Infoviz Update

Jason Hartford Neil Newman

November 2015

1 Update

We built a server with a REST API for interfacing between the javascript visualization code and the python code that performs computations. We have started learning D3 through tutorials, and experimented with NVD3 as a quick way to get basic plots going.

2 Previous Work

2.1 Model Visualisation

The closest prior work that we are aware of is [2], which develops an interactive system for validation of relatively complex regression models that allows researchers to understand the model's behaviour under different optimisation regimes. They demonstrate the usefulness of allowing engineers to visualise slices of the target function in order to get a feeling for the model's behaviour around a particular point in space.

The deep learning literature has also attempted to visualise components of their models in order to better understand how they work. In [3], the authors build functions that simulate the loss surfaces of challenging non-convex optimisation problems. To support this work, they use random projections to visualise slices of their target function.

In [1], the authors present a framework for interactively building regression models. Their system is centered around two small multiples overviews that show both single and pairwise relationships between features and the regression target value, ranked by a variety of relevance metrics. This approach is particularly effective for regression models because users can see the difference between the target and predicted value for each data point on each feature axis. Unfortunately, in our context, the model outputs a probability distribution and hence can't be directly represented by a single target value. Because regression models can be fit quickly, their system also allows interactive changes to the regression model. They use linked 3d views effectively for validation of their model.

[5] is a tool to systematically explore a multi-dimensional parameter space that affects the quality of image segmentation algorithms. Unlike regular pairwise small-multiples which typically only show the lower triangle of the pairwise matrix, Tuner uses both triangles in order to display two different optimisation objectives simultaneously.

[4] describes *visual parameter space analysis*: systems where the inputs contain parameters that can be tuned that will affect the quality of the output. The paper distinguishes between different types of input: *control parameters*, which are tuned, and *environmental parameters* which come from measured data. The paper also codifies analysis tasks: using this paper's terminology, we have an *optimization* task, since we are interested in finding the best parameters given a loss function, as well as a *sensitivity* task, since we are interested in studying the impact on outputs of varying the model's input parameters. However, unlike this paper, our main task cannot be achieved by viewing the model as a black box, since we want to understand what it is that the black box is doing.

In contrast, [6] provides a viz tool with the goal of understanding the relative importance of input and hidden units in neural networks. The importance of a weight (a metric derived from propagating the weight throughout the network) is encoded in the width of the connection, and the importance of an input unit is encoded by its size. The authors were able to use these visualizations to identify what the important features were in a spam classifier, and to remove unimportant nodes from the hidden layers (leading to a more compact network with similar accuracy). However, the examples in the paper only have a single hidden layer, and it is unclear how these techniques would generalize to networks with multiple hidden layers.

2.2 Feature discovery

[7] visualises the layers in a convolutional network that is used for image classification, using a derived data technique that allows them to infer the input pixels that resulted in particular outputs in intermediate layers in the network. Unfortunately, this technique relies on the visual structure encoded in the parameters of the convolutional network for be useful, and thus doesn't generalise to our domain. This limitation motivates our interest in comparing learnt features to hand-crafted features to understand the intermediate representations in our model.

References

[1] T. Muhlbacher and H. Piringer. A partition-based framework for building and validating regression models. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):1962–1971, Dec 2013.

[2] Harald Piringer, Wolfgang Berger, and Jürgen Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation. In

Computer Graphics Forum, volume 29, pages 983–992. Wiley Online Library, 2010.

[3] Tom Schaul, Ioannis Antonoglou, and David Silver. Unit tests for stochastic optimization. *CoRR*, abs/1312.6055, 2013.

[4] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Moller. Visual parameter space analysis: A conceptual framework. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2161–2170, 2014.

[5] T. Torsney-Weir, A. Saad, T. Moller, H.-C. Hege, B. Weber, J. Verbavatz, and S. Bergner. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1892–1901, Dec 2011.

[6] Fan-Yin Tzeng and Kwan-Liu Ma. Opening the black box-data driven visualization of neural networks. In *Visualization, 2005. VIS 05. IEEE*, pages 383–390. IEEE, 2005.

[7] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer International Publishing, 2014.