

# Engenharia de Redes Neurais Artificiais

Aluno: Matheus Paul Lopuch

Exercícios das Aulas

Aula 04\_Linguagem\_Natural

Histórico de Redes Neurais para Processamento de Linguagem Natural

Exercícios:

1. Verifique os exemplos da aula 4 para série temporal e processamento de linguagem natural

Responda as seguintes perguntas:

2. Porque a LSTM foi mais eficiente que a MLP?
3. Qual a grande desvantagem de Transformers?
4. Altere a LSTM por uma GRU no exemplo 4.a. Qual foi o resultado?

Respostas:

1. O material fornece exemplos de redes neurais aplicadas a dados sequenciais, destacando RNNs, LSTM, GRU, BiLSTM, BiGRU e Transformers.
2. LSTM é adaptada para dados sequenciais, possui memória de longo prazo controlada por portas, capturando dependências temporais e contextuais. Já a MLP (Multilayer Perceptron) trata as entradas de modo independente, sem considerar ordem ou contexto, perdendo informações relevantes ao processar sequências, como textos ou séries temporais. Assim, em NLP e séries temporais, a LSTM supera a MLP devido à sua capacidade de representar e manipular dependências de longo prazo dentro dos dados.
3. A principal desvantagem dos Transformers está no custo computacional elevado, especialmente devido ao mecanismo de auto-atenção, cujo uso de memória e processamento cresce quadraticamente com o tamanho da sequência de entrada. Isso torna Transformers exigentes em hardware, limitando seu uso em aplicações com grandes sequências ou recursos restritos.

4. Ao trocar a LSTM pela GRU no exemplo 4.a, o resultado normalmente é similar: A GRU é mais leve, possui menos parâmetros e pode treinar mais rápido, mantendo desempenho equivalente ao da LSTM na maioria das tarefas.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense

model = Sequential()
model.add(GRU(units=64, input_shape=(timesteps, features)))
model.add(Dense(NUM_CLASSES, activation='softmax'))
model.compile(optimizer='adam',
              loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(x_train, y_train, epochs=30,
          validation_data=(x_val, y_val))
```

Parte que substitui no código.

Com a GRU, espere resultados muito próximos aos da LSTM, porém com treinamento mais rápido e uso de menos recursos computacionais.