Mika Hämäläinen

Summary of *Parsing Natural Scenes and Natural Language with Recursive Neural Networks*[1]

In their paper, Socher et al. introduce a new general method for extracting semantic information from images and parsing natural language syntax using recursive neural networks (RNNs). Both natural language sentences and scene images are recursive in nature: sentences can have relative clauses inside of each other to an infinite depth and a picture of a car consists of windows, wheels, doors, etc. which in their turn consist of smaller structures. The existence of this similar kind of recursion is indeed what makes RNNs a general tool for tackling both problems.

However the main focus of the paper is on scene understanding rather than in natural language processing (NLP). In the task for scene understanding, images are oversegmented into smaller regions which represent parts of objects in the scene or background. Vision features are obtained from these segments, which makes it possible for a neural network to map the segments' features into a semantic space. These features form an input for the RNN, which in its turn, calculates a score indicating the need to merge neighboring regions into larger ones. When the merging has been done, the RNN also produces a new semantic feature representation for the new larger region along with a label for its classification. A label can, for instance, be *building* or *street*. The merging decisions define a tree structure in which the higher nodes represent larger elements of the image, such as handle, door and then car.

The RNN approach, according to its creators, has a long list of contributions to the current state of its related scientific field. It's the first deep learning method producing state-of-the-art results in scene understanding. The hierarchical tree structures predicted by the algorithm outperform those of every other method. The scene classification part of the algorithm is better than state-of-the-art methods such as Gist descriptors. The algorithm is not only better than its alternatives, but it also applies to a seemingly completely different task of NLP.

Scene understanding is a problem that consists of three different subproblems: annotation, segmentation and classification. The term annotation refers to understanding which objects are present in the scene under analysis, whereas segmentation provides insight in the location of these objects in the scene. Finally, classification is used to determine the general type of the scene. However, all the existing approaches focusing on classification have problems in obtaining a deeper understanding of the objects in the scene.

[1] Socher, R., Lin, C. C., Manning, C., & Ng, A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 129-136).

What makes the RNN's segmentation more powerful than that of the existing systems is that it makes super segments recursively learning their representations automatically.

Syntactic parsing is considered a crucial part of NLP because its importance in dovetailing the linguistic expression and the intended meaning. RNN represents phrases in a vector space of features learning at the same time to parse its input. Both semantic and syntactic information is captured by the feature representations. This learned data can be used to accurately parse a sentence and to find similarities between phrases and sentences.

Deep learning can find lower dimensional representations for input images of the same sizes. The way RNN does deep learning differs from previous attempts on this field. Previously raw or whitened pixels have been used to learn features form images. The RNN, however, uses off-the-shelf vision features by using the smaller segments calculated from the oversegmented image of the entire scene. The other fundamental difference to other algorithms is that the hierarchical structure is not built upon convolutional and max-pooling layers, but instead, is constructed in a recursive manner applying the same network to merged segments categorizing them with a label of their respective semantic categories.

The RNN architecture is evaluated in its performance on visual and NLP tasks. There are three parameters one can modify in the RNN: the size of the hidden layer, the penalization for incorrect parsing decisions and ultimately the regularization parameter. Despite of this configurability, the RNN has been tested to be robust enough that altering these parameters only affects on the outcome by a few percent.

For the vision experiments, the Stanford background dataset is selected, while for the NLP testing, the Wall Street Journal section of the Penn Treebank is used. The Stanford data set has three types of scenes: city, countryside and sea-side. The accuracy of the RNN in scene classification is 88.1 % using this dataset, which is better than the 84.0 % of the state-of-the-art Gist classification.

Nearest neighbor super segments are visualized in order to show that the classification and capturing of features is meaningful even in the higher levels of the tree hierarchy. This is done by firstly passing all the test set images through the trained RNN and then finding subtrees the nodes of which have same labels and saving the vector representations of the top node of each subtree. These representations can be used to find the nearest neighbors across all the images and all the subtrees. When these neighbors are outputted, one can clearly see that they are most of the time related: pictures of roads, grass and cars get outputted nicely as each other's nearest neighbors.

In the syntactical parsing task, the F-measure of the results is 90.29 % while the same measure for an existing parser, namely that of Berkeley, is 91.63 %, which means that despite of the good result, the RNN doesn't quite overshadow the state-of-the-art. A key difference to existing parsers is that the RNN one doesn’t provide a parent node with syntactic information about its children. The semantic relations are captured by nearest neighbor phrases. This is achieved by embedding complete sentences from the WSJ dataset in the syntactico-semantic feature space and then comparing the features of the new sentences to those stored in the feature space.