TP: Introduction to Pytorch

1 Auto-differential and optimization in Pytorch

1.1 Setup

Check the python3 environment and make sure that pytorch is installed.

```
>>> import torch
>>> torch.__version__
'1.10.0+cu102'
```

Make a test first of using CPU and GPU.

```
import torch
cuda = torch.device('cuda:0')
print('compatibility gpu')
print(torch.cuda.get_device_capability(device=cuda))
a = torch.tensor([11,11],device=cuda)
print(a+a)
```

1.2 Forward/Backward propagation and SGD

Assume $x_t \in \mathbb{R}$ and $\theta \in \mathbb{R}$. We want to optimize θ so that x_T is close to a target value under the following linear dynamics,

$$x_{t+1} = x_t + \theta$$

Assume that x_0 is random, our goal is to minimize the following objective function

$$L_T = \mathbb{E}_{x_0 \sim p}(x_T - 1)^2$$

To compute x_T, you are supposed to call the following function multiple times,

```
def f(x,theta):
    return x + theta
```

• Assume that x_0 follows $p = \mathcal{N}(1,1)$. Use $mb = 1024$ samples to estimate the expectation in L_T.

• Compute automatically the derivative of L_T with respect to θ.

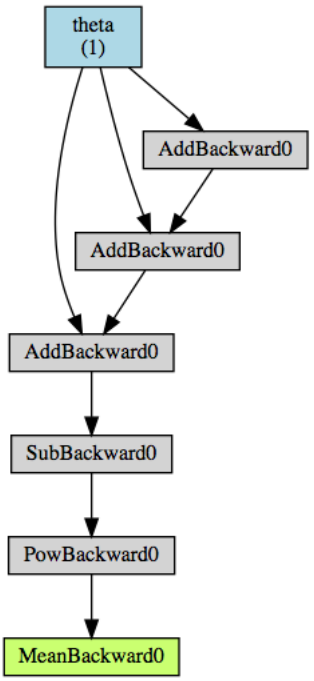

Figure 1 – Computational graph of the loss L_T with respect to θ.

• Assume $T = 20$ and $\theta = 1$, what is the loss L_T and its gradient ? Implement a gradient descent algorithm with constant step-size to minimize L_T with respect θ, started from $\theta^{(0)} = 1$. Choose a proper step-size (learning rate).

• Use torch.optim.SGD to minimize L_T. What is the optimal solution that you have found ?

• What happens if $T = 200$ with the same learning rate ? How to set learning rate for an arbitrary T ?

• Test the code on both CPU and GPU and check if you have obtained the same results. Measure their computational time.

1.3 Visualize the computational graph

Install torchviz (if not installed)

```
pip install torchviz --user
```

• For $T = 3$, check that if you obtain Figure 1.