

Aplicação da Conexão Preferencial no Domínio de Séries Temporais

Daniil Sveshnikov¹, Gildo G. A. Oliveira², Sofia F. Martins²

¹Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (USP)
São Carlos, SP - Brasil

²Instituto de Física de São Carlos - Universidade de São Paulo (USP)
São Carlos, SP - Brasil

daniil_s@usp.br, gildogutemberg2@usp.br, sofiafreitasmartins@usp.br

Abstract. *Weight initialization in artificial neural networks (ANN) has a direct impact on training efficiency and model generalization. In this work, we explore Preferential Attachment Rewiring (PARw), an approach that aims to improve this step. This technique uses concepts from network science to reorganize random weights, balancing the distribution of neuron strengths. This method has been shown to improve training stability, initial performance, and final accuracy in several deep learning tasks, especially image classification. The goal of this work is to apply PARw to the time series domain, which was never done before.*

Resumo. *A inicialização de pesos em redes neurais artificiais (ANN) possui impacto direto na eficiência do treinamento e na generalização de modelos. Neste trabalho, exploramos o Preferential Attachment Rewiring (PARw), abordagem que visa aprimorar essa etapa. Tal técnica utiliza conceitos de ciência de redes para reorganizar pesos aleatórios, equilibrando a distribuição de forças dos neurônios. Este método tem demonstrado melhorar a estabilidade do treinamento, o desempenho inicial e a precisão final em diversas tarefas de aprendizado profundo, em especial a classificação de imagens. O objetivo deste trabalho é aplicar o PARw no domínio de séries temporais, algo nunca feito antes.*

1. Introdução

A popularidade das redes neurais artificiais no aprendizado profundo se deve principalmente à sua capacidade de aprender coisas complexas de grandes quantidades de dados. Todavia, o sucesso do treinamento dessas redes depende de importantes fatores como a inicialização de pesos. Inicializadores tradicionais, como os de métodos Kaiming e Glorot, foram projetados para reduzir problemas de gradientes explosivos. No entanto, questões como saturação de neurônios e a variância no desempenho entre diferentes inicializações ainda permanecem.

Recentemente, foram explorados avanços na ciência de redes para solucionar e melhorar a compreensão e o controle da complexidade de redes neurais. Dentre os

métodos, surge o PARw (Preferential Attachment Rewiring), que realoca pesos aleatórios de forma a suavizar a distribuição de forças dos neurônios. Esta redistribuição baseia-se no princípio da conexão preferencial com o objetivo de evitar valores extremos de força e facilitar o aprendizado coletivo durante o treinamento. A pesquisa empírica de [Scabini et al. 2023] mostra que o PARw melhora o desempenho inicial quanto a generalização de modelos treinados em tarefas como classificação de imagens.

2. Trabalhos relacionados

[Scabini et al. 2023] introduziu um novo método chamado Preferential Attachment Rewiring (PARw) com o objetivo de melhorar a inicialização de redes neurais artificiais (ANN). O método é realizado por meio da reorganização de pesos aleatórios após a inicialização e se baseia em alguns conceitos da ciência de redes, de modo que a distribuição da força dos neurônios seja equilibrada. Os autores demonstraram que o método ajuda no desempenho da inicialização e na generalização das redes neurais artificiais para diferentes tarefas, incluindo classificação de imagens em conjuntos de dados MNIST e CIFAR10. Além disso, a abordagem é caracterizada por um baixo custo computacional. Neste trabalho, vamos testar essa técnica no domínio de séries temporais, utilizando diversos conjuntos de dados disponibilizados por [Dau et al. 2018] para testar a validade do método em diferentes séries temporais.

Um dos primeiros esforços para unir séries temporais e redes neurais foi feito por [Wang et al. 2016]. Eles propuseram uma linha de base para classificação de séries temporais com a utilização de redes neurais profundas. Em especial, aproveitaremos suas ideias para arquiteturas de perceptron multicamadas (MLP) e redes totalmente convolucionais (FCN).

3. Arquiteturas de redes neurais

3.1. MLP

A arquitetura MLP é uma implementação clássica de redes neurais, adaptada para atender às demandas da classificação de séries temporais modernas. Ela consiste em três camadas totalmente conectadas (fully connected), cada uma composta por 500 neurônios. Esta arquitetura foi projetada seguindo dois princípios fundamentais:

1. Dropout para Regularização: O dropout é aplicado em cada camada para evitar overfitting, com taxas configuradas em 0.1 na camada de entrada, 0.2 nas camadas ocultas e 0.3 na camada de saída. Essa abordagem reduz a coadaptação entre os neurônios e melhora a capacidade de generalização do modelo, especialmente em datasets pequenos.
2. Função de Ativação ReLU: A Unidade Linear Retificada (ReLU) foi escolhida como a função de ativação padrão em todas as camadas. A ReLU resolve problemas clássicos de saturação do gradiente em funções sigmóides e tangentes hiperbólicas, permitindo que redes mais profundas sejam treinadas de forma eficiente.

A saída do modelo é gerada por uma camada softmax, que transforma as ativações em probabilidades para a classificação. Embora simples, esta arquitetura demonstrou ser competitiva, alcançando desempenho semelhante ao método 1-NN Dynamic Time Warping (DTW), considerado um padrão na classificação de séries temporais. A combinação

de dropout e ReLU contribuiu significativamente para o desempenho robusto da MLP, mesmo em conjuntos de dados com características variadas.

3.2. FCN

As Redes Totalmente Convolucionais (Fully Convolutional Networks - FCNs) representam uma abordagem moderna e eficaz para a classificação de séries temporais, aproveitando os avanços obtidos por Redes Neurais Convolucionais (CNNs) em tarefas de visão computacional. Diferentemente das arquiteturas tradicionais que utilizam camadas totalmente conectadas, as FCNs substituem estas camadas por operações convolucionais seguidas de pooling global, o que possibilita a extração direta de padrões discriminativos sem a necessidade de etapas intermediárias de pré-processamento ou engenharia de características.

Estrutura da Rede: A arquitetura das FCNs é composta por três blocos convolucionais sequenciais, cada um com as seguintes características principais:

1. Camadas Convolucionais: Cada bloco convolucional utiliza kernels unidimensionais com tamanhos 8, 5, 3, associados a profundidades de filtro 128, 256, 128, respectivamente. Esses filtros são projetados para capturar padrões em diferentes escalas temporais, otimizando a representação das características extraídas.
2. Normalização em Lote: Após cada camada convolucional, é aplicada a normalização em lote (Batch Normalization), que estabiliza e acelera o treinamento ao reduzir o desvio covariante interno, além de facilitar a generalização do modelo.
3. Função de Ativação ReLU: Cada bloco convolucional emprega a função de ativação ReLU (Rectified Linear Unit), que introduz não-linearidade ao modelo, permitindo a captura de interações complexas entre as características.

Após os blocos convolucionais, os recursos extraídos passam por uma camada de pooling médio global (Global Average Pooling). Esta etapa sintetiza cada canal convolucional em um único valor, substituindo as camadas totalmente conectadas convencionais. Essa abordagem reduz significativamente o número de parâmetros, diminuindo o risco de overfitting e tornando o modelo mais eficiente. A etapa final da arquitetura é uma camada softmax, responsável por converter os recursos extraídos em probabilidades para classificação.

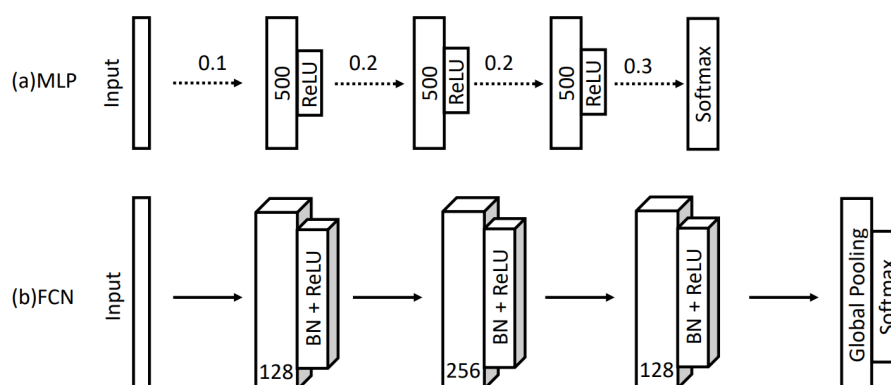


Figura 1. A estrutura de rede de (a) MLP e (b) FCN. A linha tracejada indica uma operação de dropout. Fonte: [Wang et al. 2016].

4. Metodologia

4.1. Conexão Preferencial

Um processo de diversos modelos propostos para explicar a evolução topológica de redes complexas é a conexão preferencial, entendida simplificada como "os ricos ficam mais ricos", isto é, os nós altamente conectados aumentam sua conectividade mais rápido do que seus pares menos conectados. A probabilidade $\Pi(k)$ de que um link do novo nó se conecte ao nó i depende do grau k_i , dada por [Barabási and Pósfai 2016]:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \quad (1)$$

A conexão preferencial é portanto um mecanismo probabilístico, em que um novo nó é livre para se conectar a qualquer nó na rede, podendo este ser um hub ou ter apenas um link. Todavia, (1) mostra que se um novo nó pode escolher entre um nó de grau três e um nó de grau seis, por exemplo, é duas vezes mais provável que ele se conecte ao nó de grau seis. Esse mecanismo é utilizado no algoritmo PARw, explicado na seção 4.2.

4.2. PARw

O PARw (Preferential Attachment Rewiring) é um método de reorganização de pesos em redes neurais artificiais proposto por [Scabini et al. 2023] e introduzido para lidar com limitações dos inicializadores tradicionais de pesos, que trata o processo de inicialização como algo totalmente aleatório, isto é, sem considerar como a organização inicial afeta a aprendizagem.

Para aplicar o PARw, cada camada de uma ANN é representada como um grafo bipartido. Nesta representação:

- Os nós de entrada (V_l) correspondem aos neurônios da camada anterior.
- Os nós de saída (V_{l+1}) representam os neurônios da camada atual.
- As arestas (E) simbolizam os pesos das conexões entre neurônios, sendo os valores desses pesos os responsáveis pela "força" das conexões.

O peso de uma conexão entre um neurônio i na camada de entrada e um neurônio j na camada de saída é denotado por W_{ij} . Assim, a camada inteira pode ser expressa matematicamente como um grafo bipartido $G(W)$, onde:

$$G(W) = (V_l \times V_{l+1}, E) \quad (2)$$

e $E = W$. Essa representação permite modelar as camadas de forma generalizada, incluindo camadas densas (totalmente conectadas) e convolucionais, ao expressar pesos como matrizes bidimensionais.

Quando uma rede neural é inicializada de forma aleatória, os pesos das conexões entre os neurônios podem gerar neurônios com forças muito altas ou muito baixas, chamados de hubs. A força de um neurônio $s(i)$ é simplesmente a soma dos pesos das conexões que chegam ou saem dele, sendo o grau do nó ponderado:

$$s(i) = \sum_j W_{ij} \quad (3)$$

Essa força pode ser interpretada como uma estimativa da taxa média de disparo do neurônio, com base em uma soma ponderada de entradas, sem incluir viés, e assumindo que as entradas são estatisticamente independentes e seguem a mesma distribuição, com função de ativação $f_i(\sum_j W_{ij} X_j)$. Esse termo indica a soma das entradas X_j multiplicadas pelos pesos W_{ij} cuja soma representa a força (ou importância) da conexão entre o neurônio j e i .

O problema de gerar hubs é que eles podem indicar que um neurônio está contribuindo de maneira desproporcional, causando saturação ou inatividade. A fim de resolver isso, o PARw ajusta a distribuição dos pesos na camada, reorganizando as conexões de maneira que as forças fiquem balanceadas. A redistribuição é feita seguindo uma probabilidade calculada a partir da força atual de cada neurônio. A probabilidade $P(i)$ de um neurônio i receber uma nova conexão é dada por:

$$P(i) = \frac{s_t(i) + |\min(s_t)| + 1}{\sum_j (s_t(j) + |\min(s_t)| + 1)} \quad (4)$$

em que $s_t(i)$ é a força do neurônio i na iteração atual t e $|\min(s_t)| + 1$ é um termo de ajuste que significa que deslocamos a distribuição da força para o lado positivo e adicionamos 1 para evitar probabilidades nulas. Nesse sentido, a probabilidade varia desde a menor força negativa (probabilidade menor) até a maior força positiva (probabilidade maior).

Repare que $P(i)$ segue a lógica invertida da conexão preferencial, descrita na seção 4.1, ou seja, neurônios com forças muito positivas têm maior chance de receber pesos negativos e vice-versa. Em cada iteração t , uma nova força é calculada após a última reorganização realizada na iteração anterior $t - 1$:

$$s_t(i) = \sum_{x=1}^{t-1} W_{ix} = s_{t-1}(i) + W_{i \ t-1} \quad (5)$$

Uma nova reorganização com respeito aos neurônios de entrada pode então ser realizada com os pesos de um novo neurônio $x = t$. Após reorganizar todos os neurônios na camada $l + 1$, esse procedimento reorganizará os pesos da rede neural de forma que a força neuronal dos neurônios de entrada se aproxime de zero. O processo também é descrito no pseudocódigo em 2:

Algoritmo 1: A python-like pseudo-code of the proposed PARw method.

```

1 function PA_rewiring( $\mathcal{W}$ );
   Input : An  $n_l \times n_{l+1}$  weight matrix  $\mathcal{W}$  filled by some weight initializer
   Output:  $\mathcal{W}$  after rewiring
2  $s_1 = \text{zeros}(n_l)$ 
3 for  $t$  from 2 to  $n_{l+1}$ , do
4    $s_t = s_{t-1} + \mathcal{W}[:, t-1]$ 
5    $P = s_t + \text{abs}(\min(s_t)) + 1$ 
6    $P = P / \text{sum}(P)$ 
7   targets = random_choice( $[i \in [1, \dots, n_l]]$ , prob= $P$ , size =  $n_l$ )
8   new_edges = argsort( $\mathcal{W}[:, t]$ )
9    $\mathcal{W}[\text{targets}, t] = \mathcal{W}[\text{new\_edges}, t]$ 
10 end
11 return  $\mathcal{W}$ 

```

Figura 2. Pseudocódigo do PARw. Fonte: [Scabini et al. 2023].

O PARw é ilustrado em 7, onde se pode observar como uma distribuição inicial de forças com caudas longas (devido à presença de hubs) gerada por um inicializador tradicional desaparece quando as conexões são reorganizadas.

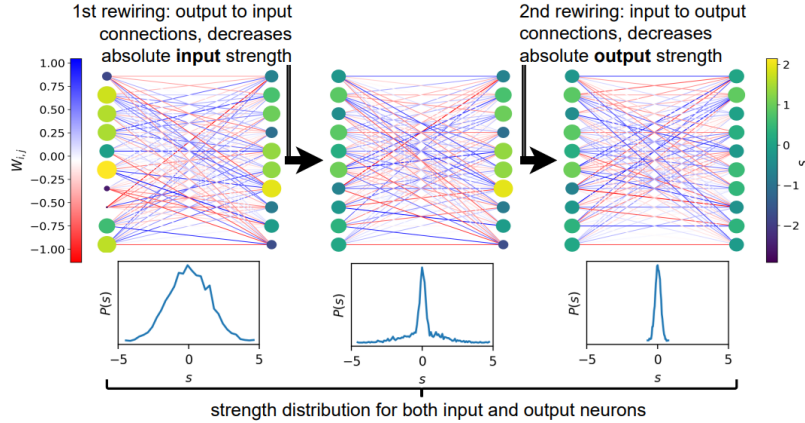


Figura 3. Ilustração do funcionamento do PARw na reorganização de pesos para diminuir a força neuronal absoluta. O tamanho dos neurônios representa sua força, do mais negativo (menor) ao mais positivo. Mostra-se uma pequena camada com 10 neurônios e 10 entradas, enquanto as distribuições de força são calculadas para uma camada com 1024 neurônios e 1024 entradas. Fonte: [Scabini et al. 2023].

4.3. Conjunto de dados

Os dados usados no trabalho são disponibilizados por [Dau et al. 2018].

Para avaliar o desempenho dos métodos propostos, utilizamos uma seleção de conjuntos de dados amplamente reconhecidos na literatura de classificação de séries temporais. Os conjuntos de dados escolhidos abrangem uma variedade de domínios, incluindo biomédico, químico e geométrico, permitindo uma avaliação abrangente das abordagens em diferentes contextos de classificação. Abaixo, detalhamos os conjuntos de dados utilizados neste trabalho:

1. **Adiac**: O conjunto de dados Adiac é composto por séries temporais que representam medições de formas tridimensionais de conchas. Cada instância do conjunto de dados corresponde a uma sequência de dados extraída das medições da borda de uma concha, observada em diferentes direções. O conjunto contém 37 classes diferentes, com cada classe representando uma espécie distinta de concha. A principal dificuldade ao trabalhar com este conjunto de dados é a alta variabilidade dentro das classes e as semelhanças visuais entre diferentes classes, o que torna a tarefa de classificação desafiadora. Este conjunto de dados é particularmente útil para testar algoritmos de classificação multiclasse em cenários com alta complexidade e sobreposição de padrões.

2. **ChlorineConcentration**: O ChlorineConcentration é um conjunto de dados que reflete medições da concentração de cloro na água ao longo do tempo. A tarefa é classificar diferentes níveis de concentração de cloro, algo fundamental no controle de qualidade da água. O conjunto possui 3 classes que representam diferentes intervalos de concentração de cloro. A principal característica deste conjunto é a alta dimensionalidade das séries temporais, o que pode aumentar o custo computacional dos algoritmos de classificação. Além disso, o modelo precisa ser sensível a pequenas variações nos dados, dado que mudanças sutis nos níveis de concentração podem ter grande impacto nas decisões.

3. **ECG200**: O conjunto de dados ECG200 é composto por séries temporais de eletrocardiogramas (ECG), com o objetivo de classificar padrões normais de ECG de anomalias cardíacas. Este conjunto contém 2 classes, sendo uma para sinais de ECG normais e outra para sinais que indicam possíveis problemas cardíacos. A característica chave do conjunto é que ele possui um número relativamente pequeno de instâncias, com 100 exemplos de treino e 100 exemplos de teste. Essa limitação no número de exemplos torna o conjunto de dados um bom desafio para testar a generalização de modelos de classificação em pequenos conjuntos de dados.

4. **ECG5000**: O conjunto ECG5000 é uma versão mais extensa do ECG200, contendo registros de 5 classes diferentes, incluindo um padrão normal de ECG e quatro tipos de anomalias cardíacas. Com 500 instâncias de treino e 4500 instâncias de teste, este conjunto oferece um desafio mais robusto para algoritmos multiclasse. A principal dificuldade no uso do ECG5000 é o desequilíbrio entre as classes, com a classe "normal" sendo substancialmente mais representada do que as classes de anomalias cardíacas. Este conjunto de dados é adequado para avaliar a performance de algoritmos de classificação em contextos com grandes volumes de dados e desequilíbrios de classe.

Esses conjuntos de dados foram escolhidos por sua relevância em contextos práticos e pela diversidade de características que oferecem. Vale ressaltar que devido às capacidades limitadas do computador (GPU), não foi possível testar totalmente todos os conjuntos de dados.

4.4. Configurações dos Experimentos

Buscando padronizar as configurações experimentais, seguimos as propostas apresentadas por [Wang et al. 2016], aplicando-as tanto ao modelo MLP quanto ao FCN. Para o MLP, utilizamos o otimizador Adadelta com uma taxa de aprendizado (learning rate) de 0,1, $\rho = 0.95$ e $\epsilon = 1e - 8$. No caso do FCN, empregamos o otimizador Adam, configurado com uma taxa de aprendizado de 0,001, $\beta_1 = 0,9$, $\beta_2 = 0,999$ e $\epsilon = 1e - 8$.

Em ambos os casos, adotamos a função de perda cross-entropy e utilizamos um tamanho de lote (batch size) de 16 e 100 épocas. Essa padronização visa garantir a comparabilidade dos resultados e minimizar vieses associados às diferenças nas configurações dos modelos.

Conforme os experimentos descritos em [Scabini et al. 2023], conduzimos todos os nossos testes modificando as seeds iniciais para mitigar o viés associado ao efeito do "bilhete de loteria", que pode ocorrer em situações onde uma seed específica favoreça resultados devido à inicialização dos pesos aleatórios da rede. Para assegurar robustez estatística, utilizamos 100 seeds diferentes ao longo dos experimentos.

Nosso objetivo principal não é alcançar ou superar as métricas do State of the Art (SOTA), mas sim verificar se as redes que passam pelo procedimento PARw apresentam desempenho superior àquelas inicializadas de forma puramente aleatória. Para isso, optamos por dividir o conjunto de dados em treino, validação e teste. Para manter o equilíbrio entre o tamanho dos conjuntos de validação e treino, redefinimos a divisão tradicional: utilizamos metade do conjunto originalmente destinado ao teste como conjunto de validação. Essa abordagem busca garantir a comparabilidade das métricas sem comprometer a integridade do experimento.

Para a avaliação da métrica no conjunto de teste, selecionamos o modelo que apresentou o melhor desempenho no conjunto de validação ao longo do treinamento. Essa abordagem busca assegurar que os resultados reportados no conjunto de teste reflitam a capacidade generalizada do modelo,

5. Resultados e Discussões

Inicialmente, submetemos a arquitetura do MLP a um rigoroso teste utilizando os quatro datasets selecionados para validação. Cada experimento consistiu em um treinamento de 100 épocas, repetido para 100 diferentes inicializações de seeds, garantindo uma avaliação robusta da inicialização padrão dos pesos. Em seguida, repetimos o mesmo processo de treinamento, desta vez aplicando o método PARw para a inicialização. Os resultados obtidos estão resumidos na Tabela da imagem 4.

MLP				
Dataset	Peso	ep. 1 acc Tre.	ep. 1 acc Val.	Teste Avr.
Adiac	Normal	2,58 +/-0,82	2,73+/-1,55	16,07+/-2,28
	PARw	2,58+/-0,85	3,01+/-1,46	17,29+/-2,33
Chlorine	Normal	36,45+/-3,69	52,07+/-3,13	54,75+/-1,41
	PARw	37,89+/-2,49	52,49+/-2,76	54,23+/-1,23
ECG200	Normal	54,96+/-9,17	61,54+/-14,10	84,02+/-2,79
	PARw	56,64+/-7,18	68,02+/-8,07	86,02+/-2,32
ECG5000	Normal	45,91+/-8,64	89,51+/-1,79	93,87+/-0,15
	PARw	49,99+/-5,55	90,88+/-0,89	94,16+/-0,13

Figura 4. Os dados apresentados referem-se ao treinamento e teste da arquitetura MLP aplicada aos datasets selecionados.

Na Tabela da figura 4, apresentamos as métricas de acurácia para os conjuntos de treino e validação na primeira época, bem como a média dos valores de teste. Esses resultados foram calculados com base na média dos desempenhos obtidos em todas as seeds.

A análise visual da tabela indica que o método PARw demonstrou um desempenho ligeiramente superior em relação à inicialização padrão. No entanto, para determinar se essas diferenças são estatisticamente significativas, é necessário realizar um teste estatístico adequado.

Nas imagens 5 até 8, apresentamos os gráficos da acurácia média por seed ao longo das épocas, tanto para o dataset de treinamento quanto para o de validação. Observa-se que, em alguns casos, há um deslocamento superior do método PARw em comparação com a abordagem original. No entanto, devido ao pequeno número de datasets analisados, não é possível generalizar essa impressão, que foi observada apenas neste conjunto específico de 4 datasets e ainda depende da faixa de épocas que estamos analisando. Por exemplo, em 5, o método analisado apresenta uma diferença positiva para o dataset de validação apenas nas últimas épocas analisadas.

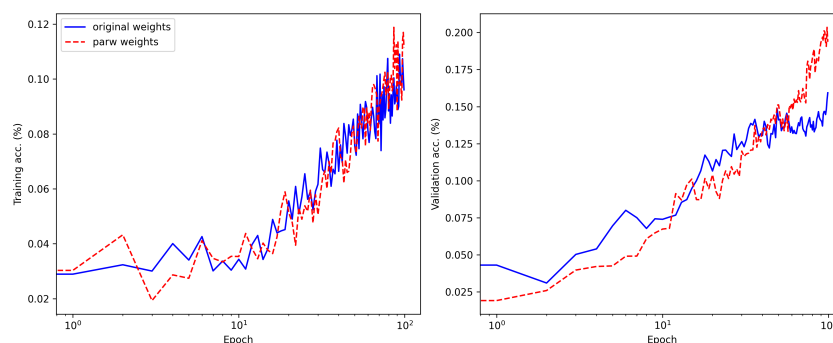


Figura 5. Resultados de acurácia de treinamento e validação da arquitetura MLP aplicada ao dataset Adiac.

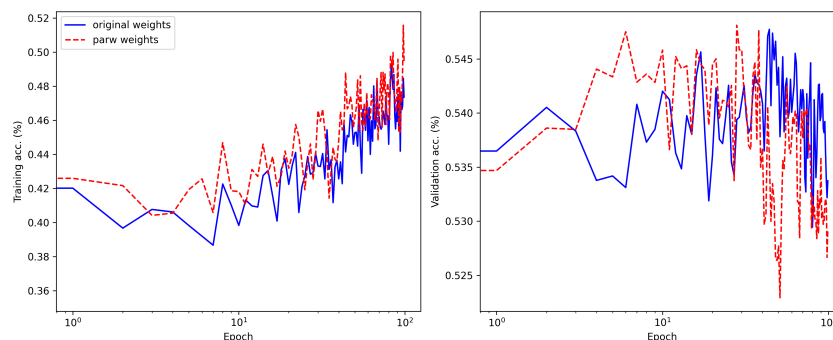


Figura 6. Resultados de acurácia de treinamento e validação da arquitetura MLP aplicada ao dataset ChlorineConcentration

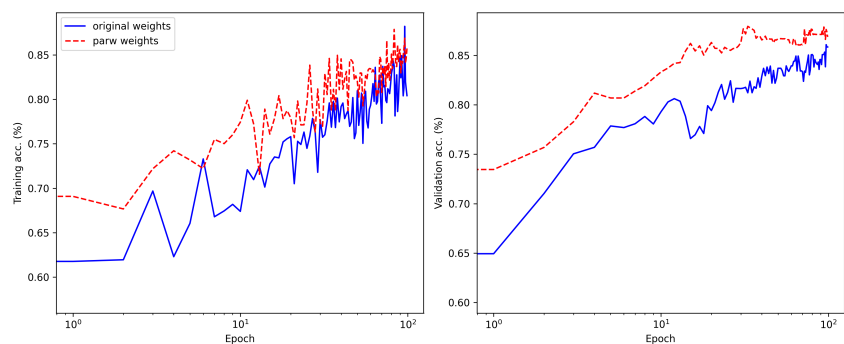


Figura 7. Resultados de acurácia de treinamento e validação da arquitetura MLP aplicada ao dataset ECG200

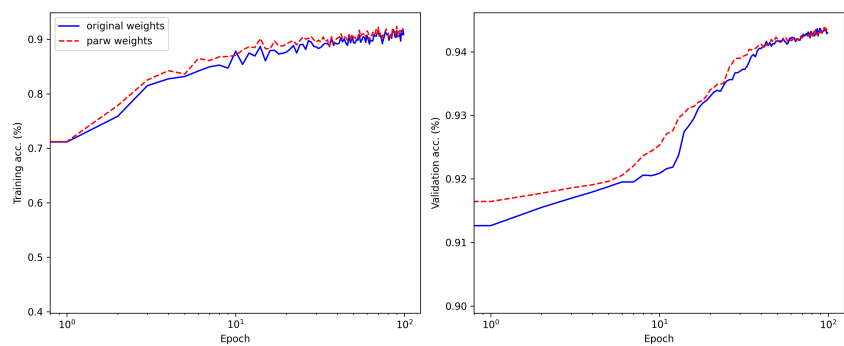


Figura 8. Resultados de acurácia de treinamento e validação da arquitetura MLP aplicada ao dataset ECG5000

Em seguida, foram realizados os testes para a arquitetura FCN, com os resultados apresentados na tabela da figura 9. Para essa arquitetura, os resultados foram mais inconclusivos, devido aos elevados erros observados durante os testes.

FCN				
Dataset	Peso	ep. 1 acc Tre.	ep. 1 acc Val.	Teste Avr.
Adiac	Normal	3,24 +/-0,8	4,08+/-1,77	63,79+/-9,33
	PARw	3,47+/-1,06	4,03+/-1,87	62,61+/-9,91
Chlorine	Normal	54,11+/-1,59	54,08+/-1,98	46,08+/-14,23
	PARw	54,30+/-1,23	54,14+/-2,03	49,47+/-13,60
ECG200	Normal	69,16+/-5,70	62,34+/-17,39	84,12+/-8,77
	PARw	71,39+/-4,52	68,84+/-13,03	85,90+/-6,23
ECG5000	Normal	86,98+/-1,77	88,98+/-4,07	93,42+/-2,36
	PARw	88,00+/-0,84	88,45+/-5,05	93,29+/-2,59

Figura 9. Os dados apresentados referem-se ao treinamento e teste da arquitetura FCN aplicada aos datasets selecionados.

Os resultados das curvas de acurácia também foram menos distintos quando comparados a arquitetura MLP.

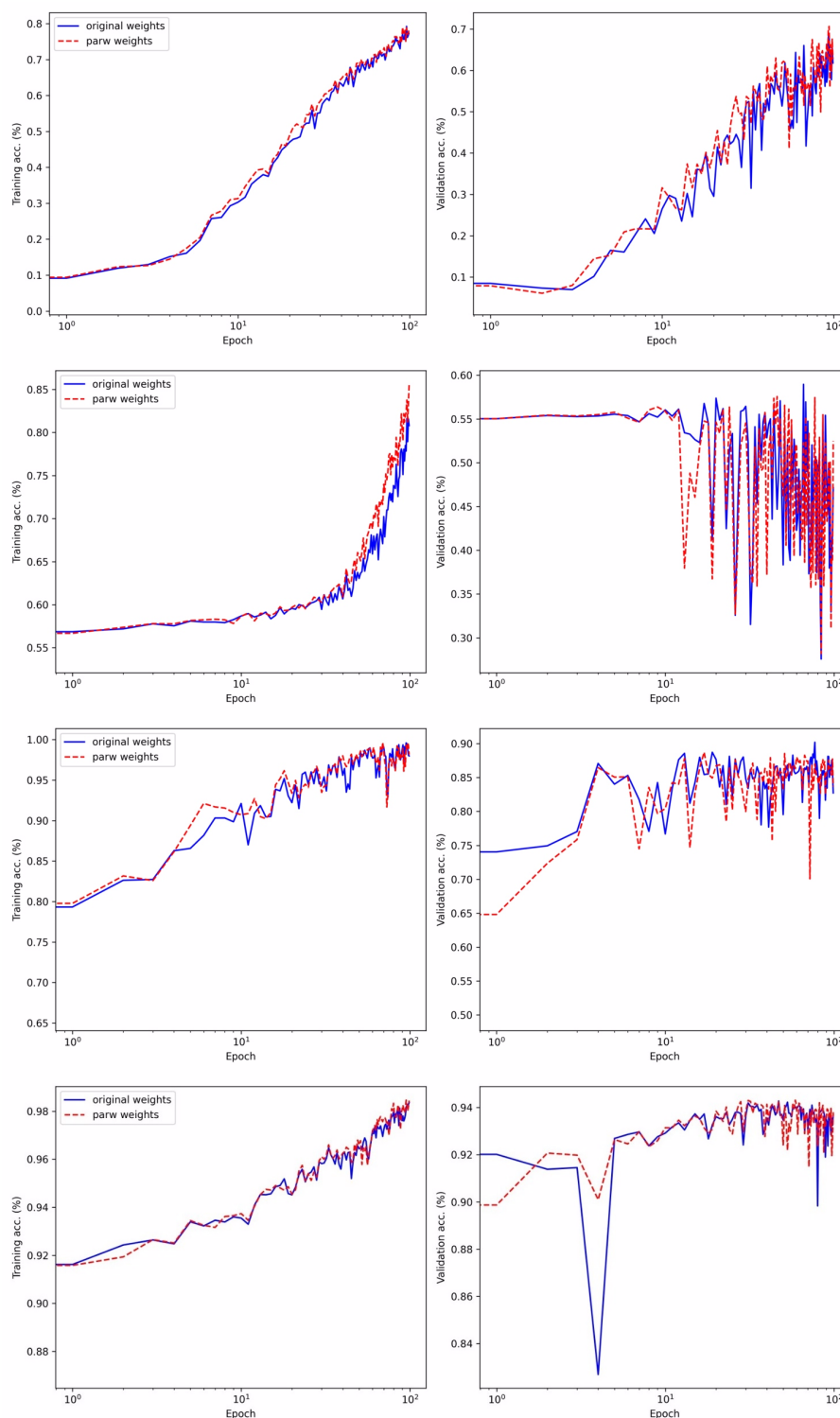


Figura 10. Curvas de acurácia dos datasets Adiac, Chlorine, ECG200 e ECG5000 respectivamente

6. Conclusão

Observa-se que, embora o MLP tenha apresentado um desempenho superior à FCN na comparação com a inicialização padrão dos pesos da rede, não podemos afirmar

que o bom desempenho observado nesse método no domínio de imagens seja, de fato, replicável no domínio de séries temporais.

Devido às limitações computacionais, os testes foram realizados em apenas 4 datasets. Para uma avaliação mais robusta, um estudo futuro poderia expandir a análise para todos os datasets do UCR Time Series Classification Archive e realizar todas as avaliações estatísticas pertinentes, a fim de concluir se o método PARw realmente oferece desempenho superior em comparação com a inicialização padrão.

7. Repositório

Todos os códigos utilizados na realização deste trabalho podem ser encontrados no seguinte repositório: https://github.com/gildex174/conexao-preferencial_series_temporais.

Referências

- [Barabási and Pósfai 2016] Barabási, A.-L. and Pósfai, M. (2016). *Network science*. Cambridge University Press, Cambridge. Section 5.2.
- [Dau et al. 2018] Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., and Hexagon-ML (2018). The ucr time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- [Scabini et al. 2023] Scabini, L. F. d. S., Baets, B. D., and Bruno, O. M. (2023). *Patterns and randomness in networks for computer vision: from graphs to neural networks*. PhD thesis, Universidade de São Paulo.
- [Wang et al. 2016] Wang, Z., Yan, W., and Oates, T. (2016). Time series classification from scratch with deep neural networks: A strong baseline.