WEEK 2

Objective : To develop a Graph Neural Network for Social Recommendation

Topic : Learn Neural Networks & Use Tensorflow for Implementing MF based Recommender System.

Neural Networks Basics –

A neuron of the human brain has the capability of taking messages as input through dendrites, doing some computations and producing output through axons. There are millions of neurons in the human brain which together constitute the neural network.

Artificial Neural Network was developed as simulating neurons or networks of neurons. In the computer world, we use a simple model of human brain neurons to implement a neuron as a logistic unit for an Artificial Neural Network. It was developed for computers to mimic the human brain. Just like the human brain does most of the work unconsciously, a computer is trained to function like a human brain using artificial neural networks, where the neuron takes input in the input layer, does some computations (in the hidden layer) and accordingly produces output through the output layer. Neural Networks are an effective study-art technique for modern day machine learning applications.

Machine learning models are mostly used to construct a model; a hypothesis that is used to estimate Y based on X. The hypothesis is used for mapping inputs to outputs.

General representation of hypothesis –

$$h_\theta (x) = \theta_0 + \theta_1(x)$$

{where θ value are the parameters }

Cost Function is the measure of accuracy of a machine learning model, i.e. how wrong the model is in terms of its ability to estimate the relationship between X and Y. There are different functions which are often used to measure the difference between the estimator (the dataset) and the estimated value (the prediction), such as Mean Squared Error (MSE) etc.

A machine learning model needs to find out the correct parameters (weights and biases) for the model to be more accurate and Backward Propagation is one such algorithm to help reduce error.

Backward Propagation (or Back Propagation) Algorithm is a supervised learning algorithm, for training Multi-layer Perceptrons (Artificial Neural Network). Backward Propagation algorithm is used to train our model in such a way so as to reduce the error. The Back propagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent. The weights that minimize the error function are considered to be a solution to the learning problem.

Methods Used :

Matric Factorization is used, which is implemented using Tensorflow. Matrix factorization is a specific type of collaborative filtering method that in general sense tries to approximate sub-matrices that are equivalent to a larger matrix. Matrix factorization grew in popularity because of the sparse and expanse of data being collected by companies.

Dataset :

The dataset rating.csv includes the rating information of movies. It includes five columns which are :

- userid
- productid
- categoryid
- rating
- helpfulness

  The figures provide information about a user rating a specific product of a particular category and the helpfulness of the rating to recommend further items to the user. The model is trained with the given stats of userid, productid and rating so as to build a proper recommender system.

Packages and Libraries :

There are a lot of commands used from different libraries as well as from Tensorflow explicitly with the help from petamind link code.

- Different libraries used in the code are –
  1. Pandas
  2. Tensorflow

3. Numpy
4. Sklearn
5. Keras

- Different layers of keras used –

  Embedding – It is the first hidden layer of a network and is used for neural network on text data. It is used when we want to create the embeddings to embed higher dimensional data into lower dimensional vector space.

  Flatten – It is used to flatten the input, without affecting the batch size. Flattening a tensor means to remove all of the dimensions except for one. The flatten operation reshapes the tensor to the shape which is equal to the number of elements contained in tensor.

  Dot – The dot layer is the layer that computes a dot product between samples in two tensors.

  Dense – It is one of the most commonly and frequently used layers of Keras. Dense layer is the regular deeply connected neural network layer.

- Optimizers used in the code while trial and error –

1. SGD – SGD is an iterative method for optimizing an objective function with suitable smoothness properties. It picks single data from the dataset randomly per iteration so as to reduce the computations enormously.
2. RMSprop – Root Mean Square Propagation is a gradient based optimization technique designed for training artificial neural networks.
3. Adagrad – Adagrad is effectively SGD with a per-node learning rate scheduler built into the algorithm thus improving SGD by giving weights historically accurate leaning rates.
4. Adadelta – Adadelta optimization is a stochastic gradient descent method that is based on adaptive learning rate per dimension. It is basically a robust extension of Adagrad.

5. Adam – Adaptive Moment Estimation is a combination of Adagrad and RMSprop, which is used to calculate the individual adaptive learning rate for each parameter from estimates of first and second moments of the gradients.
6. AdaMax – It a variant of Adam based on the infinity room. AdaMax provides an important advanyage of being much less sensitive to the choice of the hyper-parameters.
7. Nadam – Nesterov – accelerated Adaptive Moment Estimation is a combination of NAG and Adam. It is employed for noisy or high curvature gradients. The learning process is accelerated by summing up the exponential decay of the moving averages for the previous and current gradient.
8. Ftrl – Follow the Regularized Leader optimizer is a per-coordinate learning rate system.

- A Metric function is used to judge the performance of a model. Functions mse() and mae() are Regression Metrics which used for loss determination or to calculate accuracy of the model.

Findings :

Latent Factors=20, Epochs=10

Optimizer	**Loss at start**	**Loss at end**	**Mae at start**	**Mae at end**	**Time for each epoch(approx.)**
Adagrad	18.4499	18.4477	4.1624	4.1622	6s
Adam	18.0541	0.3843	4.1090	0.4550	155s
RMSprop	18.4485	11.6930	4.1623	3.0608	96s
Adadelta	18.4498	18.4496	4.1624	4.1624	7s
Adamax	18.4499	18.3033	4.1624	4.1446	57s
Nadam	18.4242	7.8622	4.1592	2.4254	146s
Ftrl	18.4499	18.4499	4.1624	4.1624	7s
SGD	18.4499	18.4423	4.1624	4.1624	6s

Experimental result :

With Latent Factor = 20 and Epochs = 10, we have found the accuracy for all the optimizers and it can be concluded that Adam is the most suitable optimizer for the model.

Conclusion :

On conclusion of the second week we have totally grasped over matrix factorization and also have gotten a basic idea of Neural Networks and its cost functions. We have also completed the code of collaborative filtering using General Matrix Factorization and have tried several optimisers with different value of epochs before choosing Adam. Through this code we have also become familiar with deep learning library Keras and the python library Tensorflow.