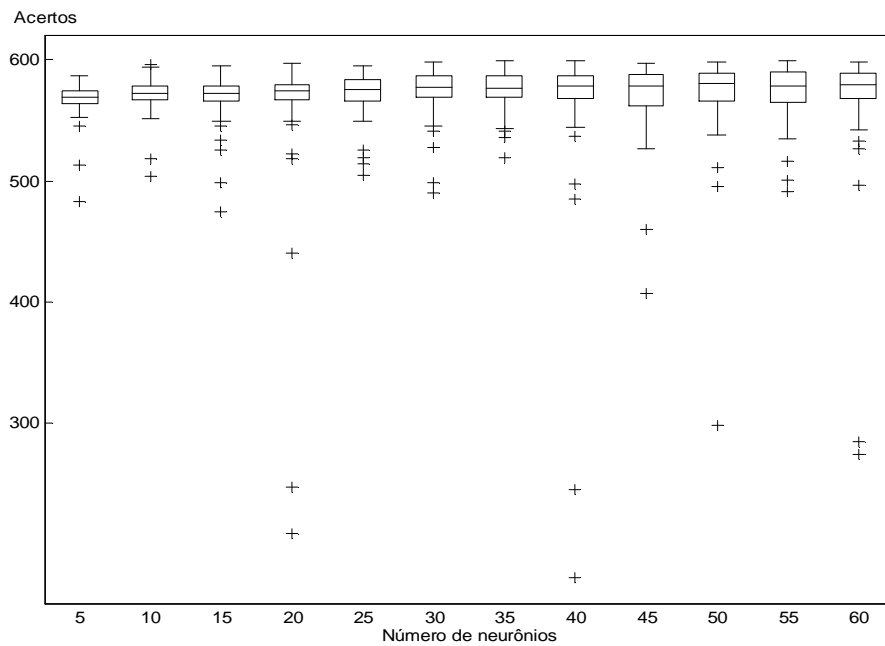


Figura 4-9 – Números de classificações corretas, *dummy 1-de-c*, uma camada oculta.

Tabela 4-10 – Número de classificações obtidas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	567,53	569,0
10	571,56	572,0
15	569,94	572,0
20	564,07	574,0
25	572,82	575,5
30	574,44	577,5
35	575,32	576,0
40	568,03	578,0
45	571,97	578,5
50	573,37	580,0
55	575,29	578,5
60	570,35	579,5

4.1.2.2 Codificação *Dummy 1-de-c*: duas camadas ocultas

O gráfico da Figura 4-10 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de duas camadas ocultas, com diferentes números de neurônios, usando codificação dos dados de entrada por *Dummy 1-de-c*. A Tabela 4-11 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

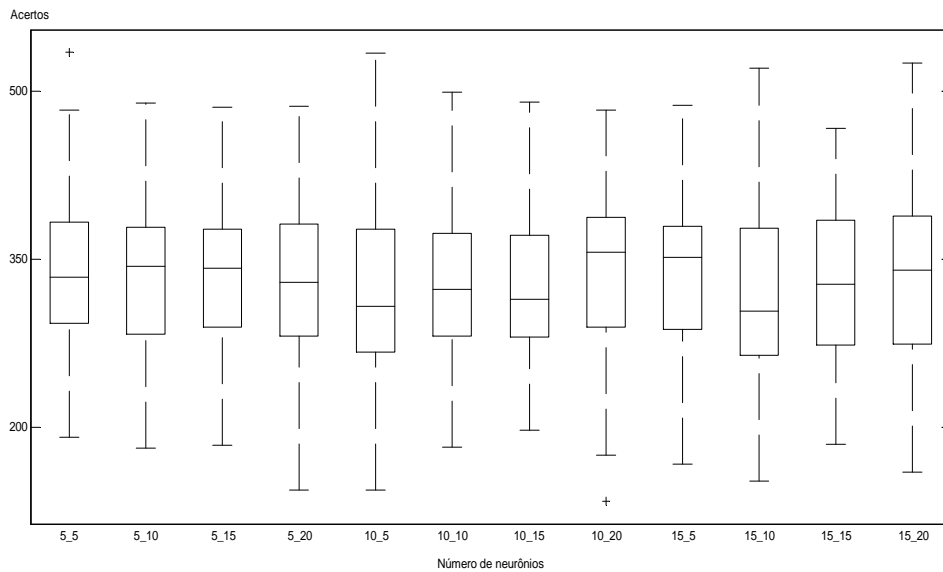


Figura 4-10 – Número de classificações corretas, *dummy 1-de-c*, duas camadas ocultas.

Tabela 4-11 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Média	Mediana
(5,5)	340,71	334,5
(5,10)	333,18	343,5
(5,15)	337,14	342,5
(5,20)	330,11	330,0
(10,5)	320,98	308,5
(10,10)	327,42	323,5
(10,15)	325,53	314,0
(10,20)	337,18	356,0
(15,5)	338,33	351,5
(15,10)	322,24	304,0
(15,15)	331,18	327,5
(15,20)	336,03	340,0

4.1.2.3 Codificação *Dummy 1-de-(c-1)*

O gráfico da Figura 4-11 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação dos dados de entrada por *Dummy 1-de-(c-1)*. A Tabela 4-12 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

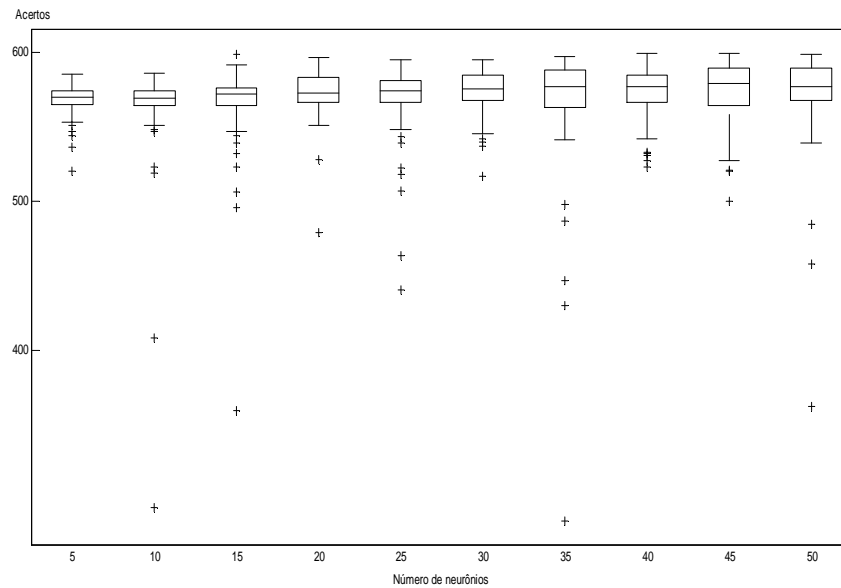


Figura 4-11 – Número de classificações corretas, *Dummy 1-de-(c-1)*, uma camada.

Tabela 4-12 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	567,77	570,0
10	563,64	569,0
15	566,69	571,5
20	572,27	572,5
25	569,91	574
30	573,92	575,5
35	568,34	577
40	573,23	577
45	574,58	579
50	572,48	577

4.1.2.4 Codificação Termômetro

O gráfico da Figura 4-12 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Termômetro* dos dados de entrada. A Tabela 4-13 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

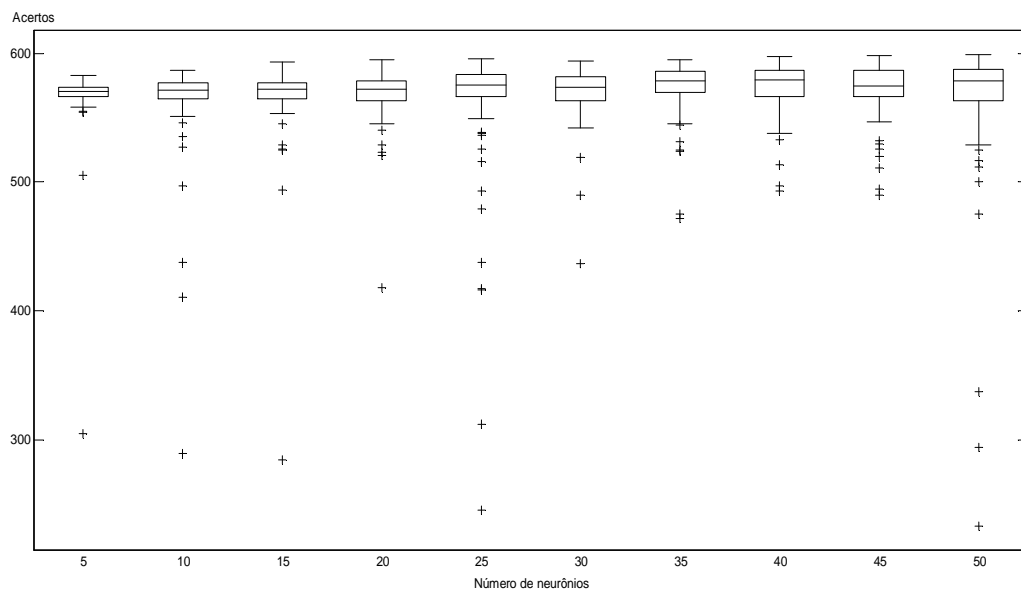


Figura 4-12 – Número de classificações corretas, Termômetro, uma camada oculta.

Tabela 4-13 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	566,94	570,5
10	563,54	571,0
15	566,47	572,0
20	569,01	572,0
25	561,79	575,0
30	570,34	574,0
35	574,23	578,5
40	573,82	579,0
45	571,82	574,5
50	564,29	578,5

4.1.2.5 Codificação Binária

O gráfico da Figura 4-13 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Binária* dos dados de entrada. A Tabela 4-14 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

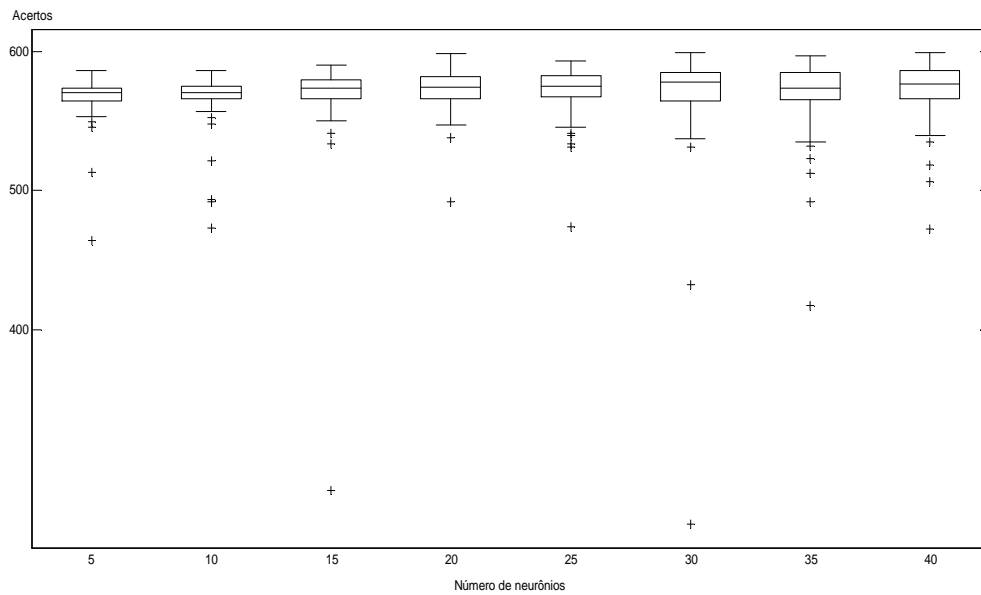


Figura 4-13 – Número de classificações corretas, codificação Binária, uma camada oculta.

Tabela 4-14 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	567,68	570,0
10	567,55	570,0
15	569,04	573,0
20	572,39	574,0
25	571,54	575,0
30	570,43	578,0
35	570,18	573,0
40	573,19	576,5

4.1.2.6 Codificação *Gray*

O gráfico da Figura 4-14 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Gray* dos dados de entrada. A Tabela 4-15 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

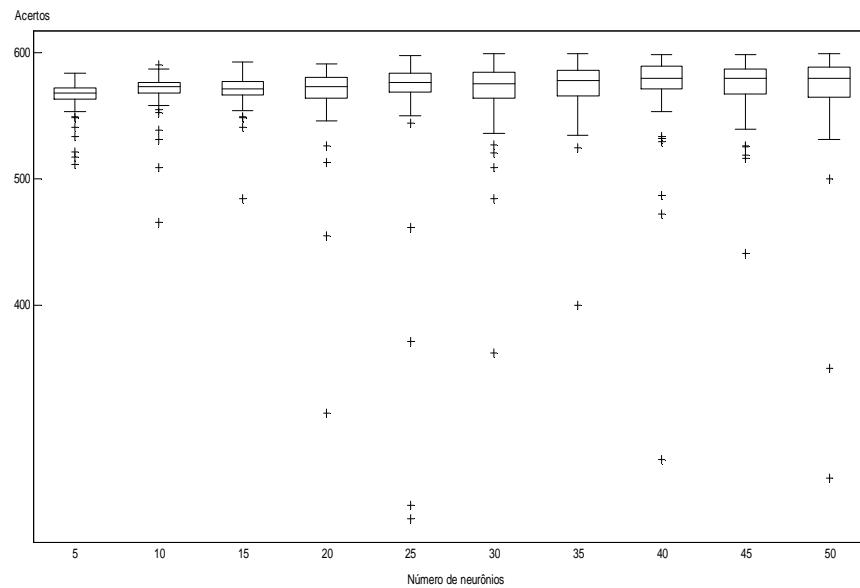


Figura 4-14 – Número de classificações corretas, codificação *Gray*, uma camada oculta.

Tabela 4-15 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	565,89	568,0
10	569,60	573,0
15	570,06	571,0
20	567,38	572,5
25	566,18	576,0
30	569,64	575,0
35	573,77	577,5
40	573,42	579,0
45	574,11	579,0
50	570,82	579,0

4.1.2.7 Codificação Numérica

O gráfico da Figura 4-15 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Numérica* dos dados de entrada. A Tabela 4-16 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

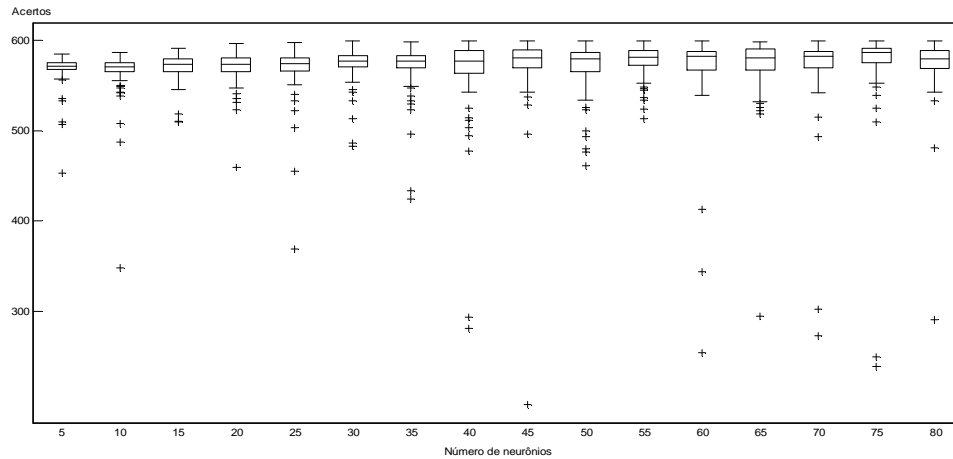


Figura 4-15 – Número de classificações corretas, codificação Numérica, uma camada oculta.

Tabela 4-16 – Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	568,01	571,0
10	565,16	570,0
15	570,18	573,0
20	569,55	573,0
25	569,28	574,0
30	573,71	576,5
35	571,52	577,0
40	566,13	577,0
45	573,04	580,0
50	570,80	579,0
55	577,20	581,0
60	570,19	582,0
65	573,31	580,0
70	571,73	582,0
75	574,61	586,0
80	573,68	579,0

4.1.2.8 Codificação AC: uma camada oculta

O gráfico da Figura 4-16 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação dos dados de entrada por meio dos fatores obtidos por *análise de correspondência*. A Tabela 4-17 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

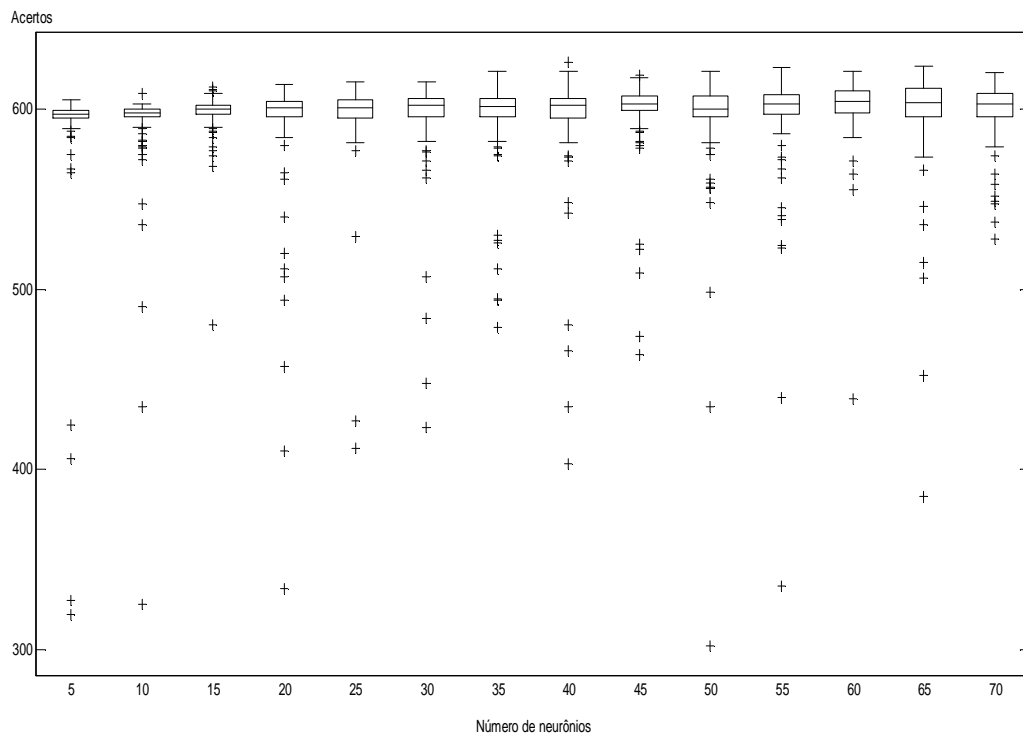


Figura 4-16 – Número de classificações corretas, codificação AC, uma camada oculta.

Tabela 4-17 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Mediana
5	587,26	597,0
10	590,01	598,0
15	597,61	600,0
20	589,82	601,0
25	594,91	600,5
30	594,37	602,0
35	595,48	601,5
40	593,58	602,0
45	597,43	603,0
50	593,07	600,0
55	594,56	603,0
60	601,15	604,0
65	596,03	603, 5
70	598,98	603,0

4.1.2.9 Codificação AC: duas camadas ocultas

Os gráficos da Figura 4-17 e da Figura 4-18 mostram os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de duas camadas ocultas, com diferentes números de neurônios, usando codificação dos dados de entrada por meio dos fatores obtidos por *análise de correspondência*. A Tabela 4-18 mostra as médias e medianas do número de acertos de cada modelo, na amostra de treino.

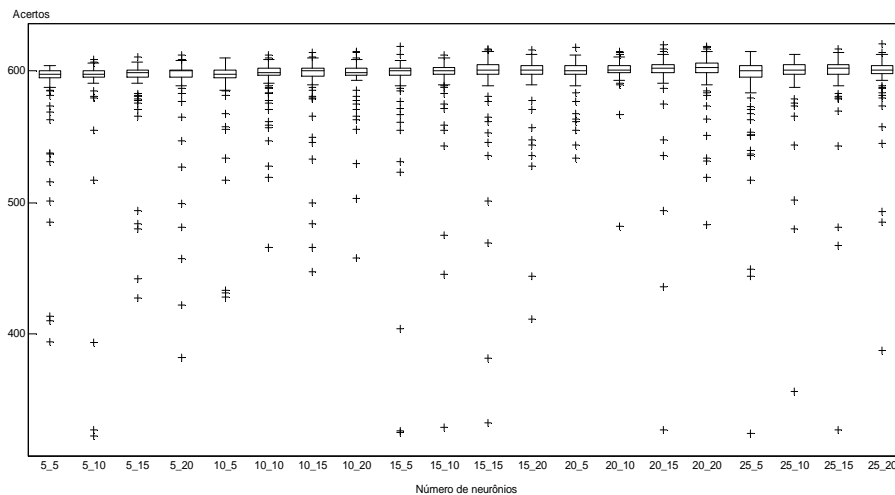


Figura 4-17 – Número de classificações corretas, codificação AC, duas camadas ocultas.

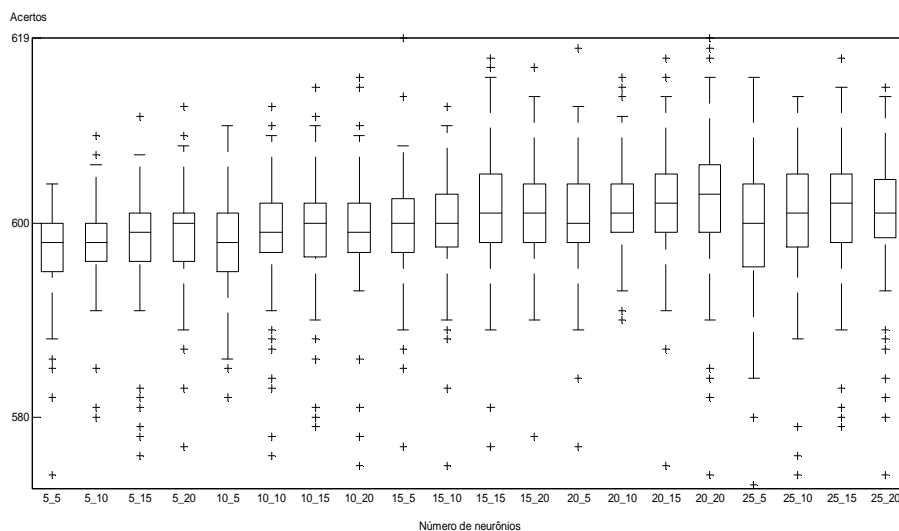


Figura 4-18 - Número de classificações corretas, codificação AC, duas camadas ocultas (Zoom).

Tabela 4-18 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Média	Mediana
(5,5)	586,01	598,0
(5,10)	589,39	598,0
(5,15)	590,71	599,0
(5,20)	590,27	600,0
(10,5)	590,62	598,0
(10,10)	594,44	599,0
(10,15)	592,30	600,0
(10,20)	595,12	599,0
(15,5)	589,75	600,0
(15,10)	592,81	600,0
(15,15)	592,17	601,0
(15,20)	595,22	601,0
(20,5)	597,96	600,0
(20,10)	599,92	601,0
(20,15)	594,82	602,0
(20,20)	598,41	603,0
(25,5)	589,69	600,0
(25,10)	595,18	601,0
(25,15)	595,32	602,0
(25,20)	595,62	601,0

4.1.2.10 Conclusão

Avaliando as simulações realizadas para este banco de dados, a melhor arquitetura para as codificações na fase de teste é a apresentada na Tabela 4-19.

Tabela 4-19 – Número de neurônios escolhidos para cada codificação.

Codificação	Neurônios
<i>Dummy</i>	50
<i>Dummy</i> 1 - de - (c - 1)	45
<i>Dummy</i> Dupla Camada	10-20
Binário	30
<i>Gray</i>	40
Numérico	55
Termômetro	40
AC Dupla Camada	20-20
AC	60

4.1.3 Banco de Dados Consumo

Nesta seção são vistos os resultados das simulações do banco de dados *Consumo*; a métrica utilizada para avaliar o desempenho foram os erros *médio quadrático* (MSE) e *percentual absoluto médio* (MAPE). Os gráficos e as tabelas apresentados nesta seção são referentes somente ao MAPE, para título de ilustração da técnica.

As codificações utilizadas neste banco de dados foram apresentadas na seção 3.4.

4.1.3.1 Codificação *Dummy 1-de-c*: uma camada oculta

O gráfico da Figura 4-19 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação dos dados de entrada por *Dummy 1-de-c*. A Tabela 4-20 mostra as médias e o desvio padrão de cada modelo, na amostra de treino.

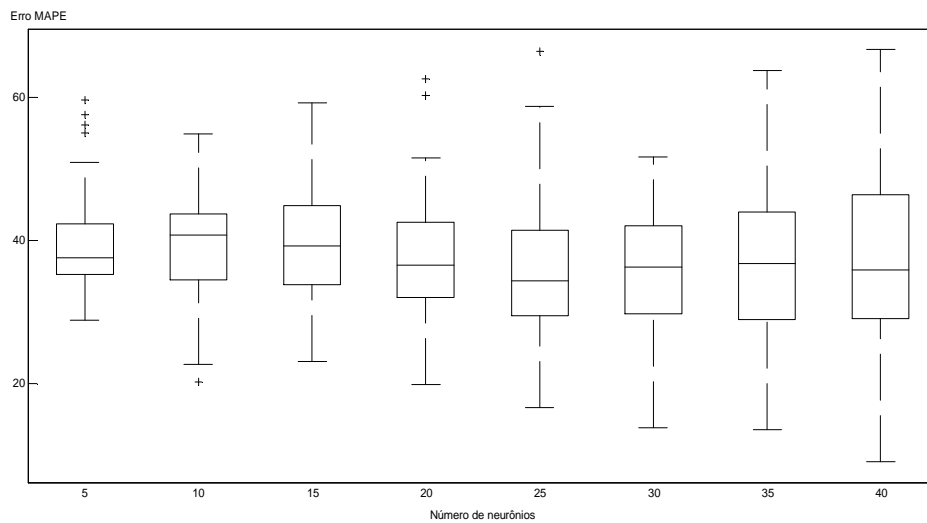


Figura 4-19 – Números de classificações corretas, *dummy 1-de-c*, uma camada oculta.

Tabela 4-20 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Desvio Padrão
5	39,15	6,04
10	39,58	6,71
15	39,61	7,51
20	37,77	7,81
25	35,80	8,81
30	35,77	8,48
35	36,76	10,54
40	37,19	12,92

4.1.3.2 Codificação Termômetro

O gráfico da Figura 4-20 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Termômetro* dos dados de entrada. A Tabela 4-21 mostra as médias e o desvio padrão de cada modelo, na amostra de treino.

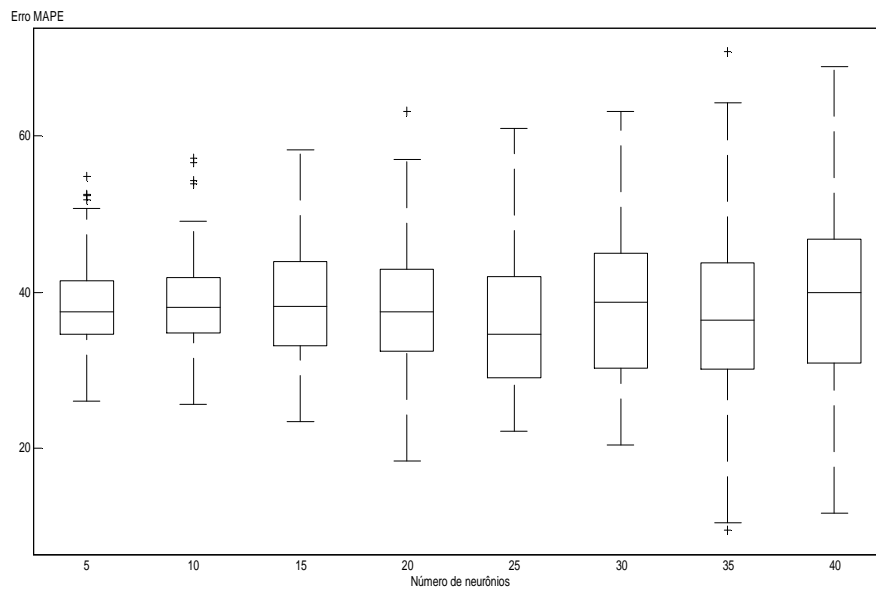


Figura 4-20 – Número de classificações corretas, Termômetro, uma camada oculta.

Tabela 4-21 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Desvio Padrão
5	38,51	5,79
10	38,55	6,05
15	38,72	6,82
20	37,83	6,12
25	36,06	8,28
30	38,14	9,11
35	37,17	11,72
40	38,55	12,49

4.1.3.3 Codificação Binária

O gráfico da Figura 4-21 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Binária* dos dados de entrada. A Tabela 4-22 mostra as médias e o desvio padrão de cada modelo, na amostra de treino.

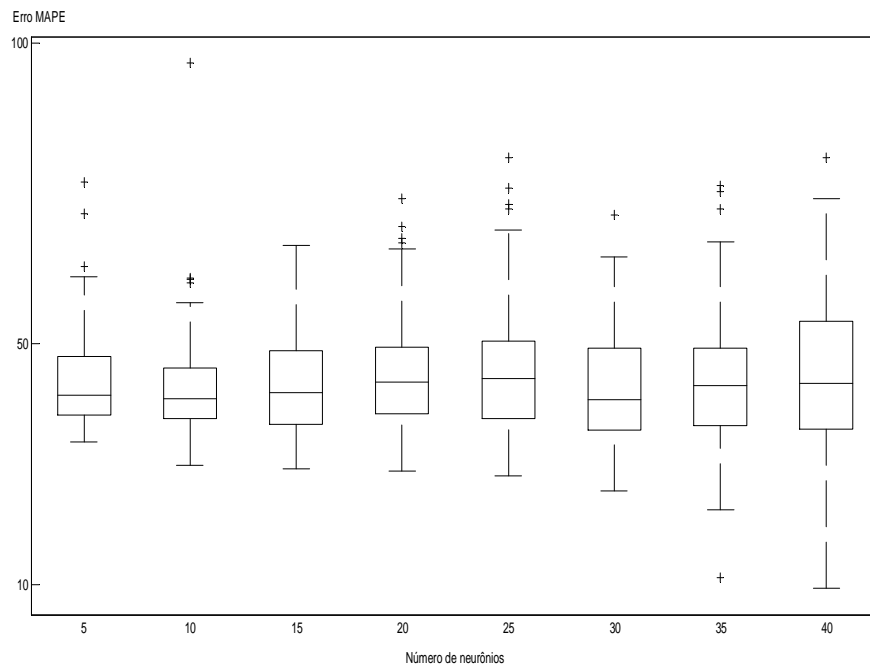


Figura 4-21 – Número de classificações corretas, codificação Binária, uma camada oculta.

Tabela 4-22 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Desvio Padrão
5	43,99	8,22
10	42,82	8,70
15	43,73	8,80
20	44,67	9,13
25	45,55	10,46
30	42,43	9,10
35	43,93	11,35
40	44,68	13,15

4.1.3.4 Codificação *Gray*

O gráfico da Figura 4-22 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Gray* dos dados de entrada. A Tabela 4-23 mostra as médias e o desvio padrão de cada modelo, na amostra de treino.

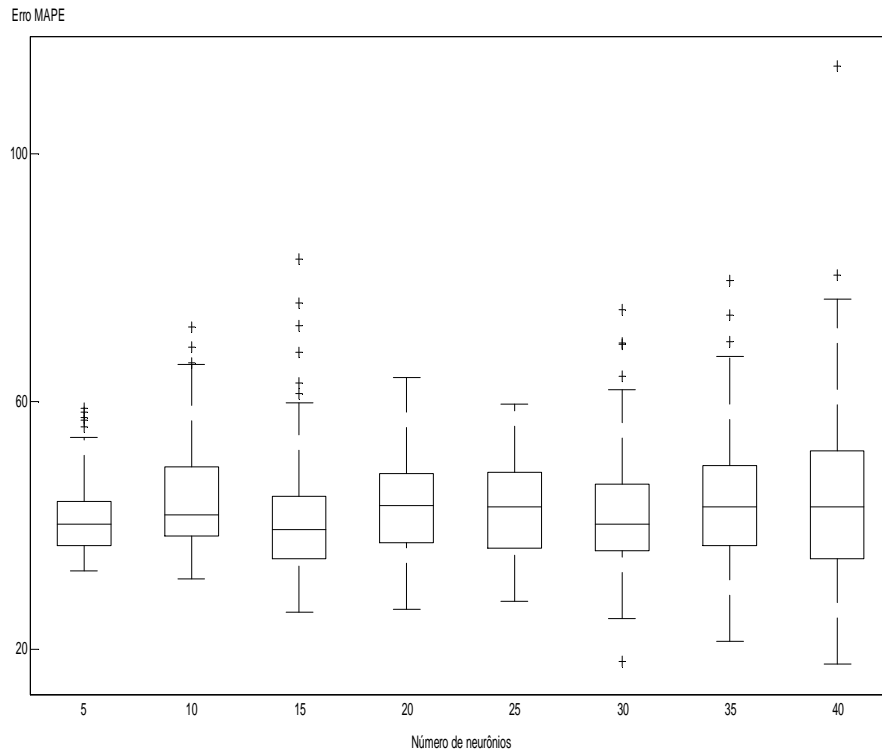


Figura 4-22 – Número de classificações corretas, codificação *Gray*, uma camada oculta.

Tabela 4-23 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Desvio Padrão
5	41,73	6,30
10	44,53	9,38
15	41,49	9,90
20	43,51	8,29
25	42,88	7,81
30	42,22	9,77
35	44,06	11,01
40	45,01	14,26

4.1.3.5 Codificação Numérica

O gráfico da Figura 4-23 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação *Numérica* dos dados de entrada. A Tabela 4-24 mostra as médias e o desvio padrão de cada modelo, na amostra de treino.

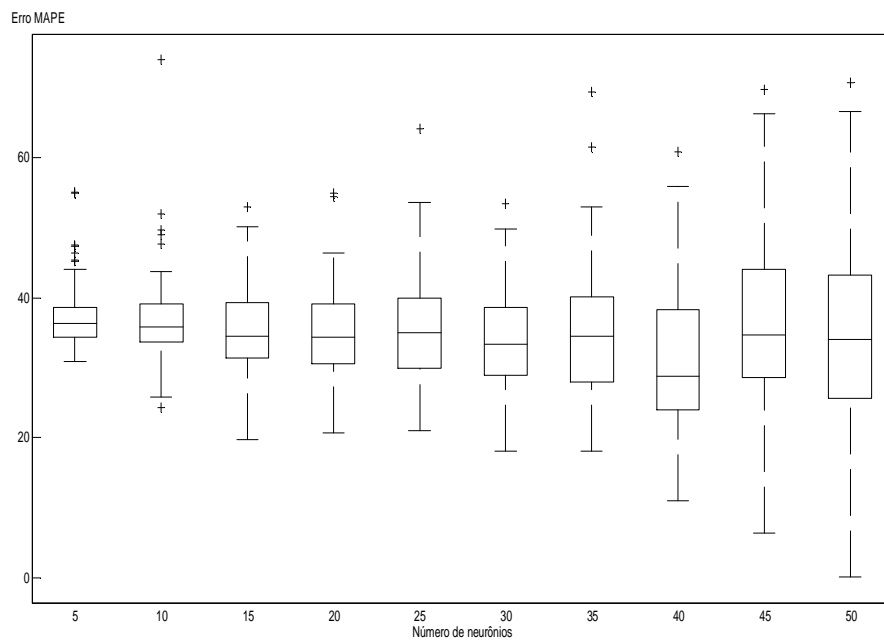


Figura 4-23 – Número de classificações corretas, codificação Numérica, uma camada oculta.

Tabela 4-24 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Desvio Padrão
5	37,29	4,49
10	36,60	6,25
15	35,54	5,99
20	34,91	6,40
25	35,77	7,51
30	33,62	7,17
35	34,69	9,00
40	31,63	10,90
45	36,16	12,09
50	34,47	13,49

4.1.3.6 Codificação AC

O gráfico da Figura 4-24 mostra os números de classificações corretas obtidas nos dados de treino em 100 simulações de redes MLP de uma camada oculta, com diferentes números de neurônios, usando codificação dos dados de entrada por meio dos fatores obtidos por *análise de correspondência*. A Tabela 4-25 mostra as médias e o desvio padrão de cada modelo, na amostra de treino.

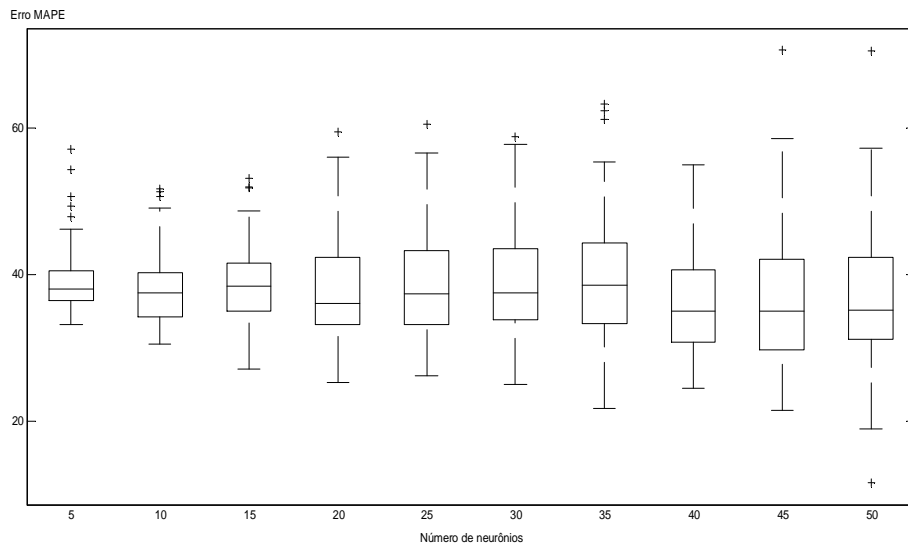


Figura 4-24 – Número de classificações corretas, codificação AC, uma camada oculta.

Tabela 4-25 - Classificações corretas em 100 simulações de cada rede.

Neurônios	Médias	Desvio Padrão
5	39,04	4,18
10	37,99	4,84
15	38,69	4,90
20	38,12	7,03
25	38,49	7,35
30	38,79	7,20
35	39,56	8,58
40	35,96	6,79
45	36,36	9,18
50	36,69	9,54

4.1.3.7 Conclusão

A partir da análise das simulações realizadas para os erros MAPE e MSE, concluímos que a melhor arquitetura para as codificações na fase de teste para cada um dos erros são as apresentadas nas Tabela 4-26 e Tabela 4-27.

Tabela 4-26 - Número de neurônios escolhidos para cada codificação – MAPE.

Codificação	Neurônios
<i>Dummy</i>	25
Binário	30
<i>Gray</i>	15
Numérico	40
Termômetro	25
AC	40

Tabela 4-27 - Número de neurônios escolhidos para cada codificação – MSE.

Codificação	Neurônios
<i>Dummy</i>	40
Binário	25
<i>Gray</i>	35
Numérico	35
Termômetro	35
AC	40

4.2 Comparação Final dos Modelos e Discussão

Para cada uma das bases de dados utilizadas neste trabalho, e para cada codificação, a melhor arquitetura da rede foi escolhida de acordo com o critério visto anteriormente (Seção 4.1). Nesta seção, comparamos por meio de tabelas e os gráficos os resultados obtidos por 100 simulações de cada um destas redes selecionadas para cada banco de dados e para cada forma de codificação, nas amostras de teste, e discutimos estes resultados. O objetivo é avaliar o desempenho da codificação por AC em relação às outras formas mais usuais de codificação.

Neste trabalho, experimentamos a técnica do *backward elimination*, para a seleção de variáveis, nos bancos de dados *Córneas* e *Seguros*. A técnica, no entanto se mostrou ineficiente e levou a resultados insatisfatórios. Parece que, nestes bancos de dados, todas as variáveis presentes eram essenciais, e não podiam ser descartadas. Isto já era em parte esperado, uma vez que estes bancos de dados já foram usados antes por outros pesquisadores, e já foram, portanto bastante depurados. Resolvemos, assim, não incluir aqui as simulações obtidas a partir desta técnica. Os resultados mostrados abaixo foram, portanto, conseguidos usando-se todas as variáveis de entrada disponíveis nos bancos de dados.

4.2.1 Banco de Dados *Córneas*

A amostra de teste deste banco de dados continha 57 casos. A Figura 4-25 e a Tabela 4-28 apresentadas abaixo comparam os desempenhos das diferentes codificações dos dados, em termos dos números de classificações corretas obtidos em 100 simulações de cada modelo. Analisando a Figura 4-25, observamos que o melhor resultado mediano foi atingido com a codificação *Termômetro*, que apesar de apresentar mesmo valor de mediana (42,5 acertos) que a codificação *Gray*, precisou de um número menor de neurônios e com isso se mostrou mais eficiente. Através da análise gráfica podemos perceber também que a técnica AC se mostrou nitidamente menos eficiente que as demais, tendo resultado em um número mediano de classificações corretas menor que o de todas as outras formas de codificação.

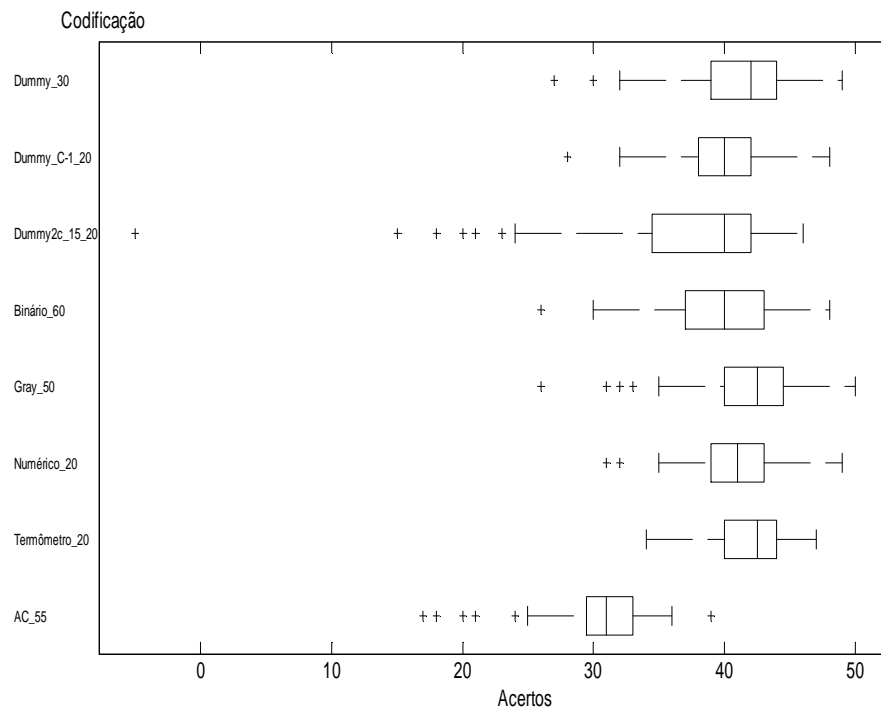


Figura 4-25 – Comparação entre as codificações - *Córneas*.

Tabela 4-28 – Comparação entre as codificações – número de classificações corretas, dados *Córneas*.

Codificação	Neurônios	Média
<i>Dummy</i>	30	41,32
<i>Dummy</i> 1 - de - (c - 1)	20	39,87
<i>Dummy</i> Dupla camada	15-20	36,44
Binário	60	39,89
<i>Gray</i>	50	41,63
Numérico	20	40,91
Termômetro	20	41,96
AC	55	30,63

4.2.2 Banco de dados *Seguros*

A amostra de teste deste banco de dados continha 221 casos, que não tinham sido utilizados anteriormente por nenhum dos modelos. Para este banco o resultado na amostra de teste é exposto na Tabela 4-29 e na Figura 4-26. Alcançamos melhor resultado novamente com a codificação *Termômetro*, a qual possui mediana no valor de 187 acertos; contudo, as demais codificações, exceto *Dummy* com dupla camada, chegaram a valores muito próximos, podendo qualquer uma dessas ser usada para esse fim com eficiência.

Quanto à codificação *Dummy* com dupla camada, nota-se que a mesma parece ter sofrido de *overfitting*, que é o sobreajuste da rede devido a um número excessivo de neurônios utilizados na camada oculta, o que levou a erros de previsão muito altos (cf. Cerqueira *et al.*, 2001). Daí, o que parecia ser uma boa escolha de neurônios na amostra de treino devido à sua performance, não se mostrou satisfatória para seu uso na amostra de teste. Usamos a técnica de *validação cruzada* para a parada do treinamento; esta técnica é geralmente recomendada para evitar o *overfitting*, mas não parece ter sido de grande auxílio neste problema. De qualquer forma, este resultado nos mostra que é desnecessário o uso de duas camadas ocultas, já que os melhores resultados foram conseguidos por modelos mais simples, usando apenas uma camada oculta.

Neste banco de dados, a simulação por AC se mostrou uma boa opção para a codificação dos dados; o número mediano de acertos em cada rodada foi aproximadamente equivalente aos das outras formas de codificação, mas a dispersão dos resultados nestas rodadas foi menor. É possível que o desempenho melhor do AC (em relação ao banco de dados anterior) seja devido à maior quantidade de casos agora disponíveis.

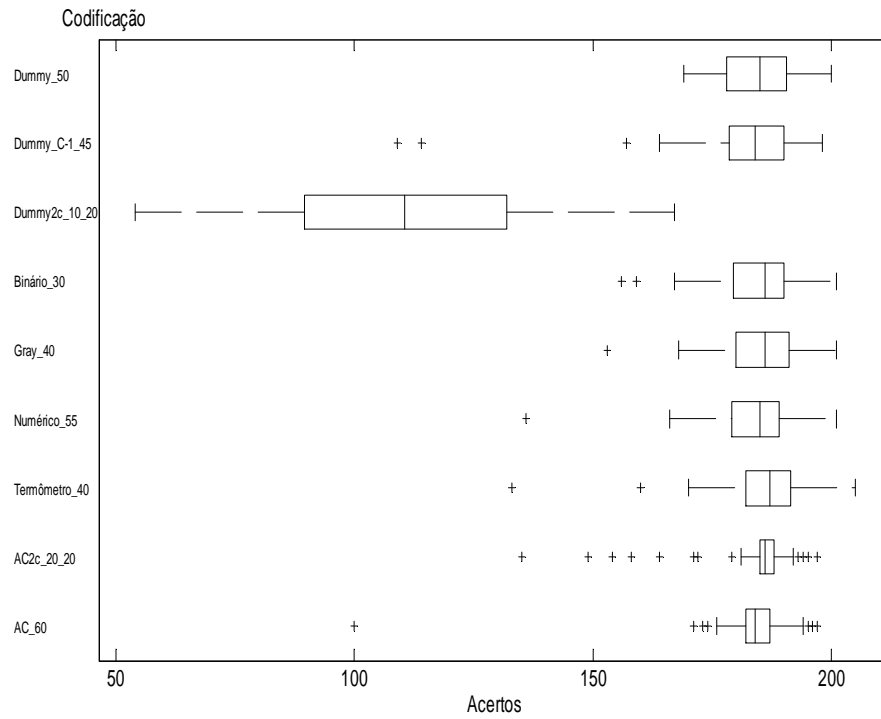


Figura 4-26 – Comparação entre as codificações - Seguros.

Tabela 4-29 – Comparação entre as codificações – número de classificações corretas no banco de dados *Seguros*.

Codificação	Neurônios	Média
<i>Dummy</i> 1-de-c	50	184,57
<i>Dummy</i> 1-de-(c-1)	45	182,20
<i>Dummy</i> Dupla Camada	10-20	110,76
Binário	30	184,89
Gray	40	185,20
Numérico	55	183,91
Termômetro	40	185,86
AC Dupla Camada	20-20	184,94
AC	60	183,43

4.2.3 Banco de dados *Consumo*

A amostra de teste deste banco de dados continha 86 casos, que não tinham ainda sido utilizados em nenhum modelo. Os resultados obtidos em 100 simulações de cada modelo são apresentados na Tabela 4-30 e Tabela 4-31, e na Figura 4-27 e Figura 4-28 (em termos do erro MAPE e do erro MSE, respectivamente). Como podemos observar, o MAPE nos dá como melhor resultado a codificação *Gray*, que apresenta a menor mediana do erro; o MSE, contudo, indicou a codificação *Numérica* como a de melhor desempenho. Uma vez que estas duas medidas de erros se baseiam em conceitos distintos - erro absoluto e erro quadrático - esta diferença de resultados é frequentemente observada. Vale observar que os engenheiros eletricitas que trabalham com previsão de consumo preferem usar o erro MAPE, por ser mais interpretável que o MSE (Hippert *et al*, 2001).

A codificação AC mostrou um desempenho razoável neste banco de dados, tanto na avaliação com MAPE quanto na avaliação por MSE, não se destacando das demais formas de codificação.

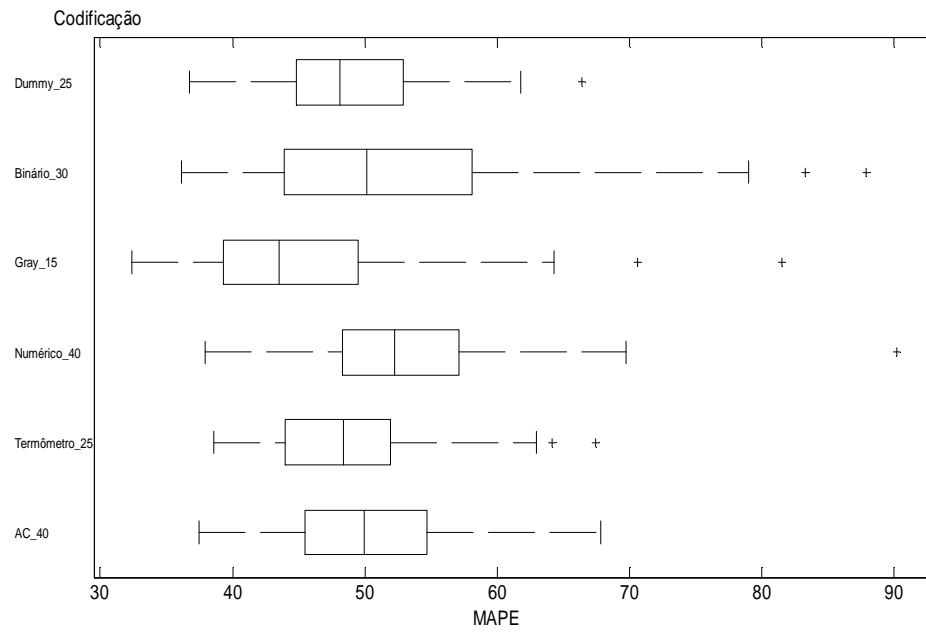


Figura 4-27 – Comparação entre as codificações do BD Consumo – Erro MAPE.

Tabela 4-30 – Comparação entre as codificações – Erro MAPE.

Codificação	Neurônios	MAPE (mediana)
Dummy	25	48,14
Binário	30	50,17
Gray	15	43,50
Numérico	40	52,25
Termômetro	25	48,36
AC	40	49,95

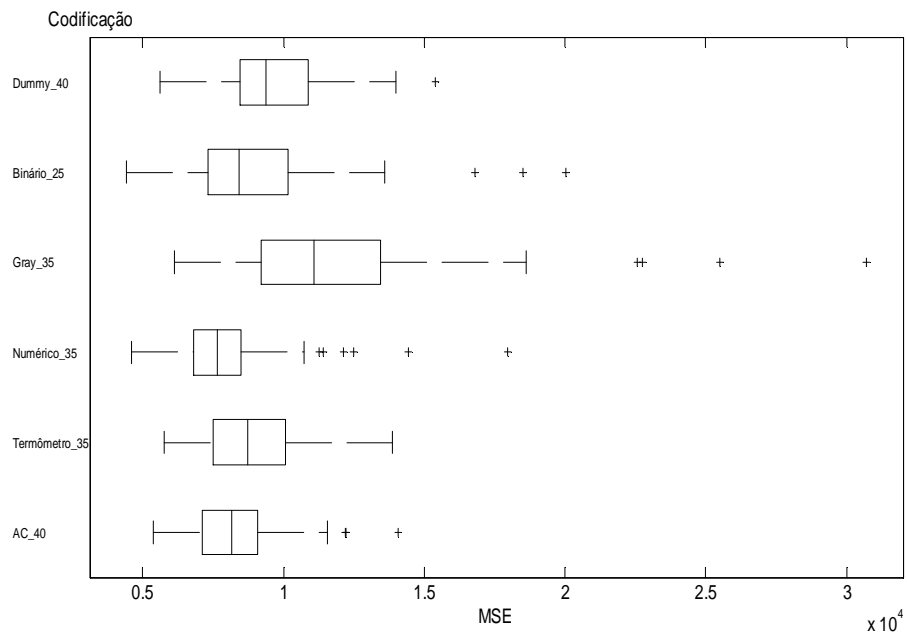


Figura 4-28 – Comparação entre codificações do BD Consumo – erro MSE.

Tabela 4-31 – Comparação entre as codificações – Erro MSE.

Codificação	Neurônios	MSE (mediana)
Dummy	40	9375,95
Binário	25	8432,35
Gray	35	11086,50
Numérico	35	7641,10
Termômetro	35	8723,95
AC	40	8165,45

5 Discussão e conclusão

Analisando os resultados apresentados na seção anterior, concluímos que, nos bancos de dados utilizados, a codificação das variáveis de entrada qualitativas usando os fatores de AC não mostrou vantagens em relação às outras formas de codificações, em termos do desempenho dos modelos em amostras de teste. Nos bancos de dados *Seguros* e *Consumo*, os resultados alcançados por esta técnica foram próximos aos resultados obtidos pelas demais técnicas empregadas; no banco de dados *Córneas*, porém, o resultado da AC foi nitidamente pior que os das outras técnicas, já que o valor alcançado por sua mediana na Figura 4-25 nos dá como resultado um valor aquém do desejado, ou seja, destoa das demais técnicas apresentadas para o banco em questão, fato esse que associamos ao pequeno tamanho da amostra empregada (*Córneas* foi a menor base utilizada e continha o menor número de casos) e a AC ser comumente aplicada em grandes matrizes de dados.

A principal utilização da técnica de AC é na análise exploratória dos dados, uma vez que permite a representação de dados qualitativos de forma quantitativa, em espaço de dimensões reduzidas, o que resulta em gráficos como o da Figura 3-3.

A redução de dimensionalidade conseguida com esta técnica, contudo, não se mostrou útil nos bancos de dados utilizados neste trabalho. O uso de AC introduziu dificuldades no trabalho (fatores de AC são evidentemente mais difíceis de calcular, e mais difíceis de entender conceitualmente do que, por exemplo, um vetor de *dummies*), mas o desempenho das redes não foi melhorado, em relação ao das outras formas de codificação. É possível que o mau desempenho da AC tenha ocorrido porque os bancos de dados usados têm pequeno número de variáveis; como apresentado na Seção 3.4.6, a técnica mostra mais eficiência ao ser aplicada em matrizes de dados de grandes dimensões. Para que esta hipótese seja confirmada, serão necessários novos estudos com bancos de dados maiores, da ordem de centenas de milhares de casos e milhares de variáveis; esta possibilidade merece ser explorada em trabalhos futuros sobre o tema.

REFERÊNCIAS

- Alves, L. B.; Belderrain, M. C. N.; Scarpel R. A. 2007. Tratamento Multivariado de Dados por Análise de Correspondência e Análise de Agrupamentos. *XIII ENCITA - Encontro de Iniciação Científica e Pós-Graduação do ITA*.
- Araújo, K. A. M.; Lotufo, A. D. P. 2009. Previsão de cargas elétricas através de uma rede neural *feedforward* com treinamento por *Levenberg-Marquardt* utilizando as *toolboxes* do Matlab. *Anais do XXI Congresso de Iniciação Científica da UNESP. São José do Rio Preto*.
- Baesens, B.; Setiono, R.; Mues, C. & Vanthienen, J. 2003. Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation. *Management Science*, 49(3), pp. 312-329.
- Baker, Samuel L. 2006. Dummy Variables. Pp.1-6. Disponível em <http://hspm.sph.sc.edu/courses/J716/pdf/716-6%20Dummy%20Variables%20and%20Time%20Series.pdf>. Acesso em 28 de abril de 2012.
- Barbosa, A. H.; Freitas, M. S. R.; Neves; F.A. 2005. Confiabilidade Estrutural Utilizando o Método de Monte Carlo e Redes Neurais. *Revista Escola de Minas*. 58(3). Pp. 247-255. Ouro Preto.
- Braga, A. P.; Carvalho, A. C. P. L. F.; Ludemir, T. B. 2000. *Redes Neurais Artificiais: teoria e aplicações*. Rio de Janeiro: LTC.
- Carvalho, M. S.; Struchiner, C. J. 1992. Análise de Correspondência: Uma Aplicação do Método à Avaliação de Serviços de Vacinação. *Cadernos de Saúde Pública*. Rio de Janeiro, 8 (3): 287-301.
- Cazes, Thiago Broerman. 2005. Interpretação Baseada em Conhecimento de Imagens de Sensores Remotos de Alta Resolução. Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da PUC-Rio. Rio de Janeiro. Disponível em <http://www.maxwell.lambda.ele.puc-rio.br>. Acesso em 27 de abril de 2012.
- Cerqueira, Eduardo O. de; Andrade, João C. de; Poppi, Ronei J. and Mello, Cesar. 2001. Redes neurais e suas aplicações em calibração multivariada. *Química Nova*, São Paulo, v. 24, n. 6.
- Coutto Filho, Milton B. Do; Souza, Julio C. Stacchini; Schilling, Marcus Th. 2007. Sobre o problema da integração generalizada de dados. *Sba Controle & Automação*, Natal, v. 18, n. 1.
- Cunha Jr., Marcus Vinícius Moretti da. 1997. Análise Multidimensional de Dados Categóricos: A Aplicação das Análises de Correspondência Simples e Múltipla em Marketing e Sua Integração com Técnicas de Análise de Dados Quantitativos. *Caderno de Estudos do PPGA/EA/UFRGS*, nº 16/97, p. 43.

Cyrino, Carolina Dutra. 2011. Análise Geométrica de Dados através de Análise de Correspondência Múltipla. Monografia apresentada ao curso de Estatística da Universidade Federal de Juiz de Fora.

Czermainski, A.B. 2004. Análise de correspondência. Piracicaba. Seminário apresentado na ESALQ – USP. Disponível em <http://ce.esalq.usp.br/tadeu/anabeatriz.pdf>. Acesso em 01 de abril de 2012.

Demuth, H.; Beale, M.; Hagan, M. *Neural Network Toolbox™ 6 User's Guide*. The MathWorks, 2012. Disponível em http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf. Acesso em 28 de abril de 2012.

Freitas, H.; Janissek-Muniz, R. 2008. Aprofundando a análise de dados através da Análise de Correspondência. *Revista Quanti & Quali*.

Gomes, Cristiane Ruiz. 2006. Localização de falhas em linhas de transmissão utilizando decomposição harmônica e redes neurais artificiais. Dissertação submetida à banca examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA. Belém. Disponível em http://www.repositorio.ufpa.br:8080/jspui/bitstream/2011/1645/4/Dissertacao_FalhasLinhasTransmissao.pdf
Acesso em 10 de junho de 2012.

Gonçalves, A. R. 2011. Otimização em ambientes dinâmicos com variáveis contínuas empregando algoritmos de estimação de distribuição. Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Campinas. Disponível em ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/theses/andre_goncalves.pdf
Acesso em 04 de junho de 2012.

Gonçalves, M. T.; Santos, S. R. dos. 2009. Aplicação da Análise de Correspondência à Avaliação Institucional da Fecilcam. IV EPCT – Encontro de Produção Científica e Tecnológica.

Greenacre, Michael; Blasius, Jörg. 2006. *Multiple Correspondence Analysis and Related Methods*. Chapman & Hall/CRC.

Greenacre, Michael; Hastie, Trevor. 1987. The Geometric Interpretation of Correspondence Analysis. *Journal of the American Statistical Association*. Vol. 82, No. 398, pp. 437-447. Published by: American Statistical Association.

Guedes, T. A.; Ivanqui, I. L.; Martins, A. B. T.; Cochia, E. B. R. 1999. Seleção de variáveis categóricas utilizando análise de correspondência e análise procrustes. *Revista Acta Scientiarum*, 21(4), pp. 861-868. ISSN 1415-6814.

Haykin, S. 2001. *Redes neurais princípios e prática*. Porto Alegre: Bookman.

Hippert, H.S.; Pedreira, C.E.; Souza, R.C. 2001. Neural networks for short-term load forecasting: a review and evaluation. *IEEE Transactions on Power Systems*, vol.16, nº1, pp.44-55. doi:10.1109/59.910780

Jeon, Yunho. Choi, Chong-Ho. 1999. Thermometer coding for multilayer perceptron learning on continuous mapping problems. IJCNN'99. *International Joint Conference on Neural Networks*, vol.3, no., pp.1685-1690 vol.3, 1999.
doi: 10.1109/IJCNN.1999.832628

Kak, Subhash. Unary Coding for Neural Network Learning. 2010. Disponível em <http://arxiv.org/ftp/arxiv/papers/1009/1009.4495.pdf>. Acesso em 28 de abril de 2012.

Lamelas, Carlos. 2012. Sistema de Bonificação no Seguro Automóvel (Bónus-Malus). Disponível em http://www.omundodosseguros.com/carlos.lamelas/carlos_lamelas_sistema_bonus_malus.html. Acesso em 13/05/12.

Larsen, Pia Veldt. 2008. Module 8: Selecting regression models. Master of Applied Statistics – Syddansk Universitet. Disponível em <http://statmaster.sdu.dk/courses/st111/module08/>. Acesso em 23 de abril de 2012.

Leone Filho, Marcos. 2008. Aula 16: RNAs: Arquiteturas de Rede e Modelos de Aprendizagem 1. Tópicos em Computação I - 2008/1. Disponível em http://www.cose.fee.unicamp.br/~marcos/aulas/unifeob/TOP_I/aula16.pdf. Acesso em 23 de abril de 2012.

Lopes, M.L.M. 2005. Desenvolvimento de redes neurais para previsão de cargas elétricas de sistemas de energia elétrica. Tese de doutorado submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia de Ilha Solteira – UNESP. Ilha Solteira. Disponível em http://www.athena.biblioteca.unesp.br/exlibris/bd/bis/33004099080P0/2005/lopes_mlm_dr_ilha.pdf
Acesso em 30 de maio de 2012.

Mathworks. n.d. Design and simulate neural networks. Disponível em <http://www.mathworks.com/products/neural-network/index.html>. Acesso em 21 de março de 2012.

Mathworks. Matlab® - Licença nº 334543. Conforme: http://www.mathworks.com/support/contact_us/ts/popups/prodrel.html

Mathworks. n.d. MATLAB – The Language of Technical Computing. Disponível em <http://www.mathworks.com/products/matlab/>. Acesso em 21 de março de 2012.

Mendes, Débora Cosme Pereira. 2009. Diagnóstico da evasão ocorrida na UFRN para alunos que ingressaram entre 2000 e 2008: uma aplicação de Análise de Correspondência. Monografia apresentada ao Departamento de Estatística da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do título de Estatístico. Natal. Disponível em

- http://www.comperve.ufrn.br/conteudo/observatorio/arquivos/monografias/debora_cosme_pereira_mendes.pdf. Acesso em 11 de junho de 2012.
- Mingoti, Sueli Aparecida. 2005. *Análise de Dados Através de Métodos de Estatística Multivariada: Uma Abordagem Aplicada*. Belo Horizonte: Editora UFMG. Pp. 257-266.
- Missio, Fabrício. Jacobi, Luciane Flores. 2007. Variáveis dummy: especificações de modelos com parâmetros variáveis. *Revista Ciência e Natura*, vol. 29, nº 1.
- Núcleo de Computação Eletrônica (NCE) – UFRJ. Laboratório de Inteligência Computacional (LabIC). n.d. Dicas para a Configuração de Redes Neurais. Disponível em http://www.labic.nce.ufrj.br/downloads/dicas_cfg_rna.pdf. Acesso em 22 de abril de 2012.
- O’Neal, Monte R. Engel, Bernard A. Ess, Daniel R. Frankenberger, Jane R. 2002. AE — Automation and Emerging Technologies: Neural Network Prediction of Maize Yield using Alternative Data Coding Algorithms. *Biosystems Engineering*, Volume 83, Issue 1, Pages 31-45, ISSN 1537-5110, 10.1006/bioe.2002.0098.
- Pamplona, Admur Severino. 1998. *Análise de Correspondência para Dados com Estrutura de Grupo*. Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como requisito parcial para obtenção do Título de Mestre em Estatística. Campinas – SP.
- Sampaio, Tiago Pitchon. 2009. *Ajuste de Histórico de Produção Usando Redes Neurais Artificiais*. Projeto final submetido ao corpo docente de Engenharia de Petróleo da Escola Politécnica da UFRJ. Rio de Janeiro. Disponível em <http://www.petroleo.ufrj.br/lorde/?q=node/56>. Acesso em 10 de junho de 2012.
- Santos, Ana Carla dos Oliveira. 2000. *Tolerância a Falhas para Sistemas Embarcados*. Trabalho de Graduação em Engenharia da Computação. Universidade Federal do Pernambuco. Recife. Disponível em www.cin.ufpe.br/~tg/2000-1/acos.doc. Acesso em 05 de junho de 2012.
- Sarle, Warren S. 2002. Questions. SAS. Cary, NC, USA. Disponível em <ftp://ftp.sas.com/pub/neural/FAQ2.html>. Acesso em 21 de março de 2012.
- Silva, Renato Maia. 2005. *Redes Neurais Artificiais aplicadas à Detecção de Intrusão em Redes TCP/IP*. Dissertação de Mestrado apresentada à Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro. Disponível em <http://www.maxwell.lambda.ele.puc-rio.br/>. Acesso em 22 de abril de 2012.
- Souza; Augusto Carvalho; Bastos, Ronaldo Rocha; Vieira, Marcel de Toledo. 2010. *Análise de Correspondência Simples e Múltipla para Dados Amostrais Complexos*. *Anais do 19º SINAPE – Simpósio Nacional de Probabilidade e Estatística*.
- Steiner, Maria Teresinha Arns et al. 2007. Extração de regras de classificação a partir de redes neurais para auxílio à tomada de decisão na concessão de crédito bancário. *Pesquisa Operacional* [online]. Vol.27, n.3, pp. 407-426. ISSN 0101-7438.

Steinmetz, T. R. 2009. Extração de Regras de Conhecimento a Partir de Redes Neurais Artificiais Aplicadas para Previsão de Demanda por Energia Elétrica. Dissertação de Mestrado apresentada à Unisinos. São Leopoldo. Disponível em http://bdtd.unisinos.br/tde_arquivos/1/TDE-2009-07-21T094737Z-778/Publico/SteinmetzTarcisioRobertoComputacao.pdf. Acesso em 23 de abril de 2012.

Tocci, Ronald J.; Widmer, Neal S.; Moss, Gregory L. 2007. *Sistemas Digitais - princípios e aplicações*. 10ª Ed. São Paulo: Pearson Prentice Hall.

TRENN, S. 2008. Multilayer Perceptrons: Approximation Order and Necessary Number of Hidden Units. *IEEE Transactions on Neural Networks*, vol.19, no.5, pp.836-844. doi:10.1109/TNN.2007.912306

Tun, U Aung That. 2007. Binary To Gray Code Converter Department of Electrical Power Technological University Pakokku. Disponível em <http://www.most.gov.mm/techuni/media/BinaryToGrayCodeConverter.pdf>. Acesso em 10 de junho de 2012.

UCLA: Academic Technology Services, Statistical Consulting Group. n.d. What is Dummy Code? Disponível em http://www.ats.ucla.edu/stat/mult_pkg/faq/general/dummy.htm. Acesso em 21 de março 2012.

Winandy, Charles-Edouard; Borges Filho, Estillac; Bento, Lisânia Vieira. 2007. Algoritmos para Aprendizagem Supervisionada. Seminários. Disponível em http://winandy.voila.net/IA_ApSuperv_NotasAula.pdf. Acesso em 04 de junho de 2012.