

# VU Visualization and Visual Data Analysis

M4



universität  
wien

Wintersemester 2017

Christoph C. H. Langer (0500580)

Jane Doe and John Doe

## Disclaimer:

Although I tried to meticulously plan the last 2 weeks of this semester ahead. I thought the report was due at the weekend before the presentation and I realized my mistake only on Thursday after class. I am doing my best to work on my time management so this simple mistake feels particular annoying since I wanted to end this semester on a high note and without a disclaimer. Thus this report is missing a week of my planning and this is all I could salvage.

A tableau dashboard in a .zip file accompanies this report. The final version will also be found on <http://www.christophlangner.com/vis/> and it will be updated before the live-demo during the final presentation.

## Motivation:

Visualize the parameter space of machine learning algorithms in particular Neural Networks for this we conduct a pilot design study.

## Problem:

There is a lot of hype concerning neural networks and deep learning. However it is still not well understood how to pick the parameters that make them work well. For example, these questions are largely still unanswered:

How many layers should the network have?

How many internal nodes should it have?

Which activation function work best?

How should we set the learning rate?

How do properties of the underlying data influence the result?

Our idea is to create an environment that lets the user explore the impact of such

parameters in a structured way. We try to give according to Shneiderman's mantra: an overview, provide filtering and show details on demand (zooming or in our case semantic zooming is currently outside the scope of M4 but it is planned in the future).

These goals are hard to accomplish with existing standard tools. The difficulty is not only to visualize this high dimensional data but also to first understand which parameters are important. With this project we can systematically explore the individual parameters alone as well as in combination using standard dimensionality reduction approaches.

### Tasks:

A user wants to get an overview and understanding, which parameters are important and in which range they created useful results and how they impact different quality metrics like, precision, recall and f-measure.

The parameter exploration task was split as in [1] into following subtasks. Our current tool performs only optimization, fitting, and outlier detection. Uncertainty and sensitivity can not be addressed without generating more data and in a different format, this was discussed in more detail during the M2 report. The partitioning task is only partially achieved in our prototype. We partition and can select according to the output space. However we would like to be able to do this in more details in the next iteration, by looking at clusters of final segmentations as in [8], looking at the learning over behavior time using bundled lines as in [4] or detecting different types of network topology's using a parallel coordinates as in [3].

### User:

The user will be anybody who is interested in using or trying to understand neural networks. This can be student with minimal computer science background and who is trying to understand the concepts behind neuronal networks, up to a senior researcher that tries to gather new insights about the behavior of neuronal networks or tries to understand how to tune them to his particular use case. However the use cases described above are similar, both want to achieve a deeper understanding. Only the level of guidance necessary and explanations needed to use our tool differ.

## Data:

We are using the datasets created using a python implementation of the online demo created for the [CASS challenge. \[7\]](#)

It is based on the web-based tensor flow playground demo found here: [Tensorflow Playground](#) [6]. The dataset will be extended during the next iteration of this project.

We are systematically looking at data generated from the tensor flow playground [6]. The playground gives works on an informed trial and error basis and might even be seen as a local-to-global view (i.e. if we play enough with the individual parameters, we get a rough understanding how they effect the NN). Our approach however tries to go a step further and using a large number of precomputed settings (i.e. with different parameters), we try to give the user a global-to-local view.

Before you start visualizing we start with the “what” part (Munzner Ch.2 – What Data abstraction) and have closer look at our data.

Let me divide in data into 3 distinct parts:

1) These attributes describe the data our network is working on:

Attribute Name	Attribute type	Type details	Description
data	categorical	{circle, gauss, xor, spiral}	dataset shape
noise	numerical discrete*	int: [0 .. 50]	data noise percent
training_ratio	numerical discrete*	int: [10 .. 90]	ratio of training to test data

*Remark:* Instead of training\_ratio the actual training/test numbers would have been more useful. Summary statistic or aggregated values can be calculated in later steps - the raw data should not include these or at least keep the original values.

2) Then we have attributes that describe the parameters of our the TensorFlow neuronal network, which I would like to subdivide into two further groups:

a) Network topology specific Parameters:

X1, X2, X1Squared, X2Squared, X1X2, sinX1, sinX2	numerical	binary	1 if the feature is an input to the network, 0 otherwise
layer_count	numerical discrete	int: [0 .. 6]	number of hidden layers
neuron_count	numerical discrete	int: [H1 .. H6]	sum of neurons in all hidden layers
H1 .. H6	numerical discrete	int: [0 .. 8]	number of neurons in hidden layer 1 .. 6

b) Learning specific Parameters:

learning_rate	numerical discrete*	float	learning rate
---------------	---------------------	-------	---------------

activation	categorical	{ReLU, Tanh, Sigmoid, Linear}	activation function for hidden layers
regularization	categorical	{None, L1, L2}	regularization type
regularization_rate	numerical discrete*	float	regularization rate
batch_size	numerical discrete	int	batch size
epoch	numerical discrete	int: {25, 50, 100, 200, 400}	epoch for which the stats were generated

**Remark:** In the data description epoche is called an output but I would argue, in the way that it is used (defined as a strict cutoff beforehand and not as the result of self terminating algorithm) it is more reasonable to call it a learning specific parameter (i.e the time a NN is allowed to learn).

3) Last but not least we have attributes that describe output characteristics of our NN, these have are all of the numerical continuous attribute type (respectively as continuous as computer data can be).

total_time	numerical continuous	float	Total time at this epoch
mean_time	numerical continuous	float	Mean time per epoch
train_loss	numerical continuous	float	Training loss at epoch
train_TPR	numerical continuous	float	True positive rate for training data
train_FPR	numerical continuous	float	False positive rate for training data
test_loss	numerical continuous	float	Test loss at epoch
test_TPR	numerical continuous	float	True positive rate for test data
test_FPR	numerical continuous	float	False positive rate for test data

Certain discrete attributes are marked with a star '\*'. These are not discrete per se but only due to our need for sampling the parameter hyperspace. In general the input parameter are very discrete or binned to keep the numerical combinations of the hyperspace in check.

We only have TPR (true positive rate) and FPR (false positive rate) and the loss. Unfortunately we do not have the precession of the data and without any additional information is not possible to generate them out of the TPR and FPR alone. Future iterations of the simulation should output the full confusion matrix and then it is easy to calculate a metric of our choice (e.g the 1D F1-Score and the 2D precision and recall combination).

## **Related Work:**

### **Other visualization solutions:**

Conceptual design is based on the paper: Visual Parameter Space Analysis: A Conceptual Framework [1].

The solutions in the current iteration are based on the ideas of the author alone. So far no similar tool has been created to view the mixture of input parameter (binary categorical and numerical data) and to explicitly try to explore a multiple dimensional output space.

### **Any previous Ideas incorporated into your solution:**

The tool described in this report will be the overview part and could be the first part in a multitier analytic pipeline.

During the conceptualization we looked at [8] for a clustering of the final output. We looked at the recent scatter plot paper [5] for solutions to solve the over plotting problem.

We took squares [3] as an inspiration to use a parallel coordinated based approach to depict the trends in network topology.

We looked at sliceplorer [4] to view the variance and general learning behavior of groups of neuronal networks. All the ideas can be added downstream of our approach and were discussed in more detail during M2.

### **Reference to both academic and commercial tools used.**

Only Tableau is used in the implementation, however we will switch for next iteration to D3.js.

## Approach

### Description of your visualization design

The main element of the design is the scatterplot. It is chosen to depict the output space, since they are all numerical and continuous. Further 2-dimensional correlation can be depicted. Through color we have an additional channel to depict a 3<sup>rd</sup> dimension. Here the user will have a choice through a dropdown menu to either show a third dimension or directly depict a correlation of one of the parameters to the output space. The user can then chose a region of interested and view the changes in the histograms of the individual parameters. Thus the user gets direct feedback on the summary statistic of individual parameters. The user can also directly interact with the histograms to restrict a certain parameter to a particular range and see how the output space is changed, for example which regions are not covered anymore.

### Reasons for your design choices

One of the still unsolved problems when working on this project is how many graphics can one add to dashboard without overloading the user with visual clutter. During the design process we removed all elements to a minimum. Thus the concept is to start with the bare minimum but let the user add elements later if he likes.

We reduced the original dual scatterplot to a singular one. In a later step the scatterplot may be expanded to multiples to handle over plotting or the user might choose multiple axis and coloring schemes to view multiple effects at the same time. These are the only expandable and user definable element in this dashboard. The histogram overview offers all input parameter at a glance. This works well for categorical and numerical values. The current design should also be viewable without scrolling.

Only after gaining an overview and after filtering is provided, the user can then move on to the next phase of analysis and look at elements later described but not implemented in this iteration. (i.e. the network topology view, correlation between parameters, learning behavior over time and analysis of uncertainty and sensitivity).

Keeping the first part simple lets us later add new layers and views in a form of sematic zooming.

## Implementation

### Brief description of how the system was implemented (toolkits, languages, platforms)

The main bulk of the tool will be implemented in tableau since this allows the fast implementation of interactive applications and giving the best general and full connected view of the data without vastly exceeding the allotted time for this project. In the next iteration we switch to D3.

I played with the data in Tableau, tensorflow - playground and used the information gathered during the VIS lecture and CASS2017.

If I could restart:

During our book club we got introduced to vega-lite. I had a quick look at the basic graphs used in my final dashboard and similar functionality is setup far faster as in D3.js but on still is far more able to customized and combine elements in a more customizable fashion. Many of the implementation challenges bellow could have been easily resolved there. The extra time in coding would have been recuperated trying to search through forums to find a way to solve problems that later turn to be out unsolvable in Tableau anyway.

Since this project will not end with this Tableau implementation I did not bother to solve some of the problems or fully restrict my design space to functionality that can be achieved in tableau. Further the discussions will include many elements not fully implemented in the tableau environment yet and will show proof of concepts for the next iteration in D3.js.

If we want a quick standard figure, I would compare Tableau to Excel for drawing graphs and gg-plot to vega-lite. I would only create very basics plots with Excel and only if I know the default forming and settings are roughly enough, if I want anything of higher production value or a reasonable important report I would switch to gg-plot since the initial higher time invested to setup the graph, is easily recuperated if I need multiple similar ones with different data and when I want to edit its appearance to my personal choice. Since here I can interact in a far more structured way and do not have to click and find the settings in obscure and version dependent menus.

### **Any serious implementation challenges you encountered and how you handled them**

As described above most of the challenges involved limitations of tableau.

Relative to filtering values and multiple filtering: If one uses multiple elements and switches between multiple elements filter selection can become stuck and can not be removed with the usual interactions. (have to be deleted on the corresponding sheets themselves, since the dashboard has a huge number of linked sheets finding them is very tedious). Moving the dashboard onto the web (Tableau Public) mitigates this by being able to simply refreshing the Page thus reload the Dashboard with the default parameters, however any progress and settings so fare are lost. Further the direct interaction with the filtered/not filtered data efficiently was not possible, it was impossible to display the relative changes in the histograms created by parameter or output space selection. Since only the final number after filtering is accessible.

Selecting Multiple bars, the original concepts of using the histograms as scented widgets was nice however only a singular bucket can be selected thus normal sliders had to be added to achieve the result originally intended, this sorts of interaction can far better be achieved in D3.js

Selecting regions in the output space: only square regions can be selected this can not be changed.

Url rules: are very restrictive you can not choose which of the multiple web-site containers in a dashboard is target, this is only performed for the first one. This is annoying if you have multiple however over or even clickable events to happen in

one dashboard. Like showing help features in on web-site containers, and showing the a resolution of for example the image path by hovering over the a particular point in the scatterplot. This was resolved by removing the detailed image view of a data point upon hover, the final dashboard has now a static image as placeholder. The detailed explanation upon hovering of elements and help question was added instead since this was asked by most early testers and would have been necessary to work with the tool without any prior explanations.

Scatterplot:

Scatterplots come as is, over plotting is a real problem shown by a graphic in M2 that adds histograms to the axis.

A recent paper about scatterplot [5] addresses many issues however none of the concepts can be implemented in tableau. On of the ideas I played with was using alpha blending to show the overall level of over-plotting but this does not work with multiple colors thus the colors were separated in multiple juxtapose, however since filters are again connected and can not be individually applied to individual sheets individual this is not in the final result but instead I show a stitched together screenshot that proves this concept [SCREENSHOT]

Reloading Parameters: No efficient way to save and reload parameters were possible. Future D3.js implementations will created savable JSON-files that can be saved and loaded for better comparison in downstream elements of our tool as well the possibility to easily save interesting findings and share them since only the filtered data output could be accessed and saved over the Tableau-API.

Full tool: Too many shortcoming of the tableau framework were identified, thus polishing the tool to have production value was omitted because of time efficiently and instead the implementation in D3.js was started. The life demo during the presentation will show a careful chosen path through the program and two use cases. The Tableau implementation thus is not ready for extensive and unsupervised beta-testing. This will also not be resolve in this implementation.

## Results

### Scenario(s) of use, including screen shots of the system being used:

Two scenarios that came up by the user were:

- A) getting an overview of a certain subsection of the output space – how this influences the parameter choices i.e which parameters will lead to this result.
- B) Filtering according to a particular feature and directly view how the output space would be effected, meaning people wanted to see how the multiple dimensional output space is affected by restricting a particular parameter.

### Performance of the system:

The overall performance on the 100.000 item dataset was rather slow, the 2-4 second lag when readjusting the filtering was perceived as a major annoyance by the first testers. Since the final product will include an even bigger dataset and the overall performance of tableau is not really influence able performance, reimplementing in a different framework will be necessary.



Filtering takes for ever – not smooth enough for final production tool.

### **Feedback from the evaluations about the design and functionality of your tool:**

Feedback of the design and functionality was gathered by friends and roommates, since the prototype is still very rudimentary. Explanations were given, slight steering was performed when a functionality did not match the expected behavior during their testing. However one of the problems that still exist was that how to interact and what they interacted was not clear from the beginning the tool will need a description like tensor flow has at the bottom but a general feedback was that an help or info buttons that explained the functionality on hovering over elements or made visible by clicking on tiny info buttons would have been nice. This will defiantly implemented in a later iteration. One of the simplest ways to do this will be done up to the final presentation. The implementations in tableau will be a url-action that displays this kind of information into a web-page container on the dashboard upon hovering over and clicking on certain elements.

When asked what conclusion they achieved concerning the output space, the data and the parameter space, 3 out of 4 came up with feasible but also non overlapping novel insights.

Further a way to store the settings of the filters and easily with between two of them or display the results in juxtaposition was asked 4 out of 4 for as well as the question, further 2 out of 4 if it is possible to output the a subset of results and browse through them and visualize what happened here like on tensorflow.

Last but not least 2 out of 4 users asked if it would be possible to switch the data that created the output space as well as the coloring. This shows that my initial guess how the user would intuitively want to interact with this tool and this feature will have a high priority when going through the next iteration.

1 out of 4 asked for additional summary statistics value, he wanted the ability to quantify the perceptual change created by interacting with our tool

## **Discussion**

### **Strengths and weaknesses of your approach and implementation**

**Strengths:** The solution is broken down into very basic and simply understandable elements. The solution can be easily expanded and used as the staring point for semantic zooming. A number of alternatives were looked at and considered during the initial testing phase. The current concept produced a number of important insights, which are not part of the current implementation but are crucial for the next iteration.

**Weaknesses:** Many of the original ideas were dropped due to inability to implement them in tableau. However a reasonable partition of the project was achieved. The current setup helped to create a rough idea what the real prototype would need. Future steps were identified

### Lessons you learned:

- Project management. During the last semester with multiple lectures, I ran in a general time management problem. Some were due to my very naïve estimation of work effort and my drive to achieve something far better than reasonable in the allotted time. Further I simply discounted that I was working alone on the project and went for something too ambitious and partitioned the project badly. I worked on too many problems and looked at too many different aspects right from the beginning. I also underestimated how much effort and work a design study entails. Further many of the subtasks I originally created for myself could have been 3-man projects on their own as it later turns out.
- Even simple tasks like designing a scatterplot, can turn out to be a far more complex problem as [5] shows.
- Tableau is only a tool to generate a quick overview of the data. I would not use it again to create a prototype of a visualization tool unless one can and wants to reduce the design to something very basic.
- Final report testing and setting up a version for a live demo takes at least 2 full weeks, losing one to a planning error is not reparable.
- The perfect tool for rapid prototyping would be a library that produces quick D3 implementations for default graph types. This tool would be like a website builder, with which one can rapidly create a good looking and working webpage but can still inspect and modify the individual elements to one's liking if necessary.
- The Tableau API basically only lets you interact with the input and output of the Dashboard, however it does not let you modify and interact with anything happening within Tableau. These restrictions have to be taken into consideration earlier when planning the whole design of a tool.
- This project is a very good example, why working with new tools and frameworks is a very dangerous endeavor on a tight time budget. Further it shows that after 2 months of development the whole system might need to be completely re-implemented and re-designed.
- Many concepts of visualization and the exploration of the data during this project are of great importance for the next iteration.

**A clear separation of tasks between the group members: you have to detail who did what for this milestone.**

Christoph Langer did everything.

### References:

- [1] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Oller, "Visual Parameter Space Analysis: A Conceptual Framework," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, 2014.
- [2] T. Munzner, *Visualization Analysis and Design*. 2014.
- [3] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 61–70, 2017.
- [4] T. Torsney-Weir, M. Sedlmair, and T. Möller, "Sliceplorer: 1D slices for multi-dimensional continuous functions," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 167–177, 2017.
- [5] A. Sarikaya, S. Member, and M. Gleicher, "Scatterplots : Tasks , Data , and Designs," vol. 24, no. 1, pp. 402–412, 2018.
- [6] Daniel Smilkov and Shan Carte: <http://playground.tensorflow.org/>
- [7] Visualization assignment for 2017 Czech-Austrian Summer School on "Deep Learning and Visual Data Analysis": [https://github.com/hyounesy/cass2017\\_vis](https://github.com/hyounesy/cass2017_vis)
- [8] B. Fröhler, T. Möller, and C. Heinzl, "GEMSe: Visualization-Guided Exploration of Multi-channel Segmentation Algorithms," *Comput. Graph. Forum*, vol. 35, no. 3, pp. 191–200, 2016.