Demonstration of Krypton: Incremental and Approximate Inference for Faster Occlusion-based Deep CNN Explanations

Supun Nakandala [1] **Arun Kumar** [1]

Abstract

In this demonstration we present KRYPTON, a system for accelerating occlusion based deep convolution neural network (CNN) explainability workloads. Driven by the success of CNNs in image understanding tasks, there is growing adoption of CNN technology in various domains including high stake applications such as radiology. However, users of such applications often seek an "explanation" for why a CNN predicted a certain label. One of the widely used approaches for explaining the CNN predictions is the occlusion based explainability (OBE) method. This approach is computationally expensive due to the large number of re-inference requests produced. In this demo we present KRYPTON , a system that reduces the runtime of OBE by up to 35x by enabling incremental and approximate inference optimizations. We allow the audience to interactively diagnose CNN predictions from several use cases including radiology and natural images.

1 Introduction

Deep convolution neural networks (CNNs) are now the state of the art method for many image prediction tasks. Thus, there is growing interest in adopting deep CNNs in various application domains, including high stake applications such as radiology (Kermany et al., 2018). Despite their successes, a key criticism of CNNs is that their internal workings are unintuitive to non-technical users. Thus, users often seek an "explanation" for why a CNN predicted a certain label. How to explain a CNN prediction is still an active research question, but in the practical literature, an already popular mechanism for CNN explanations is a simple procedure called *occlusion-based explanations*, or OBE for short.

OBE works as follows. Place a small patch (usually gray or black) on the image to occlude those pixels. Rerun CNN inference on the occluded image. The probability of the predicted label will change. Repeat this process by moving the patch across the image to obtain a sensitivity *heat map* of the probability changes. This heat map will highlight regions of the image that were highly sensitive or "responsible" for the prediction (see red/orange color regions in Figure 1). Such *localization* of the regions of interest allows users to gain intuition on what "mattered" for the CNN prediction.

However, OBE is highly computationally expensive. Deep CNN inference is already expensive; OBE just amplifies it by issuing a large number of CNN re-inference requests. Such long wait times can hinder users' ability to consume explanations and reduce their productivity.

KRYPTON uses a database-inspired lens to formalize, optimize, and accelerate OBE. We start with a simple but crucial observation: *the occluded images are not disjoint but share most of their pixels; so, most of CNN re-inference computations are redundant.* Instead of treating a CNN as a "blackbox," we open it up and formalize *CNN layers* as "queries." Just like how a relational query converts relations to other relations, a CNN layer converts *tensors* (multidimensional arrays) to other tensors. So, we reimagine OBE as *a set of tensor transformation queries* with incrementally updated inputs.

KRYPTON is implemented on top of popular deep learning framework PyTorch. It works on both CPU and GPU and currently supports a few popular deep CNNs. KRYPTON can enable up to 35X speedups over the current dominant practice of running re-inference with just batching for producing high-quality approximate heat maps and up to 5X speedups for producing exact heat maps.

2 Technical Novelty

The novelty of our system comes from the optimization techniques that it uses for accelerating the OBE workload. Our first optimization is *incremental CNN inference*. We *materialize* all tensors produced by the CNN's layers on the given image. For every re-inference request in OBE, instead of rerunning CNN inference from scratch, we treat it as an incremental view maintenance (IVM) query, with the "views" being the tensors. We rewrite such queries to *reuse* as much of the materialized views as possible and recompute only what is needed, thus *avoiding computational redundancy*. Such rewrites are non-trivial because they are

[1] University of California, San Diego. Correspondence to: Supun Nakandala <snakanda@eng.ucsd.edu>.

Proceedings of the 2^{nd} SysML Conference, Palo Alto, CA, USA, 2019. Copyright 2019 by the author(s).

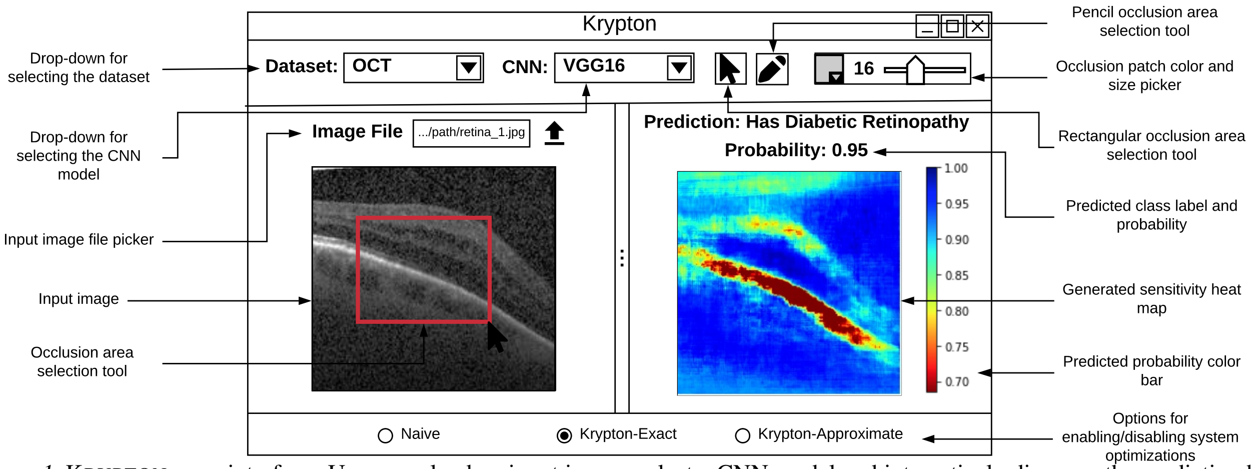
Krypton
Dataset: OCT
CNN: VGG16
16
Image File .../path/retina_1.jpg
Prediction: Has Diabetic Retinopathy
Probability: 0.95
1.00
0.95
0.90
0.85
0.80
0.75
0.70
○ Naive
● Krypton-Exact
○ Krypton-Approximate
Drop-down for selecting the dataset
Drop-down for selecting the CNN model
Input image file picker
Input image
Occlusion area selection tool
Pencil occlusion area selection tool
Occlusion patch color and size picker
Rectangular occlusion area selection tool
Predicted class label and probability
Generated sensitivity heat map
Predicted probability color bar
Options for enabling/disabling system optimizations

Figure 1. KRYPTON user interface. Users can load an input image, select a CNN model and interactively diagnose the prediction by occluding parts of the image using brush or selection tool (left image). KRYPTON generates a sensitivity heat map (right image) and iteratively refines it as the user progresses.

closely tied to the complex geometric dataflows of CNN layers. We have formalized such dataflows to create an *algebraic framework* of CNN query rewrites. Going further, we batch all re-inference requests in OBE to reuse the *same* materialized views. This is a form of multi-query optimization (MQO), albeit interwoven with our IVM, leading to a novel *batched incremental CNN inference* procedure. To the best of our knowledge, this is the first instance of IVM being fused with MQO in query optimization, at least for CNN inference.

KRYPTON also performs two novel *approximate inference* optimizations that allow users to tolerate some degradation in visual quality of the heat maps produced to reduce runtimes further. These optimizations build upon our incremental inference optimization to trade off heat map quality in a user-tunable manner. Our first approximate optimization, *projective field thresholding*, draws upon an idea from neuroscience and exploits the internal semantics of how CNNs work. Our second approximate optimization, *adaptive drill-down*, exploits the semantics of the OBE task and the way users typically consume the heat maps produced. We also present intuitive automated parameter tuning methods to help users adopt these optimizations. More details on KRYPTON optimizations can be found in our Technical Report (Nakandala et al., 2019).

3 DEMONSTRATION

Datasets and CNN models. In this demonstration we will present an evaluation of KRYPTON with three real-world image datasets: 1) identifying diabetic retinopathy from retinal images, 2) identifying pneumonia from chest X-ray images, and 3) identifying objects from natural images in the ImageNet dataset. KRYPTON currently supports three popular CNN architectures, VGG, ResNet, and Inception models. Altogether, each participant will be able to interact with the system on nine different settings.

Walkthrough. Each Participant will be first made familiar with OBE method and the KRYPTON system. They will interact with the system using the user interface shown in Figure 1. First the participant will pick the dataset and the CNN model that they want to explore using the drop down menus. Then the participant can select a particular image from the selected dataset that they want to diagnose using the image file picker. This will load the input image to the left pane of the user interface and also show the predicted label and the probability on the right pane. For occluding the input image participants have two options. First one is to use the rectangular selection tool. The other option is to use the brush tool which will enable the participant to select an occlusion region in free form. When selecting the brush tool participants also have the option of changing the brush size. For both tools users have to select the color of the occlusion patch using the color picker dropdown menu. After selecting occlusion tool, participants can then interactively occlude parts of the input image, probably starting from the regions that they think contributes most to the prediction. While the participant interactively updates the occluded region, KRYPTON will show the sensitivity heat map generated so far in the right pane and will continue update it with incremental updates as the participant progresses.

REFERENCES

Kermany, D. S. et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

Nakandala, S. et al. Incremental and approximate inference for faster occlusion-based deep cnn explanations. https://adalabucsd.github.io/papers/TR_2019_Krypton.pdf, 2019.