DARVIZ: A Visually IDE to build Deep Learning Models

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

Akshay Sethi[2*], **Rahul Aralikatte**[1], **Neelamadhav Gantayat**[1]
akshay14133@iiitd.ac.in, {rahul.a.r, neelamadhav}@in.ibm.com
[1]IBM Reseach [2]IIIT Delhi

Abstract

With an abundance of research papers in deep learning, adoption 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 powerful 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

The growth of artificial intelligence (AI) has challenged and even outperformed the human capabilities in recent times due to the emergence of deep learning algorithms. These algorithms have shown out of the box successful results in various tasks such as automatic speech recognition, image recognition, natural language processing, recommendation systems and, etc.

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, Torch, etc based out of different programming langauges. The variety of deep learning libraries means that each library have their own syntax, design principles and merits. Further, 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 also hinder resuability of existing public code 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. The primary features of DARVIZ as shown in Figure 1 are:

1. Platform agnostic no-code designing of deep learning models.
2. Easy interoperability across various deep learning platforms by converting execution ready code.
3. Extract and code-generation for deep learning design flow diagrams and tables available in a research paper.
4. Validation and verification of deep learning designs.

Key Features

Visual IDE for DL Design

Deep learning models are graph-like structures having a set of predefined layers and hyper-parameters, driven by data. DARVIZ provides an intuitive drag-and-drop UI, where a user can construct the complete network using basic units like layers. DARVIZ provides support to 23 distinct layers such as Convolution2D, Dense, Text Data, Accuracy. These layers can be used to build huge deep learning networks processing both text and image data. The deep network built can be translated to any of the publicly available deep learning platforms like Caffe, Tensorflow, Theano etc. A network designed is converted to an abstract computational graph using platform-specific comprehensive rule base. This is done using an inference engine that acts as a converter to map the basic units (layers) using the grammar of the platform. The inference engine consists of a comprehensive list of templates and dictionaries built manually for platforms. 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.

DARVIZ also enables data handling and manipulation using the user interface. Data in any format and structure can be pre-processed based on the models within a few clicks. Consider one of the deep learning models used in computer vision called VGG19 network (Chatfield et al. 2014). Typically in Tensorflow, this constitutes 400 lines of code (LOC) and takes a few hours to days to implement, based on the developer's expertise. However, the same model could be designed using the drag-and-drop UI within minutes.

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,

[*]Akshay Sethi was working as an intern in IBM Research, India during his contribution to this project.

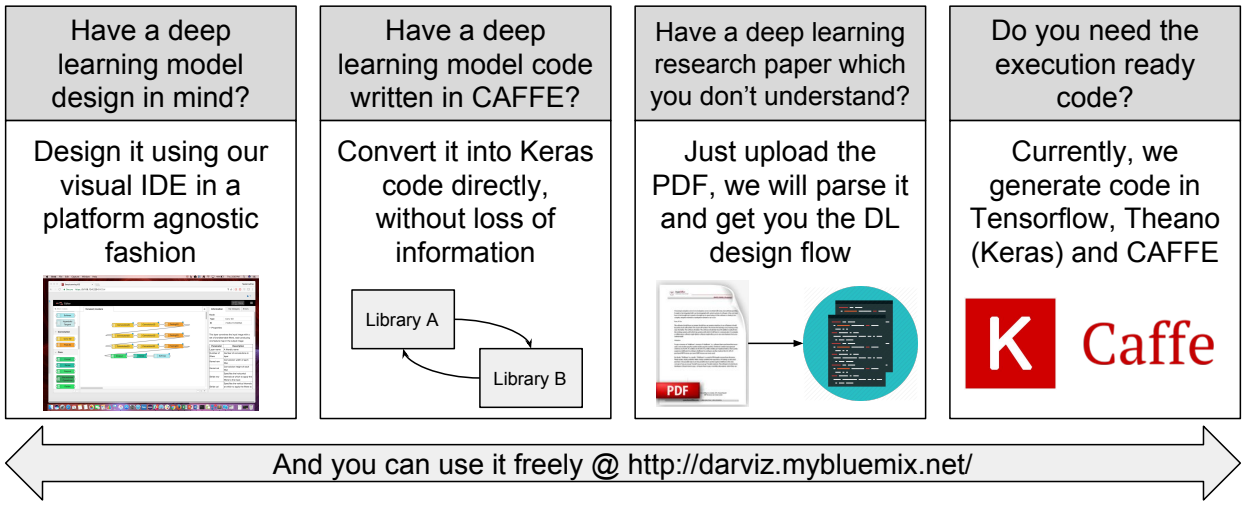

Figure 1: The primary features and use of the proposed DARVIZ system.

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. This real-time suggestion is performed by translating the knowledge and efforts of researchers in the deep learning community to a comprehensive list of rules.

Interoperability across Libraries

Researchers working in deep learning tend to release their code-base implemented in a particular library. DARVIZ enables easy code conversion of a deep learning design from one library to another. Currently, any code available in Caffe (design and solver prototxt) could be easily be converted to a Keras (Tensorflow) code. This easy importing of deep learning models designed in one platform to another adds to the efficiency of the developers, considerably. DARVIZ performs interoperability by creating a 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. Given a Caffe based deep learning network, the Caffe specific inference engine translates the code to the DARVIZ's abstract universal schema, as mentioned previously, this schema then gets realized to Keras code using a Keras based inference engine.

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. This computational graph can then be converted to platform-specific code. After extraction of tables and figures from the pdf of the paper, it provides the entire design in the visual drag-and-drop framework. The users could edit the design and further generate the 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 create a sophisticated grammar that defines deep learning models. A dataset of more than 185,000 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.; Aralikatte, R.; Mani, S.; Khare, S.; Panwar, N.; and Gantayat, N. 2017. Darviz: deep abstract representation, visualization, and verification of deep learning models. In *Proceedings of the 39th International Conference on Software Engineering: New Ideas and Emerging Results Track*, 47–50. IEEE Press.