Assignment 3: Action Recognition from Videos using Recurrent Neural Networks

Juan Belieni e Juliana Carvalho

14 de Outubro de 2023

1 Introdução

No trabalho disponível no seguinte notebook do Colab utilizamos três redes neurais recorrentes, Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU) e Deep Long Short Term Memory (Deep LSTM) para realizar a classificação de frames de vídeos em quatro classes de esportes diferentes. Esses esportes são Basketball (Basquetebol), Diving (Mergulho), Golf Swing (Balanço de Golfe), Skiing (esqui). Iremos também comparar a performance e complexidade dessas redes.

2 Conceitos e implementação

2.1 Extração de features

Para que seja possível analisar cada *frame* dos vídeos, as imagens são processadas por uma VGG16 sem as camadas densas, para que as principais *features* das imagens sejam extraídas (como em um processo de *encoding*). No código, esse processo é feito a partir dos seguintes métodos:

- `get_model`: retorna o modelo VGG16 sem as camadas densas e treinado na *ImageNet*.
- `extract_features`: utiliza o modelo convolucional especificado para extrair as *features* dos primeiros 40 *frames* dos vídeos.
- `extrac_features_from_set`: extrai as features das imagens de um determinado conjunto de nomes de arquivo.

Essa operação de extração de *features* é importante para economizar memória e processamento em tempo de treinamento, já que o modelo convolucional é estático e não sofre atualização nos pesos.

2.2 Definição do modelo, do treinamento e dos testes

O modelo base (que será posteriormente adaptado nos experimentos) é definido na seção 'Define the trainable LSTM model' como um modelo sequencial, onde acontece a adição de uma LSTM com um determinado número de unidades e uma camada densa de *output* de 4 unidades, uma para cada classe.

Esse modelo é treinado utilizando os dados de treino, dos quais 10% será usado pelo *Tensorflow* como dados de validação. Durante o processo de treinamento, acontece o monitoramento da *loss* de validação por meio do método de *early stopping*, que está definido com paciência igual a 3.

No teste, contamos com métricas de acurácia, *F1 score* médio e *F1 score* para cada classe, que são realizados ao final do processo de treinamento.

2.3 Construção dos experimentos

Os experimentos foram feitos a partir de 3 funções para cada tipo de experimento (com LSTM, GRU e Deep LSTM). Além disso, existe uma função para realizar o treino e retornar as métricas de teste (`train_and_test`). Esses métodos realizam as mesmas operações ditas nas subseções anteriores.

3 Dados

Os dados foram extraídos da base de dados de reconhecimento de ações UCF101 da Universidade Central da Flórida. Quatro classes foram selecionadas: basquetebol, mergulho, balanço de golfe e esqui.

Cada vídeo é composto de 40 frames. Esses frames são utilizados como entrada da rede, cuja arquitetura é composta de um encoder com os pesos congelados, seguido por uma camada de global average pooling, e posteriormente por uma rede neural recorrente dentre as três analisadas (LSTM, GRU e RNN). O output final é o resultado produzido pela camada densa que segue a RNN.

4 Experimentos

Nos experimentos utilizamos os seguintes hiperparâmetros:

Batch Size	**Otimizador**	**Loss**	**Early Stopping Patience**	**Min Delta**	**Epochs**
64	Adam	Categorical Crossentropy	3	0	100

Tabela 1: Detalhes do experimento.

5 Resultados

Na tabela a seguir resumimos a quantidade de parâmetros de cada experimento, sendo todos os parâmetros treináveis, ou seja, não congelados pela rede. Vemos que a quantidade de parâmetros aumenta com a quantidade de unidades internas no mesmo tipo de rede neural recorrente. Mantendo-se fixa a mesma quantidade de unidades internas, a GRU tem menos parâmetros que a LSTM, a qual por sua vez tem menor quantitdade de parâmetros que a Deep LSTM. Isso reflete a arquitetura de cada uma das redes, sendo a GRU uma versão mais simplificada da LSTM e a Deep LSTM uma versão amplificada da LSTM.

Exp.	RNN Network	Internal Units	Quantidade de Parâmetros
1.1	LSTM	50	112804
1.2	LSTM	100	245604
1.3	LSTM	200	571204
1.4	LSTM	500	2028004
2.1	GRU	50	84804
2.2	GRU	100	184604
2.3	GRU	200	429204
2.4	GRU	500	1523004
3.1	Deep LSTM	50	133004
3.2	Deep LSTM	100	326004
3.3	Deep LSTM	200	892004
3.4	Deep LSTM	500	4030004

Tabela 2: Quantidade de Parâmetros

Em relação aos tempos de execução e a quantidade de épocas necessárias podemos observar na tabela 3 que:

- A medida que a quantidade de épocas aumenta, o tempo também aumenta
- A GRU e a LSTM apresentam, no total, quantidade de épocas e tempo de execução menores ou igual do que a Deep LSTM, revelando assim a maior complexidade desta última.
- Com 100 e 500 units a GRU apresentou tempo de execução e quantidade de épocas menores que a LSTM, sendo pior nos demais casos.

Exp.	RNN Network	Internal Units	Qtd Épocas	Tempo de Execução (s)
1.1	LSTM	50	66	6.2713
1.2	LSTM	100	100	11.77
1.3	LSTM	200	26	4.7539
1.4	LSTM	500	100	11.286
2.1	GRU	50	100	12.469
2.2	GRU	100	14	3.3069
2.3	GRU	200	100	12.094
2.4	GRU	500	10	3.3906
3.1	Deep LSTM	50	100	14.747
3.2	Deep LSTM	100	100	12.954
3.3	Deep LSTM	200	100	13.844
3.4	Deep LSTM	500	100	23.156

Tabela 3: Resultados - Experimentos e seus internal units, quantidade de épocas necessárias e tempo de execução

As métricas estão descritas na tabela 4, onde destacamos os melhores resultados F1-score de cada uma das classes e da acurácia.

- Por meio dessa tabela, podemos concluir que a LSTM com 500 unidades internas obteve uma melhor performance na acurácia geral, na média dos F1-scores, e no F1-score das classes "Diving"e "Skiing".
- Por outro lado, a GRU com 100 unidades internas teve uma performance melhor nos F1-scores de "Golf Swing"e "Skiing". Assim, como o score dessa classe 96.91% é o mesmo para a LSTM, caso tivessemos que detectar apenas "Skiing", poderíamos optar pela GRU como um modelo alternativo mais simples da LSTM.
- A LSTM de 200 unidades internas obteve a melhor performance na classe "Diving".
- De modo geral, a LSTM com 500 unidades apresentou melhor performance que as LSTM de 50 e 100 unidades. Apresentou também melhor performance que a LSTM de 200 unidades, apenas sendo pior ao classificar o esporte "Diving". É importante notar que a LSTM de 50 unidades apresentou a pior performance de acurácia e média dos F1-scores. Entretanto, a LSTM de 100 unidades não se mostrou melhor que a LSTM de 200 unidades, revelando que o aumento de unidades não necessariamente é responsável por aumentar a performance.
- No caso das GRUs, o melhor score de acurácia de F1 média foi para a de 100 unidades, seguida da GRU de 200 unidades, da de 500 unidades e por último da de 50 unidades.
- Por outro lado, a Deep LSTM também apresentou uma melhor performnace com 100 unidades, seguida pela Deep LSTM com 500 unidades em segundo lugar e pela de 50 em terceiro lugar. A pior performance ficou com a rede de 200 unidades.

Exp.	RNN Network	Internal Units	Acc.	Basket.	Diving	Golf Swing	Skiing	AVG F1
1.1	LSTM	50	89.97%	82.87%	93.90%	88.57%	93.81%	89.81%
1.2	LSTM	100	93.73%	88.66%	94.88%	95.05%	96.26%	93.71%
1.3	LSTM	200	93.48%	86.67%	94.93%	95.10%	96.45%	93.28%
1.4	LSTM	500	94.24%	89.25%	94.88%	95.57%	96.91%	94.14%
2.1	GRU	50	89.72%	80.93%	90.50%	92.38%	94.74%	89.43%
2.2	GRU	100	92.73%	84.44%	92.44%	96.48%	96.91%	92.53%
2.3	GRU	200	91.98%	83.70%	93.46%	94.12%	95.92%	91.79%
2.4	GRU	500	90.73%	85.57%	92.11%	92.15%	93.26%	90.77%
3.1	Deep LSTM	50	89.72%	82.05%	93.95%	89.73%	92.61%	89.62%
3.2	Deep LSTM	100	91.23%	86.70%	93.46%	90.71%	93.94%	91.21%
3.3	Deep LSTM	200	88.97%	77.30%	91.67%	90.32%	94.06%	88.33%
3.4	Deep LSTM	500	92.23%	84.78%	93.33%	95.48%	94.74%	92.08%

Tabela 4: Resultados de acordo com as métricas, onde 'Basket.', 'Diving', 'Golf Swing' e 'Skiing' referem-se ao F1-score de cda uma das respectivas classes

6 Conclusão

Neste assignment, comparamos a LSTM (Long Short Term Memory), a GRU (Gated Recurrent Unit) e a Deep LSTM (Deep Long Short Term Memory) utilizando frames de videos para classificação de esportes em quatro classes diferentes. Podemos ver que a GRU é mais simples e demora menos no treinamento que a LSTM, contudo apresenta uma performance melhor considerando a média dos scores F1 e a acurácia.

A Deep LSTM, no entanto, acabou demorando muito tempo no treinamento ao mesmo tempo que não atingia bons resultados, o que pode ser explicada pela sua alta quantidade de parâmetros a ser treinada, que não obteve êxito no treinamento com 100 épocas no máximo.

Temos evidências que, para essa tarefa, a LSTM se sai melhor que os outros métodos testados. No entanto, para uma melhor avaliação, seria interessante testar a convergência dos modelos com um número maior de épocas.