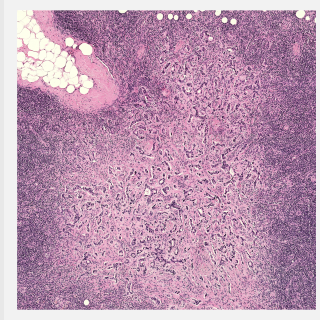
Training convolutional neural networks with megapixel images

Hans Pinckaers, Geert Litjens
Computational Pathology Group, Radboud University Medical Center, Nijmegen, The Netherlands

Some overlap between the tiles is needed to counteract for loss in convolutional layers.

up to 97% less memory required



typical input 299×299 — our input 8192×8192

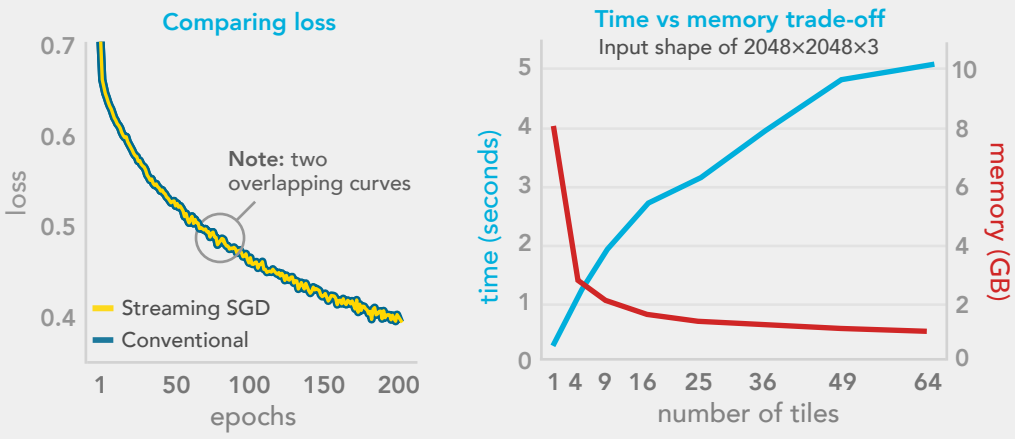
1. Some images are too big to copy to the GPU or to perform convolutions on.
2. To solve this we divide the image in a predefined number of tiles.
3. Tiles are processed one by one to reduce memory requirements.
4. Pooling layers (or larger strides) reduce the sizes of the intermediate feature maps.
5. Once an intermediate feature map can fit on the GPU we reconstruct it from the tiles.
6. The reconstructed intermediate feature map is passed through the final layers.
7. The final output will be based on the whole input image.

Introduction

In various domains researchers work with gigapixel images which contain both global contextual information and local details. Training convolutional neural networks with those images typically involves either downsizing them significantly or training on image patches. The former results in a loss of local details, whereas the latter results in losing global spatial information. Additionally, when training on image patches, there needs to be a label for every patch, which can be time-intensive to produce. **We propose two solutions to enable training of neural networks with multiple-megapixel images:**

Serial mini-batch training

Instead of processing all images of a mini-batch at once, we process them one-by-one and average the gradients before taking the gradient descent step.



https://github.com/DIAGNijmegen/StreamingSGD

Streaming SGD

Processing an image tile-by-tile (see figure on top). We use partial forward passes (gradient checkpointing) to recover feature maps of tiles during backpropagation.

Results & discussion

We show (see figure on the left) that the difference in loss of training with vanilla SGD and our methods is negligibly small while providing large memory savings. In the paper we sucessfully fit a neural network on 64-megapixel histopathology images of the Camelyon 2016 challenge using 7GB of memory instead of 235GB it would normally take. However, as shown in the figure, a disadvantage of StreamingSGD is that it requires recomputation of the activation maps of the tiles during the backward pass, which can cause significant computation overhead. Beside the time trade-off, architectures are less flexible since operations which depend on the whole activation map being present, e.g. batch normalization, cannot be used.

Institute for Health Sciences
Radboudumc