Deep Learning based Clustering

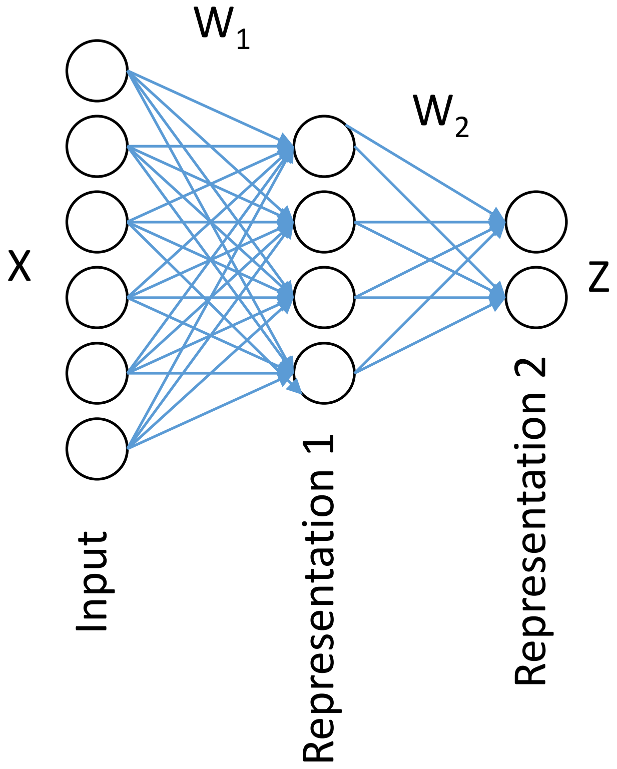
Presented by

Angshul Majumdar

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Unsupervised Learning Frameworks

Formalism



Representation 1:

$$Z_1 = \varphi\left(W_1 X\right)$$

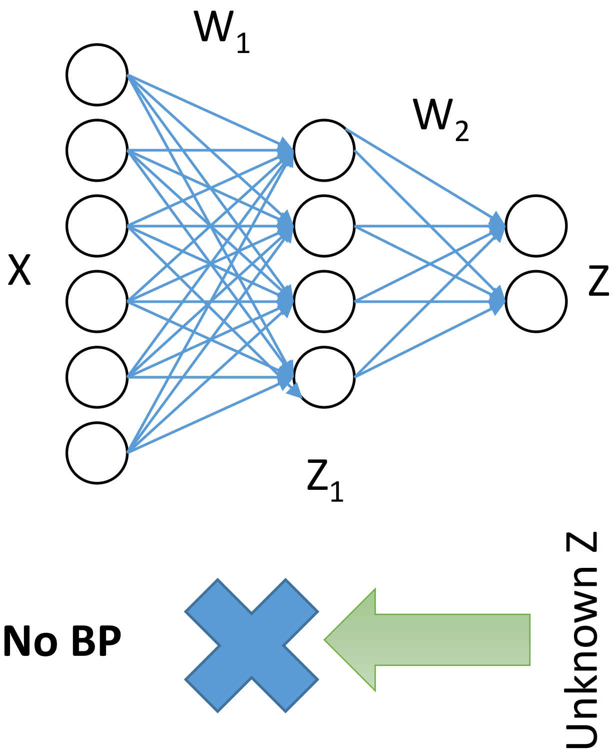
Representation 2:

$$Z = \varphi\left(W_2 H_1\right) = \varphi\left(W_2 \varphi\left(W_1 X\right)\right)$$

Cost Function:

$$\min_{W_1, W_2, Z} \left\| Z - \varphi\left(W_2 \varphi\left(W_1 X\right)\right) \right\|_F^2$$

Bottleneck – Trivial Solution



$$\min_{W_1,W_2,Z}\left\|Z-\varphi\left(W_2\varphi\left(W_1X\right)\right)\right\|_F^2$$

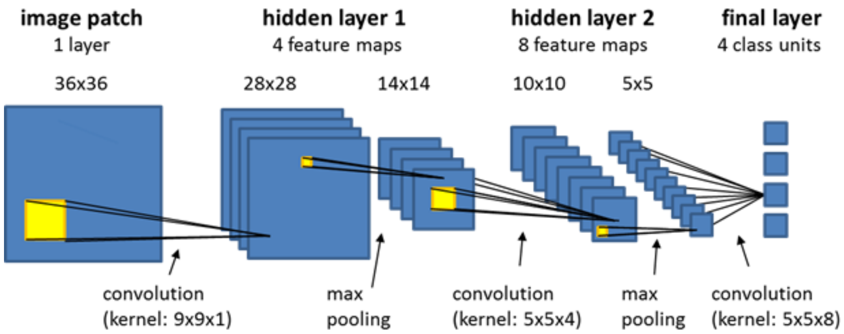
Nothing to backpropagate from!

Trivial Solution:

$W_1=0, W_2=0$ and $Z=0$

Satisfied for any X!

Do Convolutions Help?



$$y[k] = w[n] \otimes x[n] = \sum_{i=-\infty}^{\infty} x[i]w[k-i]$$

$w[n]$: length 4, $x[n]$: length 3

Computing Convolutions

$$y[0] = \sum_{i=-\infty}^{\infty} x[i]w[-i] = x[0]w[0] + 0 + 0$$

$$y[1] = \sum_{i=-\infty}^{\infty} x[i]w[1-i] = x[0]w[1] + x[1]w[0] + 0$$

$$y[2] = \sum_{i=-\infty}^{\infty} x[i]w[2-i] = x[0]w[2] + x[1]w[1] + x[2]w[0]$$

$$y[3] = \sum_{i=-\infty}^{\infty} x[i]w[3-i] = x[0]w[3] + x[1]w[2] + x[2]w[1]$$

$$y[4] = \sum_{i=-\infty}^{\infty} x[i]w[4-i] = x[1]w[3] + x[2]w[1] + 0$$

$$y[5] = \sum_{i=-\infty}^{\infty} x[i]w[5-i] = x[2]w[3] + 0 + 0$$

Toeplitz Form

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \end{bmatrix} = \begin{bmatrix} w[0] & 0 & 0 \\ w[1] & w[0] & 0 \\ w[2] & w[1] & w[0] \\ w[3] & w[2] & w[1] \\ 0 & w[3] & w[2] \\ 0 & 0 & w[3] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix}$$

$y = \widehat{W}x$, $\widehat{W}$ is the Toeplitz representation of w

Back to square one!

$$Z = \varphi\left(\widehat{W}_2\left(\varphi\left(\widehat{W}_1 X\right)\right)\right)$$

For two layer convolutional NN

Cost function

$$\min_{\widehat{W}_1, \widehat{W}_2, Z} \left\| Z - \varphi\left(\widehat{W}_2\left(\varphi\left(\widehat{W}_1 X\right)\right)\right) \right\|_F^2$$

Satisfied by the trivial solution

$\widehat{W}_1 = 0$ or $w_1 = 0$

$\widehat{W}_2 = 0$ or $w_2 = 0$

$Z = 0$

Adding clustering loss

Two examples

K-means

$$\min_{z_j, h_{ij}} \sum_{i=1}^{k} \sum_{j=1}^{n} h_{ij} \left\| z_j - \mu_i \right\|_2^2$$

$$h_{ij} = \begin{cases} 1 & \text{if } x_j \in \text{Cluster i} \\ 0 & \text{otherwise} \end{cases}$$

Equivalent to matrix factorization

$$\min_{Z,H} \left\| Z - ZH^T \left(HH^T \right)^{-1} H \right\|_F^2$$

Sparse Subspace

$$\min_{c_i} \sum_i \left\| z_i - Z_{i^c} c_i \right\|_2^2 + \lambda \left\| c_i \right\|_1$$

$\forall i$ in $\{1, \ldots, m\}$

$$A = |C| + |C|^T$$

N-cuts on A

$$\min_C \left\| Z - ZC \right\|_F^2 + \lambda \left\| C \right\|_1$$

$$s.t.\ C_{ii} = 0$$

Clustering embedded NN

$$\min_{W_1,W_2,Z,H} \underbrace{\left\| Z-\varphi\left(W_2\varphi\left(W_1X\right)\right)\right\|_F^2}_{NN} + \underbrace{\left\| Z-ZH^T\left(HH^T\right)^{-1}H\right\|_F^2}_{K-means}$$

Trivial solution, again!

$W_1 = 0, W_2 = 0, Z = 0, H = 0$. For all X

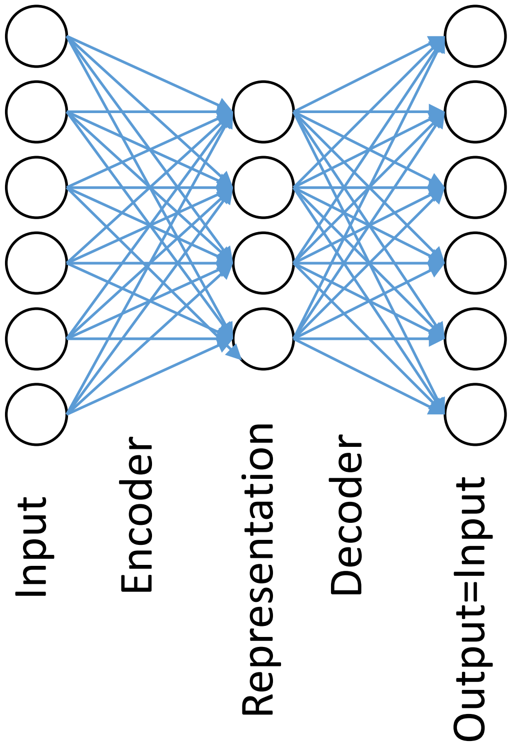
$$\min_{W_1,W_2,Z,H} \underbrace{\left\| Z-\varphi\left(W_2\varphi\left(W_1X\right)\right)\right\|_F^2}_{NN} + \underbrace{\left\| Z-ZC\right\|_F^2+\lambda\left\|C\right\|_1}_{Sparse\ Subspace\ Clustering}$$

As before. Trivial solution!

$W_1 = 0, W_2 = 0, Z = 0, C = 0$. For all X

Way Out

Autoencoder



Self supervision

$$\min_{W_{Enc},W_{Dec}} \left\| X - W_{Dec}\varphi(W_{Enc}X) \right\|_F^2$$

Similarly one can also have CAE

Pros

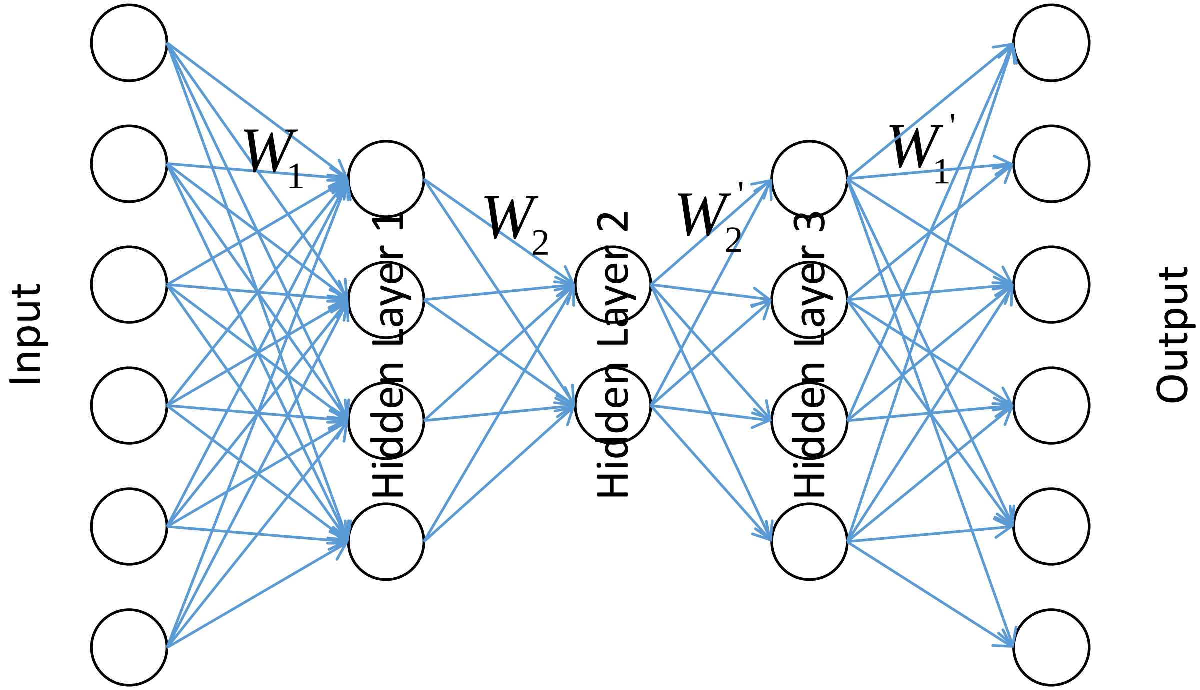
- Mathematically amenable cost function. Easy to add penalties.
- Solvable by BP (gradient descent).

Cons

- Need to learn double the number of parameters. Overfitting.

Stacked Autoencoder

$$\min_{W_1,W_2} \left\| X - W_1' \varphi\left(W_2' \varphi\left(W_2 \varphi\left(W_1 X\right)\right)\right)\right\|_F^2$$



- Same Pros and Cons extend to stacked (deep) autoencoders.

Publications – almost all on SAE!

1. F. Tian, B. Gao, Q. Cui, E. Chen and T. Y. Liu, "Learning deep representations for graph clustering," AAAI conference on Artificial Intelligence, pp. 1293-1299, 2014.

2. X. Peng, S. Xiao, J. Feng, W. Y. Yau and Z. Yi, "Deep Sub-space Clustering with Sparsity Prior," International Joint Conference on Artificial Intelligence, pp. 1925-1931, 2016.

3. J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” International Conference on Machine Learning, pp. 478-487, 2016.

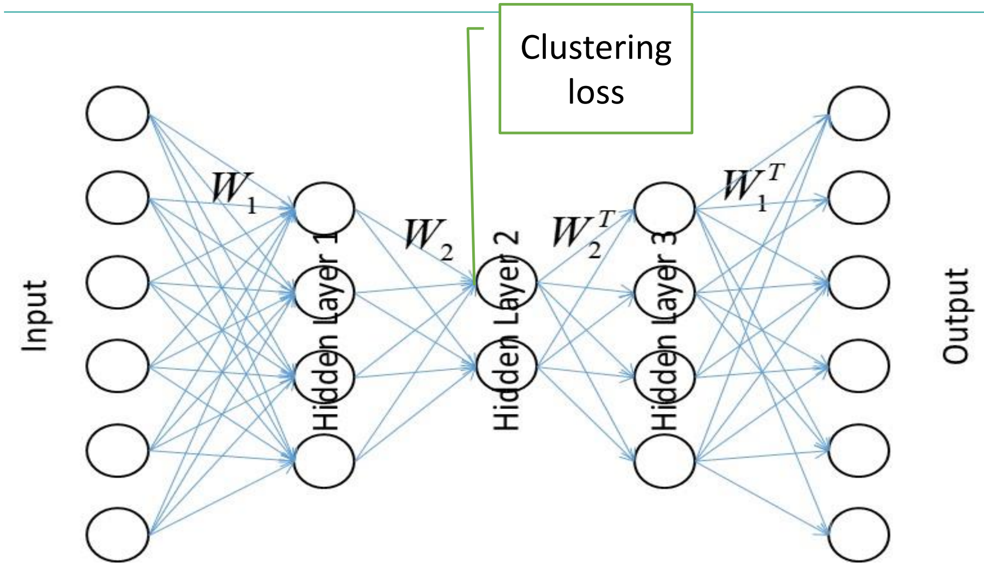
4. B. Yang, X. Fu, N. D. Sidiropoulos and M. Hong, “Towards k-means-friendly spaces: Simultaneous deep learning and clustering,” International Conference on Machine Learning, pp. 3861-3870, 2017.

5. M. M. Fard, T. Thonet and E. Gaussier, “Deep k-means: Jointly clustering with k-means and learning representations,” Pattern Recognition Letters, vol. 138, pp.185-192, 2020.

6. X. Guo, X. Liu, E. Zhu and J. Yin. "Deep clustering with convolutional autoencoders." International Conference on Neural Information Processing, pp. 373-382, 2017.

7. X. Yang, C. Deng, F. Zheng, J. Yan and W. Liu, "Deep Spectral Clustering Using Dual Autoencoder Network," IEEE Conference on Computer Vision and Pattern Recognition, pp. 4061-4070, 2019.

SAE based Clustering


Clustering loss

X. Peng, S. Xiao, J. Feng, W. Y. Yau and Z. Yi, "Deep Sub-space Clustering with Sparsity Prior," International Joint Conference on Artificial Intelligence, pp. 1925-1931, 2016.

$$\min_{W_1',W_2',W_1,W_2,C} \underbrace{\left\| X - W_1'\varphi\left(W_2'\varphi\left(W_2\varphi\left(W_1X\right)\right)\right)\right\|_F^2}_{Stacked\ Autoencoder} + \underbrace{\left\| Z - ZC \right\|_F^2 + \lambda \left\| C \right\|_1}_{Sparse\ Subspace\ Clustering}$$

$$s.t.\ Z = \varphi\left(W_2\varphi\left(W_1X\right)\right)$$

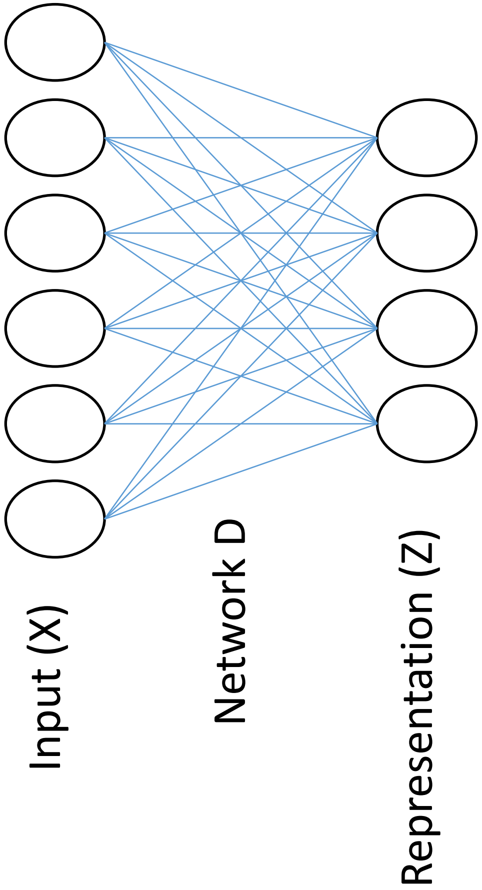
One more

$$\min_{W_1',W_2',W_1,W_2,H} \underbrace{\left\| X - W_1' \varphi\left(W_2' \varphi\left(W_2 \varphi\left(W_1 X\right)\right)\right) \right\|_F^2}_{Stacked\ \ Autoencoder}$$

$$+ \underbrace{\left\| Z - ZH^T \left(HH^T\right)^{-1} H \right\|_F^2}_{K-means} \quad s.t.\ Z = \varphi\left(W_2 \varphi\left(W_1 X\right)\right)$$

B. Yang, X. Fu, N. D. Sidiropoulos and M. Hong, “Towards k-means-friendly spaces: Simultaneous deep learning and clustering,” International Conference on Machine Learning, pp. 3861-3870, 2017.

RBM – Another Possibility



- RBM Cost Function:

$$E(z,x) = -a^T z - b^T x - x^T Dz$$

$$P(z,h) = \frac{e^{-E(z,h)}}{Z}$$

- Deeper extensions possible – DBM and DBN.

Issues with RBM

- Cost function is unwieldy and mathematically inflexible.
- Contrastive divergence is only an approximate solution.
- Difficult to incorporate other penalties.

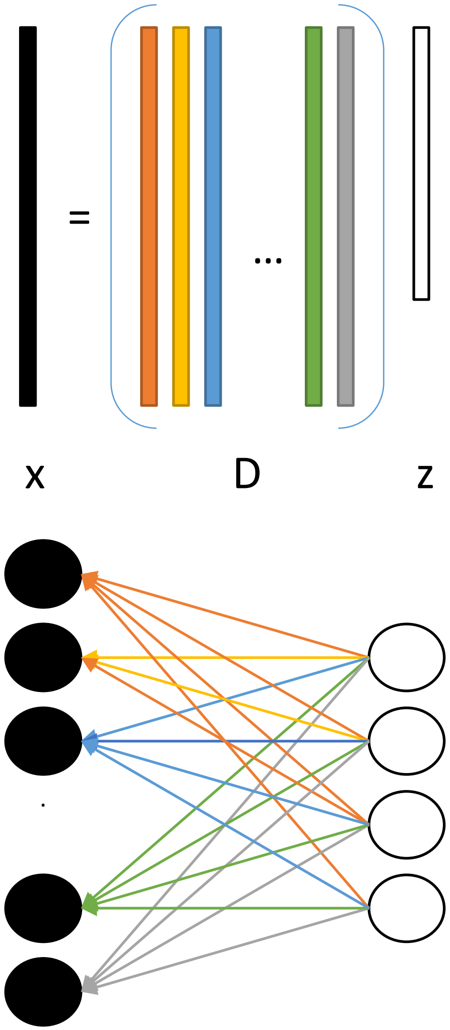
No Clustering Formulation

Newer paradigm

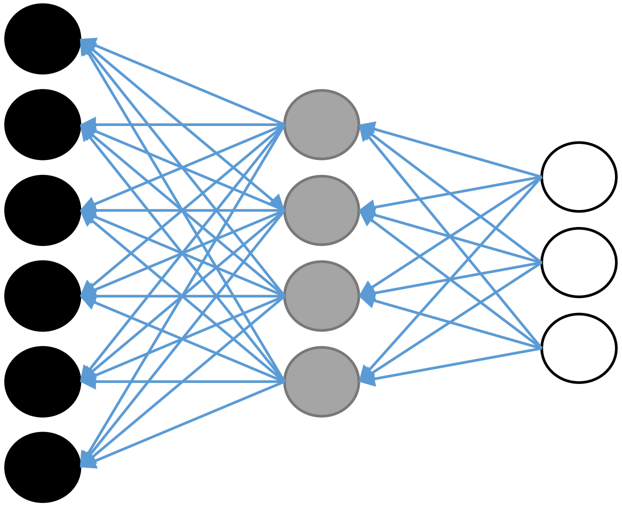
Dictionary Learning

- The basis act as connections between representation layer and input.

- The direction is opposite to that of a conventional neural net.



Deep Dictionary Learning



- Repeat the same building block again and again.
- Start with data at the input. Ins subsequent layers, use the representation from the previous layer as input.

Optimization

- In reality the problem is solved using a single optimization.
- Can harness powerful techniques like ADMM and PPXA.

Shallow dictionary learning

$$\min_{D,Z} \|X - DZ\|_F^2$$

Deep dictionary learning

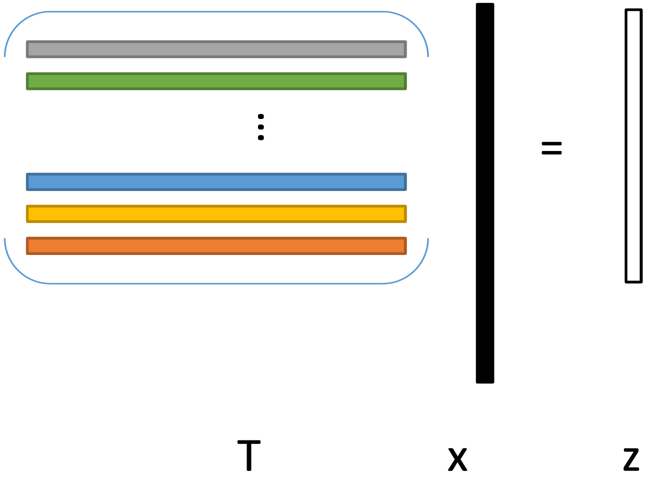
$$\min_{D,Z} \|X - D_1 D_2 D_3 Z\|_F^2$$

$$\text{s.t. } Z \geq 0, D_3 Z \geq 0, D_2 D_3 Z \geq 0$$

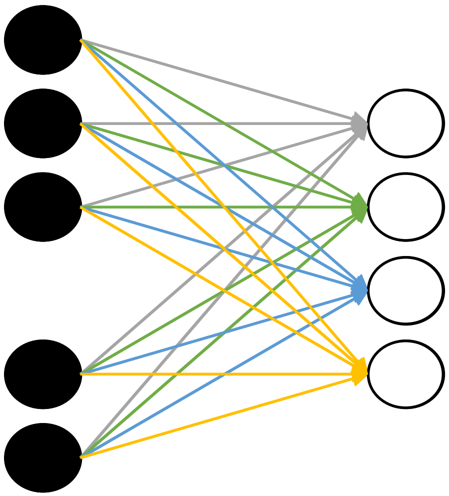
- Fully unsupervised framework. No trivial solution.
- Implies possibility of adding clustering penalty.

Transform Learning

- Analysis formulation. Basis operates on the signal to generate coefficients.



- In a neural network representation, it is feedforward. Akin to standard neural network.



Formulation

- Learns a basis such that it operates on samples to produce coefficients.

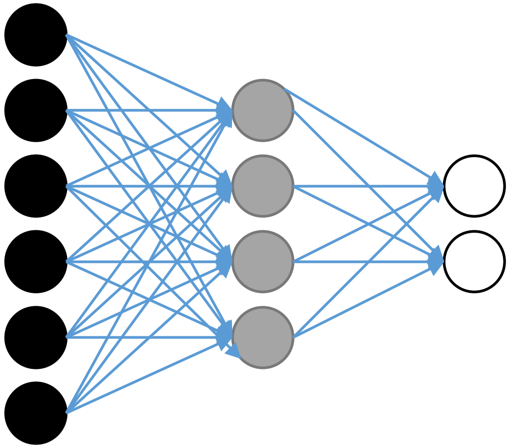
$$\underset{\textit{transform}}{T}\ \underset{\textit{samples}}{X} = \underset{\textit{coefficients}}{Z}$$

- Learning is expressed as follows …

$$\min_{T,Z} \|TX - Z\|_F^2 + \lambda\left(\|T\|_F^2 - \log\det T\right)$$

- Prevents the trivial solution with additional penalties

Deep Transform Learning



- Repeat units of transform learning to form deep architecture.

$$\min_{T_1,T_2,T_3,Z} \|T_3T_2T_1X - Z\|_F^2 + \lambda\sum_{i=1}^{3}\left(\|T_i\|_F^2 - \log\det T_i\right)$$

$$\text{s.t. } T_1X \geq 0, T_2T_1X \geq 0, Z \geq 0$$

DDL based clustering

- Main idea is to incorporate clustering loss at the last layer of dictionary learning.

$$\min_{D_1,D_2,D_3,Z,H} \underbrace{\left\| X - D_1 D_2 D_3 Z \right\|_F^2}_{\text{DDL}} + \mu \underbrace{\left\| Z - ZH^T \left(HH^T \right)^{-1} H \right\|_F^2}_{\text{K-means}}$$

$$\text{s.t. } \underbrace{D_2 D_3 Z \geq 0, D_3 Z \geq 0, Z \geq 0}_{\text{ReLU activation}}$$

- Can be easily solved using ADMM.
  - Update each of the variables separately.

A. Goel and A. Majumdar, "Sparse Subspace Clustering Friendly Deep Dictionary Learning for Hyperspectral Image Classification," IEEE Geosciences and Remote Sensing Letters (in print)

Solution

$$D_1 \leftarrow \min_{D_1} \left\| X - D_1 D_2 D_3 Z \right\|_F^2$$

$$D_1^k = X Z_1^{\dagger}, \text{ where } Z_1 = D_2^{k-1} D_3 Z^{k-1}$$

$$D_2 \leftarrow \min_{D_2} \left\| X - D_1 D_2 D_3 Z \right\|_F^2$$

$$D_2^k = \left(D_1^k \right)^{\dagger} X Z_2, \text{ where } Z_2 = D_3^{k-1} Z^{k-1}$$

$$\min_{W',W,x} \left\| y - Ax \right\|_2^2 + \lambda \sum_i \left(\left\| P_i x - W' \varphi(W P_i x) \right\|_2^2 + \mu \left\| \varphi(W P_i x) \right\|_1 \right)$$

$$D_3 \leftarrow \min_{D_3} \left\| X - D_1 D_2 D_3 Z \right\|_F^2$$

$$D_3^k = \left(D_1^k D_2^k \right)^{\dagger} X \left(Z^{k-1} \right)^{\dagger}$$

$$Z \leftarrow \min_{Z} \left\| X - D_1 D_2 D_3 Z \right\|_F^2 + \mu \left\| Z - Z H^T \left(H H^T \right)^{-1} H \right\|_F^2$$

$$H^k \leftarrow \min_{H} \left\| Z - Z H^T \left(H H^T \right)^{-1} H \right\|_F^2$$

DTL based clustering

- Same as before. Plug-in a clustering loss after the last layer of deep transform learning

$$\min_{T_1,T_2,T_3,Z,C} \underbrace{\left\|T_3T_2T_1X - Z\right\|_F^2 + \lambda\sum_{i=1}^{3}\left(\left\|T_i\right\|_F^2 - \log\det T_i\right)}_{Deep\ Transform\ Learning}$$

$$+\gamma \underbrace{\sum_i \left\|z_i - Z_{i^c}c_i\right\|_2^2 + R(c_i)}_{Subspace\ Clustering}$$

$$s.t.\ T_1X \geq 0 \text{ and } T_2T_1X \geq 0$$

$$\min_{T_1,T_2,T_3,X_2,X_3,Z,C} \left\|T_3X_3 - Z\right\|_F^2 + \left\|T_2X_2 - X_3\right\|_F^2 + \left\|T_1X - X_2\right\|_F^2$$

$$+\lambda\sum_{i=1}^{3}\left(\left\|T_i\right\|_F^2 - \log\det T_i\right) + \gamma\sum_i \left\|z_i - Z_{i^c}c_i\right\|_2^2 + R(c_i)$$

$$s.t.\ X_3 \geq 0 \text{ and } X_2 \geq 0$$

Solution

$$\text{P1}: \min_{T_1} \left\|T_1 X - X_2\right\|_F^2 + \lambda\left(\left\|T_1\right\|_F^2 - \log\det T_1\right)$$

$$\text{P2}: \min_{T_2} \left\|T_2 X_2 - X_3\right\|_F^2 + \lambda\left(\left\|T_2\right\|_F^2 - \log\det T_2\right)$$

$$\text{P3}: \min_{T_3} \left\|T_3 X_3 - Z\right\|_F^2 + \lambda\left(\left\|T_3\right\|_F^2 - \log\det T_3\right)$$

$$\text{P4}: \min_{X_3} \left\|T_3 X_3 - Z\right\|_F^2 + \left\|T_2 X_2 - X_3\right\|_F^2 \; s.t.\; X_3 \geq 0$$

$$\text{P5}: \min_{X_2} \left\|T_2 X_2 - X_3\right\|_F^2 + \left\|T_1 X - X_2\right\|_F^2 \; s.t.\; X_2 \geq 0$$

$$\text{P6}: \min_{Z} \left\|T_3 X_3 - Z\right\|_F^2 + \gamma \sum_i \left\|z_i - Z_{i^c} c_i\right\|_2^2$$

$$\text{P7}: \min_{C} \sum_i \left\|z_i - Z_{i^c} c_i\right\|_2^2 + R(c_i)$$

J. Maggu, A. Majumdar, E. Chouzenoux and G. Chierchia, “Deeply Transformed Subspace Clustering”, Signal Processing, Vol. 174, 107628, 2020.
J. Maggu, A. Majumdar and E. Chouzenoux, “Transformed Subspace Clustering”, IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1796-1801, 2021

What next?

Convolutional transform learning

- Convolutional neural networks have replaced almost all other DNN frameworks in recent years.
- Unfortunately, they can only be used in a supervised fashion.
- How do we make them unsupervised?
  - Naïve approach doesn't work. Discussed before.
  - Leads to trivial solution with all ZERO weights.
- Cues from transform learning for preventing trivial solution.

$$\min_{T,Z} \|TX - Z\|_F^2 + \lambda\left(\|T\|_F^2 - \log\det T\right)$$

CTL formulation

Convolution operation

$$t_m * x^{(k)} = z_m^{(k)}, \ \forall \ m \in \{1 \ldots M\} \text{ and } \forall k \in \{1 \ldots K\}$$

Optimization

$$\min_{(t_m)_m, \left(z_m^{(k)}\right)_{m,k}} \sum_{k=1}^{K} \sum_{m=1}^{M} \left(\left\| t_m * x^{(k)} - z_m^{(k)} \right\|_2^2 + \psi\left(z_m^{(k)}\right) \right) + \lambda \left\{ \left\| T \right\|_F^2 - \log\det(T) \right\}$$

Matrix vector form

$$\min_{T,Z} \left\| T \bullet X - Z \right\|_F^2 + \Psi(X)) + \lambda \left\{ \left\| T \right\|_F^2 - \log\det(T) \right\}$$

$$X = \left[x^{(1)} | \ldots | x^{(K)} \right]^T, \ Z = \left[z_1^{(k)} | \ldots | z_M^{(k)} \right]_{1 \le k \le K}$$

$$T \bullet X = \begin{bmatrix} t_1 * x^{(1)} & \ldots & t_M * z^{(1)} \\ \ldots & \ldots & \ldots \\ t_1 * x^{(K)} & \ldots & t_M * z^{(K)} \end{bmatrix}$$

Deep CTL

- Easy to extend to multiple layers

$$\min_{T_1,T_2,T_3,Z} \left\| T_3 \bullet \left(T_2 \bullet \left(T_1 \bullet X \right) \right) - Z \right\|_F^2 + \Psi\left(Z \right)\big) + \lambda \sum_{i=1}^{3} \left\{ \left\| T_i \right\|_F^2 - \log\det\left(T_i \right) \right\}$$

J. Maggu, E. Chouzenoux, G. Chierchia and A. Majumdar, “Deep Convolutional Transform Learning”, ICONIP, pp. 300-307, 2020.
J. Maggu, E. Chouzenoux, G. Chierchia and A. Majumdar, “Convolutional Transform Learning”, ICONIP, pp. 162-174, 2018.

- Work on CTL based clustering is under submission at EUSIPCO 2022!

Properties

- Easily solvable via existing optimizers like Adam.
- Guarantees unique (linear independent) of convolutional filters.
    - Unlike CNN, which ‘hopes’ to learn different filters.

THANK YOU

angshul@iiitd.ac.in