Data Visualization Milestone #3 – Project Book
A high-dimensional inspection tool for deep neural network safety

Kyle Matoba and Arnaud Pannatier

June 4, 2023

1 Introduction

Since we began this project, the topic of *AI safety* has passed firmly into the mainstream. Deep learning experts are meeting with world leaders,[1] NVIDIA has realized the largest ever one-day gain to a US stock following its earnings release (driven by deep learning),[2] and the world's richest person is discussing the topic on mainstream US news.[3] As we write this report, a major statement on AI risk[4] is being widely reported on.[5]

At the center this discussion are concerns about oversight of increasingly powerful deep learning systems. The essential fact being that the manner in which deep neural networks reason can be highly nonintuitive to humans. In essence: deep neural networks can demonstrate remarkable performance on a task *most of the time*, but can sometimes exhibit completely nonsensical behavior for reasons that are not well understood. One way to help make AI systems safer is to build tools that enable humans to effectively view their prospective behavior. Our course project addresses this problem for a class of deep learning models that control physical systems.

Our visualization is constructed to dynamically evaluate the outputs of neural networks. Due to the high cost and latency of the extensive computation,[6] *observing best practice is crucial in this area. And best practice is to use Python-based PyTorch ([7]).*[7]

2 The path to our final project

Note: we are attempting to effectively and efficiently convey the essence of complex data. However, our tool does not propose to visualize a particular problem instance, but be applicable to any deep neural network with a manageable preimage and behavior which could definitely be recognized as "wrong" by a domain expert. Put simply, we are less statisticians or graphics designers, and more ML engineers in developing this tool. There is more need for technical correctness and architecting and perhaps less scope for aesthetics and presentation than the median project. And this reality guides our process book as well.

2.1 Inspiration

[9] is perhaps the most elegant study in large the literature that proposed to prove the safety[8] of a deep neural network, and an important antecedent to this work. [5] took this a step further with a method for analyzing networks that would never wrongly term unsafe a truly safe network, and

[1] https://edition.cnn.com/2023/05/16/tech/sam-altman-openai-congress/index.html

[2] https://www.ft.com/content/b074781a-683f-4b20-8c70-33bb71399d94

[3] https://www.foxnews.com/video/6325259104112

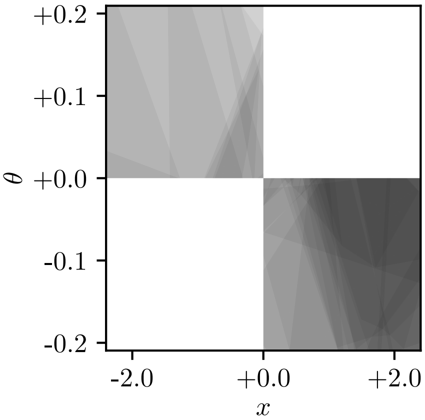
[4] https://www.safe.ai/statement-on-ai-risk

[5] https://www.bbc.com/news/uk-65746524

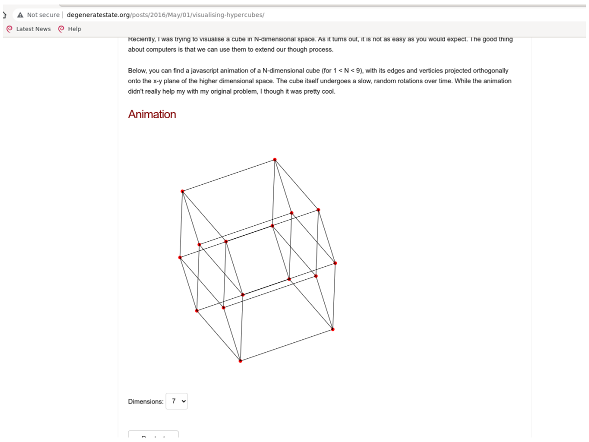
[6] Which can easily run into the many millions of dollars to train a large model – training the GPT 4 large language model reportedly cost upwards of USD 100 million, for instance, cf. https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over/.

[7] OpenAI, for example, officially uses PyTorch, cf. https://openai.com/blog/openai-pytorch. There are some plausible alternatives, such as TensorFlow, but there are for sure no JS-based plausible alternatives.

[8] "Safety" has a high-level definition as "absence of unexpected behavior around a set of inputs" that is operable at both a technical and informal level.



(a) The most direct antecedent of our work [6], depicts a two-dimensional slice of a four-dimensional control policy.


degeneratestate.org/posts/2016/May/01/visualising-hypercubes/

Reciently, I was trying to visualise a cube in N-dimensional space. As it turns out, it is not as easy as you would expect. The good thing about computers is that we can use them to extend our though process.

Below, you can find a javascript animation of a N-dimensional cube (for 1 < N < 9), with its edges and verticies projected orthogonally onto the x-y plane of the higher dimensional space. The cube itself undergoes a slow, random rotations over time. While the animation didn't really help my with my original problem, I though it was pretty cool.

Animation

Dimensions: 7

(b) An online app showing a rotating high dimensional hypercube projected onto a computer monitor, a direct inspiration for our "rotation" plot. From [1].

extending it to operate over the entire input domain, albeit with a much higher complexity. [6] further introduced a method for representing deep neural networks by computing the set of all inputs that give each possible outcome.

Perhaps the most direct antecedent of our work is presented in Figure 1b, which is repurposed from an early version of [6], and which depicts a two-dimensional slice of a four-dimensional deep neural network control policy. This policy is trained using reinforcement learning and achieves a good criterion. The precise experimental setup is fully described in subsection 4.2. Within quadrants II and IV, it is possible to unambiguously characterize the correct behavior for any controller on physical principles – nonetheless even a well-performing policy makes clearly wrong moves, with the shading in these quadrants is proportional to the volume (in $\dot{\theta}, \dot{x}$) space) of the input space on which an action contrary to the physics of the problem is emitted.

2.2 Planning Ahead

We were faced with the challenge of finding an effective and simple approach to present our high-dimensional data. After careful consideration, we decided to tackle this task by splitting the visualization into two main axes, each offering a distinct perspective on the data.

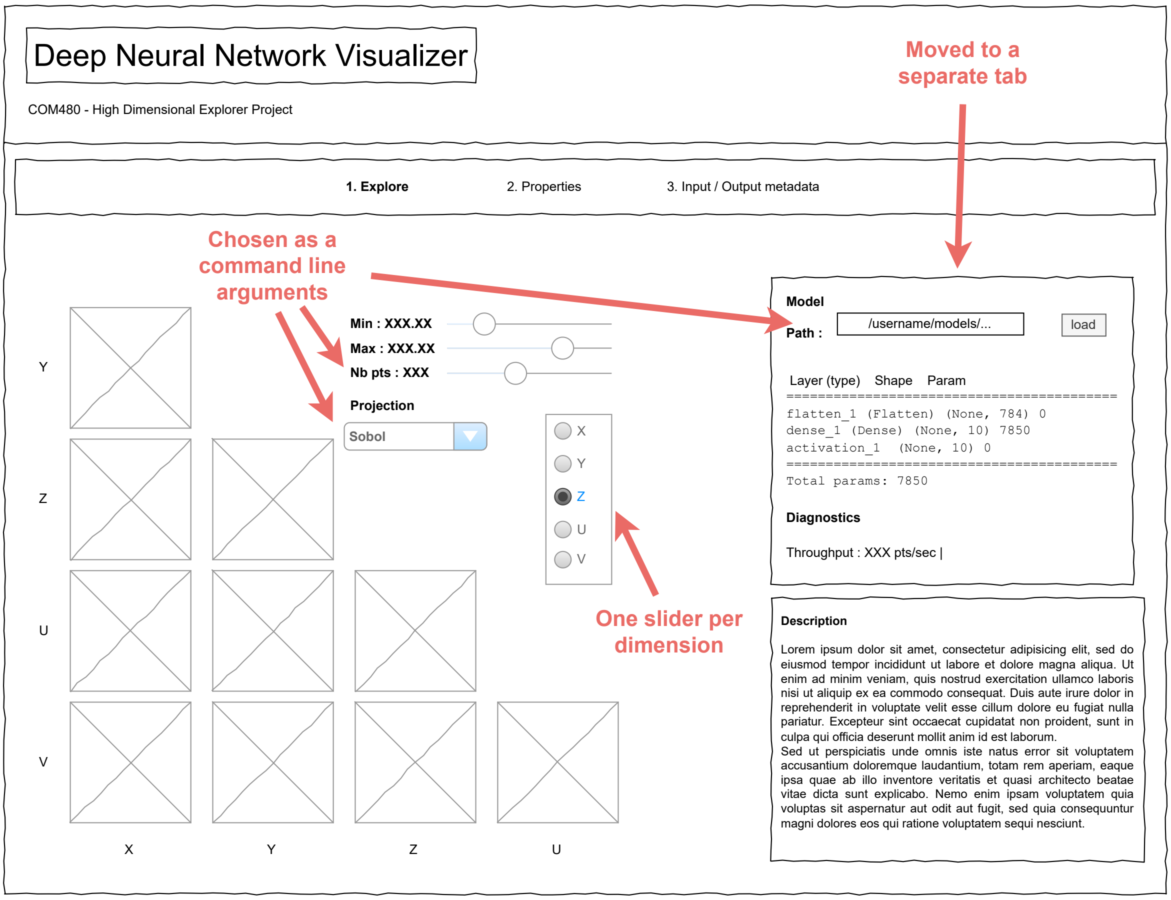
The first axis, illustrated in Figure 2a, follows a more traditional approach. We chose to project the data onto every possible subset of two dimensions within the entire input space. This allowed us to capture different combinations of variables and their interactions. To enhance the visual experience, we organized the visualization so that each column and row corresponded to a specific dimension, creating a structured grid-like arrangement. Furthermore, we ensured that the range of values was shared across all plots, facilitating the comparison of different dimensions. By adopting this layout, practitioners can easily identify patterns and relationships between different axes, leading to deeper insights.

For the second axis, depicted in Figure 2b, we aimed to provide a unique and interactive way of presenting the data. This approach empowers the user to select specific regions of interest and observe how the corresponding data points are distributed across multiple dimensions. By allowing such region-based exploration, practitioners can gain a better understanding of the relationships and characteristics within localized areas of the data.

By employing these two complementary visualization strategies, we offer a comprehensive and insightful view of the high-dimensional data, enabling practitioners to extract valuable insights and make informed decisions.

2.3 "I know it when I see it"

As mentioned in the first milestone, we want not to only be able to investigate hypotheses about incorrect behavior, but we also want a method to simply apply human intuition to discover such



Deep Neural Network Visualizer

COM480 - High Dimensional Explorer Project

1. Explore 2. Properties 3. Input / Output metadata

Min : XXX.XX

Max : XXX.XX

Nb pts : XXX

Projection: Sobol

○ X
○ Y
● Z
○ U
○ V

Model

Path : /username/models/... load

```
Layer (type)    Shape    Param
===============================================
flatten_1 (Flatten)  (None, 784)  0
dense_1 (Dense)  (None, 10)  7850
activation_1   (None, 10)  0
===============================================
Total params: 7850
```

Diagnostics

Throughput : XXX pts/sec |

Description

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

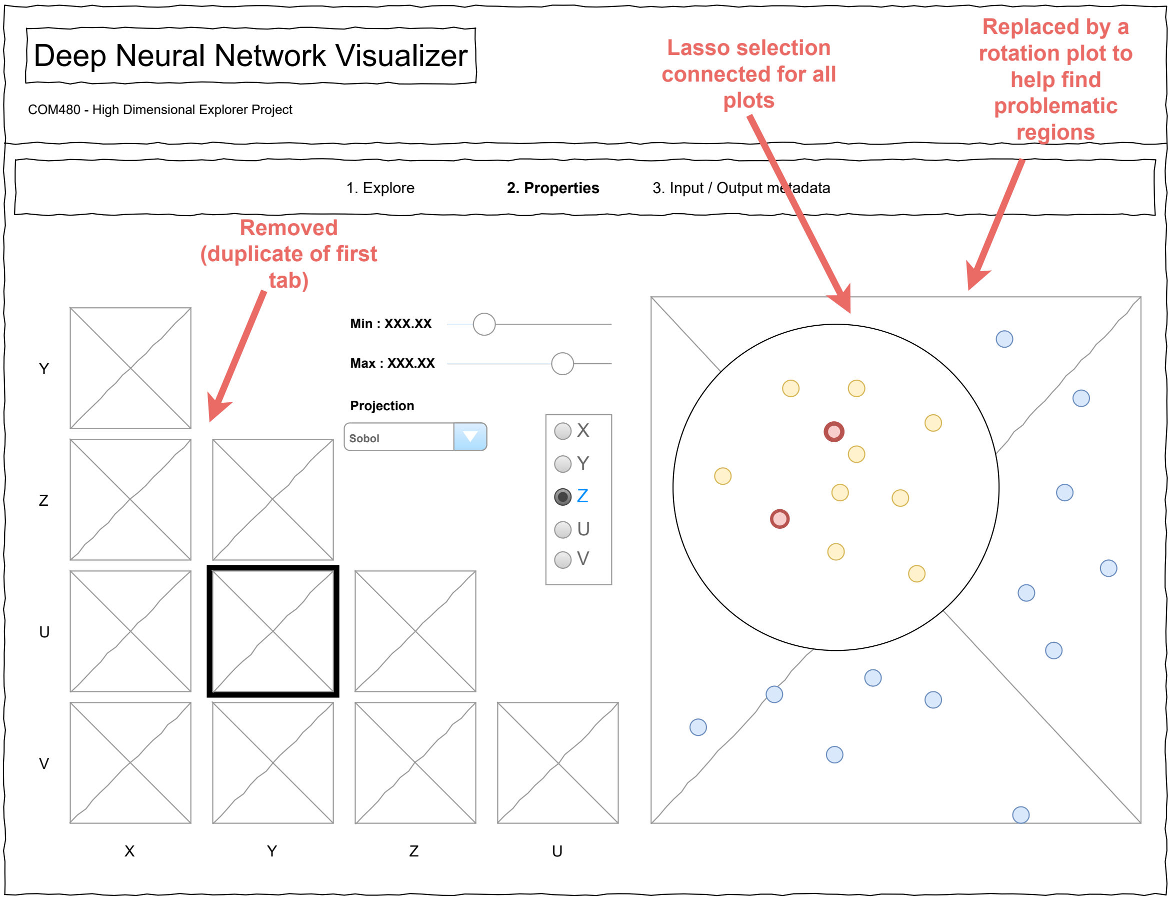
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

Moved to a separate tab

Chosen as a command line arguments

One slider per dimension

(a) Wireframe of the plots tab and modifications.



Deep Neural Network Visualizer

COM480 - High Dimensional Explorer Project

1. Explore **2. Properties** 3. Input / Output metadata

Min : XXX.XX

Max : XXX.XX

Projection: Sobol

○ X
○ Y
● Z
○ U
○ V

Lasso selection connected for all plots

Replaced by a rotation plot to help find problematic regions

Removed (duplicate of first tab)

(b) Wireframe of the second tab and modifications.

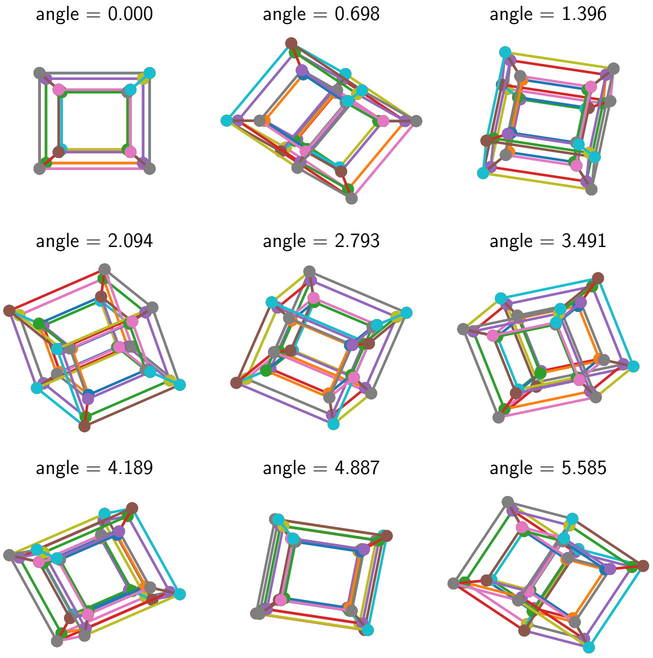


Figure 3: First steps in matplotlib: Nine projections of the five-dimensional unit cube.

behavior, prior to deployment. This is really the essence of AI safety: a concern that we do not even really understand the nature of bad outcomes that could arise from too-intelligent computer systems.

We include a tool that dynamically presents different slices of the predictions, inspired by [1]. The idea is that a person can passively observe this behavior and through noticing patterns may be able to formulate hypotheses about its operation. The screencast that accompanies this document demonstrates this functionality best.

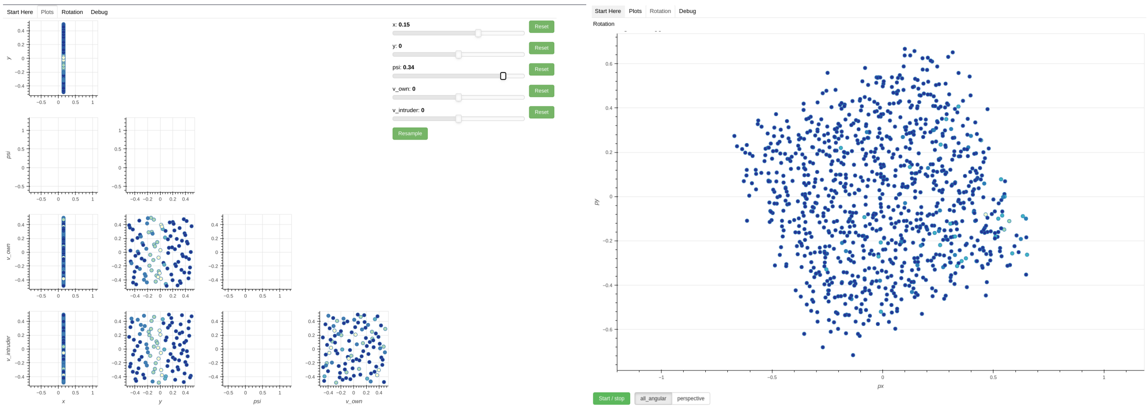
2.4 Final Project

In the final project, we made changes to our initial design by incorporating a rotation approach. These changes aimed to enhance the overall usability and user experience of our interface. To illustrate these modifications, we have annotated the wireframes with red highlights, as shown in Figure 2a and Figure 2b. Additionally, we have included screenshots of the tool in action, which can be seen in Figure 4a and Figure 4b.

In the first panel, we made some adjustments to simplify the interface. Initially, we had planned to incorporate a UI tool for managing the static elements of our interface. However, during the development process, we realized that this approach was introducing unnecessary clutter. To streamline the design and enhance usability, we opted to rely on a powerful configuration management system, Hydra ([10]) instead. This tool makes it simple to build and flexibly compose (via command line argument, a single yaml file, or a sophisticated hierarchy of nested configurations) various aspects of a data and training pipeline and the eventual visualization. By doing so, we were able to prioritize simplicity and readability, ensuring that users can interact with the interface more intuitively.

This “rotation” plot, incorporates a convenient pause button, allowing users to start and stop the rotation with ease. This empowers users to navigate through the data in a fluid and interactive manner. It also allows the selection of two projection methods, each of which offers distinct views of the model’s behavior.

To maintain consistency and aid users in their analysis, we synchronized the selection between the second panel (rotation plot) and the first panel. This means that if users identify a problematic region in the rotation plot, they can effortlessly locate the same region in the first panel. The first panel, employing a more conventional approach, offers sliders that allow users to manipulate the data points directly. By utilizing these sliders, users can navigate to the specific region they are interested in and observe the corresponding data point.



x: **0.15**

y: **0**

psi: **0.34**

v_own: **0**

v_intruder: **0**

(a) Final grid plots.

(b) Final rotation plots.

3 The tool

3.1 Configuration and extensibility

Since our customer is assumed to be sophisticated, and with access to advanced computing resources we have paid very close attention to making our tool *powerful* to such a user. This means offering flexible configuration through clear and simple interfaces, even if text-based and perhaps requiring a restart. Where a shiny GUI could be more ergonomic for a throwaway analysis, our approach is vastly superior as a basis for industrial-scale work. For example, although Bokeh makes it relatively straightforward to change colors, markers, etc. we jettisoned early on support for these features as relatively shallow, and ultimately not justifying the space they occupy, given how infrequently they would need to be modified.

One aspect in which our tool could be extended is with additional logic to address subcomponents of the analysis. Different point sets, projection methods, or additional on-the-fly analysis, for instance. Our code is also sufficiently modular to allow straightforward extension in this manner. Furthermore, our analysis is notably simplistic in its deep learning pipeline – even a relatively simple, fully connected, shallow and narrow deep neural such as that used in the experiment described at subsection 4.1, cannot be evaluated at scale on general-purpose hardware. Because of this, our app currently includes further settings about the regeneration of pointsets – for most use cases this would be unnecessary with a hardware accelerated (for instance using graphics processing units, GPUs, which would enable several orders of magnitude greater throughput), as a sufficient mesh could be computed in real time.

One extension our tools composes very well with is specialized 3d visualization hardware, for instance via VR headset or projector. With this, the science fiction trope of visually debugging faulty AI logic via hologram might be truly realized! Sadly, this is clearly out of scope for a course project, but it is straightforward to see how our project could be extended thus. Nonetheless, our visualization scales well with additional surface area; we expect that industry users could easily benefit from four or more high resolution, dedicated monitors. It's easy to imagine a “control room”-like atmosphere where specialized technicians “patch” unsafe behavior observed in production (for instance using the method of [8]).

3.2 Technology

The Bokeh [2] webpage states “Tools and widgets let you and your audience probe “what if” scenarios or drill down into the details of your data.” – this is exactly our use-case, and we found it to be very true.

Implementing the stateful approach for updating plots in Bokeh was a frustrating paradigm shift to the more functional way we tend to write machine learning models. Idiomatic Bokeh creates a “data source” object at the top-level scope and reaching upwards from asynchronous callbacks to mutate it in-place. This implicit use of Python scope leads to, in our opinion, some fairly unreadable code that static analyzers hate. We are fairly sophisticated at building data analysis pipelines, but unfortunately our experience was not super helpful within Bokeh's conventions.

Although Bokeh is Python-based, it can be extended with JavaScript. By combining JavaScript with Python, we were able to, for example, reset the slider position when the reset button is pressed.

Despite these challenges, Bokeh remains an exceptional tool for interactive data exploration. Its features, such as linked lasso selection, allow users to select data points on one graph and have them displayed on related graphs projected in different dimensions. Additionally, if no further computations are needed after the initial visualization, users can generate static HTML and JS files without the need for an additional backend. However, in our case, we did not utilize this capability.

Another technological challenge we faced within the scope of this course was hosting the visualization tool. While this is not a significant issue in an industrial-strength use case where the tool is likely launched on a user's machine or a GPU-accelerated cluster, we still needed to find a suitable service that could run a Python backend and correctly route information between the endpoint and the backend. Ultimately, we chose Heroku but had to carefully develop the application to accommodate this choice.

4 Use-cases

4.1 ACAS

To reiterate from Milestone #1, the raw data for this use-case is generated by the Julia code provided by [4]. The code formulates and solves a partially observed Markov decision process (POMDP) to represent the ACAS policy. This policy maps three aircraft relative positions and three relative velocities to five advisories, encompassing strong left, weak left, clear of conflict, weak right, and strong right. In our analysis, we employ a fully connected neural network architecture with five layers, each consisting of 25 neurons and interwoven with rectified linear unit (ReLU) nonlinearity. The model fitting process follows standard procedures commonly employed in neural network training.

All data points are normalized within the range of -.5 to +.5, a convention inherited from [4].

4.2 Cartpole control

This description and figure are largely copied from an early version of [6].

In the "cartpole" control problem a pole is balanced atop a cart which moves along a one dimensional track (Figure 5). Gravity pulls the pole downward, the falling of the pole pushes the cart, and external movement of the cart pushes the pole in turn. The control problem is to keep the pole upright by accelerating the cart.

In the formulation of Brockman et al. [3] controller inputs are: the position of the cart (negative values are left of the origin), x, velocity of the cart $\dot{x}$ ($\dot{x} \leq 0$ corresponds to leftward), the angle of the pole θ from upright ($\theta \leq 0$ means left of vertical), and the angular velocity of the pole $\dot{\theta}$ ($\dot{\theta} \leq 0$ means falling left). For brevity, denote an arbitrary state $(x, \dot{x}, \theta, \dot{\theta})$ by s.

Possible actions are to accelerate the cart in the positive or negative x direction, thus a controller is a function $s \mapsto \{-1, +1\}$. The reward environment encourages balancing by a unit reward per period before failure, where failure means that the pole is not sufficiently upright ($\theta \notin [-\pi/15, +\pi/15]$), or the cart not near enough the origin ($x \notin [-2.4, +2.4]$). The Gym environment itself has no prescribed limits for $\dot{x}$ and $\dot{\theta}$, but via some simple arguments, we can bound these states as taking values in $[-3.0, +3.0] \times [-3.5, +3.5]$ for any model that starts in a physically plausible state.

Consider a still cart and pole ($\dot{x} = \dot{\theta} = 0$), with the cart left of zero ($x \leq 0$) and the pole left of vertical ($\theta \leq 0$). Reasoning that keeping x and θ both near zero is better, since these are further from failure, we see that moving left will steady θ but worsen x, and vice versa. Nonzero velocities make this reasoning more complicated, but one configuration is unambiguous: if $x \leq 0, \dot{x} \leq 0, \theta \geq 0, \dot{\theta} \geq 0$, then pushing right is clearly the correct action. Figure 5 gives depicts a value in this orthant.

5 Peer assessment

We share a PhD supervisor, are officemates, coauthors, and friends. In short: we had a good working relationship before beginning the class, and collaborated productively and equitably on this project. The work was not only equally split in the aggregate, but in most individual aspects as well.

Kyle conceived of the early idea and wrote some of the preliminary prototypes in matplotlib and did relatively more of the milestone report writing. The test problem we presented came from some of Kyle's earlier work, so that was Kyle-specific.

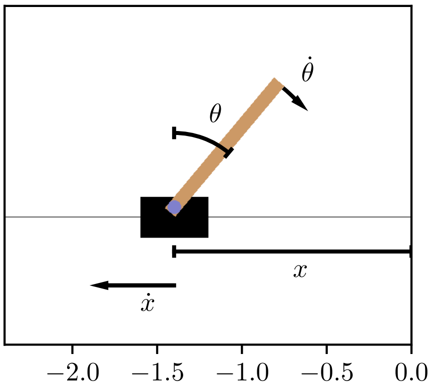


Figure 5: The state space of the cart pole problem, schematically. Here $x \leq 0$ (the cart is left of the origin), $\dot{x} \leq 0$ (the cart is moving leftward), $\theta \geq 0$ (the pole is right of vertical), and $\dot{\theta} \geq 0$ (the pole is moving rightward).

Arnaud handled relatively more of the technical aspects of scaling the visualization, deploying the server, and generally turning a collection of ideas into a cohesive whole.

We both contributed ideas to the final features of the visualization and discussed together all aspects of the aesthetics and usability. We jointly decided on the technology infrastructure and together worked through different aspects of learning Bokeh. We conceived of and built the main logic together. Once we had the basic "tabbed" layout decided, we each worked separately on aspects of the tabs, but again in close communication and with tight feedback loops.

Works Cited

[1] Iain Barr. *Visualising Hypercubes.* May 2016. URL: `http://www.degeneratestate.org/posts/2016/May/01/visualising-hypercubes/`.

[2] Bokeh Development Team. *Bokeh: Python library for interactive visualization.* 2018. URL: `https://bokeh.pydata.org/en/latest/`.

[3] Greg Brockman et al. "OpenAI Gym". In: (2016). URL: `http://arxiv.org/abs/1606.01540`.

[4] Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. "Deep Neural Network Compression for Aircraft Collision Avoidance Systems". In: *Journal of Guidance, Control, and Dynamics* 42.3 (2019), pp. 598–608. URL: `https://doi.org/10.2514/1.G003724`.

[5] Guy Katz et al. "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks". In: *Computer Aided Verification.* Ed. by Rupak Majumdar and Viktor Kunčak. Springer International Publishing, 2017. URL: `https://doi.org/10.1007/978-3-319-63387-9_5`.

[6] Kyle Matoba and François Fleuret. "Exact Preimages of Neural Network Aircraft Collision Avoidance Systems". In: *Machine Learning for Engineering Modeling, Simulation, and Design Workshop at Neural Information Processing Systems 2020.* Nov. 2020. URL: `https://ml4eng.github.io/camera_readys/24.pdf`.

[7] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS 2017 Workshop Autodiff Program.* 2017.

[8] Chongli Qin et al. "Adversarial Robustness through Local Linearization". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems.* Red Hook, NY, USA: Curran Associates Inc., 2019.

[9] Eric Wong and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope". In: *arXiv e-prints* (2017). URL: `http://arxiv.org/abs/1711.00851`.

[10] Omry Yadan. *Hydra - A framework for elegantly configuring complex applications.* Github. 2019. URL: `https://github.com/facebookresearch/hydra`.