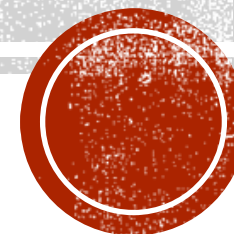


# REDES NEURAIS ARTIFICIAIS

## WEBCONFERÊNCIA 1

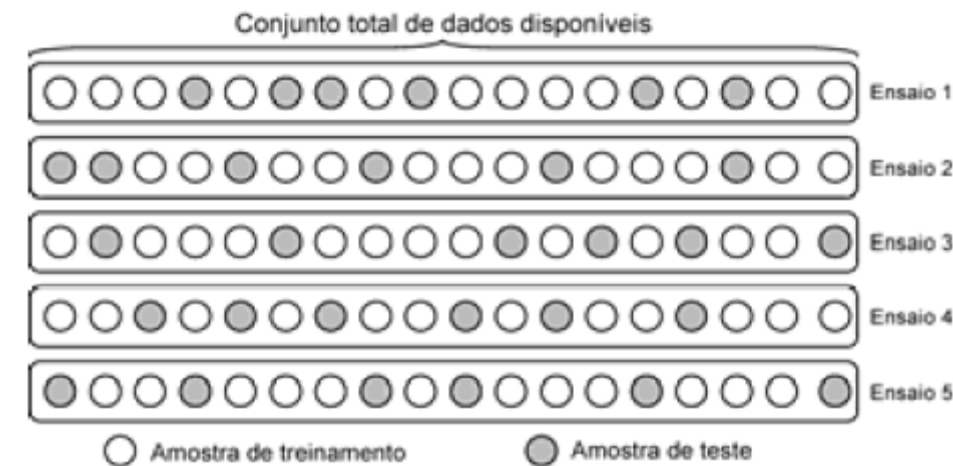
Prof. Rodrigo Palácios  
[rodrigopalacios@utfpr.edu.br](mailto:rodrigopalacios@utfpr.edu.br)



# TÉCNICAS DE VALIDAÇÃO CRUZADA

## ■ Princípios da validação cruzada (amostragem aleatória)

- O conjunto total de dados (amostras) disponíveis é aleatoriamente dividido em duas partes, isto é, subconjunto de treinamento e subconjunto de teste (validação).
  - **Subconjunto de treinamento** → utilizado para treinar todas as topologias candidatas.
  - **Subconjunto de teste** → utilizado para selecionar aquela que estará apresentando os melhores resultados de generalização.
- As amostras do subconjunto de teste não participaram do treinamento, o que possibilita avaliar o desempenho da generalização proporcionada em cada uma das topologias candidatas.
- Para tanto, basta-se comparar os resultados produzidos em suas saídas frente aos respectivos valores desejados.
- A partir do conjunto total de amostras, cerca de 60 a 90% delas são aleatoriamente escolhidas para o subconjunto de treinamento, enquanto o restante ficará alocado ao subconjunto de teste.
- Esta sistemática de partição é repetida várias vezes durante o aprendizado das topologias candidatas, permitindo-se (em cada ensaio) a possibilidade de contemplação de amostras diferentes tanto no subconjunto de treinamento como naquele de teste.
- O desempenho global de cada topologia candidata será então compilado a partir da média dos desempenhos individuais em cada experimento.



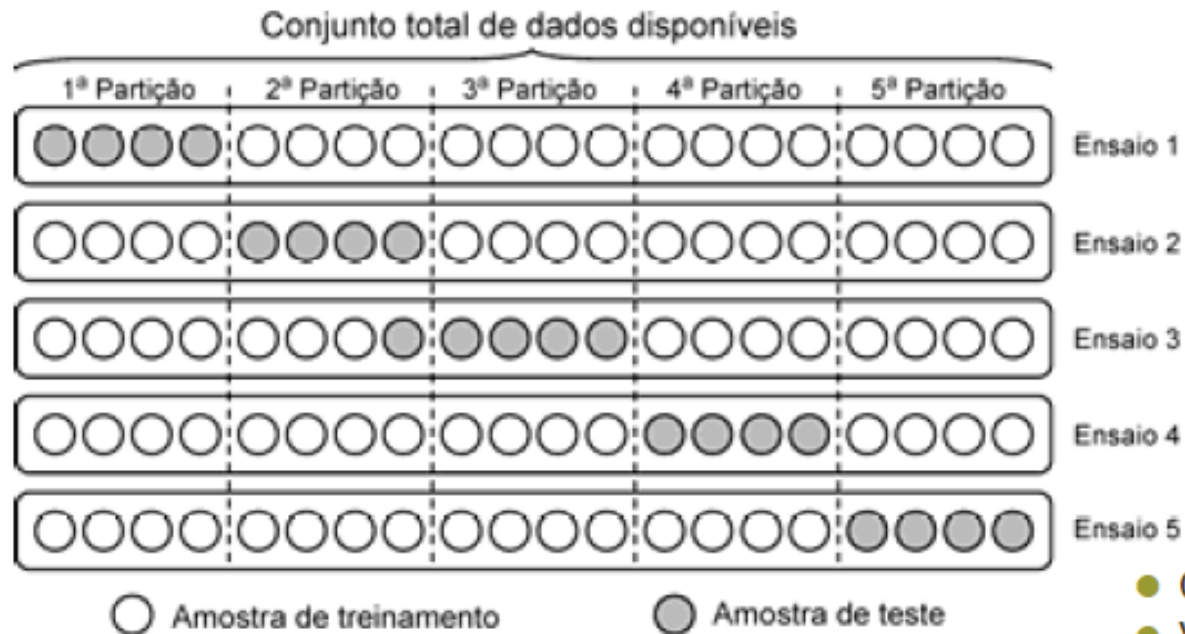
Conjunto total de amostras → 18  
Conjunto de treinamento → 12  
Conjunto de teste → 6



# TÉCNICAS DE VALIDAÇÃO CRUZADA

## ■ Princípios da validação cruzada ( $k$ -partições)

- Realiza-se aqui a divisão do conjunto total de amostras em  $k$  partições, sendo que  $(k-1)$  delas serão usadas para compor o subconjunto de treinamento, ao passo que a partição restante constituirá o subconjunto de teste.
- Por conseguinte, o processo de aprendizado se repete  $k$  vezes até que todas as partições tenham sido utilizadas como subconjunto de teste.
- O valor do parâmetro  $k$  está atrelado à quantidade total de amostras disponíveis, sendo usualmente atribuído um número compreendido entre 5 e 10.
- O desempenho global de cada topologia candidata será agora também obtido em função da média entre os desempenhos individuais observados quando da aplicação das  $k$  partições.



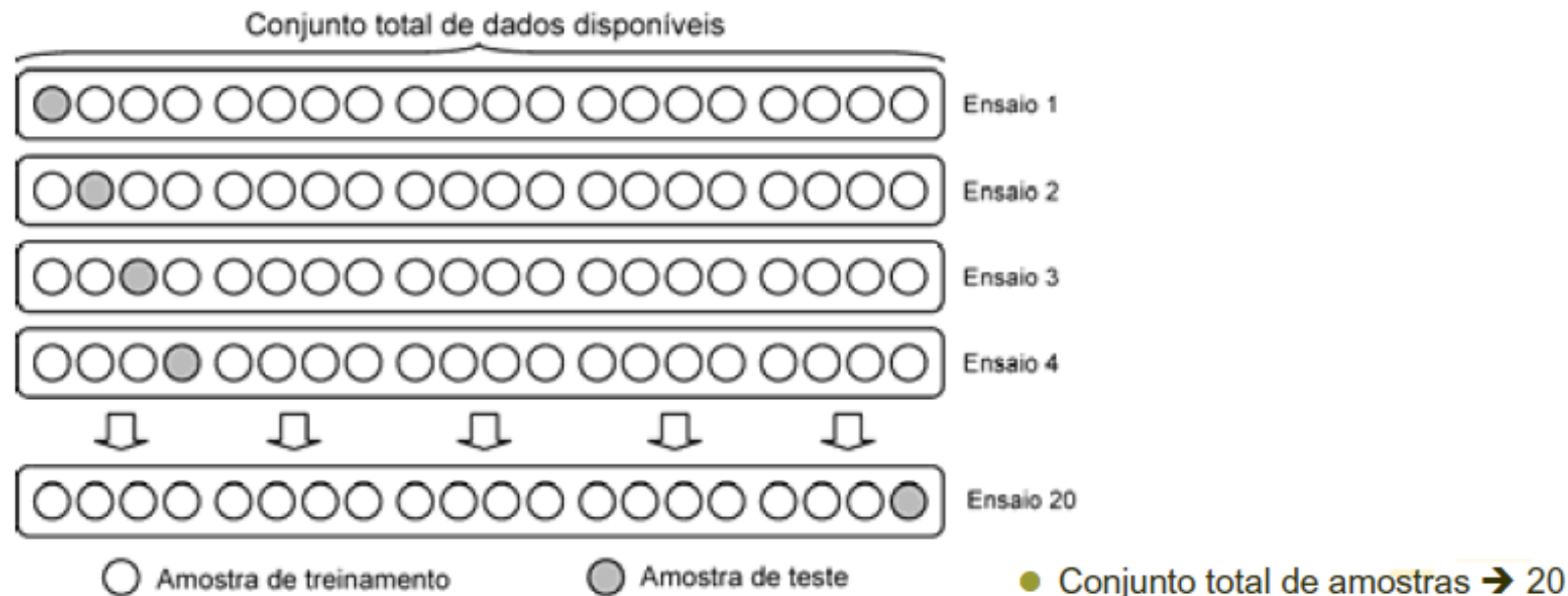
- Conjunto total de amostras → 20
- Valor do parâmetro  $k$  → 5



# TÉCNICAS DE VALIDAÇÃO CRUZADA

## ■ Princípios da validação cruzada (por unidade)

- Consiste da utilização de uma única amostra para o subconjunto de teste, sendo todas as demais alocadas para o subconjunto de treinamento.
- O processo de aprendizado é então repetido até que todas as amostras sejam individualmente utilizadas como subconjunto de teste.
- Esta técnica acaba sendo um caso particular do método de  $k$ -partições, pois se basta atribuir ao parâmetro  $k$  o valor que corresponde ao número total de amostras disponíveis.
- Contudo, tem-se aqui um elevado esforço computacional, pois o processo de aprendizagem será repetido, considerando cada uma das topologias candidatas, um número de vezes que será igual ao tamanho do conjunto total de amostras.



# TÉCNICAS DE VALIDAÇÃO CRUZADA

- Aplicação prática usando o Sklearn (Python)
  - Link para documentação oficial: [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html)



# TÉCNICAS DE VALIDAÇÃO CRUZADA

- Pipeline Sklearn (Python)

- <https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html>
- *from sklearn.pipeline import Pipeline*
- A classe Pipeline é uma funcionalidade do Scikit-Learn para geração de códigos que possuam um padrão de execução.
- Toda vez que o modelo é treinado, todos os passos definidos no Pipeline são executados na ordem em que são definidos, com o output de um passo servindo como input para outro. O Pipeline também possui o método *fit* e todos os passos definidos são executados na ordem em que aparecem.
- Uma vez que encapsulado o modelo dentro de um Pipeline, fica mais fácil validar. Com o Pipeline o modelo não consiste apenas do algoritmo, mas também de todo o pré-processamento usado antes de aplicar o algoritmo nos dados.



# REFERÊNCIA

- SILVA, Ivan Nunes da e SPATTI, Danilo Hernane e FLAUZINO, Rogério Andrade. Redes neurais artificiais para engenharia e ciências aplicadas. . São Paulo: Artliber Editora. . Acesso em: 23 dez. 2023. , 2010