

Keras Without Keras

Chu-Sheng Ku, Hansol Yoon, Si Shen, Ganesh Chandra Satish, Rahul Chowdhury

Introduction:

Motivation:

Deep Learning has been used in a cross variety of applications since it was introduced. However, for those who have not yet learned deep learning, deep learning is always considered as future work although they would like to apply it to their own research. Deep learning can also be a difficult learning curve if the user doesn't have prior knowledge and expertise in programming and machine learning skills^[1]. Even if they understand what it is, people without coding experience with Tensorflow or Keras cannot build usable neural net models. Deep learning is just one of the famous techniques which they cannot use. Also, it is impossible to encourage middle school or high school students to use or understand deep learning. The biggest problem in this situation is that there is no testbed that anyone can test deep learning to see what results they will get if they use their own data even if they do not know how to code in Python. That's why there is not enough opportunity to get to know about the benefits of deep learning, and it is difficult to motivate people to learn deep learning. Tensorflow offers a simple demonstration that shows people how deep learning works. In the TensorBoard Neural Network Playground^[2], people can see visually how deep Learning solves the classification or regression problem by changing the number of hidden layers, the number of neurons, and the activation function. The user can change the options such as learning rate and regularization methods by clicking the mouse without coding. But the problem is that users can not use their own data, only the data given at the site is available. Also, the number of layers available, the number of neurons are limited.

The other use case that we are hoping to for is the usage of deep learning models as a tool similar to jmp. JMP^[3] is a statistical tool used by many companies for data analysis. We are hoping to create a deep learning tool that can be used for analyzing data on a practical scale.

Our proposal:

We propose a deep learning framework called '*Keras without Keras*' which expands the Neural Network Playground so that the user can put in their own data and check the results by setting various options without coding. Using this framework, users can build up a network by building hidden layer blocks without knowing any coding and use the various functions of Keras just by specifying a few options with the mouse. Students can see the network visually and researchers will be motivated to study deep learning more deeply if they get nice results using their own

data. And companies will be motivated to use this tool to develop deep learning models for commercial use.

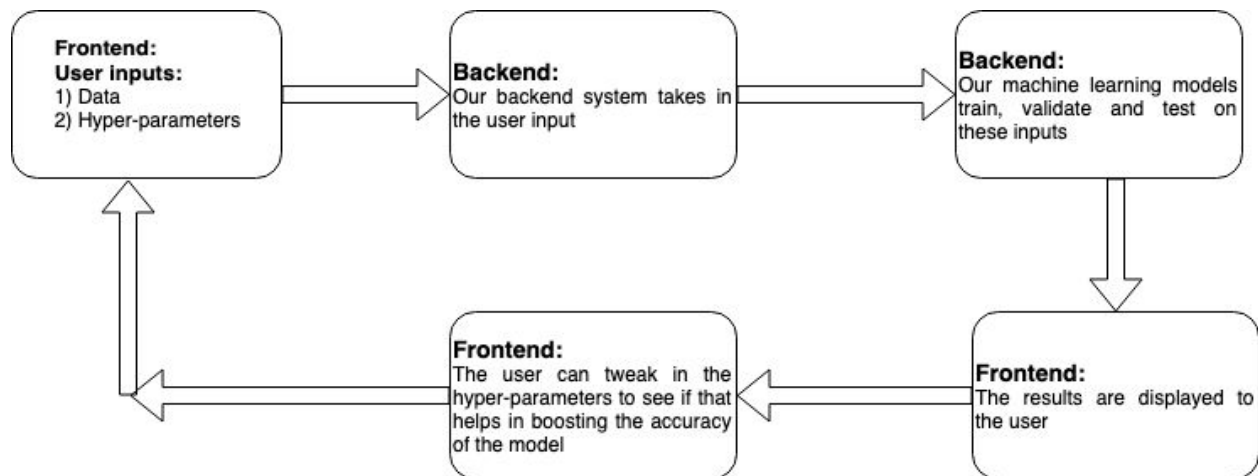


Fig 1: A high-level architecture of our system.

Feasibility:

In class, we learned how to create a Tensor framework and train neural networks by the forward and backward architecture on top of this framework. In addition, we learned two useful network architectures, CNN and RNN. Currently, Tensorflow's Neural Network Playground provides a feed-forward network, and we want to add CNN, RNN and so on. The goal is to provide more options than the Neural Network Playground, starting from the feed-forward network, and adding CNN and RNN if the project period is sufficient. It is not a problem to make a backend because we learned the theoretical part of the class. In the case of frontend such as UI, we can use Python's UI library or React JS to create it. In Tensorflow's Neural Network Playground uses Javascript. By comparing various UI development tools, we will select a better language (tool) considering running time and how convenient it is for users.

Proposed Approach:

Backend: We will use Keras. Users can bring their own data and build a deep learning network using blocks like playing with LEGOs. When a user determines the number of hidden layers and neurons using blocks, the corresponding Keras code is entered in the backend. That is, when a user put five neurons in one hidden layer, Keras code is automatically generated like "model.add(Dense(5))." Many options that Keras has including choosing activation functions, Dropout rate or initialization methods are available.

Frontend: We will choose the best methods to make a UI. The first way is to extend the Javascript code used in the Neural Network Playground (<https://github.com/tensorflow/playground>). We can set the Neural Network Playground as a

baseline, add the necessary functions and remove unnecessary functions. In case the application is implemented on the website, it is possible to build UI and Server using React JS and Node JS. Other visualization, GUI and web development libraries in python will be investigated (like D3.js, matplotlib, Flask, Django), and we will choose the most user-friendly language or tool. **Deployment:** We will use AWS or Google Cloud for the demo. Our main idea is to provide an installable light system on the user laptop and installing the backend on the same system or a cloud platform of their choice.

References:

- [1] <http://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html>
- [2] <https://playground.tensorflow.org/>
- [3] https://www.jmp.com/en_us/home.html