Deep Learning based Recommender System: A Survey and New Perspectives (Autoencoder based Recommendation System)

Shuai zhang, Lina yao and Aixin sun

Presented by Boyoung Kim

November 22, 2017

Contents

Introduction : Auto-encoder

- Unsupervised learning version of Neural Network.
- AE can be used for dimensionality reduction of high-dimensional data.
- AE generate a hidden representation from an input, and reconstruct the output as the input from the hidden representation.
- Setting the target values to be equal to the input : $h_{W,b}(x) \approx x(\hat{x} \approx x)$.

Introduction : Auto-encoder

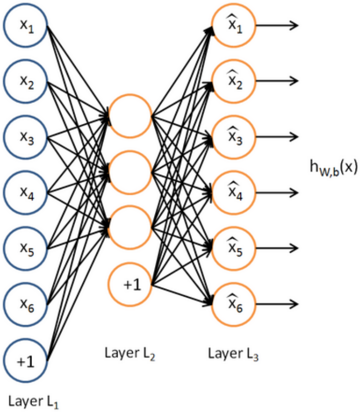


Figure: Architecture of autoencoder

- $h_{W,b}(x) = f(W_2 \cdot g(W_1 x + b_1) + b_2)$
- Stacked Auto-encoder : Auto-encoder with more than 1 hidden layer

Contents

AutoRec: Autoencoders Meet Collaborative Filtering

- Suppose we have M users, N items.
- We use different Autoencoder for each user or each item.
- Each Autoencoder only has input units for the users who rate that item.
- Every Autoencoder has the same number of hidden units.
- Each autoencoder only has a single training case, but all of the corresponding weights and biases are tied together.

Item-based AutoRec model

- The input, output units model ratings as real values.
- Let $r^{(i)}$ denote partial observed vector for item i.

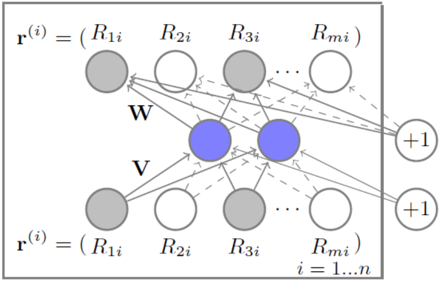


Figure: Item-based AutoRec model

Item-based AutoRec model

- Suppose that the item is rated by n users.
- Then the hidden and output units are :

$$h_j = g(\sum_{i=1}^{n} V_{ij} r_j^{(i)} + a_j) \quad \text{and}$$

$$\hat{r}_j^{(i)} = f(\sum_{k} W_{ik} h_k + b_j)$$

where $f(\cdot)$ and $g(\cdot)$ are activation functions.

ν Using identity $f(\cdot)$ and sigmoid $g(\cdot)$ functions has good performance.

AutoRec : Learning

- Gradient descent method using "Backpropagation algorithm".
- The objective function for a single training example :

$$\min_{W,V,a,b} \frac{1}{N} \sum_{i=1}^{N} \| r^{(i)} - \hat{r}^{(i)} \|_{\mathcal{O}}^2 + \lambda \cdot Regularizer$$

where $\| \cdot \|_{\mathcal{O}}^2$ means that we only consider the contribution of observed ratings.

ν I-AutoRec performs better than U-AutoRec, since the average number of ratings per item is much more that those per user.

ν Stacking more layers improves the performance.

Contents

Collaborative Filtering Neural network(CFN)

- Extension of AutoRec
- Denoising AutoEncoder
  - In this paper, masking noise is imposed.
  - $\tilde{r}^{(i)}$ denotes the corrupted input of $r^{(i)}$

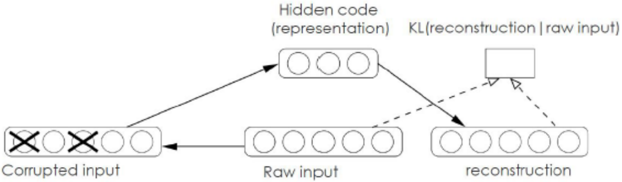


Figure: A denoising AE

Collaborative Filtering Neural network(CFN)

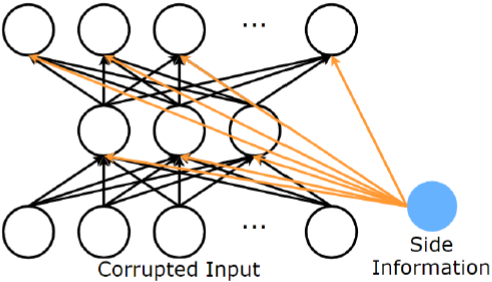
- DAE loss

$$\mathcal{L} = \alpha \left(\sum_{(i,j) \in I(\mathcal{O}) \cap I(\mathcal{C})} [h(\tilde{r}^{(i)})_j - r_j^{(i)}]^2 \right) + \beta \left(\sum_{(i,j) \in I(\mathcal{O}) \setminus I(\mathcal{C})} [h(\tilde{r}^{(i)})_j - r_j^{(i)}]^2 \right) + \lambda \cdot \mathit{Regularization}$$

  - $I(\mathcal{O})$ and $I(\mathcal{C})$ are the indices of observed and corrupted elements
  - α and β are two hyper parameters which balance the reconstruction and prediction error

Collaborative Filtering Neural network(CFN)

- Further extension of CFN also incorporates side information in every layer.
- It can be stacked.



$$h(\{\tilde{r}^{(i)}, s_i\}) = f(W_2 \cdot \{g(W_1 \cdot \{\tilde{r}^{(i)}, s_i\} + b_1), s_i\} + b_2)$$

where s_i is side information of item i.

Contents

Collaborative Denoising Auto-Encoder(CDAE)

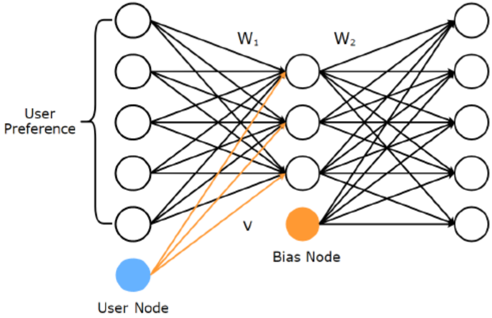
- Implicit feedback dataset
- If the user likes the item, the entry value is 1, otherwise 0.
- Gaussian noise or Mask-out/drop-out noise is used.
- Mask-out/drop-out corruption :

$$P(\tilde{r}_d^{(u)} = \delta r_d^{(u)}) = 1 - q, P(\tilde{r}_d^{(u)} = 0) = q$$

To make the corruption unbiased, one sets $\delta = \frac{1}{1-q}$.

Collaborative Denoising Auto-Encoder(CDAE)

- $\mathbf{V}_u \in \mathbb{R}^k$: weight vector for the user input node where k is the number of hidden units. Note that $\mathbf{V}_u$ is a user-specific vector.



$$h(\tilde{r}^{(u)}) = f(W_2 \cdot g(W_1 \cdot \tilde{r}^{(u)} + V_u + b_1) + b_2)$$

Collaborative Denoising Auto-Encoder(CDAE)

- Parameters are learned by

$$\underset{W_1, W_2, V, b_1, b_2}{\operatorname{argmin}} \frac{1}{M} \sum_{u=1}^{M} \mathbb{E}_{p(\tilde{r}^{(u)} | r^{(u)})} [l(\tilde{r}^{(u)}, h(\tilde{r}^{(u)}))] + \lambda \cdot Regularization$$

The loss function $l(\cdot)$ can be square loss or logistic loss.

- Negative sampling : Sampling small subset from negative set and user's preferences of items are used for computing gradients reduces the time complexity.

Contents

Collaborative Deep Learning(CDL)

- Hierarchical Bayesian model which integrates SDAE and MF
- Modeling the noise to get robust result.
- Implicit feedback dataset

- Notation
  - X_c : $N \times S$ item content matrix (clean output)
  - $X_{c,j*}$: item j's content. j-th row of X_c
  - X_0 : corrupted input
  - X_l : $N \times D_l$ the output of layer l of the SDAE.
  - L : number of layers

Generative process of CDL

1. For each layer l of the SDAE network,
   (a) $W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{D_l})$
   (b) $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{D_l})$
   (c) For each row j of X_l, $X_{l,j*} \sim \mathcal{N}(\sigma(X_{l-1,j*} \cdot W_l + b_l), \lambda_s^{-1} I_{D_l})$
2. For each item j,
   (a) Draw a clean input $X_{c,j*} \sim \mathcal{N}(X_{L,j*}, \lambda_n^{-1} I_s)$
   (b) Draw a latent item offset vector $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$, and set the latent item vector:

$$v_j = X^T_{\frac{L}{2},j*} + \epsilon_j$$

3. Draw a latent user vector for each user i: $u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$.
4. Draw a rating R_{ij} for each user-item pair (i, j):

$$R_{ij} \sim \mathcal{N}(u_i^T v_j, C_{ij}^{-1})$$

where C_{ij} is a confidence parameter $C_{ij} = a$ if $R_{ij} = 1, C_{ij} = b$ o.w. $(a > b > 0)$

Collaborative Deep Learning(CDL)

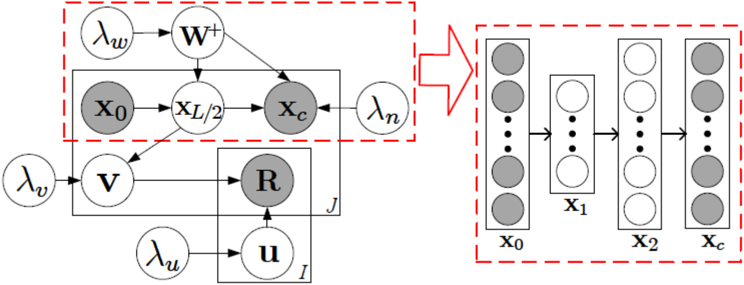


Figure: Graphical model of CDL when λ_s approaches positive infinity

Collaborative Deep Learning(CDL)

- Maximizing a posterior probability is equivalent to maximizing the joint log-likelihood of parameters.

$$
\begin{aligned}
\mathcal{L} = &- \frac{\lambda_u}{2} \sum_i \|u_i\|_2^2 - \frac{\lambda_w}{2} \sum_l (\|W_l\|_F^2 + \|b_l\|_2^2) \\
&- \frac{\lambda_v}{2} \sum_j \|v_j - X_{\frac{L}{2},j*}^T\|_2^2 - \frac{\lambda_n}{2} \sum_j \|X_{L,j*} - X_{c,j*}\|_2^2 \\
&- \frac{\lambda_s}{2} \sum_{l,j} \|\sigma(X_{l-1,j*} W_l + b_l) - X_{l,j*}\|_2^2 \\
&- \sum_{i,j} \frac{C_{ij}}{2} (R_{ij} - u_i^T v_j)^2
\end{aligned}
$$

Contents

Deep Collaborative Filtering Framework(DCF)

- DCF unifies the deep learning models with MF which makes use of both rating matrix and side information.
- Let X and Y denote side information of user and item.
- The objective function of mDA-CF is

$$\operatorname*{argmin}_{U,V,W_1,W_2} l(R,U,V) + \beta(\|U\|_F^2 + \|V\|_F^2) + \gamma\mathcal{L}(X,U) + \delta\mathcal{L}(Y,V)$$

where β, γ, δ are the trade-off parameters.

- In particular, the latent factors are extracted from the hidden layer of deep networks.

Deep Collaborative Filtering Framework(DCF)