

Deep Spreadsheets with ExcelNet

David Fouhey
Carnegie Mellon University
Pittsburgh, PA
dfouhey@cmu.edu

Daniel Maturana
Carnegie Mellon University
Pittsburgh, PA
dimatura@cmu.edu

	Caffe/Theano/MCN/etc	EXCELNET
Nerd coding stuff	☹	
Commie open source license	☹	
Backpropagation	☺	
Synergy		☺
Enterprise ready		☺
Turnkey solution		☺
WYSIWYG Weight Editing		☺

Table 1. EXCELNET versus lesser platforms.

As summarized in Table 1, this approach has many advantages over other complicated solutions like Caffe [2], Theano [1], or MatConvNet [4]. EXCELNET does not have backpropagation, but it is not needed, because the interface puts the weights at the tip of the user’s fingers! To help you get started though, we provide some initial weights using a strategy similar to the “pre-training” process in common use: we build a CNN in MatConvNet [4] that recognizes digits and transfer these weights to an Excel spreadsheet. We then allocate activation maps in other spreadsheets and define these activations using Excel’s powerful functions. Once done, Excel performs all memory management and calculations.

The advantage is that the power of Deep Learning is now in *your* hands. There is no need for worrying about CUDA or BLAS: Excel’s proven and battle-tested numerical codebase handles it for you! There is no need to worry about platforms: Excel provides the necessary abstractions, and you can seamlessly transition between Windows, Linux, and Mac! This enables the ultimate in computer vision deployment flexibility: you can develop on a Mac, and deploy simultaneously to client and cloud devices.

We show screenshots of EXCELNET in action in Figure 1. Figure 1(a) shows the i/o layer sheet. Here, you can input new data for the CNN to classify. EXCELNET takes care of the rest and gives you the answer, both in terms of a final probability as well as the most likely answer. Figure 1(b) shows extracting activations. Typically, this requires fussing with file formats, editing .prototxts, or writing code. EXCELNET enables you to execute your vision for Deep Learning without learning all this useless nerd information. *You’re in control.*

SELLING POINTS OF EXCELNET

EXCELNET offers the ultimate in Deep Learning abilities. We see a number of unique advantages.

Proven numerical abstraction layer: Worried about the correctness of your new app? Excel is the most trusted numerical computing platform: BLAS and ATLAS are used for fluid simulations and other underwater-basket-weaving-like endeavours; Excel is used for finance. By building on a solid foundation, we provide unsurpassed guarantees in terms of algorithmic correctness.

Enterprise-grade security: Worried about your autonomous driving platform in EXCELNET getting into the wrong hands when you leave your phone in a bar in SF? Don’t sweat it – EXCELNET is compatible with the most secure forms of document protection from Microsoft.

ABSTRACT

We introduce EXCELNET, the premier solution for Deep Learning in Excel¹. Visit deepexcel.net to try it today!

Author Keywords

Deep; neural; Excel; spreadsheet; Convolutional Neural Network

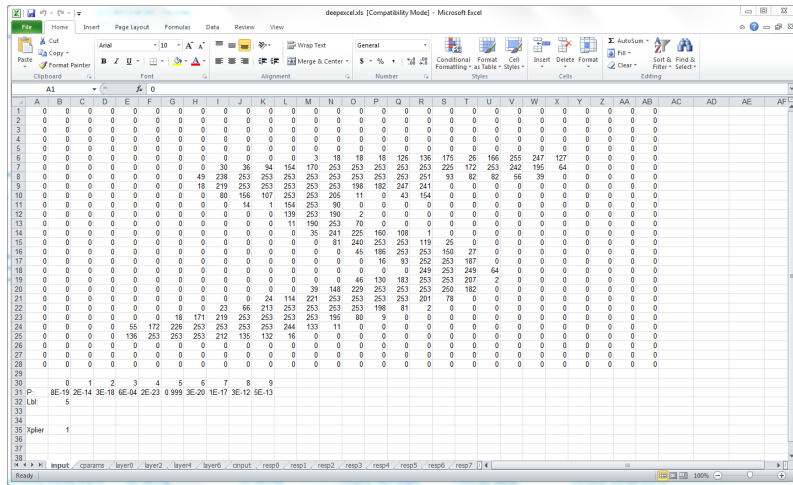
INTRODUCTION

As anyone in machine learning and computer vision will tell you, Deep Learning is the right tool to solve the problem. And as anyone in business and finance will tell you, Excel is the right platform to implement your solution. But until now, there has been no way to do Deep Learning in Excel. To fill this gap we have developed EXCELNET, the ultimate synergy of spreadsheets and Deep Neural Networks.

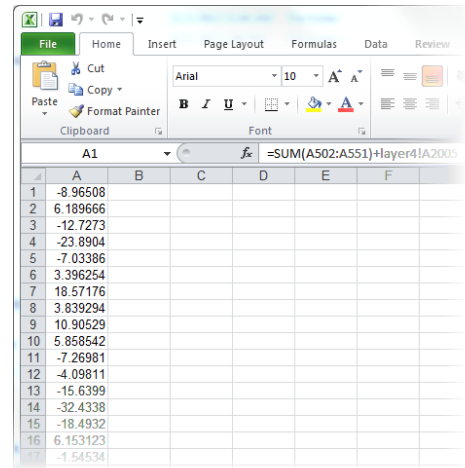
APPROACH

Like the human brain, Excel has “cells”. In Excel, as in the brain, these “cells” are organized in “columns”. Our approach is to type *weights* into those “cells”. We group the *weights* into “layers”, and put each layer in a “sheet”, which can be easily previewed and modified by the user. We then leverage the sophisticated multiplication and addition capabilities of Excel to multiply and add these *weights* to perform advanced tasks such as recognizing handwritten digits [3].

¹Or LibreOffice, if you’re an open source communist. Excel is a trademark of Microsoft, we believe.



(a)



(b)

Figure 1. EXCELNET in action: (a) the interface, where you are in control of what goes into the network and where the results can be obtained instantly; (b) examining an activation map. Typically getting access to the power of Deep Learning requires hiring a nerd or learning obscure nerdy things. EXCELNET puts you in control, and lets anyone harness the power of Deep Learning!

Low cost of entry: Not willing to invest in pricey GPUs? With a basic laptop, and a copy of LibreOffice, you too can hop on the Deep Learning bandwagon with ExcelNet.

No programming required: Afraid of missing out of the building the next big thing in Deep Learning because you do not know how to program? EXCELNET makes it easy by leveraging a basic level knowledge of MS Excel.

Artisanal data science: Instead of slurping heaps of data from the Internet, EXCELNET enables you to hand-enter both weights and inputs, just like in the good old days. The resulting numbers are more authentic.

HOW DOES EXCELNET WORK?

Machine learning practitioners may wonder: how does EXCELNET work?

It's simple: it's a self-contained .xls file that contains a sheet for I/O, and then sheets for the filters and biases of several layers of a convolutional neural net. The remaining sheets apply mean-subtraction, convolutions, rectified linear units (ReLU), fully connected layers, and a softmax function. All of these operations can be implemented using Excel functions. The I/O sheet then references the final output of the softmax function.

HOW DO I USE EXCELNET TO CLASSIFY NEW DATA?

Simply enter in your data in the rows and columns of the data-entry matrix, and watch the posterior distribution change in real-time²! You may find it helpful to use the multiplier cell at the bottom that multiplies your entered numbers by a fixed

value so you can type in 1's and 0's as opposed to 0's-255's. When you have your prediction – save your .xls file, which will save all the activations and the current prediction.

²At least if you're having a branded experience and using real MsExcel: LibreOffice is definitely slower.

CONCLUSIONS

Please visit deepexcel.net to try EXCELNET today!

REFERENCES

1. James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU Math Expression Compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*.
2. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).
3. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 2001. Gradient-Based Learning Applied to Document Recognition. In *Intelligent Signal Processing*, S. Haykin and B. Kosko (Eds.). IEEE Press, 306–351.
4. A. Vedaldi and K. Lenc. 2015. MatConvNet – Convolutional Neural Networks for MATLAB. In *Proceeding of the ACM Int. Conf. on Multimedia*.