

Impactos da Ferramenta KerasTuner em Modelo de Predição Baseado em LSTM para Pecuária Sustentável

Davi L. Lemos¹, Bianca O. Durgante¹, Naylor B. Perez², Leonardo B. Pinho¹

¹Universidade Federal do Pampa (UNIPAMPA) - Campus Bagé
96.413-172 – Bagé – RS – Brasil

²Embrapa Pecuária Sul
96.401-970 – Bagé – RS – Brasil

{davilemos.aluno,biancadurgante.aluno,leonardopinho}@unipampa.edu.br

naylor.perez@embrapa.br

Resumo. *O trabalho analisa o impacto da ferramenta de autoajuste KerasTuner em um modelo de Rede Neural Recorrente LSTM de predição de massa de forragem para pecuária sustentável, originalmente ajustado empiricamente. Resultados preliminares indicam melhora de 30% na acurácia, associada a um aumento entre 80% e 110% no tempo de treinamento, proporcional à quantidade de hiperparâmetros autoajustados, divergindo das escolhas originais.*

1. Introdução

Algoritmos de aprendizado de máquina representam ferramentas significativas na resolução de uma variedade de questões relacionadas à vida cotidiana. Nos últimos anos, tem-se dedicado considerável atenção ao papel das ferramentas computacionais no que diz respeito à pecuária de precisão e, no caso em questão, melhorar o processo de tomada de decisão no manejo dos animais, equilibrando produtividade com sustentabilidade. Este processo depende da predição da oferta de massa de forragem (MStotal) em uma determinada área de pastagem, necessário para que se faça um ajuste eficiente na taxa de lotação, isto é, na quantidade de animais de acordo com a disponibilidade de alimento em quilogramas por hectare (kg/ha) visando a otimização do ganho de peso.

Neste trabalho é analisada a predição de MStotal, sob o aspecto do tempo de execução de um modelo proposto anteriormente e a hipótese de que é possível aperfeiçoar a sua acurácia com ajuste fino de hiperparâmetros, usualmente realizado de maneira empírica, que na presente abordagem, é feito com a ferramenta de autoajuste KerasTuner, porém com aumento no custo computacional. Na revisão da literatura realizada [Lemos et al., 2023], não foram encontrados trabalhos correlatos abordando este tópico no contexto da pecuária, destacando a potencial contribuição de um estudo exploratório capaz de analisar o impacto desta abordagem de ajuste automatizado no tempo médio de execução desse algoritmo, em diferentes ambientes computacionais, bem como na acurácia do método de predição.

2. Metodologia

Esta pesquisa exploratória experimental parte do trabalho publicado por Schulte et al. [2019] que dedicou-se a produzir um algoritmo como parte de um sistema de suporte à

decisão para manejo de pastagens através da predição da MStotal em determinada área. Mais especificamente, foram realizadas modificações no algoritmo original para a aplicação da ferramenta de autoajuste KerasTuner [O'Malley et al., 2019] e o planejamento de experimentos capazes de viabilizar a análise do efeito na acurácia do modelo e no tempo de execução em diferentes ambientes computacionais, permitindo evidenciar e discutir sobre a interferência desses fatores na execução do algoritmo.

O algoritmo se caracteriza como uma Rede Neural Recorrente (RNN) com arquitetura *Long Short-Term Memory* (LSTM), a partir de um banco de dados espacial contendo séries temporais estratificadas sobre determinado espaço geográfico e de dados meteorológicos, usada para predição de MStotal em áreas de pastagem natural do bioma Pampa. No processo de criação das RNN, é necessário definir dois grupos de amostras: um que é aplicado no processo de treinamento da rede e outro para validação do aprendizado. Definiu-se os doze últimos valores contidos no arquivo de entrada como *dataset* de validação (correspondente a um ano), enquanto o restante caracteriza os dados de treino. A estrutura da rede consiste em três camadas, conforme o modelo original, sendo duas delas LSTM e a última densa, visto que é responsável pela constatação do resultado, contendo, respectivamente, 30, 15 e 1 neurônios. O modelo é treinado durante 5000 épocas (quantidade ideal identificada empiricamente) e ao final expressa a relação entre os valores reais e preditos através de gráficos de caixa e dispersão, gerados a partir das métricas de desempenho.

A partir da incorporação do KerasTuner no processo de ajuste dos hiperparâmetros, foi possível eliminar o processo empírico de busca pela melhor combinação entre tais fatores. No seu funcionamento, é possível realizar a sintonização de quatro formas: Hyperband, BayesianOptimization, Sklearn e, a escolhida por ter sido frequentemente usada em trabalhos similares (que integram a arquitetura LSTM com KerasTuner), RandomSearch (busca aleatória). A definição do modelo mantém duas camadas, contudo, estas não vêm de um número pré estabelecido de neurônios em cada uma delas, sua variação ocorre de 32 a 512 e este valor é definido ao longo da busca aleatória, assim como a taxa de aprendizado, o estado do dropout e a função de ativação LSTM a ser usada: Tangente Hiperbólica (Tanh) ou Unidade Linear Retificada (ReLU), sendo estas as que tiveram melhor desempenho quando aplicadas por Schulte [2019]. Com a definição do modelo é possível configurar o sintonizador, estabelecendo o número máximo de 10 tentativas, com 3 execuções a cada uma delas. Além disso, institui-se que a variável alvo da ferramenta será a acurácia através dos dados de validação, redefine-se o modelo para a melhor combinação realizando o treinamento com 5000 épocas e a sintetização das métricas de avaliação.

Tendo em vista a demanda de recursos computacionais para utilização de algoritmos de aprendizagem profunda, realiza-se sua execução em diferentes configurações disponibilizadas no ambiente Google Colaboratory (Colab) em sua versão gratuita. Desta forma, foram realizados testes com Unidade Central de Processamento (CPU), Unidade de Processamento Gráfico (GPU) e Unidade de Processamento de Tensor (TPU), comparando os diferentes tempos de execução. O equipamento disponível conta, nos respectivos ambientes, com o processador Intel Xeon 2,2 GHz utilizando de um núcleo com duas *threads* (possuindo, em sua totalidade, dez núcleos com vinte *threads*), GPU Tesla T4 15 GB GDDR6 com 2560 núcleos e uma TPU v2 16 GB *High Bandwidth Memory* (HBM) com oito núcleos. Em se tratando de um ambiente compartilhado de execução, foram

realizados experimentos em diferentes horários evidenciando eventual impacto pelo uso intensificado do serviço em determinado período do dia. O programa foi dividido e analisado em seis partes, sendo elas: manipulação dos dados, definição e treinamento do modelo original, cálculo de métricas do modelo original, definição e busca aleatória do KerasTuner, treinamento do modelo adaptado e cálculo das métricas, desconsiderando os processos de sintetização. Buscando analisar estatisticamente os tempos obtidos, para cada condição repetiu-se a execução do algoritmo cinco vezes, uma vez que, como citado anteriormente, são limitados os recursos da versão gratuita do Colab. A divisão dos horários se deu a cada 8 horas, estipulando-se os horários de 6h, 14h e 22h no fuso horário *Greenwich Mean Time* (GMT), no qual opera o ambiente de execução.

3. Resultados e Discussões

Ao realizar a execução dos modelos descritos, com os dados correspondentes à área de pastagem na qual o modelo original havia apresentado melhor acurácia, observou-se, um erro médio quadrado (RMSE), principal métrica para a acurácia do algoritmo, igual a 1.993,24 kg/ha para o Modelo Original e 1.392,568 kg/ha para o Modelo Adaptado, correspondente a uma melhoria de 30,14% na acurácia, superando outros trabalhos que usam KerasTuner com LSTM [Lemos et al., 2023].

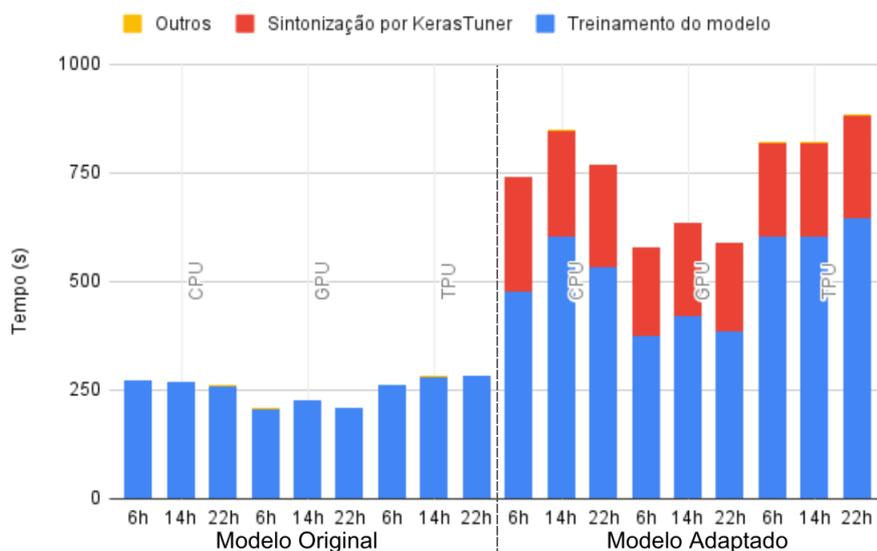


Figura 1. Gráfico de barras dos tempos de execução

Em contrapartida, por conta da aplicação de métodos otimizados para busca de hiperparâmetros, atividade de considerável custo computacional, cabe analisar os tempos observados em cada processo nos diferentes ambientes de execução (Figura 1). Dentre as configurações e horários testados, é possível perceber que não há diferença significativa na duração do processo de sintonização, assim o maior impacto não está necessariamente no horário de execução mas, nos hiperparâmetros selecionados para o modelo naquela sessão. Ainda sobre as escolhas da ferramenta, KerasTuner definiu ReLU como função de ativação, diferentemente do que havia sido apresentado por Schulte [2019]. Destaca-se como menor tempo a execução feita em GPU às 6h e como o pior caso aquele em que

se utilizou TPU, às 22h. Ademais, é perceptível que o uso do KerasTuner resultou em um maior tempo para o treinamento do modelo, cerca de 82% de acréscimo de tempo com GPU e de 110% com TPU, além do tempo de sintonização. Por outro lado, concluiu-se que mesmo com um número relativamente pequeno de amostras, os tempos foram significativos, totalizando cerca de 10 minutos para definição, sintonização, treinamento do modelo com KerasTuner e validação, ao aumentar a quantidade de amostras ou ampliar o máximo de tentativas ou épocas, a tendência é um aumento expressivo no tempo.

4. Conclusão

Este trabalho trouxe uma análise do desempenho do modelo de predição em diferentes ambientes computacionais executados em diferentes turnos, usando ou não, KerasTuner. Com isso, foi possível perceber que tais fatores interferem de diferentes formas: quanto aos diferentes ambientes computacionais, foi possível perceber que a escolha do GPU representou considerável redução no tempo de treinamento com KerasTuner que, por sua vez, foi responsável pelo aumento da acurácia na predição da MStotal do algoritmo superando as expectativas de Lemos et al. [2023] em 50%. Contudo, quando se trata dos horários de execução, é possível afirmar que não houve impacto expressivo, mas sim, que este se deu por conta dos diferentes hiperparâmetros definidos pela ferramenta. Em suma, esta pesquisa visa a contribuição com o estado da arte, incorporando aspectos ambientais, sociais e de governança (ESG), *Smart Farming* e tecnologias modernas na agricultura e pecuária de precisão. Salienta-se a intenção de pesquisa e trabalhos futuros voltados ao aprimoramento da técnica e das variáveis do modelo, experimentando a arquitetura de rede alternativa à LSTM denominada *Gated Recurrent Unit* (GRU) e buscando constatar um padrão comportamental de KerasTuner, tornando possível a reprodução de seus métodos e escolhas usuais manualmente (mitigando o tempo de sintonização), além de uma análise da influência dos parâmetros configurados para a ferramenta no que tange sua eficiência.

Agradecimentos

Reconhece-se o apoio da CAPES - Código de Financiamento 001 e dos Programas Institucionais de Bolsas de Iniciação Científica PIBIC/CNPq e PROBIC/FAPERGS.

Referências

- Lemos, D. L., Durgante, B. O., Rocha, H. W. B., Perez, N. B., and Pinho, L. B. (2023). Aprimorando modelos de aprendizagem profunda para pecuária de precisão com kerastuner. *Anais do 15o SIEPE - Salão Internacional de Ensino, Pesquisa e Extensão da UNIPAMPA: Pesquisa e Inovação*, 3(15).
- O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). Kerastuner. <https://github.com/keras-team/keras-tuner>.
- Schulte, L. G. (2019). Suporte à decisão em pastagens: Análise espaço-temporal e aprendizado de máquina para predição da disponibilidade de forragem no contexto de smart farming. Master's thesis, Universidade Federal do Pampa (UNIPAMPA) – Programa de Pós-Graduação em Computação Aplicada (PPGCAP).
- Schulte, L. G., Perez, N. B., Pinho, L. B., and Trentin, G. (2019). Sistema de apoio à decisão para pecuária de precisão: Módulo para ajuste de taxa de lotação. In *Anais do XV Simpósio Brasileiro de Sistemas de Informação (SBSI)*, pages 95–102, Porto Alegre, RS, Brasil. SBC.