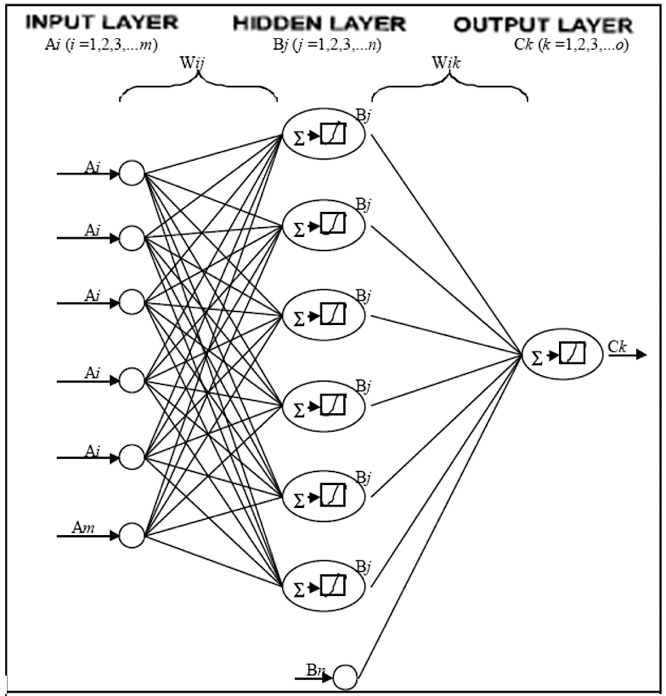
Multilayer Perceptron Regression

Multilayer Perceptron(MLP) is an extension of linear classifier(perceptron). It is a neural network. Each layer assign each feature a weight w and a constant bias w0 then threshold it with a non-linear activation function for binary classification. MLP utilizes backpropagation supervised learning technique. It classifies data that are not linearly separable. [?]



This approach is attempted with the following setups:

* MLP(PCA(BoWdescription\_nouns, 1000), FC1000) with 1000 components.
* MLP(PCA(BoWdescription, 2048), POOL5) with 2048 components.
* MLP(PCA(BoWdescription, 2048), BoWtags) with 91 components.

The MLP models were latter outperformed by PLS models.

Ranking pictures based on tags:

Modeling Approach:

Tag is a great intermediate resource used to map descriptions to relevant pictures. It captures the essential objects that could be found in a picture. Also, tags are highly likely to appear in the description. Therefore, a bag of words of tags is created by converting tags to stemmer words and filtering with stop words. There are in total 91 unique tags left.

All tags are of following the format below:

category:subcategory

animal:cow

Category only captures the high-level class of the object which, as the above example “animal”, may not appear in the description. Therefore, when creating the bag of words, subcategory tags are assigned with twice as much the weights as category tags.

Tag space needs to be mapped to description space. As before, PCA with 1000 components was applied to the description bag of words to reduce noise and computation expense. Tag space only has 91 columns, it does not need PCA processing.

Two regression methods are employed to map BoWdescription Space to BoWtag space:

* MLP(PCA(BoWdescription, 2048), BoWtags) with 91 components.
* PLS(PCA(BoWdescription, 2048), BoWtags) with 91 components.

PLS outperform MLP because it better describes relations between 2 spaces. Therefore, it is chosen to be included in the final model.

Ensemble Methods:

Method 1: Plain average

The first try of the ensemble model is by weighting each model equally. Since it is not clear why one model is performing better than the other. Plain average of every model is a good start. It is later proven to be outperformed by weighted average.

Method 2: Weighted average

When having a few working predicting models, an ensemble method boosts performance by aggregating the results from each model. Each model has a knn ranked 80 most relevant pictures, so no relevant pictures would be loft out from the aggregation. The number 80 is a chosen after validation. Accuracy score would drop if it’s above or below 80.

The priorities of the picture ranking are:

1. If a picture appears in multiple models, it is more relevant.
2. If the picture is ranked higher in a model, it is more relevant since models use knn to rank pictures.
3. If a picture is in a model with a higher accuracy, it is more relevant.

Here is the proposed ensemble model that incorporates the above three priorities:

Final ranking =

Final ranking is the ranking of the pictures;

Rank is the rank of the picture in a single model;

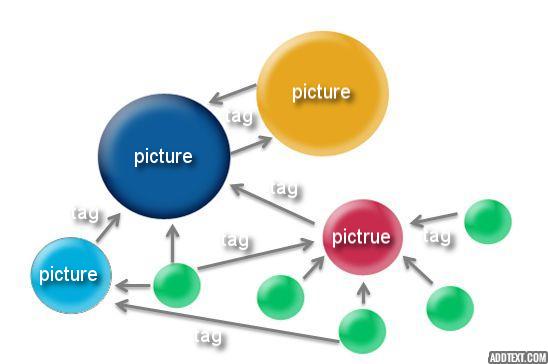
Weight is the accuracy score of the model.

Also, a few combinations of weights were employed to validate that the accuracy based weighted ensemble model indeed performs the best.

Method 3: Page rank

Despite that the result is ranked, other ways to optimize ranking was employed. Pictures should be ranked by their relevance to the description. Page rank is employed to rank pictures by relevance.

A network of all pictures as nodes was created. Pictures(nodes) are linked to each other if they share a common tag. Then for the final 20 result, the rankings are ranked by page rank with ε = 0.07 on the subnetwork of the 20 nodes.

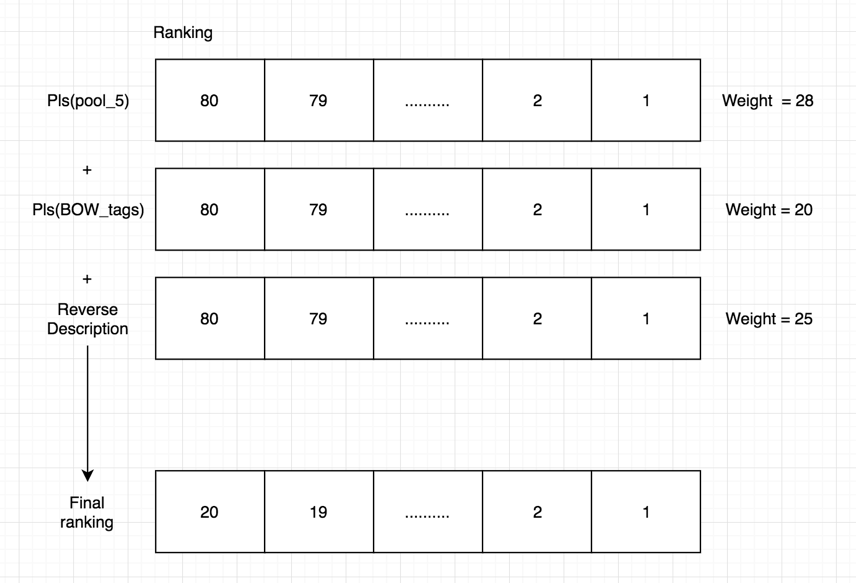


However, the result was not satisfactory. The accuracy dropped from 33% to 14%. There could be 2 reasons to explain the drop. First, the rankings were optimized by knn and the weighted average ensemble. Secondly, since the tags only describe pictures in high level, the network wasn’t a good representation of the picture description relations.

Final Version:

The final model is a combination of three models. 80 nearest neighbor pictures with cosine distance was selected for each model to higher the chance of including the right picture. For each picture, the scores are computed as 28\*Pls(Pool\_5)rank + 20\*Pls(BOW\_tags)rank + 25\*Reverserank description.

Here is a graph representation of the final model:



The higher the total score, the higher the final ranking. The 20 pictures come from the top 20 pictures in the aggregated score.

The accuracy for this final model is 0.36735.

Reference:

[?] http://www.sciencedirect.com/science/article/pii/S1352231097004470