DELFT UNIVERSITY OF TECHNOLOGY

Artificial Neural Networks

PROJECT PROPOSAL

Authors:
Michiel Bongaerts
Marjolein Nanninga
Maniek Santokhi
Tung Phan

March 25, 2015

Contents

1. Introduction

Mapping the world around us has always been a human endeavour to advance economical output. A better understanding of the places around us makes for more efficient travelling and exploitation of the land. However, it has always been a very slow and tedious process to produce these maps, something technology has not changed just yet.
A new opportunity has arisen with the arrival of satellite imagery and an ever increasing amount of computational power. An opportunity where this mapping can be done automatically so that this tedious and slow job can be processed even more quickly and perhaps more accurately. It is with this in mind we further analyze any possibilities.

1.1 Satellite image classification with Neural Networks

The project facilitates the following: analyzing satellite images and classifying each pixel in different classes such as roads, cropland, forest, water and cities. Since Neural Networks (NN) can be used for pattern recognition [1], we came up with the idea to try to develop our own `Google Maps`. The general approach is to use supervised learning with labeled satellite images. Making use of cross-validation we can determine the accuracy of our model. If a certain accuracy is reached we will present new images to the network and look into the performance of the model.

The implementation of the system will be done in Python. A programming library called PyBrain will be used to model and experiment with [2]. The flow is as follows: the user provides a satellite image, the system classifies each pixel and any subsequent steps can be filled in with extra features.

1.2 Extra features

Vector map as output

Depending on how fast we will be able to attain a sufficient performance, we would like to add some extra features. We want to convert the output image to a vector graphical representation which can be adjusted by the user. In this way we can for example adjust sloppy, sinuous, roads and make them straighter. This output could in its turn be fed into the Neural Network again such that the classification can even be more in tune with what is expected from a map (straighten up roads and make them neater).

Behavioral analyses

If even more time is available, we could add the following feature: simulate data generated by users of the developed map, 'the `Google Map` users', and according to their behavior. So for example the speed of the people on certain roads can be used to distinguish between walking tracks, biking tracks and motorways. Also we could use this data to improve the classified images to adjust roads and to find out about roads the automatic pattern recognition missed out on. Lastly, using this simulated data the optimum path people could take when they want to commute from point A to point B can be computed. So certain post map-creation processes can be thought of as an addition to the existing proposal.

2. Literature review

In the past decade a lot of research has been done to the use of Neural Networks for pattern recognition. Extensive documentation exists on the different methods available, the theoretical explanation and practical utilities [1] [5]. Moreover, in different programming languages Pattern Recognition toolboxes based on NN exist, proving its use in real world applications [6] [4]. These could therefore also be used in our project.

Next to this general approach of implementing pattern recognition based on NN, we read a paper that focuses more specifically on what we are trying to achieve: segmenting objects from images and classifying them in certain categories [5]. In this research, improvements are made on the conventional implementation of Convolutional Neural Networks (CNN) in pixel-to-pixel classification [5]. Common implementation of CNN involves convolutional kernels and kernels to extract features from images. In general, the patch-by-patch approach is used where patches are selected from images. In the following steps different convolution and pooling kernels operate on each patch to extract specific features, called feature maps. These feature maps can be used to classify the specific patch.

Lastly, research had been done to specifically classification of high-resolution satellite images into asphalt, vegetation, building and bare soil [3]. In this research multilayer perceptron networks are used as a tool for classification of multispectral images.

So in terms of already done research the literature shows possibilities. This implies a certain feasibility to our problem we would like to solve. More research will be conducted while actually tackling the problem. For now this seems like a good start.

3. Research questions

In this project we want to address the following questions:

1. What exactly should be used as the input vector: RGB values of single pixels, RGB values averaged over patches or more sophisticated data as multispectral images?
2. What activation function has the best performance for this purpose?
3. Are Convolutional Neural Networks necessary for classification, or do ordinary multilayer perceptrons suffice?
4. What kind of architecture balances best between accuracy and computation time:
   (a) What would be the minimal amount of hidden layers to give good classification?
   (b) How many nodes should be used?
   (c) What parameters should be learnt by the model: could we use an Extreme Learning Machine?
5. What accuracy can be achieved using Neural Networks for pixel-to-pixel classification? And what accuracy suffices to obtain the objectives?
6. Could we increase performance by using edge-based or other segmentation procedures to post-process the classified images?
7. Does this Neural Network approach to map making complement contemporary methods of map making?

4. Approach

Our approach to the project is as follows:

1. From literature it is known that Neural Networks can be used to perform pixel-to-pixel classification. The most common approach uses the Convolutional Neural Networks. This approach works with multiple layers of NN's. From each image we take multiple different CNN's which are just the convolutional images of the input image. In this way we can extract features from the image in interest. This process can be repeated creating new CNN's which are decreased in size since convolution reduces the size of the (convoluted) images. Other operation may be included such as pooling layers. Eventually, we end up with many output nodes which can be combined in such a way that we return a classified image of our input. In the first stage we will investigate these Convolutional Neural Networks since this approach looks promising on first sight. Next, we try to determine a working architecture. In this stage we might also investigate other (similar) approaches.

2. The second stage consist of the implementation of the architecture using Python as our main programming language. This might involve the use of PyBrain depending of the complexity of the CNN architecture. PyBrain can also be used for validation.

3. In the third stage we will train the Network. We will use satellite images with its labeled version to train the Network. In the beginning we will consider a few classes which can be extended in future versions.

4. When the Network is trained it must be capable of classifying new satellite images. This stage include the validations of the network. Furthermore, we will adjust the Network to solve the errors.

5. Extra features can be implemented in this stage. This involves the Vector Graphical map where users can adjust the labeled maps and/or the Behavior analyses described in the Introduction.

5. Time schedule

Week	Description of work	Deadline	Labour (hours)
13	Reading literature	Project proposal	16
14	Reading literature + developing architecture	-	24
15	Reading literature + developing architecture	Discussion	24
16	Reading literature + developing architecture	Updated project proposal (if needed)	24
17	Implementation	-	40
18	Implementation	-	40
19	Implementation	-	40
20	Implementation	-	40
21	Implementation	-	40
22	Implementation	-	40
23	Implementation + preparation for demo	-	40
24	Preparation for demo	Demo	30
25	Project finalization	-	30
26	Project finalization	Project report + individual document	40
			Total: 468

Bibliography

[1] Christopher M Bishop et al. Neural networks for pattern recognition. 1995.

[2] CogBotLab and Idsia. *Pybrain, The Python Machine Learning Library*. http://pybrain.org/docs/.

[3] Fabio Del Frate, Fabio Pacifici, Giovanni Schiavon, and Chiara Solimini. Use of neural networks for automatic classification from high-resolution images. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(4):800–809, 2007.

[4] M. et al. Hanke. *PyMVPA: A Python toolbox for multivariate pattern analysis of fMRI data*, 2009.

[5] Hongsheng Li, Rui Zhao, and Xiaogang Wang. Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification. *CoRR*, abs/1412.4526, 2014.

[6] Mathworks. *Classify Patterns with a Neural Network*. http://nl.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html.