Krish Rai

DATASCI 224

Final Paper

06/11/24

Capsule Neural Networks: PneumoniaMNIST X-Ray Binary Classification

Goal, Dataset, and Algorithm

A tool to classify medical images correctly through machine learning, such as for pneumonia diagnosis, would be incredibly beneficial in helping physicians provide prompt and efficient medical care. On the other hand, misclassification can be a significant failure and affect patients negatively. Many medical images, including chest X-rays have information coded into just a small number of pixels. With poor resolution, blurry and unclear markings, and lack of color, classification models like the Convolutional Neural Network (CNN) may decrease in performance and be more prone to error [1]. CNNs, commonly used for image classification, work through translational invariance. Through pooling, the maximum or average values are taken in layers which reduce the feature information in an output [2]. However, through pooling CNNs may lose important feature information [2]. This could include orientation, position, and thickness, affecting classification accuracy.

The dataset used, PneumoniaMNIST from MedMNIST, contains pediatric Chest X-ray images [3]. These images are used to help doctors determine whether a patient has pneumonia. This dataset has a binary classification for labels: presence of pneumonia (1) or a normal (0) x ray. This dataset includes 5856 images, 4708 for training and 624 for testing (remaining for validation). All images are in greyscale, and all images are converted to size 28x28 pixels.

A. **B.**

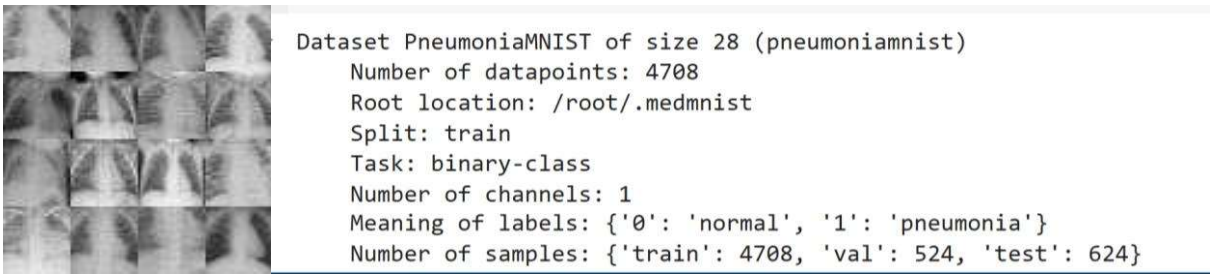

Figure 1. ***A.*** *16 X-ray images from PneumoniaMNIST (using montage).* ***B.*** *Printed snippet of information including labels and number of samples from the dataset.*

The individual features of the x-ray lung images in the dataset vary significantly from each other. Therefore, there may be benefit in alternatives to CNNs in medical image classification.

The goal is to test the accuracy of Capsule Neural Networks, a promising model on MNIST, on the classification of medical Pneumonia chest X-ray imaging. The purpose is to see whether this predicted model can maintain or even improve on standard metrics for classification (above 90% accuracy).

Functioning of Capsule Neural Networks

Capsule neural networks are built on Convolutional Neural Networks (CNNs) for feature inclusion in image classification. In Geoffrey Hinton's seminal paper in 2017 on capsule networks, this novel algorithm was tested to see if it could mitigate the problem of losing feature information like orientation in CNNs. This paper successfully classified MNIST, handwritten digits data, with low test error [4]. However, the Capsule network is also mentioned to struggle with complex data and with distinguishing crowded objects [5]. Therefore, its success on medical images, which likely have poorer resolution and are not standard usage as MNIST, is not clear. Since this research is very new, there are limited studies and Hinton's paper is cited as primary evidence for any Capsule related analysis.

This model contains a vector capsule as the fundamental unit rather than a scalar neuron as in CNNs [6]. While the convolutional layer is the same in both, the primary capsule layer that follows it transforms the scalar activation from the Convolutional layer to vector capsules. These capsules contain all the feature components that may be lost from a CNN, including orientation, overlap, and thickness [4]. As a lower-level capsule feeds into a higher-level capsule layer, a transformation matrix that accounts for the relation between a feature and how it fits into the larger space is introduced. The dot product of this transformation matrix and the specific feature region produces the higher layer capsule vector [6]. This vector is then squashed:

$$v_j = \frac{||s_j||^2||}{1+||s_j||^2} \frac{s_j}{||s_j||}$$

Figure 2.

The unnormalized vector sj is reduced to between 0 and 1 per squashing function. This marks the probability of this feature being present. It is multiplied by the unit vector (2nd half of equation sj/||sj||), so it maintains a sense of direction [6].

Finally, the fully connected layer also differs between the two models. In CNN, the Fully Connected Layer weighs the sum of the previous layer's outputs. In Capsule Networks, there is a method called dynamic routing/routing by agreement, which iterates over the previous capsules and adjusts the weights to enhance strong connections and weaken poor ones. If there is agreement between the prediction of a capsule and the output, the most important prediction capsules are given higher weightage [4,7]. This is repeated (dynamic routing) to minimize loss.

The marginal loss function is: $Lk = Tk \max(0, m^+ - ||vk||)^2 + \lambda(1 - Tk) \max(0, ||vk|| - m^-)^2$, where $m^+ = 0.9$, $m^- = 0.1$, and $\lambda = 0.5$ [4]. In this equation, Tk is a binary indicator of the presence (Tk=1) or absence (Tk=0) of a feature. Depending on if presence or absence of feature is being measured, one of the two components cancels out (1-Tk when Tk =1 or Tk when Tk =0). m^+ is an upper threshold, set to 0.9 to represent high certainty of the presence of a feature, and m^- is a lower threshold set to 0.1 to represent high certainty of the absence of a feature. The L2 norm of the feature vector Vk is then subtracted to calculate the marginal loss, Lk. This loss is minimized by picking the weights during routing by agreement. Through this process, capsules have potential in accounting for more information and improving image classification. Therefore, the capsule network was considered the best for pneumonia classification in these differing images.

Results

After applying a Capsule Neural Network on PneumoniaMNIST with 2 output capsules, (See **Appendix**), the following results were computed for pneumonia image binary label classification. The final accuracy of the training data for 30 epochs and batch size of 128 was 0.679688 and the training loss was 0.003164. The corresponding testing accuracy was 0.601562 and the testing loss was 0.001582.

A. B.

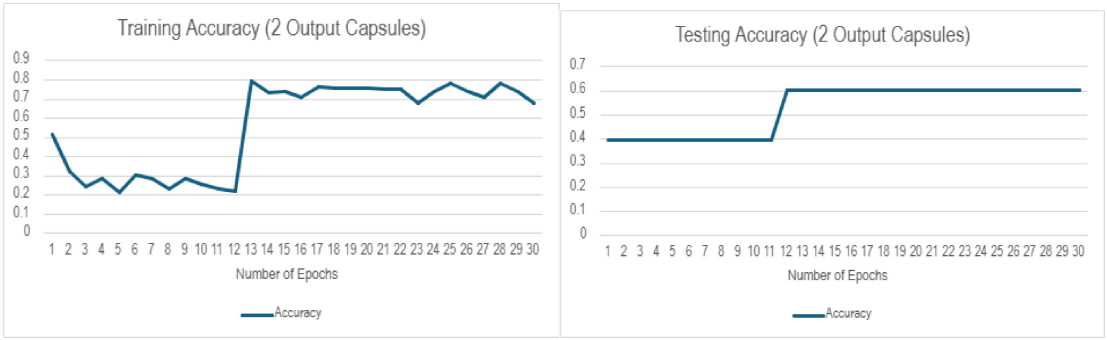

Figure 3. A. *Training accuracy for the 2-capsule output Capsule Network (30 epochs).* ***B.*** *Testing accuracy for the 2-capsule output Capsule Network (30 epochs).*

The parameters were tweaked to increase the number of capsules (Ex: 4 output capsules and changed the number of epochs by 10), but the final accuracy for testing did not improve beyond 0.6. Both the training and testing loss did not reduce significantly after epoch 12, which was when convergence occurred.

Therefore, the capsule neural network algorithm did not get a high enough accuracy to significantly justify its use in medical image classification. This may be due to the quality of the images; crowding and various image background colors may affect the error as it did on CIFAR-10 [4]. Additionally, further parameter tuning or adjustment (adding convolutional layers) may be necessary, or a greater number of images may improve accuracy.

Additionally, implementation on other MedMNIST datasets may shed more light on the Capsule Network. Running these epochs took a sizable amount of time on CPU (GPUs needed), and clearer image datasets like DermaMNIST failed in any capsule network adaptation; DermaMNIST was labeled as multi-class classification and conversion to tensor for analysis posed challenges. Therefore, PneumoniaMNIST was chosen. The capsule neural network has potential conceptually for this type of data, but further testing is needed.

Appendix

Capsule Net Architecture for PneumoniaMNIST: Decoder Layer present

Input shape = (1,28,28)

Convolutional layer parameters

- Output channels= 256
- Kernel size = 9x9

- stride =1

Primary Capsule Layer Parameters

- Number of capsules = 8
- Output channels = 32
- Kernel size = 9
- stride = 2

Output/object capsule parameters

- Output/Object number of capsules = 2
- Output/Object output channels = 16

References

1. Bria, Alessandro, Claudio Marrocco, and Francesco Tortorella. "Addressing class imbalance in deep learning for small lesion detection on medical images." *Computers in biology and medicine* 120 (2020): 103735.

2. Sun, Zengguo, et al. "Overview of capsule neural networks." *Journal of Internet Technology* 23.1 (2022): 33-44.

3. Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, Bingbing Ni. Yang, Jiancheng, et al. "MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification." Scientific Data, 2023.

4. Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." Advances in neural information processing systems 30 (2017).

5. Dombetzki, Luca Alessandro. "An overview over capsule networks." *Network architectures and services* 10 (2018).

6. Aurélien Géron. “Capsule Networks (CapsNets) – Tutorial.” YouTube, 21 Nov. 2017, www.youtube.com/watch?v=pPN8d0E3900. Accessed 14 June 2024.

7. Count From Zero. “Capsule Network Explained.” YouTube, 29 Dec. 2020, www.youtube.com/watch?v=v0tgo3c_7Xs. Accessed 14 June 2024.

8. Hinton, Geoffrey E., Alex Krizhevsky, and Sida D. Wang. "Transforming auto-encoders." Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21. Springer Berlin Heidelberg, 2011.

9. Iwasaki, Kenta. “Gram-Ai/Capsule-Networks.” GitHub, 8 June 2024, github.com/gram-ai/capsule-networks. Accessed 14 June 2024.

10. Dulat, Yerzat. “Higgsfield/Capsule-Network-Tutorial.” GitHub, 23 May 2024, github.com/higgsfield/Capsule-Network-Tutorial/tree/master. Accessed 14 June 2024.