(Not Too) Deep Clustering via Clustering the Local Manifold of an Autoencoded Embedding

simple but effective deep clustering

Ryan McConville
Raúl Santos-Rodríguez, Robert Piechocki, Ian Craddock

University of Bristol

Introduction

Deep Clustering

- learning a good representation is beneficial to clustering
- *and* clustering provides supervisory signals for the representation
- *so* integrate both into a single model

This is a popular idea within deep clustering.

Autoencoders

- An autoencoder is a (usually deep) neural network which takes an take an input x and predicts an output x.
- Often the code z that is of a smaller dimension than x.

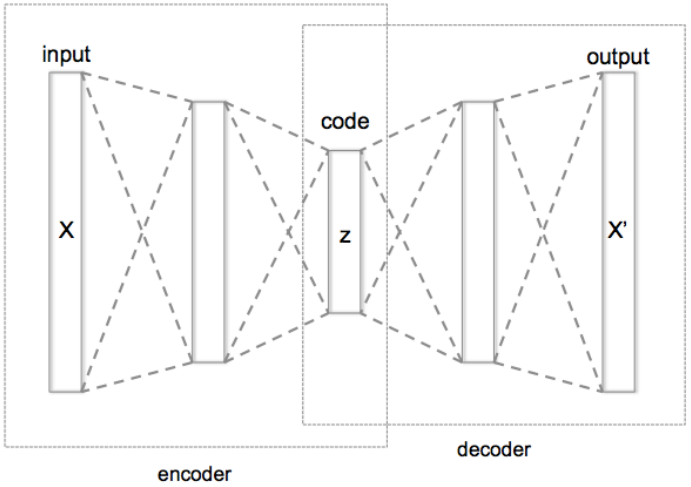


Image: wikimedia

Self-Supervision

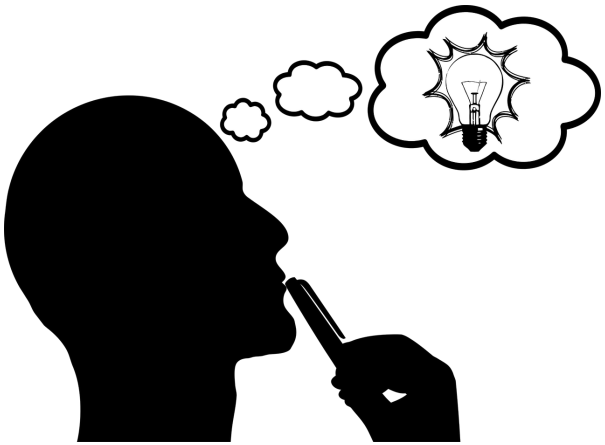
Idea: Let the data provide the supervision!

How existing approaches (typically) work:

1. Pretrain the autoencoder without the clustering objective.
2. Cluster the representation using your favourite clustering algorithm to obtain initial centroids.
3. Use these centroids as the labels for each cluster.
4. Introduce a new loss function (clustering objective) that minimizes the difference between the representation and the centroids.

Many deep clustering approaches take this approach (DEC, IDEC, etc.). However, we had an idea..

- Why do approaches such as tSNE and UMAP produce visualizations that humans often use for interpreting data and clusters?
- Can we replace the self-supervision with something which humans consider good clusters rather than the algorithm?

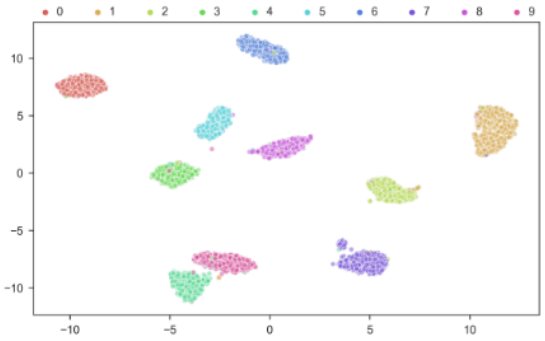


The N2D method

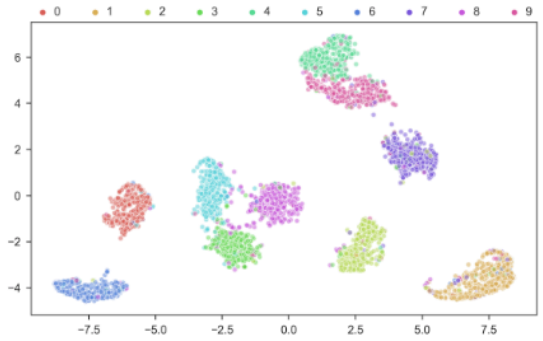
The Not-Too-Deep (N2D) method for deep clustering.

1. Train the (MLP) autoencoder with a regular reconstruction loss.
2. Apply a manifold learner to the autoencoded embedding.
3. Apply a standard clustering algorithm (e.g., k-means, GMM) to the output of the mainfold learner.
4. Done!

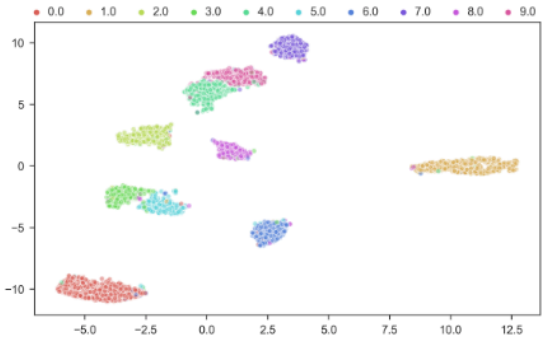
So lets run some experiments



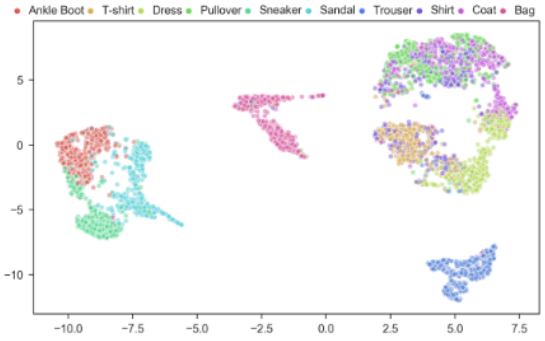
(a) MNIST



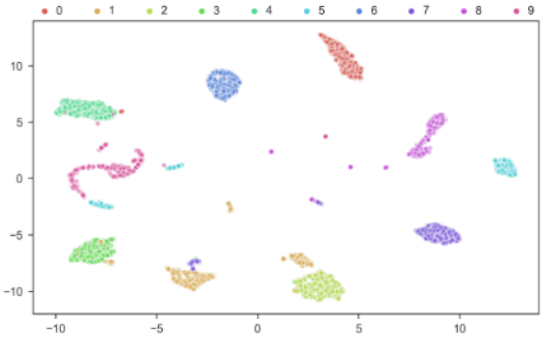
(b) MNIST-test



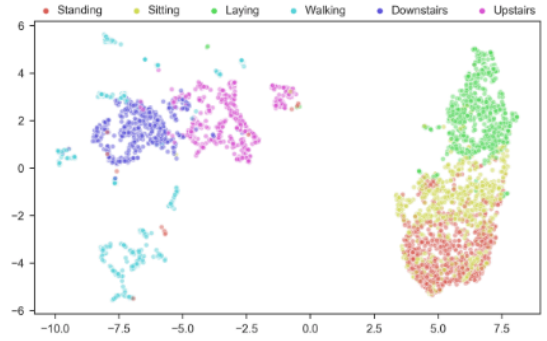
(c) USPS



(d) Fashion



(e) pendigits



(f) HAR

Why UMAP?

	MNIST		MNIST-test		USPS		Fashion		pendigits		HAR	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
GMM	0.389	0.333	0.464	0.465	0.562	0.541	0.463	0.514	0.674	0.683	0.585	0.648
AE	0.809	0.835	0.769	0.745	0.707	0.705	0.569	0.591	0.807	0.757	0.552	0.483
Isomap	—	—	0.896	0.755	0.781	0.787	—	—	0.756	0.792	0.608	0.677
N2D (Isomap)	—	—	0.797	0.77	0.660	0.727	—	—	0.826	0.820	0.632	0.580
TSNE	0.768	0.810	0.806	0.823	0.720	0.830	0.608	0.651	0.893	0.872	0.647	0.736
N2D (TSNE)	0.978	0.940	0.948	0.883	0.810	0.858	0.588	0.658	0.785	0.826	0.768	0.670
UMAP	0.825	0.880	0.857	0.819	0.804	0.845	0.588	0.656	0.819	0.856	0.552	0.696
N2D (UMAP)	0.979	0.942	0.948	0.882	0.958	0.901	0.672	0.684	0.885	0.863	0.801	0.683

Results Table

	MNIST		MNIST-test		USPS		Fashion		pendigits		HAR	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means	0.532	0.450	0.546	0.501	0.668	0.626	0.474	0.512	0.666	0.681	**0.599**	0.588
SC	0.680	0.759	0.667	0.712	0.656	0.796	0.551	**0.630**	0.724	**0.784**	0.538	**0.741**
GMM	0.389	0.333	0.464	0.465	0.562	0.540	0.463	0.514	0.673	0.682	0.585	**0.648**
DeepCluster	0.797	0.661	0.854	0.713	0.562	0.54	0.542	0.510	—	—	—	—
DCN	0.830	0.810	0.802	0.786	0.688	0.683	0.501	0.558	.72	.69	—	—
DEC	0.863	0.834	0.856	0.830	0.762	0.767	0.518	0.546	0.701	0.678	0.565	0.587
IDEC	0.881	0.867	0.846	0.802	0.761	0.785	0.529	0.557	**0.784**	0.723	**0.642**	0.609
SR-*k*-means	0.939	0.866	0.863	0.873	0.901	**0.912**	0.507	0.548	—	—	—	—
VaDE	0.945	0.876	0.287	0.287	0.566	0.512	0.578	**0.630**	—	—	—	—
ClusterGAN	0.964	0.921	—	—	—	—	**0.630**	**0.640**	**0.770**	**0.730**	—	—
JULE	0.964	0.913	**0.961**	**0.915**	**0.950**	**0.913**	0.563	0.608	—	—	—	—
DEPICT	0.965	0.917	**0.963**	**0.915**	0.899	0.906	0.392	0.392	—	—	—	—
DBC	0.964	0.917	—	—	—	—	—	—	—	—	—	—
DAC	**0.978**	**0.935**	—	—	—	—	—	—	—	—	—	—
ASPC-DA	**0.988**	**0.966**	**0.973**	**0.936**	**0.982**	**0.951**	**0.591**	**0.654**	—	—	—	—
N2D	**0.979**	**0.942**	0.948	0.882	**0.958**	0.901	**0.672**	**0.684**	**0.885**	**0.863**	**0.801**	**0.683**

Conclusions

- Deep neural networks (e.g., autoencoders) are very effective at clustering.
- Manifold learning on top of the autoencoder seems to do something interesting to the autoencoder representation that makes it more cluster-able.
- **Simple approach, but performs well.**
- More research is needed to better understand why, and exploit this idea further.
  - Already numerous follow-up papers from other authors further improving this line of research.
- Code is available on GitHub (`https://github.com/rymc/n2d/`).