בס"ד

316281047 רון סיידר

336208657 ג'ורדן טנג'י

Work Description:

Here is our github link : Please visit it https://github.com/jordantangy/Recommender-System

As mentioned in the proposal, our work consist of creating a neural network in order to predict the rating on a movie using linear regression.

Our data-set is composed of three main files that we are using for our work.

The first contains the movie Ids, userIds and ratings given by the users.

The second file is a list of 1128 tags for the movies. (Example of tags : absurd, realistic, fantastic…). There are 1128 words “describing” the content of the movie. Those 1128 tags are used as features for our neural network and that’s where the third file is used.

In the third file, for each movie we have a relevance number to know how relevant each tag is for each movie. For example, Super-Man has 0.8 “fantastic” relevance (almost 1) because indeed, Super Man is a fantastic movie and 0.0001 for “drama” because it is not a drama movie, and so on for each tag there is a relevance number.

What we want to predict is the rating that a user gave a movie based of the ratings he gave on other movies. Let’s have a look on how looks the rating table for each user : Une image contenant table

Description générée automatiquement

(In general, one user gave approximately ratings so consider this list to be longer).

Let’s say that user1 gave a 100 ratings. We split that data in 80% training set and 20% test set.

Training phase :

For the first movie (in the tab, movieId 296), we retrieve the features for that specific movie (1128 relevance numbers), and we feed the input layer of our neural network with those 1128 numbers.

We initialize the weights as a tf Variable of shape [1128,1].

We initialize a mean loss squared function that will take as input the predicted y value and the actual y value, that is, the predicted rating and the actual rating.

We updated the weights with gradient Descent optimization.

Then for each movie in the training set, we take the related relevance numbers (features) and we are starting a session that will update the weights and the loss function. At the end we obtain our trained neural network with updated weights.

Test phase :

Now that we finished to train our network, we obtained updated weights.

We retrieve the data of our test set. For a given movieId, we retrieve it’s features (relevance) and are multiplying them by the weights. The result should be the predicted rating that user1 for this same movie.

(Our prediction are working for single users only, that is, we split the rating data by users. Meaning that we predict the rating that userX would give on a movie based on the previous movies that he rated. And once our Network is built, we can feed do the same for any user is the dataset).

Here are the results

Une image contenant texte

Description générée automatiquement

Unfortunately we didn’t get the last prediction because of an error, we didn’t succeed to get trough it.

In a simple linear regression, we would have much less features (one or two), and the prediction would have been very inaccurate.