

# Explorando Redes de Múltiplas Camadas para Classificação de Modulações de Sinais de Rádio

Roger Danilo Figlie - [rogerfiglie@hotmail.com](mailto:rogerfiglie@hotmail.com) - RA:189957

Thiago Arruda Navarro do Amaral - [navarro.ime@gmail.com](mailto:navarro.ime@gmail.com) - RA:159121

**Resumo** – Este trabalho tem por objetivo explorar o artigo [1] que aborda redes para classificação de modulações de 11 sinais de rádio. Esta temática é importante para cenários onde o receptor não sabe a priori que tipo de sinal o transmissor está emitindo, como, por exemplo, em Rede de Rádios Cognitivos e em Guerra Eletrônica. Neste trabalho foram feitas modificações na estruturas das redes *CNN2* e *CNN* apresentadas originalmente em [1], com a finalidade de analisar e comparar a acurácia resultante com as redes originais. Foi possível verificar que técnicas, *Dropout* e *MaxPooling*, e estruturas baseadas em redes *MLP* e na rede Inception pouco influenciaram na acurácia da rede *CNN*. Já na rede *CNN2*, a técnica de *Dropout* apresentou uma melhora considerável na acurácia do classificador.

**Palavras-chave:** modulações de rádio, ruído, redes convolucionais, problema de classificação.

## 1. Introdução

Este trabalho tem por objetivo revisitar o artigo [1] que explora o tema de classificação de modulações de sinais de rádio. O trabalho se baseou no dataset [3] que contém amostras de sinais de áudio separados por SNR e rótulo da classe de modulação do sinal. Tal dataset já foi explorado anteriormente por outros pesquisadores, de forma a estudar o comportamento de alguns tipos de redes neurais para o mesmo problema. Em um primeiro momento, serão implementadas algumas das redes originais do artigo [1] e em seguida serão realizados testes com basicamente dois intuitos distintos: entender como alguns dos hiperparâmetros e camadas foram selecionados e experimentar estruturas diferentes de redes neurais.

## 2. O Problema

O problema abordado com o dataset disponível neste relatório é o de classificação de modulações de sinais de rádio. Hoje em dia quase todos os sistemas de comunicação são desenvolvidos com a premissa de que o transmissor e o receptor sabem, a priori, qual tipo de forma de onda será usada. Entretanto, existem cenários em que essa premissa não é válida. Um exemplo clássico é em redes que suportam rádios cognitivos. Os novos terminais que entram na rede não necessariamente precisam saber que tipo de forma de onda está sendo utilizada. Além disso, existem aplicações de guerra eletrônica que visam varrer o espectro em busca de informações e padrões que podem ser utilizados para extrair ou dificultar a comunicação de um inimigo, por exemplo.

## 3. Os Dados

O dataset que utilizamos, disponível em [3], é um dataset público e é constituído por amostras de sinais de rádio sintéticos gerados conforme [2]. O dataset utilizado neste trabalho é diferente do usado originalmente no artigo [1] sendo, segundo o site, uma versão com menos amostras do dataset utilizado originalmente. O dataset está dividido em 11 modulações de rádio diferentes, cada uma dessas modulações de rádio está subdividida em 20 categorias de relação sinal-ruído variando de -20 a 18. Cada uma dessas modulações e relações sinal-ruído possui 1000 amostras totalizando 220000 amostras, balanceadas em relação aos rótulos. Ainda conforme [2] cada uma das amostras é uma matrix 2 x 128 na qual a dimensão 2 representa os dois canais I e Q do sinal de áudio e a dimensão 128 representa a dimensão temporal (corresponde a uma janela de 128  $\mu$ s). Desta forma cada amostra de áudio é constituída por 256 atributos. O dataset não estava standarlizado em relação a cada um desses atributos e portanto ele foi standarlizado e em seguida foi separado em um conjunto de treinamento e outro de teste com respectivamente 70% e 30% dos dados, mantendo as proporções dos dados entre modulações e relações sinal ruído.

## 4. Revisitando o artigo

No artigo [1] foram modeladas vários tipos de redes para classificações de modulações de rádio. Dentre elas, as que melhor apresentaram desempenho foram as redes do tipo convolucional denominadas *CNN2* e *CNN*. Tais redes têm em comum o número e tipo de camadas, o tamanho dos kernels e o stride, conforme a figura 1, mas são diferentes no número de kernels e

neurônios e uso de regularização conforme tabela.

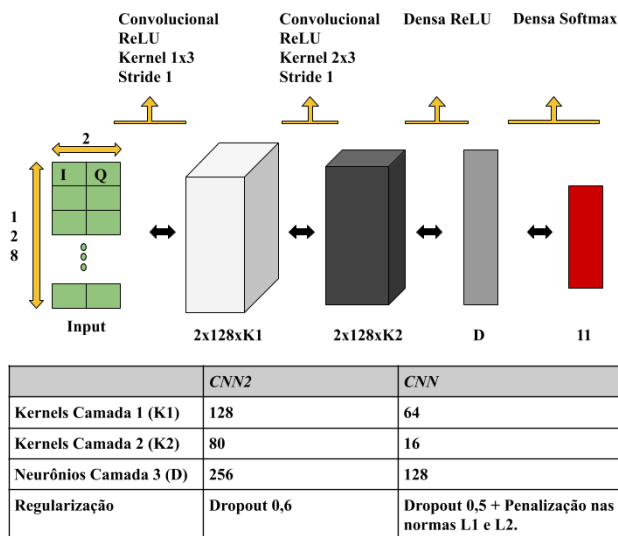


Figura 1 - Modelo das redes *CNN2* e *CNN* do no artigo.

O trabalho foi iniciado implementando as duas redes, com a diferença de que, na *CNN*, foi utilizada apenas a regularização por *Dropout*. O principal motivo de remover a regularização por penalização da norma foi para reduzir o custo computacional dos treinamentos. Esse custo está relacionado ao algoritmo *Busca em Grade*, utilizado para encontrar valores ótimos dentro de um conjunto busca. Esses valores representam a ponderação das normas para minimizar a função de custo. O *Dropout* é utilizado para evitar overfitting em redes neurais, desligando aleatoriamente pesos dos neurônios. Desta forma implementamos rede *CNN2* sem o *Dropout* para verificar se a rede apresentaria *overfitting*.

Essas redes foram treinadas por 200 épocas, o que foi suficiente para que a acurácia se estabilizasse com o otimizador Adam (o mesmo utilizado em [1]), e como rede final foi selecionada a que obteve melhor acurácia de teste durante o treinamento.

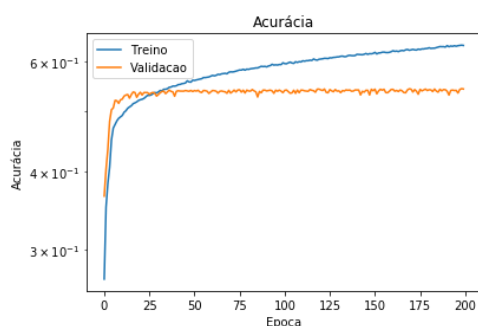


Figura 2 - Gráfico da evolução da acurácia durante o treino com rede *CNN2* com *Dropout* 0,6.

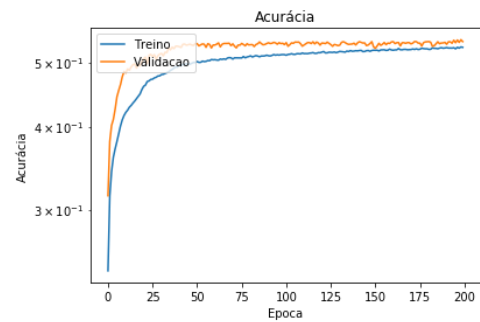


Figura 3 - Gráfico da evolução da acurácia durante o treino com rede *CNN* com *Dropout* 0,5.

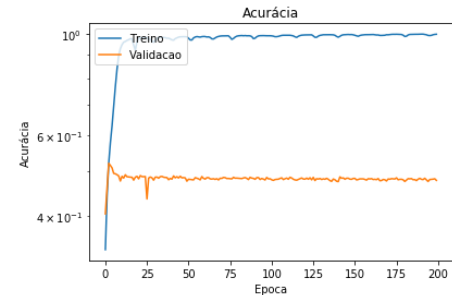


Figura 4 - Gráfico da evolução da acurácia durante o treino com rede *CNN2* sem *Dropout*.

Assim, é possível verificar nas figura 2, 3 e 4 a convergência das redes a medida que o número de épocas vai aumentando, atingindo o valor máximo de acurácia média de 54,82%, 52,06% e 54,10% no conjunto de teste para as redes *CNN2* com *Dropout*, *CNN2* sem *Dropout* e *CNN* respectivamente.

Também obteve-se um gráfico de acurácia por cada SNR muito similar ao do artigo [1]:

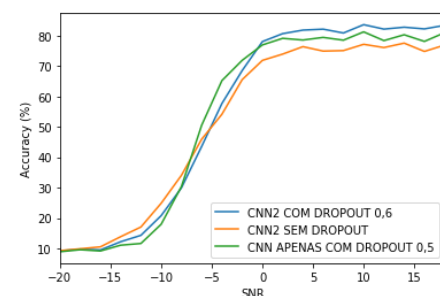


Figura 5 - Gráfico da acurácia vs SNR

Assim, observa-se que, para SNRs abaixo de -10, ambas as redes apresentam um desempenho similar e inferior a 20%. Já para SNRs maiores que 0, foi atingido um platô nas redes, mas com um desempenho ligeiramente melhor na *CNN2*: acima dos 80%. Entretanto, o treino da rede *CNN2* é quase 5 vezes mais demorado que o da *CNN*, tornando inviável demais testes com base na arquitetura da *CNN2* com os

hardwares disponíveis. Logo, os próximos testes serão realizados com base na rede *CNN*.

## 5. Explorando Outras Idéias

A partir da rede *CNN* com *Dropout* que teve uma acurácia de 54,10%, foram exploradas algumas variações. A primeira delas foi adicionar duas camadas *MaxPooling*, em amarelo, conforme figura abaixo:

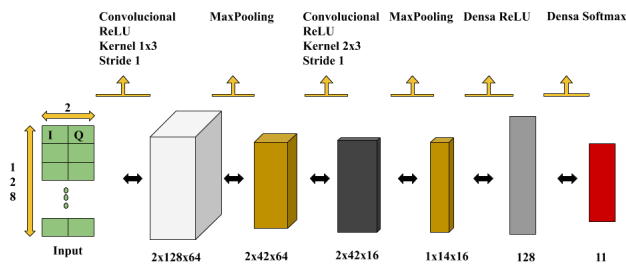


Figura 6 - Gráfico da acurácia vs SNR

Em seguida, foram feitos dois testes da *CNN*: o primeiro utilizando as camadas de *MaxPooling* sem o *Dropout* e o segundo com o *MaxPooling* e com *Dropout*.

A idéia de introduzir a camada de *MaxPooling* é tornar a rede mais invariante a deslocamentos no tempo dos sinais de áudio e também diminuir o número de pesos a serem treinados. Com esses teste, foram obtidas uma acurácia de 53,99% para a rede apenas com *MaxPooling* e 51,88% para a rede com *MaxPooling* e *Dropout*.

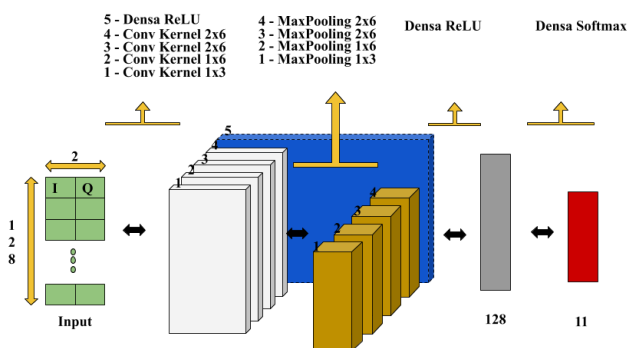


Figura 7 - Arquitetura da rede com blocos em paralelo, o bloco em azul é o bloco denso utilizado no segundo teste.

Outra idéia implementada foi a de utilizar vários kernels em paralelo na mesma camada (idéia essa inspirada na rede *Inception* do Google [5]), seguido de uma camada densa. A idéia nesta arquitetura de rede é usar vários kernels em paralelo na primeira camada de

forma com que a própria rede escolha quais kernels utilizar. Nesta rede, foi utilizado o *MaxPooling* para diminuir o número de parâmetros a serem treinados. Seguindo essa idéia, foram feitos também dois testes: o primeiro deles com 4 blocos convolucionais na primeira camada seguido de *MaxPooling* em cada bloco convolucional, uma camada Densa ReLU e uma Densa Softmax. O segundo teste foi feito adicionando também um bloco denso ReLU na primeira camada. A ideia do bloco denso é pegar informações pontuais que possam ajudar na classificação das amostras. Em ambas as redes, foi utilizado *Dropout* de 0,5 antes de entrar na camada densa para evitar o *overfitting* (ver figura 6).

Com os blocos convolucionais, foi obtido uma acurácia média de teste de 52,07% e com a adição do bloco denso 52,42%. Com a realização desses testes obtemos a figura 8.

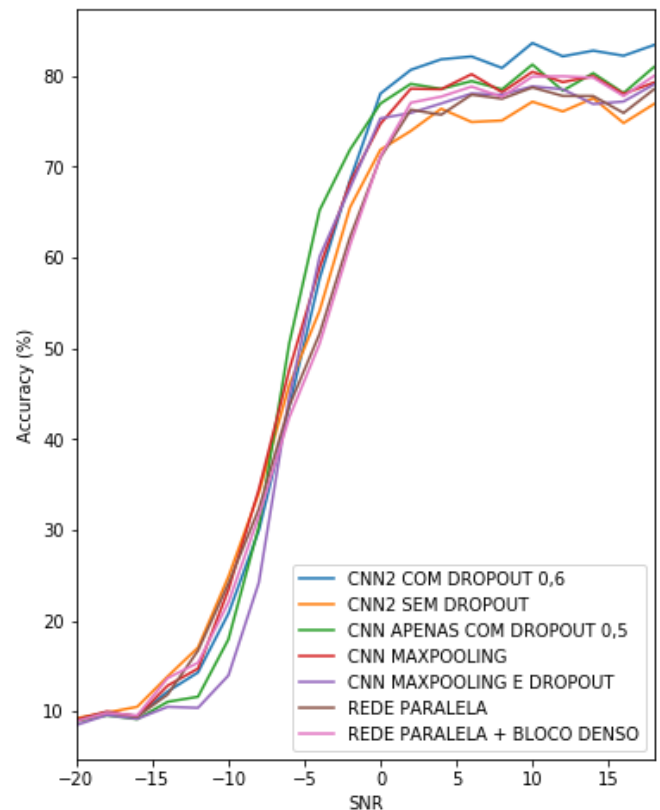


Figura 8 - Composição dos gráficos SNRs vs Acurácia de todos os modelos explorados.

Em SNRs abaixo de -15, todas as redes se comportaram basicamente da mesma maneira tendo uma baixa acurácia. Acima de -15 e abaixo de 0, já é possível notar que a *CNN* com *MaxPooling* e *Dropout* tem uma performance inferior. Especificando um pouco mais a faixa de -5 a 0, é particularmente interessante por mostrar que a rede *CNN* com *Dropout* ganha em

desempenho da rede *CNN2* que tem quase o dobro de tamanho. Já para *SNRs* positivos, a rede *CNN2* com *Dropout* é a que melhor que classifica os dados, o que faz sentido devido a seu tamanho se comparado às *CNNs* que são bem menores.

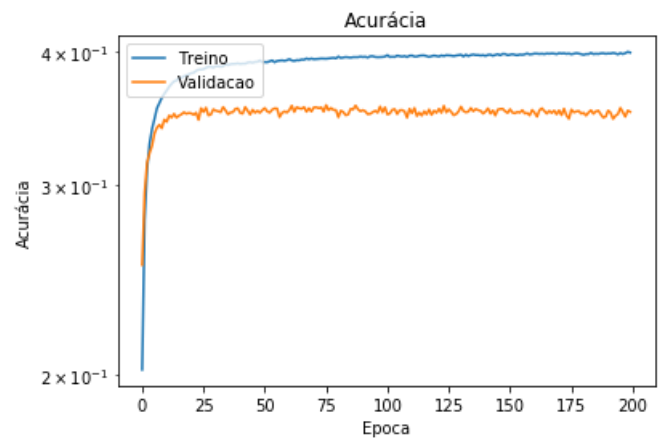
Outro ponto interessante é notar a importância do *Dropout*, pois a *CNN2* sem o *Dropout* possui um desempenho inferior a *CNN*. Além disso, a *CNN* com *Dropout* e a *CNN* com *MaxPooling* têm performance muito similar em todos os *SNRs*.

O último teste foi implementar uma *MLP* com uma camada intermediária e com função de ativação *ReLU*. Os demais parâmetros foram idênticos aos utilizados nas redes *CNN*. Para ver como essa rede se comporta, foi variado primeiramente o número de neurônios e, em seguida, foi adicionada a camada de *Dropout* com diferentes valores para verificar seria possível obter uma rede melhor. Essas redes também foram treinadas com 200 épocas o que foi suficiente para que a acurácia estabilizasse:

Número Neurônios	Acurácia de Treino	Acurácia de Validação
64	39,80	35,63
128	44,92	35,76
256	55,37	35,73
512	53,22	36,23
1024	94,47	37,63
2048	98,15	40,15

**Tabela 1 - Tabela com os valores da acurácia de treino e de validação em função do número de neurônios na rede *MLP*.**

Tal acurácia foi acompanhado de um claro overfitting, pois nessas redes a acurácia de treino ficou acima dos 90%. A melhor rede foi a com 2048 neurônios (ver figura 9).



**Figura 9- Gráfico de treino da MLP com 2048 neurônios.**

Agora usando *Dropout* na MLP com 2048 neurônios:

<i>Dropout</i>	Acurácia Treino	Acurácia Teste
0,2	59,85	36,92
0,4	42,61	37,48
0,6	31,25	37,09
0,8	23,30	32,78

**Tabela 2 - Tabela com os valores da acurácia de treino e de validação em função do valor de *Dropout* na rede *MLP*.**

Como não foi obtido um desempenho satisfatório com as *MLPs* não prosseguimos com seus testes.

## 6. Conclusão

O trabalho em questão explorou algumas variações de redes usadas para classificação de modulações de sinais de rádio. Foi possível constatar que técnicas como *MaxPooling* e *Dropout* pouco influenciaram no resultado final das redes baseadas na *CNN*. Por outro lado, com as *CNN2*, o uso da técnica de *Dropout* influenciou no aumento significativo da acurácia final de aproximadamente 10% para *SNRs* maiores que 10. O modelo de rede inspirada na rede *Inception* e *MLP* também não apresentaram resultados melhores que as *CNN* e *CNN2*. Sobre as *MLPs*, apesar de pararmos com seus testes, uma coisa interessante pode ser notada. Para valores muito altos de *Dropout* a rede fica tão instável que tem-se uma rede que classifica melhor no teste do que no treinamento.

Como perspectivas futuras, além de explorar a rede a estrutura da rede *CNN2* do mesmo modo que foi feita com a *CNN*, o trabalho poderia ser direcionado para entender o comportamento da rede ao variar seus hiperparâmetros. Vale lembrar que existe um custo computacional elevado para treinar a rede *CNN2*. Então, adicionar camadas de *MaxPooling* poderia ajudar nessa questão. Ou ainda, poderiam ser usadas plataformas de processamento dedicadas (e.g. TPU) para realizar as operações tensoriais, melhorando a performance durante a fase de treino.

Por fim, outro trabalho que poderia ser explorado é realizar um ensemble de redes, visto que dependendo do SNR temos redes diferentes com melhor acurácia.

## Referências

- [1] Convolutional Radio Modulation Recognition Networks Timothy J. O'Shea<sup>1</sup>, Johnathan Corgan<sup>2</sup>, and T. Charles Clancy<sup>1</sup>
- [2] Radio Machine Learning Dataset Generation with GNU Radio Tim O'Shea OSHEA@VT.EDU Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA
- [3] Dataset disponível em <http://opendata.deepsig.io/datasets/2016.10/RML2016.10a.tar.bz2> (acessado 30 de novembro de 2019)
- [4] DeepSig Inc. Reinventing Wireless with Deep Learning, <https://www.deepsig.io/> (acessado em 30 de novembro de 2019)
- [5] Tutorial sobre Rede Inception <https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202> (acessado em 30 de novembro de 2019)
- [6] Tutorial rede convolucional 1D. <https://missinglink.ai/guides/keras/keras-conv1d-working-1d-convolutional-neural-networks-keras/> (acessado em 30 de novembro de 2019)
- [7] Human activity recognition <https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/> (acessado em 30 de novembro de 2019)