Universidade de Brasília - UnB
Instituto de Ciências Exatas - IE
Estatística

Recalibração de Redes Neurais Artificiais

Autor: João Gabriel Rodrigues Reis
Orientador: Professor Guilherme Souza Rodrigues

Brasília, DF
2021

João Gabriel Rodrigues Reis

Recalibração de Redes Neurais Artificiais

Monografia submetida ao curso de graduação em Estatísticada Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Estatística.

Universidade de Brasília - UnB

Instituto de Ciências Exatas - IE

Orientador: Professor Guilherme Souza Rodrigues

Brasília, DF

2021

Agradecimentos

À minha mãe, Fátima, por todo amor e apoio dedicados a mim ao longo de todos esses anos e por sempre fazer o que esteve ao seu alcance para me proporcionar uma educação de qualidade. A ela eu dedico todas as conquistas que já obtive, além das que virão.

A toda minha família por estar sempre disponível nos momentos que precisei, sei que invariavelmente poderei contar com eles em períodos difíceis.

Ao longo da minha vida acadêmica pude contar com excelentes professores, que me ajudaram a chegar até aqui. Agradeço especialmente meu orientador, Guilherme Rodrigues, por todos os conselhos e ensinamentos que me guiaram durante boa parte da graduação.

Às grandes amizades que fiz durante o período de graduação, Gauthier, Letícia, Rodrigo e Tamara, que me proporcionaram momentos inesquecíveis e foram excelentes companhias durante nossos almoços do RU.

Aos amigos da *Tukey* que sempre estiveram dispostos me ajudar e compreenderam minha ausência enquanto me dedicava à realização deste trabalho: Eduardo Felipe, Gabriel Reis, Laura Cristina e Marcelo Fleury.

Aos amigos que carrego desde a infância e considero irmãos. Agradeço especialmente à Stéfane, que me ajudou na revisão dos textos deste trabalho.

Por último, mas não menos importante, agradeço à minha namorada, Juliana, por todo o carinho e atenção dedicados a mim, que me deram vitalidade para continuar trabalhando neste projeto.

"É um erro acreditar que é possível resolver qualquer problema importante usando batatas.
(Douglas Adams)

Resumo

Redes Neurais Artificiais são algoritmos computacionais extremamente poderosos que estão em evidência devido a sua enorme flexibilidade, que permite trabalhar com uma grande variedade de problemas não lineares. Porém, como todo modelo, as redes neurais possuem desvantagens, sendo uma delas a frequente falta de uma caracterização probabilística das suas previsões. O presente trabalho apresenta um novo algoritmo baseado em uma técnica conhecida como recalibração, com o objetivo de gerar intervalos de predição para redes neurais em problemas de regressão. O método proposto apresentou resultados satisfatórios na estimação da distribuição de probabilidade em dados simulados. Além disso, a acurácia da rede melhorou sem que houvesse a necessidade de aumentar a complexidade da mesma.

Palavras-chaves: Redes Neurais Artificiais. Regressão. Recalibração.

Sumário

1 Introdução e Justificativa

Redes Neurais Artificiais (RNA) são modelos computacionais inspirados no funcionamento do cérebro humano. Tal como seu arquétipo original, são ferramentas muito poderosas utilizadas para o reconhecimento de padrões. As redes podem ser utilizadas em diversos tipos de problemas, incluindo de classificação e de regressão. Este trabalho tem como foco o uso das RNAs em problemas de regressão, com variável resposta contínua e unidimensional.

Uma das principais vantagens das Redes Neurais, em relação a outras abordagens, é a flexibilidade (*capacidade*, na nomenclatura técnica), uma vez que sua estrutura em camadas permite representar as não linearidades presentes nas relações entre a variável resposta e as covariáveis, levando a previsões frequentemente mais acuradas. Apesar da grande versatilidade das redes neurais, elas estão sujeitas a diversas limitações, dentre as quais destacamos a possível ausência de medidas de incerteza associadas às previsões, a frequente instabilidade resultante da falta de regularidade no espaço das covariáveis - as redes neurais possuem uma estrutura que facilita o superajuste do modelo, levando muitas vezes duas observações parecidas no espaço das entradas à previsões substancialmente distintas - e a dificuldade de interpretação dos parâmetros (*pesos* e *vieses*). Este trabalho busca alternativas para os dois primeiros impasses, fazendo uma recalibração da distribuição preditiva da rede neural por meio da confrontação das distribuições preditivas com suas respectivas observações do banco de validação, de modo a satisfazer a propriedade de cobertura.

2 Revisão da Literatura

2.1 Redes Neurais Artificiais

2.1.1 Introdução

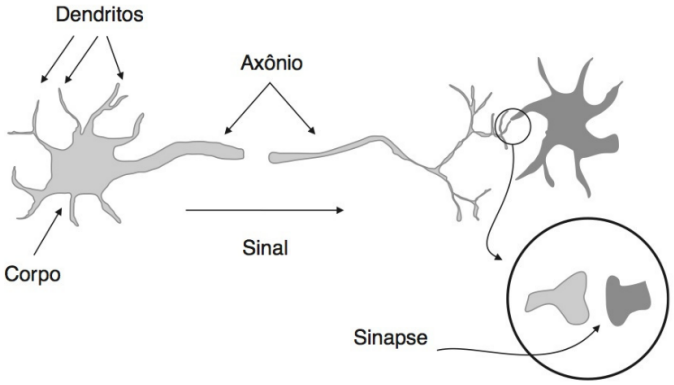
Inteligência Artificial (IA) é um campo vigoroso da Ciência que tem sido largamente explorado por profissionais de diferentes áreas do conhecimento. O conceito de IA é extremamente abrangente englobando diferentes técnicas. Segundo McCarthy (1998) a IA pode ser definida como "a ciência e a engenharia de fazer máquinas inteligentes, particularmente programas de computador inteligentes". Sendo a natureza de inteligência subjetiva e imprecisa, o autor parte do argumento de não haver nenhuma definição desassociada com a inteligência humana. Dessa maneira, as Redes Neurais Artificiais surgiram como uma tentativa de reproduzir a capacidade humana de absorver, processar e responder a informações em máquinas, utilizando o próprio cérebro humano como protótipo para a criação de algoritmos capazes de aprender a partir da experiência.

Para Haykin (2009), "o cérebro é um computador altamente complexo, não linear e paralelo (sistema de processamento de informações)". A unidade básica do cérebro, o neurônio, composto de dendritos, corpo e axônio, serviu como ponto de partida para o desenvolvimento do primeiro modelo neural, proposto por Mcculloch e Pitts (1943). Posteriormente, esse modelo foi aperfeiçoado por Rosenblatt (1957), e ficou conhecido como Perceptron, o primeiro modelo formal de um neurônio que, diferentemente do modelo anterior, era capaz de aproximar funções lineares por intermédio de um conjunto de pesos que podem ser estimados, introduzindo, portanto, o conceito de aprendizado.

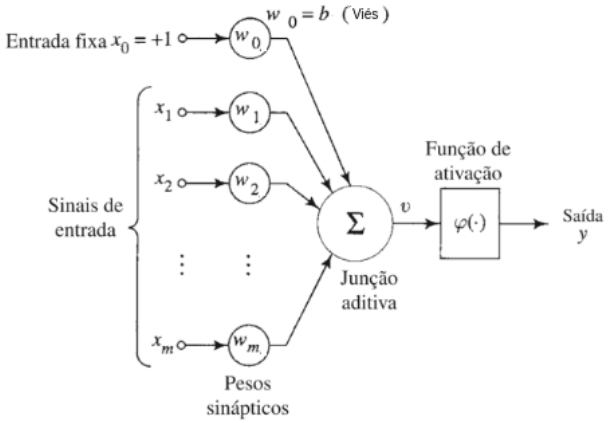
2.1.2 Perceptron

Simplificadamente, o funcionamento de um neurônio se dá pela seguinte forma: os dendritos recebem um estímulo nervoso de uma fonte externa e o transmite ao corpo celular, onde o sinal será processado, criando-se um novo impulso, que é enviado para os axônios, de onde é finalmente transmitido para os próximos neurônios. O esquema pode ser visualizado na Figura 1a.

Tendo esse mecanismo como base e somando-se ao modelo de McCulloch-Pitts, surgiu o Perceptron. Em uma analogia, o Perceptron recebe os estímulos (valores de entrada), os processa (aplicando transformações lineares e funções de ativação) e passa adiante um valor de saída. Portanto, o modelo implementado por Rosenblatt nada mais é que uma função matemática. Um esquema completo pode ser visto na Figura 1b.



(a) Neurônio biológico simplificado. Fonte: Faceli et al. (2011)



(b) Perceptron. Fonte: Haykin (2009)

Figura 1 – Neurônios. Em (a) temos uma visão simplificada do funcionamento de um neurônio. Em (b), apresentamos o modelo Perceptron, onde uma série de sinais de entrada é recebida, processada e retornada.

Matematicamente o perceptron é definido pelas equações

$$v = \sum_{j=0}^{m} w_j x_j,$$
$$y = \varphi\Big(v\Big),$$

onde y é o valor retornado pelo perceptron, $\varphi(.)$ é a função de ativação, x_j é o valor do *j-ésimo* sinal de entrada (também conhecido como covariável), w_j é o peso sináptico aplicado à *j-ésima* covariável e m é a quantidade de covariáveis. O valor x_0 é fixado em 1 e o peso w_0 é conhecido como viés (na literatura estatística é chamado de intercepto).

Em analogia à célula neural, a função $\varphi(v)$ seria equivalente ao axônio, responsável por permitir ou não a ativação, a depender do estímulo recebido (v). Existem diversas funções de ativação, como a função limiar, sigmoidal, linear e etc.

2.1.3 Arquitetura mlp

Embora o Perceptron tenha revolucionado o contexto do estudo de inteligência artificial, o modelo ainda apresenta uma grande limitação: ele só é capaz de lidar com problemas lineares. Devido a isso o estudo de redes neurais entrou em hiato durante décadas, até que uma nova arquitetura com múltiplas camadas de Perceptrons e um algoritmo capaz de estimar a grande quantidade de pesos presente nessa nova estrutura, o *backpropagation*, foi proposto por Rumelhart, Hinton e Williams (1986).

Assim surgiram as redes *mlp* (*multilayer perceptron*), uma solução inovadora, capaz de lidar com problemas com grau de complexidade muito mais elevado. O modelo conta com múltiplos Perceptrons organizados de forma que as saídas de um servem de entrada para o próximo.

A primeira camada da rede é chamada de camada de entrada e é por ela que os estímulos (covariáveis) são percebidos. As camadas seguintes, com exceção da última, são conhecidas como camadas escondidas, pois não fazem parte da entrada nem saída da rede, porém podem representar importantes padrões presentes nas covariáveis de entrada. Por fim, a última camada, conhecida como camada de saída, é responsável por retornar um valor que geralmente possui um significado prático, o qual depende do problema. Tal propagação sequencial recebe o nome de propagação *feedfoward*. O gráfico apresentado na Figura 2 ilustra o funcionamento de uma rede com uma camada escondida com dois neurônios.

As redes *mlp* também são conhecidas como redes neurais profundas. Mais detalhes sobre essas redes podem ser encontradas no Capítulo 6 do livro de Goodfellow, Bengio e Courville (2016).

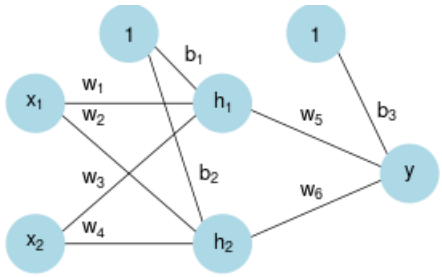


Figura 2 – Arquitetura de uma rede *MLP*.

A propagação feedforward também pode ser compreendida como um conjunto de operações matriciais. A sequência de fórmulas a seguir apresenta a notação matricial da

rede com duas camadas escondidas ilustrada na Figura 2.

$$\begin{aligned}\boldsymbol{a} &= \boldsymbol{W}^{(1)\top}\boldsymbol{x} + \boldsymbol{b}^{(1)},\\ \boldsymbol{h} &= \phi_1(\boldsymbol{a}),\\ y &= \phi_2(\boldsymbol{W}^{(2)\top}\boldsymbol{h} + b^{(2)}),\end{aligned}$$

onde

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, \quad \boldsymbol{h} = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix},$$

$$\boldsymbol{W}^{(1)} = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}, \quad \boldsymbol{W}^{(2)} = \begin{pmatrix} w_5 \\ w_6 \end{pmatrix}, \quad \boldsymbol{b}^{(1)} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \text{e} \quad b^{(2)} = b_3.$$

Uma rede com duas camadas é capaz de representar qualquer função Cybenko (1989). Portanto, a nova arquitetura proposta possui alta flexibilidade, lidando não apenas com problemas lineares. Vale destacar a importância da função de ativação, uma vez que utilizando apenas ligações lineares qualquer rede com múltiplas camadas pode ser facilmente reescrita como uma rede com uma única camada.

2.1.4 Função de Perda

Considere um conjunto de observações $\mathbf{x}$ e $\mathbf{y}$. $\mathbf{x}$ é conhecido na literatura estatística como matriz de delineamento (cada uma das n unidades observacionais aparece em uma linha, enquanto as variáveis ocupam as colunas) e $\mathbf{y}$ é um vetor que armazena os respectivos valores da variável resposta. Desejamos prever o valor de $\mathbf{y}$ em função de $\mathbf{x}$. Ao alimentar uma rede *MLP* com $\mathbf{x}$, queremos obter $\hat{\mathbf{y}}$, que representa a previsão para cada linha $\mathbf{x}$. A função de perda estabelece uma medida de distância entre os valores previstos pela rede $\hat{\mathbf{y}}$ e os valores efetivamente observados, $\mathbf{y}$. Quanto mais ajustada a rede neural estiver menor será o valor da função de Custo, denotada por $J(\theta)$. Existem diferentes funções de custo, que podem ser selecionadas de acordo com o problema em estudo. Dentre as mais conhecidas destaca-se o erro quadrático médio, que possui a forma

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

onde θ é o vetor de todos os pesos da rede neural.

2.1.5 Otimização

O aprendizado da rede consiste em obter o melhor conjunto de parâmetros θ. Ou seja, é necessário obter o vetor θ que minimize a função $J(\theta)$. Para isso, o ajuste da rede neural se divide em duas fases: *feedforward* e *backpropagation*. A primeira já foi

abordada na seção anterior, enquanto a segunda é responsável pelo cálculo do gradiente da função $J(\theta)$ em relação a θ, responsável por apontar a direção para a qual os pesos devem ser incrementados para que a função de custo diminua. Usando múltiplas iterações de *feedforward* e *backpropagation* a rede aos poucos aprende a combinação de pesos ideal.

Existem diversas técnicas de otimização e dentre as mais conhecidas destacam-se o *Stochastic Gradient Descent* e o *Adam*. Este e outros algorítimos de otimização famosos podem ser vistos em detalhe no Capítulo 8 do livro de Goodfellow, Bengio e Courville (2016)

2.1.6 Aprendizagem representacional

As redes neurais *feedforward* podem ser pensadas como uma forma de aprendizado representacional. Na verdade, as camadas escondidas da rede nada mais são que representações do conjunto de covariáveis original. No ensaio escrito por Carvalho (2019), o autor buscava identificar o sexo de indivíduos a partir de suas digitais. Para tanto, o pesquisador contava com um banco de 200 imagens de digitais de homens e mulheres, de onde foram extraídas manualmente por um especialista as informações da densidade de linhas e a contagem de linhas brancas para cada indivíduo e com essas informações tentar predizer o sexo.

Caso o problema fosse abordado utilizando redes neurais, o tamanho amostral poderia aumentar radicalmente. Isso se daria pois a própria rede seria capaz de aprender essas *features* de maneira automática em suas camadas escondidas, eliminando a necessidade de um especialista investigar imagem a imagem. Porém, vale destacar que apesar de informativas as variáveis aprendidas pela máquina quase sempre não possuem algum significado prático para os humanos, diferentemente da densidade de linhas ou contagem de linhas brancas. As *features* aprendidas pela rede são representações da imagem original nas camadas escondidas, e essa ideia pode ser expandida para outros conjuntos de dados além de imagens. Existem estruturas de redes neurais especializadas em aprender subespaços informativos, como os *autoencoders*.

2.2 Técnicas de Recalibração

2.2.1 Diagnóstico em modelos de regressão

Conforme definido por Neter, Wasserman e Kutner (1983), "regressão linear é uma ferramenta estatística que utiliza a relação entre duas ou mais variáveis quantitativas, onde uma variável pode ser predita em função de outra, ou outras". A partir de um conjunto de dados observados, pode-se estimar a relação entre uma variável resposta $\boldsymbol{Y}$ e um conjunto de variáveis explicativas $\boldsymbol{X} = (X_1, \ldots, X_n)$. Esses modelos usualmente

representam apenas versões simplificadas de algum fenômeno de interesse. Portanto, ao ajustar um modelo, é preciso investigar suas propriedades, em um processo conhecido como análise de diagnóstico. Assim é possível determinar se o modelo construído é capaz de reproduzir o conjunto de observações de maneira verossímil e verificar se as previsões do modelo são aproximadamente não viesadas.

Existem diversas técnicas de diagnóstico baseadas nos resíduos, ou seja, na comparação direta entre os valores observados e as respectivas previsões feitas pelo modelo. Alternativamente, podemos basear a avaliação do ajuste na comparação dos valores observados com as respectivas densidades condicionais estimadas pelo modelo. Nessa abordagem, destaca-se o gráfico de probabilidades acumuladas, que permite avaliar a qualidade do ajuste a partir dos histogramas dos valores acumulados das probabilidades de cada observação. Seja y_i o valor da i-ésima observação da variável resposta Y. A probabilidade acumulada pode ser definida como

$$p_i = \Phi_i(y_i|\boldsymbol{x}_i),$$

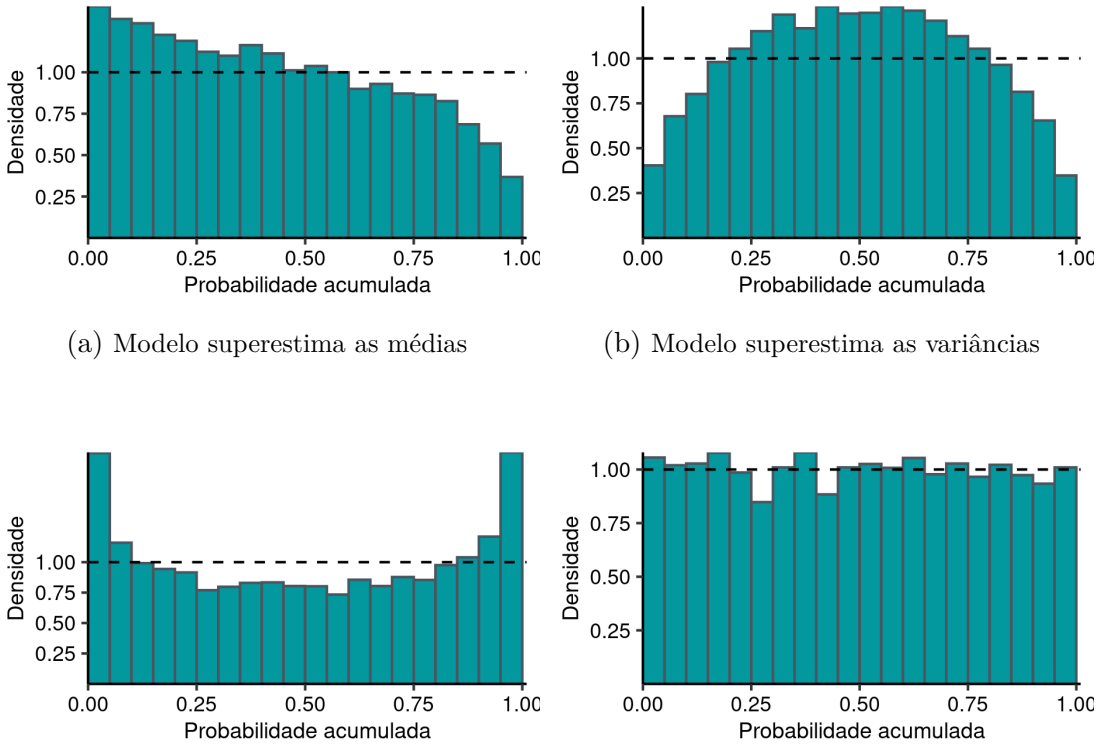
onde Φ_i é a função de distribuição condicional acumulada estimada pelo modelo.

Teorema (Probability Integral Transformation):

Seja X uma V.A. contínua com Função de Distribuição Acumulada (FDA) $F_X(x)$, então $U = F_X(X) \sim$ Uniforme$(0, 1)$. Rizzo (2008).

A partir do teorema *Probability Integral Transformation* deriva-se que se o modelo ajustado estiver bem especificado, então $p_i \sim$ Uniforme(0, 1), para todo i. Desta maneira o histograma dos p_i indica se essa propriedade pode ser observada no modelo.

A Figura 3 apresenta alguns possíveis histogramas de probabilidades acumuladas (também conhecidas na literatura como p-valores) resultantes de diferentes modelos ajustados. Em 3a, observa-se um modelo que superestima suas previsões, uma vez que a maioria das observações encontra-se na região mais à esquerda ($p_i < 0.5$) da respectiva distribuição estimada. Em 3b, nota-se que o histograma possui densidade maior em torno da mediana, e portanto superestima a variabilidade das distribuições (poucas observações estão nas caudas das distribuições). De maneira oposta, no painel 3c, tem-se um modelo que subestima a incerteza. Por fim, o histograma apresentado em 3d apresenta uniformidade nos p-valores, o que não caracteriza uma descalibração. Entretanto, esse gráfico não é suficiente para garantir que o modelo esteja bem calibrado, uma vez que não garante uniformidade para todo i.



(a) Modelo superestima as médias

(b) Modelo superestima as variâncias

(c) Modelo subestima as variâncias

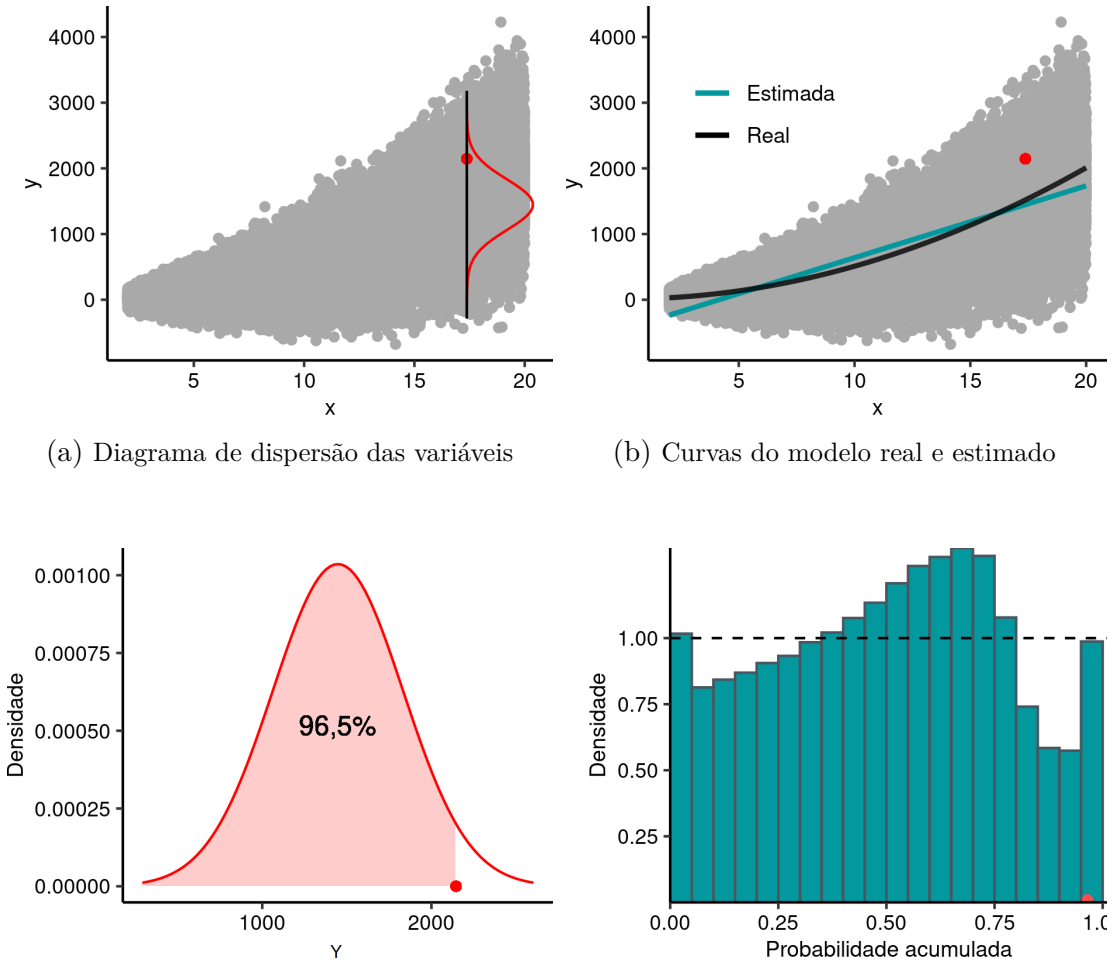
(d) Modelo (possivelmente) bem especificado

Figura 3 – Histograma das probabilidades acumuladas. O painel (a) ilustra um histograma das probabilidades acumuladas de um modelo que superestima as médias. Em (b), temos o histograma de um modelo que superestima as variâncias. Em (c), o histograma indica que o modelo subestima as variâncias. O painel (D) apresenta o histograma de um modelo possivelmente bem ajustado.

2.2.2 *Toy example* 1: modelo Gaussiano quadrático heterocedástico

A título de ilustração, considere um modelo Gaussiano quadrático heterocedástico da forma $Y = 10 + 5X^2 + e$, onde $e \sim N(\mu = 0,\ \sigma = 30X)$. Geramos $n = 100.000$ amostras desse processo para servirem como o conjunto de observações. O gráfico de dispersão pode ser observado na Figura 4a. Em seguida ajustamos um modelo linear homocedástico mal especificado da forma $\hat{Y} = \beta_0 + \beta_1 X$. A reta de regressão estimada se encontra na Figura 4b, onde verifica-se o contraste com a curva da média real que gerou os dados. Uma observação escolhida ao acaso foi destacada em vermelho. Essa observação possui coordenadas $(17,4;\ 2145,3)$ e os valores estimados pelo modelo linear para $\hat{y}$ e $\hat{\sigma}$ são $1447,3$ e $385,2$, respectivamente. A função de densidade condicional estimada para essa observação encontra-se na Figura 4c, na qual a área abaixo da curva realçada em vermelho indica a probabilidade acumulada no ponto $2145,3$ por uma V.A. com distribuição Normal$(1447,3,\ 385,2)$. A Figura 4d apresenta o histograma das probabilidades acumuladas de todas as observações.

Um olhar mais atento à Figura 4 permite constatar a má qualidade de ajuste do



(a) Diagrama de dispersão das variáveis

(b) Curvas do modelo real e estimado

(c) FDA do modelo ajustado no ponto

(d) Histograma das probabilidades acumuladas

Figura 4 – Exemplo ilustrativo. O painel (a) exibe o diagrama de dispersão do modelo quadrático gerado. No painel (b), temos as curvas real (preto) e estimada através da regressão linear (azul). A função de distribuição de probabilidade estimada pelo modelo no ponto destacado em vermelho está disponível no painel (c). O painel (d) expõe o histograma das probabilidades acumuladas do modelo para todas as observações e destaca a FDA do ponto selecionado anteriormente.

modelo original, conforme esperado devido à má especificação. Utilizando o teste de Frosini, reportado na Tabela 1, rejeitou-se a hipótese da distribuição dos p_i's ser uniforme, a 5% de significância. Apura-se pelo histograma das probabilidades ajustadas que há uma combinação de padrões que indicam tanto superestimação quanto subestimação da variância do modelo (em diferentes regiões do espaço da covariável). Além disso, há indícios de vieses positivos e negativos na estimação das médias das distribuições.

Estatística B	P-valor
7,396	<0,001

Tabela 1 – Teste de Frosini para uniformidade dos p_i's.

2.2.3 Recalibração Global

O procedimento de recalibração é uma técnica que usa a informação contida nas probabilidades acumuladas para recalibrar a distribuição preditiva. Isso é, se o gráfico das probabilidades acumuladas aponta para um certo padrão de viés, tal informação pode ser explorada para mitigar tal comportamento indesejado. A sequencia necessária para realizar a recalibração está elucidada a seguir.

Considere o exemplo da Seção 2.2.1, e seja p_i a i-ésima probabilidade acumulada gerada pelo ajuste. Para obter uma amostra da distribuição preditiva recalibrada, seleciona-se primeiramente uma amostra com reposição de tamanho m das probabilidades acumuladas $\boldsymbol{r} = (r_1, r_2, \ldots, r_m)$.

A amostra da distribuição preditiva recalibrada de tamanho m para a i-ésima observação é obtida tomando

$$y^*_{ik} = \Phi_i^{-1}(r_k|\boldsymbol{x}_i), \quad k = 1, 2, ..., m.$$

Realizando o mesmo procedimento para todo i, obtém-se o modelo recalibrado, como uma amostra de Monte Carlo da distribuição recalibrada para cada observação. O i-ésimo valor previsto recalibrado pode ser obtido tomando a média da amostra recalibrada.

$$\hat{y}^*_i = \frac{1}{m}\sum_{j=1}^{m} y^*_{ij}.$$

Caso deseje-se realizar a previsão da variável resposta para uma nova observação x_{new} não contida no conjunto de dados de treinamento, aplicando o método de recalibração, basta realizar o procedimento descrito acima, porém tomando $y^*_{new,k} = \Phi^{-1}_{new}(r_k|\boldsymbol{x}_{new})$. Ou seja, sorteia-se uma nova amostra, $\boldsymbol{r}$, do vetor de probabilidades acumuladas, e em seguida usa-se a distribuição preditiva estimada pelo modelo para obter uma amostra da distribuição recalibrada. Repetindo esse passo m vezes constrói-se uma a distribuição recalibrada empiricamente. Apesar dessa distribuição gerada muitas vezes ser mais fidedigna que a distribuição estimada pelos modelos, a técnica possui como grande desvantagem a perda da interpretabilidade dos parâmetros do modelo. Esse, porém, não é um problema no contexto de Redes Neurais Artificiais, uma vez que os pesos ajustáveis não são facilmente interpretáveis, mesmo sem a recalibração.

A divergência de Kullback-Leibler é uma generalização da ideia de entropia que permite a comparação de informação entre duas distribuições de probabilidades $\mathbf{P}$ e $\mathbf{Q}$. Portanto, se conhecermos a distribuição verdadeira dos dados, podemos verificar se a distribuição preditiva de um modelo melhora após a recalibração. Sua fórmula para funções $\mathbf{P}$ e $\mathbf{Q}$ contínuas pode ser escrita da seguinte forma.

$$D_{KL}(P||Q) = \int_x p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

,

Onde $p(x)$ e $q(x)$ são funções de densidade de probabilidade.

A Figura 5 evidencia alguns resultados do procedimento de recalibração global encontrados quando aplicamos a técnica ao conjunto de dados apresentado no exemplo da seção anterior. Para que o procedimento pudesse ser avaliado, dois novos conjuntos foram gerados, o de validação e teste. O conjunto de validação será utilizado somente para o calculo do vetor de probabilidades acumuladas $\mathbf{p}$, enquanto o conjunto de teste será usado para comparar o desempenho entre o modelo original e o recalibrado.

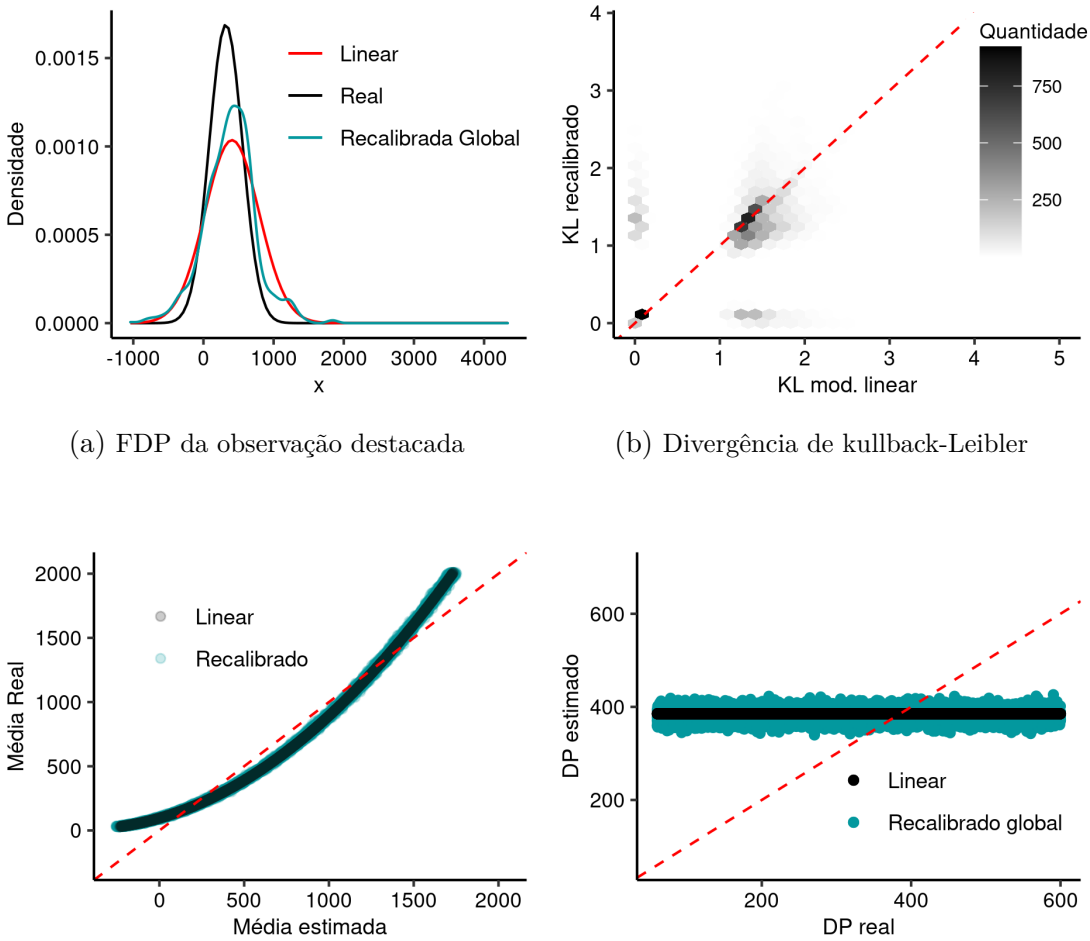
Em 5a observa-se, para a mesma observação destacada em vermelho na Figura 4a, a função de densidade de probabilidade de cada um dos modelos (linear e recalibrado) em contraste com a densidade real gerada. Na Figura 5b, apresentamos a divergência de Kullback-Leibler entre a densidade estimada pelo modelo linear e a densidade real em comparação com a divergência entre a distribuição recalibrada globalmente e a distribuição real. O gráfico das médias reais pelos valores ajustados por cada modelo se encontra na Figura 5c. A Figura 5d mostra o diagrama de dispersão entre o desvio padrão real gerado e o desvio padrão estimado pelos modelos.

De maneira geral, os resultados obtidos após a recalibração global não foram melhores que os obtidos a partir do modelo linear. A densidade recalibrada globalmente para a observação destacada não se aproxima muito da distribuição verdadeira. O modelo recalibrado apresentou uma distância de Kullback-Leibler média aproximadamente 3% menor que o original, indicando um ganho insignificante de performance na estimação da distribuição dos dados. A Figura 5b indica que o modelo recalibrado melhora as estimativas em alguns lugares do espaço de predição e piora em outros. O gráfico de dispersão entre a média verdadeira e a média estimada mostra que os dois modelos geram estimativas bem similares. Além disso nenhum dos dois modelos é capaz de lidar com a heterocedasticidade do fenômeno, produzindo estimativas inconsistentes para o desvio padrão, um bom modelo deveria apresentar uma distribuição dos postos em torno da reta identidade.

O procedimento de recalibração global trouxe muito pouco no ganho de informação em relação a densidade real dos dados, inclusive houve uma perda na capacidade de predição do modelo conforme apresenta a tabela 2. Na seção 2.2.4 será apresentada uma variação mais poderosa do método de recalibração.

Modelo	Erro Quadrático Médio	Distancia de KL
Linear	14804,5	1,174
Recalibrado global	14957,1	1,133

Tabela 2 – Comparação de performance entre o modelo linear original e o recalibrado



(a) FDP da observação destacada

(b) Divergência de kullback-Leibler

(c) Média real por média estimada

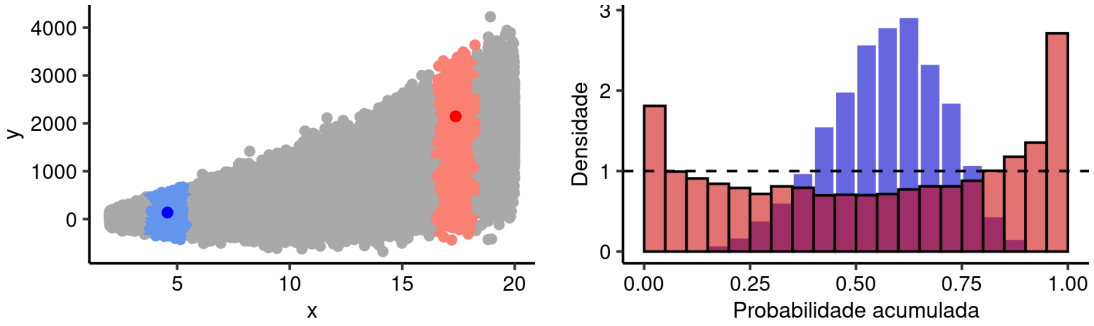
(d) Desvio padrão real por estimado

Figura 5 – Exemplo de recalibração global. Em (a) apresentamos a densidade real (preto), estimada pelo modelo (vermelho) e recalibrada globalmente (Azul). No painel (b), podemos verificar se houve uma melhora na distribuição estimada após a recalibração. No eixo das abscissas temos o valor da divergência de Kullback-Leibler entre o modelo linear e o real, enquanto no eixo das coordenadas temo a divergencia entre o modelo recalibrado global e o real, a quantidade de pontos é indicada pela cor. Na figura (c), está disponível o diagrama de dispersão entre as médias reais e estimada pelos modelos linear(preto) e recalibrado globalmente (azul). A figura (d) exibe o diagrama de dispersão entre o desvio padrão estimado pelos modelos e o valor real.

2.2.4 Recalibração Local

Aprofundando o exemplo apresentado anteriormente, considere agora as duas observações destacadas em tons escuros na Figura 6a. Nota-se que cada um desses pontos se encontra em uma região diferente do eixo x, enquanto as observações realçadas em tons mais claros indicam seus respectivos 10% vizinhos mais próximos. Os histogramas apresentados em 6b mostram a densidade dos p_i's na vizinhança de cada ponto evidenciado. Para cada localidade o histograma apresenta um padrão de viés diferente. Olhando somente a vizinhança de cada ponto é possível notar padrões mais semelhantes ao apresentado

na Figura 3. Na vizinhança do ponto vermelho, é possível ver que o modelo claramente subestima a variância, enquanto a vizinhança do ponto azul tende a superestimá-la. Esse comportamento é esperado devido a suposição de homogeneidade assumida pela regressão linear. Como cada região do espaço pode possuir um padrão de viés distinto, essa informação pode ser acrescentada ao modelo recalibrado.



(a) Diagrama de dispersão das variáveis (b) Probabilidades acumuladas em cada ponto

Figura 6 – Exemplo vizinhança em uma recalibração local. Em (a) destacamos dois pontos selecionados em diferentes regiões do espaço, os pontos coloridos em tons mais claros indicam os 10% vizinhos mais próximos. O Painel (b) apresenta o histograma das probabilidades acumuladas dos 10% vizinhos mais próximos a observação cada um dos pontos destacados em (a).

O procedimento de recalibração explicado na seção anterior é conhecido como recalibração global pois utiliza todas as observações durante a etapa de reamostragem para recalibrar a distribuição preditiva. Um procedimento alternativo, conhecido como recalibração local, pode ser realizado, porém no momento de fazer a amostragem com reposição das probabilidades acumuladas, selecionam-se apenas as observações mais similares à i-ésima observação, capturando assim o comportamento local do modelo. Tal similaridade pode ser verificada tanto no espaço das covariáveis quanto no espaço das previsões. Uma possível abordagem para verificar a similaridade é a distância euclidiana, cuja fórmula que retorna a distância entre as observações a e b é dada por

$$d_{ab} = \sqrt{\sum_{j=1}^{p}(x_{aj} - x_{bj})^2},$$

onde x_{aj} e x_{bj} é o valor da variável j para as observações a e b, respectivamente. Dessa maneira, pode-se selecionar uma proporção π dos quantis aleatorizados mais próximos. Além de selecionar apenas os vizinhos mais próximos, pode-se atribuir pesos para cada um desses vizinhos usando o kernel de Epanechnikov

$$k(d) = \frac{3}{4} \times \left(1 - \left(\frac{d_i}{\max(\mathbf{d})}\right)\right)^2,$$

onde $\mathbf{d}$ é um vetor com a distância da observação para todos os seus π vizinhos mais próximos. Dessa maneira as observações mais parecidas possuem maior probabilidade de

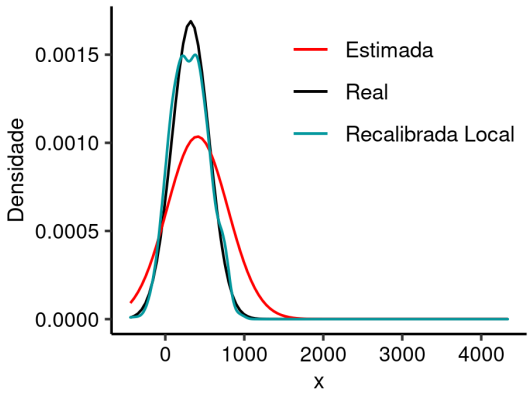
serem selecionadas, gerando assim a amostra das probabilidades acumuladas $\boldsymbol{r}$. Realizando o procedimento análogo ao relatado na recalibração global, obtém-se a amostra de quantis y^*_{ik}, porém agora recalibrada localmente. Note que o procedimento localizado generaliza a versão globalizada, dado que as abordagens se equivalem quando os referidos pesos são fixados em uma valor constante (igual para todas as observações).

Dois pontos a respeito da recalibração local merecem destaque: a magnitude das covariáveis e a "maldição da dimensionalidade".

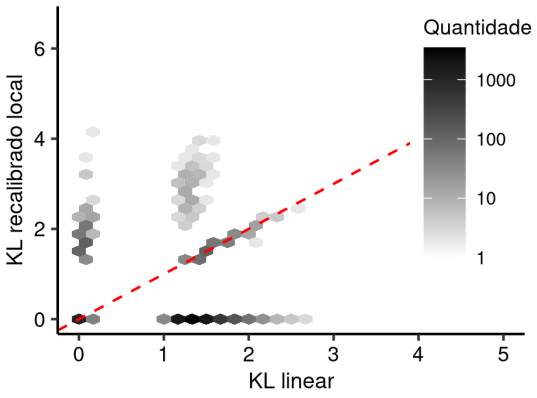
No momento de fazer a reamostragem dos p_i's, na recalibração local, somente as observações mais parecidas com a i-ésima serão consideradas. Dessa forma, a unidade de medida das covariáveis é um fator que pode tornar um atributo indevidamente mais relevantes que os demais. Como exemplo, imagine que deseja-se realizar o procedimento de recalibração em um espaço de duas covariáveis, idade e altura. Dependendo da escala utilizada para a variável altura (centímetros, metros, quilômetros, ...) esse atributo pode tornar-se mais ou menos relevante ao calcular a distância desse vizinho para os demais. Ou seja, as medidas de distância são afetadas pelas escalas dos dados. Uma alternativa seria realizar uma padronização das covariáveis antes de calcular a distância entre os pontos.

A maldição da dimensionalidade é um problema bastante discutido na literatura. Calcular a distância entre as observações pode se tornar bastante difícil conforme a dimensão aumenta. Beyer et al. (1999) mostram que, sob certas condições, com o aumento da dimensionalidade a distância para o vizinho mais próximo se aproxima da distância para o vizinho mais distante, em outras palavras, o contraste das distancias entre dois pontos se torna inexistente. Portanto, realizar o procedimento de recalibração no espaço das covariáveis pode se tornar muito complexo dependendo da quantidade de variáveis preditoras adotadas no modelo.

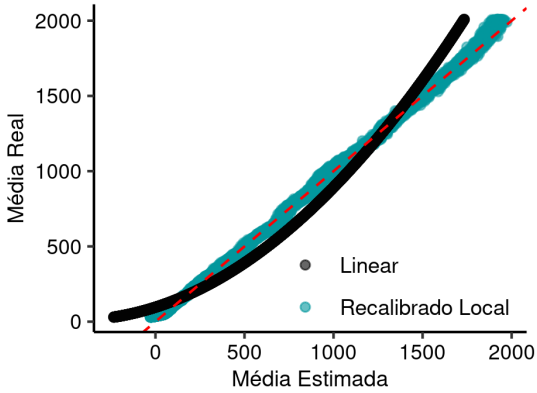
Foi dito que a localização dos vizinhos pode ser feita tanto no espaço das covariáveis quanto no espaço das predições. Cada uma das alternativas conta com vantagens e desvantagens. Realizar a recalibração no espaço das covariáveis faz com que, usualmente, o pesquisador tenha que buscar alternativas para lidar com os dois problemas citados anteriormente, porém esse procedimento garante que o viés local do histograma dos p_i's seja utilizado na reamostragem. Fazer a localização no espaço das previsões evita tanto o problema da escala dos dados quanto de dimensão, uma vez que a previsão terá sempre uma dimensão. Entretanto, essa abordagem não garante que os vizinhos selecionados para a reamostragem sejam de fato parecidos, pois apenas o atributo alvo está sendo levado em consideração. Isso implica que o padrão de viés local apontado pelo histograma não necessariamente retrata o viés local verdadeiro.



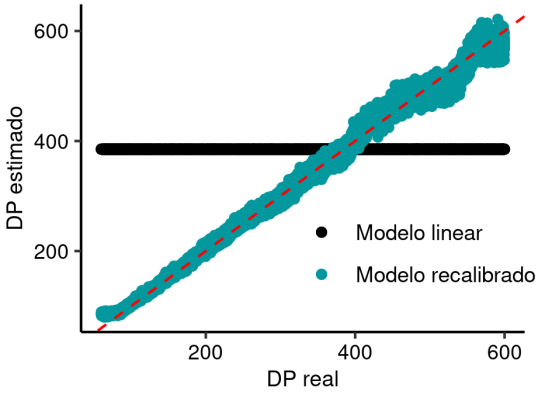
(a) Densidade para a observação destacada



(b) Diagrama de dispersão das variáveis



(c) Curvas do modelo real e estimado



(d) Curvas do modelo real e estimado

Figura 7 – Exemplo de recalibração local. Em (a), estão disponíveis a densidade real (preto), estimada pelo modelo (vermelho) e recalibrada localmente (Azul). Em (b), está expresso o valor da divergência de Kullback-Leibler entre o modelo linear e o real no eixo das abscissas, enquanto no eixo das coordenadas, temos a divergência entre o modelo recalibrado global e o real. A figura (c) expõe o diagrama de dispersão entre as médias reais e estimada por cada um dos modelos (linear e recalibrado localmente). Em (d), temos um diagrama de dispersão entre o desvio padrão estimado pelos modelos e o valor real.

Retomando o exemplo anterior, percebe-se que as estimativas feitas por meio da recalibração local se saíram muito melhores que as do modelo linear. Na Figura 7a nota-se que o modelo recalibrado localmente produz uma estimativa da densidade melhor para a observação destacada na Figura 4a. Na Figura 7b, as distâncias de Kullback-Leibler ficaram em sua maioria abaixo da reta identidade, assinalando que existe um ganho de informação ao utilizar-se a distribuição recalibrada em relação a densidade estimada pelo modelo. Alinhado a isso, a Tabela 3 mostra uma grande redução na distancia de KL média, indicando um ganho de informação satisfatório. Na Figura 7c, o diagrama de dispersão mostra que a média prevista pelo modelo recalibrado se aproxima bem da média real (os pontos se encontram próximos a reta identidade). O modelo de recalibrado localmente

é capaz de capturar a heterocedasticidade do fenômeno, conforme aponta a Figura 7d, porém para valores mais altos o modelo tende a subestimar a variância. Além disso, a Tabela 3 mostra que o erro quadrático médio do modelo recalibrado é cerca de 70% menor que o modelo original.

Vale destacar que o ajuste do modelo recalibrado localmente depende fortemente do tamanho da vizinhança escolhida para a reamostragem dos p_i's. Nesse exemplo selecionou-se os 10% vizinhos mais próximos a cada observação para realizar a recalibração. Caso a vizinhança selecionada seja de 100% o método de recalibração local será igual ao global. Caso a proporção de vizinhos seja muito baixa a técnica terá pouca informação na reamostragem dos p_i's e a estimação a densidade recalibrada pode ser prejudicada.

Modelo	Erro Quadrático Médio	Distancia de KL
Linear	14804,5	1,174
Recalibrado local	400,8	0,193

Tabela 3 – Desempenho dos modelos linear original e recalibrado localmente.

3 Redes Neurais Artificiais Recalibradas

Redes neurais artificiais são modelos bastante flexíveis, capazes de se ajustarem a várias relações não lineares entre as covariáveis de entrada e a variável resposta. Porém, a depender da implementação, esses modelos não provêm nenhuma medida de incerteza associada à sua previsão.

Nas Seções 3.1 e 3.2 introduzimos um algoritmo para viabilizar a mensuração da incerteza associada às previsões. A solução é composta de 3 etapas principais: ajuste da rede neural, previsão para o conjunto de teste por meio da RNA ajustada e geração de amostras de Monte Carlo da distribuição preditiva recalibrada.

3.1 Ajuste da Rede Neural

O primeiro passo para o ajuste da rede consiste em separar o conjunto de dados em três partes (treinamento, validação e teste). O banco de treinamento é utilizado para ajustar o modelo, o de validação é utilizado tanto para verificar a qualidade do ajuste quanto para realizar o procedimento de recalibração da distribuição preditiva. Por fim, o de teste serve para avaliar o desempenho em dados não explorados no processo de modelagem.

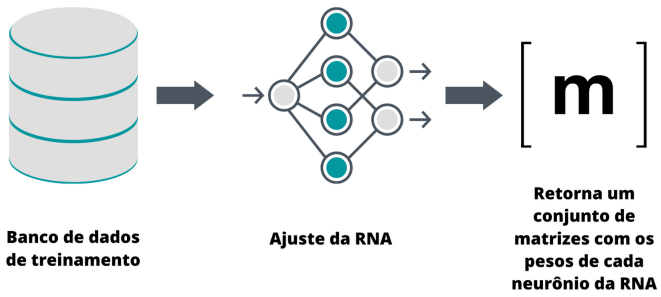


Figura 8 – Etapa 1: Ajuste da rede neural usando o conjunto de treinamento.

A Etapa 1 do estudo é o ajuste da RNA no conjunto de treinamento. O retorno dessa etapa é um conjunto com k matrizes com os pesos (chamado de parâmetros na literatura estatística) associados a cada par de neurônios. Dessa maneira, cada matriz é responsável pela propagação dos valores de uma camada para a camada imediatamente seguinte.

O próximo passo é estimar, para cada observação do conjunto de treinamento, o valor dos neurônios nas duas últimas camadas e guardar os resultados junto com o valor observado da variável resposta, o conjunto de dados resultante da etapa 2 é composto por $c + 2$ colunas, onde c é o numero de neurônios na penúltima camada da RNA. Para este estudo selecionou-se sempre a penúltima camada da rede para realizar a recalibração, porém alguma outra camada intermediária poderia ter sido selecionada.

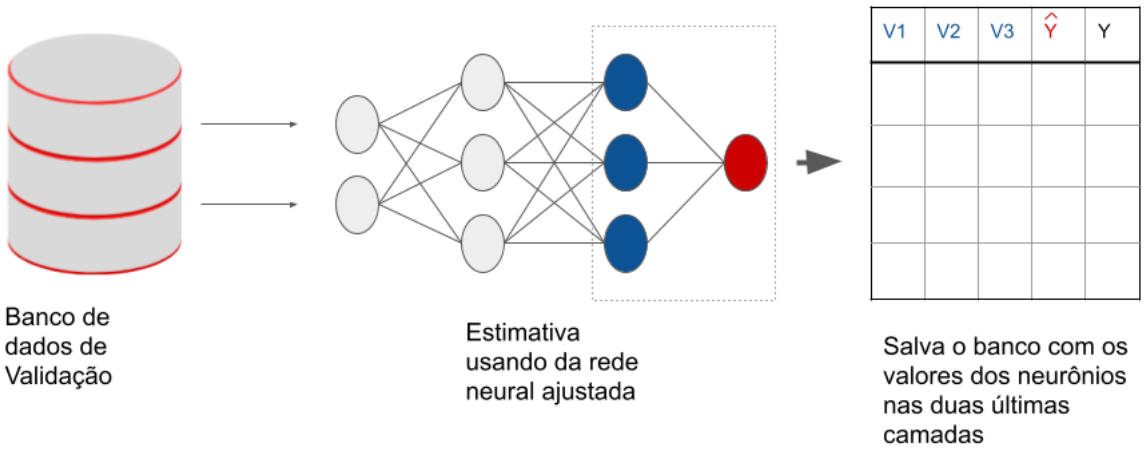


Figura 9 – Etapa 2: Estimação das duas ultimas camadas da RNA para o conjunto de validação

O Banco de dados gerado na Etapa 2 é chamado de banco de recalibração, pois é a partir dele que o método é realizado.

3.2 Recalibração

O procedimento de recalibração utiliza o padrão de comportamento apresentado pelas probabilidades acumuladas de um dado modelo para reconstruir a distribuição preditiva das observações. Para realizar o procedimento é necessário primeiramente assumir uma distribuição de probabilidade para as previsões da RNA, para que através do método seja encontrada uma distribuição preditiva mais fidedigna aos dados. Por exemplo, pode-se assumir que as distribuições preditivas têm a forma da distribuição normal, com média centrada no próprio valor previsto pela rede e variância estimada pelo $\text{MSE} = \frac{1}{n}\sum_{y=1}^{n}(y_i - \hat{y}_i)^2$, obtido do banco de dados separado para validação. A recalibração exige uma reamostragem dos dados do banco de recalibração – no método proposto nesse estudo atribui-se probabilidades maiores de seleção para os vizinhos mais próximos à observação para a qual deseja-se realizar a previsão.

A proporção da vizinhança selecionada para a reamostragem é um aspecto muito importante para a recalibração. Quanto menor o valor selecionado para π, melhor será capturado o viés local do modelo, porém caso a quantidade de vizinhos disponíveis seja muito baixa o modelo irá gerar poucas amostras recalibradas, levando a um alto erro de Monte Carlo. Portanto é necessário fazer um balanço entre o valor de π e o tamanho da amostra, de modo que existam observações suficientes para reamostrar os dados.

Conforme mencionado em 2.2.4, tanto a localização no espaço das covariáveis quanto no das previsões apresentam vantagens e desvantagens. Portanto, uma solução intermediária é verificar a similaridade das observações em um espaço de dimensão reduzida que uma RNA é capaz de gerar.

A Etapa 3 consiste em realizar o procedimento de recalibração, resultando em um vetor contendo uma amostra da distribuição recalibrada do valor previsto para uma nova observação. A partir dessa amostra é possível estimar a densidade da distribuição recalibrada. O valor previsto recalibrado é estimado utilizando a média do vetor amostrado.

O procedimento pode ser resumido na seguinte sequência de passos:

1. Divisão dos dados em 3 conjuntos: treinamento, validação e teste.
2. A rede neural aprende com base no conjunto de treinamento e o banco de validação é usado apenas para checar a sua capacidade de generalização.
3. Usando o banco de validação calcula-se a probabilidade acumulada p_i para cada observação, assumindo que a rede gera previsões com distribuição Normal($\hat{y}$, eqm).
4. Sobre o conjunto de teste faz a propagação feedforward, salvando os valores dos neurônios das duas últimas camadas.
5. Para cada observação de teste, utilizando o espaço da penúltima camada, localiza-se as observações do conjunto de validação mais parecidas e prossegue com o procedimento de reamostragem explicado na seção 2.2.3.

O pseudo-algoritmo do procedimento listado nas seções 3.1 e 3.2 está apresentado no Algoritmo 1.

Algoritmo 1 Ajuste e recalibração da rede neural

1: **Entrada**

- Matriz de covariáveis $\boldsymbol{x}$ e vetor de observações $\boldsymbol{y}$.
- Arquitetura do modelo de rede neural com o número de neurônios por camada $\boldsymbol{c} = (c_1, \ldots, c_k)$ e um vetor especificando as funções de ativação $\mathbf{a} = (\phi_1, \cdots, \phi_k)$.
- Proporção da vizinhança selecionada para a recalibração π.
- Camada na qual será feita a localização, k'.
- Proporção do conjunto de dados separado para treinamento α.

Ajustando a rede neural

2: Defina $n_1 = n\alpha$ e $n_2 = n(1-\alpha)$.
3: Separe a matriz $\boldsymbol{x}$ nos conjuntos de treinamento $\mathbf{t}$, validação $\mathbf{v}$ e teste $\boldsymbol{m}$.
4: Padronize as covariáveis do conjunto de treinamento fazendo $\tilde{t}_{ij} = \frac{t_{ij} - \bar{t}_j}{\hat{s}_{Tj}}$, onde $\bar{t}_j$ e $\hat{s}_{Tj}$ representam, respectivamente, a média e o desvio padrão da covariável j.
5: Padronize as covariáveis do conjunto de validação utilizando as estatísticas do conjunto de treinamento $m_{ij} = \frac{m_{ij} - \bar{t}_j}{\hat{s}_{T_j}}$.
6: Ajuste o modelo de redes neurais com arquitetura $\boldsymbol{c}$ usando os conjuntos $\mathbf{t}$ e $\mathbf{v}$ e calcule a raiz do erro quadrático médio $\hat{\sigma}$.
7: Calcule os valores previstos $\hat{y}_1, \ldots, \hat{y}_{n_2}$ e as respectivas probabilidades acumuladas fazendo $p_i = \Phi_i(y_i | \hat{y}_i, \hat{\sigma})$.
8: Para $i = 1, \ldots, n_2$, calcule o vetor $v_i = (v_{i1}, \ldots, v_{ic_{k'}})$ contendo os valores dos neurônios na camada k' referente à i-ésima observação do conjunto de teste.

Recalibração

9: **para** $i \leftarrow 1$ **até** n_2 **faça**
10: **para** $j \leftarrow 1$ **até** n_2, $i \neq j$ **faça**
11: Calcule a distância euclidiana entre i e j, $d_{ij} = ||v_i - v_j||$.
12: Para cada j satisfazendo $d_{ij} < \tilde{d}$, onde $\tilde{d}$ é o π-quantil do vetor $(d_{i1}, \ldots, d_{in_2})$, defina o peso da observação j usando o *kernel Epanechnikov*

$$w_{ij} = \frac{3}{4} \times \left(1 - \left(\frac{d_{ij}}{\tilde{d}}\right)\right)^2.$$

13: **fim para**
14: Gere um vetor de p-valores $p_i^* = (p_1^*, \ldots, p_m^*)$ amostrando, com reposição, do vetor $p = (p_1, \ldots, p_{n_2})$ utilizando os pesos $(w_{i1}, \ldots, w_{n_2})$.
15: Gere amostras da distribuição preditiva recalibrada $y_{ik}^* = \Phi_i^{-1}(p_k^*, \hat{y}_i, \hat{\sigma})$.
16: **fim para**

17: **Saída** Amostras de Monte Carlo das distribuições preditivas recalibradas.

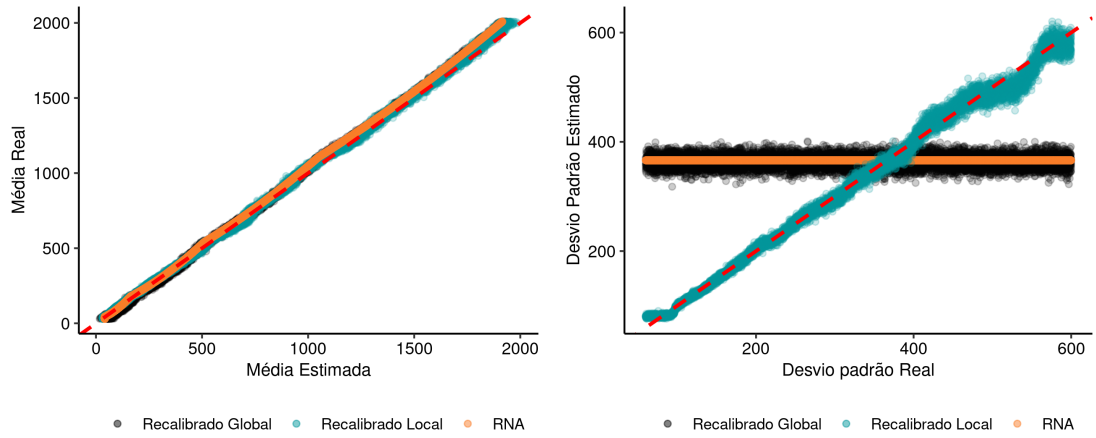
4 Estudo de dados Simulados

Nesse capítulo, apresentamos 3 exemplos simulados para ilustrar o funcionamento do método proposto. No primeiro, revisitamos o exemplo explorado na seção 2.2.2. No segundo, tratamos um problema no qual a variável dependente segue um modelo de mistura de normais. Por fim, abordamos a estimação de um modelo Gama altamente não linear.

4.1 *Toy example* 1 (revisitado)

Considere, novamente, o exemplo discutido na seção 2.2.2. Ajustamos aqui, a titulo de ilustração, um modelo de rede neural com duas camadas intermediárias (escondidas), com seis e com dois neurônios, respectivamente, e função de ativação *relu*. A camada de saída contém um único neurônio e função de ativação *linear*. A função de perda adotada foi o erro quadrático médio. Essa arquitetura foi escolhida arbitrariamente devido a sua estrutura simples e ilustrativa.

Assumimos, por conveniência didática, que tal rede fornece distribuições preditivas Gaussianas, com média centrada no valor previsto pela RNA, e variância igual ao erro quadrático médio do conjunto de treinamento. Em seguida, foram ajustados os dois métodos de recalibração, global e local. Neste último, selecionamos os 10% vizinhos mais próximos, localizando no espaço da penúltima camada da RNA. Utilizando as observações de teste, a Figura 10a, apresenta as médias de cada um dos modelos em contraste com a média verdadeira. A Figura 10b mostra o desvio padrão estimado pelo real, onde é possível observar que o modelo recalibrado localmente é o único capaz de lidar com a heterocedasticidade do modelo, gerando boas estimativas para o desvio padrão.



(a) Estimação da média (b) Estimação do desvio padrão

Figura 10 – Exemplo de recalibração de uma rede neural. A linha pontilhada representa a reta identidade. Em (a) é possível perceber que os três modelos geram boas estimativas para a média real do conjunto de dados simulado. Já em (b) nota-se que apenas o modelo recalibrado localmente foi capaz de lidar com a heterocedasticidade dos dados.

A Tabela 4 compara a performance dos modelos ajustados no conjunto de teste.

Modelo	Erro	Distância KL	Cobertura observada
Linear	14804,5	1,174	93,88
Linear Recalibrado Globalmente	14957,1	1,133	94,94
Linear Recalibrado Localmente	400,8	0,193	94,87
Rede Neural Artificial	178,5	1,186	93,63
RNA Recalibrada Globalmente	307,3	1,035	94,81
RNA Recalibrada Localmente	460,4	0,250	94,77

Tabela 4 – Comparação de performance entre os modelos.

Já havia sido mostrado que o modelo linear passa a produzir melhores estimativas pontuais após a recalibração local. O modelo original de rede neural já possui uma boa acurácia, após a recalibração a rede passou a gerar previsões pontuais menos acuradas. Porém ainda assim o procedimento de recalibração se mostrou bastante eficiente ao reduzir o valor médio da distância de Kullback-Leibler, indicando que a distribuição recalibrada provê uma melhor aproximação que a distribuição normal assumida pela RNA.

A cobertura observada representa o percentual de intervalos de confiança de 95% estimados pelo modelo que contém o valor verdadeiro, observado no conjunto de testes. À primeira vista, todos os intervalos parecem estar capturando bem o valor verdadeiro, porém a Figura 11 aponta que apenas o recalibrado localmente funciona de acordo com o esperado. Por mais que os modelos original e recalibrado global errem apenas 5% dos intervalos, ainda assim eles estão mal especificados, possuindo uma taxa de acerto maior

para baixos valores de x, para os quais a variância é superestimada, enquanto para valores maiores o modelo erra mais de 5% das vezes.

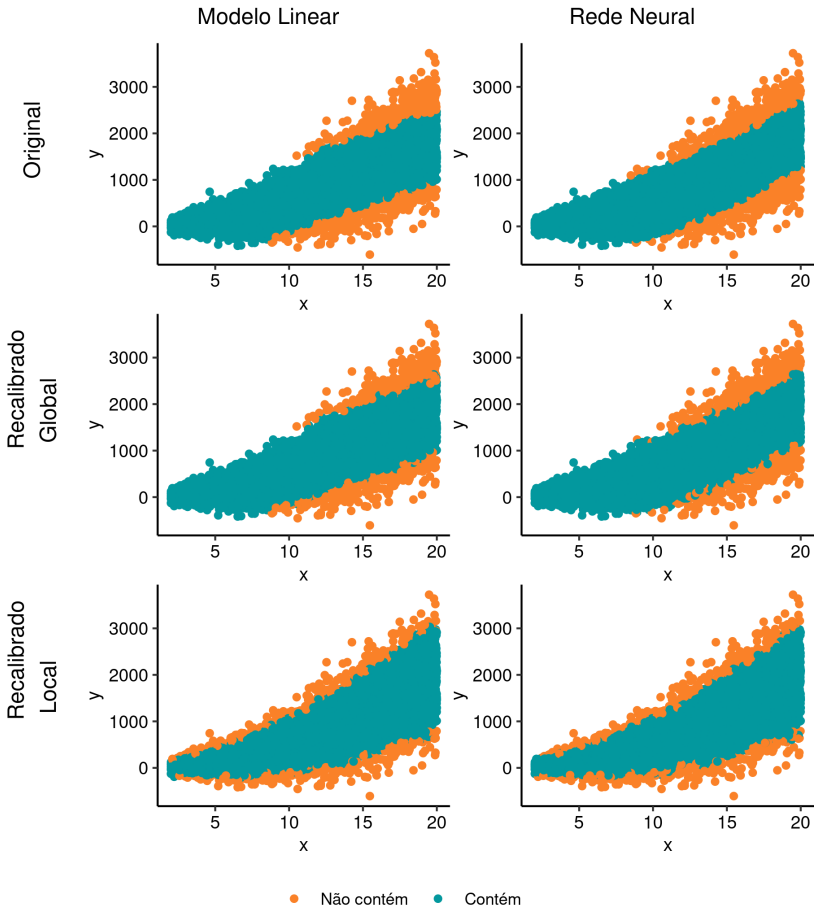


Figura 11 – Região de cobertura dos modelos. A figura indica em verde os pontos que foram capturados pelo intervalo de confiança de cada modelo. Todos os modelos apresentaram 95% de taxa de acerto, porém os modelos descalibrados ou recalibrados globalmente falham em capturar a heterocedasticidade do modelo, acertando mais apenas em regiões onde a variância dos dados está sendo superestimada.

4.2 *Toy example* 2: modelo de mistura de normais

Nesse exemplo, geramos amostra simuladas de uma matriz de 10 covariáveis $\mathbf{X}$ e da variável resposta $\mathbf{y}$. A densidade de probabilidade assumida para realizar a amostragem de $\mathbf{y}$ segue uma mistura de normais, que pode ser escrita como

$$f(x) = \sum_{i=1}^{2} wp_i(x) = \sum_{i=1}^{2} w \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{\frac{-(x-\mu_i)^2}{2\sigma^2}\right\},$$

onde $w = \frac{1}{2}$, $\mu = (g(\mathbf{X}) + 1000, \quad g(\mathbf{X}) + 5000)$ e $\sigma = 1000$.

A função $g(X)$ indica a relação entre as covariáveis e a variável resposta. A relação escolhida foi

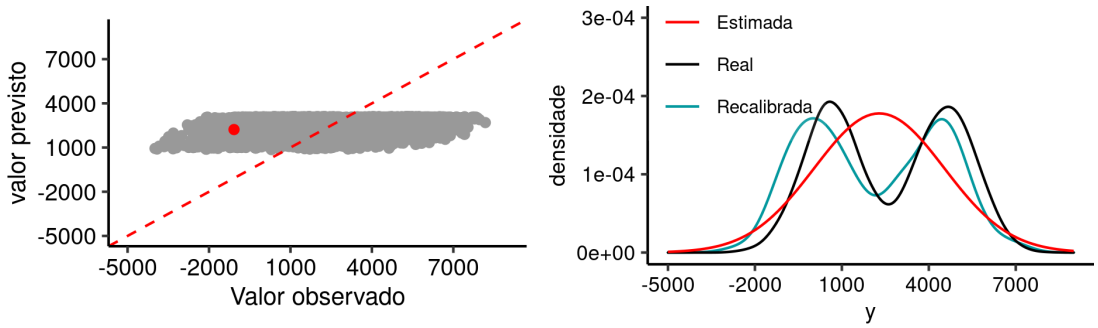
$$g(\mathbf{X}) = \mathbf{D} - \mathbf{X} + \mathbf{B},$$

onde $D_{ij} = X_{ij}^2$, para todo i, j e $\mathbf{B} = [b_1, b_2, ..., b_n]$ é o vetor que contém os vieses.

As covariáveis $\mathbf{X}$ foram geradas a partir de uma normal multivariada. Cada elemento de $\mathbf{B}$ foi gerado de uma normal$(0, 100)$.

Em seguida, uma rede neural *feedfoward* foi ajustada sobre a amostra gerada com objetivo de aprender $g(\mathbf{X})$. A rede treinada possui 5 camadas densas escondidas, as três primeiras contam com com 128, 64, 32 neurônios e função de ativação *Sigmoide* e as duas últimas com 16 e 4 neurônios, com função de ativação linear, além de uma camada de saída com um único neurônio. A função de perda adotada para a otimização do modelo foi o erro quadrático médio, enquanto o otimizador escolhido foi o *Adam* com taxa de aprendizado de 5×10^{-4}.

Como a distribuição $f(x)$ não é conhecida habitualmente, assumimos normalidade com média $\hat{y}$ e variância $\hat{\sigma} = \frac{1}{n}\sum_{y=1}^{n}(y_i - \hat{y}_i)^2$. Posteriormente usamos essa distribuição para realizar a recalibração. Seguindo o procedimento analogamente ao exemplo anterior, realizando a recalibração na penúltima camada da rede, conseguimos estimativas mais consistentes para a estimação da densidade de probabilidade.



(a) Diagrama de dispersão das previsões da rede

(b) Densidade de probabilidade da observação destacada em vermelho

Figura 12 – Exemplo mistura de normais. Em (a) destacamos um pontos selecionados ao acaso. A figura (b) apresenta as curvas de densidade de probabilidade real, estimada e recalibrada localmente. Após a recalibração observa-se uma boa aproximação à distribuição real geradora dos dados.

Esse exemplo serve para ilustrar que a recalibração continua produzindo bons resultados quando usamos um subespaço para realizar a localização dos vizinhos. No exemplo anterior, a recalibração também era feita na penúltima camada da rede, porém esta possuía dimensão maior que o espaço das covariáveis. Na prática, estaremos interessados em localizar usando as camadas da rede principalmente quando possuirmos uma grande

quantidade de covariáveis, para que a partir da projeção do espaço de entrada nas camadas escondidas possamos contornar a maldição da dimensionalidade e calcular distâncias de maneira menos custosa.

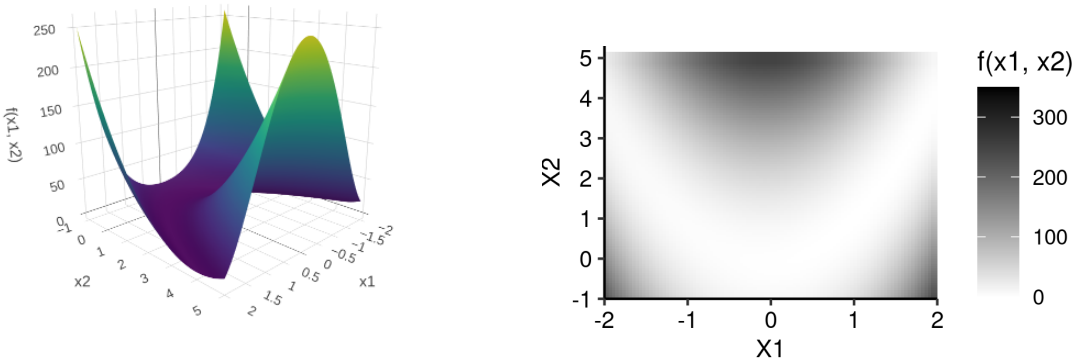
Além disso, a distribuição assumida inicialmente nesse exemplo estava mal especificada. Ainda assim a recalibração foi capaz de gerar amostras de uma distribuição bimodal.

4.3 Exemplo 3: modelo Gama não linear

Um exemplo mais complexo foi desenvolvido para melhor aprofundamento do método proposto. Considere a função de Rosenbrock, também conhecida como vale de Rosenbrock. Essa função se notabiliza por sua larga utilização como teste para algoritmos de otimização. Sua fórmula pode ser expressa da seguinte forma.

$$f(x_1, x_2) = (a - x_1)^2 + b(x_2 - x_1^2)^2,$$

onde assumimos $a = 1$ e $b = 10$. A curva da função nos intervalos $x_1 \in (-2, 2)$ e $x_2 \in (-1, 5)$ pode ser observada na Figura 13.

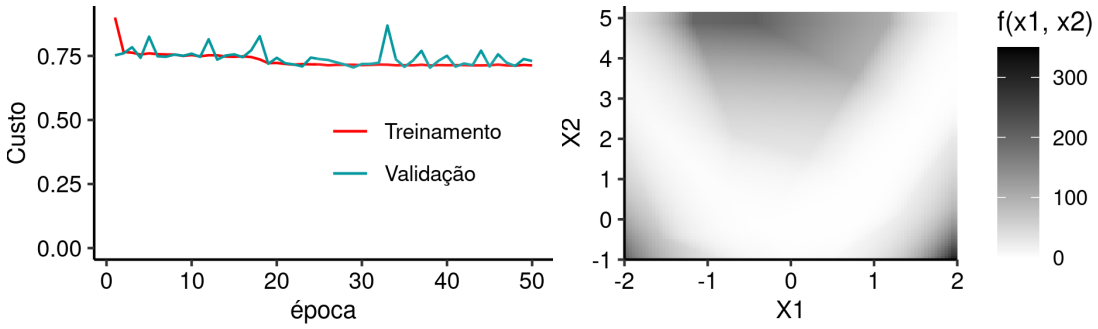


(a) Representação 3d da função de Rosenbrock (b) Mapa de calor da função de Rosenbrock

Figura 13 – Função de Rosenbrock com $a = 1$ e $b = 10$. A função será utilizada como a média condicional de uma distribuição de probabilidade Gama.

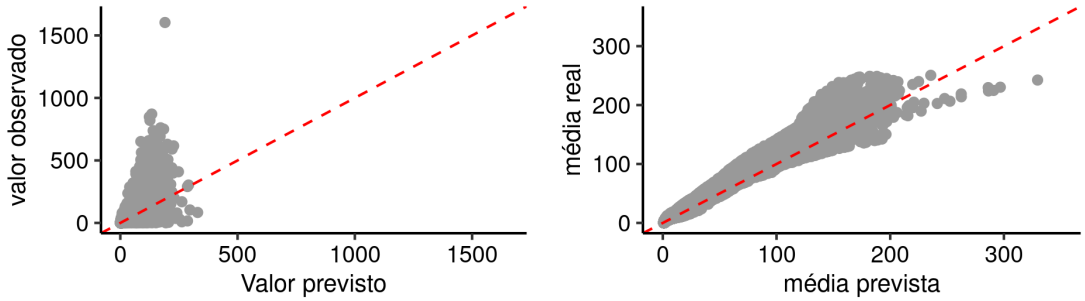
Para simular a relação entre um par de covaráveis e uma variável resposta, foram gerados dois vetores de 120 mil observações, $\mathbf{x}_1$ e $\mathbf{x}_2$, com distribuições $uniforme(-2; 2)$ e $uniforme(-1; 5)$, respectivamente. Em seguida, foram sorteadas 120 mil amostras de uma variável aleatória Y condicionada a x_1, x_2, sendo $Y|x_1, x_2 \sim Gama\left(1, 5; \frac{1,5}{f(x_1,x_2)}\right)$. Portanto para cada par amostrado de x_1, x_2 temos um valor y com distribuição Gama com média igual a $f(x_1, x_2)$.

Após a geração dos dados separou-se a amostra nos conjuntos de treinamento (100 mil observações), validação (10 mil observações) e teste (10 mil observações). Em seguida uma rede neural com arquitetura relativamente simples foi treinada. A arquitetura do modelo proposto possui duas camadas escondidas, a primeira com 10 neurônios e a segunda com 5, ambas com ativação *ReLu*, e uma camada de saída composta de um único neurônio e ativação exponencial. A função de perda escolhida foi o logaritmo do erro quadrático médio e o otimizador foi o *Adam*.



(a) Otimização da Rede

(b) Mapa de calor das superfícies de nível



(c) Previsão pelo valor observado da RNA

(d) Previsão pela média real da RNA

Figura 14 – Ajuste da Rede Neural. Resultados do conjunto de validação. Em (a) temos o otimização da rede ao longo das iterações. Em (b) temos uma representação da superfície gerada pela rede em função das covariáveis de entrada. A figura (c) ilustra o valor observado pelo valor estimado, enquanto a figura (d) aponta a média estimada pela rede em função da média real, dada pela função de Rosenbrock.

A arquitetura simples da rede não garante flexibilidade suficiente para o aprendizado da média real geradora do conjunto de dados. Com a finalidade de obter um melhor desempenho do modelo, será realizado o procedimento de recalibração local, usando o espaço de entrada para localizar os vizinhos mais próximos.

Como a rede neural não é um modelo probabilístico, e sim uma função que aproxima a média, é necessário assumirmos uma distribuição de probabilidade inicial que será recalibrada. Essa distribuição pode ser encontrada utilizando um modelo linear generalizado (MLG). Os modelos lineares generalizados propostos por Nelder e Wedderburn

(1972) flexibilizam a modelagem da variável resposta utilizando qualquer distribuição da família exponencial. Portanto, ajustou-se um MLG utilizando o valor real de $\mathbf{Y}$ como variável resposta e o valor estimado pela rede $\hat{y}$ como variável explicativa, assumindo que $\mathbf{Y}|log(\hat{y}) \sim exp(\lambda)$. O modelo foi ajustado utilizando a função *glm* do pacote *stats* do *R*, o resultado dos parâmetros ajustados está indicado na Tabela 5. Vale destacar que a distribuição assumida para $\mathbf{Y}$ foi propositalmente mal especificada para verificar a capacidade da recalibração uma vez que em problemas reais a distribuição verdadeira dos dados é desconhecida.

	Estimado	Erro padrão	valor t	P-valor
Intercepto	0,264	0,02	12,98	<0,001
$\log(\hat{Y})$	0,984	0,006	158,20	<0,001

Tabela 5 – Resultado dos Parâmetros estimados pelo MLG ajustado de $Y/\hat{Y}$ com os dados de validação.

Dessa maneira, para cada valor gerado pela rede neural temos uma distribuição de probabilidade $exponencial(\lambda)$ associada, onde o parâmetro λ é estimado utilizando o modelo linear generalizado. O fluxo para a estimação de $\hat{\lambda}$ está ilustrado na Figura 15.

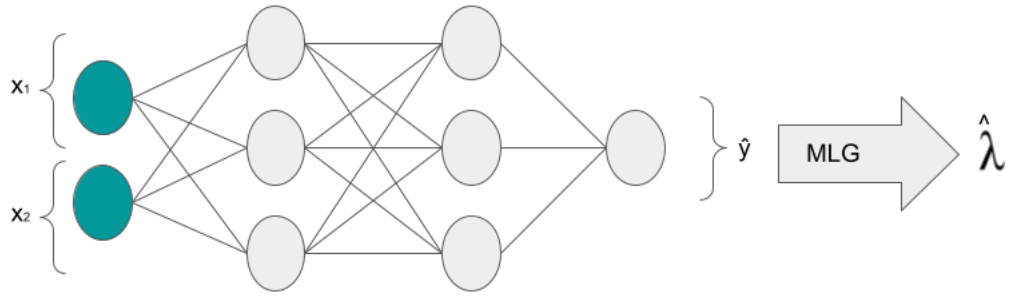


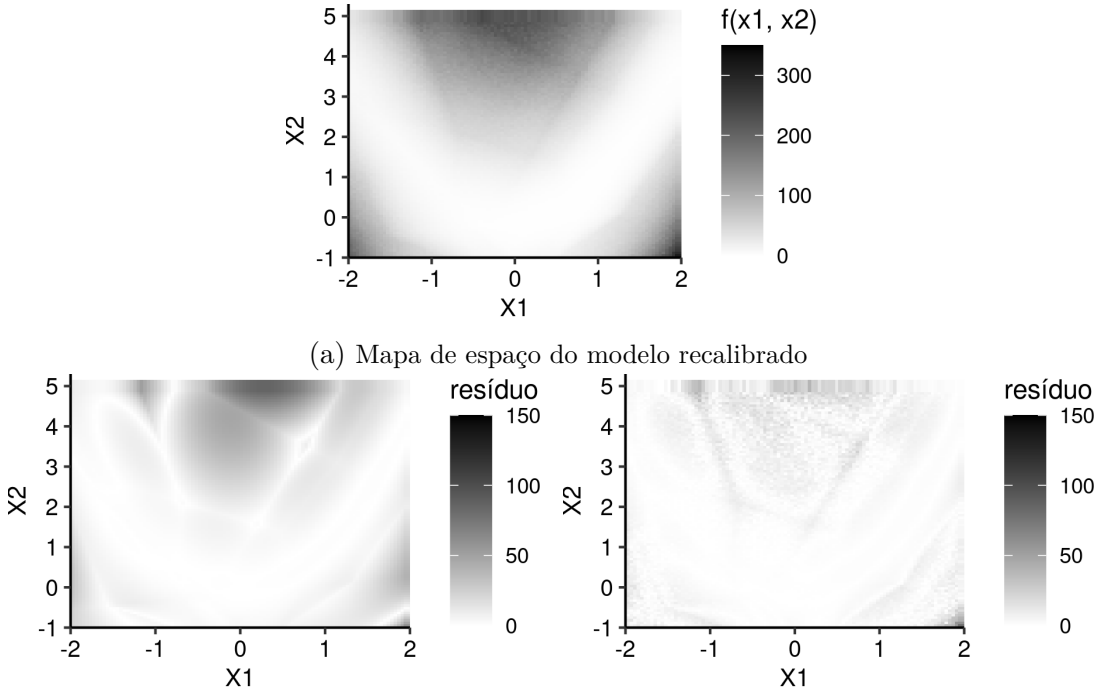
Figura 15 – Procedimento para a obtenção da distribuição inicial que será recalibrada

Agora o cenário já está montado para que o procedimento de recalibração. O banco usado para a recalibração será o mesmo utilizado para validação da rede neural e ajuste do *MLG*. Faremos a recalibração dos dados de teste seguindo os seguintes passos, para cada observação desse conjunto:

1. Realiza a previsão da média usando os pesos ajustados da RNA;
2. Encontra o $\hat{\lambda}$ com o *MLG* assumindo distribuição exponencial.
3. Para cada observação da matriz de delineamento $\boldsymbol{x}$ do conjunto de teste, localiza as observações do conjunto de validação com distância euclidiana de no máximo 0,71.

4. Realiza, para os vizinhos selecionados, o procedimento de recalibração definido na seção 2.2.3, com o tamanho de reamostragem igual a 1000.

Até agora, os vizinhos sempre foram escolhidos usando uma proporção de vizinhança. Aqui o método de seleção das observações mais próximas foi baseada na distância máxima entre as observações do conjunto de validação para cada observação do conjunto de teste. Essa abordagem foi escolhida por conta das fronteiras impostas aos dados, observações que se encontram nas bordas possuem um raio de vizinhança maior que o raio de observações centrais. A Figura 16 expõe, no painel (a), a superfície geradas pelo modelo recalibrado. Para facilitar a visualização e comparar as curvas geradas pelo modelo com e sem recalibração, os painéis (b) e (c) mostram os resíduos dos modelos com e sem recalibração. É possível perceber que houve uma diminuição no valor absoluto dos resíduos em praticamente todas as regiões do espaço.



(a) Mapa de espaço do modelo recalibrado

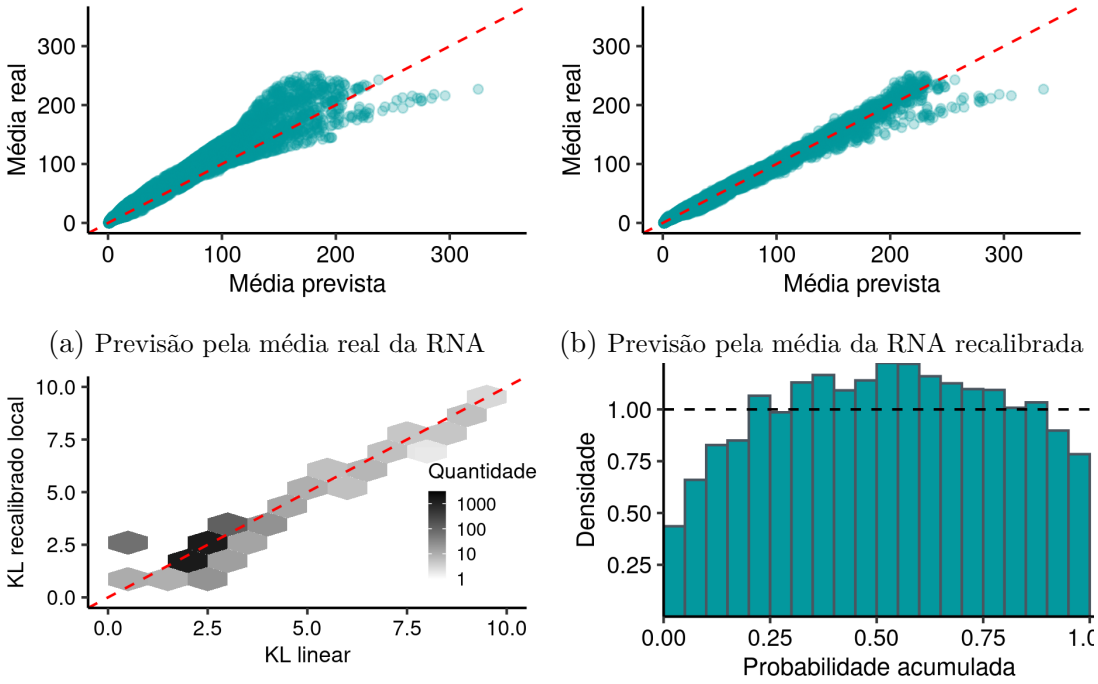
(b) Mapa dos resíduos sem recalibração

(c) Mapa dos resíduos recalibrados

Figura 16 – Modelo Recalibrado. Em (a) está apresentado o mapa de calor do espaço gerado pelo modelo após a recalibração. Em (b) identifica-se a diferença entre a média estimada pela rede neural e a média real (função de Rosenbrock). O painel (c) exibe a diferença entre a média estimada pelo modelo recalibrado e a média real. Pode-se observar que a média real do modelo recalibrado é melhor em todos os lugares do espaço

A capacidade de generalização do modelo recalibrado pode ser verificada usando as 10 mil amostras separadas para teste. Comparando os painéis (a) e (b) da Figura 17, nota-se que após a recalibração o modelo apresentou um desempenho de previsão melhor. O painel (c) contrasta a divergência de Kullback-Leibler para cada observação do conjunto

de teste antes e depois da recalibração. O quadro (d) apresenta o histograma das probabilidades acumuladas antes da recalibração. É possível identificar que o modelo original superestima a variância. De acordo com a Tabela 6 houve uma melhora na divergência média que caiu de 1,88 para 1,03 após o procedimento de recalibração. A mesma tabela também aponta para uma melhora significativa na acurácia do modelo e indica uma leve redução da proporção de observações não contidas no intervalo de confiança gerado por cada um dos dois modelos.



(a) Previsão pela média real da RNA

(b) Previsão pela média da RNA recalibrada

(c) KL no modelo original e recalibrado

(d) Histograma das probabilidades acumuladas

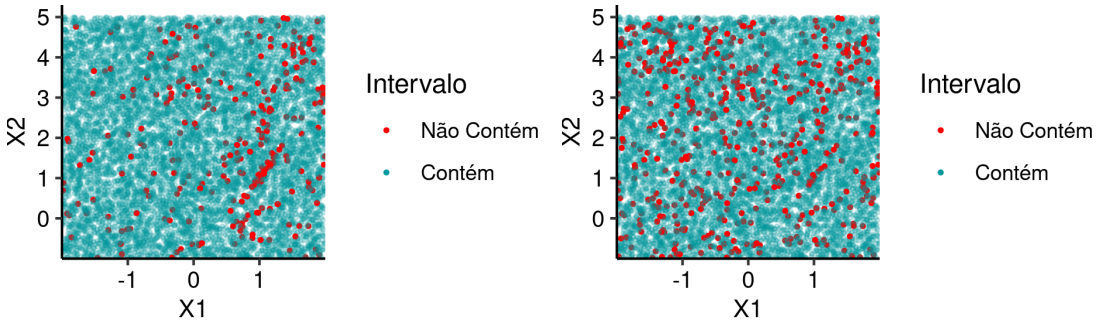
Figura 17 – Avaliação do desempenho do modelo recalibrado no conjunto de testes. Em (a) nota-se que o modelo não se comporta muito bem para algumas regiões de previsão. O ajuste em (b) funciona de maneira mais satisfatória, possuindo muitos pontos sobre a reta identidade, que identifica previsões próximas a valores reais. A figura (c) mostra que houve uma melhora na previsão da distribuição dos dados em algumas regiões do espaço, porém a distribuição prevista piorou em outros lugares. A figura (d) mostra o histograma das probabilidades acumuladas. Nota-se uma superestimação na variância do modelo descalibrado.

Medida	RNA	RNA recalibrada
Acurácia (EQM)	250,99	43,86
Kullback-Leibler médio	1,92	1,105
Cobertura	97,16%	94,91%

Tabela 6 – Comparação entre os modelos utilizando os dados de teste

A Tabela 6 confirma que a recalibração foi bem sucedida, levando a estimativas mais acuradas e distribuições mais próximas à distribuição real dos dados. O histograma

apresentado em 17d indica que o modelo neural superestima a variância real. Isso se reflete no erro quadrático médio da rede e, consequentemente, faz com que os intervalos de confiança de 95% capture mais de 95% das observações.



(a) (b)

Figura 18 – Região de cobertura dos modelos. Os pontos em vermelho representam os valores observados de y do conjunto de testes que não foram capturados pelo intervalo de confiança do modelo de 95%. O modelo original apresenta uma percentual de cobertura maior que o modelo recalibrado, porém isso só ocorre devido a uma má especificação, que superestima a variância das previsões.

5 Conclusão e Trabalhos Futuros

Existem na literatura algumas técnicas para lidar com a limitação da falta da quantificação da incerteza associada as previsões das redes neurais. Este trabalho introduz um novo método de recalibração em RNA, o qual se mostrou eficiente para a estimar os intervalos de predição.

Foram apresentados duas versões de recalibração, global e local, sendo a segunda em geral mais efetiva. Foi nesse caso que a divergência de Kullback–Leibler entre a distribuição de probabilidade das previsões e a distribuição real apresentou os menores valores em todos os cenários propostos. Isso indica que após o procedimento há uma melhor aproximação do processo gerador de dados. Além disso, avaliou-se a proporção de intervalos de predição estimados que contém o valor observado para verificar se os intervalos gerados cumprem as propriedades básicas de cobertura. Outro ponto avaliado foi a acurácia do modelo recalibrado. A recalibração local levou à previsões pontuais melhores em redes neurais com arquiteturas não muito complexas.

Em trabalhos futuros será necessário avaliar o efeito da recalibração em modelos mais complexos e computacionalmente mais custosos, assim como verificar se há ganho computacional ao ajustar a recalibração em uma rede simples como alternativa aos modelos de redes mais profundas. Outro ponto bastante interessante para ser estudado é a qualidade da recalibração em cada camada da rede, tal qual o estudo das arquiteturas que levam o conjunto de covariáveis à representações mais informativas nas camadas escondidas, potencializando a obtenção de melhores resultados pela recalibração. Por fim, a própria localização dos vizinhos também pode ser otimizada usando algoritmos mais eficientes para o cálculo das distâncias entre os pontos.

Referências

BEYER, K. et al. When is “nearest neighbor” meaningful. International Conference on Database Theory, 1999. Citado na página 19.

CARVALHO, D. da S. *Determinação do sexo a partir da densidade de cristas papilares e da contagem de albodatilares em brasileiros*. 146 p. Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2019. Citado na página 11.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. Math. Control Signal Systems 2, 1989. Citado na página 10.

FACELI, K. et al. *Inteligencia Artificial*: Uma abordagem de aprendizado de máquina. 1. ed. [S.l.]: LTC, 2011. 378 p. Citado na página 8.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 2 vezes nas páginas 9 e 11.

HAYKIN, S. *Neural Networks and Learning Machines*. 3. ed. McMaster University, Canada: Prentice Hall, 2009. 906 p. Citado 2 vezes nas páginas 7 e 8.

MCCARTHY, J. What is artificial intelligence. Stanford University, 1998. Citado na página 7.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. University Of Illinois, 1943. Citado na página 7.

NELDER, J. A.; WEDDERBURN, R. W. M. Generalized linear models. Journal of the Royal Statistical Society. Series A (General), Vol. 135, No. 3 (1972), pp.370-384, 1972. Citado na página 32.

NETER, J.; WASSERMAN, W.; KUTNER, M. H. *Applied linear regression models*. 2. ed. [S.l.]: R.D. Irwin, 1983. 547 p. Citado na página 11.

RIZZO, M. L. *Statistical Computing with R*. Bowling Green State University Bowling Green, Ohio, U.S.A: Chapman Hall/CRC, 2008. 393 p. Citado na página 12.

ROSENBLATT, F. *The Perceptron*: A perceiving and recognizing automaton. [S.l.], 1957. Citado na página 7.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. Nature, 1986. Citado na página 9.