Animal Adoptability Analysis for CS230-Winter 2018

Hermann Qiu
Department of Computer Science
Stanford University
hq2128@stanford.edu

Xu Zhao
Department of Computer Science
Stanford University
pglory@stanford.edu

Chun Kit Chan
Department of Computer Science
Stanford University
cckit@stanford.edu

https://github.com/HermannQQQ/Animal-Adoptibility-Analysis

Abstract

The long-term motivation of this work is to create an application that guide shelters and rescues around the world on improving their pet profiles' appeal, reducing animal suffering and euthanization. The first step is to properly predict the adoption speed given a pet profile data. In order to achieve this, we employ deep learning architectures. In particular, we use different methods of feature generation, and different networks for each type of data. One method employs a pre-trained CNN to extract feature vectors from pet images, then we concatenate the image vectors to pre-processed structured vectors, and feed into a simple network to predict the adoption speed. We use Quadratic Weighted Kappa (approximate) Loss as our loss function, which measures the degree of closeness between two measures -- our prediction vs. true label.

1 Introduction

The goal of this project is to apply deep learning to predict the adoptability of pets – specifically, how quickly a pet is adopted. In the United States alone, there are about 1.5 million companion animals are euthanized in shelters. [8]. This project can guide shelters and rescues around the world on improving their pet profiles' appeal, reducing animal suffering and euthanization. The input to our algorithm is the profile for each pet, including structure data(age, color, etc) and non-structure data (pet images and descriptions of the pet). We use pre-trained ResNet50 on image data to extract a vector of 2048 neurons, then concatenate the vector to the pre-processed existing structure data columns, and run through a dense layer neural network. The output consists of 5 neurons with one-hot encoding which represents 5 bins of adoption speed. We use softmax for the output layer and *QW-Kappa* loss function (for our best model).

2 Related Work

Our project is based on a 2019 Kaggle competition. The public leaderboards top entry has 0.492 Quadratic Weighted Kappa Score as of Mar,19th, 2019. Although research on this topic is not abundant, earlier ones found that age, sex, coat color and reason for relinquishment are the major factors. [5][7]We noted we were not provided the reason for relinquishment in this data set, but this has been the most dominating factor from earlier researches. Physical characteristic, personality, breed features are themes among adopters seek. [9]When predicting the adoptability, people also looked at behaviors and contextual predictors: [4]Protopopva and Wynne found that dogs that were adopted spent half as much time ignoring play initiation by and twice as much time lying in proximity to the adopter than dogs that were not adopted. Overall, these earlier work indicate the adoptiblity depends on highly complicated interactions among many factors, some are not able to be provided in current dataset, which may explain why so far no one has ever broken the 50.0% (*QW-Kappa* Score) in the leaderboard.

3 Dataset and Features

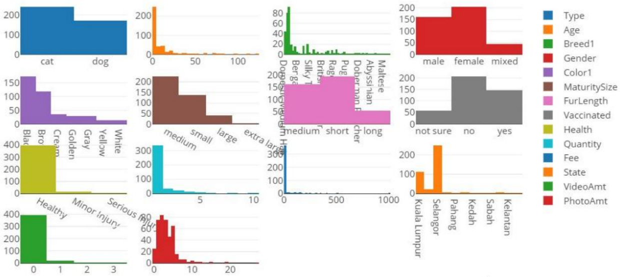
Overview

Data overview: 14993 pet profiles with labels from 0 to 4 indicating the adoption speed (the smaller the label value, the faster), and also 58311 pet images. Data source: https://www.kaggle.com/c/petfinder-adoption-prediction/data

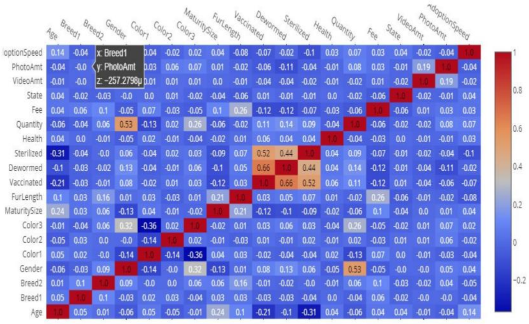
Types of data

a. Structured data: 24 columns in total, including numerical data, nominal data and ordinal data.

The plot on the right shows different distribution of animal adoption speed by sample features - some of them are numerical features (e.g. age) and some of them are categorical features (e.g. type).



And the correlation heatmap furthur shows that only a few numerical features have certain degree linear correlations with the animal adoption speed. Categorical data is further processed into one-hot encoding vectors. Among those, only age stand out, but even that only has a correlation of 0.14. This number is probably lower than people would expect.



b. Image data: 58311 pet images. One pet profile can have more than one photo, and some photos include multiple pets.

4 Methods

Learning algorithm

For structured data, we preprocessed data by dropping irrelevant columns (e.g. pet_id) and applying one-hot encoding to categorical columns. In the end, every sample has a feature vector size as of 5970. For image data, we picked and scaled the first image and used a dummy black image for those don't have. All images are (224x224). Transfer learning is done here using ResNet50, dropping the last layer of the pretrained network, and appending a global 2D pooling layer with 2048-length vector.

After all, vectors from structured data and image data are concatenated into single vector. In order to speed up the training process, 2048-length image vectors are precomputed. Please note that we are allowing options here either to concatenate two data at the beginning of the network or at later stage.

Then the input is passed to a 6-dense-layer network with each having 256 neurons, performing batch normalization, using ReLU activation and dropout in series. Softmax activation is applied at the output layer which has 5 neurons to represent 5 different classes for the label.

Batch normalization weakens the coupling between layers and reduces the effect of covariate shift so that mode learning becomes faster. ReLU activation does not have the gradient saturated problem like sigmoid and tanh so learning is faster in our case because our input matrix is as of sizes (, 8018). Dropout acts as a tools for regularization.

Loss function and metrics

The objective of the learning algorithm is to classify an ordinal value. The built-in loss function categorical_crossentropy and metric categorical_accuracy from Keras are not closely aligned with the task. For instance, if the true adoption speed of dog A is 8-30 days and one algorithm α says that dog A will be adopted on the same day and algorithm β says it should need 1-7 days, algorithm β obviously outperforms algorithm α. If categorical_crossentropy is used, α and β would have the same cost, not reflecting β better performance.

0 - Pet was adopted on the same day as it was listed.
1 - Pet was adopted between 1 and 7 days (1st week) after being listed.
2 - Pet was adopted between 8 and 30 days (1st month) after being listed.
3 - Pet was adopted between 31 and 90 days (2nd & 3rd month) after being listed.
4 - No adoption after 100 days of being listed. (There are no pets in this dataset that waited between 90 and 100 days).

In order to tackle this problem, quadratic weighted kappa Loss(1- *QW-Kappa* Score) is used instead and quadratic weighted kappa score(*QW-Kappa* Score) as our metrics. Using *QW-Kappa* ensures the more distant prediction compare to true label is punished much more than the closer one. Using the notation from *Cohen's Kappa* where p_{ij} are the observed probabilities, $e_{ij} = p_i q_j$ are the expected probabilities and w_{ij} are the weights (with $w_{ji} = w_{ij}$) , then:

$$\kappa_w = 1 - \frac{\sum_{i,j} w_{ij} p_{ij}}{\sum_{i,j} w_{ij} e_{ij}}$$

5 Experiments/Results/Discussion

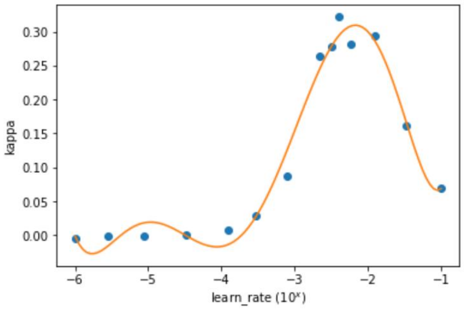
Dataset preparation

Raw data is divided into 3 datasets: training set, dev set and test set. The ratio of the division is 90%, 5% and 5%. We also prepared a training-dev set from 5% of training set for the potential need to verify whether the model s has data mismatch problem..

Hyperparameter search

Manual search: Hyperparameter search are conducted in two forms: A random search phase,where all major hyperparameters including learning rate, drop-out rate, l2 regularization lambda, average number of unit per layer and batch_size. Variation of whether the last few layers of ResNet is also trainable, whether to keep the last layer of ResNet, and whether to concatenate certain ResNet layers before feeding into dense blocks are also tested. Performance metrics are recorded and used to guide further efforts in refining our model.

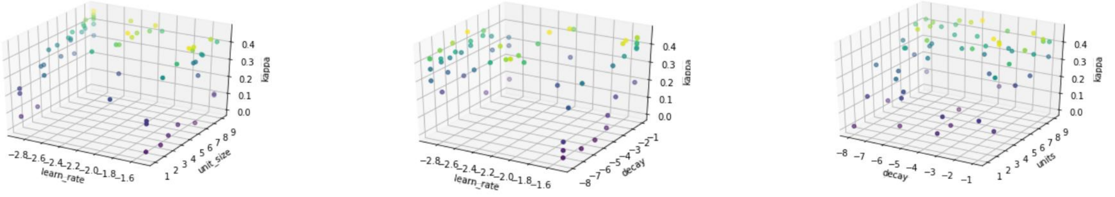
We found that for certain parameters where we can assign different values among different layers, e.g. dropout or l2 lambda,if we choose to assign these values randomly for each individual layer, we consistently get poorer performance. Further investigation indicated, if we choose to randomize individually, given the high number of random run, some will get a high number and this layer usually control the performance (to the poor side). So in later phase of this project, we choose one dropout rate/one regularization constant. Another thing we notices is that, the last layer is very sensitive to drop-out, and removing drop-out from the last layer seems at least stabilize the dev-set performance.



Automated search: to search hyperparameter systematically, Bayesian optimization is applied to look at good candidates for further search. The learning rate is usually the most important hyperparameter to be tuned. In the initial coarse search, learning rate is searched between 10^{-6} and 10^{-1}. The results shows that the distribution is similar to Gaussian distribution and the values between $10^{-2.8}$ and $10^{-1.6}$ perform much better than others. As the result, the range of learning rate for next fine search is narrowed down to this range.

The fine search is then extended to other parameters like hidden unit size (from 2^1 to 2^9) and learning rate decay (from 2^1 to 2^9). From the results, we can conclude some insights and strategies about hyperparameter tuning:

- Learning rate does not greatly affect the performance once the suitable range is picked
- Larger hidden unit size helps the performance: higher complexity empowers the learning ability of model
- Effect of learn rate decay is not very important, especially when learning rate is small



Neural network with Structured data

We explored a series of neural network architectures and hyperparameters. Hyperparameters are either randomly generated or assigned individually. The neural network are layered with standard dense block (See Methods) with neurons of 256-1024 units. In the early phase of the experiments, we featured our data mainly on the pet's characteristics, for example, age, sex, colors, breed etc, we dropped other data. In the later phase of the experiments, we added these non-pet characteristics back. For example, we added back the rescuer (by its ID), each rescuer is represented by a one-hot column. This will be discussed further in other sections , this non-direct information contains significant prediction power. In the later phase of the experiments, we settled with learning rate of 0.005 across different trials.

Neural network with Image data

We explored a series of neural network architectures and hyperparameters. Hyperparameters are either randomly generated or assigned individually. The neural network are layered with resnet pretrained network , stacked with standard dense block (See Methods) with neurons of 256-1024 units at the end. The pretrain network is either totally frozen (not trained) or the last few layers (1-3) are unfrozen (subject to training). Another variation we tested is merging the last few activation layers of resnet before fitting into dense blocks. In the later phase of the experiments, we settled with learning rate of 0.005 across different trials.

Neural network with merged structure data and Image data (precomputed)

Similar to above architectures and hyperparameter search mechanism, here the image data is precomputed into a 2048 vector, concatenated with existing structure data. Beyond the variation we used earlier, we also tested to fit precomputed image data into separate dense blocks and merge them at later layers. The merge point is also subjected to experimentation.

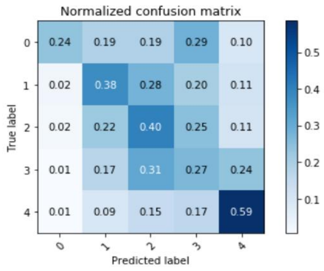
Results

The below table summarize the best performing model for each specific data-architecture pair:

Data Used	*Network*	*Loss Function*	*Accuracy - Dev Set*	*QW-Kappa Score - Dev Set*
Structure Data	*1- layer NN*	*Cross-entropy*	*30.40%*	**N/A*
		MSE	*27.00%*	**N/A*
Image Data	*Pre-trained ResNet + 2-layer NN*	*Cross-entropy*	*33.9%*	**N/A*
		MSE	*25.90%*	**N/A*
Structure Data	*6- layer NN*	*Cross-entropy*	*32.13%*	*24.56%*
		QW-Kappa Loss	*39.00%*	*42.00%*
Structure Data + Image Data	*Step1:Pre-trained ResNet for image feature extraction, then feed into 2-layer NN; Step2: Feed structure data to a 6-layer NN; Step3: Concatenate the results from 1 and 2, then run through softmax layer*	*Cross-entropy*	*40.40%*	*38.98%*
		QW-Kappa Loss	*****43.60%***	*****48.01%***

** Please note that we didn't run the QW-Kappa Score on the baseline cases, because the accuracy is too low.*
*** Best result so far.*

Error Analysis



The confusion matrix shows relatively positive results, considering that this is a very challenging project.

Below, We will focus on wrong predictions which are two categories away from true label:

1. Rescuer Effect: Result turns out to be significantly different with or without rescuer data. Most rescuer only has one animal profile. Some "star rescuer" has more than 10 animals profiles. The rescuers who rescued many more animals turn to have an average speed of 2.24 compared to 1.97 from single profile rescuer. Second, from the description, we find that rescuer acts differently which would impact the adoption speed, for example, certain rescuer is able and willing to host the pet longer and wait for best adopter. All these info needs to be provided.

2. Description Data: Also mentioned in the Future Work below, we will need to dig more information from the description that hasn't been touched at this stage. Some of these info has proven significant by previous work

3. Quantity Effect: When there are multiple pets in the pet profile, the adoption speed is not well-defined. This make the prediction harder.

4. Image Quality: Majority of the profiles have photos (98.5%), for profile without photo, or it is blurry, we've seen a large prediction gap.

6 Conclusion/Future Work

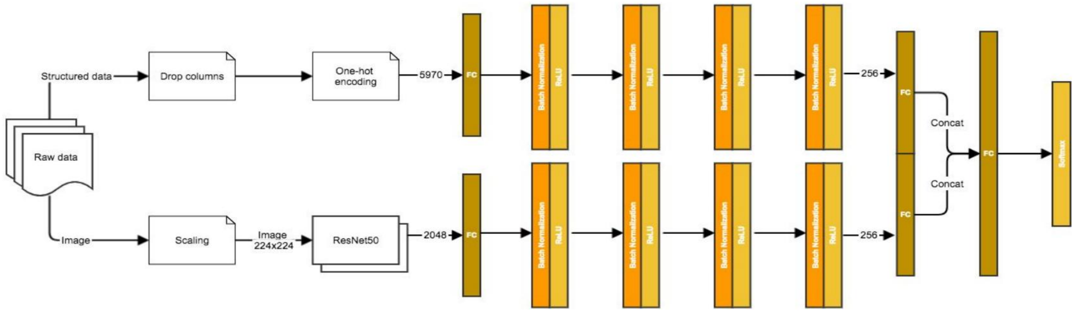
Summary

We were able to analyze different types of data in the rescued pet profile and predict their adoption speed, and indicated both current structured data and associated images contained certain prediction power. We explored couple of methods to combine these two types of data. In the end, the best result is obtained when data is merged at the very late phase. Given that these two types of data are quite different in many ways -- structured data is better represented in one-hot vector(as our experiments found out) but the image data is better fed into ResNet and represented as a real feature vector -- this may explain why a late merging performs better, as the neural network may need to learn to represent them in a consistent form before combining them.

As for the difficulties of this project, we knew from beginning that it's very challenging for -- the small dataset size(<15k samples), the weak relationship among different adoption factors, and the lack of relevant research. Earlier research has indicated the largest factor in determining adoptability is "reason for relinquishment", which is arguably absent from our dataset. Certain entries has descriptions mentioning the reason of relinquishment, but that is limited and the description data in the dataset are in different languages. As a comparison, the kaggle competition is still ongoing as the time of this writing, and the highest QW-Kappa score on the published leader board is at 0.49 The kaggle community also recognizes this issue and there are various discussions on "how to break the 0.5". Our data is evaluated on the dev set with an QW-Kappa score of 0.4801.

Best Model

As shown above in the Results section, the performance from different algorithms varies, and the one below is considered the best model we have, which has the highest QW-Kappa Score(*48.01%*), highest category accuracy(*43.60%*) and performs more stably than the others:



Future Work

1. We need to clean up the description text data for each profile, translate all non-English sentences to English, then feed the data through a sequence model for data analysis and feature extraction. We have seen the benefit from merging image features to structured features, and so believe that we could also benefit from the text features.
2. One alternative way to think about description text data is that -- based upon previous research study, the most promising action is trying to extract the "reason for pet relinquishment", which was indicated as the most crucial adoptability factor.
3. Current Resnet is trained to classify a large pool of class objects, which pets may all receive similar consideration. We can stack more layers
4. To our surprise, the rescuer ID played a significant role in predicting the adoption speed. Looking back, this may relate to implicit information linked to the rescuer, for example, the experience of the rescuer, the region where the rescuer works on. The result indicated we should look into better ways to extract/process this type of information.
5. Our teammate has already spent some time modularizing the code, and make it more convenient for future code addition and editing. The next step is improve the code structure and running efficiency, so that it will allow other users/companies/researchers to seamlessly feed in their data and see results.

7 Contributions

Hengkai Qiu	Exploratory data analysis / Data processing and Feature engineering / Model optimization with different loss function
Xu Zhao	Training environment setup / Code framework with image data and pretrained network / Error Analysis and merged model search
Chun Kit Chan	Code framework with structured data / Dataset generation and separation / Semi-automated hyperparameter searcher

References

[1] Liwei Wang, Yin Li, Jing Huang, Svetlana Lazebnik (2018). “Learning Two-Branch Neural Networks for Image-Text Matching Tasks”. https://arxiv.org/pdf/1704.03470.pdf

[2] Classifying e-commerce products based on images and text. http://cbonnett.github.io/Insight.html#So-what-about-classes-that-are-not-in-the-training-sample

[3] Smarter Ways to Encode Categorical Data for Machine Learning https://towardsdatascience.com/smarter-ways-to-encode-categorical-data-for-machine-learning-part-1-of-3-6dca2f71b159

[4] Adopter-dog interactions at the shelter: Behavioral and contextual predictors of adoption https://towardsdatascience.com/smarter-ways-to-encode-categorical-data-for-machine-learning-part-1-of-3-6dca2f71b159

[5] Prediction of adoption versus euthanasia among dogs and cats in a California animal shelter. https://www.pubfacts.com/detail/12738587/Prediction-of-adoption-versus-euthanasia-among-dogs-and-cats-in-a-California-animal-shelter

[6] Paedomorphic Facial Expressions Give Dogs a Selective Advantage

[7] Animal shelter dogs: factors predicting adoption versus euthanasia. https://soar.wichita.edu/bitstream/handle/10057/3647/d10022_DeLeeuw.pdf

[8] Shelter Animal Adoption Research: A 2015 Reviewhttps://faunalytics.org/shelter-animal-adoption-research-2015-review/

[9] The truth about humans: The decision to adopt dogs & cats. https://www.researchgate.net/publication/280317580_The_Truth_about_Humans_The_Decision_to_Adopt_Dogs_Cats