FORCED TO LEARN: DISCOVERING DISENTANGLED REPRESENTATIONS WITHOUT EXHAUSTIVE LABELS

Alexey Romanov & Anna Rumshisky
Department of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854, USA
{aromanov,arum}@cs.uml.edu

ABSTRACT

Learning a better representation with neural networks is a challenging problem, which was tackled extensively from different prospectives in the past few years. In this work, we focus on learning a representation that could be used for clustering and introduce a novel loss component that substantially improves the quality of produced clusters, is simple to apply to an arbitrary cost function, and does not require a complicated training procedure.

1 INTRODUCTION

In the past few years, a substantial amount of work has been dedicated to learning a better representation of the input data that can be either used in downstream tasks, such as KMeans clustering, or to improve generalizability or perfromance of the model. In general, these works can be divided into two categories: (1) approaches that require a complicated training procedure; (2) approaches that introduce a new loss component that can be easily applied to an arbitrary cost function; For example, approaches by Liao et al. (2016) and Xie et al. (2016) can be assigned to the first category, as they propose to iteratively refine the clusters during the training. In contrast, approaches by Cogswell et al. (2015) and Cheung et al. (2014) introduce new loss components that can be added to a cost function while training the model with a gradient descent algorithm. Our work belongs to the second category and focuses on the challenging problem of learning disentangled representations while having access to labels that do not fully reflect the underlying partitioning of the data, but still separate it into distinguishable groups.

For example, consider a case of predicting in-hospital mortality using multivariate physiological time series. This is a binary classification task which can be solved using a Recurrent Neural Network model, depicted on the Figure 1. During the regular training procedure with a sigmoid cross-entropy loss, the model tends to learn weights that lead to a strong activation of one of the neurons in the penultimate layer (`FC1`) for the instances that belong to the positive class and a strong activation of another neuron for the instances that belong to the negative class, whereas all other neurons tend to be not active for both classes (see the Figure 2a and Figure 2c). However, we would like to separate patients into more groups than just binary outcomes by applying a clustering algorithm to the learned representations of the patients. Thus, we would need the model to learn disentangled representations that can not only differentiate between the patients with different outcomes, but also between the patients with the same outcome using latent characteristics of the time series and properties for which we do not have labels.

In order to force the network to learn such disentangled representations, we propose a novel loss component that can be applied to an arbitrary cost function. Although it can be used in any type of model, including autoencoders, this paper focuses on a task of multivariate time series classification.

2 THE PROPOSED METHOD

Inspired by the work of Cheung et al. (2014) and Cogswell et al. (2015), we propose a loss component that, despite its simplicity, significantly increases the quality of the clustering over the produced

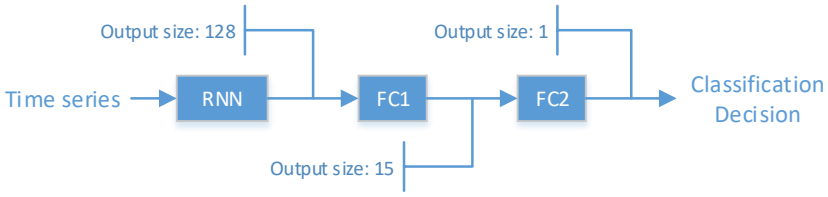
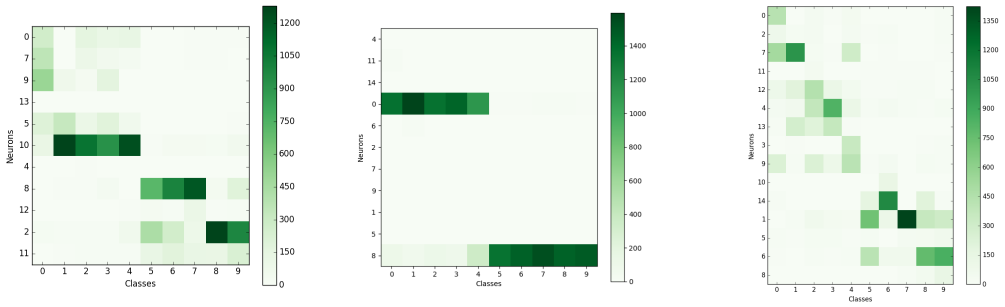


Figure 1: An RNN model for time series classification



(a) Baseline (b) Baseline + `DeCov` (c) Baseline + proposed loss

Figure 2: Number of samples for which the neurons on the y axis were active the most in the binary classification task. The classes 0-4 have the label 0, and the classes 5-9 have the label 1. See the section 3 for details.

by the model representation. Consider the model on the Figure 1. The layer `FC2` has size 1 and produces a binary classification decision, and the output of layer `FC1` is used to perform a KMeans clustering. Recall from the example in the introduction that we want to force the model to produce different representations for the samples that belong to the same class, but are different from each other. One way to do it would be to force the *rows* of the weight matrix W_{FC1} of the `FC1` layer be different from each other, leading to different patterns of activations in the output of the `FC1` layer.

Formally, it can be expressed as follows:

$$\mathcal{L}_W = \sum_{i=0}^{k} \sum_{j=i+1}^{k} \max(0, m - D_{\text{KL}}(d_i || d_j)) + \max(0, m - D_{\text{KL}}(d_j \parallel d_i)), \tag{1}$$

where k is the number of neurons in the target layer, m is the a hyperparameter that defines the desired margin, and $D_{\text{KL}}(d_i||d_j))$ is the Kullback-Leibler divergence between probability distributions d_i and d_j. d_l is obtained by converting the row l of the weight matrix W of the given layer into a probability distribution with the softmax function: $d_l = \text{softmax}(W_l)$.

3 Experiments

We performed initial experiments on the MNIST strokes sequences dataset de Jong (2016)[1] to validate our hypothesis. We performed our experiments with two different tasks: supervised binary classification task and unsupervised autoencoder model.

For the binary classification task, we split the examples into two groups: classes from 0 to 4 were assigned to the first group, and classes from 5 to 9 were assigned to the second group. During the training, the model should correctly predict the group of the given sample without having the access to the underlying classes. This experiment is a simplified version of the example we discussed in

[1] `https://github.com/edwin-de-jong/mnist-digits-stroke-sequence-data`

the introduction where we wanted to cluster the patients into meaningful groups, while having only the binary mortality outcomes, rather than the more fine-grained labels. The goal of the autoencoder model is to just reconstruct the input sequence. In both cases, we developed a GRU-based model using TensorFlow (Abadi et al., 2016) and used the output of the penultimate or intermediate layer to perform the KMeans clustering using the Scikit-learn package (Pedregosa et al., 2011). We report the average Adjusted Mutual Information score (Vinh et al., 2010) across three runs in the Table 1. We compare our loss component with `DeCov` (Cogswell et al., 2015) and `XCov` (Cheung et al., 2014) as these losses use similar ideas, even though they do not directly target improving the quality of clustering.

Table 1: Adjusted Mutual Information score for the MNIST strokes sequences experiments

Model	Binary classification			Autoencoder		
	9 clusters	10 clusters	11 clusters	9 clusters	10 clusters	11 clusters
Baseline	0.446	0.467	0.464	**0.443**	0.454	**0.468**
Baseline + DeCov (Cogswell et al., 2015)	0.285	0.287	0.288	0.427	0.443	0.465
Baseline + XCov (Cheung et al., 2014)	0.507	0.525	0.529	0.396	0.414	0.428
Baseline + proposed loss	**0.512**	**0.544**	**0.555**	0.439	**0.457**	0.460

4 DISCUSSION

Our initial experiments showed that the proposed loss component substantially increases the quality of KMeans clustering in terms of Adjusted Mutual Information Score. As we can see from the Table 1, our approach outperforms previously proposed loss components that use similar ideas and performs the best in the binary classification settings.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014.

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

Edwin D de Jong. Incremental sequence learning. *arXiv preprint arXiv:1611.03068*, 2016.

Renjie Liao, Alex Schwing, Richard Zemel, and Raquel Urtasun. Learning deep parsimonious representations. In *Advances in Neural Information Processing Systems*, pp. 5076–5084, 2016.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, 2016.