

Democratization of Deep Learning Using DARVIZ

Anush Sankaran,¹ Naveen Panwar,¹ Shreya Khare,¹ Senthil Mani¹
{anussank, naveen.panwar, shkhare4, sentmani}@in.ibm.com

Akshay Sethi,^{2*} Rahul Aralikkatte,¹ Neelamadhav Gantayat¹
akshay14133@iiitd.ac.in, {rahul.a.r, neelamadhav}@in.ibm.com
¹IBM Reseach ²IIIT Delhi

Abstract

With an abundance of research papers in deep learning, adoption and reproducibility of existing works becomes a challenge. To make a DL developer life easy, we propose a novel system, DARVIZ, to visually design a DL model using a drag-and-drop framework in an platform agnostic manner. The code could be automatically generated in both Caffe and Keras. DARVIZ could import (i) any existing Caffe code, or (ii) a research paper containing a DL design; extract the design, and present it in visual editor.

Introduction

In the recent years, deep learning algorithms have played a crucial role in pushing the boundaries of artificial intelligence (AI). These algorithms have shown out of the box successful results in various tasks such as automatic speech recognition, image recognition, natural language processing, and recommendation systems. Deep learning algorithms use a cascade of many layers to transform the raw data and perform feature extraction in an unsupervised manner. To easily enable implementation of such algorithms, there are many deep learning libraries such as like Caffe, Tensorflow, and Torch based out of different programming languages. The variety of deep learning libraries implies that each library have their own syntax, design principles, and merits. These libraries have minimum communication and interoperability across them, making the code and model reproducibility minimum. Thus, software engineers are limited in designing applications and are also hindered in reusing existing public implementations (Sankaran et al. 2017).

To address these challenges, we propose a novel framework called DARVIZ: Deep Abstract Representation, Visualization, and Verification. This framework is made publicly available at <http://darviz.mybluemix.net>. As shown in Figure 1, the primary features of DARVIZ are:

1. Platform agnostic “no-code” visual designing of deep learning models
2. Easy interoperability across various deep learning platforms by converting code from one library to another

* Akshay Sethi was working as an intern in IBM Research, India during his contribution to this project.
Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

3. Extract information from deep learning design flow diagrams and tables available in a research paper
4. Static validation and verification of deep learning designs.

Key Features

Visual IDE for Deep Learning Design

Deep learning model designs are graph-like structures having a set of layers as nodes and each node having a set of hyper-parameters. DARVIZ provides an intuitive drag-and-drop UI, where a user can construct the complete network using basic units like layers. While the visual coding environment is not completely new to the community with the likes of *SubGraphs*¹, *Machine UI*², *Fabrik*³, DARVIZ is one of the oldest and highly flexible platform built over the open-source *Node-Red*⁴ library. DARVIZ provides support to 23 distinct layers that can be used to build complex deep learning networks, processing both text and image data. The designed deep network design can be translated to an execution ready deep learning code in Caffe and Keras (Tensorflow, Theano). The network designed is converted to an abstract computational graph using platform-specific comprehensive rule base. The inference engine consists of a comprehensive list of templates and dictionaries built manually for every library. These template-based structures transfer each component from the abstract representation into a platform-specific structure using dictionary mappings. Further, this inference engine is extremely flexible, permitting easy extension and addition of new layer definitions and new libraries. A dictionary mapping can be written to support a new layer converting the abstract representation to the library specific code. For instance, one of the deep learning models used in computer vision called VGG19 network (Chatfield et al. 2014), could be designed using the drag-and-drop UI within minutes.

DARVIZ also enables data handling and manipulation using the user interface. Data in different formats and structures can be pre-processed by providing specifications in the

¹<https://subgraphs.com/#/>

²<https://machineui.co/>

³<http://fabrik.cloudcv.org/>

⁴<https://nodered.org/>

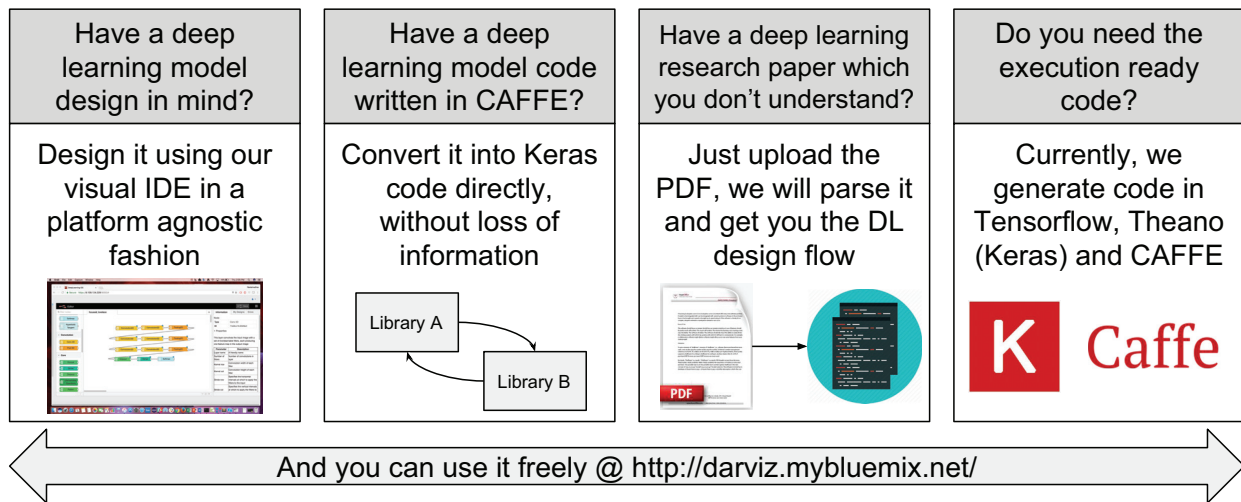


Figure 1: The primary features and use of the proposed DARVIZ system.

UI. Another common challenge developers face is debugging their implemented deep learning code for logical errors. Often, design errors are identified after hours of GPU training. To overcome this, DARVIZ performs real-time static validation of the model reducing error probability of the model, significantly. For example, if a network contains a series of operations which reduces the image size significantly (below zero), the system prompts user at that particular set of layers and also provides constructive suggestions.

Interoperability across Libraries

Deep learning model implementations are typically available in a particular library. DARVIZ enables easy code conversion of the design code from one library to another. Currently, any code available in Caffe (design and solver prototxt) could be easily be converted to a Keras (Tensorflow, Theano) code. This easy conversion of deep learning models designed in one library to another adds to the efficiency of the developers, considerably. DARVIZ performs interoperability by creating an abstract universal schema and platform-specific inference engine. A platform-specific inference engine consists of templates and dictionaries, which maps code generated in a particular platform to the abstract universal schema. This follows a popular principle of model-driven development and thus supporting other libraries could be easily extended by adding importers and exporters.

Research Paper to Code

DARVIZ also provides the capability to understand deep learning models available in research paper in the form of figures and tables. Given a research paper, DARVIZ extract figures and tables providing the architecture details, parses them to generate the abstract computational graph (Sethi et al. 2018). After extraction of tables and figures from the research paper, the entire design is provided in the visual drag-and-drop framework. The users could edit the design and further generate the execution ready source in both Caffe and Keras.

Due to the lack of ground truth availability, evaluating such a feature is a huge challenge. Thus, we created a sophisticated grammar using which a dataset of more than 216,000 valid deep learning models was generated along with its source code in both Caffe and Keras. As both these libraries provide visualizations, extracting the deep learning design from the image was more than 99% for Keras images and more than 92% for the Caffe images.

Potential Applications

The easy-to-use DARVIZ framework has multiple applications where it could be potentially used in many ways:

1. **Rapid Prototyping:** In industries, the large group of developers could use DARVIZ as a rapid prototyping tool for designing complex deep learning architectures and building applications over them.
2. **Teaching Tool:** In academic setting, DARVIZ could be used as a classroom teaching tool, to enable students visually learn how to design deep learning models.
3. **Open Research:** New research papers could make their design available in a platform agnostic DARVIZ format. The community could then generate the source code for the design in the library and language of their choice.

References

Chatfield, K.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

Sankaran, A.; Aralikkatte, R.; Mani, S.; Khare, S.; Panwar, N.; and Gantayat, N. 2017. Darviz: deep abstract representation, visualization, and verification of deep learning models. In *ICSE-NIER Track*, 47–50. IEEE Press.

Sethi, A.; Sankaran, A.; Panwar, N.; Khare, S.; and Mani, S. 2018. DLPaper2Code: Auto-generation of code from deep learning research papers. In *AAAI Conference on Artificial Intelligence*.