Greedy Deep Disaggregating Sparse Coding

Authors: Shikha Singh and Angshul Majumdar

Presenter: Manoj Gulati

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

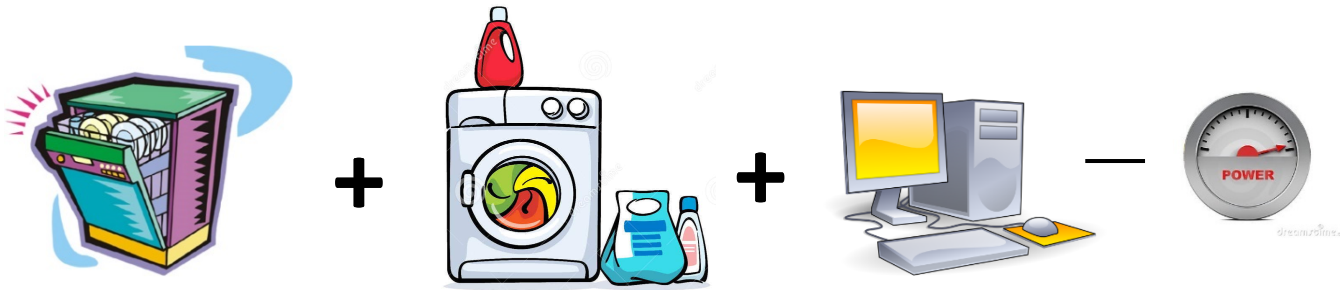
Sparse coding - Training

$$X_{dishwasher} = D_1 Z_1 \equiv \min_{D_1 Z_1} \left\| X_{dishwasher} - D_1 Z_1 \right\|_F^2 + \lambda \left\| Z_1 \right\|_1$$

$$X_{washer} = D_2 Z_2 \equiv \min_{D_2 Z_2} \left\| X_{washer} - D_2 Z_2 \right\|_F^2 + \lambda \left\| Z_2 \right\|_1$$

$$X_{desktop} = D_3 Z_3 \equiv \min_{D_3 Z_3} \left\| X_{desktop} - D_3 Z_3 \right\|_F^2 + \lambda \left\| Z_3 \right\|_1$$

Sparse Coding - Disaggregation



$$X_{dishwasher} + X_{washer} + X_{desktop} = X$$

$$D_1 Z_1 + D_2 Z_2 + D_3 Z_3 = X$$

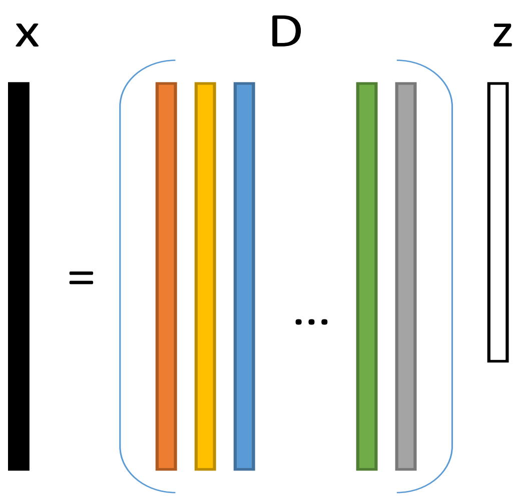
Dictionaries are already learnt in the training phase

$$\min_{Z_1, Z_2, Z_3} \left\| X - [D_1 \mid D_2 \mid D_3] \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \right\|_F^2 + \lambda \left\| \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \right\|_1$$

$$\hat{X}_{dishwasher} = D_1 Z_1 ; \hat{X}_{washer} = D_2 Z_2 ; \hat{X}_{desktop} = D_2 Z_2$$

Dictionary Learning Interpretation

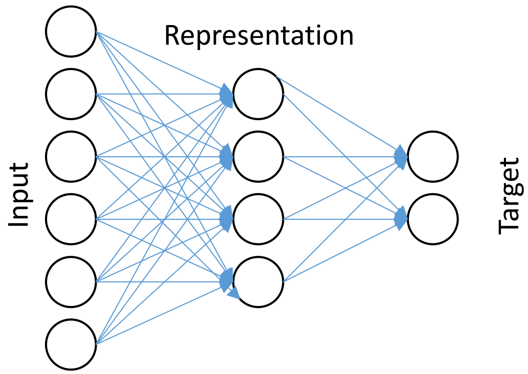
- Given a dataset X, can we learn a basis D so that the data can be represented in terms of sparse features Z?

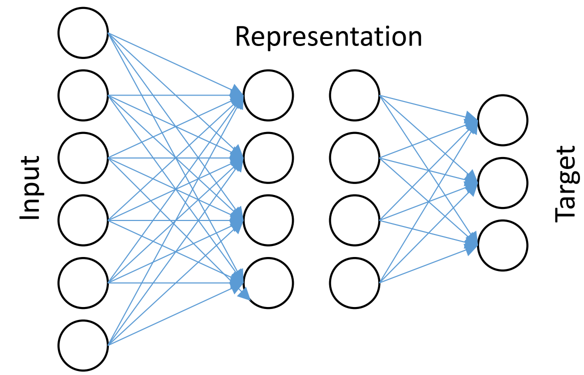


$$X = DZ$$

$$\min_{DZ} \|X - DZ\|_F^2 + \lambda \|Z\|_1$$

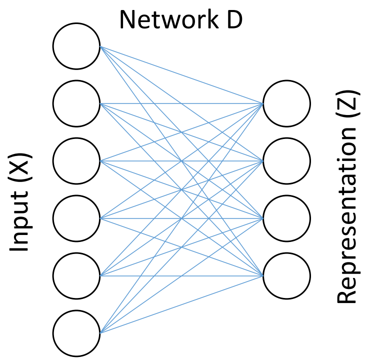
Neural Network





- This can be segregated into two parts
  - Input to representation
  - Representation to target
- The second part is trivial!
- Learning the first part – Representation Learning

RL – Restricted Boltzmann Machine

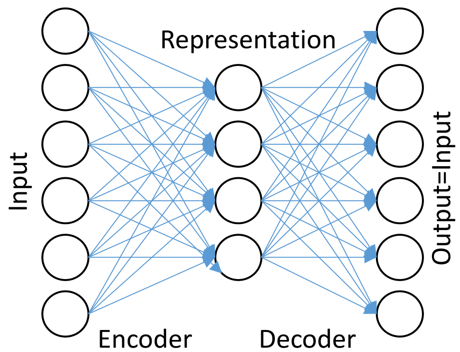


Boltzmann Function

$$p(D,Z) = e^{-Z^T DX}$$

- Maximizes similarity between the projection of the input (DX) and the representation (Z).
- After training RBM, for representation, the Targets are attached to form the neural net.

RL – Autoencoder

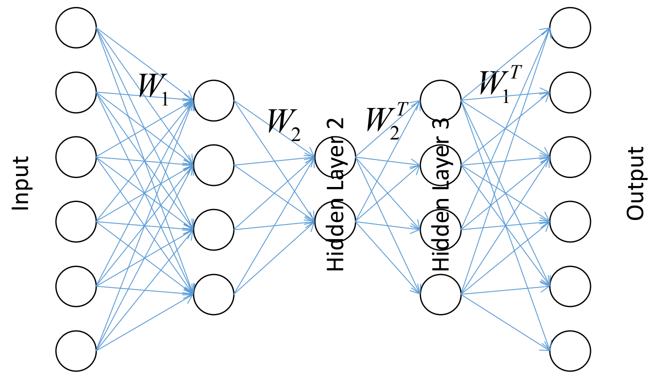


Euclidean Mismatch

$$\min_{W,W'} \left\| X - W'\phi(WX) \right\|_F^2$$

- Encodes the input to the representation and then decodes the representation to form the input / output such that the cost function is minimized.
- After training, decoder is deleted and the Targets attached to form the neural net.

Stacked Autoencoders



- To learn deeper architectures, autoencoders are nested inside each other.

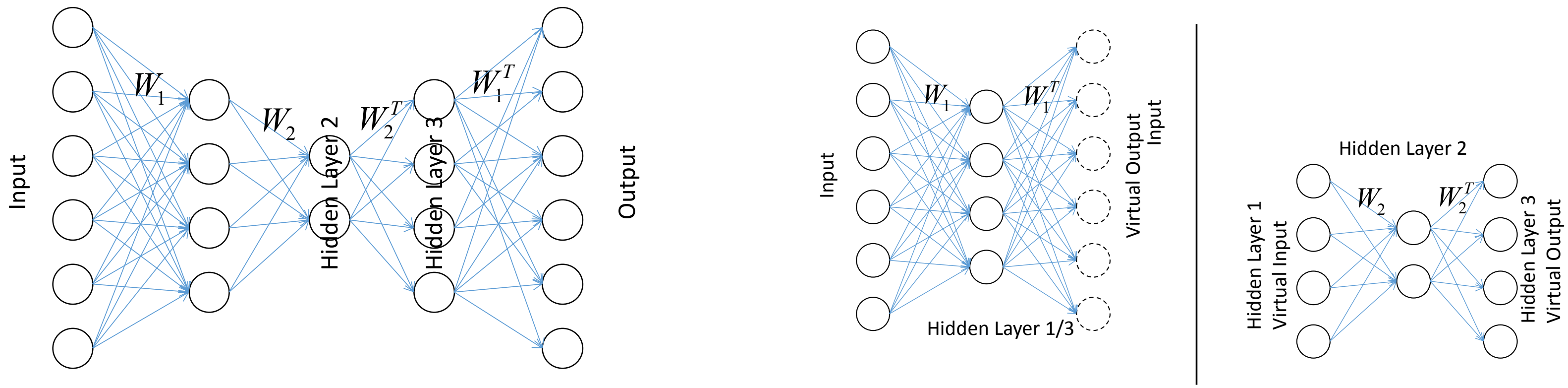
$$\underset{W_1 \ldots W_{L-1}, W'_1 \ldots W'_L}{\arg\min} \left\| X - g \circ f(X) \right\|_F^2$$

$$\text{where } g = W_1' \phi\left(W_2' \ldots W_L'\left(f(X)\right)\right)$$

$$\text{and } f = \phi\left(W_{L-1}\phi\left(W_{L-2} \ldots \phi(W_1 X)\right)\right)$$

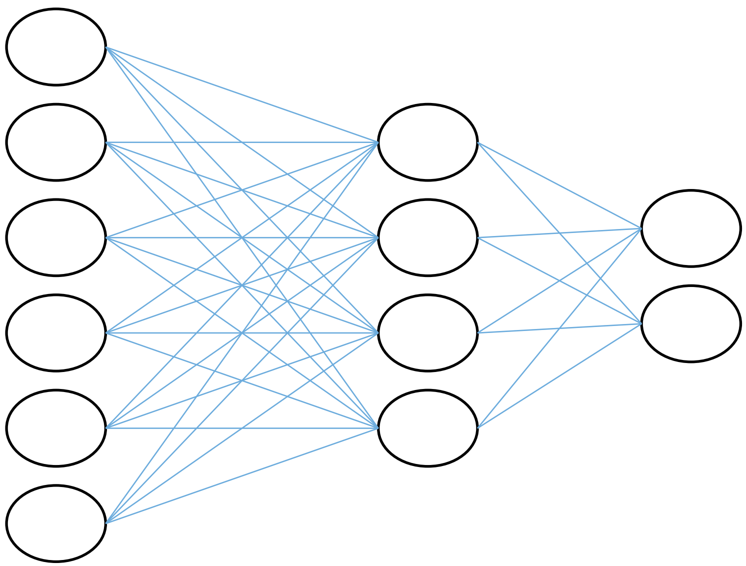
Difficult beast to optimize

Greedy Learning



- First the outermost layer is learnt.
- The features from the outermost layer now act as inputs for the nested layer.
- This continues till the deepest layer.
- Deep / bottleneck layer is used.

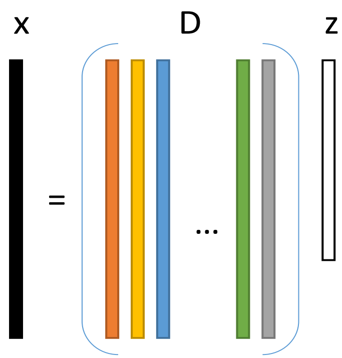
Deep Belief Network

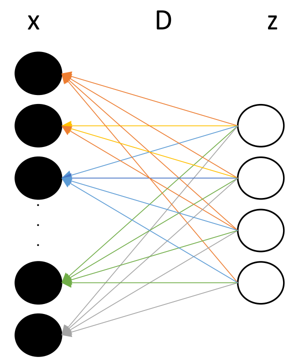


- A DBN is a cascade of RBMs.
- Can be used for feature extraction.
- Can be converted to a deep neural network with targets at the output.

G. E. Hinton, S. Osindero and Y. W. Teh, "A fast learning algorithm for deep belief nets", Neural Computation, Vol. 18, pp. 1527-1554, 2006.

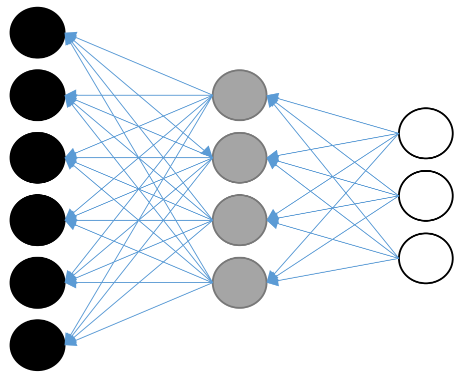
Sparse Coding Alternate Look





- The basis can be interpreted as connections between the feature / represention to the input.
- It is 'synthesis learning' since the dictionary is synthesizing the input from the features.

Deep Sparse Coding



- Layers can be appended one after the other to form deeper architecture.
- The features from the deepest layer will be used for the task.
- Features from shallower layer acts as input for deeper layer.

Formulation

- This is the exact formulation

$$\min_{D_1,D_2,\ldots,D_N,Z} \left\|X - D_1 D_2 \ldots D_N Z\right\|_F^2 + \lambda \left\|Z\right\|_1$$

- Solving this is as complex as the stacked autoencoder.
- One can use Bregman Splitting …

$$\min_{D_1,D_2,\ldots,D_N,Z,Y_1,Y_2,\ldots,Y_N} \left\|X - D_1 Y_1\right\|_F^2 + \mu_1 \left\|Y_1 - D_2 Y_2 - B_1\right\|_F^2$$

$$+ \ldots + \mu_{N-1} \left\|Y_{N-1} - D_N Z - B_{N-1}\right\|_F^2 + \lambda \left\|Z\right\|_1$$

- But needs tuning of too many hyper-parameters

Greedy Learning

Substitute $Y_1 = D_2 \ldots D_N Z$ in $\min\limits_{D_1, D_2, \ldots, D_N, Z} \| X - D_1 D_2 \ldots D_N Z \|_F^2 + \lambda \| Z \|_1$

Greedily learn: $\min\limits_{D_1, Y_1} \| X - D_1 Y_1 \|_F^2$

Then substitute $Y_2 = D_3 \ldots D_N Z$ in $Y_1 = D_2 \ldots D_N Z$

Greedily learn: $\min\limits_{D_2, Y_2} \| Y_1 - D_2 Y_2 \|_F^2$

Continue ... Till penultimate level

In the last level you have $Y_{N-1} = D_N Z$

Solve: $\min\limits_{D_N, Z} \| Y_{N-1} - D_N Z \|_F^2 + \lambda \| Z \|_1$

Doesn't introduce any extra hyper-parameter

Current Work

- Greedy learning offers plug and play options.
- One can use any type of dictionary learning in any layer.
- This work uses Kolter's formulation in first layer.
- Following layers use simple sparse coding.

- Tested on Pecan Street

Results

No. of Houses	Sparse Coding (SC)	1st Layer (DDSC)	2nd Layer (SC)	3rd Layer (SC)
72	63.13	67.25	69.34	69.45

*Disaggregating Discriminative Sparse Coding

Conclusion

- New framework for representation learning.
- NILM – serves as a nice application.

- For NILM, deep sparse coding will be coupled with our associated work on robust learning.

THANK YOU