UNIVERSITY OF ZAGREB
FACULTY ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS nu. 1382

Image Based Phylogenetic Classification

Vinko Kodžoman

Zagreb, travanj 2017.

Umjesto ove stranice umetnite izvornik Vašeg rada.

Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`*.*

Thank you...

CONTENTS

1. Introduction

Since the dawn of time, people have tried to explain their surroundings. Life is all around us in many forms, and as such people have tried to categorize it by keen observation, both through its visual and genetic features. Today, it is organised into a taxonomic hierarchy of eight major taxonomic ranks. The number of known species on Earth is in the millions and climbing every year. Great numbers of species make it difficult to classify species based on images and requires domain knowledge. Therefore, an algorithm with the capability to classify species on the filed or from an image using only the image itself could provide great benefits for field researches.

Machine learning allows computers the ability to learn without being explicitly programmed (Samuel). It, together with an incrase in avaiable quallity data (CIFAR, Imagenet) has yielded great results in the area of deep learning - a class of machine learning algorithms. Deep learning algorithm's accuracy scales with the amount of data used by the algorithm (referenca), that together with the improvements in hardware - mainly general purpose graphic units (GPUs) - has yielded significant perfomance gains in the last couple of years. One of the most rapidly advancing filed of deep learning is image recognition (Krizhevsky et al.; Simonyan i Zisserman; Szegedy et al.; He et al.) with new neural network architectures being developed almost at a yearly basis, the pefromance of deep neural networks on image recognition has achived results perviously tought impossible.

In this thesis I propose a solution for a scalable classification of species from images, based on convolution neural networks and recent modern deep learning techqniues.

2. Research context

To fully understand the depth of the image recognition using deep learning, we need a better understand of the underlying algorithms and methods in machine learning, as well as fundemental terms and concepts. In the next section, an introduction of basic terms is given, followed by a detailed explenation of fundemental machine learning algorithms.

2.1. Definitions and notation

2.1.1. Image representation

Matrix is a rectangular array of numberes. It is used because some numbers are naturally represented as matricies. Matrix A with m rows and n columns often writtens as $m \times n$ has $m * n$ elements and is denoted as $A_{m,n}$. Elements are denoted as $a_{i,j}$ where i and j corespond to row and column number respectivly, as shown in 2.1.

$$A_{m,n} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{2.1}$$

Each image is represented as a 3 dimensional matrix. One pixel in the image represent a single element in the matrix and as images have multiple channels (RGB) each channel is a 2 dimensional matrix. Image I denoted as $I_{k,m,n}$ where $k \in [0, 2]$ represent the channel - red, green or blue - and $m, n \in [0, 255]$ represent the pixels in a particular channel as 2 dimensional matricies. Figure 2.1 shows a representation of an image as a 3 dimensional matrix where each pixel is denote as $I_{k,m,n}$.

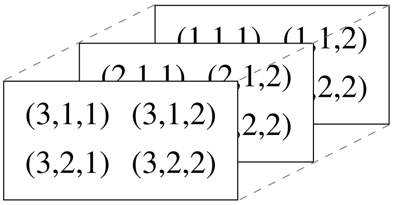


Figure 2.1: RGB image with 4 pixels represented as a 3 dimensional matrix

2.1.2. Gradient

A gradient is a generalization of the derivative in multi-variable space and as such it is represented as a vector. Like the derivative, it represents the slope of the tanget of the graph of the function. Therefore, it points in the direction of the greatest rate of incrase of the function. Gradients are widely used in optimization theory as they allow the parameters to shift in a direction which will minimize or maximize a given function. In machine learning the function we want to minimize will be the loss function, which we will define in further chapters in more detail. Gradient of f is denoted as ∇f, where every component of ∇f is a partial derivavate of f, denoted as $\frac{\partial f}{\partial x}\vec{e}$. Notice that gradient components are vectors denoted as $\vec{e}$. Every vector is written as a bolded letter. The gradient for a n dimensional space is defined in 2.2.

$$\nabla f = \frac{\partial f}{\partial x_1}\vec{e_1} + \ldots + \frac{\partial f}{\partial x_n}\vec{e_n} \tag{2.2}$$

2.1.3. Activation functions

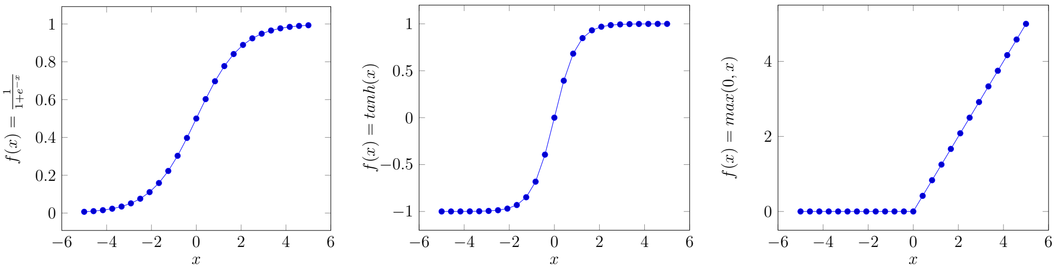
Machine learning models use nonlinear functions to gain more capacity - expressiveness . The most popular nonlinear functions are $sigmoid$, $tanh$, $relu$. All nonlinear functions have to have easy to compute gradients, as they are compunted on parameters in order to reduce loss as explained above.

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

$$tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2}} \tag{2.4}$$

$$relu(x) = max(0, x) \tag{2.5}$$

The order of nonlinear functions is given in order of their discoveries. Today relu is used the most, since it solves the problem of vanishing gradiens for very deep neural networks, this does not apply to all network types. Reccurent neural networks (RNN)



(a) *simgoid* **(b)** *tanh* **(c)** *relu*

Figure 2.2: Nonliear activation functions

are a class of neural networks that often use $tanh$ as it is better suited for the particular reccurent architecture.

2.1.4. Metrics

In order to compare different models a set of metrics is employed. Accuracy which gives the accuracy of a model, it is often used on balance datasets (2.14). The problem with unbalanced datasets can be easily explained with a short example. Image having 2 klasses $K = \{dog, cat\}$ and there are a total of 100 images in the dataset, of which only 2 are dogs. The model if optimized for accuracy might say the whole dataset is cats which will yeild an accuracy of 98%. To solve the previous problem, more metrics where introduced for the task of classification; precision (2.15), recall (2.8) and F1 score (2.9). Precision - positive predictive value - is defined as a fraction of retrived instances that are relevant. Recall - sensitivity - is a fraction of relevant instances that are retreived. In order to represent the perfromance of a model as a single variable F1 score was introduced, it represent a harmonic mean of accuracy and precision.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.6}$$

$$Precision = \frac{tp}{tp + fp} \tag{2.7}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.8}$$

$$F1score = 2 * \frac{precision * recall}{precision + recall} \tag{2.9}$$

Classification results are often represented as a confusion matrix, also known as an error matrix. It is a performance visualisation of a classification model - classifier. To

Table 2.1: Confusion matrix

	prediciton positive	**prediction** negative
actual positive	True Positive (TP)	False Positive (FP)
actual negative	False Negative (FN)	True Negative (TN)

build the classification matrix, conditions of the experiment must be labeled as positive and negative. Using the cats and dogs example from before and marking the cats and a positive and dogs as a negative class. Doing so creates a $2x2$ matrix of actual and predicted values as shown in table 2.1.

2.1.5. Data

The input data of the machine learning algorithm is labled as D, and it consists of X and y_t, where X is one input data (an image in our case) and y_t is the true lable of the picture - species' name. Written formally the whole input dataset is represented as $D = \{X^i, y^i\}_{i=1}^N$, where i is the i-th data point and N in the number of data points. Prediction of the algorithm is labeled as y_p.

The input data set is usually split into two datasets called the *training* a *test* dataset. The training dataset is used to optimize them models parameters while the test data is used to evaluate the model's performance. Sometimes the training dataset is split further into training and *validation* where the validation dataset is used to tune the models *hyperparameters*. Hyperparameters are parameters that do not belong to the model but non the less effect the model's performance. Depth of the neural network is a hyperparameter and will be discussed in later chater in more detail.

2.2. Machine learning

As said in the Introduction chapter, machine learning allows computers the ability to learn. Giving data to a machine learning algorithm - model - allows it to find patterns withing the dataset and to infere. The function that maps the input X to y_p is called a *hypotesis* and is denoted as h (2.10). The hypotesis $h(X; \vec{\theta})$ is parametrized with $\vec{\theta}$ - model's parameters.

$$h(X; \vec{\theta}) : X \to y \tag{2.10}$$

The model is defined as a ste of hypotesis H, $h \in H$. Machine learing is the search of the best hypotesis h from the hypotesis space H - typical optimization problem. The algorithm tries to minimize the emircal error function $E(h|D)$ - loss function. The error indicates the accuracy of the hypotesis and is called empirical because it is computed on D. Therefore, every machine learning algorithm is defined with the model (2.11), error function (2.12) and the optimization method (2.13).

$$H = \{h(X;\vec{\theta})\}_{\theta} \tag{2.11}$$

$$E(h|D) = \frac{1}{N}\sum_{i=1}^{N} I\{h(X^{i}) \neq y^{i}\} \tag{2.12}$$

$$\theta^{*} = argmin_{\theta}E(\theta|D) \tag{2.13}$$

Machine learning algorithms are devided into groups depending on the task, the groups are classification and regression. Each can be represented as a result of $h(X;\vec{\theta})$. Classification hypotesis takes the input X and returns a klass k, example of this method would be image classification. Regression hypotesis takes the input X and returns a number, for example predicting house prices.

$$Regression \equiv h(X;\vec{\theta}) : X \to y, y \in \mathbb{R} \tag{2.14}$$

$$Classificatio \equiv h(X;\vec{\theta}) : X \to y, y \in K = \{k_0, ..., k_n\} \tag{2.15}$$

2.2.1. Supervised and unsupervised learning

Supervised vs unsupervised.

2.2.2. Models

2.2.3. Model selection

2.3. Deep learning

GPUs

2.3.1. Feedforward Neural Networks

2.3.2. Convolutional Neural Networks

2.3.3. Backpropagation

2.3.4. Vanishing Gradient

2.3.5. Batch Normalization

2.3.6. Data Augmentation

3. TaxNet

Let’s hope it is any good.

3.1. Implementation

4. Dataset

4.0.1. ImageNet

5. Results

Graphs graphs graphs...

6. Conclusion

Zaključak.

BIBLIOGRAPHY

Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Deep residual learning for image recognition. U *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, stranice 770–778. URL http://www.cv-foundation.org/openaccess/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Alex Krizhevsky, Ilya Sutskever, i Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. U *Advances in neural information processing systems*, stranice 1097–1105. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-network

A. L. Samuel. Some studies in machine learning using the game of checkers. 3(3): 210–229. ISSN 0018-8646. doi: 10.1147/rd.33.0210.

Karen Simonyan i Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. URL https://arxiv.org/abs/1409.1556.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, i Andrew Rabinovich. Going deeper with convolutions. U *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, stranice 1–9. URL http://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html.

Image Based Phylogenetic Classification

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.