SPECTRAL CAPSULE NETWORKS

Mohammad Taha Bahadori
Amazon Web Services
bahadorm@amazon.com

ABSTRACT

In search for more accurate predictive models, we customize capsule networks for the learning to diagnose problem. We also propose *Spectral Capsule Networks*, a novel variation of capsule networks, that converge faster than capsule network with EM routing. Spectral capsule networks consist of spatial coincidence filters that detect entities based on the alignment of extracted features on a one-dimensional linear subspace. Experiments on a public benchmark learning to diagnose dataset not only shows the success of capsule networks on this task, but also confirm the faster convergence of the spectral capsule networks.

1 INTRODUCTION

The potential for improvement of the quality of care via artificial intelligence has led to significant advances in predictive modeling for healthcare (Lipton et al., 2015; Choi et al., 2016; Razavian et al., 2016; Che et al., 2016; Miotto et al., 2017; Suresh et al., 2017; Rajkomar et al., 2018). For accurate prediction, the models in healthcare need to not only identify risk factors, but also distill the complex and hierarchal temporal interactions among symptoms, conditions, and medications.

It has been argued that traditional deep neural networks might not be efficient in capturing the hierarchical structure of the entities in the images (Oyallon & Mallat, 2015; Cohen & Welling, 2016; Dieleman et al., 2016; Gens & Domingos, 2014; Worrall et al., 2017). They argue that networks that preserve variations in the input perform superior to those that drop variations (equivariant vs. invariant architectures), as the upper layer can have access to the spatial relationship of the entities detected by the lower layers. In particular, in capsule networks (Hinton et al., 2011; Sabour et al., 2017; Hinton et al., 2018) the capsules are designed to have both activation and pose components, where the latter is responsible for preserving the variations in the detected entity.

In this work, we first develop a version of capsule networks with EM routing (EM-Capsules) and show that it can accurately predict diagnoses. We observe that EM-Capsules converge slowly in our dataset and are sensitive to the selection of hyperparameters such as learning rate. To address these issues, we propose Spectral Capsule Networks (S-Capsules) that are also spatial coincidence filters, similar to EM-Capsules. In contrast to EM-Capsules, S-Capsules measure the coincidence as the degree of alignment of the votes from below capsules in a one-dimensional linear subspace, rather than centralized clusters. In S-Capsules, the variation (pose) component is the normal vector of a linear subspace that preserves most of the variance in the votes coming from below capsules and the activation component is computed based on the ratio of preserved variance.

Our experiments on a benchmark learning to diagnose task (Harutyunyan et al., 2017) defined on the publicly available MIMIC-III dataset (Johnson et al., 2016) highlight the success of capsule networks. Moreover, we confirm that the proposed S-Capsules converge faster than EM-Capsules. Finally, we show that the elements of the S-Capsules' variation (pose) vector are significantly correlated with the commonly used hand-engineered features.

2 METHODOLOGY

The learning to diagnose (phenotyping) task (Lipton et al., 2015; Harutyunyan et al., 2017) is a multivariate time series classification task, where we need to predict a patient's diseases based on his time series of vital signs and lab results. It is a multi-label classification task, meaning that a patient can be diagnosed with multiple diseases or no disease.

Due to lack of space, we describe both our customized EM-Capsules for this task and the proposed S-Capsules by outlining and comparing the three steps in their forward pass and defer the details to

Figure 2 in the appendix. Only step 3 is different between the two architectures. In both networks, capsules have two components: an activation $\alpha \in [0, 1]$ and a pose (variation)[1] vector $\mathbf{u} \in \mathbb{R}^d$. The choice of having a pose vector instead of a matrix in (Hinton et al., 2018) is due to the time series nature of our features.

Step 1: Extracting features. First, we use one-dimensional convolutions to extract lower dimensional features for processing by capsules. We use three residual blocks (He et al., 2016) with increasing dilation inspired by (Van Den Oord et al., 2016) to not only increase the receptive field of the convolutions but also reduce the dimension of the input with fewer layer and parameters. Finally, we flatten the output of the residual network to have a 120-dimensional vector ready to be processed by capsule layers.

Step 2: Primary capsules. In both architectures, we use two dense residual networks per capsule to create the activation and pose components of the primary capsules. The choice of residual blocks as transformation operations instead of the linear map is because in healthcare we do not have a formal understanding of the deformations in the data, whereas in computer vision distortions such as rotation are well-studied (Dieleman et al., 2016; Worrall et al., 2017). Residual blocks allow simple non-linear transformations without over-parameterization of the network.

Step 3: Capsule to capsule computation. The EM-Capsule network uses the EM-routing procedure as described in (Hinton et al., 2018) expect the fact that we choose the transformations to be residual blocks. To describe the capsule computations in S-Capsule networks, consider two layers L and $L + 1$ with n_L and n_{L+1} capsules, respectively. The jth capsule in layer $L + 1$ computes the weighted votes from the layer L capsules as $\mathbf{y}_{j,i} = \alpha_i R_{j,i}(\mathbf{u}_i)$ for each capsule i in layer L, where $R_{j,i}(\cdot)$ is a dense residual block. Then it concatenates all the weighted votes from layer L as a matrix $\mathbf{Y}_j \in \mathbb{R}^{n_L \times d}$ and computes the singular value decomposition of $\mathbf{Y}_j = \tilde{\mathbf{U}}\mathbf{S}\mathbf{V}^\top$. The pose vector for capsule j is simple the first (dominant) right singular vector $\mathbf{u}_j = \mathbf{V}[0, :]$, which is the normal vector of a linear subspace preserving most of the variance in the vote vectors of the capsules in layer L. The activation for the jth capsule is computed using the singular values s_k as

$$\alpha_j = \text{sigmoid}\left(\eta\left[\frac{s_1^2}{\sum_{k=1}^{\min(d,n_L)} s_k^2} - b\right]\right) = \text{sigmoid}\left(\eta\left[\frac{\|\mathbf{Y}_j\mathbf{u}_j\|_2^2}{\|\mathbf{Y}_j\|_F^2} - b\right]\right),$$

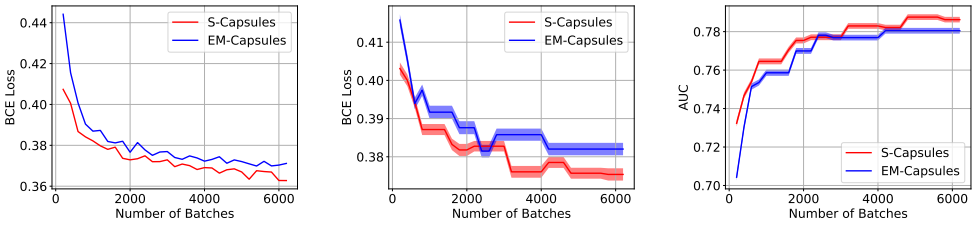
where b is discriminatively trained and η is linearly annealed during the training. Note that the ratio $\frac{s_1^2}{\sum_{k=1}^{\min(d,n_L)} s_k^2}$ is the fraction of variance of the votes from the below layer captured in the one-dimensional subspace defined by $\mathbf{u}_j$ and measures the agreement between the votes for the pose of the jth capsule.

We train the entire network in an end to end discriminative training with binary cross-entropy loss. We also extended the spread loss in (Hinton et al., 2018) to the multi-label setting, but it performed measurably worse than the binary cross-entropy loss in our experiments. We also found that adding a skip connection from the features extracted in Step 1 to the activations of the last capsule layer improves the performance.

Remarks. Step 3 in S-Capsules only need the top-1 SVD which is more efficient than the full decomposition, reducing the computational cost of the network.[2] The activations and poses computed in Step 3 are inherently normalized: we always have $\|\mathbf{u}_j\|_2 = 1$ and $\frac{s_1^2}{\sum_{k=1}^{\min(d,n_L)} s_k^2} \in \left[\frac{1}{\min(d,n_L)}, 1\right]$. Given a stable SVD implementation, the inherently normalized outputs stabilize the training of S-Capsule networks. Computation of the activations using the variance preserved in the top singular value has an additional self-annealing impact: random matrices usually have a sizable rank-1 component (Vershynin, 2010; Tropp et al., 2015), which can prevent the death of capsules in the initial phases of training.

[1] The term 'pose' is not meaningful in here, we use it to adhere to the terminology of (Hinton et al., 2018).

[2] At the time of submission, the `svd()` function in PyTorch did not support top-1 decomposition and batch computation, slowing down the algorithm from the ideal case. We hope to fix these issues soon.



(a) Training loss (b) Test loss (c) Test AUC

Figure 1: Faster convergence of S-Capsules compared to EM-Capsules using the same learning rate. In practice, we found S-Capsules can operate with larger learning rates.

3 Preliminary Experiments

The learning to diagnose benchmark in (Harutyunyan et al., 2017) extracts 78-dimensional multivariate time series for each patient from MIMIC-III dataset (Johnson et al., 2016). The data are divided into training/validation/test partitions of sizes 29,250/6,371/6,281 patients. We follow the preprocessing and discretization process in the benchmark and also crop the time series to the last 50 time stamps. We train all algorithms using Adam (Kingma & Ba, 2014) with batch size 64 and half the learning rate whenever the validation accuracy plateaus. We tune the other hyperparameters using the provided validation set.

Figure 1 shows the convergence behavior of EM-Capsules and S-Capsules as we train more batches. Figures 1a and 1b show the decrease of binary cross entropy and Figure 1c shows the increase of micro-AUC. The small rises and declines on figures b and c, respectively, are due to model selection with a separate validation set. The intervals in figures b and c are one standard deviation intervals obtained by 1000 times bootstrapping on the test set. For fairness, the learning rate is set to be equal for both algorithms, though S-Capsules could learn with higher rates. The figures confirm that S-Capsules learn faster and generalize better compared to EM-Capsules. S-Capsules achieve the final AUC of **80.50%** beating the accuracy of EM-Capsules and deep GRU networks —the common baseline—, which are 80.17% and 80.02%, respectively.

Interpreting pose vector of the output capsules. It is interesting to analyze if the pose vectors of the output capsules preserve the variations in the input data. While in computer vision this analysis can be done by visual inspection (Sabour et al., 2017), understanding the patterns in medical data requires significantly more expertise. Instead of visual inspection, we choose to construct the common hand-engineered medical features (Pollack et al., 1996; Lipton et al., 2015) for the continuous variables in the input time series and measure the correlation between each dimension of the pose vector and them. Given 13 continuous input variables and 7 features extracted from each, we construct 91 hand engineered features. We test the Spearman correlation between each of the 15 dimensions of the pose vector and the 91 features. After Bonferroni correction (Shaffer, 1995), at a p-value of 5%, we observe that **47.40%** of times the pose vector elements are significantly correlated with the hand-engineered features. This result indicates that the pose vectors do preserve a significant amount of variations in the input data. Clearly, we do not like this percentage to be too large either, as we know that the hand-engineered features are not the perfect summary of the input data.

4 Conclusion and Discussion

In this work, we customized capsule networks with EM routing (Hinton et al., 2018) for learning to diagnose task. We also proposed spectral capsule networks to improve stability and convergence speed of the capsule networks. Similar to EM-Capsules, S-Capsules are also spatial coincidence filters and look for agreement of the below capsules. However, spectral capsules measure the agreement by the amount of alignment in a linear subspace, rather than a centralized cluster. Setting aside the attention mechanism in EM-Capsules, the connection between S-Capsules and EM-Capsules is analogous to the connection between Gaussian Mixture Models and Principal Component Analysis. This analogy suggests why S-Capsules are more robust during the training. Our preliminary results confirm the superior convergence speed of the proposed S-Capsule network and preservation of variations in the data in its pose vectors.

ACKNOWLEDGMENTS

We would like to thank the developers of PyTorch for the awesome software that they make.

REFERENCES

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pp. 301–318, 2016.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999, 2016.

Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *arXiv preprint arXiv:1602.02660*, 2016.

Robert Gens and Pedro M Domingos. Deep symmetry networks. In *Advances in neural information processing systems*, pp. 2537–2545, 2014.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *arXiv preprint arXiv:1703.07771*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pp. 44–51. Springer, 2011.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix Capsules with EM Routing. In *International Conference on Learning Representations*, pp. 3859–3869, 2018.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 2017.

Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *CVPR*, volume 3, pp. 6, 2015.

Murray M Pollack, Kantilal M Patel, and Urs E Ruttimann. Prism iii: an updated pediatric risk of mortality score. *Critical care medicine*, 24(5):743–752, 1996.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Peter J Liu, Xiaobing Liu, Mimi Sun, Patrik Sundberg, Hector Yee, et al. Scalable and accurate deep learning for electronic health records. *arXiv preprint arXiv:1801.07860*, 2018.

Narges Razavian, Jake Marcus, and David Sontag. Multi-task prediction of disease onsets from longitudinal laboratory tests. In *Machine Learning for Healthcare Conference*, pp. 73–100, 2016.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pp. 3859–3869, 2017.

Juliet Popper Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46(1):561–584, 1995.

Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical intervention prediction and understanding with deep neural networks. In *Machine Learning for Healthcare Conference*, pp. 322–337, 2017.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

A Details of the architecture

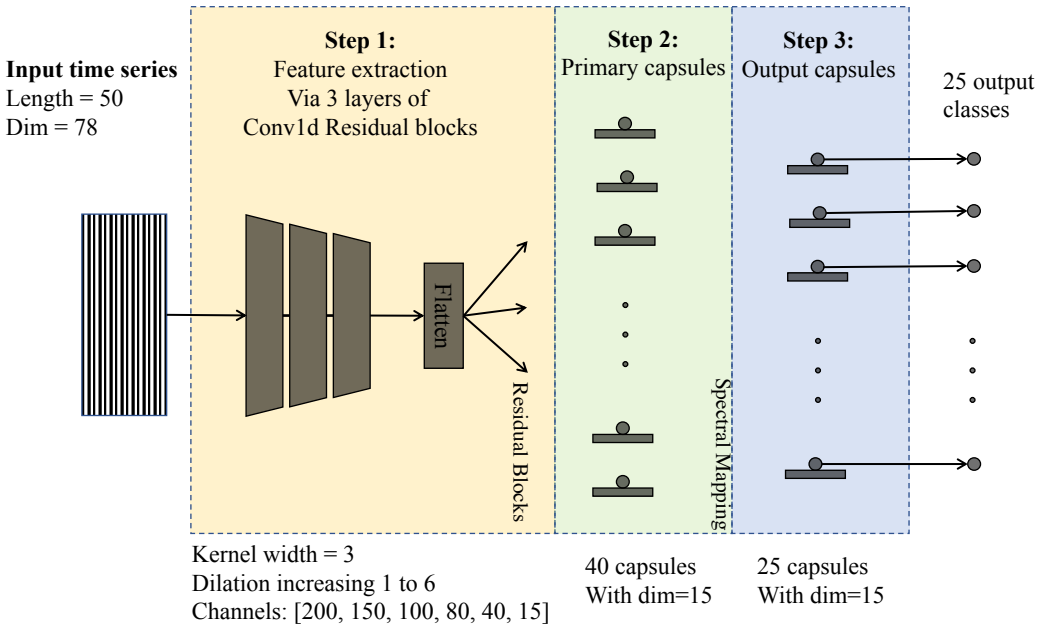


Figure 2: Details of the S-Capsules architecture. The mapping from flattened features and between two capsule layers are described in details in Section 2.