Institute of Informatics Silesian University of Technology

Faculty of Automatic Control, Electronics and Computer Science

Department of Graphics, Computer Vision and Digital Systems

**Biologically Inspired Artificial Inteligence**

***Report***

*Cats and Dogs Image Classification*

*Monika Dudzińska*

Short introduction presenting the project topic.

The theme of the project is to classify images of cats and dogs using a neural network. Distinguishing images of dogs and cats is easy for humans, but it is particularly difficult to tell apart automatically. The aim of the project is to create a model that will classify the images as accurately as possible.

Analysis of the task:

1. Methodology

The following types of neural networks were compared during the research:

* Artificial Neural Networks (ANN)

ANN consists of 3 layers – input layer, hidden layer (the one that processes the inputs), and output layer. Inputs are processed only in the forward direction. Thanks to activation functions, ANN are capable of learning any nonlinear function. They can be used to solve problems related to: Tabular data, Image data, Text data. However, one of it’s main disadvantages is that it loses the spatial features (arrangement of pixels and the relationship between them in an image).

* Recurrent Neural Networks (RNN)

The main difference between RNN and ANN is that RNN has a recurrent connection on the hidden state. This looping constraint ensures that the input data contains sequential information.  
RNN are used to solve the problems related to: Time Series data, Text data, and Audio data.

* Convolution Neural Networks (CNN)

Although CNN also performs well on sequential inputs, it was introduced to solve problems related to image data. CNN automatically learns the filters, which help in extracting the relevant features from the input data by using the convolution operation. In opposite to ANN, CNN captures the spatial features from an image, what helps in identifying the object, the location of an object, and its relation with other objects in an image.  
The basic CNN structure is as follows: Convolution -> Pooling -> Convolution -> Pooling -> Fully Connected Layer -> Output

Since the project concerns image classification, convolutional neural network was chosen.

1. Datasets

Google search results for dataset, which contains images of cats and dogs, seem to be limited to almost only Asirra dataset provided by Microsoft Research. Since it is commonly used and met the project requirements (the size and availablity), I also decided to use it.   
  
This dataset was supposed to contain 25,000 images, but at the time I was choosing it, I wasnot aware that 54 of them will turn out to be corrupted. In order to keep the dataset balanced I had to remove 6 images of cats. This way the training set contained 24,900 images, including 12,450 images of dogs and 12,450 images of cats. The remaining 40 images were taken to create a testing set.

1. The tools and libraries

While doing the research I acknowledged that ones of the most common Python Machine Learning IDEs are:

* Spyder

This IDE is known for being simple, light-weight and easy to install. One of the prons is the Documentation Viewer that shows the documentation related to classes or functions in a project. Spyder is considered best for testing and development of scientific applications and scripts that use libraries such as SciPy, NumPy and Matplotlib.

* Geany

This is also a light-weighted IDE but as capable as any other IDE present out there. Although it has features that can come in handy and make writing code more convinient/confortable, there are no special prons that would draw our attention towards this environment.

* Rodeo

This Python IDE was built for the purpose of machine learning and data science. It uses IPython kernel. It lets users explore, compare and interact with the data frames and plots, however, it does not consist of code analysis, PEP 8, etc.

* PyCharm

PyCharm is most famous in the professional world for data science and conventional Python programming. It provides support for important libraries like Matplotlib, NumPy and Pandas. It also includes code completion, auto-indentation, runtime debugger, PEP-8 that enables writing neat codes. It has documentation viewer and video tutorials.

* JuPyter

It’s an open source platform that supports sharing live codes, and documents with equations and visualizations. It has Big Data integration within to help the data scientists.

The tool of my first choice was PyCharm since I was already familiar with it. Due to issues that occurred during implementation of a model I learnt that the hardware configuration of my machine was not proper for this IDE. Therefore, I switched to Google Colaboratory, in which it is not necessary to configure the environment and one can easily import libraries and datasets.

As we can read on google research page - “Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more.”   
Colab is especially well suited to machine learning, data analysis and education.

Deep learning libraries comparison:

|  | Keras | PyTorch | TensorFlow |
| --- | --- | --- | --- |
| API Level | High | Low | High and Low |
| Architecture | Simple, concise, readable | Complex, less readable | Not easy to use |
| Datasets | Smaller datasets | Large datasets, high performance | Large datasets, high performance |
| Debugging | Simple network, so debugging is not often needed | Good debugging capabilities | Difficult to conduct debugging |
| Does It Have Trained Models? | Yes | Yes | Yes |
| Popularity | Most popular | Third most popular | Second most popular |
| Speed | Slow, low performance | Fast, high-performance | Fast, high-performance |
| Written In | Python | Lua | C++, CUDA, Python |

source: https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article  
  
Keras can run on top of other open-source machine learning libraries such as TensorFlow. Considering all the above prons and cons I decided to use both TensorFlow and Keras.

Short summary of other used libraries:

| THE LIBRARY | THE REASON FOR USE |
| --- | --- |
| numpy | array operations |
| matplotlib | displaying the image |
| opencv | image operations |
| os | joining paths of directories |
| tqdm | progress bars |
| pickle | saving and loading data |
| time | tensorboard |

The visualization tool used in this project is TensorBoard provided with TensorFlow. It enables tracking experiment metrics like loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more.

Internal and external specification of the software solution.

Experiments:

Summary, overall conclusions, possible improvements, future work etc.

References – list of sources used during the work on the project.

https://www.kaggle.com/c/dogs-vs-cats

https://www.microsoft.com/en-us/download/confirmation.aspx?id=54765

https://www.thecrazyprogrammer.com/2017/11/best-python-machine-learning-ides.html

https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/

https://en.wikipedia.org/wiki/Convolutional\_neural\_network