Aprendizado Incremental de Redes RBF via Agrupamento Evolutivo de Fluxos de Dados

Thais Macela de Lira Menegaldi*†, Rodrigo Amador Coelho†‡, Cristiano Leite de Castro†‡
*Graduanda em Engenharia de Sistemas, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil
†Machine Intelligence and Data Science Laboratory, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil
‡Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil
§Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil
*†thaismenegaldi@ufmg.br, †‡toluenotnt@gmail.com, †§crislcastro@ufmg.br

***Resumo*—Em virtude da era do *Big Data* e dos dispositivos IoT (*Internet of Things*), onde diversas aplicações presentes no cotidiano das pessoas geram dados a todo momento, surgiu uma demanda por modelos de aprendizado de máquina que são capazes de operar eficientemente em fluxos de dados contínuos. Na literatura há inúmeros algoritmos propostos para esse cenário, entretanto, em sua maioria, tratam-se de modelos de elevada complexidade e que tipicamente requerem o ajuste de diversos hiperparâmetros. O presente trabalho propõe uma maneira simples e eficaz de se definir a topologia da camada escondida da rede neural de base radial, dando a ela a capacidade de aprender incrementalmente. Esta abordagem baseia-se no algoritmo de clusterização evolutiva MicroTeda, tornando possível a atualização da arquitetura da rede à medida que novas amostras de dados são recebidas. Resultados preliminares obtidos sobre bases de dados sintéticas e reais demonstram que a abordagem proposta é promissora mesmo diante de mudanças de conceito (*concept drifts*) abruptas e graduais.**

***Palavras-chave*—Agrupamento evolutivo, aprendizado incremental, mudança de conceito, rede neural de função de base radial.**

I. Introdução

Um dos desafios existentes ao se trabalhar com Redes Neurais de Função de Base Radial (Redes RBFs) [1] se dá na definição de sua topologia. Esse processo, geralmente, ocorre no estágio inicial da construção do modelo de aprendizado. Mesmo diante de um cenário no qual o conjunto de dados é estático, isto é, em que a rede recebe todos as observações de uma só vez, a determinação da arquitetura da rede RBF não se trata de uma tarefa simples. Há uma variedade de hiperparâmetros que precisam ser especificados, e quando isso ocorre inadequadamente, a generalização do modelo pode ser comprometida.

No que se refere ao cenário de fluxo de dados contínuos, definir a topologia da rede é um problema ainda mais complexo, uma vez que os dados são recebidos ao longo do tempo e estão susceptíveis à mudanças em sua distribuição. A rede neural, no entanto, deve ser capaz de adaptar a sua estrutura para cada novo cenário imposto, de modo que seu desempenho seja consistente.

Dado o crescimento na produção de dados nos últimos anos (*Big Data*), o aprendizado *online* ou incremental, como é conhecido na literatura, ganhou bastante atenção da comunidade científica e da indústria. Assim, uma série de abordagens surgiram para lidar com os desafios de se receber dados gradualmente [2] [3]. A *Online Evolving Spiking Neural Network* (OeSNN) [4], por exemplo, consiste em uma rede neural de três camadas, onde a camada de saída é evolutiva e se adapta à medida que uma nova amostra de dados é recebida. Este método codifica os dados de entrada em uma sequência de *spikes* que são ligados à camada escondida, onde seus neurônios têm o formato de uma função Gaussiana. O número de neurônios, no entanto, é recebido como parâmetro e sua escolha impacta no desempenho da rede. Há também o método RIT2-TSK-FNN, que propõe uma nova estrutura para um controlador de uma rede neural recorrente nebulosa [5]. Tal método utiliza o aprendizado por reforço para adaptação online dos hiperparâmetros do controlador, melhorando o desempenho de redes não lineares em fluxos contínuos de dados.

Nesse mesmo contexto, foi proposto o método *Adaptive Random Forest* (ARF) [6], que se trata de uma adaptação do algoritmo *Random Forest* [7] para o aprendizado incremental. O ARF possui um mecanismo de reamostragem e operadores adaptativos que são capazes de lidar com mudanças na distribuição dos dados. De maneira similar, uma variação incremental da rede neural recorrente *Long Short-Term Memory* (LSTM) [8] é proposta, tal que seu processo de adaptação ao fluxo de dados contínuos não requer um detector de mudança de conceito ou um sistema de gerenciamento de memória.

Embora os algoritmos supracitados apresentem resultados satisfatórios quando aplicados ao contexto de aprendizado *online*, muitos deles são bastante complexos e requerem o ajuste de diversos hiperparâmetros para cada problema. Nesse sentido, o presente trabalho propõe uma maneira de determinar a arquitetura da camada escondida da rede RBF com um algoritmo de agrupamento evolutivo que requer apenas dois hiperparâmetros, facilitando seu uso em casos em que não há um conhecimento prévio do domínio do problema. A adaptação contínua das funções de densidade à medida que os dados chegam (via agrupamento evolutivo), provê à rede RBF a capacidade de aprender incrementalmente, podendo, assim, lidar com problemas comuns encontrados em cenários dinâmicos, tais como a adaptabilidade à mudanças nas distribuições e a escassez de recursos para armazenamento de grandes volumes de dados.

II. Método Proposto

O método proposto, intitulado IRBF (*Incremental Radial Basis Function Network*), consiste em três principais etapas sequenciais, conforme ilustrado pelas Figuras 1, 2 e 3: (i) construção iterativa dos micro-grupos à medida que os dados são disponibilizados, (ii) obtenção dos macro-grupos e (iii) definição da camada escondida da rede RBF e ajuste dos pesos (e bias) da camada de saída. Na primeira etapa, as observações são agrupadas de acordo com o grau de similaridade existente entre elas. Tais conjuntos de observações são denominados micro-grupos. Como uma observação pode pertencer a mais de um micro-grupo simultaneamente, a próxima etapa consiste na integração dos micro-grupos que compartilham uma ou mais observações, isto é, apresentam sobreposição. Ademais, micro-grupos que não apresentam tanta relevância para o agrupamento são desativados. A integração desses micro-grupos, intitulados macro-grupos, são utilizados para estimar os principais hiperparâmetros da camada escondida da rede RBF, como o número de neurônios e os centros e raios de cada um deles. Por último, os pesos da camada de saída são ajustados usando a pseudo-inversa da matriz de ativações da camada escondida. A partir do segundo *batch* de dados a rede realiza a classificação das observações contidas nele e, em seguida, repete todas as etapas descritas anteriormente, a fim de manter a estrutura de agrupamento e a topologia da rede atualizadas. Essas três etapas sequenciais da abordagem proposta são ilustradas a seguir. A Figura 1 apresenta os micro-grupos obtidos pelo algoritmo TEDA. Em seguida, na Figura 2 tem-se a mistura de densidades resultante deste agrupamento. Por fim, a Figura 3 exemplifica a topologia da rede RBF formada com base nos estágios anteriores.

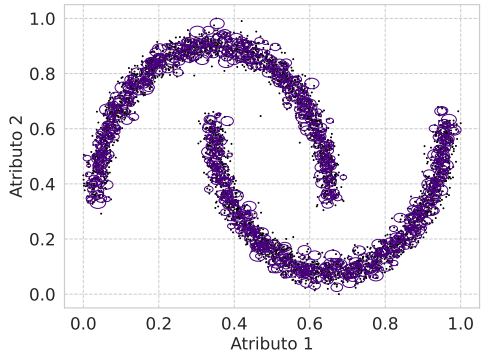

Figura 1. (i) Micro-grupos obtidos pelo algoritmo TEDA são representados pelos círculos que contêm uma ou mais observações do conjunto de dados.

A. TEDA

TEDA (*Typicality and Eccentricity Data Analytics*), proposto por Angelov et al. [9], é um algoritmo de detecção de anomalias em fluxos de dados contínuos. Trata-se de uma metodologia não-paramétrica de estimação de densidades que se baseia na proximidade de cada nova observação em relação a todos os outros pontos que compõem o conjunto de dados.

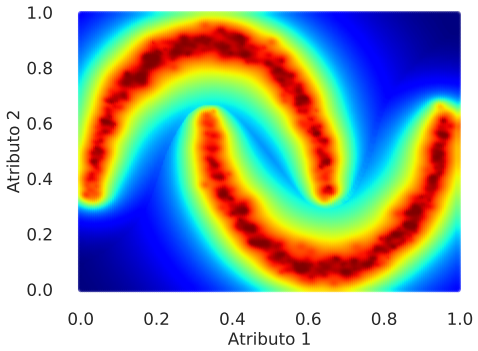

Figura 2. (ii) Mistura de densidades resultante.

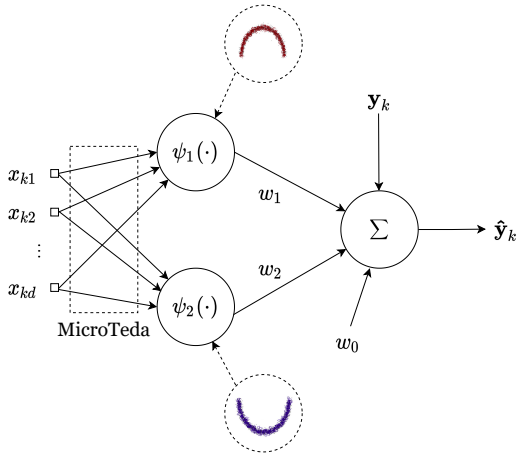

Figura 3. (iii) Topologia da rede RBF formada com base nas etapas (i) e (ii).

Nos casos em que a distância é superior a um valor limite, de acordo com a desigualdade de Chebyshev [10], considera-se que o ponto observado é um *outlier*. Para caracterizar cada observação do conjunto de dados o TEDA utiliza os conceitos de proximidade acumulada, tipicidade e excentricidade.

1) Proximidade acumulada: Seja $\mathbf{x}_k \in \mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ um vetor de entrada com d dimensões no instante de tempo k proveniente de um *batch* com n observações, sua proximidade acumulada π_k é dada por [9]

$$\pi_k(\mathbf{x}_k) = \sum_{i=1}^{k} dist(\mathbf{x}_k, \mathbf{x}_i), \tag{1}$$

em que $dist(\mathbf{x}_k, \mathbf{x}_i)$ é a distância entre os pontos $\mathbf{x}_k$ e $\mathbf{x}_i$. Essa distância pode ser Euclidiana, Mahalanobis, Cosseno, entre outras.

2) Excentricidade: Uma vez calculada a proximidade acumulada, torna-se possível calcular a excentricidade, que se trata de uma medida de dissimilaridade de $\mathbf{x_k}$ em relação aos dados disponíveis até o instante de tempo k. Em outras palavras, a excentricidade mensura o quão excêntrico um vetor de entrada $\mathbf{x_k}$ é em relação ao conjunto de dados. Esta medida

é dada por [9]

$$\varepsilon_k(\mathbf{x}_k) = \frac{2\pi_k(\mathbf{x}_k)}{\sum_{i=1}^{k}\pi_k(\mathbf{x}_i)},\ \sum_{i=1}^{k}\pi_k(\mathbf{x}_i) > 0,\ k \geq 2. \quad (2)$$

A excentricidade também pode ser calculada de maneira recursiva [11]

$$\varepsilon(\mathbf{x}_k) = \frac{1}{k} + \frac{(\mu_k - \mathbf{x}_k)^T(\mu_k - \mathbf{x}_k)}{k\sigma_k^2}, \quad (3)$$

em que μ_k representa a média e σ_k^2 a variância dos dados até o instante de tempo k. Tanto a média quanto a variância também podem ser calculadas recursivamente [12]

$$\mu_k = \frac{k-1}{k}\mu_{k-1} + \frac{\mathbf{x}_k}{k},\ k \geq 1,\ \mu_1 = \mathbf{x}_1. \quad (4)$$

$$\sigma_k^2 = \sigma_{k-1}^2 + \frac{1}{k-1}||\mathbf{x}_k - \mu_k||^2,\ \sigma_1^2 = 0. \quad (5)$$

Por fim, vale ressaltar que os valores de excentricidade variam no intervalo de $[0, 1]$ e que seu valor normalizado é dado por

$$\zeta(\mathbf{x}_k) = \frac{\varepsilon(\mathbf{x}_k)}{2} \quad (6)$$

3) Tipicidade: além da excentricidade, um outro conceito importante utilizado pelo TEDA é a tipicidade. Tal medida consiste na similaridade de $\mathbf{x}_k$ em relação aos demais dados, isto é, mensura o quão típica uma observação é em relação ao conjunto de dados. Além disso, a tipicidade é uma medida complementar à excentricidade e, portanto, é definida como

$$\tau(\mathbf{x}_k) = 1 - \varepsilon(\mathbf{x}_k),\ k \geq 2 \quad (7)$$

Assim como a excentricidade, a tipicidade também tem seus valores contidos no intervalo $[0, 1]$ e a tipicidade normalizada é definida por

$$t(\mathbf{x}_k) = \frac{\tau(\mathbf{x}_k)}{k-2} \quad (8)$$

4) Limiar para detecção de outliers: A partir da excentricidade normalizada $\zeta(\mathbf{x}_k)$ é possível definir um limiar para a detecção de *outliers* tomando como base a desigualdade de Chebyshev [10]

$$\zeta(\mathbf{x}_k) > \frac{m^2+1}{2k},\ m > 0 \quad (9)$$

em que m representa o número de desvios padrão distantes da média que uma amostra deve estar para ser considerada um *outlier*. Tendo em vista que o tamanho da amostra influencia no rigor do limiar para definição de *outliers*, Neto et al. [13] propuseram um critério de detecção de *outlier* m que varia em função de k. Desse modo, tem-se uma função $m(k)$ que permite que a definição da quantidade de desvios padrão necessária para que uma amostra seja considerada *outlier* seja ajustada de maneira dinâmica ao longo do tempo. A função $m(k)$ se assemelha a uma sigmoide, conforme

$$m(k) = \frac{3}{1 + e^{-0.007(k-100)}} \quad (10)$$

B. Micro-grupos

Micro-grupos tratam-se de estruturas de agrupamento formadas com base nas características comuns entre observações já recebidas de um fluxo de dados. Esses micro-grupos possuem um mecanismo de detecção automática de *outliers* feita através da função $m(k)$.

A cada novo ponto do conjunto de dados que chega, é verificada a condição de *outlier* dessa observação em relação a todos os micro-grupos já existentes. Similarmente ao conceito de conjuntos nebulosos, uma observação pode pertencer simultaneamente a mais de um micro-grupo tendo diferentes graus de pertinência para cada um.

Seja $\mathbf{m}_i$ o i-ésimo micro-grupo em um determinado instante, cada micro-grupo é definido por um conjunto de parâmetros que são atualizados sempre que uma nova observação é obtida. Tais parâmetros são [13]:

- S_k^i: número de amostras;
- $m_k^i(S_k^i)$: parâmetro de *outlier*;
- μ_k^i: média (ou centro);
- $(\sigma_k^i)^2$: variância;
- $\varepsilon^i(\mathbf{x}_k)$: excentricidade;
- $\zeta^i(\mathbf{x}_k)$: excentricidade normalizada;
- $\tau^i(\mathbf{x}_k)$: tipicidade;
- $t^i(\mathbf{x}_k)$: tipicidade normalizada;
- $D_k^i = \frac{1}{\zeta^i(\mathbf{x}_k)}$: densidade do micro-grupo;
- ρ_k^i: energia, a ser definida adiante.

No momento em que se recebe a primeira observação, é criado o primeiro micro-grupo $\mathbf{m}_1$ com os seguintes parâmetros:

$$q = 1,\ S_1^1 = 1,\ \mu_1^1 = \mathbf{x}_1,\ (\sigma_1^1)^2 = 0$$

em que q representa a quantidade de micro-grupos. Vale ressaltar que apenas alguns parâmetros são calculados quando se tem um único ponto ($S_i^k = 1$), uma vez que a excentricidade e a tipicidade só podem ser calculadas quando há, no mínimo, 2 observações. Quando um novo ponto $\mathbf{x}_k$ chega num instante de tempo $k > 1$, calcula-se a tipicidade e a excentricidade de $\mathbf{x}_k$ em relação a todos os micro-grupos existentes. Além disso, a condição de *outlier* é verificada de acordo com

$$\zeta^i(\mathbf{x}_k) > \frac{m_k^i(S_k^i)^2 + 1}{2S_k^i} \quad (11)$$

$$m_k^i(S_k^i) = \frac{3}{1 + e^{-0.007(S_k^i - 100)}} \quad (12)$$

Ao analisarmos a Equação 11, quando $m_k^i(S_k^i) \geq 1$ nota-se que o segundo ponto pertencente a qualquer micro-grupo $\mathbf{m}_i$ nunca será considerado um *outlier*, mesmo que este seja muito distante da primeira observação de $\mathbf{m}_i$. Isto é indesejável, visto que desse modo podem ser criados micro-grupos muito grandes, englobando observações que não são tão similares entre si. Por esse motivo, foi adicionado o parâmetro r_0 com o intuito de limitar a variância dos micro-grupos quando $S_k^i = 2$. Desse modo, é possível evitar que um micro-grupo cresça indefinidamente ao adicionar o termo referente à r_0 ao teste

de *outlier* quando se tem apenas duas observações no micro-grupo, i.e, $S_k^i = 2$ [13]

$$\left(\zeta_2^i(\mathbf{x}_2) > \frac{(m^i(2))^2 + 1}{4}\right) AND \left((\sigma_2^i)^2 < r_0\right) \tag{13}$$

Após a verificação da condição de *outlier* para uma dada observação $\mathbf{x}_k$ em relação aos micro-grupos existentes, tem-se duas condições possíveis. A primeira delas é que $\mathbf{x}_k$ não é *outlier* para, ao menos, um micro-grupo. Nesse caso, todos os micro-grupos aos quais $\mathbf{x}_k$ pertence, devem ter seus parâmetros atualizados

$$\begin{aligned}
S_k^i &= S_{k-1}^i + 1 \\
\mu_k^i &= \frac{S_k^i - 1}{S_k^i}\mu_{S_k^i - 1} + \frac{\mathbf{x}_k}{S_k^i} \\
(\sigma_k^i)^2 &= \frac{S_k^i - 1}{S_k^i}(\sigma_{k-1}^i)^2 + \frac{1}{S_k^i - 1}||\mathbf{x}_k - \mu_k^i||^2 \\
\varepsilon^i(\mathbf{x}_k) &= \frac{1}{S_k^i} + \frac{(\mu_k^i - \mathbf{x}_k)^T(\mu_k^i - \mathbf{x}_k)}{S_k^i(\sigma_k^i)^2} \\
\rho_k^i &= 1
\end{aligned} \tag{14}$$

em que a energia ρ representa o tempo desde a última atualização do micro-grupo [14], isto é, o tempo decorrido desde que uma observação foi atribuída ao micro-grupo pela última vez. Nos instantes de tempo em que nenhuma observação foi considerada como pertencente a um micro-grupo $\mathbf{m_i}$, a sua energia ρ é atualizada de acordo com a função de decaimento a seguir

$$\rho_k^i = \rho_{k-1}^i - \frac{1}{\lambda} \tag{15}$$

em que $\frac{1}{\lambda}$ é denominado fator de esquecimento e λ é um hiperparâmetro do algoritmo. Quando a energia ρ é menor ou igual a zero, o micro-grupo é desativado. Isso ocorre, pois significa que há muito tempo nenhuma observação é atribuída a ele e, portanto, tem pouca relevância na distribuição do conjunto de dados.

A segunda condição, no caso em que $\mathbf{x}_k$ é *outlier* para todos os micro-grupos, cria-se então um novo micro-grupo

$$q = q + 1;\ S_k^{new} = 1;\ \mu_k^{new} = \mathbf{x}_k;\ (\sigma_k^{new})^2 = 0;\ \rho_k^{new} = 1 \tag{16}$$

C. Macro-grupos

Dado que se tem todos os micro-grupos atualizados no instante de tempo k, inicia-se a etapa de obtenção dos macro-grupos, que consistem em um conjunto de micro-grupos que se sobrepõem. A atualização dos macro-grupos pode ser feita de maneira *offline*, possibilitando a obtenção do agrupamento resultante de um fluxo de dados sempre que se desejar. Os macro-grupos são formados a partir de um grafo em que os micro-grupos representam os vértices e as arestas, as interseções existentes entre eles [14]. Feito isso, são extraídos desse grafo seus componentes fortemente conectados [15], isto é, os micro-grupos que estão conectados entre si. A verificação da interseção entre dois micro-grupos é feita por

$$dist(\mu_k^i, \mu_k^j) < 2(\sigma_k^i + \sigma_k^j),\ \forall i \neq j \tag{17}$$

Ao se utilizar a Equação 17 pode ocorrer de todos os micro-grupos estarem conectados entre si, resultando em um grande macro-grupo. Com o intuito de evitar esse cenário, utiliza-se um filtro dos micro-grupos baseado na densidade de cada um deles. Seja $\mathfrak{M}_j = \{\mathbf{m}_1^j, \mathbf{m}_2^j, ..., \mathbf{m}_l^j\}$ o j-ésimo macro-grupo composto por l micro-grupos conectados. Os micro-grupos ativos de $\mathfrak{M}_j$ são aqueles cuja densidade D_k^l é maior ou igual à densidade média do macro-grupo $\mathfrak{M}_j$ [16]

$$ativo(\mathbf{m}_l^j) = \begin{cases} 1,\ D_k^l \geq \frac{1}{|\mathfrak{M}_j|}\sum_{l=1}^{|\mathfrak{M}_j|} D_k^l, l = 1, ..., |\mathfrak{M}_j| \\ 0,\ c.c. \end{cases} \tag{18}$$

A aplicação desse filtro resulta na desativação dos micro-grupos que se encontram em regiões de baixa densidade, enquanto aqueles que estão em regiões de densidade elevada continuarão ativos. Desse modo, considera-se que as regiões de baixa densidade correspondem à separação dos macro-grupos que se sobrepõem. Além disso, é importante notar que, à medida que novas observações são disponibilizadas, a atualização dos micro-grupos pode modificar as regiões de maior densidade e, consequentemente, os macro-grupos resultantes do agrupamento.

Por último, o cálculo da estimativa de densidade de cada um dos macro-grupos é dada pela soma das tipicidades normalizadas multiplicada pela densidade normalizada de cada micro-grupo w_k^l que constitui um macro-grupo $\mathfrak{M}_j$, similar à uma mistura de densidades

$$\mathcal{T}_j(\mathbf{x}_k) = \sum_{l \in \mathfrak{M}_j} w_k^l t_k^l(\mathbf{x}_k) \tag{19}$$

em que:

$$w_k^l = \frac{D_k^l}{\sum_{l \in \mathfrak{M}_j} D_k^l} \tag{20}$$

Sendo assim, uma observação $\mathbf{x}_k$ é atribuída ao macro-grupo para o qual tem maior pertinência baseada na mistura de tipicidades $\mathcal{T}_j(\mathbf{x}_k)$.

D. Rede Neural de Base Radial

As Redes Neurais de Função de Base Radial (RBFs), desenvolvidas em 1988 por David S. Broomhead e David Lowe [17], possuem em sua arquitetura três camadas de neurônios: (i) camada de entrada, (ii) camada escondida e (iii) camada de saída. Os neurônios da camada escondida utilizam funções de base radial com o intuito de tornar os dados linearmente separáveis e assim ser capaz de realizar o mapeamento necessário da camada de entrada para a camada escondida.

Dado um neurônio p na camada escondida, a função de base radial utilizada para modelá-lo é tipicamente a função Gaussiana:

$$\psi(\mathbf{x}_k, \boldsymbol{\mu}_p, \sigma_p) = e^{-||\mathbf{x}_k - \boldsymbol{\mu}_p||_2/2\sigma_p^2} \tag{21}$$

em que $\boldsymbol{\mu}_p$ e σ_p representam, respectivamente, o centro e o raio da função Gaussiana e $||\mathbf{x}_k - \boldsymbol{\mu}_p||_2$ a distância Euclidiana entre $\mathbf{x}_k$ e o centro da função de base radial $\boldsymbol{\mu}_p \in \mathbb{R}^d$. Por fim, o mapeamento da entrada $\mathbf{X} \in \mathbb{R}^{n \times d}$ à camada escondida realizado por todos os K neurônios compõe a matriz de ativações $\mathbf{A} \in \mathbb{R}^{n \times K}$.

Com o intuito de se obter um bom desempenho em tarefas de classificação, os hiperparâmetros $\boldsymbol{\mu}_p$ e σ_p, assim como o número de neurônios K da camada escondida, devem ser ajustados adequadamente. Existem diversos métodos para esse fim. Dentre os mais recorrentes, tem-se o algoritmo de clusterização *K-Means* [18] e sua versão nebulosa *Fuzzy C-Means* [19]. Ambos algoritmos não supervisionados consistem em agrupar os dados por proximidade em um número de *clusters* previamente definido, onde cada um desses *clusters* corresponde a um neurônio na camada escondida da rede RBF. No caso do *K-Means*, esse número de *clusters* é representado pelo parâmetro K e no *Fuzzy C-Means* por C. No presente trabalho assume-se a representação K para ambos algoritmos, de modo a simplificar as explicações.

Tendo em vista que os neurônios são modelados por funções de base radial, seus hiperparâmetros podem ser calculados a partir dos *clusters* obtidos. A estimativa mais usual considera o centro $\boldsymbol{\mu}_p$ como o centroide do *p*-ésimo *cluster* e o raio σ_p como [20]

$$\sigma_p = \frac{d_{\max}}{\sqrt{2K}} \quad \forall \quad p \in [1, K] \tag{22}$$

em que $d_{\max}$ é a distância máxima entre dois centros quaisquer dos *clusters*. É interessante ressaltar que existem inúmeras outras heurísticas para determinar o raio da função de base radial, inclusive algumas que atribuem larguras distintas para cada neurônio da camada escondida.

No entanto, uma notável desvantagem dos algoritmos citados é justamente a necessidade de determinar o número de *clusters* como parâmetro de entrada para cada problema. A definição desse valor pode ser realizada através da validação cruzada [21] ou do *Elbow Method* [22], por exemplo, o que pode ser bastante custoso computacionalmente. A fim de solucionar essa premissa, o algoritmo *Geometric Vectors* estima todos os hiperparâmetros necessários para a composição dos neurônios da camada escondida diretamente a partir do conjunto de dados, sem demandar conhecimento prévio do usuário [23]. Contudo, esse método não foi avaliado no contexto de aprendizado *online*, no qual requer uma adaptação da estrutura da rede RBF em virtude de possíveis alterações na distribuição dos dados (*concept drift*) à medida que novas amostras são disponibilizadas.

Sob essa perspectiva, propõe-se no presente trabalho uma nova abordagem para a definição da topologia da rede RBF para o contexto de aprendizado *online*. Tal metodologia tem como princípio o MicroTeda, algoritmo de agrupamento de fluxos de dados contínuos baseado na mistura de tipicidades, proposto em [16] e aqui descrito na Seção anterior. A cada nova amostra de dados recebida, o MicroTeda atualiza os macro-grupos e os utiliza para modelar a camada escondida da rede neural. A quantidade de neurônios K é definida conforme o número de macro-grupos obtidos no instante de tempo k e os hiperparâmetros $\boldsymbol{\mu}_p$ e σ_p da função Gaussiana do neurônio p são, respectivamente, o centro e o raio do p-ésimo macro-grupo. Essa sucessiva atualização da topologia da rede implica na adaptação do modelo a diferentes cenários e mudanças de conceito.

Posteriormente, a matriz de pesos $\mathbf{W} \in \mathbb{R}^K$, responsável por mapear a camada intermediária à camada de saída, é obtida de maneira similar às *Extreme Learning Machines* [24]

$$\mathbf{W} = \mathbf{A}^+ \mathbf{y} \tag{23}$$

onde $\mathbf{A}^+$ é a pseudo-inversa da matriz de ativações $\mathbf{A}$ e $\mathbf{y} \in \mathbb{R}^n$ são os rótulos correspondentes à matriz de entrada $\mathbf{X}$. Por fim, as predições $\hat{\mathbf{y}}$ são obtidas a partir da combinação linear dos pesos $\mathbf{W}$ com a matriz de mapeamento $\mathbf{A}$:

$$\hat{\mathbf{y}} = \mathbf{A}\mathbf{W} \tag{24}$$

III. Experimentos e Resultados

A. Dados Estáticos

Ao se propor um método para o contexto de fluxos de dados contínuos, faz-se relevante realizar um estudo preliminar, cujo objetivo é validar o seu desempenho em tarefas de classificação menos complexas, tal como um conjunto de dados estático. Sob essa perspectiva, realizou-se um experimento utilizando o conjunto de dados Iris [25], disponibilizado no repositório *UCI Machine Learning* [26], para avaliar o desempenho da integração do algoritmo MicroTeda com uma rede RBF. Para fins de comparação, o *Fuzzy C-Means* também foi integrado a uma rede RBF.

A análise do desempenho dos modelos utilizados foi feita por meio da técnica *nested cross validation*, onde emprega-se dois *5-fold cross validation* aninhados. Em outras palavras, o conjunto de dados é dividido entre treino e teste cinco vezes (primeiro *5-fold cross validation*) e cada um desses conjuntos de treino é dividido ainda entre treino e validação também cinco vezes (segundo *5-fold cross validation*). O treinamento de cada algoritmo foi então realizado em cada um desses conjuntos de treino resultantes do segundo *5-fold cross validation* e os seus respectivos hiperparâmetros foram selecionados através da acurácia média obtida nos cinco conjuntos de validação. Para o *Fuzzy C-Means* foi ajustado o valor do parâmetro m_{we}, que representa o expoente aplicado ao cálculo de atualização da matriz de pertinências, e para o MicroTeda ajustou-se o parâmetro r_0. Os valores de m_{we} e r_0 investigados foram definidos segundo um *Grid Search* com cem valores de m_{we} e r_0 uniformemente distribuídos em seus respectivos intervalos. As curvas de ajuste desses hiperparâmetros são apresentadas nas Figuras 4 e 5.

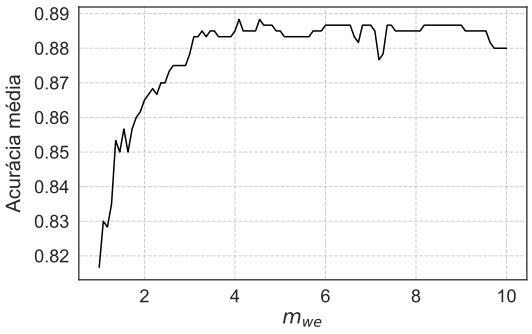

Figura 4. Influência do parâmetro m_{we} na acurácia média do conjunto de validação.

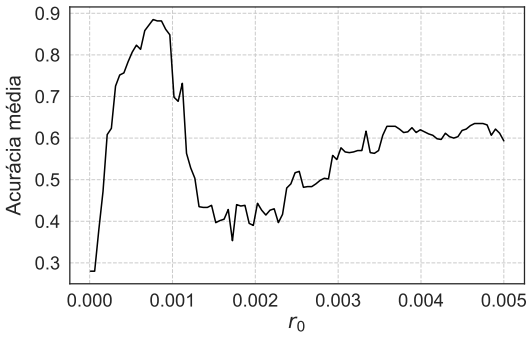

Figura 5. Influência do parâmetro r_0 na acurácia média do conjunto de validação.

Uma vez que os melhores valores de $m_{we} = 4,0909$ e $r_0 = 0,0007$ foram escolhidos no conjunto de validação, o desempenho de cada modelo foi avaliado nos cinco conjuntos de teste disponíveis. A acurácia média e o desvio padrão resultante de ambos os métodos compõem a Tabela I.

Tabela I
ACURÁCIAS MÉDIAS NO CONJUNTO DE TESTE.

Algoritmo	Acurácia
Fuzzy C-Means + RBF	0.8408 ± 0.0249
MicroTeda + RBF	0.9000 ± 0.0558

Observa-se que a integração do MicroTeda à rede RBF tem desempenho médio superior à integração do *Fuzzy C-Means* com uma diferença de 6 p.p. Ademais, vale ressaltar que para a execução do *Fuzzy C-Means* é necessário, além do ajuste do hiperparâmetro m_{we}, especificar o número de grupos que se deseja obter, o que requer um conhecimento prévio dos dados.

Para se fazer o ajuste do MicroTeda não é preciso ter conhecimento prévio dos dados, tendo em vista que a quantidade de grupos criados resulta do agrupamento evolutivo feito pelo algoritmo. Nesse caso, faz-se apenas o ajuste do hiperparâmetro r_0, utilizado para limitar a variância dos grupos ao longo de sua execução. Sendo assim, pode-se concluir que o método proposto está apto para ser testado em um contexto de fluxo de dados contínuos, diante do desempenho satisfatório apresentado em um ambiente estático.

B. Fluxos de Dados Contínuos

No cenário de fluxo de dados contínuos, cada *batch* de dados recebido é tratado de maneira análoga ao que é feito com um conjunto de dados estático. No entanto, um treinamento inicial da rede com n_0 observações é realizado. Posteriormente, recebe-se um *batch* para avalição do modelo e essa mesma amostra é utilizada para atualizar a arquitetura da rede. Enquanto houver amostras disponíveis na base de dados, esse procedimento, intitulado *test-then-train* [27], se repete.

A fim de analisar o comportamento da metodologia proposta no contexto *online* foram consideradas 5 bases de dados, onde 4 são de dados sintéticos e 1 de dados reais. A avaliação da performance do método proposto foi feita utilizando a metodologia *Prequential Evaluation* [28], comumente utilizada em cenários de classificação de fluxos de dados contínuos. *Prequential* permite quantificar a acurácia média do modelo considerando todos os *batches* recebidos na ordem em que se encontram disponíveis na base de dados.

Para que a abordagem proposta obtivesse o seu melhor desempenho em cada base de dados, foi realizado, individualmente, o ajuste de hiperparâmetros para cada umas das bases de dados descritas a seguir. Esse ajuste foi feito através do *Grid Search*, considerando todos os *batches* de dados, com exceção do *batch* utilizado para realizar o treinamento inicial da rede. As letras **`A`** e **`G`** nas nomenclaturas de algumas bases indicam, respectivamente, bases com mudanças de conceito abruptas e graduais.

- **`Sea-0123-G`** [29]: conjunto de dados de dados sintéticos, criados a partir de uma sequência de funções de classificação. Esta base de dados possui 40.000 observações, três atributos e 2 classes balanceadas, bem como uma probabilidade de 0,2 de ocorrência de ruído.
- **`Sine 1`** [27]: esta base sintética de classificação binária possui 10.000 observações uniformemente distribuídas entre 0 e 1, dois atributos sem ruído e apresenta mudança de conceito abrupta. No primeiro contexto todos os pontos sob a curva $y = sin(x)$ são classificados como positivos, após a mudança de conceito essa classificação é invertida.
- **`Sine-0123-G`** [29]: trata-se de uma base de dados sintética composta por 40.000 observações, 2 atributos sem ruídos e 2 classes balanceadas (20.000 observações para cada uma). Há mudanças de conceito nos instantes 9.500, 20.000 e 30.500.
- **`Mixed-0101-A`** [29]: esta base sintética, criada a partir de uma sequência de funções de classificação na ordem 0-1-0-1, possui 40.000 observações, 4 atributos sem ruído e classes binárias balanceadas. Além disso, apresenta mudanças de conceito nos instantes 10.000, 20.000 e 30.000.
- **`Mixed-1010-G`** [29]: este conjunto de dados, gerado por uma sequência de funções de classificação seguindo a ordem 1-0-1-0, contém 40.000 observações, 4 atributos sem ruído e 2 classes balanceadas.
- **`Electricity`** [30]: é uma base de dados relacionada à

mudança histórica de preço da energia elétrica contendo 45.312 observações, que vão desde 7 de maio de 1996 até 05 de dezembro de 1998. Cada observação corresponde a um intervalo de 30 minutos, isto é, são 48 observações por dia. Além disso, possui 5 atributos e 2 classes que retratam a mudança nos preços (1 quando o preço subiu e 0 quando o preço diminuiu). Tendo em vista que se tratam de dados reais, os instantes exatos em que as mudanças de conceito ocorrem são desconhecidos [27].

A Tabela II apresenta os valores avaliados de cada hiperparâmetro com o *Grid Search*: limiar de variância do micro-grupo (r_0), fator de esquecimento (λ), tamanho do *batch* inicial (n_0) e tamanho do *batch* regular (n).

Tabela II
HIPERPARÂMETROS UTILIZADOS NO GRID SEARCH.

Hiperparâmetro	Valores
r_0	{0,00005; 0,0001; 0,0002; 0,0005}
λ	{500; 1000; 2000; 3000; 4000; 5000}
n_0	{600; 800; 1000; 1200; 2000}
n	{100; 200; 500; 600; 800; 1200}

Após o ajuste de hiperparâmetros, os melhores valores encontrados para cada uma das bases de dados são apresentados na Tabela III. Vale ressaltar que não foi necessário realizar mais de um experimento para cada conjunto de dados, visto que o algoritmo é determinístico, isto é, apresenta o mesmo desempenho em cada problema dada a mesma configuração de hiperparâmetros.

Tabela III
MELHORES HIPERPARÂMETROS ENCONTRADOS PARA CADA CONJUNTO DE DADOS.

Base de Dados	r_0	λ	n_0	n
Sine 1	0,00005	1000	2000	500
Sea-0123-G	0,00005	4000	800	200
Sine-0123-G	0,0001	500	600	200
Mixed-0101-A	0,0005	500	600	200
Mixed-1010-G	0,0005	500	600	100
Electricity	0,0002	5000	1200	1200

O desempenho da abordagem proposta foi comparado ao *Incremental* LSTM (ILSTM) [8], que também se trata de um método de aprendizado incremental baseado em redes neurais. Nesse caso, como o algoritmo ILSTM é estocástico, foram realizadas 5 execuções em cada uma das bases de dados. Os hiperparâmetros da arquitetura, como número de camadas, número de neurônios, funções de ativação, otimizador, taxa de aprendizado, tamanho do *batch* e número de épocas para treinamento, foram mantidos de acordo com o trabalho original. A Tabela IV apresenta os resultados para comparação.

Tabela IV
RESULTADOS DOS EXPERIMENTOS.

Base de Dados	Método	Acurácia (%)
Sine 1	**IRBF**	**98,05**
	ILSTM	97,34 ± 8,63
Sea-0123-G	IRBF	70,53
	ILSTM	**72,72 ± 2,56**
Sine-0123-G	**IRBF**	**91,87**
	ILSTM	74,37 ± 1,30
Mixed-0101-A	**IRBF**	**92,59**
	ILSTM	50,00 ± 0,0
Mixed-1010-G	**IRBF**	**91,06**
	ILSTM	80,35 ± 1,48
Electricity	**IRBF**	**63,75**
	ILSTM	59,47 ± 12,27

Conforme observado na Tabela IV, a RBF incremental obteve desempenho superior aos valores médios da ILSTM em 4 das 5 bases avaliadas, mesmo com um número substancialmente menor de hiperparâmetros. Além disso, para as bases **Electricity** e **Sine 1**, a ILSTM apresenta um elevado desvio, o que indica que a abordagem proposta também é mais robusta. Ao contrário do que é evidenciado para a LSTM incremental, os resultados da IRBF sugerem uma boa adaptação para mudanças de conceito abruptas. No entanto, quando a base de dados apresenta ruído, como é o caso da **Sea-0123-G**, a IRBF apresenta um certa dificuldade de adaptação. Por fim, é importante ressaltar que o desempenho da RBF incremental não é muito sensível ao hiperparâmetro r_0, de forma que valores pequenos (na ordem de 10^{-3}) funcionaram bem para a maior parte das bases de dados.

IV. CONCLUSÃO

Este trabalho propôs uma nova abordagem para definição da topologia de uma rede RBF no contexto de fluxos de dados contínuos. A constante adaptação das funções de densidade (funções de ativação da camada escondida) via agrupamento evolutivo, provê à rede RBF a capacidade de aprender incrementalmente.

Os resultados apresentados sugerem que o método incremental proposto é capaz de se adaptar as adversidades impostas pelo ambiente de fluxos de dados contínuos. No entanto, este estudo preliminar foi realizado majoritariamente em bases de dados sintéticas e requer uma análise mais detalhada em problemas reais. Outras direções futuras de investigação incluem a implementação de um mecanismo capaz de promover uma melhor adaptação do modelo a ruídos, bem como uma comparação sistemática de seu desempenho com mais bases de dados e outros algoritmos do estado-da-arte.

REFERÊNCIAS

[1] C. S. K. Dash, A. Behera, S. Dehuri, and S. Cho, "Radial basis function neural networks: a topical state-of-the-art survey," *Open Computer Science*, vol. 6, pp. 33 – 63, 2016.

[2] Q. Yang, Y. Gu, and D. Wu, "Survey of incremental learning," in *2019 Chinese Control And Decision Conference (CCDC)*. IEEE, 2019, pp. 399–404.

[3] A. Chefrour, "Incremental supervised learning: algorithms and applications in pattern recognition," *Evolutionary Intelligence*, pp. 1–16, 2019.

[4] J. L. Lobo, I. Laña, J. Del Ser, M. N. Bilbao, and N. Kasabov, "Evolving spiking neural networks for online learning over drifting data streams," *Neural Networks*, vol. 108, pp. 1–19, 2018.

[5] A. A. Khater, A. M. El-Nagar, M. El-Bardini, and N. M. El-Rabaie, "Online learning based on adaptive learning rate for a class of recurrent fuzzy neural network," *Neural Computing and Applications*, vol. 32, no. 12, pp. 8691–8710, 2020.

[6] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9, pp. 1469–1495, 2017.

[7] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] Á. C. L. Neto, R. A. Coelho, and C. L. de Castro, "An incremental learning approach using long short-term memory neural networks," *Anais da Sociedade Brasileira de Automática*, vol. 2, no. 1, 2020.

[9] P. Angelov, "Outside the box: An alternative data analytics framework," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 8, pp. 29–35, 04 2014.

[10] J. G. Saw, M. C. Yang, and T. C. Mo, "Chebyshev inequality with estimated mean and variance," *The American Statistician*, vol. 38, no. 2, pp. 130–132, 1984.

[11] P. Angelov, "Anomaly detection based on eccentricity analysis," in *2014 IEEE symposium on evolving and autonomous learning systems (EALS)*. IEEE, 2014, pp. 1–8.

[12] C. G. Bezerra, B. S. J. Costa, L. A. Guedes, and P. P. Angelov, "A new evolving clustering algorithm for online data streams," in *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE, 2016, pp. 162–168.

[13] M. Neto, "Proposta de algoritmo evolutivo inteligente para agrupamento de fluxos continuos de dados," Master's thesis, Universidade Federal de Minas Gerais, Minas Gerais, Brasil, 2018.

[14] R. Hyde, P. Angelov, and A. R. MacKenzie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters," *Information Sciences*, vol. 382, pp. 96–114, 2017.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[16] J. Maia, C. A. S. Junior, F. G. Guimarães, C. L. de Castro, A. P. Lemos, J. C. F. Galindo, and M. W. Cohen, "Evolving clustering algorithm based on mixture of typicalities for stream data mining," *Future Generation Computer Systems*, vol. 106, pp. 672–684, 2020.

[17] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Royal Signals and Radar Establishment Malvern (United Kingdom), Tech. Rep., 1988.

[18] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[19] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[20] S. S. Haykin *et al.*, "Neural networks and learning machines/simon haykin." 2009.

[21] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[22] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.

[23] L. C. Torres, A. P. Lemos, C. L. Castro, and A. P. Braga, "A geometrical approach for parameter selection of radial basis functions networks," in *International Conference on Artificial Neural Networks*. Springer, 2014, pp. 531–538.

[24] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2. Ieee, 2004, pp. 985–990.

[25] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[26] ——, "UCI machine learning repository," 1988. [Online]. Available: http://archive.ics.uci.edu/ml

[27] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian symposium on artificial intelligence*. Springer, 2004, pp. 286–295.

[28] A. P. Dawid, "Present position and potential developments: Some personal views statistical theory the prequential approach," *Journal of the Royal Statistical Society: Series A (General)*, vol. 147, no. 2, pp. 278–290, 1984.

[29] J. López Lobo, "Synthetic datasets for concept drift detection purposes," 2020. [Online]. Available: https://doi.org/10.7910/DVN/5OWRGB

[30] M. Harries and N. S. Wales, "Splice-2 comparative evaluation: Electricity pricing," 1999.