APPLICATION OF CAPSULE NETWORK ARCHITECTURES FOR MODULATION RECOGNITION

Final Project for CSC-275

BY

SIRISH PRABAKAR

ASHUTOSH DEOWANSHI

SHIWEI FANG

Application of Capsule Network architectures for Modulation Recognition

Ashutosh Deowanshi
Dept. Of Computer Science,
California State University,
Sacramento
ashutoshdeowanshi@csus.edu

Shiwei Fang
Dept. Of Computer Science,
California State University,
Sacramento
sfang@csus.edu

Sirish Prabakar
Dept. Of Computer Science,
California State University,
Sacramento
sirishprabakar@csus.edu

Abstract

Machine Learning and Deep Learning are increasingly in demand as tools used in data communication. In this paper, we investigate the most recent deep learning framework for wireless modulation recognition. The deep learning technique can enhance the recognition performance with comparatively small scale network. We have incorporated the customized convolutional neural network(CNN) and Capsule network to refine the achievement and decrease the complexity of this deep learning platform with lower recognition error rate.. Even though capsule network will be newer concept to additional network structures, it might contain possible capability to resolve this nagging difficulty. Our approach focuses on using capsule network for the same purpose and we outperformed the proposed model by using capsule network. Capsule networks are better suited for image classification with image features having different orientations. They also can be trained on much lower number of total params while producing appreciable performance.

1. Introduction

We aim at replacing the CNN Architecture used in the proposed paper [1]. We will do this by using a capsule network meant for the purpose of classification. We also compare the performance of classification of CNN and Capsule network to get a better understanding of how these models compare. Once we are convinced with the results we plan on using Capsule network for the proposed data set to classify modulation signals. This work can be viewed as implementation and improvement of [0].

Wireless modulation recognition is generally researched in wireless communications society. This is on the grounds that the modulation schemes are significant for cognitive radio and wireless signals observations in business use and security use. A lot of work have been distributed, e.g., statistical features[2], clustering algorithm[3], decision trees[4], artificial neural network[4-6] and genetics algorithm[5].

The authors in [4] have read algorithms for programmed modulation recognition. In [4], the techniques dependent on the decision trees have been proposed to group signals modulation with conventional features. The conventional artificial neural network technique has been proposed in [4-6], in which the strategies just perform well in some specific situations, and the parameters in these strategies must be accurately tuned. In different words, the performance of these techniques isn't promising.
Because of the ongoing advances of deep learning strategies, the precision of the recognition is fundamentally improved [7-9]. In [7], a deep convolutional neural network has been proposed to improve the recognition precision for ImageNet challenges.

A deeper residual network can utilize the forward data across layers and won ImageNet 2015 [8]. In the field of speech recognition, deep recurrent neural network can incredibly decrease error rates in [9], due to the changes of network structures and optimization techniques like dropout [10], batch normalization [11], and Adam stochastic gradient decent algorithm [12]. Models can be trained and move to other related issues without any problem.
In [13C], the authors proposed a convolutional neural

Section 2 presents a few notable deep neural network architectures and presents the upsides of each architecture and concentrates the significant highlights from wireless baseband signals. Section 3 portrays the variety of datasets that are utilized in our training and prediction. Exploratory settings also, results are given in section 4 and hypothetical conclusions are appeared in section 5.

2. Problem Formulation

The dataset is organized as a collection of images. In Python this would be a dataset with Numpy arrays of (224 x 224 x 3). The labels are categorically encoded(0 to 11) to indicate the category.

Here the proposed solution has the CNN network as the main model. Data is fed as training data with the shape (110000 x 2 x 128). This is recognized as an input array and this is how it is taken for an advantage while using CNN. Using the same concept we will be trying to use capsule network in input the input data as a 4d image array for classification purpose

We also have incorporated the concept of class balancing. Using these we have assigned weights to each class according their proportion in the training samples. This is also an additional feature we have included.

Coming to the number of categories, we classified this data into 11 categories based on the dataset used in the main proposed paper. Namely '8PSK', 'AM-DSB', 'AM-SSB', 'BPSK', 'CPFSK', 'GFSK', 'PAM4', 'QAM16', 'QAM64', 'QPSK' and 'WBFM'.

Such large data could not be easily stored in one numpy array. Due to this we had to save them into small npy.files and concatenate them on google colab PRO. Once this was done we could load the final versions of the numpy arrays right before training.

3. System Design

The project is an implementation of Capsule network system. Here instead of a traditional CNN we have used capsule network as our main model. Here even in our CNN model we will not be using transfer learning to enhance the results. This is because the main data is being fed as input image array to the CNN. However the input data is not actually a set of images but is made to look like an image as the input. Same goes for Capsule network and for this reason using transfer learning would not help us.

3.1 System Architecture

3.1.1 Comparison of the Capsule Network vs. CNN Architecture.

3.1.2 CAPSULE.

A CNN model is modified to perform an optimized analysis of modulated signals. In addition to this a Capsule Network was added to determine the spatial information needed to identify various features of the input data for classification.

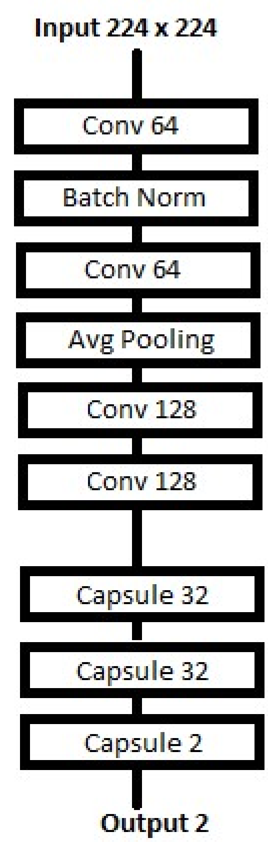


Figure #. Capsule network Architecture.

3.1.3 Proposed CNN Architecture. This is the proposed system for this project. This model uses the traditional CNN without Transfer Learning model.

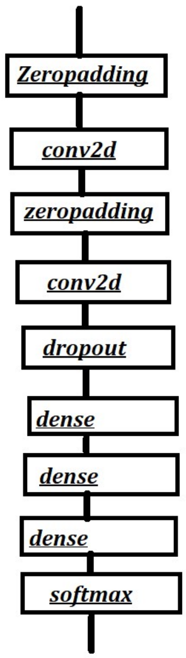


Figure #. Proposed CNN Architecture.

4. Experimental Evaluation

Both models for this project are classification models. The goal here is to classify the modulation signals according to their classes.

4.1 Methodology

In short we aim to achieve 2 tasks. One is to compare Capsule network performance to Transfer learning architecture with CNN for same random dataset. Second is to use capsule network architecture to produce same performance as the performance achieved by the proposed model in the research paper mentioned, if not beat it.

The data required for both models needed to be separated into Training and Testing data, as is typical with most neural network models. Our main focus is on the dataset proposed in the research paper and to implement Capsule network with this dataset. However we have also created another dataset with high-quality data. We designed a transfer learning model for this dataset. This is just for comparison of Capsule Network vs CNN.

The parameter tuning used for the Capsule system for the most part was very basic. The activation functions for most of the layers used RELU. The Adam optimizer was also used for the compiler. The Batch Normalization used momentums of 0.99 while epsilon was set to 0.001. All Convolution layer kernels were set to (3x3). This system was run at 100 epochs.

The CNN Learning model on the other hand used fully connected dense layers with one of them use RELU as an activation function and the final layer using Softmax. Average pooling was used as the pooling layer. The model was also set to non-trainable in all layers.

In order to compare the two systems, a series of metrics is required. They are accuracy, precision, specificity, sensitivity, and the F1 score. The ROC curve is also appropriate for this comparison.

4.1 Results

Using the same dataset for both models it is possible to get a clear picture as to which model performs better at classification purpose. The scores are compared by accuracy, precision, specificity, sensitivity, and F1-score.

4.1.1. Capsule Training and Testing on 2nd dataset.

Figure #. Precision of the Capsule model.

```
accurate predictions:  1498
Total predictions to be made in testing : 1507
accuracy :  0.9940278699402787
specificity  :  0.9960745829244357
sensitivity or precision  :  0.9897540983606558
precision  :  0.9950980392156863
f1-score  :  0.9924188748760097
```

4.1.1. CNN Transfer learning model- Training and Testing on 2nd dataset.

```
[INFO] evaluating network...
              precision    recall  f1-score   support

       covid       0.97      1.00      0.98      1019
      normal       1.00      0.93      0.97       488

    accuracy                           0.98      1507
   macro avg       0.98      0.97      0.97      1507
weighted avg       0.98      0.98      0.98      1507

[[1018    1]
 [  32  456]]
acc: 0.9781
sensitivity: 0.9990
specificity: 0.9344
```

Figure #. Precision of the CNN Transfer Learning Model.

These evalutions were performed on a second dataset which produced high accuracy in both cases. This was just for comparison of Capsule vs CNN performance. Later main dataset which is proposed in paper will be used.

Here it is quite convincing that capsule network outperforms traditional methods of transfer learning with CNN pre-trained models. Keeping this in mind we will Focus on using Capsule network to compete with proposed model in the paper, the main dataset proposed in the paper will be used for comparison. The following results show how Capsule network performed with the proposed dataset.

4.2 Capsule network performance on Main dataset

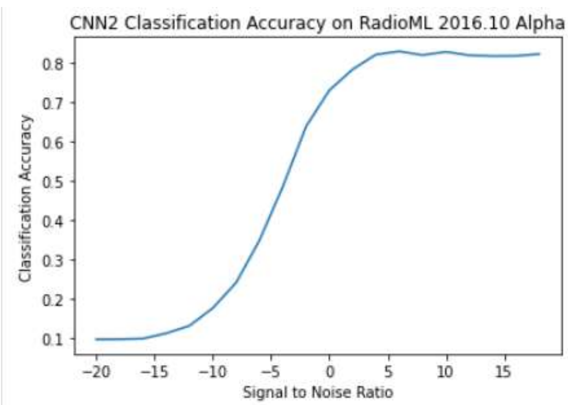


Figure #. classification accuracy figure over different SNR values

```
[[5024  313 1464  414  392  268  154  517  376 1049  138]
 [ 281 5759 1090  160  210  338   70  113   31  350 1457]
 [ 921  731 4198  666  714  489  294  403  117 1160  276]
 [ 435  333 1371 5889  305  216  351  268   68  622  136]
 [ 469  318 1353  278 5784  362  130  279  160  696  139]
 [ 391  434 1347  218  425 5763  118  193   60  477  560]
 [ 294  243 1065  676  208  195 6370  257  111  467  106]
 [ 684  226  924  299  228  191  202 4769 1784  638   84]
 [ 471  154  614  205  176  144  168 1762 5718  520   60]
 [ 710  298 1416  420  363  250  128  437  199 5663  144]
 [ 242 4250 1136  191  252  548   82  115   44  374 2840]]
Plotting confusion matrix
```

Figure #. Classification matrix

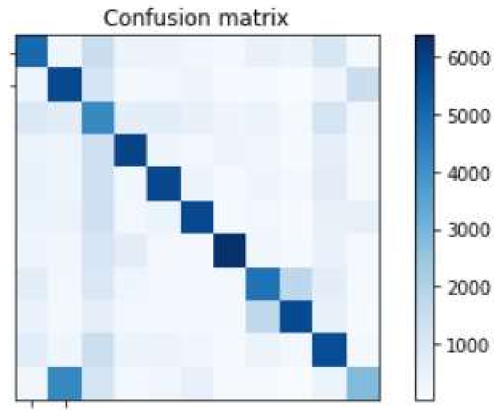


Figure #.Classification confusion matrix

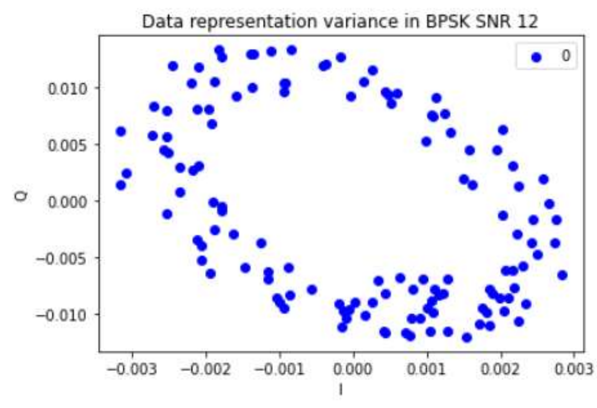


Figure #. Different modulations in the I/Q domain

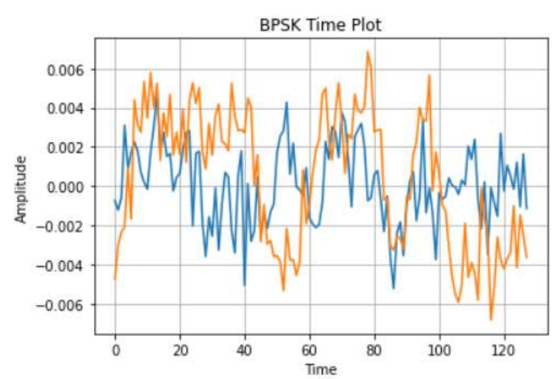


Figure #. Different modulations in the time domain

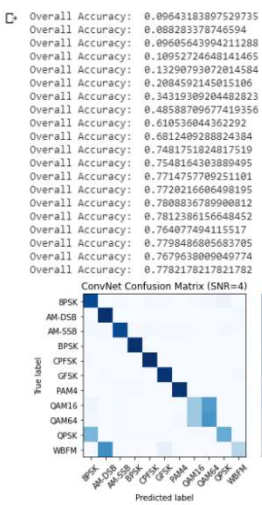
Performance comparison with proposed results:

Performance of main research paper:

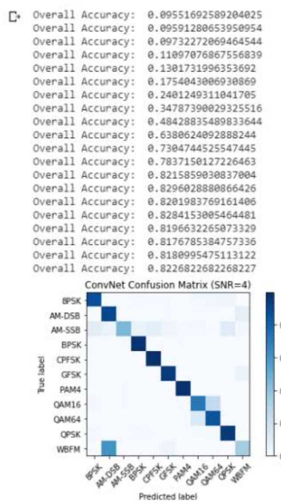
Total params: - 2,830,427
Overall Accuracy over all SNRs - 72.5%

Our model's performance:

Total params: - 282,944
Overall Accuracy over all SNRs - 82.26%



Results from research paper code using CNN



Our results using capsule network instead of CNN

We slightly did better than the proposed model in increasing the accuracy and reducing the total params during training. Here the results are shown in a number of ways as depicted by the above images.

Here we first fed the input data with dimensional shape(110000 x 2 x 128 x1). We tried all other alternatives in the combination of 2,128 and 1. Finally it was shown that combination 2,128,1 worked best and gave best results. Finally we used 2,128,1 shaped array and padded it with 0's to make it appear as an image of 224 x 224 x 3 pixels. This gave us a model able to achieve 96% accuracy during training but however performed poorly during testing phase because it was overfit during training. Using this technique would have been beneficial. However the biggest challenge in padding and using 224 x 224 x 3 is storage limitations. Once the 110,000 inputs are stored in this format it would take close to 50gb of storage space just to save the numpy array. So we were forced to have a small dataset of 11000 training samples with 1000 from each category. Even though this dataset was well balanced it did not give us the best results because its number of training samples was reduced by 90%. This gave the model less number of instances to train for each category. We howver believe this technique will give us the best results once we overcome the storage limitations.

Also number of total parameters is drastically reduced from 117 million to 295k with this approach. This is a big advantage if the task at hand requires considerable amount of params during training.

Inference:

5. Related Work

Here the main paper for reference is [1] which talks about modulation recognition with physical adversarial attacks. Paper at [0] is an improvement to paper which implements capsule network for the same classification process. Our goal was to implement paper [0] and we also improved the score obtained in paper [0] from 76% overall accuracy to 82%. Our work can be considered as an improved implementation of the original work proposed in [0] and [1].

Capsule network's performance advantage compared to CNN, helped us develop a stronger model with better performance.

6. Conclusion

We were able to obtain appreciable performance with capsule network. Our future work will focus on using real time data augmentation and making use of better dataset to build a strong DL model to handle such classification with high performance.

Overall we were very satisfied with our approach and end result. We were trying out something new and we had no idea whether it would work or not. Eventually we were able to unfold our mistakes, make improvements and produce an appreciable amount of progress. Keeping in mind that we would be commended if we put in our best efforts we worked to make progress rather than just trying to get the end result. I feel this made the whole learning experience effective and fun too.

7. LEARNING EXPERIENCE:

During the whole process of project development, we have learnt much more than we had expected. Every time we faced an obstacle we were more motivated than before to find a way around it.

From this project we became aware of new concepts and built on the knowledge we already had from this course. We were exposed to the fact that machine learning and artificial intelligence has very high potential to change the way we do things now and in the future. Most importantly during the research phase of our project we became aware of how AI/ML is dominating in so many different domains.

Coming to this project, we were introduced to capsule network which is relatively new compared to traditional CNN which are used for image classification. We learnt the differences, pros and cons of each. For example Capsule network work great with images where the image features can have different orientations. We also learnt about many fine-tuning methods like class balancing, real time data augmentation , pre-training and why transfer learning works even though they are trained on a large variety of different images. Also our coding capabilities and familiarity with tensorflow functions have sharpened by a great deal. Finally and also most importantly we have reshaped our perspective towards challenging tasks like this project. Now that we have overcome most of the obstacles we faced, we are not only prepared for such challenging projects in the future but we are highly motivated to challenge ourselves with such tasks.

CHALLENGES FACED:

One of the main difficulties was not being able to use real-time data augmentation. This was a feature we had hoped to add to our work but however we are working with large data such as this one real-time data augmentation is very computationally expensive. Our google colab pro session would crash within minutes when we tried to use this feature. However we are motivated to be able to add this in the future we have the availability of such high resources.

Preprocessing of data required us to make numpy arrays of small sizes and then concatenate them to produce the final numpy arrays needed for training and testing. This took hours to upload to google drive and then concatenate them on google colab. This could not have been possible since local PC can not assign 15gb for a single numpy array. Also due to such big numpy arrays at hand, the google colab VM would keep crashing when it ran of our RAM even when we upgraded to google colab pro forcing us to restart the whole notebook each time. Finally Google colab would cut out service to our account because we overused their GPU and we would no longer be able to load files from google drive(OSERROR 5). This forced us to create new google accounts and mount the new google drives onto google colab. And reupload all the files to the new google drive. Apart from these technical challenges we also found it challenging to understand the best parameters for training and testing during fine-tuning stage.

Acknowlegements

Thanks to Professor Xuyu Wang for allowing us to proceed with our project. Prof. Wang has always been ready to help us when ever we approached him with questions. We are very thankful for the support.

References

[0] Application of novel architectures for Modulation Recognition Yujie Sang#1, Li (Alex) Li#2

[1] M. Sadeghi, "Security and robustness of DL in wireless communication systems," https://github.com/meysamsadeghi/ Security-and-Robustness-of-Deep-Learning-in-Wireless-Communication-Systems, 2019.

[2] Wong M L D, Nandi A K. Automatic digital modulation recognition using spectral and statistical features with multi-layer perceptrons[C]//Signal Processing and its Applications, Sixth International, Symposium on. 2001. IEEE, 2001, 2: 390-393.

[3] K. T. Woo and C. W. Kok, "Clustering based distribution fitting algorithm for Automatic Modulation Recognition," 2007 12th IEEE Symposium on Computers and Communications, Aveiro, 2007, pp. 13- 18. [3] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," in IEEE Transactions on Communications, vol. 46, no. 4, pp. 431-436, Apr 1998.

[4] M.L.D. Wong, A.K. Nandi, "Automatic digital modulation recognition using spectral and statistical features with multi-layer perceptrons", Signal Processing and its Applications Sixth International Symposium on. 2001, vol. 2, pp. 390-393 vol.2, 2001.

[5] Wong M L D, Nandi A K. Automatic digital modulation recognition using artificial neural network and genetic algorithm[J]. Signal Processing, 2004, 84(2): 351-365.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[7] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.A.

[8] Graves, A. r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 6645-6649.

[9] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[10] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," CoRR, vol. abs/1412.6980, 2014.

[12] 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

[13] A. Abbas, M. Abdelsamea, M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network", Mathematics Department, Faculty of Science, Assiut University, Assiut, Egypt. March 26, 2020.
https://www.researchgate.net/publication/340332332_Classification_of_COVID-19_in_chest_X-ray_images_using_DeTraC_deep_convolutional_neural_network

[14] P. Afshar, A. Mohammadi and K. N. Plataniotis, "Brain Tumor Type Classification via Capsule Networks,"
2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 3129-3133.
https://ieeexplore-ieee-org.proxy.lib.csus.edu/stamp/stamp.jsp?tp=&arnumber=8451379

[15] P. Afshar, K. N. Plataniotis and A. Mohammadi, "Capsule Networks for Brain Tumor Classification Based on MRI Images and Coarse Tumor Boundaries,", ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, 2019, pp. 1368-1372.
https://ieeexplore-ieee-org.proxy.lib.csus.edu/stamp/stamp.jsp?tp=&arnumber=8683759

[16] P. Afshar, K. N Plataniotis, A. Mohammadi, "Capsule Networks' Interpretability for Brain Tumor Classification Via Radiomics Analyses", 2019 IEEE International Conference on Image Processing (ICIP), pp. 13816-3820, 2019.
https://ieeexplore-ieee-org.proxy.lib.csus.edu/stamp/stamp.jsp?tp=&arnumber=8803615

[17] J. P. Cohen, P. Morrison, L. Dao
COVID-19 image data collection. 2020.
https://github.com/ieee8023/covid-chestxray-dataset

[18] S. Heidarian P. Afshar, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "COVID-CAPS: A Capsule Network-based framework for identification of COVID-19 cases from chest X-ray Images"
https://arxiv.org/abs/2004.02696

[19] H. Maghdid S., A. Asaad, K. Ghafoor, A. Sadiq, M. Khan, "Diagnosing COVID-19 Pneumonia from X-Ray and CT Images using Deep Learning and Transfer Learning Algorithms", Cornell University, Mar 31, 2020
https://arxiv.org/abs/2004.00038

[20] L. Wang, A. Wong, "COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images", Department of Systems Design Engineering, University of Waterloo, Canada. March 30, 2020
https://arxiv.org/pdf/2003.09871v2.pdf

[21] L. Wang, A. Wong, Z.Q. Lin, J. Lee, P. McInnis, et. al. "Figure 1 - COVID-19 Chest X-ray Dataset Initiative"
https://github.com/agchung/Figure1-COVID-chestxray-dataset

[22] "RSNA Pneumonia Detection Challenge", "Can you build an algorithm that automatically detects potential pneumonia cases?"
https://www.kaggle.com/c/rsna-pneumonia-detection-challenge

[23] J.P Cohen, J. Tolgyesi, J.R. King, "Torchxrayvision"
https://github.com/mlmed/torchxrayvision/blob/master/torchxrayvision/datasets.py#L814