Classificação de instrumentos musicais utilizando Redes LSTM

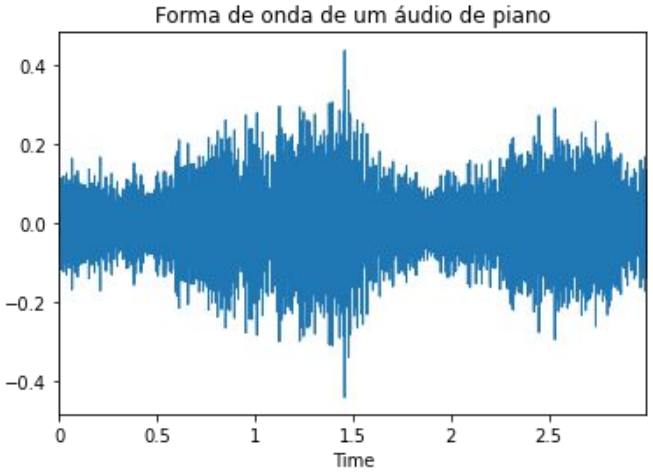
Cainã Figueiredo Pereira
Raul Baptista de Souza
Felipe Martins Fernandes de Assis
Felipe Schreiber Fernandes

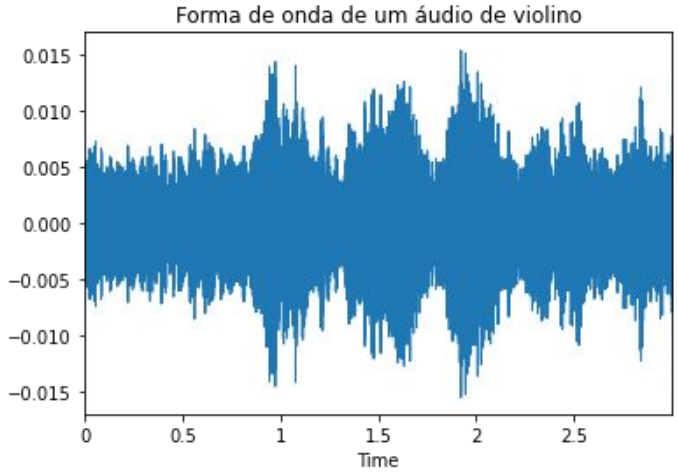
Conjunto de Dados

- IRMAS (Instrument Recognition in Musical Audio Signals) [2]
- Áudios com cerca de 3s com anotações do instrumento predominante;
  - Piano (721 áudios);
  - Violino (580 áudios);

Conjunto de Dados

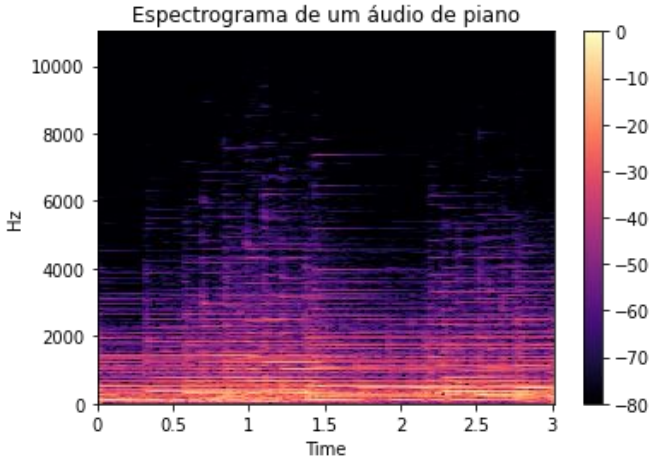
- IRMAS (Instrument Recognition in Musical Audio Signals) [2]
- Áudios com cerca de 3s com anotações do instrumento predominante;
  - Piano (721 áudios);
  - Violino (580 áudios);

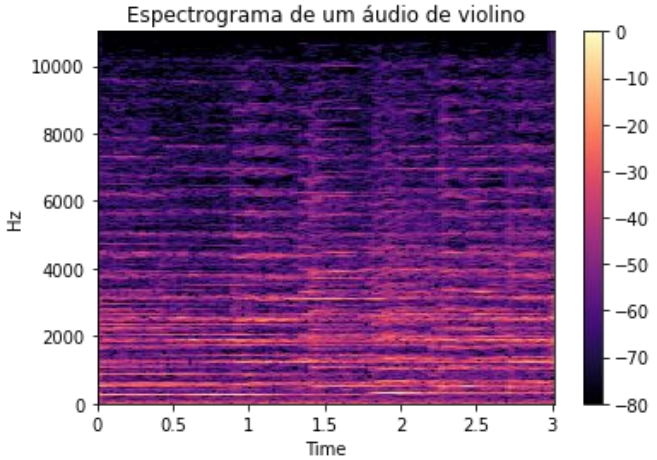




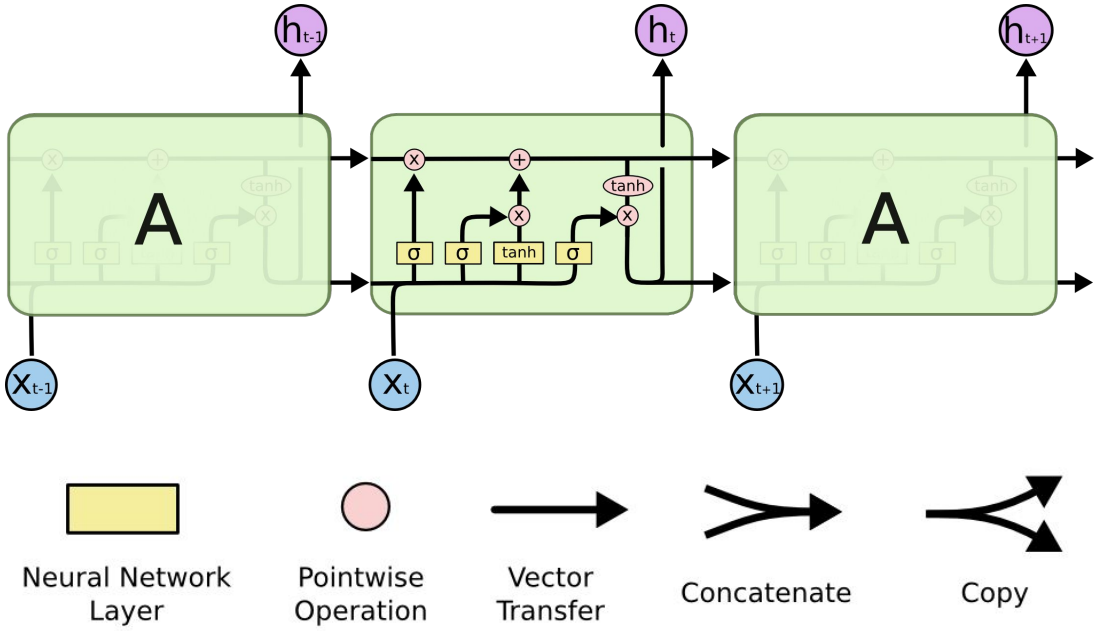
Conjunto de Dados

- IRMAS (Instrument Recognition in Musical Audio Signals) [2]
- Áudios com cerca de 3s com anotações do instrumento predominante;
  - Piano (721 áudios);
  - Violino (580 áudios);

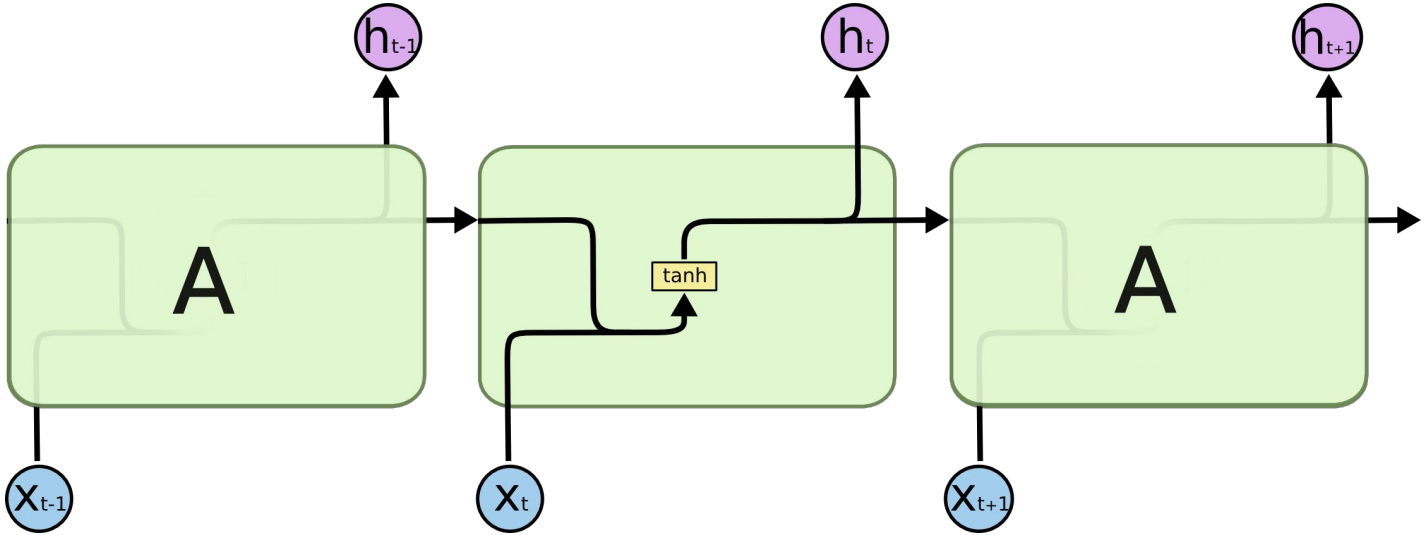




Long Short Term Memory (LSTM)

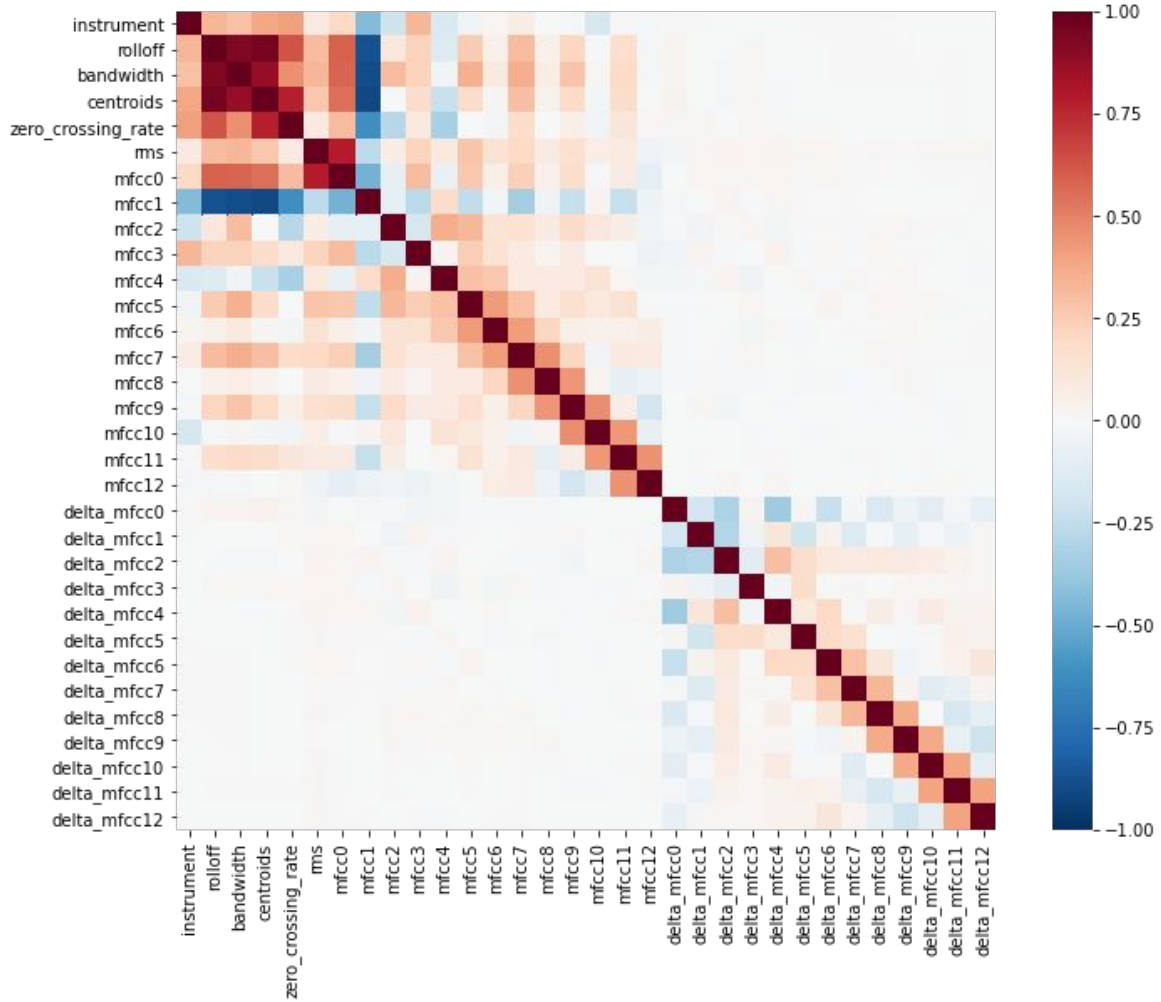


Recurrent Neural Networks (RRN)



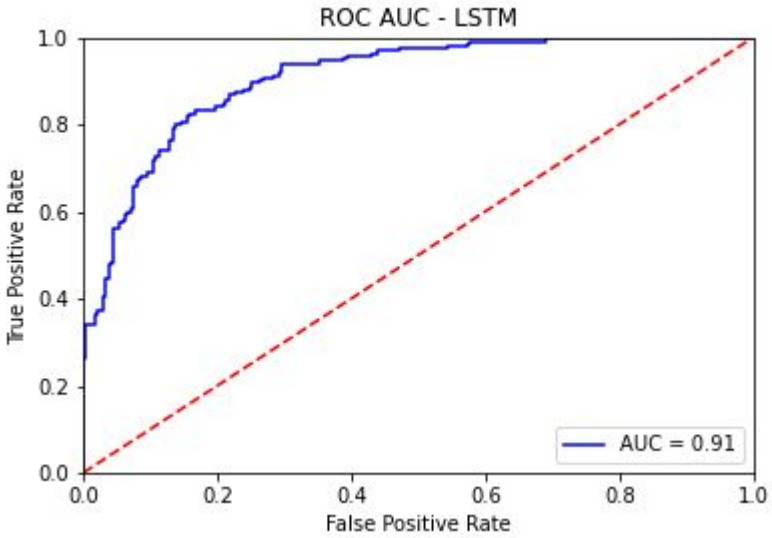
Pré-Processamento

- Áudios carregados pela biblioteca librosa
  - Normalização das amplitudes
  - Taxa de amostragem convertida de 44.1 kHz para 22.05 kHz
  - Conversão para áudios mono
- Features extraídas através da biblioteca:
  - Dois tamanhos de blocos diferentes: 93ms e 23ms
  - Spectral Centroid
  - Spectral Bandwidth
  - Spectral Rolloff
  - Zero-Crossing Rate
  - RMS Energy (Root Mean Square Energy)
  - MFCC (Mel-Frequency Cepstral Coefficients) - 13 coeficientes
  - Deltas do MFCC

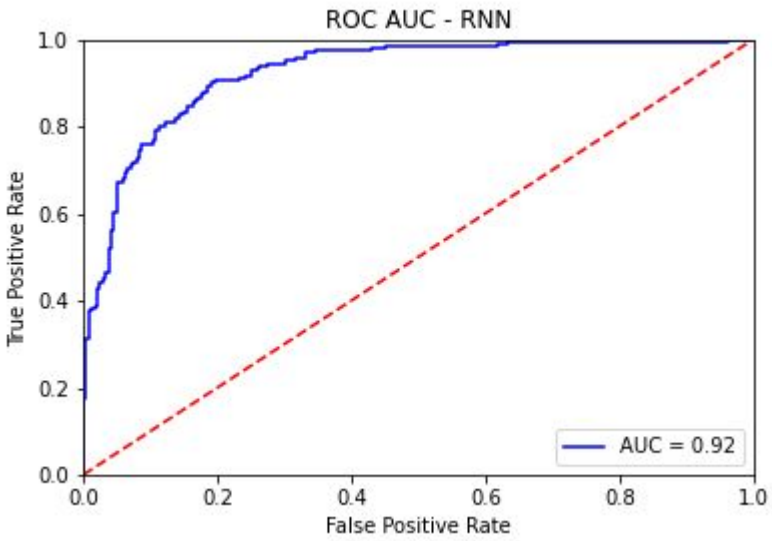

instrument
rolloff
bandwidth
centroids
zero_crossing_rate
rms
mfcc0
mfcc1
mfcc2
mfcc3
mfcc4
mfcc5
mfcc6
mfcc7
mfcc8
mfcc9
mfcc10
mfcc11
mfcc12
delta_mfcc0
delta_mfcc1
delta_mfcc2
delta_mfcc3
delta_mfcc4
delta_mfcc5
delta_mfcc6
delta_mfcc7
delta_mfcc8
delta_mfcc9
delta_mfcc10
delta_mfcc11
delta_mfcc12
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Resultados - Sem Novas Features (93ms)

LSTM:



RNN:



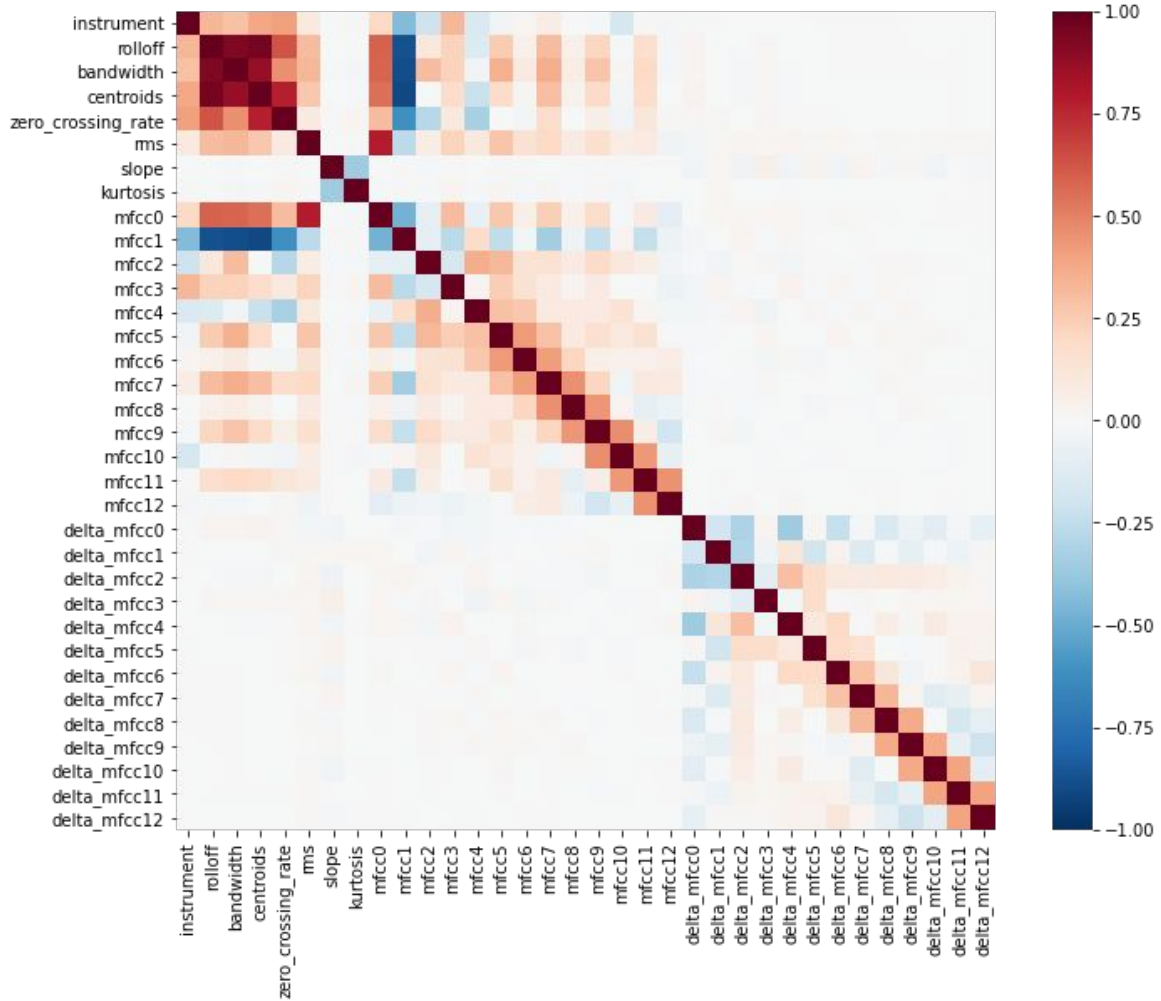
Extração de Novas Features

- Spectral kurtosis: indica a “planicidade” ou o “pico” da distribuição da energia

$$\gamma_2 = \frac{1}{\sigma^4} \int (f - \mu)^4 . p(f) df$$

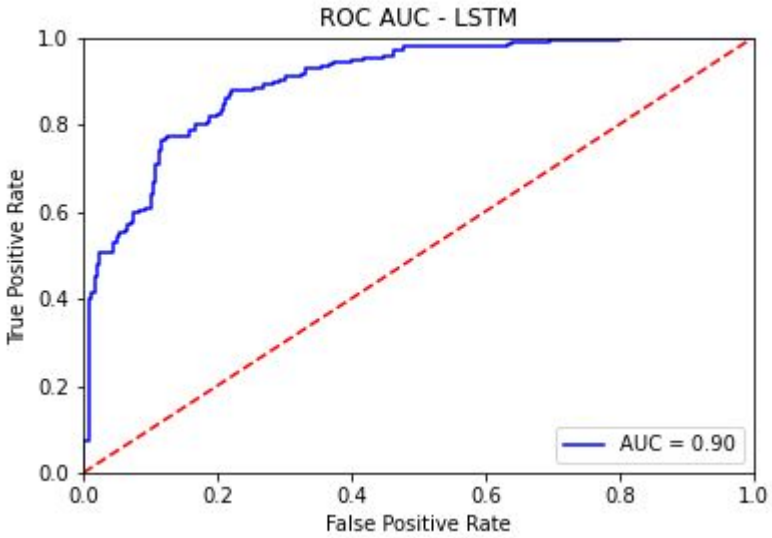
- Spectral slope: indica o quão rápido o espectro vai em direção à altas frequências. Também dá uma indicação da taxa de decrescimento do amplitude

$$m = \frac{1}{\sum_f A(f)} \frac{N \sum_f f.A(f) - \sum_f f \times \sum_f A(f)}{N \sum_f f^2 - \left(\sum_f f \right)^2}$$

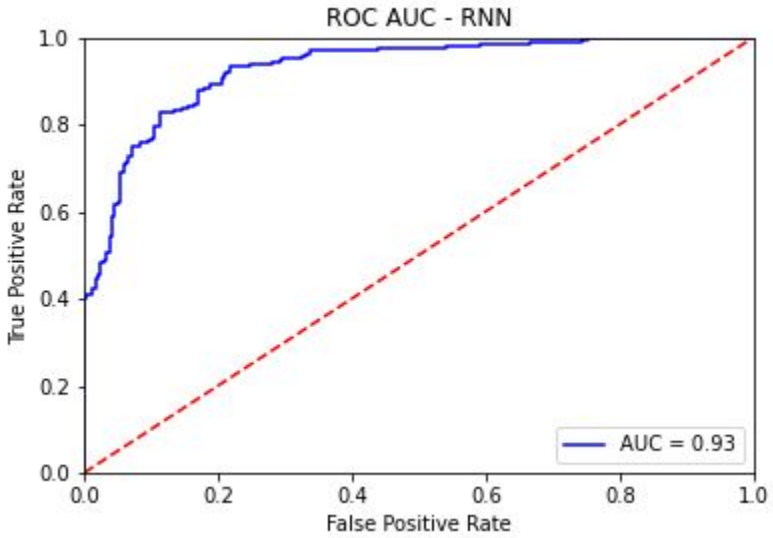

instrument
rolloff
bandwidth
centroids
zero_crossing_rate
rms
slope
kurtosis
mfcc0
mfcc1
mfcc2
mfcc3
mfcc4
mfcc5
mfcc6
mfcc7
mfcc8
mfcc9
mfcc10
mfcc11
mfcc12
delta_mfcc0
delta_mfcc1
delta_mfcc2
delta_mfcc3
delta_mfcc4
delta_mfcc5
delta_mfcc6
delta_mfcc7
delta_mfcc8
delta_mfcc9
delta_mfcc10
delta_mfcc11
delta_mfcc12
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

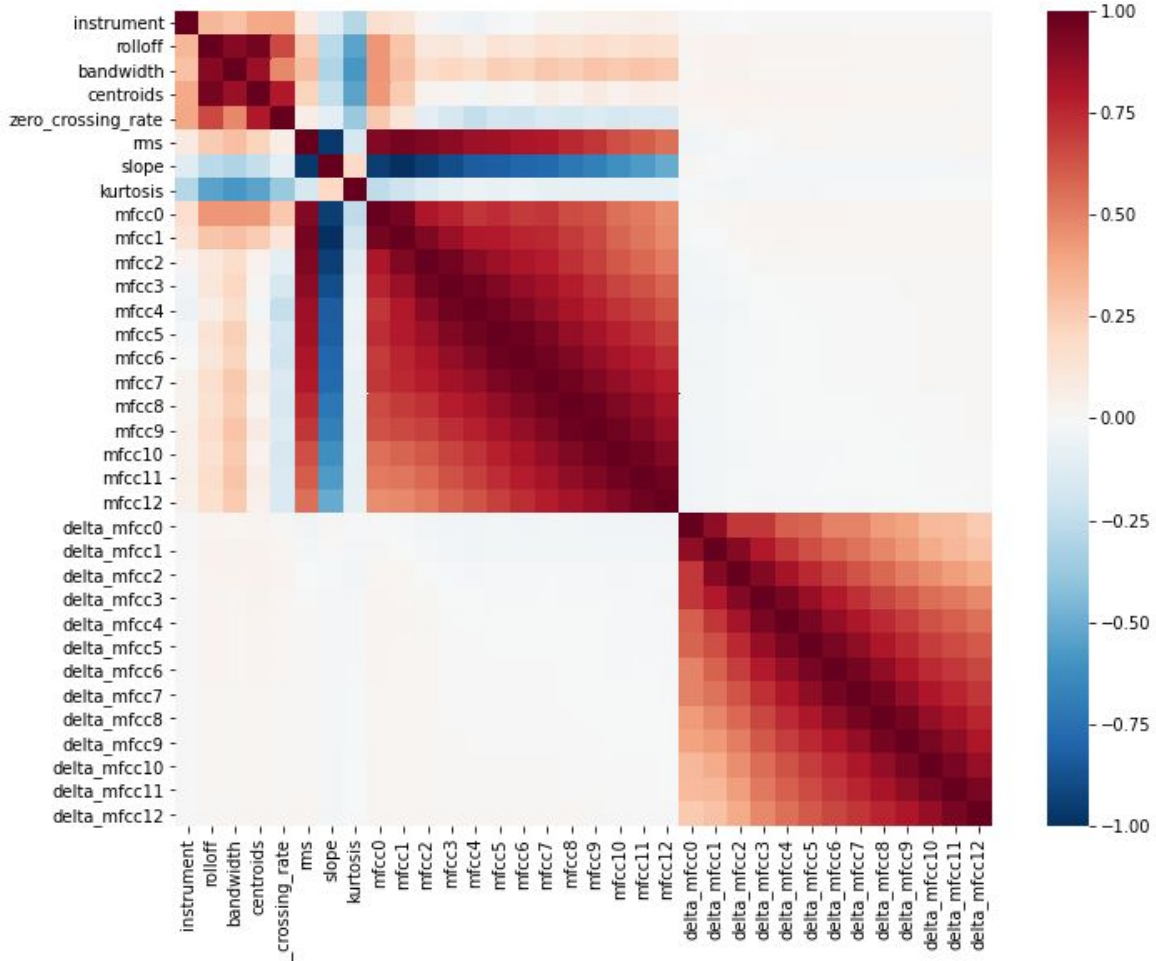
Resultados - Com Novas Features (93 ms)

LSTM:



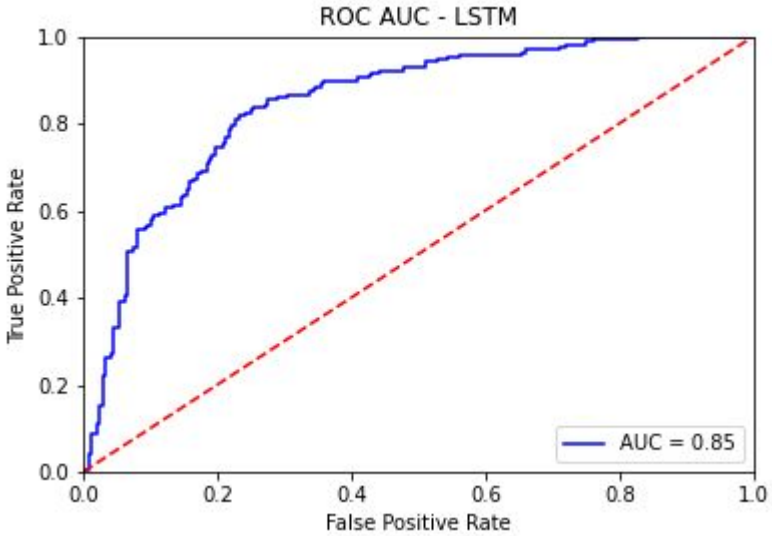
RNN:



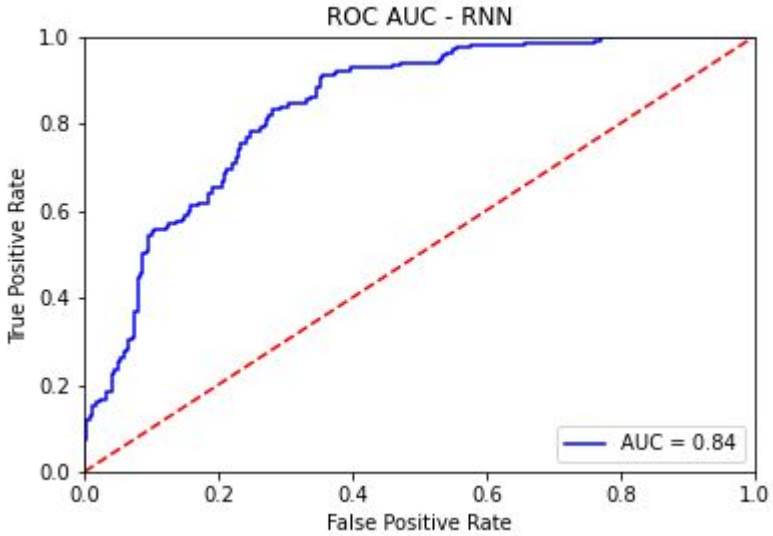

instrument
rolloff
bandwidth
centroids
zero_crossing_rate
rms
slope
kurtosis
mfcc0
mfcc1
mfcc2
mfcc3
mfcc4
mfcc5
mfcc6
mfcc7
mfcc8
mfcc9
mfcc10
mfcc11
mfcc12
delta_mfcc0
delta_mfcc1
delta_mfcc2
delta_mfcc3
delta_mfcc4
delta_mfcc5
delta_mfcc6
delta_mfcc7
delta_mfcc8
delta_mfcc9
delta_mfcc10
delta_mfcc11
delta_mfcc12
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Resultados- Sem Novas Features (23 ms)

LSTM:

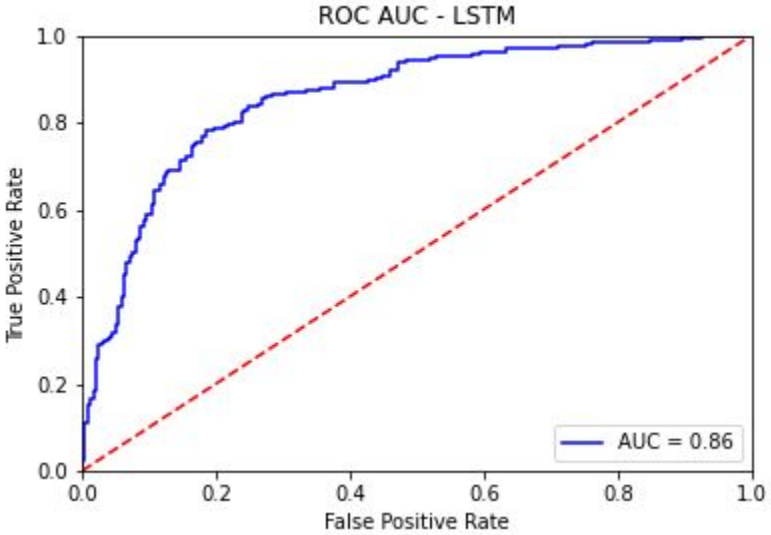


RNN:

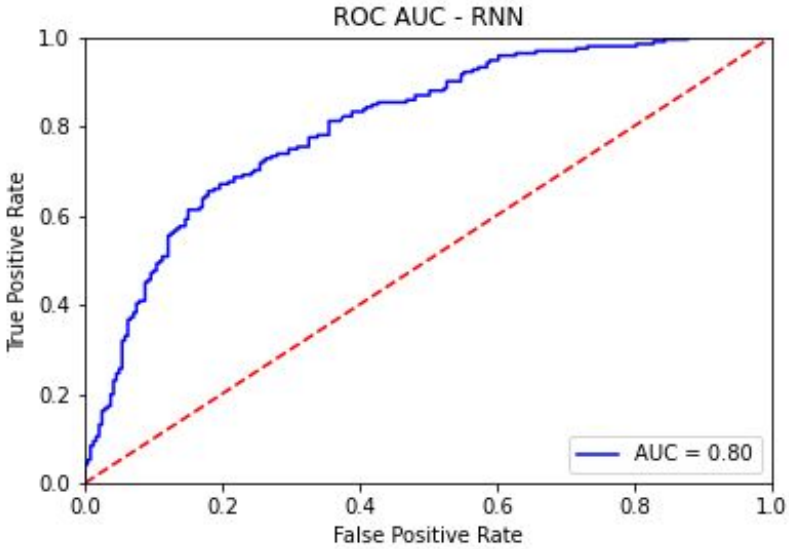


Resultados - Com Novas Features (23 ms)

LSTM:



RNN:

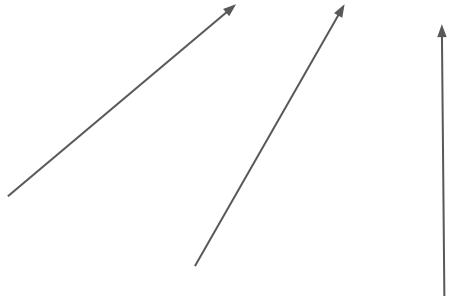


Modelo da rede neural

```
def get_LSTM_Model(input_shape):
        model = Sequential()
         model.add(layers.LSTM(128, return_sequences=True,input_shape=input_shape))
         model.add(layers.LSTM(128, return_sequences=True))
         model.add(layers.Dropout(0.8))
         model.add(layers.TimeDistributed(layers.Dense(64,activation='relu')))
         model.add(layers.TimeDistributed(layers.Dense(32,activation='relu')))
         model.add(layers.TimeDistributed(layers.Dense(16,activation='relu')))
         model.add(layers.TimeDistributed(layers.Dense(8,activation='relu')))
         model.add(layers.Flatten())
         model.add(layers.Dropout(0.5))
         model.add(layers.Dense(32,activation='relu'))
         model.add(layers.Dense(16,activation='relu'))
         model.add(layers.Dense(8,activation='relu'))
         model.add(layers.Dense(1,activation='sigmoid'))
         model.summary()
         model.compile(optimizer=opt,loss='binary_crossentropy')
         return model
```

Quantidade de parâmetros:
LSTM - 357,561
RNN - 197,433

Entrada:
X.shape = (1301,130,32)



#Arquivos, ou batch_size {32}

#Blocos/Audio (Tamanho da sequência) {517,130}

#Características/ Bloco {32,34}

Resumo Resultados

Rede Neural	SF / 93 ms	SF / 23 ms	CF / 93 ms	CF / 23 ms
LSTM	**0.91**	0.85	0.90	0.86
RNN	0.92	0.84	**0.93**	0.80

Conclusão:

1. Para um mesmo conjunto de características:
    - Maior a sequência, pior os modelos, mas o modelo RNN foi o mais afetado
    - Sequência maior -> LSTM melhor
    - Sequência menor-> RNN melhor

2. Modelos são menos sensíveis à adição de novas características, porém sensíveis ao tamanho da sequência

3. Na média, LSTM melhor:
    LSTM 0.88000 +/- 0.00065
    RNN 0.87250 +/- 0.00297

Legenda:
SF -> Sem as novas features
CF -> Com as novas features

Referências Bibliográficas

[1] Understanding LSTM Networks. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Access: October 09, 2020.

[2] IRMAS: a dataset for instrument recognition in musical audio signals. <https://www.upf.edu/web/mtg/irmas>. Access: October 09, 2020

[3] Bosch, J. J., Janer, J., Fuhrmann, F., & Herrera, P. "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals", in Proc. ISMIR (pp. 559-564), 2012

[4] MCKINNEY, Martin; BREEBAART, Jeroen. Features for audio and music classification. 2003.

[5] Audio Data Analysis Using Deep Learning with Python (Part 1). <https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html> Access: October 13, 2020.

[6]Chandwadkar, D.M; Sutaoneg, M.S. Proper Features and Classifiers for Accurate Identification of Musical Instruments, 2013

[7]Audio Content Analysis. <https://www.audiocontentanalysis.org> Access: November 11, 2020.