Report of Iterative Development

Machine Learning Project

Brian Jakobs, Vera Eising, Joram Brokkelkamp, Damla Baspinar

Minor AI

Teachers: Tim Doolan, Wouter Vrielink

Date: the 13th of January, 2022

Contents

1. Milestone I: First Model

1.1 Introduction

Sheltered cats and dogs display specific characteristics which can be considered determinants of adoption[1]. Since animals that are not adopted can be subject to suffering in different forms[2], it seems logical for shelters to attempt to focus their resources on determinants with an increasing effect on the likelihood that a given animal will be adopted[1]. Petfinder.my, a Malaysian platform regarding animal welfare, operates using a similar concept[3]. The organization uses artificial intelligence to determine a score for a Cuteness Meter, which corresponds to the attractiveness of pictures of cats and dogs[4]. A higher score correlates to a more attractive picture[4]. This score can be used to improve photo-quality in order to optimize the chances of adoption[4]. The current algorithm does not consistently function adequately as it is still under development[4]. The model described in this report seeks to provide an improved version of this algorithm. The aim of this report is to elucidate the process of formation of said model.

The first version of the model in question is a convolutional neural network which takes images of animals and designated attractiveness scores as inputs and performs a number of computations to estimate the score of attractiveness for another given image.The rationale behind this approach is that deep learning is known to be successful in tasks that require image recognition[5]. The goal regarding the performance of this model, is to achieve a model that shows a decrease in its loss over a number of epochs. This shows the designed model has the capacity to be trained.

1.2 Data Analysis and Preprocessing

The data provided for every sample is composed of the images of a set of pets and additional tabular data regarding every given image. The tabular data states whether the following composition parameters were present or absent in a given image: "*Focus (Pet stands out against uncluttered background, not too close / far), Eyes (Both eyes are facing front or near-front, with at least 1 eye / pupil decently clear), Face (Decently clear face, facing front or near-front), Near (Single pet taking up significant portion of photo (roughly over 50% of photo width or height), Action (Pet in the middle of an action (e.g., jumping)), Accessory (Accompanying physical or digital accessory / prop (i.e. toy, digital sticker), excluding collar and leash), Group (More than 1 pet in the photo), Collage (Digitally-retouched photo (i.e. with digital photo frame, combination of multiple photos), Human (Human in the photo), Occlusion (Specific undesirable objects blocking part of the pet (i.e. human, cage or fence). Note that not all blocking objects are considered occlusion), Info (Custom-added text or labels (i.e. pet name, description)), Blur (Noticeably out of focus or noisy, especially for the pet's eyes and face. For Blur entries, "Eyes" column is always set to 0)*"[6]. Additionally, the tabular data also includes the "*Pawpularity*", which reflects each pet profile's page view statistics[6]. It was noted that this dataset only contains 9912 samples, which is a relatively small amount for training.

The tabular data was analyzed using the *Normal Equation*. It was established that the relation between the features and the outcome “*Pawpularity*” was not linear. Furthermore, it was established that the tabular data does not contain any missing values.

The data used for this version of the model comprises two parts: the “*Pawpularity*” within the tabular data and the images of pets.
With regard to the tabular data, the “*Pawpularity*” was modified before use. The score, originally given in percentages, was divided by 100 to compute the decimal equivalent of the score for every image.
Regarding the images, every image was resized to a shape of 128x128x3.All provided testing data had this specific shape, whereas there the shapes of the samples in the training data differed. Therefore, it was decided to resize all training data to the same shape as testing data. Additionally, these relatively large images allow training with a large amount of information.
Furthermore, the data was split into training and testing data, corresponding with 67% and 33% of all data respectively.

1.3 Model Pipeline and Training

The current predictive model is a convolutional neural network. The input of a given sample is the image of a pet, whereas the output is the “*Pawpularity*” score.
The architecture of the convolutional neural network consists of an input layer, three hidden layers and an output layer.

The input layer is a 2D convolution layer, the input shape is (128,128,3), the kernel size is (3,3) and the number of filters is 64. In this layer, the ReLu activation function was used and the zero-padding was applied. This is followed by a max-pooling layer with a pool-size of (2,2).

The first hidden layer is a fully connected layer with 512 units and the ReLu activation function. Next, dropout with a rate of 0.1 is applied. The following hidden layer has 256 units, and the ReLu activation function was applied. The last hidden layer is a fully connected layer with128 units and the ReLu activation function.

The output layer provides one output. No activation function was applied in this layer. Given that the task at hand is a regression problem, no non-linear activation function should be necessary in the last layer.
The output shapes and parameters of all layers are presented in Figure 1.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 128, 128, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 64, 64, 64)       0
 )

 flatten (Flatten)           (None, 262144)            0

 dense (Dense)               (None, 512)               134218240

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 256)               131328

 dense_2 (Dense)             (None, 128)               32896

 dense_3 (Dense)             (None, 1)                 129

=================================================================
Total params: 134,384,385
Trainable params: 134,384,385
Non-trainable params: 0
_________________________________________________________________
```

Figure 1: The output shapes and the numbers of parameters in every layer of the model.

Concerning the training of the model, the Mean Squared Error was used as the loss function for optimization. Additionally, the Root Mean Squared Error was used as a metric to judge the performance of the model. The optimizer Adam was applied. The model was trained using 20 epochs.

No further preprocessing or augmentation was applied to the images.

1.4 Evaluation and Conclusions

Regarding the model loss, there is a clear decrease for the training data (Figure 2). Although this decrease seems to stagnate after the first epoch in Figure 2, it is actually consistent over the course of 20 epochs. The loss for the validation loss also consistently decreases through 20 epochs.
A similar trend was observed in the root mean squared error; the error decreases over the course of 20 epochs in both the training and the validation data, though this is not clearly visible in the graph of the learning curve (Figure 2).
The decreases in both the model loss as well as the root mean squared error are a manifestation of the trainability of the model, thus the goal of forming a model with the capacity to be trained was obtained.

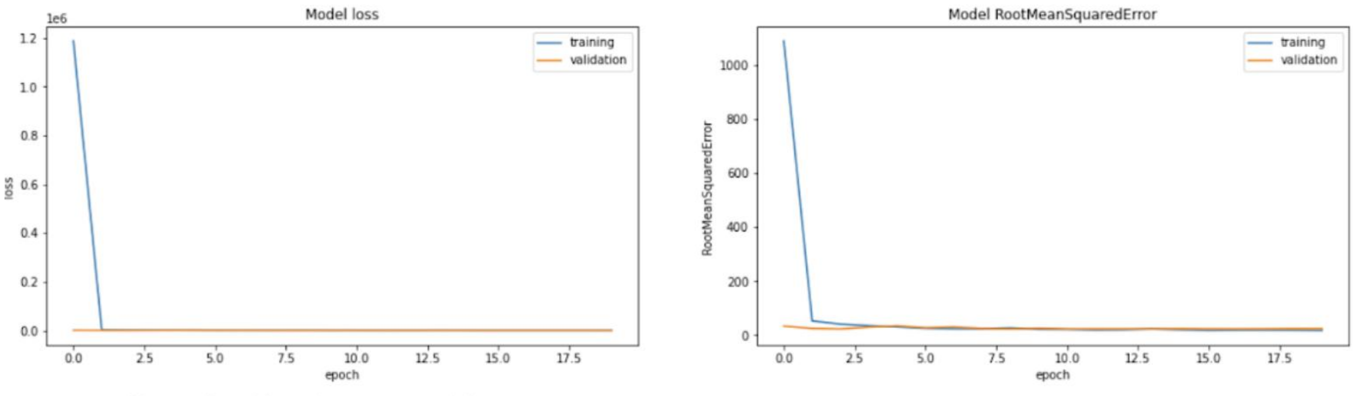

Figure 2: This figure shows the graphs of convergence of the model over 20 epochs.

Figure 2 shows that the model loss and root mean squared error were relatively high after the first epoch. Because of this the relatively small nuances between the training and the validation values are not visible in the learning curves. In order to clarify this image, the y-axis for the model loss was limited to a value of 10000 and the y-axis for the root mean squared error was limited to 100 for the model loss and root mean squared error respectively. The outcome is illustrated in Figure 3.

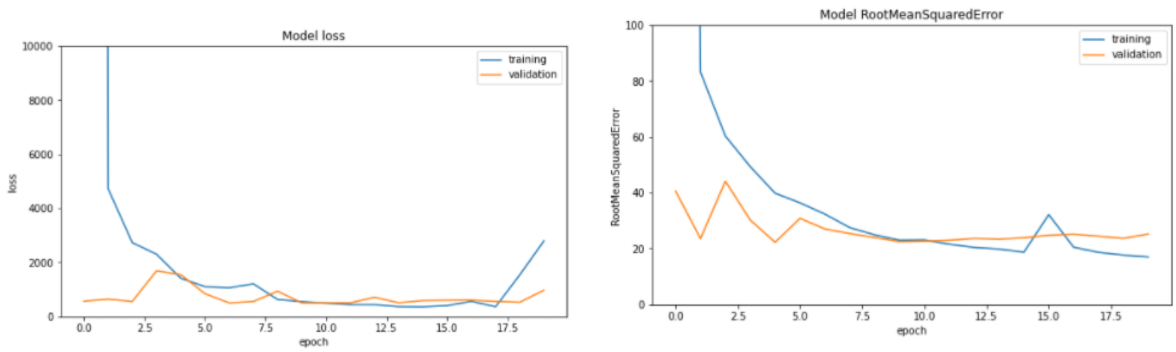

Figure 3: This figure shows the graphs of convergence of the model over 20 epochs with limited y-axes.

In Figure 3, the graph of the model loss shows that after approximately 17 epochs, the model loss and root mean square error both start to increase for the training data. This is a manifestation of divergence rather than convergence in gradient descent. This could be caused by a learning rate that is too high.

Given the small amount of available data,the application of k-cross validation and data augmentation could improve the training process of the model in future versions. Additionally, the use of preprocessing and augmentation methods in the images could derive better results. Furthermore, no activation function is applied in the last layer of the current model, whilst using a linear activation function could also be possible. Lastly, the problem of divergence should be solved.

References

1. Lepper M, Kass PH, Hart LA. Prediction of adoption versus euthanasia among dogs and cats in a California animal shelter. J Appl Anim Welf Sci. 2002;5(1):29-42.
2. PETA. The Deadly Consequences of ‘No-Kill’ Policies. [Internet]. Available from: https://www.peta.org/features/deadly-consequences-no-kill-policies/. [Accessed at 11-01-2021].
3. Petfinder.my. About Petfinder.my. [Internet]. Available from: https://www.petfinder.my/about.htm. [Accessed at 11-01-2021].
4. Petfinder.my. Cuteness Meter — How Attractive Are Your Photos?. [Internet]. Available from: https://www.petfinder.my/campaigns/cutenessmeter.htm. [Accessed at 11-01-2021].
5. Molnar C. Interpretable Machine Learning. Lulu.com; 2020.
6. Kaggle. PetFinder.my - Pawpularity Contest: Data. [Internet]. Available from: https://www.kaggle.com/c/petfinder-pawpularity-score/data. [Accessed at 12-01-2021].