Radio Modulation Classification Using Deep Learning Architectures

Kristýna Pijáčková, Tomas Gotthans
Department of Radio Electronics
Brno University of Technology
Brno, Czech Republic
xpijac02@stud.feec.vutbr.cz

***Abstract*—Over the past five years, there has been a focus on creating deep learning architectures for radio modulations recognition following the trend, where deep learning (DL) became a state-of-the-art in other fields such as computer vision and voice and natural language processing. The DL models usually contain millions of parameters and so do the models for the radio modulation classification proposed in papers such as [3] or [8]. Having a large number of parameters enable the network to learn more features from the data, but it also makes the model more complex, demands higher computational power, and results in a bigger size. This paper introduces two deep learning architectures - Convolutional Neural Network (CNN) and Convolutional Long-Short-Term Deep Neural Network (CLDNN) with a reduced number of parameters and comparable results to other research papers.**

***Index Terms*—Radio modulation, classification, neural network, CNN, CLDNN**

I. Introduction

Wireless technology has rapidly developed over the years and became an inseparable part of daily life. Automatic modulation recognition (AMR) is therefore an important part of modern wireless communication systems [1]–[3]. AMR finds its use in civilian applications such as spectrum interference monitoring, dynamic spectrum access, or in military applications to jam adversary signals and protect the friendly ones from jamming.

Common approaches nowadays to AMR are a statistical pattern recognition approach and a decision-theoretic approach. The decision-theoretic approach is mostly composed of three main steps, namely of pre-processing, a key-feature extraction, and a modulation classification. It employs probabilistic and hypotheses arguments testing for a modulation recognition problem. This method provides high classification accuracy, but its high computational power demand has led to the development of feature-based classifiers. Popular classification methods using the decision-theoretic approach are support vector machines, neural networks, decision trees, and methods combining collections of classifiers such as XGBoost.

The statistical pattern recognition approach consists of two main parts, where the first one is feature extraction. One of the suitable features for the classification of both analog and digital modulations are spectral features. High-order statistics and cyclo-stationary moments belong to one of the most used features in digital modulation. The pre-defined features are extracted from the received data. The data is then divided into several subgroups with the help of decision trees until the given modulation is determined.

Recently, deep learning (DL) has become a state of the art in computer vision and natural language processing, thanks to better and faster processing units. This inspired the authors in [3] to introduce deep learning architectures such as a deep neural network (DNN) or a convolutional neural network (CNN) to radio modulation recognition in 2016. Even though this was not the first time using neural networks for classifying modulations as the neural networks were introduced in AMR back in the '90s, it is still worth mentioning, as their results managed to outperform other so far used classification methods. The authors of the mentioned paper also created publicly available RadioML datasets [4] with synthetic modulations and after its publication, more people began to focus on creating deep learning architectures approaching the modulation recognition problem [5]–[8].

II. Dataset

RadioML dataset [9] was chosen to train and validate the proposed architectures. This dataset was chosen as it is open-source and is used in other papers, which use deep learning for the modulation classification.

The proposed architectures were designed and fitted onto the RadioML2016.10b dataset, however, the version RadioML2016.10a is introduced as well to allow more comparison.

A. RadioML2016.10b

This dataset includes 10 modulation types - QPSK, QAM16, QAM64, CPFSK, BFSK, 8PSK, and PAM4 for digital modulations and AM-DSB and WBFM for analog modulations. Signal-to-noise ratio (SNR) varies in a range from -20dB to 18dB and contains 1,200,000 samples. All the data are synthetic and stored as 128 samples long raw I/Q data. They were generated using GNU Radio software. To simulate real-life scenario channel model blocks were added consisting of sample rate offset, center frequency offset, selective fading, and additive white Gaussian noise. As a final step, the data was scaled to unity energy as a prior step to further usage of the dataset in the machine learning field.

B. RadioML2016.10a

This dataset is a smaller version of the previously described RadioML2016.10b dataset. It contains 220,000 samples with the same structure as RadioML2016.10b. The only difference, other than the size, is in modulation types as the 10a version has 11 modulations - QPSK, QAM16, QAM64, CPFSK, BFSK, 8PSK, and PAM4 for digital modulations and AM-DSB, AM-SSB, and WBFM for analog modulations.

III. Proposed Architectures

Both Convolutional Neural Networks as well Recurrent Neural Networks or their combination are commonly used for radio modulation classification in many research papers. It is also common, that these networks contain over a million trainable parameters. This can result in longer training times and bigger size of the final model, which can make it harder to include onto an embedded platform. This paper introduces two architectures with reduced number of parameters and the code for the models is shared in a Github repository [1].

A. Convolutional Neural Network Architecture

CNNs are designed for working with data with grid-like topology. Images data can be seen as a 2D grid of pixels and CNN architectures are currently state of the art in computer vision. Since time-series data can be thought of as a 1D grid structure, designing a CNN architecture is a good starting point for the radio modulation recognition network.

The CNNs consist of convolutional layers, pooling layers, and fully connected (dense) layers. The convolutional layer can only see a few neighboring data-points at a time and the size of this area is given by a chosen kernel size. It allows the network to see both low-level and high-level features, which are then activated by an activation function. The pooling layer reduces the size of the data passed through it, mostly either by finding a maximum value or by calculating an average value of a chosen target area. The fully connected layer is used at the end of the network to map the features into labels.

The final CNN architecture starts with a zero padding of size four, three convolutional layers, of which the first two have a filter size of 50 and a kernel size of 8 and are followed by a maximum pooling layer with the kernel size 2. The last convolutional layer has 50 filters as well, but a kernel size of 4. It is followed by a dropout layer with a rate of 0.6 and a maximum pooling layer with a size 2. It is then connected by a dense layer with the size of 70 and finally and a final dense layer activated by a Softmax function for a classification purpose. An overview of the architecture can be seen in Table I.

[1] https://github.com/KristynaPijackova/Radio-Modulation-Recognition-Networks

TABLE I: Architecture of the CNN model

Layers	**Output dimensions**
Input	128x2
Zero padding layer	136x50
Convolutional layer	129x50
Maximum pooling layer	64x50
Convolution	57x50
Maximum pooling	28x50
Convolutional layer	25x50
Dropout layer	25x50
Maximum pooling layer	12x50
Flatten	600
Dense layer	70
Dense layer (Softmax)	10
Overall accuracy: 61.65%	Parameters: 73,730

TABLE II: Architecture of the CLDNN model

Layers	**Output dimensions**
Input layer	128x2
Convolution layer	121x64
Maximum pooling layer	60x64
LSTM layer	60x64
Dropout layer	60x64
LSTM layer	60x64
Dropout layer	60x64
Flatten	3480
Dense layer (Softmax)	10
Overall accuracy: 64.79%	Parameters: 105,546

B. Convolutional Long-Short-Term Deep Neural Network Architecture

CLDNN combines architectures of a Convolutional Neural Network and a Long-Short-Term-Memory Neural Network. LSTM is a kind of RNN and iterates through elements of the received sequence. LSTM maintains a state with relative information based on what it has already seen during this process, first resetting this state when a new sequence is passed to the network. It also saves information for later during the training and prevents older signals from gradually vanishing.

CLDNN architecture turns a long input of time-domain sequences into shorter representations containing high-level features. This part happens in the convolutional and pooling layers. The output of the CNN part becomes an input for the LSTM layers so it can learn long-term temporal coherence of different modulations. A dropout layer after LSTM layers helps to prevent overfitting.

The proposed CLDNN architecture consists of an input layer, followed by a convolutional layer with 50 filters and 8 kernels and a pooling layer with a pool size of 2. The scaled data after the pooling layer is passed into the first LSTM layer, which has 64 filters and is followed by a dropout layer with a rate of 0.4. After that second LSTM and dropout layer follows with the same parameters. At the end is a fully connected layer with an output size 10 with a Softmax activation function.

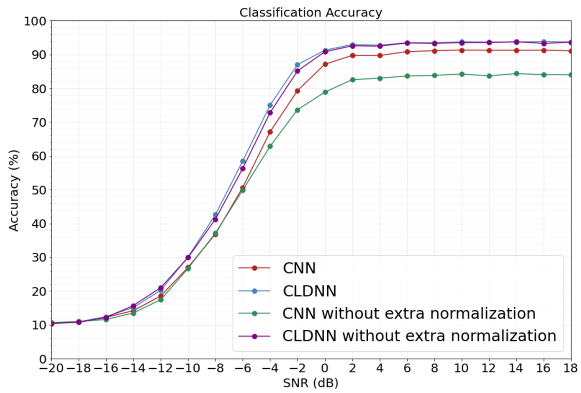


Fig. 1: Accuracy vs SNR - RadioML2016.10b

IV. Experimental Results and Discussion

A. Experimental setup

The architectures are written in Python using a deep learning API Keras [10] utilizing the free GPU offered by Google Colab.

In a pre-processing part, the dataset is split with uniform distribution, 80% of the data are assigned as training data and 20% of the data are preserved for evaluating the network on data it has not seen before. Later on in the code a validation set, using 15% from the training set, is called in Keras fit function. Applying extra feature-wise normalization seems to significantly improve the accuracy of the CNN network and was therefore additionally applied to the data. In this process, each input feature is subtracted with a mean value of the feature and divided by a standard deviation. The mean and standard deviation is computed on the training data only as any computing operation with the test data should be avoided until the validation.

The networks are trained with a batch size of 512 samples and have a starting learning rate of 0.0007. A ReduceLROn-Plateau callback from the Keras API is set to reduce the learning rate size by factor 3 if the validation loss does not improve for three epochs. With an EarlyStopping callback, the training stops if the learning 26 rate does not improve for five epochs and a ModelCheckpoint callback then saves a model with the smallest validation loss [11].

B. Network Recognition Results

1) RadioML2016.10b: Figure 1 shows achieved accuracy of the CNN and the CLDNN models for both normalized data and data without the mentioned normalization. The achieved accuracy of both models lies above 90% for high SNRs. Out of those two models, the CLDNN showed better results at the whole SNR range with an average of around 93% at the 0dB to 20dB SNR range.

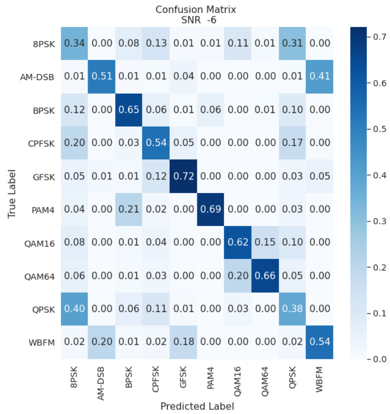
Figure 2 shows an overview of SNR levels at -6dB, 0dB, and 18dB. At -6dB there is a lot of confusion in different classes for both models, however, the main diagonal begins to form. At this SNR both models confuse BPSK and QPSK a lot as well as QAM16 and QAM64, and AM-DSB and WBFM. By the 0db SNR, the confusion between BPSK and QPSK nearly vanishes, which means that both models have found needed features to tell them apart. The confusion between QAM16 and QAM64 sinks by 20% for the CNN model and almost vanishes for the CLDNN model, with only 1-2% of confusion left for each modulation type. The AM-DSB and WBFM misclassification however stays the same for AM-DSB and becomes slightly less for the WBFM. Other than that, all the other modulation types achieve an accuracy of at least 93% for the CNN model and 98-99% for the CLDNN model. When looking at the two confusion matrices with the highest SNR of 18dB, it is to see, that there was an improvement with classifying WBFM, however, the AM-DSB confusion stays at 36% and also the QAMs confusion did not disappear. The CLDNN achieves at least 99% accuracy for all modulation types except for the AM-DSB, which stays at 36% as well. Both WBFM and AM-DSB are analog signals and use a carrier frequency to transfer signals. For the creation of the dataset, analog voice recordings containing pauses were used. When a pause appears in the recording, it doesn't contain any other information for correct classification.

2) RadioML2016.10a: The behavior of the networks on a RadioML2016.10a dataset is to be seen in Figure 3. These models were used on the proposed networks trained on the 10b version of the dataset without any further hyper-parameters tuning. Even though this dataset is almost 5.5 times smaller than the 10b version, it still achieves high accuracy similar to the one from the other dataset. The best accuracy has the CLDNN model, this time however trained on data without any extra normalization, scoring 92.36% at 18dB SNR. This outperforms the CNN model with normalized data by almost 10%.

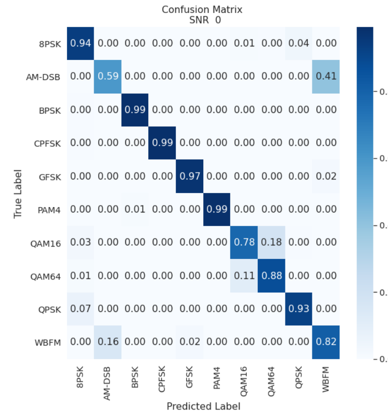
C. Results discussion

The proposed architectures, especially the CLDNN architecture prove to work well on both dataset versions and achieve comparable results to other papers using the same datasets. An overview of achieved accuracies for different conditions can be seen in Table III The accuracies achieved in [3] and [5] were comparable to the accuracy on high SNRs in the range 0dB to 20dB of the proposed CLDNN architecture when performed on the RadioML2016.10a dataset. However, the proposed model was outperformed by 10% to 20% at low SNRs.

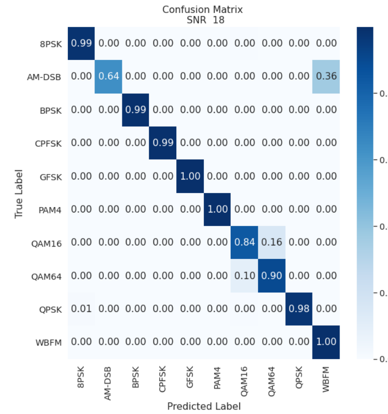
The CLDNN model from this work was able to outperform architectures from [7] and [8] with the dataset RadioML2016.10b. In [7] the author focused on comparing CNN, LSTM, and CLDNN models. The best result achieved by the convolutional network was achieved by architecture with four convolutional layers. The accuracy achieved with this model had scored 86.6% at 2dB SNR and 89.59% at 18dB SNR and were outperformed by the proposed CNN model, which has an accuracy of 89.71% at 2dB SNR and 91.10% at 18dB SNR.



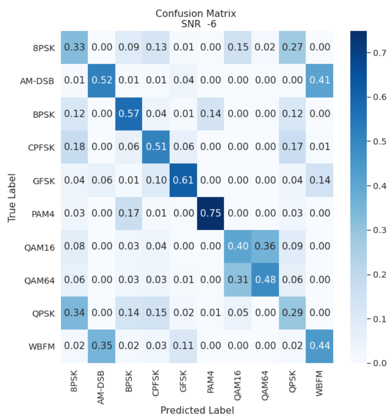
(a) CNN at SNR = -6dB



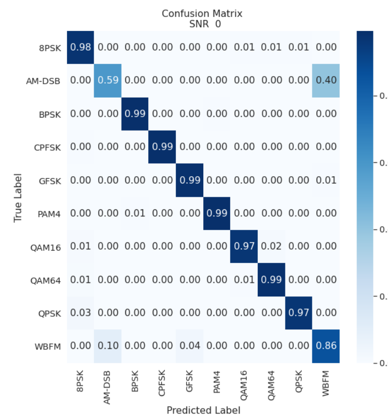
(b) CNN at SNR = 0dB



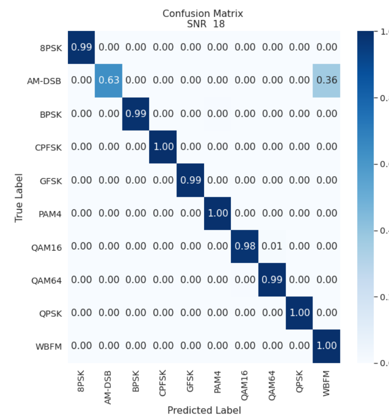
(c) CNN at SNR = 18dB



(d) CLDNN at SNR = -6dB



(e) CLDNN at SNR = 0dB



(f) CLDNN at SNR = 18dB

Fig. 2: Confusion matrices - RadioML2016.10b

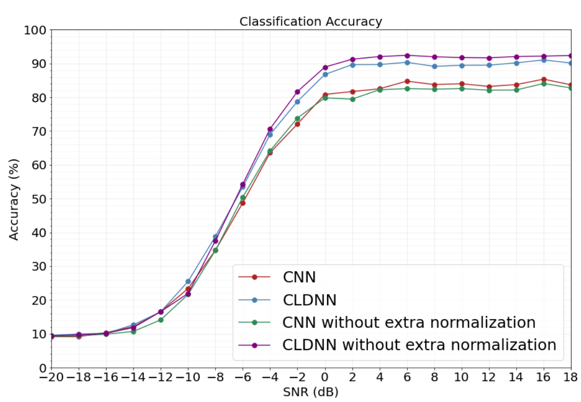


Fig. 3: Accuracy vs SNR - RadioML2016.10a

The CLDNN model showed the best result out of the three proposed architectures but was outperformed as well by the model from this paper by 2.19% at 18dB SNR. The Chosen ResNet model in [8] shows similar results to the CLDNN model in Figure 1, however, the proposed CLDNN architecture shows better results by 1.69%.

TABLE III: Accuracy overview

	RadioML2016.10b				**RadioML2016.10a**			
SNR (dB)	**CNN (%)**		**CLDNN (%)**		**CNN (%)**		**CLDNN (%)**	
Normalized	Yes	No	Yes	No	Yes	No	Yes	No
-6	50.67	49.88	58.45	56.30	48.73	50.45	53.44	54.30
0	87.11	78.84	91.26	90.79	80.84	79.82	86.74	88.92
18	91.10	83.99	93.69	93.64	83.68	82.75	90.13	92.35

V. Conclusion

Two architectures for automatic modulation recognition are presented in this paper - CNN and CLDNN. For designing the architectures, the attention was focused on reducing the number of parameters of the models. The deep learning models for the automatic modulation recognition networks in other papers such as [3], [7] or [8] use over 2 million parameters. A significant reduction of the parameters up to 20 times to the mentioned papers did not affect the resulting accuracy of the networks and in some cases even outperformed them. The proposed CNN architecture has 73,730 trainable parameters and the CLDNN architecture has 105,546 trainable parameters. Reducing the number of parameters enables the network to train faster as well as results in a smaller size of the final model and is therefore desirable for later implementation on an embedded platform.

The hyper-parameters tuning was done with the RadioML2016.10b dataset and the CLDNN model with an accuracy of 93.69% at 18dB outperforms the CNN model by 2.5% on this dataset. Applying the 5.5 times smaller version 10a of the RadioML dataset caused an accuracy drop by roughly 7.5% at the SNR of 18dB for the CNN. The CLDNN seems to be slightly less affected by the smaller size of the dataset and drops only by 3.5%. Even though the accuracy dropped it was still comparable to accuracies from other research papers as mentioned above.

Future work should focus on creating own dataset of real signals and inspecting the behavior of the networks on it. Obtaining a large dataset of real signals might be complicated, but it was shown, that even a smaller dataset can achieve quite a high accuracy. The authors of [12] used transfer learning for training the models on synthetic data and fine-tuning it on the real data. Since obtaining a large labeled dataset might be quite time consuming, this could be a good approach to get a good performance use even with a smaller dataset.

Acknowledgment

This project has been co-financed by the European Union using financial means of the European Regional Development Fund (INTERREG) for sustainable cross boarder cooperation. Further information on INTERREG Austria-Czech Republic is available at url: https://www.at-cz.eu/at.

References

[1] Z. Zhu, A. K. Nandi, "Automatic modulation classification: principles, algorithms, and applications" Hoboken, N.J.: Wiley, 2014. ISBN 978-1-118-90649-1

[2] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Piscataway, NJ, 2017, pp. 1-6, doi: 10.1109/DySPAN.2017.7920754.

[3] O'Shea T.J., Corgan J., Clancy T.C. (2016) Convolutional Radio Modulation Recognition Networks. In: Jayne C., Iliadis L. (eds) Engineering Applications of Neural Networks. EANN 2016. Communications in Computer and Information Science, vol 629. Springer, Cham. Available at: https://doi.org/10.1007/978-3-319-44188-7_16

[4] RadioML dataset from DeepSig: https://www.deepsig.ai/datasets

[5] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders and S. Pollin, "Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors," in IEEE Transactions on Cognitive Communications and Networking, vol. 4, no. 3, pp. 433-445, Sept. 2018, doi: 10.1109/TCCN.2018.2835460.

[6] D. Figueiredo, A. Furtado and R. Oliveiray, "Modulation Classification using Joint Time and Frequency-domain Data," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9128493.

[7] A. Emam, M. Shalaby, M. A. Aboelazm, H. E. A. Bakr and H. A. A. Mansour, "A Comparative Study between CNN, LSTM, and CLDNN Models in The Context of Radio Modulation Classification," 2020 12th International Conference on Electrical Engineering (ICEENG), Cairo, Egypt, 2020, pp. 190-195, doi: 10.1109/ICEENG45378.2020.9171706.

[8] S. Ramjee, S. Ju, D. Yang, X. Liu, A. E. Gamal, Y. C. Eldar Fast Deep Learning for Automatic Modulation Classification, January 2019 Avaible at: https://arxiv.org/abs/1901.05850

[9] T. O'Shea and N. E. West, "Radio Machine Learning Dataset Generation with GNU Radio,"Radio Machine Learning Dataset Generation with GNU Radio," in Proceedings of the GNU Radio Conference, vol. 1, n. 1, September 2016, Avaible at: https://pubs.gnuradio.org/index.php/grcon/article/view/11

[10] F. Chollet "Deep learning with Python" Shelter Island, NY: Manning, 2018. ISBN 9781617294433

[11] Keras API reference: https://keras.io/api/

[12] T. J. O'Shea, T. Roy and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 168-179, Feb. 2018, doi: 10.1109/JSTSP.2018.2797022.