Projeto 4 - Redes Recorrentes

Guilherme Pereira Campos RA:163787
Lucas Oliveira Nery de Araújo RA:158882
Universidade Federal de São Paulo

I. Resumo

Este projeto teve como objetivo treinar modelos *LSTM* para classificar séries temporais, onde o modelo recebe uma janela de dados e identifica sua categoria com base nos padrões observados. O objetivo escolhido para o projeto será classificar um segmento de ECG em categorias como **normal**, **arritmia** e outras.
Foram exploradas diferentes configurações de redes, ajustando a topologia das camadas *LSTM*, o número de neurônios e os mecanismos de regularização para otimizar a identificação de padrões temporais relevantes. O projeto demonstra a eficácia das redes *LSTM* na extração e análise de informações dinâmicas presentes em séries temporais.

II. Dataset

O **MIT-BIH Arrhythmia** é um dataset de série temporal contendo gravações de ECG de pacientes com diferentes tipos de arritmias. A tarefa é classificar os batimentos cardíacos com base nos sinais ao longo do tempo. Cada amostra consiste em segmentos de ECG rotulados, representando diferentes classes de batimentos. A ordem dos registros é essencial.
Foi feita a seguinte preparação nos dados:

1. Carregamento do Dataset

- O arquivo **ECG (mit-bih-arrhythmia-database-1.0.0)** foi carregado a partir do diretório do projeto utilizando a função `wfdb.rdrecord()` da biblioteca *wfdb*.
- O dataset contém **sinais de ECG** com informações de **tempo e rótulos de classes** que indicam tipos de arritmia ou a ausência delas.

2. Separação dos Conjuntos

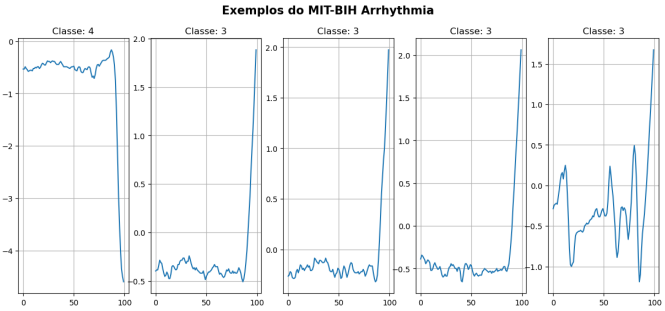
- Os dados foram divididos em **treinamento (80%)** e **teste (20%)** utilizando a função `train_test_split()` do Scikit-learn.
- Distribuição final dos dados:
  - **Treino:** 8.000 amostras
  - **Teste:** 2.000 amostras

3. Normalização e Formatação dos Dados

- Os sinais de ECG foram convertidos para **float32** e normalizados, garantindo que os valores numéricos estivessem no formato adequado para o treinamento do modelo.
- A variável alvo (tipo de arritmia) foi convertida para **representação categórica** (classes conforme o sinal de ECG).
- Os dados de entrada foram remodelados para o formato adequado ao modelo **LSTM**, com uma **dimensão adicional** para representar as séries temporais.

4. Visualização dos Dados

- Foram exibidos **5 exemplos do conjunto de treino** com seus respectivos rótulos (tipos de arritmia ou ausência), permitindo uma verificação visual dos padrões nas séries temporais de ECG antes do treinamento.



III. Redes Recorrentes Long Short-Term Memory

Definição

- A informação pode atingir um neurônio mais de uma vez.
- Característica principal: **memória** e *ordem temporal*.

Funcionamento

- O novo estado da rede depende tanto da entrada quanto do estado atual, ou seja, da memória associada ao estado interno (oculto) do modelo.
- As RNNs podem recordar características importantes dos sinais anteriores.

Rede Recorrente Padrão

- A saída h_t depende da entrada x_t e do próprio estado anterior da rede:

$$A_t = f_0(A_{t-1}, x_t)$$

$$h_t = g_0(A_t)$$

- Além da dependência espacial (fluxo camada a camada), as RNNs possuem dependência temporal do sinal.

Retropropagação

- Para calcular o gradiente de um neurônio, é necessário conhecer:
  - Os gradientes dos neurônios da camada posterior (dependência espacial, por exemplo, do neurônio de saída y).
  - Os gradientes dos neurônios da própria camada no instante posterior (dependência temporal).
- As equações envolvidas são:

$$\alpha^t = \phi(w_x x^t + w_a a^{t-1})$$

$$y^t = \phi(w_y a^t)$$

$$\delta_j = f'(v_j) \sum_k \delta_k w_{kj}$$

- O processo de atualização é idêntico ao utilizado na retropropagação tradicional aplicada em redes MLP.
- Basicamente, o algoritmo é o mesmo, mudando apenas a forma de considerar as dependências.

Problemas com o Gradiente

- **Explosão do gradiente:** Dependendo da função de ativação, o valor do gradiente pode se tornar muito elevado, saturando (estourando) as sinapses. Uma abordagem simples é estabelecer um corte, fixando o valor máximo do gradiente.
- **Desaparecimento do gradiente (vanishing):** O gradiente pode se tornar ínfimo, impossibilitando o ajuste adequado dos pesos.
- Redes recorrentes padrão (como as unidades MCP e suas variações) podem sofrer desses problemas.

Long Short-Term Memory (LSTM)

- Uma célula LSTM é composta por três gates: de entrada, de saída e de esquecimento (*forget*).
- A célula LSTM possui dois sinais de memória:

  c_t (memória de longo prazo, sinal interno)

  a_t (memória de curto prazo, sinal externo)

- Os gates são:
  - **Gate de entrada (ou de atualização):** U
  - **Gate de saída:** O
  - **Gate de esquecimento:** F
- A célula é capaz de recordar sinais arbitrários do passado a partir da configuração dos gates, que controlam o fluxo de informação.
- Teoricamente, sinais podem ser mantidos por longos períodos.

IV. Modelos

A biblioteca utilizada para implementar o VAE (Variational Autoencoder) neste projeto será o *TensorFlow* (https://www.tensorflow.org), um framework amplamente utilizado para aprendizado profundo.

A. Primeiro modelo LSTM

O primeiro modelo *LSTM* foi configurado para processar sequências de sinais de ECG com 100 timesteps e aprender padrões temporais associados às diferentes classes de arritmias. A arquitetura inclui três camadas *LSTM* com 64, 32 e 16 neurônios, respectivamente, todas utilizando a ativação *tanh*.

Para evitar *overfitting* e melhorar a generalização, cada camada *LSTM* é seguida por uma camada *Dropout* com taxa de 20%. A última camada é totalmente conectada (*Dense*) com ativação *softmax*, responsável por classificar os sinais em suas respectivas categorias.

O modelo será treinado por 20 épocas utilizando o otimizador *Adam* e a função de perda *categorical crossentropy*.

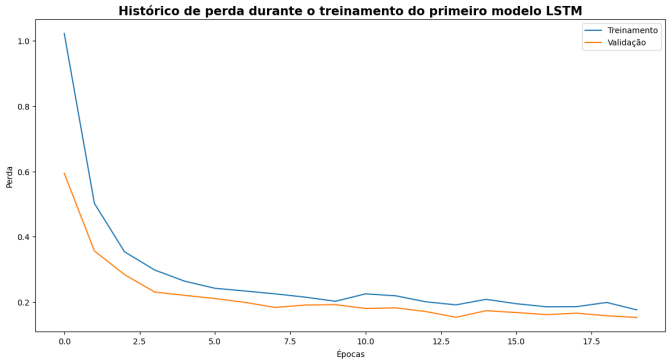
A figura abaixo conta com o *plot_model*, que permite visualizar a arquitetura do modelo de forma gráfica, exibindo a conexão entre as camadas.

1) Histórico de perda durante o treinamento: A perda durante o treinamento e validação do modelo mede o quão bem as previsões se ajustam aos dados reais, sendo calculada pela função de custo definida na compilação. No caso deste modelo *LSTM*, a perda utilizada é a *categorical crossentropy*, que mede a discrepância entre as distribuições de probabilidade predita e real. Essa função é comumente usada em problemas de classificação multiclasse e é definida pela equação:

$$H(p, q) = -\sum_{i=1}^{n} p_i \log q_i$$

onde p_i é a distribuição real das classes e q_i é a distribuição predita pelo modelo.

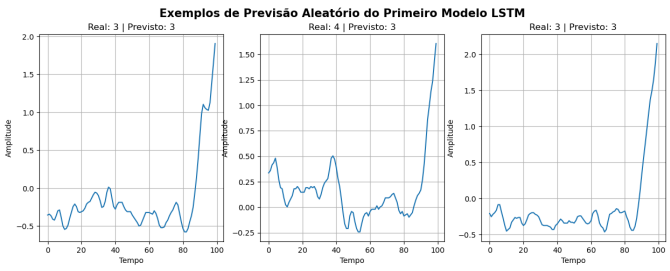
Durante o treinamento, espera-se que essa perda diminua, indicando que o modelo está aprendendo a classificar corretamente os padrões do ECG. Além disso, a métrica de acurácia acompanha o percentual de acertos. Se a perda de validação permanecer próxima à de treinamento, significa que o modelo generaliza bem sem *overfitting*.



Apesar da configuração relativamente simples, o primeiro modelo *LSTM* apresentou um desempenho satisfatório na classificação dos sinais de ECG. Com apenas três camadas *LSTM* e um total de 112 neurônios, o modelo conseguiu aprender padrões temporais relevantes e distinguir diferentes classes de arritmias de forma eficaz.

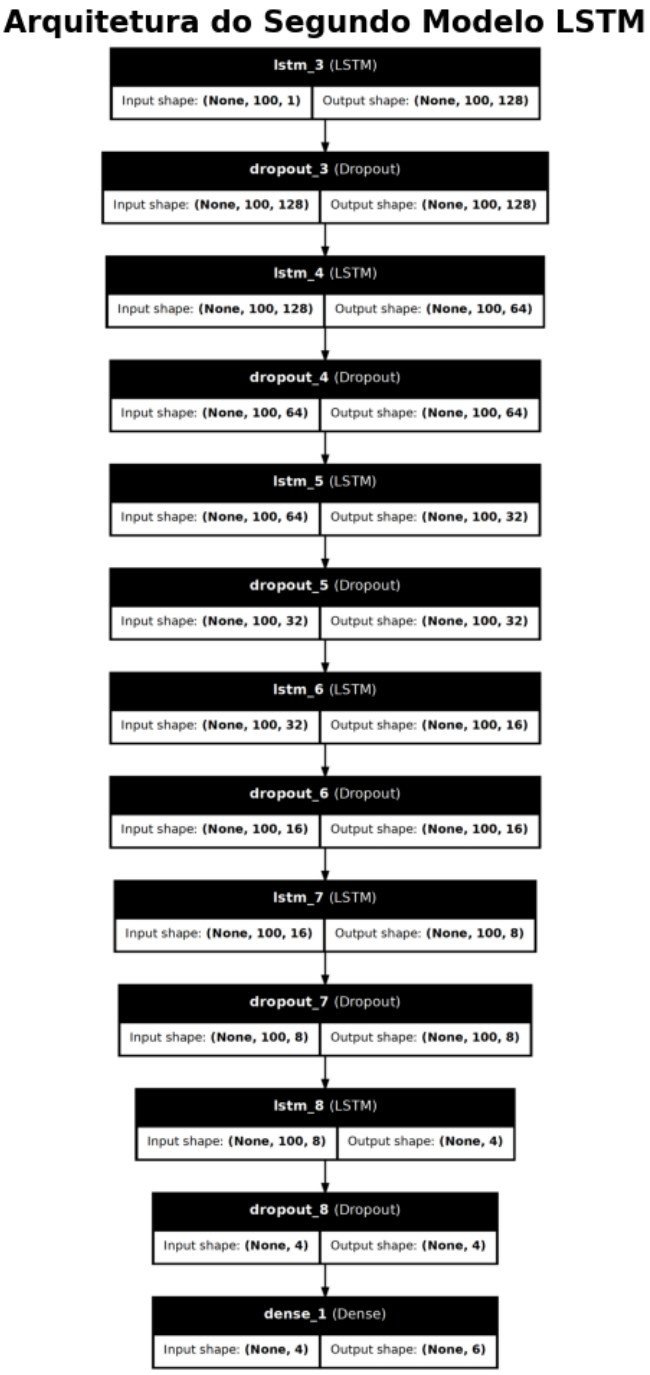
A utilização da ativação *tanh* nas camadas *LSTM*, aliada à regularização via *Dropout*, contribuiu para um treinamento estável e uma boa generalização. A acurácia obtida nos dados demonstra que, mesmo com uma arquitetura enxuta, o modelo foi capaz de capturar informações essenciais do sinal e realizar previsões coerentes com os rótulos verdadeiros, atingindo uma acurácia de **0.9623**.

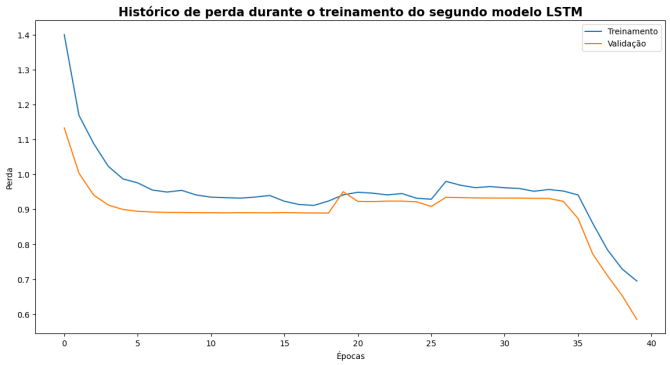
Foram realizados exemplos de previsão aleatórios, onde foram selecionados 3 exemplos do conjunto de teste e plotados os sinais de *ECG* comparando o rótulo real com a previsão do modelo, evidenciando a capacidade do sistema em identificar corretamente as classes.



B. Segundo modelo LSTM

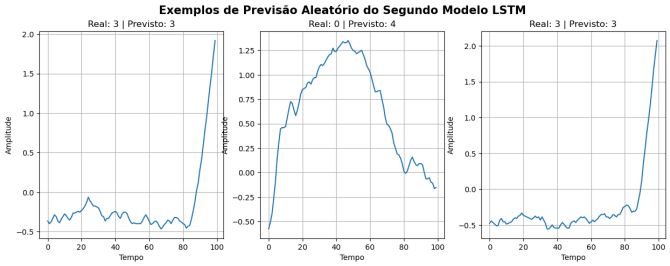
O segundo modelo *LSTM* foi configurado para processar sequências de sinais de ECG com 100 timesteps, ampliando a capacidade de aprendizado e generalização. Em vez de três camadas, essa arquitetura possui o dobro de camadas e neurônios, sendo composta por seis camadas *LSTM* com 128, 64, 32, 16, 8 e 4 neurônios, respectivamente, todas utilizando a ativação *tanh*. Para evitar *overfitting* e melhorar a generalização, cada camada *LSTM* é seguida por uma camada *Dropout* com taxa de 40% (o dobro dos 20% do primeiro modelo). A última camada é uma camada *Dense* com ativação *softmax*, responsável por classificar os sinais em suas respectivas categorias. O modelo será treinado por 40 épocas utilizando o otimizador *Adam* e a função de perda *categorical_crossentropy*.





Apesar do aumento na complexidade, o segundo modelo *LSTM*, com seis camadas e um maior número total de neurônios, apresentou desempenho inferior na classificação dos sinais de ECG. Embora a arquitetura tenha sido projetada para capturar padrões temporais mais sutis, o aumento do *dropout* e da profundidade da rede pode ter dificultado a generalização, resultando em treinamento instável e menor acurácia. Esses resultados sugerem que, para esse problema específico, uma configuração mais simples pode ser mais eficaz na extração de informações essenciais do sinal.

A acurácia obtida pelo segundo modelo foi de **0.87**. Além disso, foram realizados exemplos de previsão aleatória, nos quais foram selecionados 3 segmentos do conjunto de teste e comparados os rótulos reais com as previsões do modelo, evidenciando a capacidade, embora reduzida, do sistema em classificar os sinais de ECG.



V. Conclusão

Este projeto demonstrou a eficácia das redes *LSTM* na classificação de séries temporais, especificamente sinais de ECG. Foram testadas diferentes arquiteturas, variando a profundidade, o número de neurônios e a regularização.

Os resultados indicaram que a configuração mais simples obteve melhor desempenho, com acurácia superior e maior estabilidade no treinamento. A análise de previsões aleatórias reforçou essa conclusão, evidenciando que modelos mais profundos nem sempre resultam em melhor generalização.