Project: Comparing Various Stochastic Gradient Methods by PyTorch
Goal

- We check the use of several stochastic gradient implementations in PyTorch for training CNN
In the previous project, we tried only PyTorch’s simple stochastic gradient implementation.

Now we have discussed other variants.

Let’s try them in this project:

- Simple stochastic gradient (your previous project)
- Stochastic gradient with momentum
- Adagrad
- Adam

All settings (e.g., architectures, activation functions, data sets, etc.) are the same as the previous project.
Use their default parameter settings and initial solutions (so these methods may use different initial solutions)

See values listed in https://pytorch.org/docs/stable/optim.html

But for the simple SGD there is no default learning rate:

torch.optim.SGD(params,
lr=<required parameter>, momentum=0,
dampening=0, weight_decay=0, nesterov=False)

Let’s do
lr = 0.01

- For SG with momentum, let’s have
  momentum = 0.9

- For the loss function, let’s do
  
  \texttt{mse\_loss}

To use the MSE loss you need to convert each label to an indicator vector. Do a simple Internet search to see how people did it.

- Batch size is another issue. It seems the default is 1. Let’s use
  128
for all cases

- Pre-processing is another concern. From a guide https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html, they use

```python
transform = transforms.Compose(
    [transforms.ToTensor(),
     transforms.Normalize((0.5, 0.5, 0.5),
                          (0.5, 0.5, 0.5))])

trainset =
torchvision.datasets.CIFAR10(root='./data',
                              train=True, download=True,
                              transform=transform)
```
At this moment we don’t quite understand what it does. Let’s check two settings

- with the above setting
- without the normalization:
  `transforms.Normalize`

Let’s run 50 epochs and check the relationship between

accuracy and the accumulated # of epochs

A purpose is to see if in general a validation procedure is needed to decide the stopping condition

No tuning on other parameters yet
Please work on both MNIST and CIFAR10 data sets
In your report, give your observations and thoughts
Presentation

Students with the following UIDS (last three digits): 982, 22, 261, 793, 770, 042, 750, please do a 10-minute presentation (8-minute the contents and 2-minute Q&A)