

welove.ai | ai blog | (Part 2) Automated Vision with Convolut...

MARTIN BÄUMLER | 2017-03-31

(PART 2) AUTOMATED VISION WITH CONVOLUTIONAL NEURAL NETWORKS



Or: The right filter makes the picture

During the past five years, state-of-the-art computer vision systems have started to approach human performance. This is particularly due to a central methodology called the DCNN (Deep Convolutional Neural Network).

The fundamental idea behind deep neural networks is that features are no longer created manually and then combined with a statistical learning method. Instead, the entire system is trained “end-to-end” and during this process independently learns a hierarchy of features that build on one another depending on the respective application. It's an approach similar to the human visual cortex where increasingly complex sample recognisers are switched on one after the other along the way.

But how is it possible to learn this hierarchy of features? DCNNs comprise several layers in which so-called filters determine the similarity of the input (the following section describes what these structures look like when the network learns them). The individual filters only record a very small section of the image, but they are applied to the entire image to be able to recognise an image characteristic everywhere. You can think of the results of these operations as another series of images, one per filter per run. Large numeric values in this image mean a very high similarity with the sample. These interim results are then further processed by the following layers of the network using the same sample.

Learned and not defined

Those filters are not defined by the developer, but are learned by the network in a training process. In mathematical terms, a multi-layer network is only a series of simple functions. Each filter layer converts the output from the previous layer into a series of numbers. The last layer in the network must generate a number which corresponds to our item category “face.” We can even quantify the error in the network output for a series of images for which we already know whether they show a face. “Learning” means we just have to keep turning the filter layers so that the network error becomes smaller. It also means that we have to derive the network function, or error, using the filter parameters. As anyone who has taken Analysis 101 knows, the chain rule is the way to go to derive a nested function, but in practice, we don't need to use pen and paper because there is the backpropagation algorithm which cleverly determines how the filters should be changed to generate a better network output. First, the network output is calculated for a small sub-quantity of the image data (a batch) and the error, or the difference to the desired output, is quantified. The derivation is then used to change the filters so that they will produce a smaller error for this subset of data. This process is repeated using a new batch until the network output hardly changes.

This optimisation process is stochastic – depending on the order in which the training data is taken, the filters change in one direction or the other. That approach has a crucial advantage to deal with the fact that networks have a hard time memorising data. This so-called overfitting is often a major problem when developing pattern recognition systems. The data used to train the system is generally only a small subset of the data that will be used in the overall system. Therefore it's better if the system can generalise using unknown data. The traditional approach to combat overfitting is to reduce the model's capacity so that it's no longer able to memorise the training data. However, this also creates the problem that the system is often no longer able to model the problem sufficiently, leading to sub-par results. DCNNs suffer from this problem, too. Latest research results, though, have shown that networks can still generalise if they've been able to memorise the training data, thanks to stochastic optimisation and their hierarchical structure (source: [here](#)).

Revolution – with a caveat

DCNNs have revolutionised the field of computer vision. Classification problems that up to a few years ago were considered impossible to crack can now be solved by enthusiasts on their home computers. Such networks are used in [self-driving cars](#) and also in the [diagnosis and treatment of cancer](#). In addition, they are a key component in [deep reinforcement learning](#) and with machine learning systems have been able to beat, for the first time, the world's best players of the Chinese board game GO; they're also able to learn to play [computer games such as space invaders](#) using just the screen output. DCNNs are not only used to process image data but are also essential components in state-of-the-art [translation](#) or [speech synthesis systems](#).

Despite all the successes, though, training a DCNN still requires huge amounts of data. Networks are excellent tools to map the world's structure, but DCNNs alone can't make robots intelligent. They can reduce the variability of information to a relevant subset, and thus make robust decisions. However, a DCNN's output depends on its training, and this process has to be ongoing. No human would fall for a bluffing poker player's wink a second time.

Kommentare

Add a comment

Name*

Website

Comment

☐ Notify me of new comments by e-mail

E-mail (not published)*

Please calculate 5 plus 5.*

SUBMIT COMMENT

SHARE:  

GO BACK



AI NEWS 2018-05-28

TELEKOM DEFINES ITS OWN POLICY ...



AI NEWS 2018-05-01

AUTONOMOUS DRIVING: DRIVING WI...



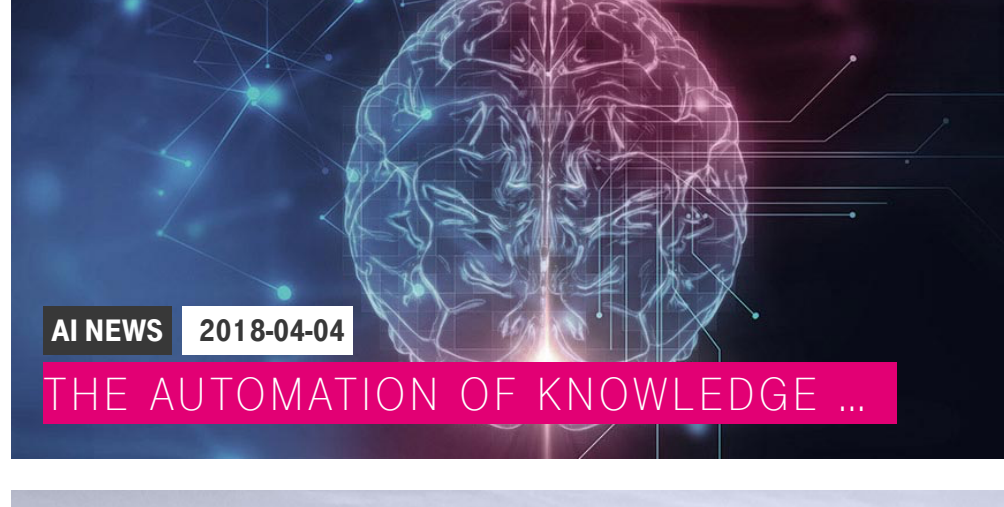
AI NEWS 2018-04-18

KNOWLEDGE IS POWER - BUSINESS ...



AI NEWS 2018-04-13

THE HISTORY OF ARTIFICIAL INTELLI...



AI NEWS 2018-04-04

THE AUTOMATION OF KNOWLEDGE ...



AI NEWS MEET&GREET 2018-02-06

EVENTS



AI NEWS STEFFAN HEUER | 2018-01-23

SILICON VALLEY LOVES AI: REPORT ...



AI NEWS 2018-01-22

FROM HAUTE COUTURE TO THE BAR...