Lowering PyTorch's Memory Consumption for Selective Differentiation

iitdelhi
Indian Institute of Technology Delhi

Samarth Bhatia, Felix Dangel

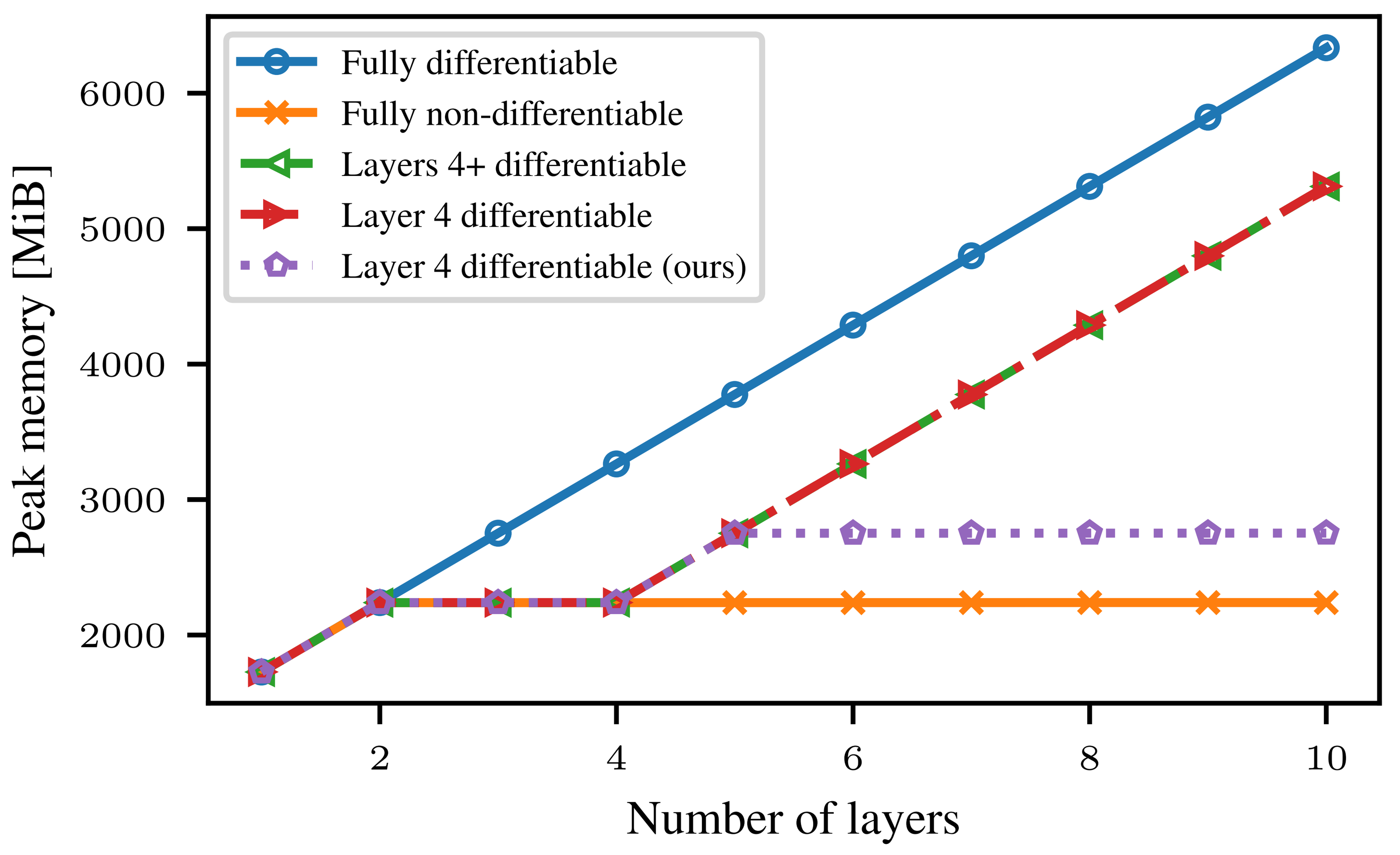
VECTOR INSTITUTE

We identify that some PyTorch layers save unnecessary tensors when parameters have `requires_grad = False` and provide a fix to reduce memory without affecting runtime.

This is useful for fine-tuning setups that only compute gradients for a sub-set of parameters.

PyTorch sometimes retains tensors which are not required for backpropagation

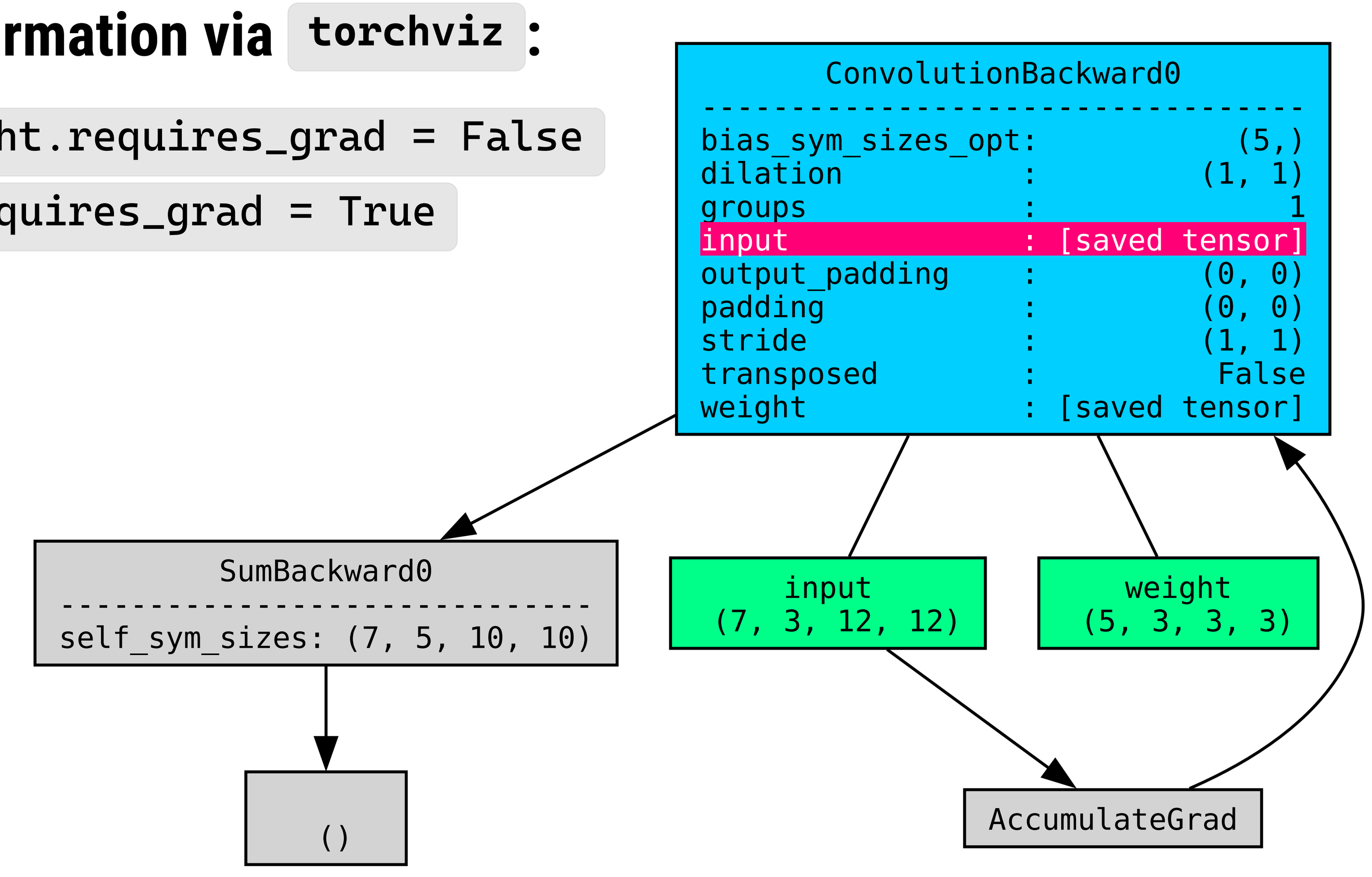
Toy example: Let's look at a CNN without pooling/activations. We feed a mini-batch which consumes 512 MiB memory. Each intermediate tensor requires 512 MiB memory. Here is the forward pass's memory consumption when different parameters are trainable (approximates computation graph size).



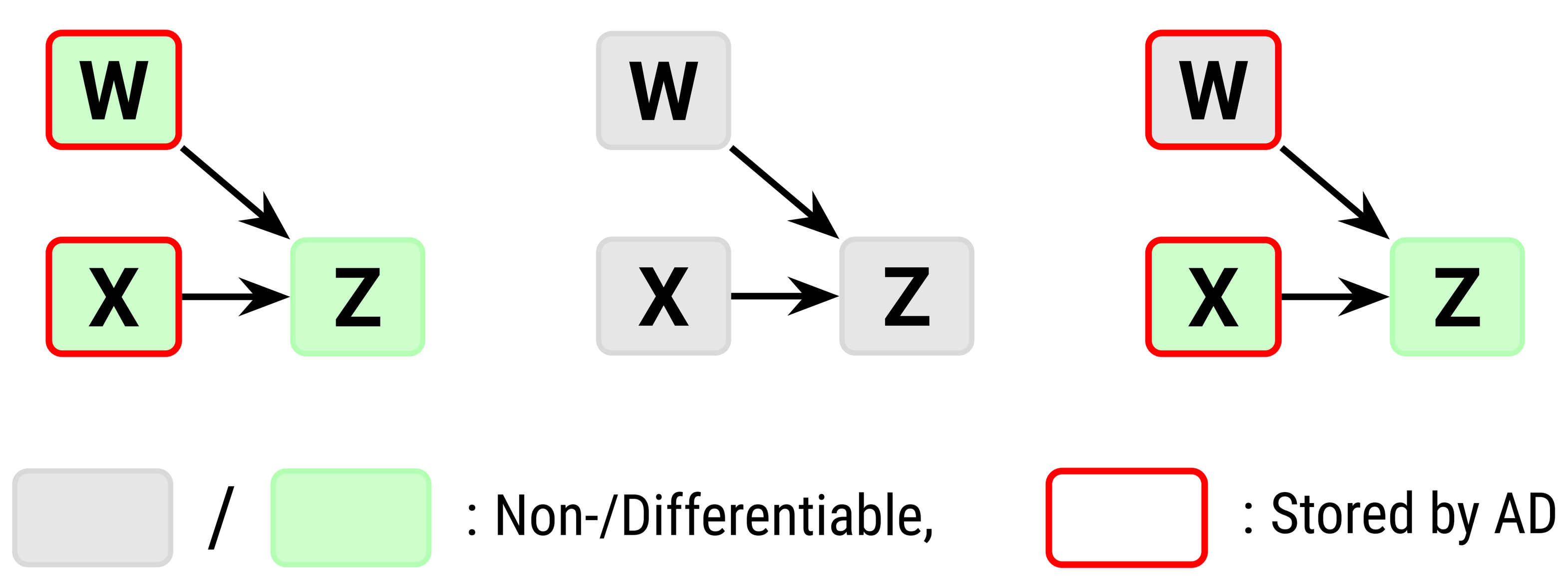
Conclusion: PyTorch's convolution stores the layer input if it is differentiable, *irrespective* of the weight's differentiability. *But we don't need the input if the weight is non-differentiable*!

Confirmation via `torchviz`:

```
weight.requires_grad = False
X.requires_grad = True
```



Summary:



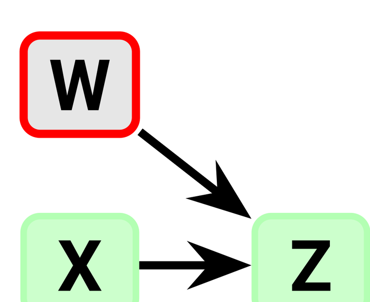
Fix: We provide a drop-in implementation which stores only the required tensors

Affected layers:

- Convolutions (`nn.ConvNd`)
- Batch norm in `eval` mode (`nn.BatchNormNd`)
- Transpose convolution (`nn.ConvTransposeNd`)

Interestingly, `nn.Linear` is optimized already!

Ours



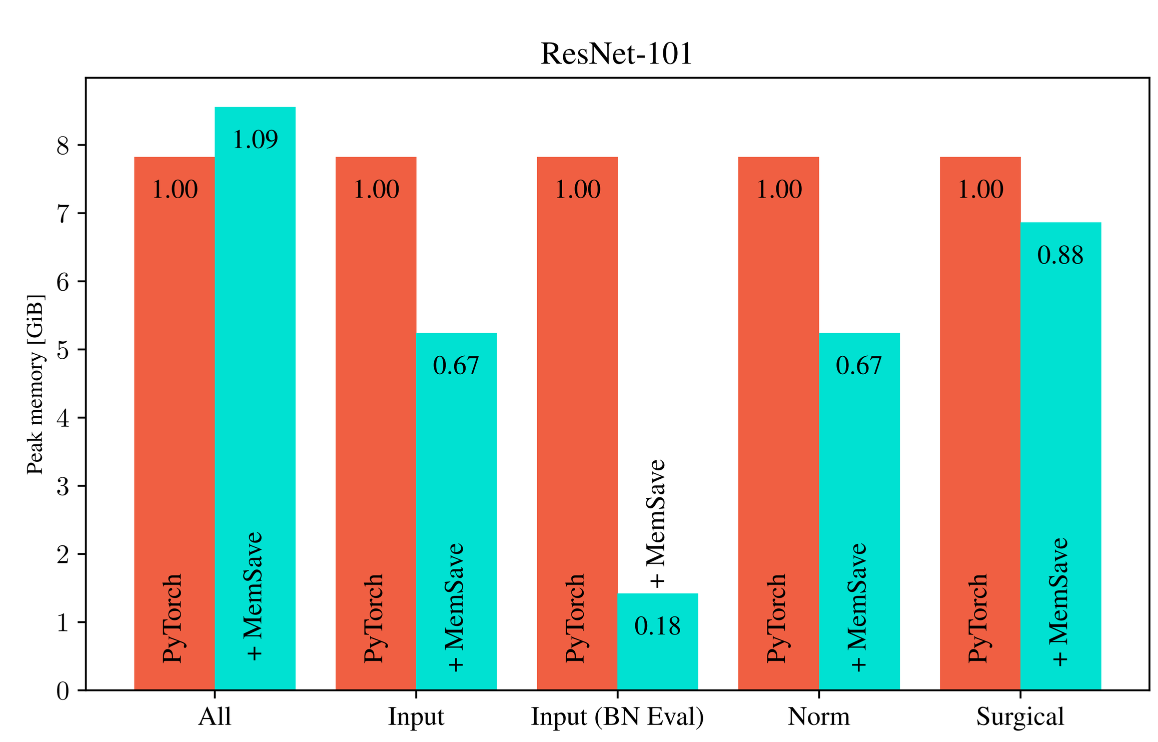
Other memory optimizations:

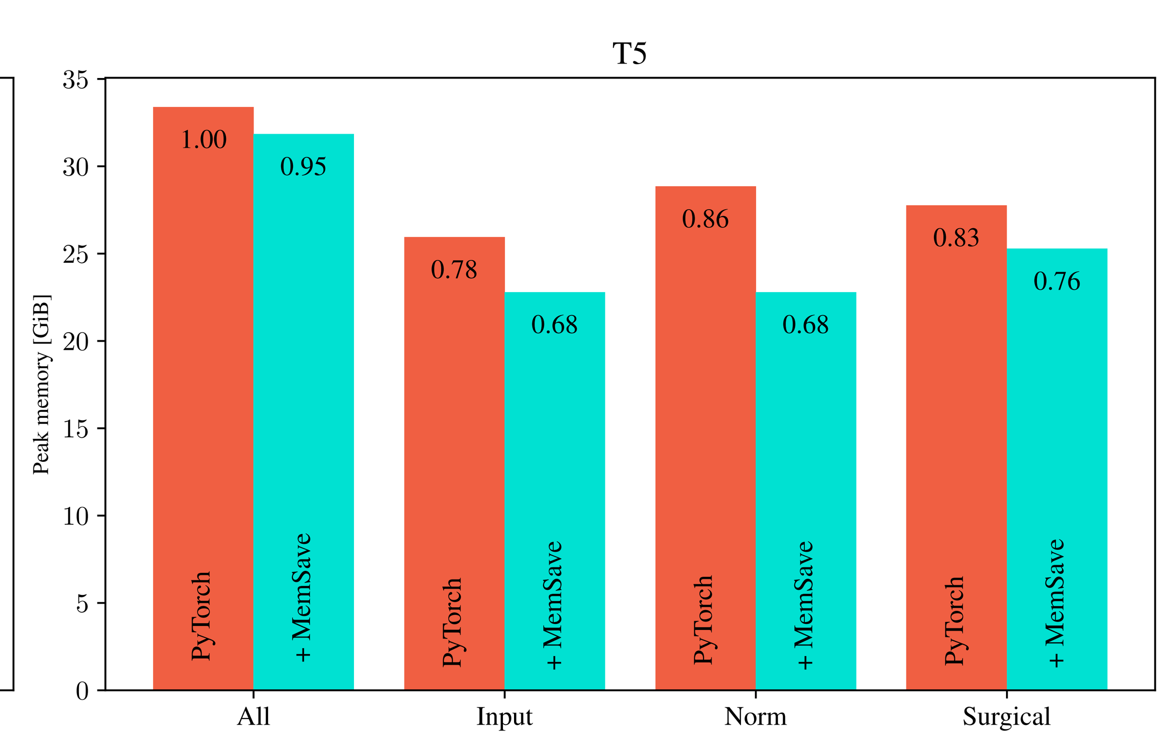
- `nn.ReLU`: Store boolean mask instead of floating point tensor (4x reduction, soon 32x when `torch.bit` is implemented)
- `nn.Dropout`: Only save random number generator state and re-compute dropout mask

Evaluation: Selective Differentiation in Practice

We evaluate on practical scenarios:

- '**All**': Training the full net (baseline)
- '**Input**': Only input differentiable (style transfer, adversarial data)
- '**Norm**': Only normalization layers trainable (layer-norm fine-tuning)
- '**Surgical**': Only trainable sub-network (surgical fine-tuning)






```
$ pip install memsave
$ model = memsave.convert(model)
```

Samarth is looking for a PhD position:

`samarth.bhatia23@alumni.iitd.ac.in`

arXiv

code