First Milestone

Team: HáLab

Team members: Stumphauser Nóra, Lestyan Bence

Motivation

In the project our goal is to compare the accuracy of simple classifiers and neural networks for a binary classification problem. We would like to show that the classifiers don't work as good on binary classification as neural networks on image classification. The problem in general is that there is a time-series dataset (for example user dataset, or like in our project weather data) and we would like to build a binary classifier (usually for churn, but we use a label invented by ourselves). Intuitively, the classifiers like decision trees, usually not work very well on time-series data. On the other hand, neural networks can classify images easily with very high accuracy. So, we create images from data (for each person, or in our case for each day) and we use already built neural networks for classification.

Literature review

About this method the literature usually contains researches about churn. In 2006 there was a huge investigation in Taiwan. [1] They used several machine learning techniques for predicting churn, for example decision trees, k-nearest neighbor. Also, they used neural networks, but not for image classification, just on the rows of the data. Similar article was published in 2012, where the researchers predict churn with a lot of machine learning techniques (Logistic Regressions, Linear Classifications, Naive Bayes, Decision Trees, Support Vector Machines and the Evolutionary Data Mining Algorithm) that we can use as methods for simple classifiers. [2] The idea of creating images from data came while reading an article from Asian researchers. [3] It is a short paper but shows that a simple neural network can do well on this type of data. They got 0.74 auc score on there data with 6 million customers. For next milestone we plan to find and read more articles about this image-idea, but for this time it seems like a new (they released their results in 2016) and not widely used technique.

Data preparation

We use a Kaggle dataset from this website: https://www.kaggle.com/muthuj7/weather-dataset

First, we load the files and change every value to a number. We divide the time column into day and hour. The raw data contained 13 columns and 96 453 rows. We realize that 96 453 not divisible by 24, so in the first part of the notebook we deal with the problem of duplicates and missing data. We deleted 11 rows which is negligible. And because of the missing data we should drop 14 days of the 4018 days, it is only 0,003% of the data so it will not mean a serious shortage. We need to delete a day that was twice in the data. We also drop a column, because it contains only 0 numbers. After deletions we got a dataset with 12 columns and 96 096 rows.

In the next step we create our target variable. We would like to choose a temperature-based target variable, so we see the median of the mean temperatures and we got that it is 12,25. So we create label 1 to those days where the mean is greater that 12,25. This case we got perfectly balanced data with 2002-2002 days in each set. After this step we drop the two columns which

contained too much information about temperature (it can be learnt easily), and saved the labels and the data into a csv file to use it in the other step of research (for simple classification).

Creating images

In this research we would like to create images, which pixels are colored by the values of the cells. We have 24 hours data for each day and 7 attributes, so we create 7x24 pixel images. For this we transform each column values into 0-255 by min-max scaling. In the end we got a 3D numpy array with 4004 7x24 colored images.

Train, validation, test sets

For Milestone 1 we create the train, valid and test datasets. We decided to make 70% for training, 15% for validation and 15% for test, but if we need to change these values during our work, we will. So we got 2802 train images and 601-601 for validation and test.

Future work

For next steps we will see how the simple classifiers work on this task. We will use logistic regression, some kind of decision trees, k-nearest neighbors, svm or some other binary classifier. During this, we will start to use some trained neural networks on our images. For conclusion we will compare the results of some well-known neural networks, and also the difference between efficacy of the neural networks and simple classifiers. We have a hypothesis that the neural networks will be better.

Bibliography

[1] Hung, S. Y., Yen, D. C., & Wang, H. Y. (2006). Applying data mining to telecom churn management. *Expert Systems with Applications, 31*(3), 515-524.

[2] Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, *39*(1), 1414-1425.

[3] Wangperawong, A., Brun, C., Laudy, O., & Pavasuthipaisit, R. (2016). Churn analysis using deep convolutional neural networks and autoencoders. *arXiv preprint arXiv:1604.05377*.