Astroinformatics
Proceedings IAU Symposium No. 325, 2016
M. Brescia, S.G. Djorgovski, E. Feigelson,
G. Longo & S. Cavuoti, eds.

doi:10.1017/S1743921317003520

Classification of galaxy type from images using Microsoft R Server

Andrie de Vries

Microsoft
Microsoft UK Ltd, Algorithms and Data Science,
2 Kingdom Street, Paddington, London, UK W2 6BD
email: adevries@microsoft.com

Abstract. Many astronomers working in the field of AstroInformatics write code as part of their work. Although the programming language of choice is Python, a small number (8%) use R. R has its specific strengths in the domain of statistics, and is often viewed as limited in the size of data it can handle. However, Microsoft R Server is a product that removes these limitations by being able to process much larger amounts of data. I present some highlights of R Server, by illustrating how to fit a convolutional neural network using R. The specific task is to classify galaxies, using only images extracted from the Sloan Digital Skyserver.

Keywords. methods: analytical, techniques: image processing

1. Introduction

Our universe contains an estimated 2 trillion galaxies (Nasa website). The shape of a galaxy, e.g. spiral vs. elliptical, tells us something about the age and history of that galaxy. Traditionally, this classification of shape could only be done by a professional astronomer. However, with the availability of large numbers of images, combined with the ubiquity of Internet access, it is possible for the public to help with this classification task.

The GalaxyZoo project (GalaxyZoo website) was one of the first citizen data science projects to engage the public in large scale participation. With the help of a web site and carefully designed questionnaire, people could participate in answering questions and ultimately helping to classify more than 240 thousand galaxies.

2. Computer vision

With a large corpus of labeled galaxy images available from GalaxyZoo, the next question is whether computer algorithms can automatically classify new images.

The state of the art in computer vision techniques is deep convolutional neural networks. IN a convolutional network a kernel gets convoluted over the input image, in the process generating features that can be used by the fully connected layer of the neural net.

In the process very low level features, e.g. lines, edges and blobs combine into a hierarchy of intermediate features. These intermediate features start to identify higher level features such as parts of faces or parts of cars (Lee:2009:CDB:1553374.1553453), see figure 1.

The input to a convolution layer is a pixel map of fixed size. From this pixel map, the convolution process generates features that ultimately influence the output class of the net.

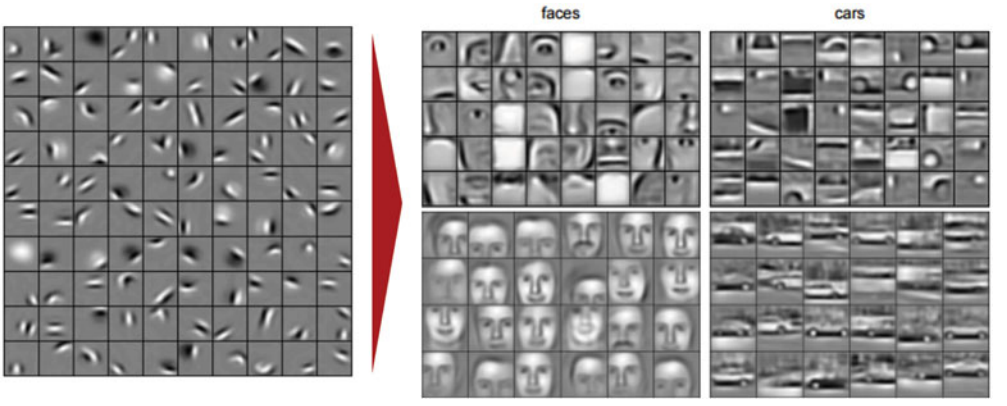

Figure 1. Low-level features combine in a hierarchy of higher level features.

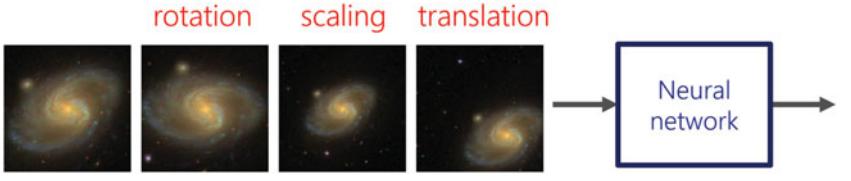

Figure 2. Tricky cases.

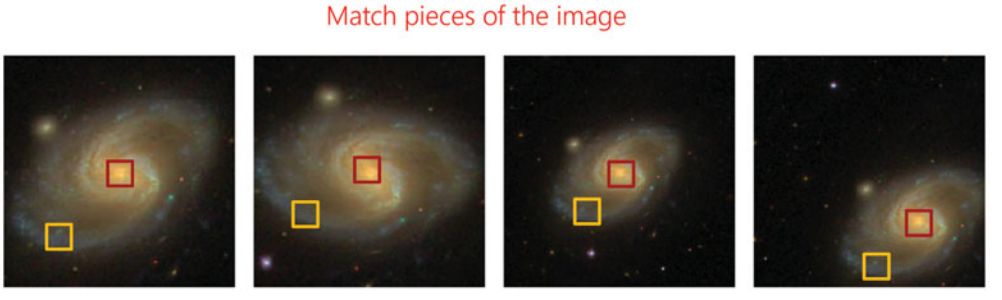

Figure 3. Matching pieces of the image.

However, simply mapping every pixel to a neuron in the network will not work very well. Rotation, scaling and translation of an image do not change the class of the image, but makes it difficult for the network to correctly classify the image (figure 2).

The intuitive solution to this problem is to find common features across the entire image. For example, in the case of galaxy classification, you need to find pieces of the image that resemble the galaxy center, or pieces that resemble the galaxy edge (figure 3)

Convolution of the input matrix is one way of dealing with the translation and scaling problem. By convolving the image, a kernel is processed over small squares of the input image. The result of this convolution gets passed to the next layer (figure 4).

This process of convolution generates features that the later layers of the network can use in the computation of the different classes.

This simplified view of convolution only explains the actual convolution process. The layers following convolution will typically include adding a non-linear transformation, for example a rectified linear (ReLU) layer, a batch normalization layer and frequently a maximum pooling layer.

https://doi.org/10.1017/S1743921317003520 Published online by Cambridge University Press

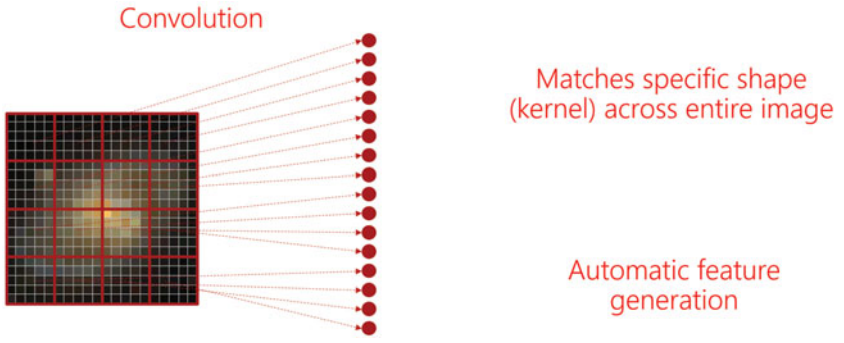

Figure 4. Convolution.

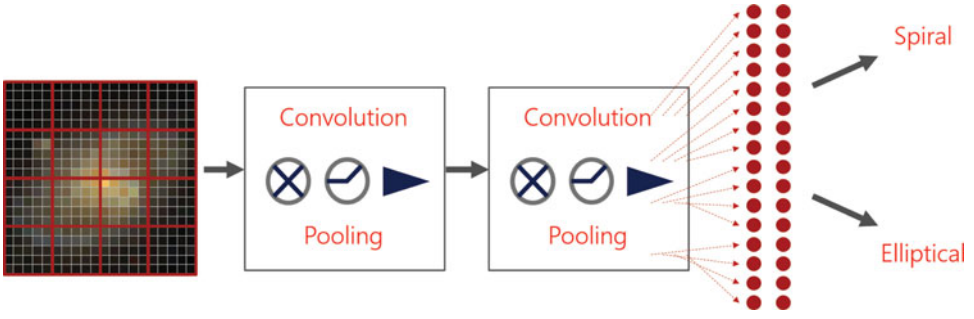

Figure 5. A deep net is the term for a neural network with multiple layers.

The term "deep neural network", or simply "deep net" refers to a network that has many of these hidden layers, added sequentially (figure 6).

3. Microsoft R Server

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS (R core Team)

Microsoft R Server is a enhanced, commercial distribution of R, provided by Microsoft. R Server handles large data sizes and computation on multiple nodes. Its big-data capable architecture includes algorithms optimised for fast parallel execution, a parallel computing framework for managing compute resources and connections to data sources (R Server website).

The R Server getting started guide (R Server getting started website) contains a comparison of the components of R server, as well as an overview of supported platforms, including Hadoop, SQL Server, Teradata and Linux.

Although R Server contains packages for dealing with large data, it doesn't change any of the base R features. In this way, it is completely compatible with open source R, and all CRAN packages will run as intended on R Server, in the same way they run on open source R.

Starting with R Server v9, a new package containing advanced analytics algorithms gets bundled with the product. This package is called *MicrosoftML* and contains new algorithms for:

- Fast linear learner (SDCA)
- Fast trees
- Fast forests

https://doi.org/10.1017/S1743921317003520 Published online by Cambridge University Press

- One-class Support Vector Machine
- Regularized logistic regression (L1 and L2)
- Neural networks

4. Training the model

It is this last item, the *neural net* capability, that allows you to train a deep convolution network using R Server.

The outline of the code is quite simple, as illustrated by this snippet:

```
library(RevoScaleR)
library(MicrosoftML)
model <- mxNeuralNet(
        formula,
        data = galaxy.images,
        netDefinition = netDefinition,
        type = ''multiClass''
        acceleration = ''gpu'',
        miniBatchSize = 32
        initWtsDiameter = 0.1,
        numIterations = 50
)
```

Note that the algorithm supports training on fast graphical processing unit (GPU) machines. In practise, we see a 10x multiplier in speed and performance when using GPU, as compared to standard CPU operations.

One important piece of the code is to specify the deep layer structure, the network definition. In R Server, you can specify this network definition using a specification language called NET sharp.

The full specification of this language is available at `https://azure.microsoft.com/en-gb/documentation/articles/machine-learning-azure-ml-netsharp-reference-guide/` .

To train a simple network to classify galaxies, the network definition file may look like the snippet below.

Note that this example illustrates:

- An input layer, consisting of an image of 50x50 pixels, with 3 channels (red, green and blue)
- A convolution layer with kernel size 5x5, with 64 different kernel maps
- A batch normalization layer
- A maximum pooling layer
- Two fully connected layers
- An output layer with a softmax activation function

```
input pixels [3, 50, 50];

hidden conv1 [64, 24, 24] rlinear from pixels convolve {
        KernelShape = [3, 5, 5];
        Stride = [1, 2, 2];
        MapCount = 64;
}
```

https://doi.org/10.1017/S1743921317003520 Published online by Cambridge University Press

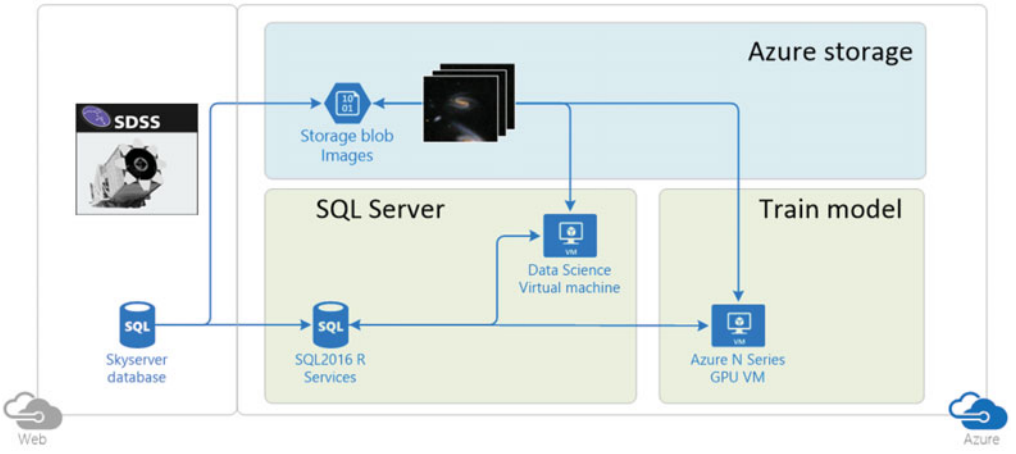

Figure 6. A deep net is the term for a neural network with multiple layers.

```
hidden rnorm1 [64, 11, 11] from conv1 response norm {
        KernelShape = [1, 4, 4];
        Stride = [1, 2, 2];
}

hidden pool1 [64, 9, 9] from rnorm1 max pool {
        KernelShape = [1, 3, 3];
}

hidden hid1 [256] rlinear from pool1 all;
hidden hid2 [256] rlinear from hid1 all;
output Class [13] softmax from hid2 all;
```

To train the model, I used a GPU enabled virtual machine on the Microsoft Azure cloud. A second virtual machine stored the labeled classes in SQL Server. In this way I could use the expensive GPU machine for high performance training, but a standard machine for predicting new classes.

To train the model I used 100 thousand images, and augmented this training set by rotating each image twice, for a total of 300K input images. With a model of 8 layers deep (176K model weights to compute), total training time was 1.8 hours. The model achieved 88% overall accuracy on the training data, and 55% on the test data.

This indicates that convolution networks have potential for this application, although clearly there is scope for improvement.

5. Conclusion

In this paper I illustrated the use of the R language, specifically Microsoft R Server to process large amounts of data. I used an example of classifying galaxy images into spiral vs elliptical, using a convolutional neural network. This functionality is available in Microsoft R Server, using the MicrosoftML package, available in version 9.0 and above.

References

Galaxyzoo web site, https://www.galaxyzoo.org/. Accessed: 2016-11-29.

https://doi.org/10.1017/S1743921317003520 Published online by Cambridge University Press

Nasa, *Hubble reveals observable universe contains 10 times more galaxies than previously thought*. https://www.nasa.gov/feature/goddard/2016/hubble-reveals-observable-universe-contains-10-times-more-galaxies-than-previously-thought Accessed: 2016-11-29.

Microsoft, Microsoft r server website https://msdn.microsoft.com/en-us/microsoft-r/. Accessed: 2016-11-29.

Microsoft, R server getting started. https://msdn.microsoft.com/en-us/microsoft-r/microsoft-r-getting-started. Accessed: 2016-11-29.

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng., Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM.

R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.