Learning Over Object Boundary Point Clouds

Christopher M. Bender
Department of Computer Science
The University of North Carolina
Chapel Hill, NC 27599-2200
bender@cs.unc.edu

Abstract

In this work, I attempt to transfer a deep neural network trained on a boundary point cloud representation of Model Net 40 to similar point clouds of brain scans to estimate if infants will be diagnosed with autism. Since point clouds are a set of unordered points typical neural network architectures are not well-matched to the problem and cannot be used without additional modification. I make use of the Deep Set architecture's inherent permutation invariance and equivariance to overcome the difficulties associated with unordered sets. By stacking several Deep Set layers, we can generate a generic feature set that can be used in classification. Transfer learning is required to overcome the limited size of the medical data set. The classification accuracy achieved through this process appears to be on-par or better than other methods but is extremely sensitive to the fine-tuned network's initialization.

1 Introduction

Diagnosis of various medical maladies from remote scans has seen significant gains in recent years with the rise of neural networks [1**?**]. Many of these works utilize single-slices through a patient's anatomy and attempt to classification from a 2D starting place. Unfortunately, this ignores necessary information that exists out-of-plane. One method to get around this issue while still using existing method passes in each 2D slice as a sequence to a recurrent network. Other works have attempted to perform classification of medical data by extending the convolution operator to non-flat surfaces.[2]

We will also limit our analysis modality to the surface of various structures in the brain. However, unlike other 3D methods, we will use neither a voxelized or regularized surface representation. Instead, we will extract points on the boundary of the objects and use the unordered set of points to represent the data. Modern CNN exploit the regular grid of 1-3D data via convolution to extract multi-resolution representations of the data, beginning at fine scales and working towards gross scales. The use of unordered data precludes the use of convolutions. Instead, we must utilize methods that are robust to permutation of the point order (since all that matters is where the points are in space, not what order they exist in a data vector). Fortunately, recent interest in sets has produced several techniques that have this property. We will use Deep Set[3] layers to generate a feature set that is robust to permutation and produces good classification results.

The analysis focuses on prediction of autism diagnosis given only scans of the bilateral hippocampi and caudate nuclei of six to twenty-four month old infants. Typically, autism cannot be diagnosed until a child has reached a certain level of maturity since diagnosis is based on behavioral observations that do not develop until approximately the child is two years old. Detection at an earlier age is desirable since it has been demonstrated that the earlier treatment can begin the less the impact the condition has on the child's life.

Unfortunately, the data set of interest (IBIS, see Sec. 2.2) is composed of less than 200 total examples. Neural networks usually require several hundred, if not thousand, examples per class to generate reliable results. We will attempt to overcome this limitation by training a Deep Set network on Model Net 40[4] and then transfer the majority of the network to the IBIS so that the IBIS data is only used to train a shallow network. The proceedure used can be found in Sec. 3 and the results in Sec. 4.

2 Data Sets

I used two data sets in this work. The first, Model Net 40 was used to train a deep set network to distinguish between 40 different everyday items. The second, the Infant Brain Imaging Study (IBIS) was classified by transferring the Model Net 40-trained network and fine-tuning.

2.1 Model Net 40

Model Net[4] is a collection of 3D CAD "images" of over 600 different classes maintained by Princeton University. This large data set has been parred down into a 40 class subset that has gone through quality checks and normalizations. The data set has become a standard in 3D dimensional classification problems. Typically, the CAD files are transformed into a binary voxel representation (on if the object is in the voxel, off otherwise) or are represented using multi-angle projections onto a 2D surface.

We will instead utilize a collection of 10,000 boundary points for each example. The number of training examples vary from 60 to over 600 per class. The number of testing examples vary from 20 to 100. This split is a predefined standard. Figure 1 illustrates one example from all 40 classes in Model Net 40.

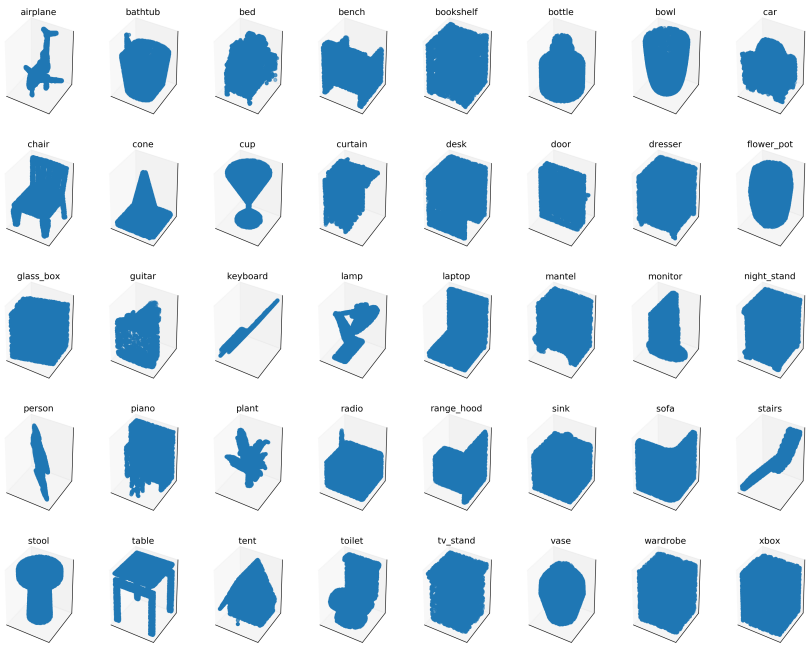


Figure 1: Model Net 40 Point-Cloud Representation

2.2 Infant Brain Imaging Study

The Infant Brain Imaging Study is a collection of MRI scans (courtesy NIH) of the bilateral hippocampi and the caudate nuclei on at-risk infants at 6, 12, and 24 months. At-risk infants are children that have an older sibling that has met the criteria for an autism diagnosis. We will explore the data collected at 6 months. This set contains 143 negative and 34 positive examples, where a positive diagnosis means the child was later found to meet the criteria for autism. The data was randomly divided into a training set (80%, 114/27 examples) and a testing set (20%, 29/7 examples). Additional details can be found in Chapter 3 of JP Hong's dissertation.[5]

Point clouds of the brain structures were constructed from fit skeletal representations[6] (s-reps) of the MRI data. A discussion of this process is outside the scope of this document, see [5] for a detailed discussion of the process. S-reps are used because they inherently contain local information (such as surface normal direction) and non-local information (such as width). It might be desirable to train a classifier using the s-reps more directly. Unfortunately, there is no network that easily extends to the representation and, even if an architecture existed, too little data is available to attempt training from scratch. Fortunately, it is easy to extract the boundary points from each spoke and these points can be used in training. It remains an open question if there might be some way to incorporate the normal direction and how much benefit the union of boundary points and normals might have on the task. Figure 2 shows the brain structure and its fitted s-rep.

(a) Smooth Boundary Representation (b) Skeletal Representation

Figure 2: Example 3D Brain Structure

3 Method

As mentioned previously, I trained a Deep Set architecture on Model Net 40 and then transferred the features from the Deep Set layers to the IBIS data set. To test that the method performs as expected, I actually split two classes out of Model Net 40 and retained them. I then transferred the network trained on the remaining 38 classes to the 2 retained classes. This was done since I knew the network should work on the Model Net classes, that the examples had gone through the same sanitation process, and that each example had the same number of points. While the architecture should be robust to some of these variations, I thought it would be helpful to check that this is the case. The results of these checks can be found in Sec. 4.

Aside: I originally broke the data out in this way because I was not sure if I would get access to the s-reps in time to use them for this work.

3.1 Deep Sets

Deep Sets[3] is a novel neural network architecture that is intended to be robust to typical variations found in sets. Namely, the number and order of points within a single set. This is achieved through the use of permutation invariant and equivariant operations, i.e. the output is the same regardless of the order (permutation invariance) or is permuted in the same way as the input (permutation equivariance). The architecture used here is composed of three permutation invariant layers and one fully connected layer. All of these layers are trained jointly on the Model Net data. The network achieved approximately 87% accuracy on the original classification task after 100 training epochs.

3.2 Transfer Learning

Transfer learning[7] is a common technique in machine learning where a network is trained on a task similar to the actual task and parts of that network are copied into a new network that is used on the primary task. The main reasons transfer learning is used are to overcome limited training samples in the primary task or to offset training times. I have used it for the former reason.

I trained a Deep Set architecture as described in Sec. 3.1. Then, I removed the final, fully-connected layer and replaced it with a new one associated with the binary classification task. Finally, I trained the last layer with stochastic gradient descent while holding the Deep Set layers fixed. It is acceptable to allow the lower layers to adjust slightly to the primary task, but I did not think this was appropriate since the task is significantly data-starved. An intuitive explanation of this process is that the original task was used to train a network to produce useful, robust features (geared towards classification) and the primary task took those features and trained a shallow network to produce new class labels.

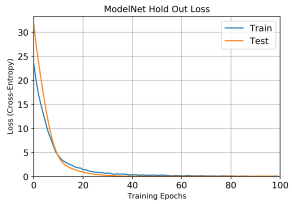
4 Results

The results are broken down into two sections. In the first section, the Deep Set network is trained using all of the 10,000 points from each of the 38 classes, where airplanes and beds were retained for testing. That network was (separately) transferred to both the airplane/bed task and the autism detection task. In the second section, only 100 points were used in training the Deep Set network and bowls/cones were retained from Model Net for testing. Section 4.2 discusses why these changes were made.

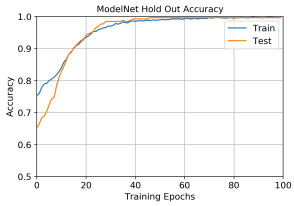
Results are presented for each binary classification problem by illustrating training performance over 100 epochs in terms of cross-entropy loss and classification accuracy. Additionally, the ROC is presented for both classes at the final epoch along with corresponding AUC. For reference, JP Hong achieved an AUC for the positive class of approximately 0.64.[5]

4.1 10,000 Point Transfer

The airplane/bed split was originally chosen as a test set because both classes have a relatively large number of train/test examples (600/100). Figure 3 illustrates the networks performance versus epoch. The network achieves low loss and high accuracy after approximately 40 epochs. The ROC curves (Fig. 4) for both classes are nearly perfect by the final epoch with AUCs of 1.00 and 0.99.

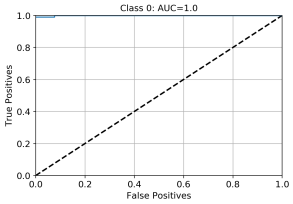


(a) Airplane vs Bed Loss

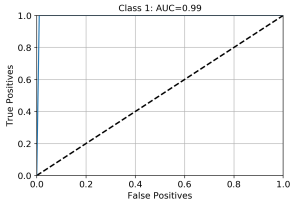


(b) Airplane vs Bed Accuracy

Figure 3: Airplane vs Bed Test/Train Evolution



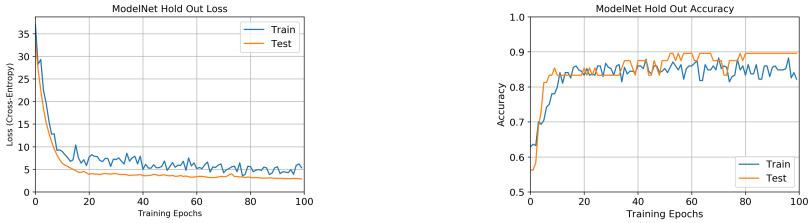
(a) Airplane ROC Curve



(b) Bed ROC Curve

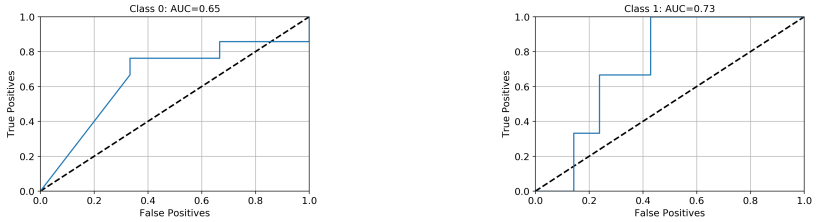
Figure 4: Airplane and Bed ROC Curves

Figure 5 illustrates the training/testing loss and accuracy over epoch. Unlike the airplane/bed split, the network does not achieve near-perfect performance on the the autism task. The step-like ROC curves in Fig. 6 correspond to the lack of testing data for both classes. Here we achieve AUCs of 0.65/0.73 (negative/positive). The main takeaway from these numbers is that we outperform the previous results by a relatively large margin (0.73 to 0.64). However, performance here appears to be extremely sensitive to the network initialization. Not shown, are AUCs that vary from 0.4 to 0.8. It is difficult to say if the results are really as impressive as they seem to be or if we are getting lucky on our relatively small test set.



(a) Medical Shape Loss (b) Medical Shape Accuracy

Figure 5: 10,000 Point Transfer Medical Train/Test Evolution



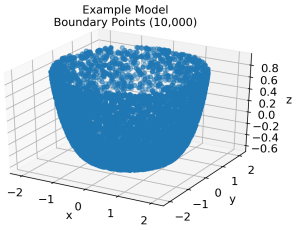
(a) No-Autism ROC Curve (b) Autism ROC Curve

Figure 6: 10,000 Point Transfer Medical ROC Curves

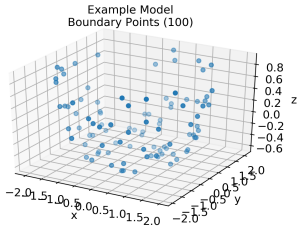
4.2 100 Point Transfer

In the tests above, the original deep set network is trained using all 10,000 points in each Model Net example. However, the medical point clouds are only composed of approximately 70 points. While the deep set architecture is capable of handling a variable number of input points, the network may learn to rely on the high levels of correlation available in these cases. Since the s-reps have compressed the data to a minimal number of spokes and the points are derived from those spokes, the medical data does not contain the same level of redundancy. To make sure this is not a problem, the original network was retrained using a random sampling of 1% of the original points (final accuracy remained unchanged, indicating that the network does not actually need the original 10,000 points). This means that the base network was trained on an equitable number of points as are naively available from the s-reps. Note, I believe it is possible to interpolate additional points from the s-rep but have not attempted to do so.

Additionally, the variation in shape caused by autism is relatively small. This is not the case with airplanes and beds. In an effort to get the prototyping set to better match the true set, I changed the two Model Net held-out sets to “bowl” and “cone.” In addition to the better shape match, there are 164 bowl examples but only 68 cone examples. This also better matches the medical data set where the negative/positive split is approximately 75%/25%. Finally, the new held-out set only contains 20 testing examples each. Figure 7 shows an example of a bowl point cloud before and after down sampling.



(a) Bowl with 10,000 Boundary Points

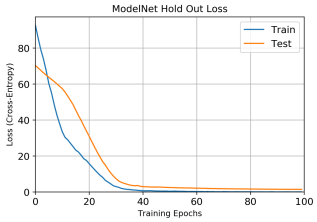


(b) Bowl with 100 Boundary Points

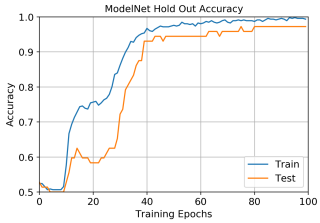
Figure 7: Bowl Downsampling

The results of the downsampled bowl versus cone training procedure are shown in Fig. 8. The network achieves reasonably good results but does not achieve the near-perfect precision found in the airplane/bed task. I believe this drop in performance is because the two objects are much more similar than the first task, this is assumption is reinforced by the consistency of the original network's performance for both the 10,000 and 100 point cases.

Figure 9 contains the ROC curves for the bowl/cone task. Both classes perform well with AUCs of 0.88/0.89 bowl/cone. It is a bit surprising that the classes have such similar performance given the difference in the number of examples per class.

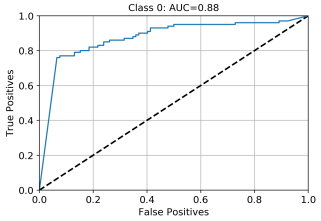


(a) Bowl vs Cone Loss

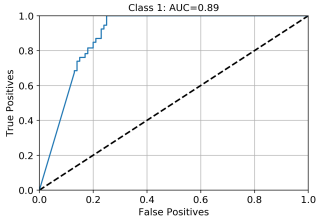


(b) Bowl vs Cone Accuracy

Figure 8: Bowl vs Cone Test/Train Evolution



(a) Bowl ROC Curve

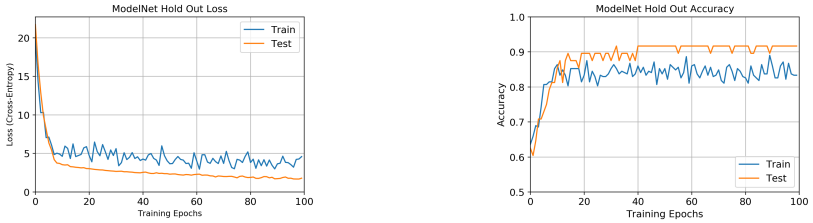


(b) Cone ROC Curve

Figure 9: Bowl and Cone ROC Curves

Finally, Fig. 10 illustrates the performance of the autism task when the Deep Set layers are trained on 100 Model Net points (per example). As in the other examples, the overall accuracy and loss do not appear much-affected by the difference in the training procedure. This indicates that the Deep Set does not use the extra points available in Model Net to inform the parameters of in the layers. However, additional points in the medical images may still assist with training accuracy. Additional research is required.

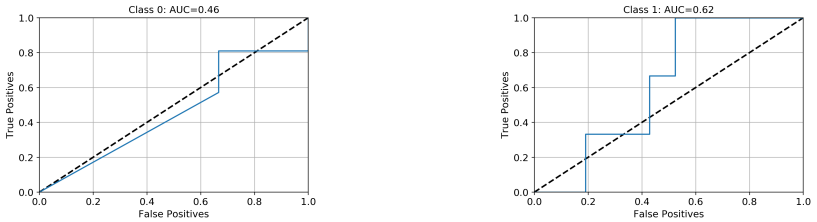
Figure 11 shows the ROC curves for the autism task. In this case, the AUCs are 0.46/0.62, negative/positive. It is somewhat disturbing that the model achieves an AUC less than 0.5 in the negative case (though perhaps diagnostically acceptable). Note, the AUC of 0.62 is equitable to the results achieved by JP in his analysis. As before, the AUC is observed to be sensitive to the initialization of the network and varies from 0.5 to 0.7. Interestingly, this variation is lower than before.



(a) Medical Shape Loss

(b) Medical Shape Accuracy

Figure 10: 100 Point Transfer Medical Train/Test Evolution



(a) No-Autism ROC Curve

(b) Autism ROC Curve

Figure 11: 100 Point Transfer Medical ROC Curves

5 Conclusion

I have demonstrated in this work that it is possible to transfer the permutation in-/equi-variant Deep Set layers to new tasks on boundary point sets. The results of transferring this information to brain scans for the purpose of diagnosing infants with autism are encouraging. Unfortunately, the lack of data and the variability of the performance with initialization draws into question if the model performs as well as it seems or if it is overfitting or getting lucky on the limited test set. Additional data or a change to a similar [medical] task would help to answer this unanswered concern.

Acknowledgments

Several individuals contributed code, data, and/or insight that made this work possible. Thanks to: Dr. J. Oliva for providing the ModelNet 40 dataset in point-cloud form, a Tensorflow implementation of the deepset architecture, and many helpful conversations and advice, Jiyao Wang for providing the refined s-rep data set, Zhiyaun Liu for providing a refined s-rep to SALT s-rep converter, Meghan Stuart for providing a Python implementation of the SALT s-rep data reader and point cloud converter, and Dr. S. Pizer for his guidance and insight into the difficulties of medical analysis.

Code

All of the experiments described in this document were done using the Tensorflow framework.[8] My working code can be found at `https://github.com/waveBender/775_project`.

References

[1] Qeethara Kadhim Al-shayea. Artificial neural networks in medical diagnosis. *International Journal of Computer Science Issues (IJCSI*, pages 150–154, 2011.

[2] Mahmoud Mostapha, Sun Hyung Kim, Guorong Wu, Leo Zsembik, Stephen M. Pizer, and Martin Styner. Non-euclidean, convolutional learning on cortical brain surfaces. In *15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, Washington, DC, USA, April 4-7, 2018*, pages 527–530, 2018.

[3] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3391–3401. Curran Associates, Inc., 2017.

[4] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920. IEEE Computer Society, 2015.

[5] Jonpyo Hong. *Classification of Neuroanatomical Structures Based on Non-Euclidean Geometric Object Properties*. PhD thesis, The University of North Carolina, 2018.

[6] Kaleem Siddiqi and Stephen Pizer. *Medial Representations: Mathematics, Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[7] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

[8] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.