**Web Programming Final Report**

**Numpy – Visualization**

Group Number: 6

Group Members: r05943094樊恩宇, 陳奕安

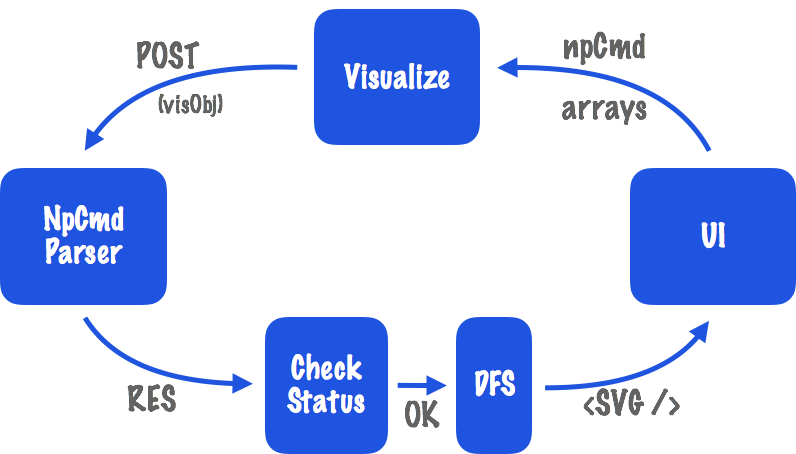
Contact: [rock15688@gmail.com](mailto:rock15688@gmail.com),

1. **Introduction**

For deep learning has gained so much attention in recent years and the Python package ‘numpy’ is widely used to cope with complex computation of multi-dimensional arrays, which is a significant part of deep learning. On grounds of that, we made a website that can visualize a numpy command readily as our Web Programming final project to make how numpy aids deep learning more explicit.

The current numpy command we support includes: identity, ones, zeros, ones\_like, zeros\_like, reshape, transpose, swapaxes, tril, triu, **+, -, x, /**. For detailed description of these commands, please refer to [Numpy](http://www.numpy.org/).

1. **System Structure**



User can define their own array as defining variables when writing programs, and specify a “npCmd” (numpy command) to be visualized. Then, this information will be convert to a “visObj”, having the form accepted by “NpCmd Parser”. The parser will response with an object which can be visualized by applying depth-first-search (DFS) on it if the object’s status is “OK”, otherwise, there will be an error message showed on the UI, for simplicity, this is not draw in the graph. Finally, an “SVG” component will be rendered onto the UI for visualization.

1. **Technique**
2. Frontend - UI:

We use “Semantic-UI-React”, which is the official React integration for Semantic UI. It is a very convenient and flexible package containing many useful features that help us quickly design a beautiful website. We used these components from “Semantic-UI-React” to build our user interface including Segment, Header, Breadcrumb, Grid, Table, Button, Input, Icon. For more information, please refer to [Semantic-UI](http://react.semantic-ui.com/introduction).

1. Frontend - Visualization:

For visualization part, we sent a “POST” request to the front end to get the well-parsed object. The responded object from backend has a nested structure with information about the command to be visualized. Each “node” inside the object has a property called “children”, which gives us the information to traverse through the object by DFS. Then, according to the order of the DFS traversal plus the “timeStep”, a state which will increment every 5 seconds when visualization starts, we render each node by “SVG”, a lightweight library for manipulating animation in javascript to complete the visualization. Simultaneously, the corresponding part of the command will be marked during the visualization. For more information about he animation package, please refer to [SVG](http://svgjs.com/).

For instance, “np.reshape(x, (1, 2))”, with x a defined array by user. Corresponding to the setting, the returned object will have a root node “np.reshape” with children “x” and “(1, 2)”. After DFS, we will get an array like this, [x, np.reshape], since (1, 2) is value, not array or command, so will be removed when performing DFS. Finally, visualization will be showed in accordance with the order of the array. Namely, creating “x” first, then reshape it.

1. Backend - Server & Parser:

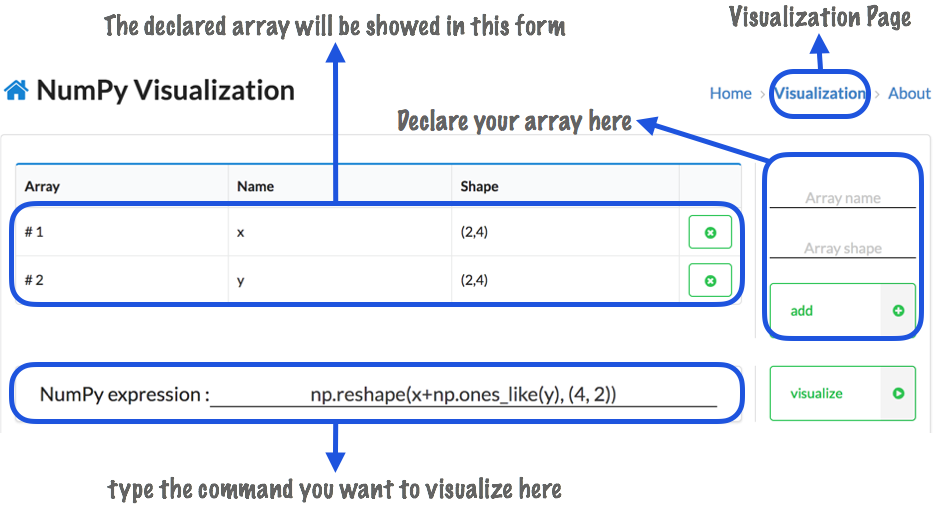
We use Python to write our backend for we want to utilize the “ast” and “asttokens” package in Python to parse the numpy command. To deal with the communication issue, we make use of “flask”, a web-development framework for Python. With just a few lines of code, cross-domain is settled. For more information, please refer to [Flask](http://flask.pocoo.org/docs/0.12/).

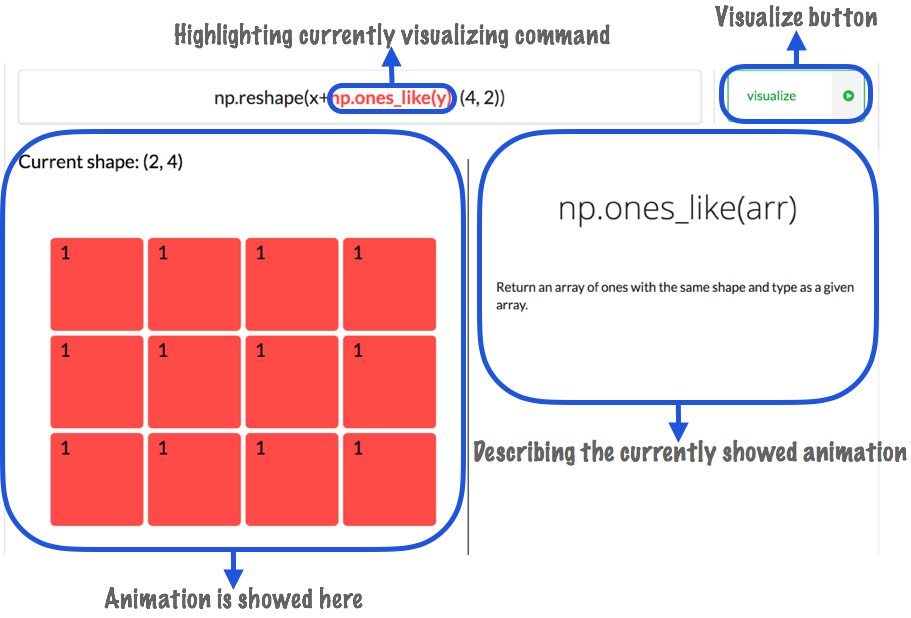
About parsing numpy command, we take advantage of built-in Python package “ast”, abstract syntax trees, a module helps Python applications to process trees of the Python abstract syntax grammar which help us to find out programmatically what the current grammar looks like. Furthermore, “asttokens”, a module annotates Python “ast” with the positions of tokens and text in the source code that generated them. This makes us capable of finding the particular text that resulted in those nodes, and highlighting that part of the command when visualizing. For more information, please refer to [AST](https://docs.python.org/2/library/ast.html) and [ASTTOKENS](https://pypi.python.org/pypi/asttokens).

1. **Usage**

The usage of our website is pretty easy. At the “Visualization” page, the user can define the array they want to use by declaring “name” and “shape” of the array on the right-hand side. Then they can specify the command to be visualized using the defined array. To be notice, we restrain array and command declaration to be exactly the same as using numpy in Python, because we want to make sure the users know that they are visualizing numpy not other package. After that, click the button visualize, the animation will begin at the below area. For more detailed usage, please refer to the home page of our website.

Following is some figure of our website:





1. Future Work
2. Add More Command:

Add more numpy command frequently used in deep learning application to be visualized or some more complicated command can be added in.

1. Extend Visualization:

Extend visualization to more than one command at a time. Perhaps with a coding like UI, where users can type multiple lines of command on and then visualize them one by one in order.

1. Extend to Keras Visualization:

Extend numpy visualization to Keras visualization, which is a more high-level package for deep learning, and is dependent to numpy a lot.

1. Linked with Database:

Maybe set up a login system for different users to organize their preferences on visualization.

1. Collaboration

樊恩宇: Frontend, Visualization(DFS), Report.

陳奕安: Backend, Visualization(SVG), Report.

1. Review
2. 樊恩宇 :

For me, this is the first time I expose myself to Web related course. Before this semester, I know nothing about the world of Web. Hence, this final is a great challenge for me, and I am very lucky to have such a powerful partner to collaborate with and learn from. I learned a lot from this project, like how to use “Semantic-UI-React”, more clear to relation between state and rendering…etc. Although these may sound like extremely fundamental stuff of web programming, this is my first “bigger” website, at least bigger than homework, so I gain many from it. Last but not the least, after all these courses, I think one of the most important thing is “testing”, since you never know what the user will do to your website whether or not you’ve specified all the rules. Especially when writing constraint to the inputs.

1. 陳奕安 :
2. Github URL

[Web Programming Final Repo](https://github.com/PatrickChen83/NumPyVisualization)