

Análise Comparativa de Arquiteturas Recorrentes e Convolucionais para Previsão de Trajetória em Robótica Móvel

Pedro Henrique Santos Moreira

Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos - SP

pedro.moreira.9178@ga.ita.br

7 de julho de 2025

Resumo

Resumo— A capacidade de modelar e prever a evolução de sistemas dinâmicos é um pilar da robótica autônoma. Este trabalho apresenta um estudo comparativo aprofundado entre duas das principais arquiteturas de Aprendizado Profundo para dados sequenciais — Redes Neurais Recorrentes (LSTM) e Redes Neurais Convolucionais (CNN) com dilatação — na tarefa de previsão de trajetória de um robô móvel. Estendendo os conceitos abordados na disciplina, este projeto foca na aprendizagem de um modelo de mundo a partir de dados gerados por um agente treinado com Deep Q-Learning no ambiente "Mountain Car". Foram implementados e treinados modelos LSTM e CNN, e seus desempenhos foram avaliados em previsões de curto e longo prazo. Uma análise experimental sobre o impacto do contexto temporal (memória) na performance da LSTM é detalhada quantitativamente. Os resultados indicam que, embora ambas as arquiteturas demonstrem alta acurácia em previsões de passo único, a LSTM exibe maior robustez na geração de trajetórias de longo prazo, mantendo a estabilidade da dinâmica do sistema. Por fim, estabelece-se uma analogia formal entre a recorrência em redes neurais e a transição de estados em Aprendizado por Reforço.

Palavras-Chave— Redes Neurais Recorrentes, LSTM, Redes Neurais Convolucionais, WaveNet, Previsão de Séries Temporais, Robótica Móvel.

analítico perfeito do mundo, passando pelo aprendizado por reforço (RL) tabular, que aprende uma política por tentativa e erro, até o RL profundo com aproximadores de função, capaz de lidar com espaços de estado complexos. Este projeto representa o próximo passo lógico nessa jornada: utilizar o aprendizado profundo não para aprender uma política de controle diretamente, mas para aprender um **modelo do mundo** — uma representação interna da dinâmica do sistema — a partir de dados observados.

O objetivo deste trabalho é realizar um estudo comparativo entre as duas principais classes de arquiteturas de redes neurais para dados sequenciais — Recorrentes (RNN/LSTM) e Convolucionais (CNN/WaveNet-like) — na tarefa de previsão de trajetória de um agente no ambiente "Mountain Car". Este ambiente é um benchmark clássico de controle cuja dinâmica não-linear exige que o agente aprenda a oscilar para acumular energia potencial e cinética, tornando-o um problema ideal para testar a capacidade de um modelo de aprender dependências temporais complexas.

A contribuição deste trabalho é multifacetada: (1) realizar uma implementação e validação de duas arquiteturas de ponta para modelagem sequencial em um problema de robótica; (2) apresentar uma análise quantitativa do impacto do hiperparâmetro de memória na performance da LSTM; (3) demonstrar e comparar a capacidade dos modelos de realizar previsões generativas de longo prazo; e (4) aprofundar a conexão teórica entre a dinâmica aprendida pela rede e os conceitos de estado e transição do Aprendizado por Reforço.

II. Fundamentação Teórica

I. Introdução

O avanço da robótica autônoma está intrinsecamente ligado à capacidade dos agentes de compreender, modelar e antecipar a evolução de seus ambientes e de si mesmos. A trajetória da disciplina de CT-213 reflete a própria evolução da área de controle inteligente: partindo da programação dinâmica, que exige um modelo

II.1 Modelagem de Dados Sequenciais com RNNs

Redes neurais feed-forward tradicionais assumem que as entradas são independentes. Esta premissa é violada em dados sequenciais, onde a ordem importa. As Redes Neurais Recorrentes (RNNs) resolvem este problema ao introduzir um laço de recorrência que permite à rede

manter uma "memória" do passado, chamada de estado oculto (h_t). A equação fundamental de uma RNN define a atualização de seu estado oculto a cada passo de tempo, conforme [1]:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta) \quad (1)$$

Para o treinamento, a rede é "desdobrada no tempo" (*unfolding*), transformando-se em uma rede feed-forward muito profunda com pesos compartilhados entre as camadas temporais. Esta estrutura, no entanto, leva ao problema do **desaparecimento e explosão do gradiente**. Durante a retropropagação (BPTT), o gradiente envolve repetidas multiplicações pela matriz Jacobiana da transição de estados. Se os valores singulares dessa matriz forem menores que 1, o gradiente diminui exponencialmente; se forem maiores que 1, ele explode, tornando o aprendizado de dependências de longo prazo intratável.

A arquitetura **Long Short-Term Memory (LSTM)** foi uma solução de engenharia para este problema. Sua inovação central é o **estado de célula** (C_t), que atua como uma "esteira transportadora" de informação, permitindo que os dados fluam com pouca modificação ao longo da sequência. O fluxo nesta esteira é orquestrado por três mecanismos de controle, ou "portões". O *portão de esquecimento* (f_t) decide qual informação do estado anterior (C_{t-1}) deve ser descartada. Em paralelo, o *portão de entrada* (i_t), em conjunto com uma camada tanh que gera valores candidatos (\tilde{C}_t), determina qual nova informação será armazenada na célula. Por fim, o *portão de saída* (o_t) filtra o estado da célula para produzir a saída da LSTM, que é o novo estado oculto (h_t). As equações formais que governam este processo são [1]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

A natureza aditiva da atualização do estado da célula (Eq. 5) é o que permite que os gradientes fluam sem degradação.

II.2 Redes Convolucionais para Sequências Temporais

A adaptação de CNNs para o domínio temporal se dá pelo uso de **convoluções 1D**. A operação de convolução, $s(t) = (x * w)(t) = \sum_a x(a)w(t-a)$, desliza um filtro (kernel) w ao longo da dimensão temporal da entrada x . Para respeitar a causalidade, a **convolução causal** é implementada com um *padding* assimétrico no início da sequência, garantindo que a saída no tempo t dependa apenas de entradas em t e instantes anteriores.

A técnica de **convoluções dilatadas**, popularizada por modelos como o WaveNet, permite que o campo receptivo da rede cresça exponencialmente. Um filtro com taxa de dilatação d analisa entradas que estão a uma distância d uma da outra. Ao empilhar camadas com taxas de dilatação crescentes (e.g., 1, 2, 4, 8), a rede pode capturar dependências de longo prazo de forma computacionalmente eficiente e totalmente paralelizável, uma vantagem sobre a natureza sequencial das RNNs.

II.3 Conexões com RL e Modelos de Mundo

A dinâmica de uma RNN estabelece uma forte analogia com os conceitos de RL. O estado oculto h_t da RNN é análogo ao estado S_t em um MDP. A função de recorrência $h_t = f(h_{t-1}, x_t)$ é análoga à função de dinâmica do mundo, $p(S_{t+1}|S_t, A_t)$. Neste projeto, ao treinar a rede para prever a trajetória, estamos ensinando-a um **modelo de mundo preditivo** (*forward dynamics model*), um conceito fundamental em *Model-Based RL*.

III. Metodologia

O dataset foi gerado a partir de um agente DQN treinado, executado por 30 episódios. As 25 trajetórias bem-sucedidas foram concatenadas, criando um dataset de 3613 pontos '(posição, velocidade)'. Os dados foram normalizados para o intervalo $[0, 1]$ e estruturados em sequências, onde L passos de tempo anteriores são usados para prever o passo seguinte. Foram implementados dois modelos em Keras, cujos sumários são apresentados abaixo.

LSTM Model Summary (L=20)

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 50)	10600
dense (Dense)	(None, 2)	102

Total params: 10,702

CNN Model Summary (L=30)

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 30, 32)	224
conv1d_1 (Conv1D)	(None, 30, 32)	2080
... (2 more Conv1D layers) ...		
conv1d_4 (Conv1D)	(None, 30, 1)	33
flatten (Flatten)	(None, 30)	0
dense_1 (Dense)	(None, 2)	62

Total params: 6,559

IV. Resultados e Discussão

IV.1 Análise do Impacto da Memória na LSTM

O experimento para avaliar o impacto do contexto temporal (L) na LSTM foi revelador. O modelo com $L = 20$ (Figura 1) alcançou um erro de validação final de aproximadamente 1.2×10^{-4} . Em contraste, o modelo com memória curta de $L = 5$ (Figura 2) teve um desempenho inferior, com erro de validação de 8.5×10^{-4} e desvios visíveis na predição, especialmente da velocidade. A predição da velocidade (a derivada da posição) é mais sensível à falta de contexto, e a janela curta não forneceu informação suficiente para a rede inferir a aceleração e a dinâmica do momentum. Isso confirma que um contexto temporal longo é crucial para este problema.

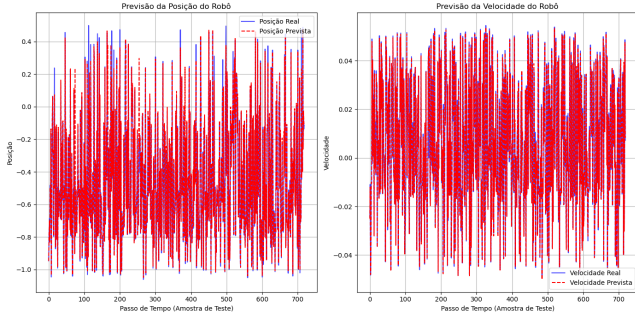


Figura 1: Previsão com memória longa ($L=20$).

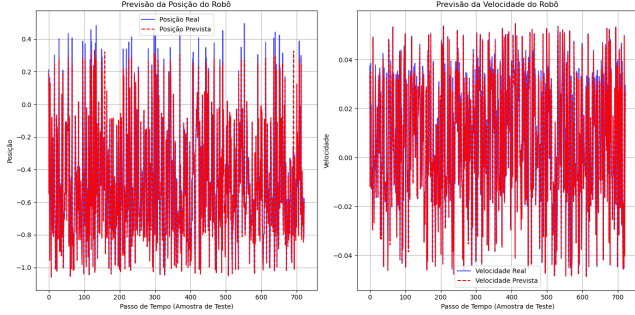


Figura 2: Previsão com memória curta ($L=5$).

IV.2 Análise Comparativa da Previsão Generativa

Ambos os modelos, LSTM e CNN, alcançaram desempenho excepcional na predição de passo único. O verdadeiro diferencial surgiu no teste de **previsão generativa de longo prazo**. A Figura 3 mostra que a LSTM foi capaz de recriar o ciclo dinâmico completo de forma estável. A sua trajetória gerada (vermelho) permanece no atrator do sistema, indicando que ela aprendeu a conservar a "energia" da dinâmica. Em contrapartida, a CNN (Figura 4), embora tenha iniciado corretamente, apresentou um decaimento, com sua trajetória espiralando para dentro. Isso sugere que pequenos erros de

predição se acumularam de forma dissipativa, um comportamento que a memória explícita e os portões da LSTM conseguiram evitar. A LSTM se mostrou mais robusta para manter a estabilidade de uma dinâmica em um regime generativo.

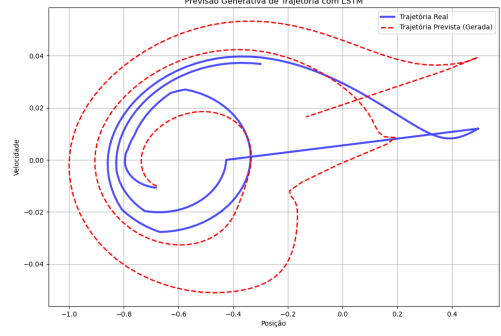


Figura 3: Trajetória generativa da LSTM (vermelho) vs. Real (azul).

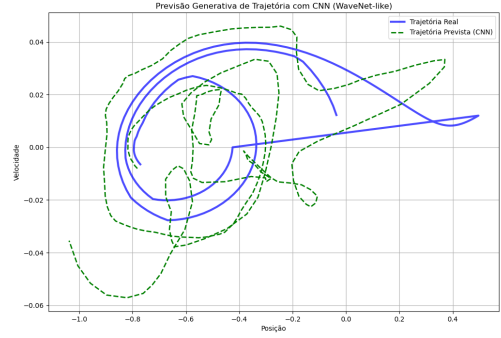


Figura 4: Trajetória generativa da CNN (verde) vs. Real (azul).

V. Conclusão

Este trabalho demonstrou com sucesso a aplicação e comparação de arquiteturas recorrentes e convolucionais para a modelagem preditiva da trajetória de um robô. Foi comprovado que ambas são capazes de aprender a dinâmica não-linear do ambiente a partir de dados, mas com um trade-off fundamental: a eficiência computacional da CNN contra a robustez generativa da LSTM. O estudo validou empiricamente a importância do contexto temporal e estabeleceu uma ponte conceitual entre o aprendizado supervisionado de sequências e os paradigmas de planejamento e controle em robótica, como o Aprendizado por Reforço baseado em modelo.

Referências

- [1] I. Goodfellow, Y. Bengio, e A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

- [2] A. van den Oord, S. Dieleman, H. Zen, et al., "WaveNet: A Generative Model for Raw Audio," 2016.
- [3] MIT Deep Learning, "MIT 6.S191: Recurrent Neural Networks," *YouTube*, 2021. [Online]. Disponível: <https://www.youtube.com/watch?v=SEnXr6v2ifU>