Pythia - A platform for vision & language research

Amanpreet Singh, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah,
Marcus Rohrbach, Dhruv Batra and Devi Parikh
Facebook AI Research

Abstract

This paper presents Pythia, a deep learning research platform for vision & language tasks. Pythia is built with a plug-&-play strategy at its core, which enables researchers to quickly build, reproduce and benchmark novel models for vision & language tasks like Visual Question Answering (VQA), Visual Dialog and Image Captioning. Built on top of PyTorch, Pythia features (i) high level abstractions for operations commonly used in vision & language tasks (ii) a modular and easily extensible framework for rapid prototyping and (iii) a flexible trainer API that can handle tasks seamlessly. Pythia is the first framework to support multi-tasking in the vision & language domain. Pythia also includes reference implementations of several recent state-of-the-art models for benchmarking, along with utilities such as smart configuration, multiple metrics, checkpointing, reporting, logging, etc. Our hope is that by providing a research platform focusing on flexibility, reproducibility and efficiency, we can help researchers push state-of-the-art for vision & language tasks.

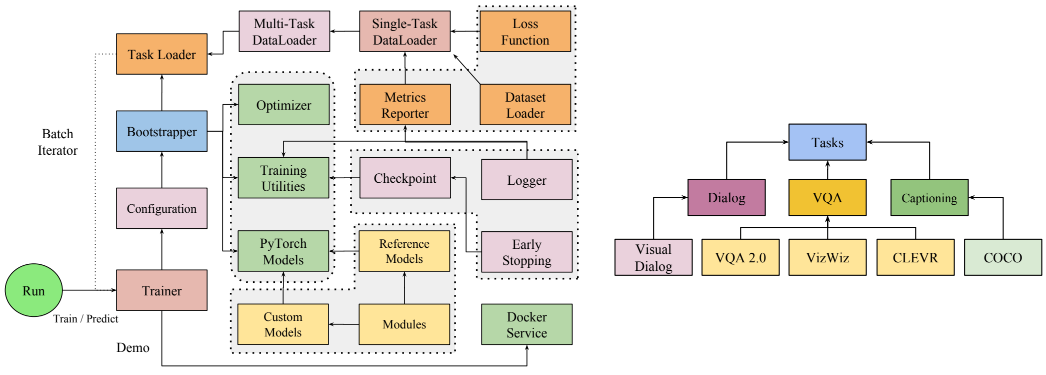
Over the last few years, we have seen impressive progress in vision & language tasks like Visual Question Answering (VQA) and Image Captioning powered by deep learning. Most of the state-of-the-art networks build upon the same techniques for generating the representations of text and images and for the network's layers. However, the devil lies in the details, hence reproducing results from the state-of-the-art models has often been non-trivial. This in-turn hinders faster experimentation and progress in research. With Pythia[1], we hope to break down these design, implementation and reproducibility barriers by providing a modular and flexible platform for vision & language (VQA and related) tasks' research ([10][6][14]) which in turn enables easy reproducibility and fosters novel research by taking care of low level details around IO, tasks, datasets and model architectures while providing flexibility to easily try out new ideas. Pythia is built on top of the winning entries to the VQA Challenge 2018 and Vizwiz Challenge 2018. Pythia includes a set of reference implementations of some current state-of-the-art models for easy comparison[2]. We derive inspiration from software suites like AllenNLP [8], Detectron [9], and ParlAI [16] which aim to break similar barriers in other machine-learning domains like natural language processing and computer vision.

Framework Design: In Pythia (refer Figure 1a), we have a central *trainer* which loads a *bootstrapper* which sets up components required for training. *Bootstrapper* builds a model based on the network configuration provided by the researcher. For loading the data, *bootstrapper* instantiates task loader which can load multiple tasks based upon the configuration. Pythia works on a plugin based registry where tasks and models register themselves to a particular key in the registry mapping. Furthermore, the datasets register themselves to one or more tasks. This registry helps in dynamic loading of models and tasks at runtime based on configuration. A task first builds, if not present, and then loads the datasets registered to it. A dataset is responsible for its metrics, logging and loss function, thus, keeping the trainer agnostic to the data details. See Figure 1b for a tree overview of tasks (second

[1] A preliminary version of Pythia (v0.2) is available at https://github.com/facebookresearch/pythia. Note that, v0.3, which is described in this abstract will be open-sourced soon.

[2] We plan to release pre-trained models for these implementations for easy comparisons in v0.3.

Preprint. Work in progress.



(a) Architecture

(b) Tasks

Figure 1: Overview of Pythia. (a) shows the architecture of Pythia's framework. Pythia's trainer loads the configuration and bootstrapper which loads the rest of the components. Task loader loads the required tasks with their datasets, loss function and metrics reporters. Bootstrapper also takes care of loading models, utilities and optimizer. (b) shows various tasks and their datasets present in Pythia - (i) VQA (ii) Dialog and (iii) Captioning. We plan to add more tasks in future.

level) and their datasets (third level). The central *trainer* delegates to appropriate components for performing the main model's training (with early stopping) and evaluation. These delegations, along with task loader and other abstractions, allows the trainer to have a generic signature and be agnostic to the data-loaders, utilities and models and be invariant across tasks.

Model Zoo: We provide a set of reference models for benchmarking which includes Bottom-Up-Top-Down [1], VQA Challenge winning entry [11] and BiLinear Attention Networks [13]. We also include implementations of common representation generation techniques and network layers like word embeddings (GloVe [17], ELMo [18]), sentence embeddings (InferSent [5][20]), attention [2], encoders and decoders [4][21] in the modules subpackage. This would help in reproducing existing results as well as implementing novel models without having to worry about details and testing of low level components.

Datasets: We package several standard ready-to-use datasets such as VQA 2.0[10], Visual Dialog [6], VizWiz [3], COCO [14] and CLEVR [12] in Pythia. We further provide standard interfaces for QA, captioning and dialog based datasets which can be inherited for new datasets to get common processing utilities and straight-forward integration with models included in Pythia.

Multi-Tasking: Recently, with the introduction of several multi-task benchmarks [22][15], there has been significant progress in field of natural language processing [19][7]. We would like to achieve similar advances in vision & language tasks through Pythia. To the best of our knowledge, Pythia is the first framework to support multi tasking in the vision & language domain. For enabling multi-tasking in Pythia, we (i) ground input from all datasets to a common signature which covers most of the tasks - $(image, context, text, output)$ (ii) For a single batch, we sample only from a particular task and a particular dataset in that task (iii) We abstract the loss function and metrics inside the task itself. This ensures that the trainer is agnostic to dataloaders, batching can be efficiently used and models with different output heads (discriminator vs generator) can be easily trained jointly. For VQA [10], the $context$ will be empty, $text$ will be the question and $output$ is from dataset's answer vocabulary. In Visual Dialog [6], the $context$ is the history of the dialog. In the captioning task [14], $context$ will be empty and $text$ becomes the word "caption". This allows use of the same high level abstractions and signatures for these tasks while keeping the trainer agnostic to the low level details.

Future: Pythia is an ongoing effort by Facebook AI Research (FAIR). We plan to continue adding more tasks, datasets, reference models and modules. We are also building tools and utilities for comparison, visualization, debugging and optimization. We believe that Pythia will lead to easy benchmarking, large scale training and faster novel research. To improve support for the community and enhance usage, we also plan to publish getting started guides, documentation and tutorials for using and building on top of Pythia.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342. ACM, 2010.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[6] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.

[9] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

[12] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

[13] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *arXiv preprint arXiv:1805.07932*, 2018.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

[16] Alexander H Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*, 2017.

[17] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[19] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*, 2018.

[20] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*, 2018.

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[22] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.