Assignment 2 - Deep Learning on Images

Luka Beverin[1], Sanya Anand[1], Prerna[1], and Dishani[1]

[1]KU Leuven, Master of Statistics and Data Science, Advanced Analytics in Business

Introduction

The objective of this assignment is to construct a deep learning model in order to predict the number of likes an image receives. Many factors could contribute to the popularity of an image such as the posting time, content, who posted the image and features of an image like size, color, quality. For instance, a greater size image could more popular as they tend to be of higher quality.
The data set used for this assignment consists of 11,695 images of recipes along with meta data including information of number of likes, photo id, url of the image, scrapping and posting time, tags and may be column to give some idea about the image. However, it was not used for training the model. For this assignment, numerous deep learning models were explored and the one model with 3 hidden layers and ADAM optimizer outperformed the others. Moreover, interpretability techniques were used to evaluate the degree to which our selected model can be understood by the humans.

Data Preprocessing

The foremost steps after data collection is data preparation and data preprocessing. The idea is to transform the data in such a way that it is appropriate for modeling. This includes data cleaning(detecting outliers, treating missing values), data selection (selecting important variables), data transformation (changing the types of variables) and feature extraction.

Images Data

A. Feature Engineering. The aesthetic appeal of an image can be a crucial contributing factor in image's popularity and is primarily determined by content (What the image depicts? in our case recipes), context (What additional information accompanies the image? eg: use of hashtags, timing of post) and composition (eg: shape, color, quality).

Image Quality. Quality of an image can be an essential factor to the popularity of an image but it is highly subjective topic. Therefore, an Image Quality Assessment technique is used which helps to quantify the perception of humans towards quality. There are two methods in image quality assessment:

1. *Reference based evaluation* wherein an image is compared with a high quality image and scored accordingly.

2. *No-Reference based evaluation* which does not require any reference image and the only information provided to it is a distorted image whose quality is being assessed.

In this assignment, we have used BRISQUE *(Blind Referenceless Image Spatial Quality Evaluator)* to quantify the quality of our images. This model uses image pixels to calculate features and scores the image by comparing it with base model computed from images of natural scenes. Due to its low computational complexity and availability of Python library "pybrisque", we have considered using this technique for quality assessment of our images. The lowest value indicates the highest quality image.

Colorfulness of an Image. Colorfulness of an image can be related to the number of likes an image receives. For instance, more colorful image is more likely to receive higher number of likes. Hence, we tried to compute the colorfulness score of an image as described in in Hasler and Süsstrunk's 2003 paper and the highest score corresponds to the most colorful images in our data set.

Shape. Shape of an image such as square, rectangle contributes to the appeal of an image and can also affect the number of likes of an image. We extracted the size of each image and categorized them into different shapes based on their sizes.

B. Treatment of Outliers. We found one image with photo id "CMmV2A2sBvY" to be corrupt and hence, removed it from the data. There were four images that had quality score and colorfulness score negative and on further investigation, we found those images to be blank. We have also removed them from our data set.

Preprocessing	No of rows	No of features	New features added
Before	11695	104	-
After	11691	105	9

Table 1. Summary of the meta data

Meta data

Feature Engineering.

Extracted Time Related Features. The day on which a user posts an image can also affect the number of likes an image receives. For instance, an image posted on weekend might have more likes as compared to the ones posted on weekday. Similarly, posting time(hour and minute) could also be an essential factor and hence, we extracted these from the "posted at" variable. In the weekday variable, we can see that the most usual weekday to post is "Saturday" and the most uncommon is "Monday". In the hour variable, the most common hour to post is about 15-20 hours range. This might be due to the fact that most people are online during this time implying that the post reach is higher and, hence higher probability of likes. For the minute variable, the distribution is balanced and we do not see any relation between post minutes and number of likes.

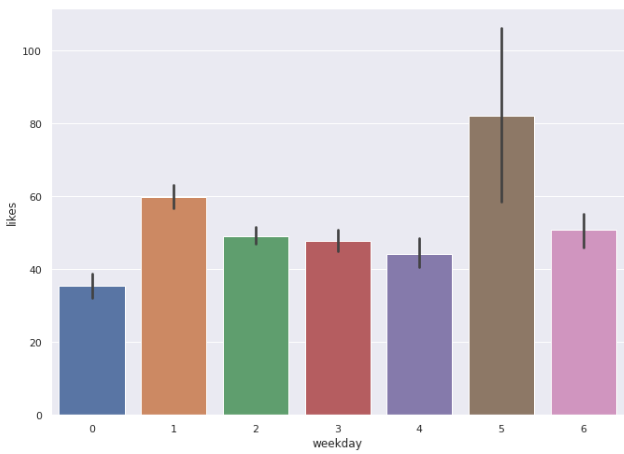

Fig. 1. Number of likes according to weekday

We have used a mathematical method proposed by Ian London (in his blog) for treating periodic features to map the hour, minute and weekday variables. The discontinuity that occurs when the hours goes from 24 to 0, minute goes from 59 to 0 and weekdays that goes from Sunday (6) to Monday (0) can be reduced by mapping the posting time and weekday onto a 2D circle. Each variable is mapped to a circle in such a way that the largest value appears right before the smallest value. These components are computed using trigonometric functions: sine and cosine.

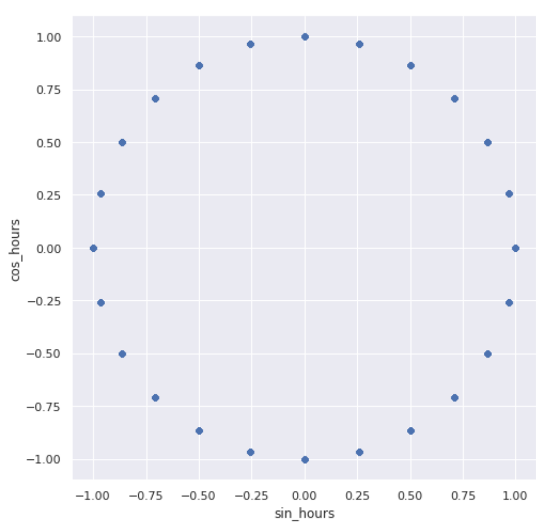

Fig. 2. Encoding Cyclical Feature - Hours

Number of Hashtags. More hashtags could mean that the image is more exposed publicly, i.e., to more people outside of the follower base of the poster, thus increasing popularity.

Number of Likes. The diff variable (time elapsed) indicates the difference between the time the image was posted and the time it was downloaded by the web scrapper. It could be the case that small value of diff variable means less popularity as people would have less time to interact with the image and like it. To take this into account, instead of predicting the likes variables, our target variable is "likes per diff" variable which is computed by dividing likes variable with diff variable.

Features Extracted from Images Data. Features extracted from images: colorfulness score, image quality score and shape variables were merged with the meta data. The shape variable was encoded into 1 representing square and 0 for any other shape.

Models

After properly preprocessing the variables, we used two sequential Keras deep learning models, with 3 layers each, that performed better than other models explored. Primary difference between both models is the optimiser used. The following parameters corresponds to the proposed model:

1. **Architechture:**

   The preprocessed variables were connected to three residual and dense layers. The dense layer transform the number of nodes and proceeds from 420 (105*4) to 210 (105*2), 105 and then ultimately applied the linear activation function to our final neuron to predict the number of likes.

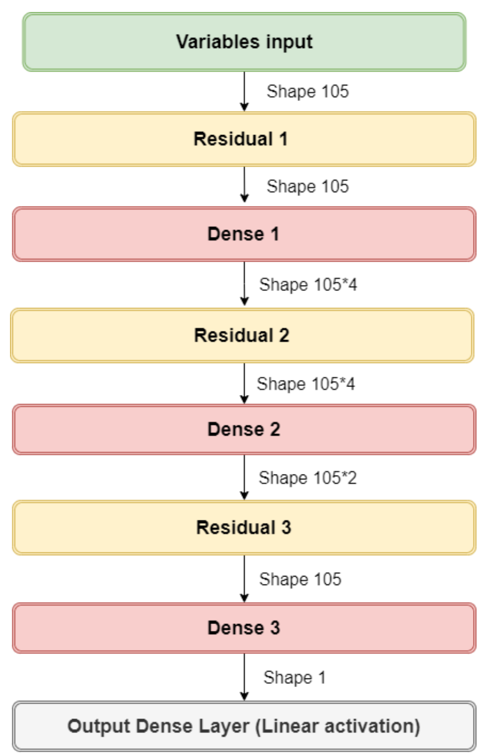

Fig. 3. Architechture of our models

2. **Optimizer - ADAM:**

To train our proposed model, we used ADAM as an optimiser. Adam is a stochastic gradient descent optimization algorithm that provides more efficient neural network weights by running repeated cycles of adaptive moment. According to Kingma, ADAM is computationally efficient, has little memory requirement and is effective for large datasets. ADAM combines the advantages of two other stochastic gradient techniques, Adaptive Gradients and Root Mean Square Propagation, in order to optimise the neural networks.

3. **Dataset Standardisation: Robust Scaler:**

   Standardization of a dataset is a common to achieve better results. Typically this is done by removing the mean and scaling to unit variance. However, outliers can often influence the sample mean / variance in a negative way. To overcome this, the median and interquartile range have been used to standardise numerical input variables, generally referred to as robust scaling.

4. **Activation function - RELU:**

   Our aim is to predict the number of likes of the pictures posted which is a positive continuous variable and thus, selected RELU as an activation function.

5. **Loss function: MSE**

   Since, our target variable is real and continuous valued, it can be viewed as a regression problem and hence, we chose the loss function as Mean squared Error (MSE). MSE measures the average of the squares of the errors, i.e., the average of the squared difference between predicted and actual values. If there are n samples of data, Y vector being the actual values of the number of likes and estimated values given by $\hat{Y}$, the MSE can be written as:

$$MSE = (1/n) \sum (Y - \hat{Y}) \quad \textbf{(1)}$$

   In order to find the best fit line, we aim to minimise the MSE.

6. **Evaluation metric - MAE:**

   To analyse the predictive accuracy of the model, we have selected Mean Absolute Error (MAE) as it is advised to be a good measure to evaluate the performance when multiple models needs to be compared on the same data. It measures the sum of differences between estimated values and actual values. The formula for MAE is given as follows:

$$MAE = (1/n) \sum (Y - x) \quad \textbf{(2)}$$

   where n is the number of samples in the data set, Y is the estimated value and x is the observed value. The reason for selecting MAE instead of RMSE (Root Mean Squared Error) is that RMSE increases with variance of the frequency distribution of error magnitudes. It means that RMSE will give more weight to large errors if the variance of the target variable is large.

7. **Methods used for improving model:**

   - Reduce Learning Rate on Plateau Schedule
   - To prevent over fitting
     - L2 - Least Square Errors (LS) Regularisation
     - Early stopping

8. **Results**

	Model 1	Model 2
Optimiser	Adam	SGD
MSE (train)	12.04	13.9
MAE (train)	1.33	1.39
MSE (test)	13.36	13.68
MAE (test)	1.8	1.27

Table 2. Summary of the Results

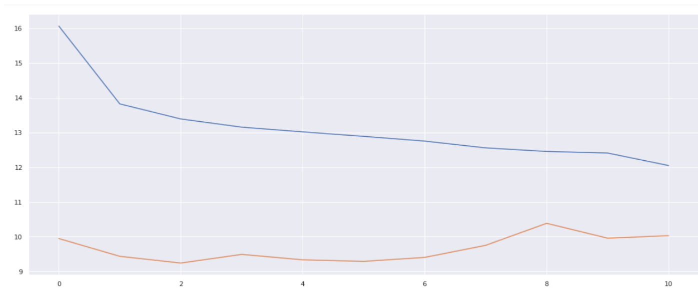

Fig. 4. Train and Validation Learning Curves for Model 1

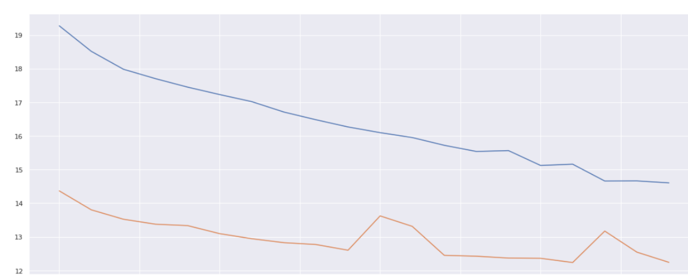

Fig. 5. Train and Validation Learning Curves for Model 2

Interpretability techniques

Due to over-parameterized black-box nature, it is often difficult to understand the prediction results of deep models. Interpretability in machine learning is useful because it can aid in trust. We tried to interpret the deep learning model using following 3 python libraries.

1. Library Shap: Plotting the mean values of the contributions shows the average contribution of each feature variable to the overall mean model output. This helps to understand the variable importance in the prediction of likes.

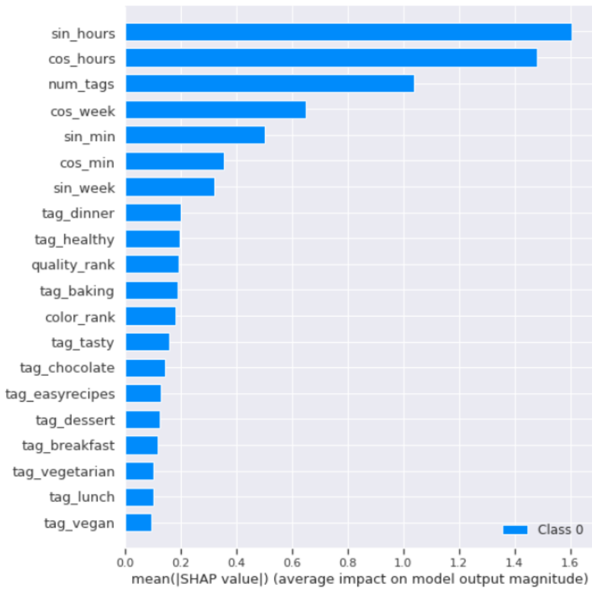

Fig. 6. Average impact of each feature

The variables with high mean values have high impact in predicting number of likes of a particular image. Variables Sin_hours, cos_hours and num_tags have highest impact while predicting the number of likes, on the other hand variables tag_vegan and tag_lunch do not contribute that much.

2. Library LIME: LIME is a clever algorithm that achieves interpretability of any black box classifier or regressor by performing a local approximation around an individual prediction using an interpretable model. It is a good algorithm to inspect or debug individual predictions.

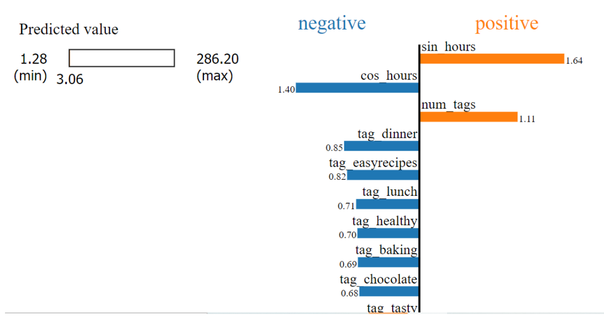

Fig. 7. Average impact of each feature

We can see that the features sin_hours and num_tags have all positive contributions to the LIME model although in not the same order as SHAP. Also, other variables in blue have negative contributions to the LIME model. This is the short snippet of some variables (*It does not have all the variables*)

3. Library SP-LIME: This library picks a series of instances of the model and their corresponding predictions in a way that is representative of the whole model performance. These picks are performed in such a way that input features that explain more different instances have higher importance score.

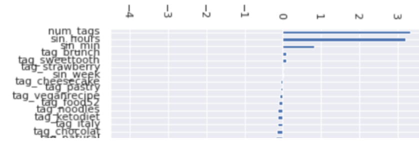

Fig. 8. Average impact of each feature

We can see that the features sin_hours, num_tags have all positive contributions to the model. Also, features tag_chocolate, tag_italy have negative contributions to the model.

Terminology

1. *L2 Regularization:* Regularisation is a technique that helps to generalise the model. L2 regularization adds a sum of the squared parameter weights term to the loss function which helps in avoiding over fitting by penalising large parameters in favor of small parameters.

2. *Early Stopping:* It is a feature that enables the training of the model to be automatically stopped when the chosen metric has stopped improving. It is a form of regularisation used to prevent over-fitting. It monitors the evolution of the given metric after every epoch and is stopped when the metric stops improving for patience epoch in a row. Patience epoch is the number of epochs without improvement after which the training will be stopped. A larger patience means the model waits longer before the model is stopped.

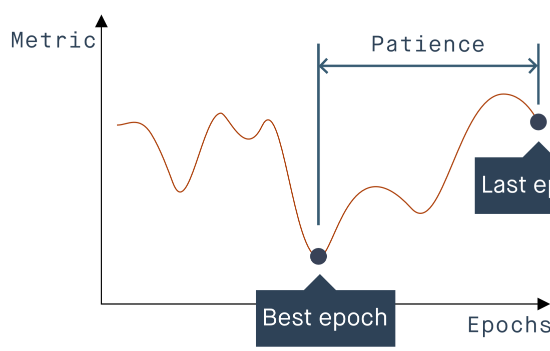

Fig. 9. Early Stopping

3. *Reduce Learning Rate on Plateau Schedule:* It decreases the learning rate by the specified Decay percentage when the metric is stagnating longer than the patience allowed.It used when we don't have information about how the data will behave and when the metrics flatten out (stops improving).

References

1. Instagram Likes for Architectural Photos Can Be Predicted by Quantitative Balance Measures and Curvature

2. Encoding Cyclical Continuous Features- 24-hour time (by Ian London)
3. Interpretability of deep-learning-models
4. Kingma:Adam: A Method for Stochastic Optimization