Dropout Training of Matrix Factorization and Autoencoder for Link Prediction in Sparse Graphs

Shuangfei Zhai, Binghamton University
Zhongfei (Mark) Zhang, Binghamton University

May 1, 2015

Background

- Link prediction: Fill possible 1s into (sparse) symmetric adjacency matrix $A \in \{0,1\}^{N \times N}$ of a graph (we assume it's undirected in our work)
- Examples:
  - Recommending friends in Facebook, Twitter etc.
  - Suggesting collaboration in coauthor networks
  - Recommendation systems in general
- Matrix Factorization (MF):

$$min \sum_{i,j} L(A_{i,j},\ g(W_2^j W_{1,i} + b_{1,i} + b_{2,j})) \tag{1}$$

  - W_2^j is the jth row of W_2, $W_{1,i}$ is the ith column of W_1
  - L and g are loss function and link function respectively.
  - We take $g(x) = (1 + \exp(-x))^{-1}$, L as cross entropy in this work.

Background Cont'd

- Autoencoder (AE): learning representations by reconstructing input $x \in R^N$

$$\begin{aligned} h &= f(W_1 x + b_1) \\ \tilde{x} &= g(W_2 h + b_2) \end{aligned} \tag{2}$$

with $W_1 \in R^{K \times N}, b_1 \in R^K, W_2 \in R^{N \times K}, b_2 \in R^N$. The parameters are learned by solving the following optimization problem:

$$\min \sum_i L(x_i, \tilde{x}_i; W_1, b_1, W_2, b_2) \tag{3}$$

- Applying to link prediction, we take $x_i = A_i$, this is "bag of neighbors" representation analogous to bag of words.
- $h \in R^K$ is the learned representation for input x, should contain community information useful for link prediction.

Relating MF to AE

- Define $\delta_i \in R^N$ as the one hot encoding of node i, rewrite MF (omitting the biases) as:

$$\min \sum_i L(A_i, g(W_2 h_i)) \quad s.t.\ h_i = W_1 \delta_i \tag{4}$$

- And AE as:

$$\min \sum_i L(A_i, g(W_2 h_i)) \quad s.t.\ h_i = f(W_1 A_i) \tag{5}$$

- Both MF and AE are reconstructing A with (almost) the same architecture: learn a representation h_i for node i.
- MF only looks at the identity of target node (δ_i); whereas AE only looks at the neighboring nodes (A_i)
- They form two complimentary and sufficient views

Model: Joint Learning of MF and AE

$$\min \sum_i L(A_i, g(W_2 h_{1,i} + b_2)) + \rho \sum_i L(A_i, g(W_2 h_{2,i} + b_4)) \quad (6)$$
$$s.t.\ h_{1,i} = f(W_1 A_i + b_1), h_{2,i} = f(W_1 \delta_i + b_3)$$

- Idea: the two views should agree with each other
- Modify MF by introducing the same nonlinear activation function f as used in AE. In our work, we use ReLu: $f(x) = \max(0, x)$
- Share the encoding the decoding matrix W_1 and W_2 between AE and MF
- Since they are both trained to reconstruct A, $W1$ and W_2 are forced to produce consistent representations from the two views.

Model: Joint Learning of MF and AE

$$\min \sum_i L(A_i, g(W_2 h_{1,i} + b_2)) + \rho \sum_i L(A_i, g(W_2 h_{2,i} + b_4)) \quad (7)$$

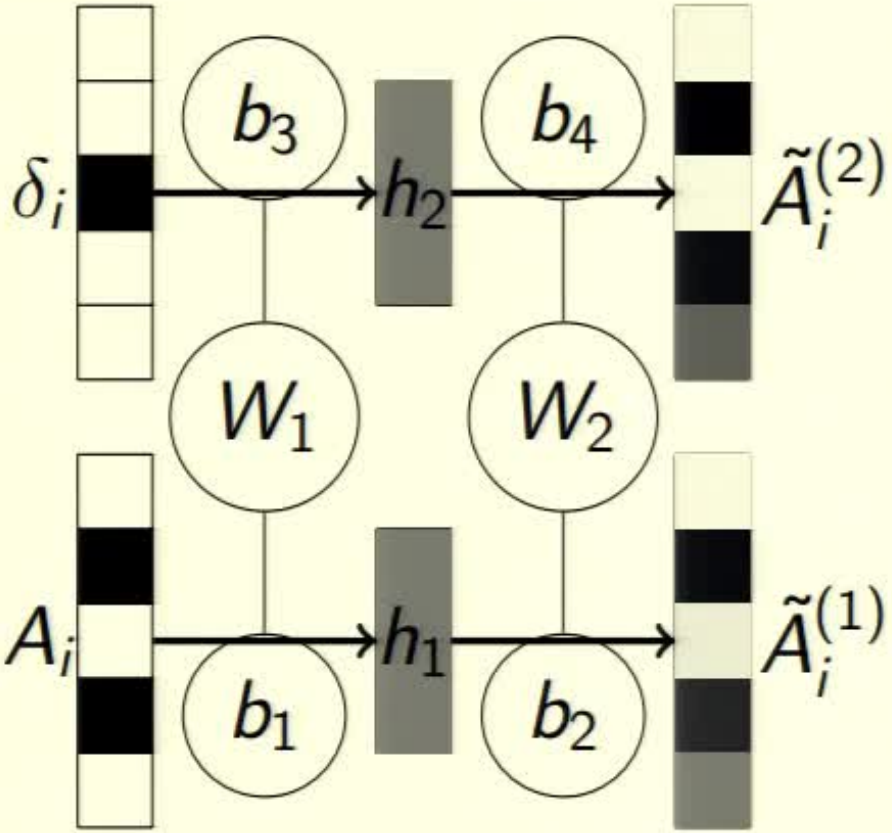
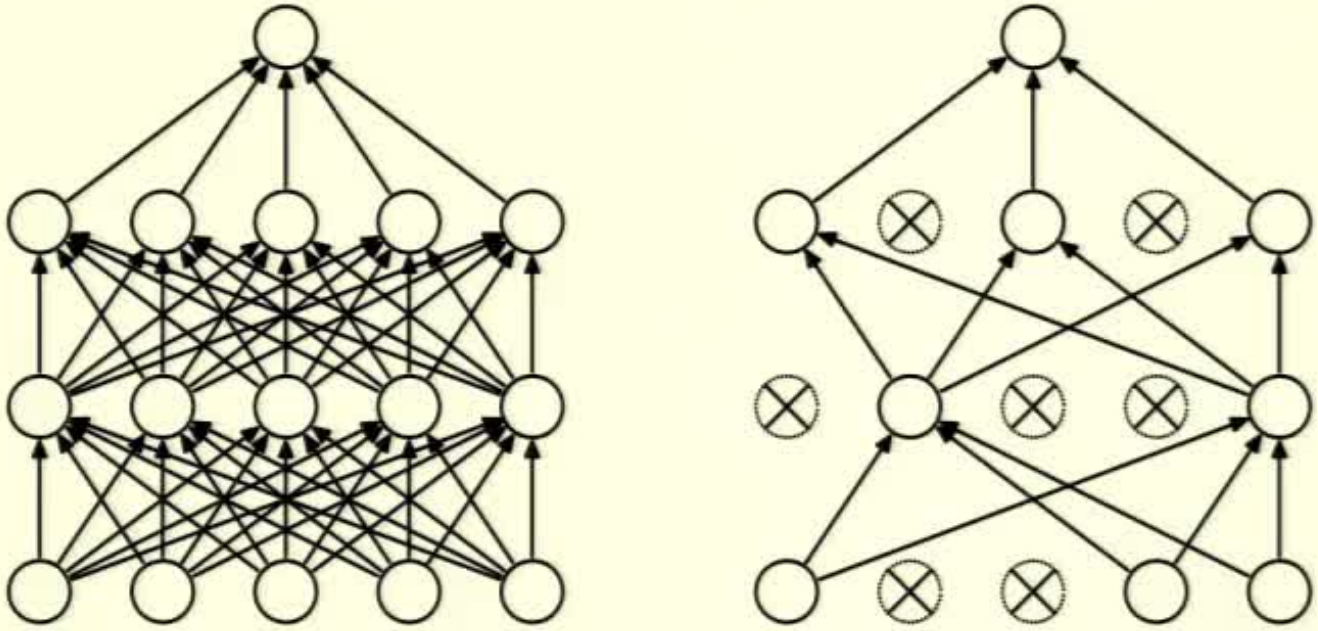


Figure 1: The architecture of the MF+AE model, prediction is achieved by averaging predictions from the two views

Training with Dropout (Regularized SGD)

- Originally designed to regularize deep neural nets, now is the standard tool for deep learning.
- Works by randomly masking some neurons (input or latent variables) to zero for each sample seen during the regular Stochastic Gradient Descent.
- We show that this very technique can also help models as shallow as MF and (single layer) AE



(a) Standard Neural Net

(b) After applying dropout.

Second Order Approximation of Dropout on MF

$$
\begin{aligned}
O &= \sum_{i,j} \mathrm{E}_{\xi_i}\{L(A_{i,j}, g(W_2^j(\xi_i \odot W_{1,i}))\} \\
&\approx \sum_i L(A_i, g(\frac{1}{2} W_2 W_{1,i})) + \frac{1}{8} \underbrace{(\sum_j (W_2^j)^2 g_{i,j}(1 - g_{i,j}))}_{\lambda^i} (W_{1,i})^2
\end{aligned}
\tag{8}
$$

- $\xi_i \in \{0,1\}^K$ is the random binary mask with each element an iid Bernoulli draw with probability 0.5
- $(W_2^j)^2$ and $(W_{1,i})^2$ are the element-wise square of the row and column vectors, respectively
- $g_{i,j}$ is short for $g(\frac{1}{2} W_2^j W_{1,i})$

Dropout on MF Cont'd

$$O = \sum_{i,j} \mathrm{E}_{\xi_i}\{L(A_{i,j}, g(W_2^j(\xi_i \odot W_{1,i}))\}$$

$$\approx \sum_i L(A_i, g(\frac{1}{2} W_2 W_{1,i})) + \frac{1}{8} \underbrace{(\sum_j (W_2^j)^2 g_{i,j}(1 - g_{i,j}))}_{\lambda^i} (W_{1,i})^2 \quad (9)$$

- The first term is equivalent to the objective function of MF, except the activation is down scaled by a half
- The second term is a weighted ℓ_2 regularization of columns of W_1
- The weight λ^i is adaptive, note $g_{i,j}$ is a function of W_1, W_2; it encourages confident prediction and small weights.
- The roles of W_1, W_2 are symmetrical, the same interpretation can be taken on the rows of W_2

Second Order Approximation of Dropout on (linear) AE

- Dropout hidden units: similar to MF, replacing W_1 with $W_1 A_i$. penalizing the linear combination of columns of W_1
- Dropout inputs (also known as Denoising Autoencoder): denote $W = W_2 W_1$

$$
\begin{aligned}
O &= \sum_i \mathrm{E}_{\xi_i}\{L(A_i, g(W(\xi_i \odot A_i)))\} \\
&\approx \sum_i L(A_i, g(\frac{1}{2} W A_i)) + \frac{1}{8} \sum_j (W^j)^2 \underbrace{\sum_i g_{i,j}(1 - g_{i,j})(A_i)^2}_{\lambda_j}
\end{aligned} \tag{10}
$$

- With $W^j = W_2^j W_1$, feature dropout performs adaptive weighted ℓ_2 regularization on the linear combination of rows of W_1
- Recall that dropout of hidden units penalize on the linear combinations of columns of W_1

Experiments

- For each of the six graphs, randomly select 10% edges for training, and rest for testing. This yields six sparse graphs.
- Train all the evaluated models to convergence, we are interested in their generalization abilities.

dataset	N	E	D
DBLP	2,958	64,674	21.9
Facebook	2,277	148,546	65.2
Youtube	1,955	102,950	52.6
Twitter	2,477	107,895	43.6
Gplus	2,129	148,306	69.7
LiveJournal	3,006	123,236	41.0

Table 1: Statistics of the datasets where N: number of nodes, E: number of links, D: average degree.

Results

Model	Facebook	Twitter	Youtube	Gplus	DBLP	LiveJournal	Average
MF+AE	**0.58057**	**0.46693**	**0.33132**	**0.41277**	**0.32462**	**0.29027**	**0.4011**
AEd	0.54643	0.44229	0.31769	0.40085	0.29942	0.28659	0.3822
AE2	0.37748	0.2773	0.087839	0.17973	0.28308	0.1722	0.2296
MFd	0.46716	0.4041	0.23636	0.28956	0.29599	0.23958	0.3221
MF2	0.45216	0.39823	0.13842	0.24594	0.30735	0.21651	0.2931
MDM	0.54255	0.41304	0.23548	0.3149	0.30286	0.25415	0.3438
RW	0.53143	0.40647	0.15805	0.21685	0.27757	0.20524	0.2993
AA	0.47439	0.34576	0.13647	0.17523	0.23712	0.18247	0.2586

Table 2: Performance evaluated by *Prec*@10

- Our proposed model trained with dropout achieved best result.
- AE and MF trained with dropout (AEd and MFd) work much better than their ℓ_2 regularized counterpart (AE2 and MF2)

Resutls Cont'd

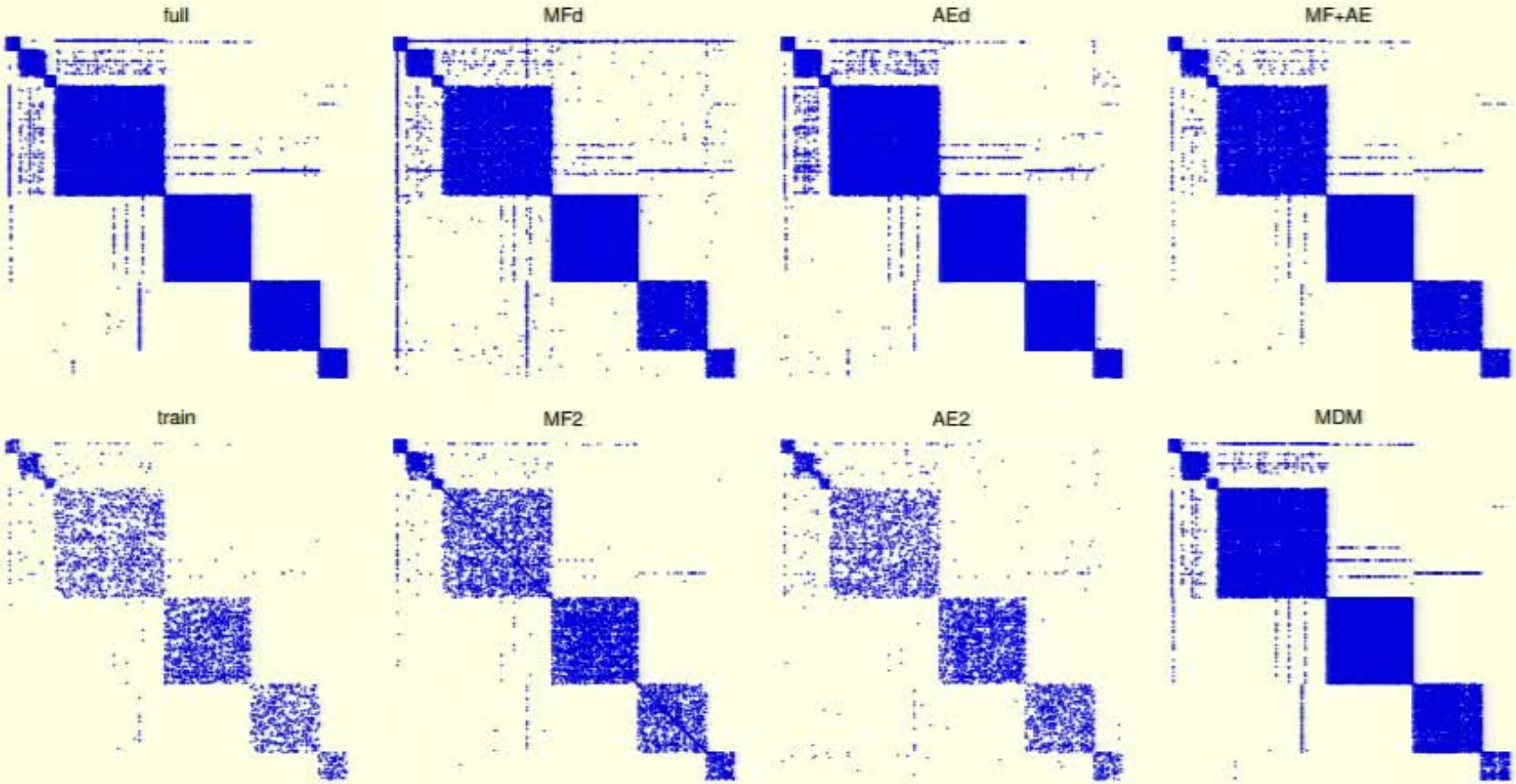


Figure 2: Visualization of the Facebook dataset. From top left to bottom right: the full graph, the predictions of MFd, AEd, MF+AE, respectively, the training graph, the prediction of MF2, AE2, MDM, respectively.

Conclusion and Future Work

Conclusion:

- We have investigated the usage of AE to graph modeling, and relate AE and MF by jointly training them together with shared parameters.
- We have applied dropout to training both AE and MF, and show that dropout act as an adaptive regularization
- We have conducted experiment on real world datasets and successfully proved our hypothesis.

Discussion:

- Bayesian approach is another popular choice for regularization, it would be interesting to investigate the relationship of Bayesian methods with dropout, and the possibility of enhancing both by combining them.

Thank You!

Second Order Approximation of Dropout on (linear) AE

- Dropout hidden units: similar to MF, replacing W_1 with $W_1 A_i$. penalizing the linear combination of columns of W_1
- Dropout inputs (also known as Denoising Autoencoder): denote $W = W_2 W_1$

$$
\begin{aligned}
O &= \sum_i \mathrm{E}_{\xi_i}\{L(A_i, g(W(\xi_i \odot A_i)))\} \\
&\approx \sum_i L(A_i, g(\frac{1}{2} W A_i)) + \frac{1}{8} \sum_j (W^j)^2 \underbrace{\sum_i g_{i,j}(1 - g_{i,j})(A_i)^2}_{\lambda_j}
\end{aligned} \tag{10}
$$

- With $W^j = W_2^j W_1$, feature dropout performs adaptive weighted ℓ_2 regularization on the linear combination of rows of W_1
- Recall that dropout of hidden units penalize on the linear combinations of columns of W_1

Dropout on MF Cont'd

$$
\begin{aligned}
O &= \sum_{i,j} \mathrm{E}_{\xi_i}\{L(A_{i,j}, g(W_2^j(\xi_i \odot W_{1,i}))\} \\
&\approx \sum_i L(A_i, g(\frac{1}{2} W_2 W_{1,i})) + \frac{1}{8} \underbrace{(\sum_j (W_2^j)^2 g_{i,j}(1 - g_{i,j}))}_{\lambda^i} (W_{1,i})^2
\end{aligned}
\tag{9}
$$

- The first term is equivalent to the objective function of MF, except the activation is down scaled by a half
- The second term is a weighted ℓ_2 regularization of columns of W_1
- The weight λ^i is adaptive, note $g_{i,j}$ is a function of W_1, W_2; it encourages confident prediction and small weights.
- The roles of W_1, W_2 are symmetrical, the same interpretation can be taken on the rows of W_2

Second Order Approximation of Dropout on MF

$$
\begin{aligned}
O &= \sum_{i,j} \mathrm{E}_{\xi_i}\{L(A_{i,j}, g(W_2^j(\xi_i \odot W_{1,i}))\} \\
&\approx \sum_i L(A_i, g(\frac{1}{2} W_2 W_{1,i})) + \frac{1}{8} \underbrace{(\sum_j (W_2^j)^2 g_{i,j}(1 - g_{i,j}))}_{\lambda^i} (W_{1,i})^2
\end{aligned}
\tag{8}
$$

- $\xi_i \in \{0,1\}^K$ is the random binary mask with each element an iid Bernoulli draw with probability 0.5
- $(W_2^j)^2$ and $(W_{1,i})^2$ are the element-wise square of the row and column vectors, respectively
- $g_{i,j}$ is short for $g(\frac{1}{2} W_2^j W_{1,i})$

Model: Joint Learning of MF and AE

$$\min \sum_i L(A_i, g(W_2 h_{1,i} + b_2)) + \rho \sum_i L(A_i, g(W_2 h_{2,i} + b_4)) \quad (7)$$

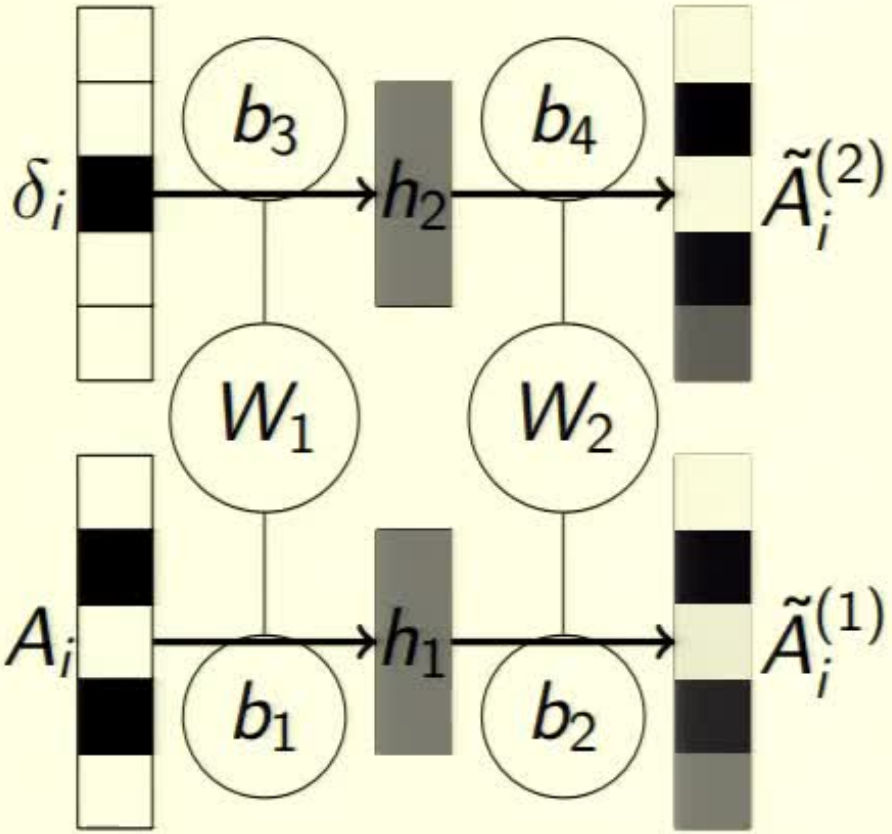


Figure 1: The architecture of the MF+AE model, prediction is achieved by averaging predictions from the two views

Model: Joint Learning of MF and AE

$$\min \sum_i L(A_i, g(W_2 h_{1,i} + b_2)) + \rho \sum_i L(A_i, g(W_2 h_{2,i} + b_4)) \quad (6)$$
$$s.t.\ h_{1,i} = f(W_1 A_i + b_1), h_{2,i} = f(W_1 \delta_i + b_3)$$

- Idea: the two views should agree with each other
- Modify MF by introducing the same nonlinear activation function f as used in AE. In our work, we use ReLu: $f(x) = \max(0, x)$
- Share the encoding the decoding matrix W_1 and W_2 between AE and MF
- Since they are both trained to reconstruct A, $W1$ and W_2 are forced to produce consistent representations from the two views.