

Redes neurais profundas para classificação

1st João Pedro Aguiar Formiga Matos *EMC-UFG*
Universidade Federal de Goiás
 Goiânia, Brasil
 Joao.formiga@discente.ufg.br

Abstract—A classificação precisa de esportes é uma tarefa desafiadora e relevante em diversos contextos. Neste trabalho, propõe-se um modelo de classificação de 100 esportes diferentes usando técnicas de aprendizado de máquina. O objetivo é categorizar automaticamente uma ampla variedade de esportes, desde os populares até os menos conhecidos. Adotou-se uma abordagem baseada em redes neurais convolucionais (CNNs) para lidar com o desafio de múltiplas classes esportivas. O modelo é capaz de classificar de forma adequada uma boa parte das classes esportivas. Neste artigo, apresentamos um estudo realizado como parte da disciplina de Redes Neurais Profundas (2023.1) na Universidade Federal de Goiás, Faculdade das Engenharias. O objetivo principal deste trabalho é avaliar o desempenho de um modelo de classificação de 100 esportes diferentes usando redes neurais profundas, considerando métricas como acurácia e função de perda. No restante do artigo, será detalhada a metodologia, a implementação do modelo, os experimentos realizados, os resultados obtidos e as considerações finais.

I. METODOLOGIA

A. Modelo 1

O objetivo do trabalho é criar uma rede convolucional. Neste trabalho, propõe-se um modelo de classificação de 100 esportes diferentes usando técnicas de aprendizado de máquina. O objetivo é categorizar automaticamente uma ampla variedade de esportes, nesta tarefa o aluno deve fazer a sua própria arquitetura, e após isso modificar essa arquitetura para utilizar uma função de convolução customizada onde será aplicada uma função com mais parâmetros atualizáveis com a expectativa de que melhorem a acurácia do modelo.

Se iniciou utilizando um modelo bem simples com 3 camadas convolucionais, cada uma com uma função de ativação relu e maxpool, os resultados obtidos não foram nada satisfatórios chegando à marca de apenas 50% de acurácia, teve-se um período de 2 dias em que houve um erro de lógica no código em que o modelo era incapaz de aprender e por causa disso não foi obtido nenhum resultado que fosse o mínimo satisfatório depois da resolução deste problema aumentou-se a complexidade do modelo e que após alguns dias de teste modificando hiperparâmetros obteve-se aumento da acurácia do modelo para 70% nos dados de teste e validação, neste período fora ministrada uma aula pelo professor da matéria de redes neurais profundas que trouxe algumas coisas novas como o batch normalization e arquiteturas famosas de redes neurais, como a GoogleLENET, Resnet e por fim a NIN.

Com a GoogleLENET obteve-se um resultado bem promissor aproximadamente 76% de acurácia de validação, porém

o tempo de cada época aumentou substancialmente e dado o escopo em que estava sendo treinado (300 épocas) dificultaria e muito a entrega do trabalho a tempo.

Com a Resnet analogamente à GoogleLENET obteve-se resultados muito promissores de 74% de validação mas após 30-40 épocas o modelo encontrava problemas devido a memória ram do computador ter se esgotado, após algumas pesquisas no Medium e em outros meios de pesquisa encontrou-se algumas formas de melhorar a performance do modelo e a otimizar a utilização de memória ram porém nenhuma das formas conseguiu resolver a questão do alto uso de memória ram do modelo, dessa forma decidiu-se a mudar de modelo pois não havia muito mais tempo para terminar o trabalho (aproximadamente 5 dias).

Com a NIN obteve-se de forma imediata, resultados muito satisfatórios com o modelo chegando a 78% de validação na época 70, porém após algumas alterações na rede, tanto de camadas como adicionando BatchNormalization conseguiu obter 80.4% na época 60 do modelo dessa forma então após treinar 300 épocas foi se traçado um ponto ótimo para o modelo na época 142 com acurácia de 83.4% de precisão de validação e 85% de precisão de teste, abaixo esta anexada uma tabela com os dados:

B. Modelo 2

No modelo 2, a abordagem utilizada consistiu em treinar a mesma rede neural do modelo 1, porém com uma modificação na função de convolução. a qual envolvia a convolução entre X e V , seguida por um processo não-linear gaussiano com média e variância. Os parâmetros μ , σ^2 e V foram definidos como atualizáveis, permitindo que μ e σ^2 fossem aprendidos automaticamente durante o processo de treinamento:

$$H = \text{bias} + \left(\frac{1}{\sigma \sqrt{2\pi}} \right) \left(-\frac{\sum (V \cdot X - \mu)^2}{2\sigma^2} \right)$$

Em relação à validação do modelo, buscou-se equacionar eficientemente o processo utilizando referências e pesquisas sobre funções de convolução 2D. A fórmula foi adaptada e implementada com o suporte da biblioteca PyTorch, aproveitando seus recursos e funcionalidades.

Durante o processo de equacionamento, foram definidos o bias e a variância como parâmetros do PyTorch, permitindo que fossem atualizados ao longo do treinamento. Essa abordagem contribuiu para o aprimoramento do modelo, possibilitando a obtenção de resultados mais precisos e melhor desempenho nas tarefas de classificação.

II. RESULTADO DAS SIMULAÇÕES

A. Modelo 1

Obteve-se estes dados de simulação do modelo 1 truncado na epoca 206:

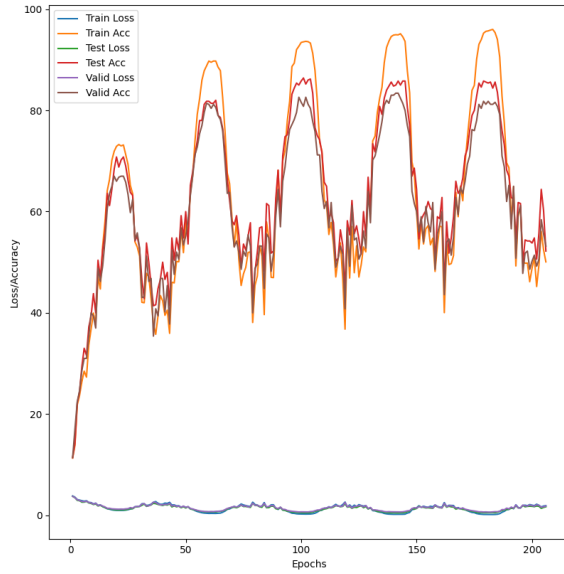


Fig. 1. dados modelo 1

Os dados do grafico sao:

epoch	train loss	train acc	test loss	test acc	valid loss	valid acc
1	3.82	11.31	3.73	11.4	3.74	11.4
20	0.97	73.26	1.06	68.8	1.21	66.8
40	2.41	39.50	2.10	46.6	2.16	40.2
60	0.38	89.51	0.68	81.4	0.74	80.4
80	2.01	47.26	1.95	52.0	2.03	49.6
100	0.24	93.54	0.51	86.4	0.68	80.8
120	2.06	46.87	1.73	55.4	1.97	49.2
140	0.20	94.95	0.51	85.0	0.67	83.4
160	1.61	56.98	1.45	62.8	1.46	60.0

TABLE I
DADOS DE SIMULAÇÃO

B.

Podemos observar que no Modelo 1, a precisão oscila e há uma tendência de aumento ao longo das épocas. Devido à falta de tempo, não foi possível aumentar significativamente o número de épocas para verificar se o modelo convergiria para uma precisão de 90%. No entanto, é importante destacar que na época 140, obtivemos uma precisão de 85%. É provável que a otimização do modelo seja necessária, juntamente com um esforço para tornar a curva de aprendizado mais exponencial em vez de senoidal.

C. Modelo 2

Obteve-se estes dados de simulação do modelo 2 truncado na epoca 206:

Obteve-se estes dados de simulação do modelo:

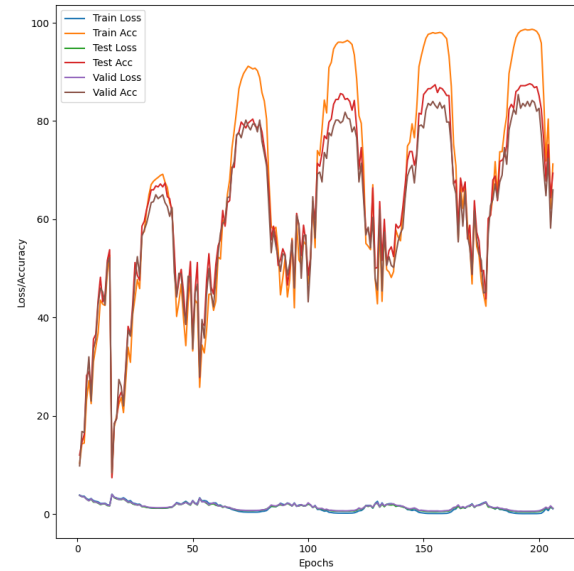


Fig. 2. Modelo 2 acuracia e loss

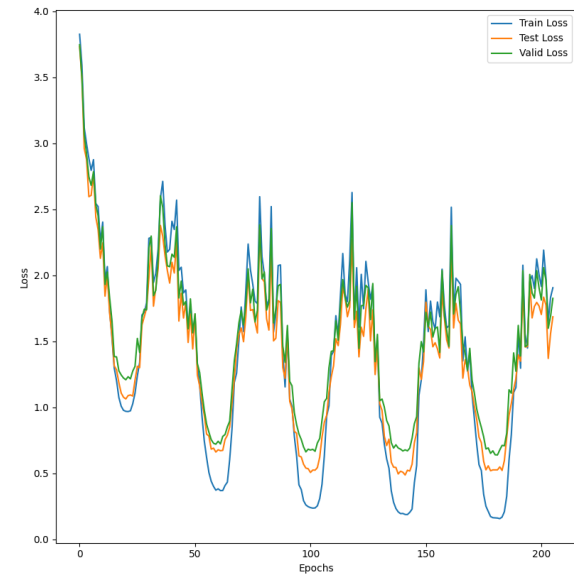


Fig. 3. loss modelo 1

D.

Analogamente, no Modelo 2, podemos observar que a precisão obtida não foi melhor do que no modelo 1, isso se deve por que o período de amostragem dos dados dispostos na tabela se dar de 20 em 20 épocas porem foi obtido uma acurácia máxima de teste de 90.6% e de validação de 86.2% na época 297, como pode ser observado no gráfico abaixo:

epoch	train loss	train acc	test loss	test acc	valid loss	valid acc
1	3.853	10.309	3.774	12.0	3.802	9.8
20	2.994	27.230	2.737	30.2	2.768	29.8
40	1.409	62.184	1.381	61.4	1.400	62.4
60	1.768	52.327	1.663	54.6	1.661	53.0
80	0.583	84.094	0.885	74.6	0.885	73.4
100	1.876	51.941	1.816	52.0	1.873	50.8
120	0.414	88.608	0.747	78.6	0.885	76.6
140	1.450	61.755	1.456	62.4	1.609	58.2
160	0.249	93.247	0.622	85.2	0.745	79.8

Todos os testes do modelo nesse artigo serão realizados na época de melhor desempenho. No entanto, notamos que ele converge mais rapidamente e com maior precisão do que o Modelo 1.

Ainda assim, há espaço para melhorias nos hiperparâmetros, a fim de realizar uma análise mais aprofundada das melhorias em relação ao Modelo 1.

III. TESTES COM IMAGENS

Nessa sessão será mostrado algumas imagens com suas previsões pelo modelo 1 e pelo modelo 2, o motivo para isso é para possivelmente detectar alguns padrões em que o modelo 2 melhor captura que o modelo 1 o que auxiliaria para a sua melhor acurácia

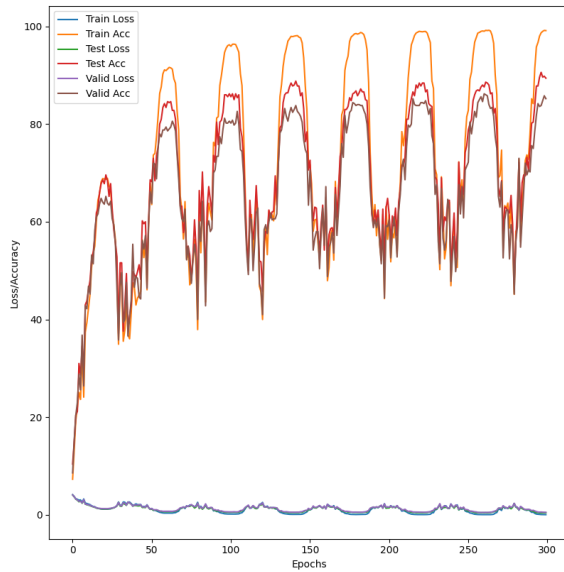


Fig. 4. Modelo 2 300, épocas



Fig. 6. Classificação correta

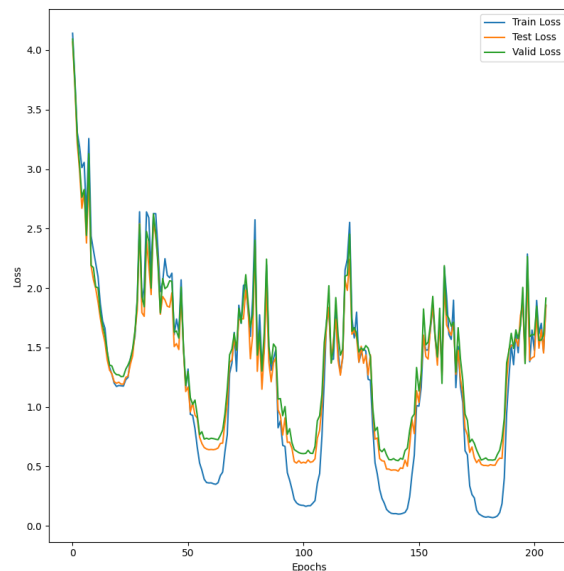


Fig. 5. loss modelo 2



Fig. 7. Classificação inadequada

Pred true: sailboat racing
 Pred model 1: sailboat racing
 Pred model 2: sailboat racing



Fig. 8. Classificação correta

Pred true: javelin
 Pred model 1: javelin
 Pred model 2: javelin

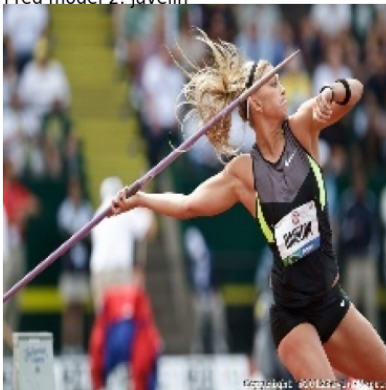


Fig. 9. Classificação correta

Pred true: croquet
 Pred model 1: golf
 Pred model 2: croquet



Fig. 10. Classificação inadequada

Pode-se observar que as imagens em que o modelo 1 não desempenhou tão bem quanto eram bem ambíguas, por exemplo quando classificou um frisbee como baseball pelo fato de ter um disco que aparenta ser uma bola indo em direção a uma pessoa, dito isso o modelo 2 aparenta ser melhor para generalizar e classifica mais corretamente em relação ao modelo 1.

IV. CONCLUSÃO

Em conclusão, os resultados obtidos estão de acordo com as expectativas, evidenciando diferenças entre o modelo 1 e o modelo 2. O modelo 2 demonstrou um desempenho superior em termos de métricas alcançadas e também em desempenho em imagens que aparentavam ser ambíguas, apresentando pontuações mais altas em comparação ao modelo 1. Além disso, o modelo 2 também se destacou por uma velocidade de treinamento ligeiramente mais rápida.

Essas diferenças sugerem que a modificação na função de convolução adotada no modelo 2 desempenhou um papel perceptível na melhoria dos resultados. A introdução do processo não-linear gaussiano com média e variância pode ter permitido uma melhor captura de características e uma maior capacidade de aprendizado da rede neural.

Como resultado, o modelo 2 conseguiu atingir métricas mais elevadas de forma mais eficiente, indicando seu potencial porem necessita de mais testes em outros modelos para ser provado a sua eficacia.

Porem apesar das métricas superiores o ambiente não é o ideal visto que o modelo 1 não desempenhou tao bem quanto o comprovado pela literatura(95% de acurácia) mas as melhoras no modelos atuais são perceptíveis.