Knowledge Distillation for Modulation Classification in Resource-Constrained Devices

Pedro Marcio Raposo Pereira

Inatel
Santa Rita do Sapucaí, Brasil
pedro.marcio@inatel.br

TP558 - Tópicos Avançados em Aprendizado de Máquina, 28 de Junho de 2024

Table of Contents

1. Introduction
2. Data Preprocessing
3. Network Architecture
4. Knowledge Distillation
5. Results
6. Conclusion and Future Research
7. Bibliography
8. Questions

Table of Contents

1. Introduction
2. Data Preprocessing
3. Network Architecture
4. Knowledge Distillation
5. Results
6. Conclusion and Future Research
7. Bibliography
8. Questions

Introduction

- Military applications for Automatic Modulation Classifier (AMCs) are vital in Electronic Warfare (EW), enhancing situational awareness and enabling quick responses to hostile transmissions.
- Acting as an intermediary between signal detection and system reaction, AMCs identify and classify signals without prior knowledge of parameters like carrier frequency or bandwidth [1].
- This capability is crucial for detecting, locating, and identifying enemy transmissions, facilitating timely and precise jamming or countermeasures.

Classification Algorithms

- Classification algorithms primarily follow two approaches: likelihood-based (LB) and feature-based (FB).
- LB algorithms, or decision-theoretic approaches, optimize classification using the likelihood function of received signals but often face high computational complexity, posing challenges for real-time implementation [2, 3, 4].
- Simplified versions of LB algorithms are typically used to balance computational demands with accuracy.
- FB algorithms use a pattern recognition approach, extracting and processing signal features for classification.
- Despite being sub-optimal, FB algorithms offer simpler implementation and lower computational overhead [3].

Deep Learning in AMC: SBCNN and IBCNN

- Deep Learning (DL) models are recognized for robust classification performance in AMC tasks, adapting to diverse modulations in heterogeneous environments.
- Signal-Based Convolutional Neural Networks (SBCNN) [5] focuses on extracting features from preprocessed signal data using a Convolutional Neural Networks (CNN).
- Extracted features from SBCNN are transformed into images to train the Image-Based Convolutional Neural Networks (IBCNN) model [6].
- Combining SBCNN and IBCNN enhances classification accuracy but increases model size and complexity.
- The double-stage CNN approach demands more computational power and memory, leading to higher latency and potential challenges in resource-limited environments.

Research Proposal

- This work aims to demonstrate the effectiveness of response-based knowledge distillation (KD) in addressing computational constraints in AMC.
- Response-based KD enables training of compact yet accurate models by transferring knowledge from larger, more complex models while maintaining high accuracy [7, 8].
- The proposed single-stage SBCNN exhibits improved performance when trained using teacher distillation compared to standalone training.
- This approach facilitates the creation of AMC models that are better suited for deployment on resource-constrained devices.

Table of Contents

1. Introduction
2. Data Preprocessing
3. Network Architecture
4. Knowledge Distillation
5. Results
6. Conclusion and Future Research
7. Bibliography
8. Questions

Dataset

- The DeepSig Dataset: RadioML 2018.01A [9] is used, containing over 2.5 million frames of wireless communication signals from 24 different modulations.
- OOK, MASK (m={4,8}), MPSK (m={2, 4, 8, 16, 32}), MQAM (m={16, 32, 64, 128, 256}), AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK
- Each frame is a 1024-sample sequence of complex time-series data, with Signal-to-Noise Ratio (SNR) levels ranging from -20 dB to +30 dB in steps of 2 dB.

Data Preprocessing

- Dataset division:
  - 80% for training, 10% for validation, and 10% for testing across SNR levels from 0 dB to 16 dB.
  - Training and validation datasets group all SNR values.
  - Testing datasets are separated by SNR levels for individual performance evaluation.
- Data Preprocessing Pipeline:
  - root-mean-square (RMS) normalization to standardize signal scales, improving model robustness to signal power variations.
  - Data transposition to position real and imaginary parts as outer dimensions.
  - Addition of a third dimension, resulting in a (time, channel, 1) format.

Table of Contents

1 Introduction

2 Data Preprocessing

3 Network Architecture

4 Knowledge Distillation

5 Results

6 Conclusion and Future Research

7 Bibliography

8 Questions

SBCNN Network Architecture

- The SBCNN network architecture, shown in Figure 1, is a convolutional neural network (CNN) composed of seven blocks in sequence:
    - An input layer
    - One A-type block
    - One B-type block
    - Two C-type blocks
    - Global Average Pooling (GAP) layer
    - Dense layer

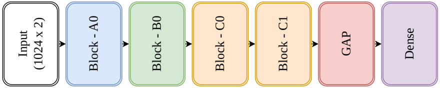

Figure 1: SBCNN Block Diagram.

SBCNN Network Architecture

- Detailed description of each block is presented in Figure 2.
- The A-type, B-type, and C-type blocks consist of Convolutional layers, Batch Normalization layers, and ReLU activation functions.

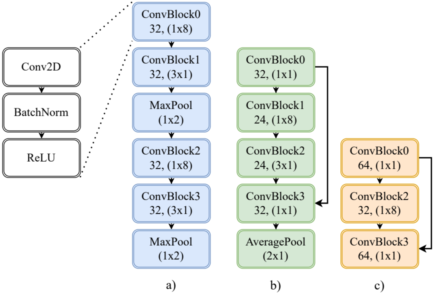

Figure 2: SBCNN Blocks architectures. Blocks a) A0, b) B0, and c) C0 and C1.

IBCNN Network Architecture

- The IBCNN architecture follows a similar design to the SBCNN, as depicted in Figure 3.
- Composed of seven blocks:
  - An input layer
  - One A-type block
  - One B-type block
  - Two C-type blocks
  - Global Average Pooling (GAP) layer
  - Dense layer
  - Dropout layer (20%)
- The addition of a dropout layer is introduced to mitigate overfitting by pruning 20% of hidden units.

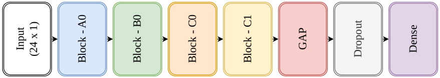

Figure 3: IBCNN Block Diagram.

IBCNN Network Architectur

- Detailed description of each block is presented in Figure 4.
- The A-type, B-type, and C-type blocks include Convolutional layers, Batch Normalization layers, and ReLU activation functions.

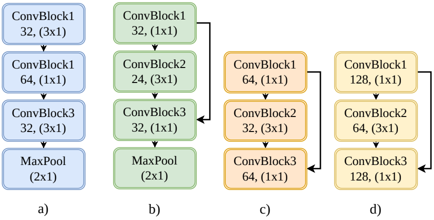

Figure 4: IBCNN Blocks architectures. Blocks a) A0, b) B0, c) C0 and d) C1

Table of Contents

1. Introduction
2. Data Preprocessing
3. Network Architecture
4. Knowledge Distillation
5. Results
6. Conclusion and Future Research
7. Bibliography
8. Questions

Knowledge Distillation (KD)

- KD transfers knowledge from a large, complex model ("teacher") to a smaller, more efficient model ("student"), as illustrated in Figure 5.
- Useful when the teacher model is computationally expensive or impractical for real-world deployment.
- The goal is to enable the student model to achieve comparable performance while being more lightweight and efficient.

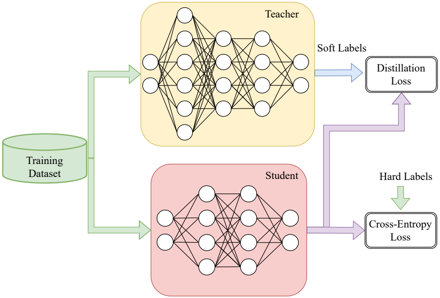

Figure 5: KD training process.

Knowledge Distillation (KD)

- Given a dataset $D = \{x_n, y_n\}_{n=1,\dots,N}$, with x_n as data and $y_n \in [1, K]$ as labels.
- Teacher model f_T and student model f_S generate logit vectors v_n and z_n.
- Logits are transformed into probability distributions using a softmax function with temperature T.
- Minimize the Kullback-Leibler (KL) divergence between the teacher's and student's distributions:
  $L_{KL}(q(v_n)||q(z_n)) = \sum_{k=1}^{K} q(v_n)(k) \log\left(\frac{q(v_n)(k)}{q(z_n)(k)}\right)$.
- Compute the student's cross-entropy loss:
  $L_{CE}(y_n, q(z_n)) = -\sum_{k=1}^{K} y_{n,k} \log q(z_n)(k)$.
- Compute Total loss function:
  $L_{KD}(y_n, q(v_n), q(z_n)) = \alpha L_{CE} + (1-\alpha) L_{KL} T^2$,where α balances the student and distillation losses.

Table of Contents

1. Introduction
2. Data Preprocessing
3. Network Architecture
4. Knowledge Distillation
5. Results
6. Conclusion and Future Research
7. Bibliography
8. Questions

Results

- The student model in this study is implemented using a simplified SBCNN architecture.
  - Composed of a single A-type block, a GAP layer, and a final dense layer for classification.
- Implementation utilized TensorFlow and Keras API.
- All models were trained with:
  - Batch size of 64 samples over 20 epochs.
  - Adam optimizer with a learning rate of 0.001.
  - Best weights saved based on validation accuracy at the end of each epoch.
- Knowledge Distillation Process:
  - teacher IBCNN
  - Parameters for distillation: $\alpha = 0.35$ and $T = 1$.
  - Distillation conducted over 20 epochs with batch size of 64.
  - Adam optimizer with a learning rate of 0.001.
- A second student model trained from scratch using the same setup.

Analysis of Model Performance Across SNR Levels

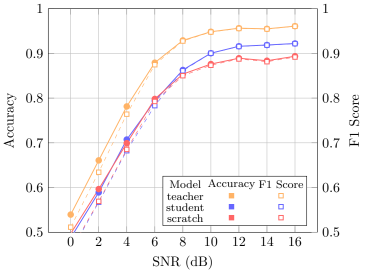

Figure 6: Comparison between models.

- The teacher model exhibits the highest accuracy and F1 score across all SNR levels.
- The student model trained by the teacher shows a significant performance improvement over the student model trained from scratch.
- The student model trained from scratch performs slightly better at lower SNR.

Classification Report

Table 1: Classification Report Comparison of Student, Student Trained from Scratch, and Teacher Models

Class	Student			Student from Scratch			Teacher		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
OOK	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4ASK	0.87	**0.97**	0.91	**0.92**	0.93	**0.93**	0.94	0.95	0.94
8ASK	**0.96**	0.85	0.90	0.93	**0.92**	**0.93**	0.95	0.94	0.94
BPSK	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
QPSK	**1.00**	0.96	**0.98**	0.98	**0.97**	0.97	0.98	1.00	0.99
8PSK	0.96	**0.82**	**0.88**	**0.99**	0.76	0.86	0.89	0.90	0.90
16PSK	**0.98**	0.49	0.65	0.57	**0.90**	**0.70**	0.98	0.63	0.77
32PSK	0.59	**0.96**	**0.73**	**0.83**	0.51	0.63	0.69	0.92	0.79
16APSK	**0.96**	0.83	0.89	0.95	**0.84**	**0.90**	0.94	0.91	0.93
32APSK	**0.92**	0.83	**0.87**	0.64	**0.94**	0.76	0.94	0.89	0.92
64APSK	0.87	**0.55**	**0.68**	**0.88**	0.49	0.63	0.96	0.66	0.78
128APSK	**0.76**	**0.61**	**0.67**	0.64	0.60	0.62	0.79	0.80	0.79
16QAM	0.59	**0.86**	0.70	**0.65**	0.81	**0.72**	0.86	0.86	0.86
32QAM	**0.85**	0.58	**0.69**	0.60	**0.66**	0.63	0.89	0.73	0.80
64QAM	**0.70**	**0.37**	**0.48**	0.66	0.30	0.41	0.94	0.48	0.64
128QAM	0.37	**0.66**	**0.47**	**0.40**	0.51	0.45	0.47	0.81	0.59
256QAM	**0.44**	**0.57**	**0.49**	0.44	0.47	0.45	0.57	0.73	0.64
AM-SSB-WC	**0.73**	**0.82**	**0.77**	0.73	0.78	0.75	0.71	0.92	0.80
AM-SSB-SC	**0.79**	0.69	**0.74**	0.76	**0.70**	0.73	0.89	0.62	0.73
AM-DSB-WC	**0.73**	0.82	0.77	0.72	**0.87**	**0.79**	0.73	0.87	0.80
AM-DSB-SC	0.79	**0.70**	**0.74**	**0.83**	0.66	**0.74**	0.84	0.68	0.75
FM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
GMSK	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
OQPSK	0.93	1.00	0.96	**0.94**	**1.00**	**0.97**	0.97	1.00	0.99
accuracy		**0.79**			0.78			0.85	
macro avg	**0.82**	**0.79**	**0.79**	0.79	0.78	0.77	0.87	0.85	0.85

Classification Report

- The student model, benefiting from KD, generally performs better than the student model trained from scratch across various classes.
- In some cases, such as 4ASK and 8ASK, the student model trained from scratch shows better performance.
- This suggests that while knowledge distillation generally yields better results, the scratch model can achieve competitive outcomes, emphasizing the role of hyperparameters like α in optimizing the distillation process.

Analysis of Confusion Matrices

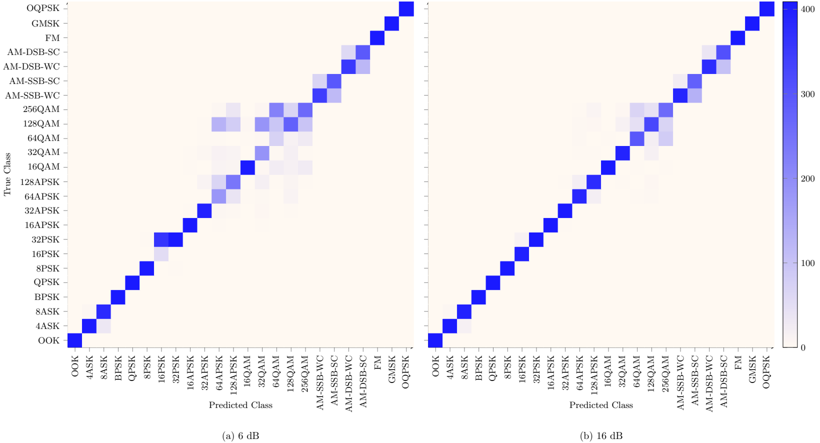

Figure 7: Confusion matrices of the classifier at SNR levels of 6dB (a) and 16dB (b). The diagonal pattern of higher values indicates strong performance for both SNR values, while off-diagonal elements suggest potential challenges in classification.

Analysis of Confusion Matrices

- Figure 7 (a) and (b) illustrate the confusion matrices of the student model at SNR levels of 6dB and 16dB, respectively.
- Both matrices exhibit a diagonal pattern of higher values, indicating strong performance.
- Notably, there are discernible off-diagonal values between modulations with higher orders such as MQAM and similar modulation techniques (AM-SSB-WC to AM-SSB-SC, AM-DSB-WC to AM-DSB-SC).
- Increasing the SNR levels results in better performance.

Table of Contents

1 Introduction

2 Data Preprocessing

3 Network Architecture

4 Knowledge Distillation

5 Results

6 Conclusion and Future Research

7 Bibliography

8 Questions

Conclusion

- This study has demonstrated the effectiveness of response-based KD in training compact AMC models.
- The proposed single-stage SBCNN model, when trained with teacher distillation, consistently outperforms the student model trained from scratch across various SNR levels.
- However, the scratch model shows competitive performance in certain modulation schemes, suggesting that the effectiveness of KD can vary depending on the modulation type, thus highlighting the importance of proper parameter optimization.
- The student model can correctly classify most instances at different SNR levels, although some confusion persists between similar modulation types, especially at lower SNRs.

Future Research

- Optimizing the parameters α and temperature T to further enhance the KD process.
- Exploring more robust models like the XGBoost classifier as teachers could provide better guidance during the distillation process.
- Incorporating advanced preprocessing methods such as Principal Component Analysis (PCA) could enhance feature extraction and improve the model's ability to differentiate signals in noisy environments.
- Lastly, validating the proposed approach in real-world scenarios with diverse environmental conditions and signal variations would provide insights into the practical applicability and robustness of the developed AMC models.

Table of Contents

1. Introduction
2. Data Preprocessing
3. Network Architecture
4. Knowledge Distillation
5. Results
6. Conclusion and Future Research
7. Bibliography
8. Questions

Bibliography

[1] V. Iglesias, J. Grajal, and O. Yeste-Ojeda, "Automatic modulation classifier for military applications," in *2011 19th European Signal Processing Conference*, 2011, pp. 1814–1818.

[2] Y. Yuan, P. Zhao, B. Wang, and B. Wu, "Hybrid maximum likelihood modulation classification for continuous phase modulations," *IEEE Communications Letters*, vol. 20, no. 3, pp. 450–453, 2016.

[3] X. Yan, G. Zhang, and H.-C. Wu, "A novel automatic modulation classifier using graph-based constellation analysis for M-ary qam," *IEEE Communications Letters*, vol. 23, no. 2, pp. 298–301, 2019.

[4] A. Kumar, S. Majhi, G. Gui, H.-C. Wu, and C. Yuen, "A survey of blind modulation classification techniques for ofdm signals," *Sensors*, vol. 22, no. 3, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1020

[5] S.-H. Kim, J.-W. Kim, V.-S. Doan, and D.-S. Kim, "Lightweight deep learning model for automatic modulation classification in cognitive radio networks," *IEEE Access*, vol. 8, pp. 197 532–197 541, 2020.

[6] S.-H. Kim, C.-B. Moon, J.-W. Kim, and D.-S. Kim, "A hybrid deep learning model for automatic modulation classification," *IEEE Wireless Communications Letters*, vol. 11, no. 2, pp. 313–317, 2022.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[8] Z. Li, H. Li, and L. Meng, "Model compression for deep neural networks: A survey," *Computers*, vol. 12, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2073-431X/12/3/60

[9] "Radioml 2018.01a," https://www.deepsig.ai/datasets/, accessed: April 16, 2024.

Table of Contents

1 Introduction

2 Data Preprocessing

3 Network Architecture

4 Knowledge Distillation

5 Results

6 Conclusion and Future Research

7 Bibliography

8 Questions

Questions

Questions?

Acknowledgment

Obrigado!