Assignment 6 - CNN

Advanced Topics in Neural Networks

7 November 2023

Homework

Deadline End-of-Day 24 November 2023

Using the previously built pipeline (or implement a new one), implement a CNN and test it on the CIFAR-10 dataset.

Implement the Conv2d layer by hand, using tensor operations. Your implementation needs to be valid. Test that on the same input and weights, your implementation, and nn.Conv2d, have the same output (use the example code below) (4 points).

Listing 1: Testing custom Conv2D Layer

```
import torch

inp = torch.randn(1, 3, 10, 12) # Input image
# kernel of size 4x5, with 3 input channels and 2 output channels
w = torch.randn(2, 3, 4, 5) # Conv weights
# Your implementation. Can be made differently, like only passing the
    kernel size for example
custom_conv2d_layer = Handmade_conv2d_implementation(weights=w)
out = custom_conv2d_layer(inp)
print((torch.nn.functional.conv2d(inp, w) - out).abs().max())
```

For the following tasks, you may opt to use your own implementation, or you can use the PyTorch equivalent.

1. Try to compile, trace and script your model. Check the runtime performance and compare it with the base model (1 point). Compiling on Windows works only from WSL. You may use Google Colab to test it.

2. Achieve 94% validation accuracy on CIFAR-10, without using a pretrained model (2 points).

3. Achieve 95.5% validation accuracy on CIFAR-10, without using a pretrained model (2 points).

4. **Bonus:** Use transfer learning to finetune a pretrained model on CIFAR-10 (1 point). Achieve 97% validation accuracy on CIFAR-10 using the finetuned model (1 point).

5. **Bonus:** Achieve 97% validation accuracy on CIFAR-10, without using a pretrained model (2 points).

6. **Bonus:** Change the dataset to CIFAR-100. Achieve over 75% validation accuracy on CIFAR-100 without using a pretrained model (2 points).

7. **Bonus:** Also test a pretrained model on CIFAR-100 and compare it with the previous results (1 point).

Your training pipeline must run both on CPU and GPU, using a device parameter.
In your submission, you must include:

- A README.md file in which you summarize what you did, your results, and the expected number of points. Include information about the runtime performance of the base/compiled/-traced/scripted model in a table.
- A small Latex document in which you explain all layers of the model you used (-2 points if not done), the forward function (-2 points if not done) and the gradient flow (-2 points if not done). Explain which layers contributed to your result and why.
- The tensorboard files or weights & biases link in which you report the metrics for the training in which you achieved the best results without using a pretrained model. Also include the files for the pretrained model, if applicable.
- A png file in which you plot the validation accuracy of your model. Also include the validation accuracy of the pretrained model in the same plot, if applicable.
- The pretrained weights of your best model, along with an inference script named "inference.py". Running the inference.py will load the weights of the pretrained model and run the validation on CIFAR-10, printing the validation accuracy. Also include the pretrained model weights if applicable.
- If you are doing your homework in a Jupyter Notebook, add the "Open in Colab" option. You must still include the inference.py script and the other requested files.

Upload your homework in the /Lab06/Solution directory.