



MATERIAL COMPLEMENTAR

Deep Learning II

Professor Jeronimo Marcondes Pinto

SUGESTÃO DE LEITURA

Material disponível na internet:

- Oliveira, E.V. 2020. **Análise de desempenho de método baseado em rede LSTM para classificação de falhas em um processo de controle de nível.** [Link](#)
 - Castelhão, R. 2018. **Utilização de Redes Neurais para Previsões no Mercado de Ações.** [Link](#)
 - Yan, S. 2016. **Understanding LSTM and its diagrams.** [Link](#)
-

1. Deep Learning

- **Aplicações em Dados Sequenciais:** Deep Learning pode ser usado para resolver problemas com dados sequenciais, como análise de texto e séries temporais. Nesses casos, a ordem dos dados importa, diferentemente de outras abordagens que tratam cada entrada como independente.
- **Limitações das Redes Neurais Feedforward (FNN):** As FNNs têm dificuldades em capturar relações sequenciais, pois consideram as entradas independentes. Isso torna as RNNs uma escolha melhor para capturar dependências temporais e contextuais nos dados.

2. RNN (Redes Neurais Recorrentes)

- **Estrutura das RNNs:** As RNNs possuem uma estrutura que "lembra" os estados anteriores através de ciclos de feedback. Cada nó nas camadas escondidas conecta-se a si mesmo em cada passo temporal, permitindo que a rede aprenda dependências entre as etapas da sequência.
- **Funcionamento:** A cada passo, a entrada atual e a saída do estado anterior são usadas para calcular a próxima saída. Assim, a saída da rede em um dado momento depende não só da entrada atual, mas também das entradas anteriores.
- **Problema do Gradiente:** Ao treinar uma RNN, o algoritmo Backpropagation Through Time (BPTT) é usado para ajustar os pesos em cada ciclo de retroalimentação. No entanto, conforme a rede "olha" para trás na sequência, os gradientes podem se tornar muito pequenos (vanishing) ou muito grandes (exploding), prejudicando o treinamento.

3. Backpropagation Through Time (BPTT)

- **Funcionamento:** O BPTT ajusta os pesos das conexões recorrentes ao "desdobrar" a rede em uma sequência de camadas ligadas, o que permite calcular como cada entrada passada impacta a saída atual.
- **Dissipação e Explosão do Gradiente:** O problema de dissipação do gradiente ocorre quando os gradientes se tornam muito pequenos para influenciar o aprendizado nas camadas iniciais da sequência, tornando

difícil para a rede aprender dependências de longo prazo. A explosão do gradiente ocorre quando o gradiente se torna muito grande, o que pode desestabilizar o processo de treinamento.

- **Técnicas para Contornar o Problema:**

- **Clipping de Gradiente:** Limita os valores dos gradientes para evitar explosão.
- **Backpropagation Truncado:** Limita o número de passos para os quais o gradiente é calculado, reduzindo o custo computacional e mitigando o problema de dissipação do gradiente.

4. LSTM (Long Short-Term Memory)

- **Introdução e Objetivo:** LSTM é uma variante das RNNs projetada para superar os problemas de dissipação e explosão de gradientes, armazenando informações por períodos mais longos.
- **Portões de Controle:**
 - **Forget Gate:** Decide quanta informação das células anteriores deve ser esquecida.
 - **Input Gate:** Controla quanta informação nova será adicionada ao estado da célula.
 - **Output Gate:** Determina a saída do LSTM para o próximo passo de tempo.
- **Estado da Célula (Cell State):** Permite que a rede transporte informações de longo prazo ao longo da sequência. É esse "estado da célula" que guarda o contexto acumulado.

- **Vantagens:** A estrutura de portões permite que a LSTM se concentre em informações relevantes e descarte as irrelevantes, tornando-a eficaz para sequências longas.

5. GRU (Gated Recurrent Unit)

- **Introdução e Estrutura:** A GRU é uma simplificação da LSTM que possui menos parâmetros, o que a torna mais eficiente. Ela tem apenas dois portões:
 - **Reset Gate:** Define quanto do estado anterior deve ser esquecido.
 - **Update Gate:** Controla quanto da nova informação será combinada com a informação anterior.
- **Estado Único:** Ao contrário da LSTM, a GRU tem apenas um estado oculto que acumula e libera informações conforme necessário.
- **Comparação com LSTM:** As GRUs são menos complexas, porém também conseguem lidar com dependências de longo prazo. Sua eficiência computacional as torna uma boa escolha para muitas aplicações onde a LSTM poderia ser usada, especialmente quando o poder computacional é um fator importante.

6. Transformadores

- **Motivação:** Enquanto as RNNs processam sequências de forma sequencial, isso limita a eficiência e o desempenho, especialmente para

textos longos. Os transformadores permitem processar toda a sequência em paralelo.

- **Mecanismo de Atenção:** Ao contrário das RNNs que tendem a "esquecer" informações distantes, os transformadores utilizam "atenção" para identificar quais partes de uma sequência são mais relevantes para a tarefa.
- **Estrutura Encoder-Decoder:**
 - **Encoder:** Codifica as informações de entrada, incluindo suas relações contextuais.
 - **Decoder:** Usa essas codificações para gerar a saída, utilizando a informação contextual capturada pelo encoder.
- **Aplicações:** Os transformadores são amplamente usados em processamento de linguagem natural (PLN) e tradução automática, pois mantêm o contexto ao longo da sequência e podem lidar com relações distantes, mantendo alta eficiência.