Visualizing Neural architecture search (NAS)

CSE 578 - Data Visualization

(Fall 2021)

I. INTRODUCTION

Visualizing a Neural Network Architecture (NAS), A project completed in course CSE 578 for the MSCS portfolio. The use of Neural Networks has become very popular these days in all sectors. Researchers working in these sectors are continuously trying to find the best Neural networks architecture for their applications by analyzing different models and validating accuracy metrics. So, building a dashboard for visualizing NAS will provide the users to steer the Neural Architecture Search and automate the process of generating the best model [2].

The idea of designing a dashboard with different visualizations for Machine learning models [3] already exists and was very successful. To which recent research and studies have been found to Steer One-shot Deep Neural Network Architecture [1] [4] where lies the inspiration of this project. The main goal of this project is to create a dashboard with different visualizations helping users to steer the Neural Architecture and select the best model Our team has created a steering dashboard with different plots like a Scatter plot, Line chart, Architecture view of the model, and summary table.

II. IMPLEMENTATION

Our team built the Visualization dashboard for Neural Network Architecture (NAS) using different visualization techniques learned in the course. The visualization starts with an initial architecture loaded and lets the user control the NAS process based on the two different datasets; (a) Random search; here the different layers in the architecture are added randomly (b) Evolutionary search; here the neural networks are added based on previously added layer. Based on every iteration the graphs and plots get updated and give the user an insight into whether the added layer was effective to the model. Datasets used and various plots designed are as follows.

Below is the dashboard created for Neural Architecture Search (NAS).

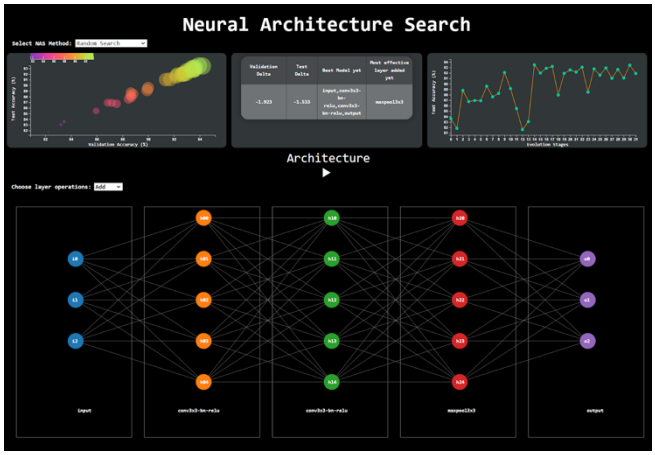

FIG.1. NEURAL ARCHITECTURE SEARCH (NAS) INTERFACE

This is the final view of our dashboard. The whole NAS system has been performed using a Random search dataset.

Dataset

The dataset for this project was a pre-trained neural architecture model having a number of layers, details of the layers like was the layer a convolution layer, max pool layer, convolution ReLU (Rectified Linear Unit), and many more, with its test and validation accuracy and a timestamp. As mentioned earlier there were 2 different datasets. These datasets were converted into JSON file format from H5 files using TensorFlow, an open-source software library for training machine learning and deep learning models.

	matrix	ops	time	best_valid	best_test	ops(str)
0	[[0, 1, 0], [0, 0, 1], [0, 0, 0]]	[input, conv3x3-bn-relu, output]	5.000066e+06	0.948918	0.941206	input conv3x3-bn-relu output
1	[[0, 1, 0], [0, 0, 1], [0, 0, 0, 1], [0,...	[input, conv3x3-bn-relu, conv1x1-bn-relu, output]	5.000167e+06	0.949519	0.942007	input conv3x3-bn-relu conv1x1-bn-relu output
2	[[0, 1, 1, 1, 1], [0, 0, 0, 1, 0], [0, 0, 0, 0...	[input, maxpool3x3, conv1x1-bn-relu, conv1x1-b...	5.000200e+06	0.950020	0.942007	input maxpool3x3 conv1x1-bn-relu conv1x1-bn-re...
3	[[0, 1], [0, 0]]	[input, output]	5.000234e+06	0.948918	0.941206	input output
4	[[0, 1, 0, 0, 1], [0, 0, 0, 0, 1, 1], [0, 0...	[input, maxpool3x3, conv1x1-bn-relu, maxpool3x...	5.000235e+06	0.951823	0.943610	input maxpool3x3 conv1x1-bn-relu maxpool3x3 co...
5	[[0, 1, 1, 1], [0, 0, 1, 1], [0, 0, 0, 1], [0,...	[input, conv3x3-bn-relu, maxpool3x3, output]	5.000274e+06	0.948918	0.941206	input conv3x3-bn-relu maxpool3x3 output
6	[[0, 1, 1, 1, 1], [0, 0, 0, 1, 0, 0], [0, 0...	[input, conv1x1-bn-relu, conv1x1-bn-relu, conv...	5.000303e+06	0.948918	0.941206	input conv1x1-bn-relu conv1x1-bn-relu conv1x1-...
7	[[0, 1, 0], [0, 0, 1], [0, 0, 0]]	[input, conv1x1-bn-relu, output]	5.000318e+06	0.948918	0.941206	input conv1x1-bn-relu output
8	[[0, 1, 1, 1], [0, 0, 1, 1], [0, 0, 0, 1], [0,...	[input, conv1x1-bn-relu, maxpool3x3, output]	5.000377e+06	0.951122	0.939603	input conv1x1-bn-relu maxpool3x3 output
9	[[0, 1, 0, 0, 0, 0], [0, 0, 1, 1, 0, 0], [0, 0...	[input, conv3x3-bn-relu, maxpool3x3, maxpool3x...	5.000424e+06	0.951122	0.939603	input conv3x3-bn-relu maxpool3x3 maxpool3x3 ma...
10	[[0, 1, 1, 0, 1], [0, 0, 0, 1, 1], [0, 0, 0, 1...	[input, conv1x1-bn-relu, maxpool3x3, maxpool3x...	5.000449e+06	0.949419	0.938802	input conv1x1-bn-relu maxpool3x3 maxpool3x3 ou...
11	[[0, 1, 1, 0, 1], [0, 0, 1, 0, 1], [0, 0, 0, 1...	[input, conv1x1-bn-relu, conv3x3-bn-relu, conv...	5.000478e+06	0.951823	0.943610	input conv1x1-bn-relu conv3x3-bn-relu conv3x3-...
12	[[0, 1, 0, 1, 1], [0, 0, 1, 1, 0], [0, 0, 0, 0...	[input, conv1x1-bn-relu, conv1x1-bn-relu, conv...	5.000481e+06	0.951122	0.939603	input conv1x1-bn-relu conv1x1-bn-relu conv3x3-...
13	[[0, 1, 1, 0, 0, 1], [0, 0, 0, 1, 0, 0], [0, 0...	[input, maxpool3x3, conv3x3-bn-relu, conv1x1-b...	5.000568e+06	0.949619	0.939904	input maxpool3x3 conv3x3-bn-relu conv1x1-bn-re...
14	[[0, 1, 1, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 1...	[input, maxpool3x3, maxpool3x3, maxpool3x3, ou...	5.000605e+06	0.951823	0.943610	input maxpool3x3 maxpool3x3 maxpool3x3 output
15	[[0, 1, 0, 1, 1, 1], [0, 0, 1, 0, 0, 1], [0, 0...	[input, conv1x1-bn-relu, conv3x3-bn-relu, conv...	5.000706e+06	0.951823	0.943610	input conv1x1-bn-relu conv3x3-bn-relu conv1x1-...
16	[[0, 1, 1, 0], [0, 0, 0, 1], [0, 0, 0, 1], [0,...	[input, conv3x3-bn-relu, conv3x3-bn-relu, output]	5.000819e+06	0.948918	0.941206	input conv3x3-bn-relu conv3x3-bn-relu output
17	[[0, 1, 1], [0, 0, 1], [0, 0, 0]]	[input, maxpool3x3, output]	5.001017e+06	0.948918	0.941206	input maxpool3x3 output
18	[[0, 1, 1, 0], [0, 0, 1, 0], [0, 0, 0, 1], [0,...	[input, conv1x1-bn-relu, conv1x1-bn-relu, output]	5.001124e+06	0.950020	0.942007	input conv1x1-bn-relu conv1x1-bn-relu output
19	[[0, 1, 1, 1], [0, 0, 1, 1], [0, 0, 0, 1], [0,...	[input, maxpool3x3, conv1x1-bn-relu, output]	5.001138e+06	0.948918	0.941206	input maxpool3x3 conv1x1-bn-relu output
20	[[0, 1, 1, 1, 1], [0, 0, 0, 1, 0, 0], [0, 0...	[input, maxpool3x3, maxpool3x3, conv1x1-bn-rel...	5.001341e+06	0.948618	0.941206	input maxpool3x3 maxpool3x3 conv1x1-bn-relu co...

FIG.2. DATASET

Architecture

This is the main visualization in the dashboard which shows the actual model architecture based on different layers added. This type of visualization consists of a node linked with many other nodes representing a neuron.

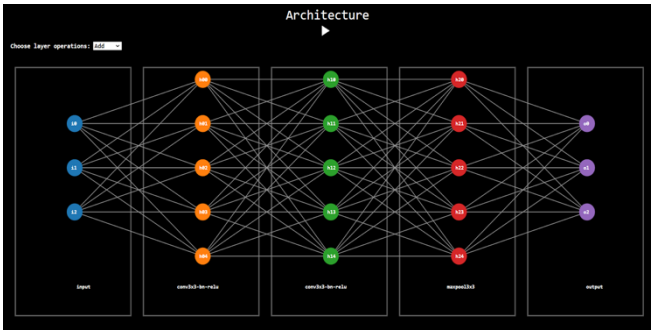

FIG.3. MODEL ARCHITECTURE

Scatter plot

The scatter plot in the dashboard focuses on the performance of different Architecture models based on their Validation and Test accuracy. This plot can be used to see what model performances look like and the user can also see the architecture view of the model by clicking over it.

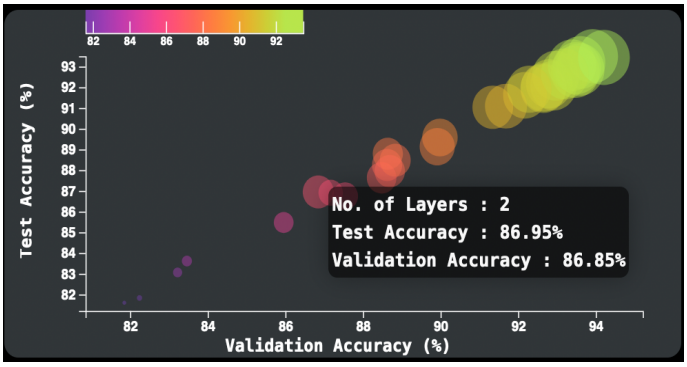

FIG.4. SCATTER PLOT

As seen from the above image, we can see that the circle on the scatter plot differs in size, as they are in increasing order of the test accuracy. On hovering the circle, a tooltip appears with all the details of the architecture.

Line Chart

The line chart shows the accuracy of different model architectures as the NAS progresses. This gives the user insight into the trend of performance of the models added so far. On adding which layer did the line graph was raised to the peak.

FIG.5. LINE CHART

The user on clicking to any of the dots can view the architecture. It also shows the tooltip on hovering.

Summary Table

This is a simple table view showing metrics that changes when any new layer is added. It shows the difference in the accuracy of the current model with the previously added layer. This is very helpful to the user in giving an insight on which layer should be added to the architecture upon which it yields maximum accuracy.

Validation Delta	Test Delta	Best Model yet	Most effective layer added yet
0.140	0.621	input,maxpool3x3,conv3x3-bn-relu,maxpool3x3,output	maxpool3x3

FIG.6. SUMMARY TABLE

III. RESULTS

With the help of the different plots described, our Neural Architecture Search (NAS) dashboard is very interactive and helpful to the users who mostly struggle with finding the right combination of the neural layers which will yield the best results based on their application. Giving the user an option to play the whole NAS search is very insightful and thoughtful on how different plots changes with change in every layer and change in datasets. Not only this, but our dashboard also facilitates human intervention as well. Like anytime a user can pause the search and manually add or delete the layer and see what effect it has brought on the overall model.

Some interesting findings while running the NAS system on our Random dataset, we found that adding a max pool layer in architecture is the most effective. Also, the architecture with 4 layers having a max pool layer outperforms with a test accuracy of 93.44 %.

IV. CONTRIBUTION

My contribution in this project was to design the scatter plot for two different datasets and make it interactive based on different animations[6]. So, I created the scatter plot which will be drawn based on the dataset selected. To create this visualization, I have used D3, HTML, CSS and Javascript [5]. The two axes in the scatter plot are validation and test accuracy present in the dataset. So, based on this a circle will be plotted which represents architecture. Not only this, the size of the circle in the plot varies based on the test accuracy. The circle with a bigger size indicates better architecture. On top of this, interaction is added to the plot; on hovering to the circle, the size of the circle enlarges and a tooltip appears with the details of the specific architecture.

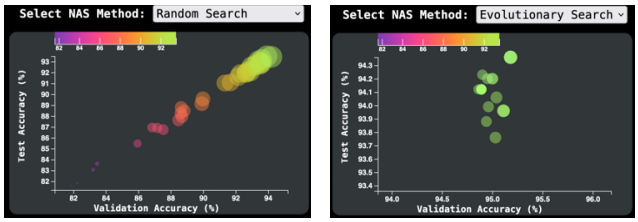

FIG.7. SCATTER PLOT FOR (A) RANDOM SEARCH (B) EVOLUTIONARY SEARCH

Along, with this, I majorly contributed to giving styling to our dashboard using CSS and integrating the Scatter plot in our system. I created the Scatterplot alone first, later integrating it with other plots in the dashboard was a complex task, as all the plots had used a common SVG to draw. Not only this, I had to scale the scatter plot to fit into a given space.

V. LESSONS LEARNED

Implementing this project helped me better understand the use of different visualization techniques like D3, Javascript, HTML, and CSS. Also, the topic in itself is a very vast and popular topic in today's time. I got to learn new concepts like how a Neural Network works and how important is to steer the whole process. Also, adding visualization to such a system is very insightful in the real world and has huge applications. Other than this, I got practical hands-on in developing a front-end using Javascript. Till now, I had only used Python and different libraries to deal with projects related to Machine Learning and Deep Learning. But this time I got an opportunity to work in this field with all different and new perspective. Also, working in a team helped me develop my teamwork skills and coordination skills. Also, to implement new features like animation, interactions, I did a lot of research, referred to implementation of previous papers and some experiments on my own. Apart from this, from an academic point of view, I got to learn how to document the project with the findings and results in a form of a technical report.

VI. TEAM MEMBERS

This project was completed by Abhineeth Mishra, Darshil Shah, Eric Murari, Fenny Zalavadia, Pravar Parekh, and Shrutwa Shah under the guidance of the mentor Jinbin Huang and Prof. Chris Bryan at Ira. Fulton School of Engineering at Arizona State University.

REFERENCES

[1]. Tyagi, Anjul, Cong Xie, and Klaus Mueller. "Visual Steering for One-Shot Deep Neural Network Synthesis." *arXiv preprint* arXiv:2009.13008 (2020).

[2]. Wang, Qianwen, et al. "Atmseer: Increasing transparency and controllability in automated machine learning." *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019.

[3]. Chatzimparmpas, Angelos, et al. "The state of the art in enhancing trust in machine learning models with the use of visualizations." *Computer Graphics Forum*. Vol. 39. No. 3. 2020.

[4]. Cashman, Dylan, et al. "Ablate, variate, and contemplate: Visual analytics for discovering neural architectures." *IEEE transactions on visualization and computer graphics* 26.1 (2019): 863-873.

[5]. Scatterplot with tooltip in d3.js - https://www.d3-graph-gallery.com/graph/scatter_tooltip.html

[6]. D3 Scatterplot Animations - http://bl.ocks.org/WilliamQLiu/raw/bd12f73d0b79d70bfbae/?raw=true