INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY, DELHI

Extreme Learning Machine for Collaborative Filtering

Sagar Verma
MT15056

November 18, 2016

1 Cold Start Problem

Cold start problem in collaborative filtering is predicting information of a new user when that user has not rated any item. Cold start problem is usually solved by using metadata information of user and item.

2 Extreme Learning Machine

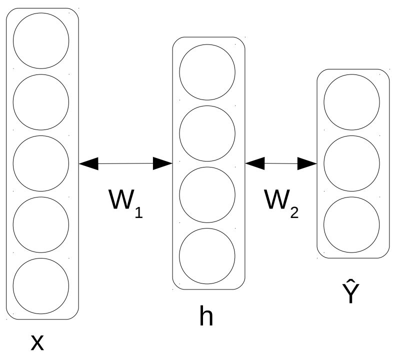

Figure 2.1: A basic ELM with one hidden layer.

Extreme learning machine(ELM) are a type of feedforward neural network used to solve classification or regression problems. The ELM network contains a single layer of hidden nodes. Weights connecting inputs to hidden nodes are randomly assigned and never updated. These weights between hidden nodes and outputs are learned in a single step. ELMs have been shown to perform good on certain class of problems were a linear model has to be learned and they achieve accuracy similar to models learned with backpropagation and at thousands times faster speed. The simplest ELM training algorithm takes the form

$$\hat{Y} = W_2 \sigma(W_1 x) \tag{2.1}$$

where w_1 is the matrix of input-to-hidden layer weights, σ is activation function and W_2 is the matrix of hidden-to-output-layer weights.

Types of activation function

1. ***Linear*** : $f(x) = x$
2. ***Sigmoid*** : $f(x) = \frac{1}{1+e^{-x}}$
3. ***Tanh*** : $f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$
4. ***Radial basis function*** : $f(x) = \sum_{i=1}^{N} a_i \rho(||x - c_i||)$ where N is the number of neurons in the hidden layer, c_i is the center vector of neuron i and a_i is the weight of neuron i in the linear output neuron. $\rho(||x - c_i||)$ is norm function given by,
   a) ***rbf_l1*** : $\rho(||x - c_i||) = exp[-\beta\, ||x - c_i||]$
   b) ***rbf_l2*** : $\rho(||x - c_i||) = exp[-\beta\, ||x - c_i||^2]$
   c) ***rbf_linf*** : $\rho(||x - c_i||) = exp[-\beta\, min(||x - c_i||)]$

3 ELM to solve cold start problem in collaborative filtering

To use ELM to solve cold start problem we need to treat it as classification problem. In this cas user metadata and item metadata form features for the ELM and label is the rating itself. For any new user we find all item ratings by giving user and item metadata as input to trained model. Each column of metadata is encoded as one-hot vector and similarly label is encoded as one-hot vector. For user cold start problem we have some item ratings from some of the users, ELM is trained on those values and prediction is done for completely new users. For item cold start problem we have some item ratings from some of the users, ELM is trained on those values and prediction is done for completely new items.

Different activation functions are tried to find best perceptron activation function. Different number of hidden nodes between 5 and 50 are used with stride of 5. Average result for all values is reported in the table 3.1.

Function	User cold start			Item cold start		
	RMSE	**NMAE**	**#Nodes**	**RMSE**	**NMAE**	**#Nodes**
linear	1.2190	0.2234	10	1.2024	0.2216	45
sigmoid	1.2054	0.2221	30	1.2032	0.2218	45
tanh	1.2048	0.2223	50	1.2068	0.2223	50
rbf_l1	1.2145	0.2227	20	1.2176	0.2233	45
rbf_l2	1.2016	0.2223	50	1.2028	0.2218	50
rbf_linf	1.2189	0.2223	35	1.1977	0.2208	45

Table 3.1: Results for user and item cold start problem. Different types of perceptron activation function have been used. Results are reported for best number of nodes in the hidden layer.

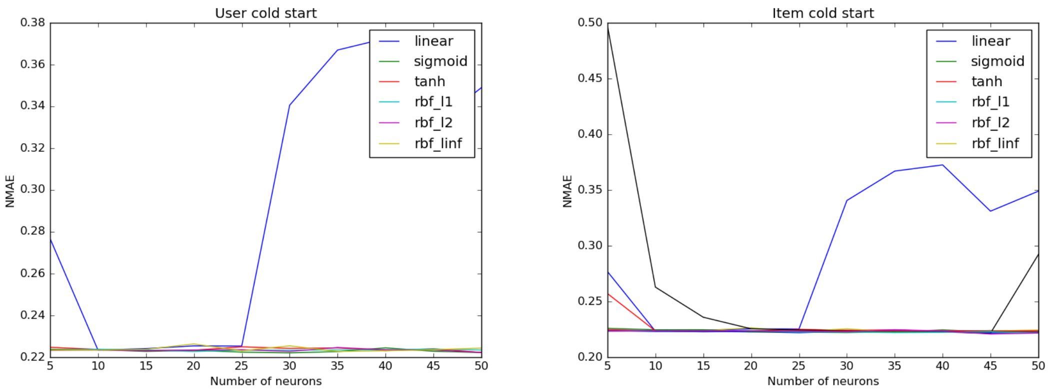

Figure 3.1: Left figure shows how different perceptron activation function perform with number of hidden nodes. Between 10-25 number of nodes every type of activation function is able to learn and give best performance for 'user cold start' problem. For 'item cold start' problem number of nodes between 20-25 gives best performance for all types of activation functions. Its evident that linear activation function performs very poorly and 'radial basis functions' perform best in both cases.