

**INSTITUTO FEDERAL**

Mato Grosso do Sul

**Minicurso: Construindo uma rede neural para previsão de séries temporais usando Keras.**

Valdemir Chaves Ribeiro

Engenharia da Computação

## O que é uma rede neural?

Uma rede neural é um modelo computacional que simula o comportamento do cérebro humano. Onde se tem a entrada dos dados (Input Layer), o processamentos dos dados (Hidden Layers) e a saída dos dados (Output Layer).

## Como a rede neural aprende?

A rede neural aprende através de treinamentos, com cada iteração entre as funções de ativação e de perda, ela vai adaptando à saída para buscar ser a mais precisa de acordo com a configuração, atribuindo pesos para cada uma das variáveis de entrada, na tentativa de ser assertiva de acordo com os dados de entrada.

## O que é necessário para gerar previsões?

A rede precisa de duas informações para treinar: a entrada (também chamada de dado de treinamento) e a saída esperada, comumente conhecida como alvo (ou target). Durante o processo de treinamento, a rede compara suas previsões (saída da rede) com os valores reais do alvo. A diferença entre as previsões e os valores reais é quantificada por meio de uma função de perda (ou função de custo), que representa o quão distantes as previsões da rede estão do resultado desejado.

## Quais são os tipos de redes neurais?

- Redes Neurais de Alimentação Direta (FNN): As informações se movem em uma única direção, da camada de entrada para a camada de saída, sem ciclos.
- Redes Neurais Recorrentes (RNNs): Permitem conexões cíclicas, o que as torna adequadas para lidar com sequências de dados, como séries temporais ou processamento de linguagem natural.
- Redes Neurais Convolucionais (CNNs): Projetadas para processar dados de grade, como imagens. São eficazes em identificar padrões espaciais.

## Os principais elementos de uma rede neural.

- A forma como os dados são inseridos na rede.
- Funções de ativação de cada camada.
- Função de perda da rede.
- Quantidade de neurônios por camada.
- O Tipo de cada camada.

## O que é uma função de ativação?

Uma função de ativação em uma camada de uma rede neural é uma função matemática aplicada a cada saída (ou ativação) de um neurônio nessa camada. Essa função é crucial para introduzir não-linearidades no modelo, permitindo que a rede neural aprenda e represente relações mais complexas nos dados.

Sem uma função de ativação não-linear, uma rede neural seria essencialmente equivalente a uma combinação linear de suas entradas. Isso significa que, mesmo com várias camadas, a rede seria equivalente a uma única camada, limitando sua capacidade de aprender padrões complexos.

# Funções de ativação

1. **Sigmoide (Sigmoid):** `'sigmoid'`
  - Usada comumente em camadas de saída de redes neurais binárias para representar probabilidades.
2. **Tangente Hiperbólica (Hyperbolic Tangent - Tanh):** `'tanh'`
  - Produz saídas no intervalo de -1 a 1.
  - Similar à sigmoide, mas com um intervalo simétrico.
3. **Unidade Linear Retificada (Rectified Linear Unit - ReLU):** `'relu'`
  - Retorna zero para valores negativos e linear para valores positivos.
  - É uma escolha popular devido à sua simplicidade e eficácia.
  - Comumente usada para previsão de séries temporais.
4. **Leaky ReLU:** `'leaky_relu'`
  - Semelhante ao ReLU, mas permite um pequeno gradiente para valores negativos, evitando alguns problemas associados ao ReLU.
5. **Linear:** `'linear'`
  - Realiza uma operação de multiplicação entre a entrada e um coeficiente (peso).
  - Se houver múltiplas entradas, a função linear seria a soma ponderada dessas entradas



## O que é uma função de perda?

A função de perda, também conhecida como função objetivo ou função de custo, é uma medida que quantifica quão bem o modelo de uma rede neural está realizando uma tarefa específica. Ela compara as previsões feitas pela rede com as saídas desejadas (rótulos) associadas aos dados de treinamento. O objetivo durante o treinamento é minimizar o valor da função de perda, indicando que as previsões da rede estão se aproximando mais das saídas desejadas.

A escolha da função de perda depende da natureza da tarefa que a rede neural está realizando.

## Funções de perda

- **Erro Quadrático Médio (Mean Squared Error):** `'mean_squared_error'`
  - Usado para tarefas de regressão.
  - Mede a média dos quadrados das diferenças entre previsões e rótulos reais.
- **Entropia Cruzada Binária (Binary Cross-Entropy):** `'binary_crossentropy'`
  - Usado para tarefas de classificação binária.
  - Mede a entropia cruzada entre distribuições de probabilidade binárias.
- **Erro Absoluto Médio (Mean Absolute Error):** `'mean_absolute_error'`
  - Usado para tarefas de regressão.
  - Mede a média das diferenças absolutas entre previsões e rótulos reais.
- **Erro de Huber (Huber Loss):** `'huber_loss'`
  - Semelhante ao erro absoluto médio, mas menos sensível a outliers.
  - Pode ser uma escolha robusta para tarefas de regressão.
- **Erro Log-Cosh (Logarithm of the Hyperbolic Cosine):** `'logcosh'`
  - Outra alternativa ao erro absoluto médio que é menos sensível a outliers.

# Tipos de camadas.

- **Dense (Densa):** 'Dense'
  - Camada totalmente conectada.
  - Todos os neurônios na camada estão conectados a todos os neurônios na camada seguinte.
  - Usada em tarefas de aprendizado profundo, incluindo classificação e regressão.
- **LSTM (Long Short-Term Memory):** 'LSTM'
  - Camada de rede recorrente que lida bem com sequências temporais longas.
  - Ajuda a superar o problema do desvanecimento do gradiente em redes recorrentes tradicionais.
  - Usada em tarefas como processamento de linguagem natural e séries temporais.
- **Conv2D (Convolutacional 2D):** 'Conv2D'
  - Camada de convolução para processamento de dados 2D, como imagens.
  - Utiliza filtros para extrair características relevantes.
  - Usada em redes neurais convolucionais (CNNs) para tarefas de visão computacional.
- **MaxPooling2D (Aglutinação Máxima 2D):** 'MaxPooling2D'
  - Camada de aglutinação máxima para reduzir a dimensionalidade espacial.
  - Reduz o número de parâmetros e operações na rede.
  - Usada em conjunto com camadas convolucionais em CNNs.

# Como eu faço a validação dos dados e confiro se a rede está errando ou acertando?

Existem métricas de validação para poder mensurar o quanto uma rede está distante ou próximo do valor esperado.

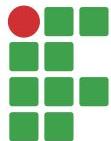
- **Mean Squared Error (MSE - Erro Quadrático Médio):**
  - Calcula a média dos quadrados das diferenças entre os valores preditos ( $\hat{y}_i$ ) e os valores reais ( $y_i$ ).
  - Sensível a outliers, pois penaliza grandes erros.
- **R-squared ( $R^2$  - Coeficiente de Determinação):**
  - Mede a proporção da variabilidade nos dados de resposta que é explicada pelo modelo.
  - Valores mais próximos de 1 indicam um modelo melhor.
- **Mean Absolute Error (MAE - Erro Absoluto Médio):**
  - Calcula a média das diferenças absolutas entre os valores preditos e os valores reais.
  - Menos sensível a outliers do que o MSE.
- **Root Mean Squared Error (RMSE - Raiz do Erro Quadrático Médio):**
  - É a raiz quadrada do MSE, fornecendo uma métrica em escala semelhante aos valores reais.
  - Como o MSE, penaliza erros maiores de forma mais significativa.

## E como eu encontro a melhor combinação para minha rede?

Executando um Grid Search, que basicamente seria uma forma de testar quais são os parâmetros que têm os melhores resultados para os dados, a técnica consiste em testar as diferentes combinações possíveis entre as funções escolhidas, onde o número total de combinações seria o produto das opções de cada hiperparâmetro. Esse procedimento pode levar **MUITO TEMPO**, particularmente recomendo utilização da biblioteca Optuna.

**Vamos a prática.**

<https://github.com/RiverCarnival/Minicurso-Keras-Magiclab>



**INSTITUTO FEDERAL**

Mato Grosso do Sul