CS231B paper review – Arbel Harpak

Learning structure in gene expression data using deep architectures, with an application to gene clustering / Gupta, Wang & Ganapathiraju

---

Autoencoders have been successfully applied for initialization of deep networks for regression, as well as for dimensionality reduction. Traditionally, the hidden layer of autoencoders has a smaller size (number of neurons, typically set in one kernel) than the input layer. A novel type of autoencoders, termed Denoising Autoencoders (DAs) involves adding a corruption layer after the input layer. Notably, the corruption layer parameters are not learned but instead preset, and the objective function is still weighted with respect to the true input. Intuitively, this procedure should force the network to guess missing/corrupted values, and thereby develop robustness to noisiness in the data.

In their work, Gupta, Wang and Ganapathiraju perform unsupervised learning of gene expression networks from gene expression data. They build a classifier by training a Denoising Autoencoder (DAs).

Much of the motivation for this work comes from Yee and Walter (2001) and from Larochelle et al. (2007). Yee and Walter (2001) applied PCA to gene expression microarray data and found that clustering patterns do not reflect gene networks very well. This perhaps suggests that linear transformation are limited in capturing a low dimensional representation of gene expression profiles. Larochelle et al. later showed that clustering following a (learned) nonlinear transformation with DAs offers substantial improvement to classification problems in the domain of image processing.

Recently, DAs have proved useful for extracting low dimensionality representation of data which is robust to increased noise or random missingness in the input. DAs learn to guess the missing or corrupted value. Gupta et al. further use a stacked DA (SDA), which is essentially a multilayered DA. Instead of setting random weights, stacked DAs perform algorithms involve an initialization step in which the weights are greedily optimized in each layer such that they minimize the next layer's loss. Then, the weights are fine-tuned by training the entire network to minimize the global (last layer) loss.

Gupta et al. build an SDA and apply it to two yeast datasets of the expression of several hundreds (237, 384) of genes at 17 time points spanning yeast cell cycle. They do not perform extensive cross validation, but instead perform validation using an evaluation statistic that is fundamentally different from the objective function—and is applied to the entire dataset used for learning. Namely, for evaluation of their clustering performance, the authors validate their results against several "gold standard" external clustering data for the same genes. They use the Adjusted Rand Index (ARI) which measures how

frequently pairs of genes agree across methods (Gupta et al. vs. the "gold standard") with respect to sharing a label (i.e. being in the same cluster).

Gupta et al. conclude that a shallow DA with a single hidden layer performs very similarly to deeper architectures investigated. As training a single-layered network significantly reduces computation time, they opt for sticking with a single-layer DA in future studies; although in the final conclusion, they mention they plan to revisit deeper architecture in their own future work. Gupta et al. also conclude that DAs outperform vanilla, no-added-noise autoencoders. However, despite the authors' general claim, this conclusion seems to be supported only by the results on one of the two datasets. Namely, for dataset 1 corresponding figure 2 DAs outperform 0% noise autoencoders; for dataset 2 corresponding to figure 3, there is no noticeable change in performance (as measured by ARI) with varying number of nodes.

Figure 2 and 3 seem very noisy—not only within the results of each dataset, but also the low reproducibility of performance across these two very similar datasets: figure 2's trend exhibits a relationship between the number of nodes and the performance, whereas figure 3 does not exhibit any clear relationship. Overall, this seems to hint that the autoencoder was not sufficiently optimized. This could be due to some of the hyperparameters, with immediate suspects being the small batch sizes (and small range) attempted, the fixed learning rate. It is unclear how much variation exists in the data and could be leveraged for learning, as there are only a few hundred genes in 17 time points but otherwise similar treatment.

Another concern that I had related to the objective function (cross entropy) measuring something quite different from the validation metric (ARI with "gold standard" classifications). It is unclear why the authors choose not to use cross validation. Finally, If the authors were to perform simulations of co-varying expression profiles, and trained their model on simulated data, we might be able to get a better sense of what the DA is capturing, and what it is missing.

I found the notion of adding a first noise layer very intriguing, and it was presented very clearly. It seems like the precise choice of the noise model could be very important in the DA setting. For DAs to be successful, intuition dictates that the model for the first layer should capture noisy aspects that typically exist in the data, but are not associated with the signal—here, by signal I mean memberships in gene networks (or regulatory cascades). The authors choose to use a dropout masking model of noise. Typically for gene expression data, a Gaussian noise model is assumed. As the authors point out themselves, with this model, the distinction between performances across noise levels remained unclear. I think that

investigating other noise models—including explicit modelling the noise in the gene expression domain—could help reveal substantially more about the utility of DAs for learning structure in gene expression data.