In [1]:

```
! python --version
```

```
Python 3.6.9
```

In [2]:

```
! pip freeze | grep -P '(torch|numpy)'
```

```
numpy==1.19.2
torch==1.8.1+cu101
torchvision==0.9.1+cu101
```

In [3]:

```
! lscpu
```

```
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              6
On-line CPU(s) list: 0-5
Thread(s) per core:  1
Core(s) per socket:  6
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               79
Model name:          Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz
Stepping:            1
CPU MHz:             2593.991
BogoMIPS:            5187.98
Hypervisor vendor:   Microsoft
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            256K
L3 cache:            35840K
NUMA node0 CPU(s):   0-5
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm c
onstant_tsc rep_good nopl xtopology cpuid pni pclmulqdq ssse3 fma cx16 pcid s
se4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm 3d
nowprefetch invpcid_single pti fsgsbase bmi1 hle avx2 smep bmi2 erms invpcid
rtm rdseed adx smap xsaveopt
```

In [4]:

```
! nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:07:16_PDT_2019
Cuda compilation tools, release 10.1, V10.1.243
```

In [5]:

```
! nvidia-smi --query-gpu=name,driver_version --format=csv
```

```
name, driver_version
Tesla V100-PCIE-16GB, 460.27.04
```

In [6]:

```
import torch
torch.set_printoptions(precision=11)
```

In [7]:

```
print(torch.__config__.show())
```

```
PyTorch built with:
  - GCC 7.3
  - C++ Version: 201402
  - Intel(R) Math Kernel Library Version 2020.0.0 Product Build 20191122 for
Intel(R) 64 architecture applications
  - Intel(R) MKL-DNN v1.7.0 (Git Hash 7aed236906b1f7a05c0917e5257a1af05e9ff68
3)
  - OpenMP 201511 (a.k.a. OpenMP 4.5)
  - NNPACK is enabled
  - CPU capability usage: AVX2
  - CUDA Runtime 10.1
  - NVCC architecture flags: -gencode;arch=compute_37,code=sm_37;-gencode;arc
h=compute_50,code=sm_50;-gencode;arch=compute_60,code=sm_60;-gencode;arch=com
pute_70,code=sm_70
  - CuDNN 7.6.3
  - Magma 2.5.2
  - Build settings: BLAS_INFO=mkl, BUILD_TYPE=Release, CUDA_VERSION=10.1, CUD
NN_VERSION=7.6.3, CXX_COMPILER=/opt/rh/devtoolset-7/root/usr/bin/c++, CXX_FLA
GS= -Wno-deprecated -fvisibility-inlines-hidden -DUSE_PTHREADPOOL -fopenmp -D
NDEBUG -DUSE_KINETO -DUSE_FBGEMM -DUSE_QNNPACK -DUSE_PYTORCH_QNNPACK -DUSE_XN
NPACK -O2 -fPIC -Wno-narrowing -Wall -Wextra -Werror=return-type -Wno-missing
-field-initializers -Wno-type-limits -Wno-array-bounds -Wno-unknown-pragmas -
Wno-sign-compare -Wno-unused-parameter -Wno-unused-variable -Wno-unused-funct
ion -Wno-unused-result -Wno-unused-local-typedefs -Wno-strict-overflow -Wno-s
trict-aliasing -Wno-error=deprecated-declarations -Wno-stringop-overflow -Wno
-psabi -Wno-error=pedantic -Wno-error=redundant-decls -Wno-error=old-style-ca
st -fdiagnostics-color=always -faligned-new -Wno-unused-but-set-variable -Wno
-maybe-uninitialized -fno-math-errno -fno-trapping-math -Werror=format -Wno-s
tringop-overflow, LAPACK_INFO=mkl, PERF_WITH_AVX=1, PERF_WITH_AVX2=1, PERF_WI
TH_AVX512=1, TORCH_VERSION=1.8.1, USE_CUDA=ON, USE_CUDNN=ON, USE_EXCEPTION_PT
R=1, USE_GFLAGS=OFF, USE_GLOG=OFF, USE_MKL=ON, USE_MKLDNN=ON, USE_MPI=OFF, US
E_NCCL=ON, USE_NNPACK=ON, USE_OPENMP=ON,
```

In [8]:

```
print('Cuda', torch.cuda.is_available())
```

```
Cuda True
```

In [9]:

```
def set_random_seed(seed_value: int, use_cuda: bool = False):
    import torch
    import numpy as np
    import random
    np.random.seed(seed_value)  # cpu vars
    torch.manual_seed(seed_value)  # cpu  vars
    random.seed(seed_value)  # Python
    torch.use_deterministic_algorithms(True)
    if use_cuda:
        torch.cuda.manual_seed(seed_value)
        torch.cuda.manual_seed_all(seed_value)  # gpu vars
        torch.backends.cudnn.deterministic = True  # needed
        torch.backends.cudnn.benchmark = True


def _run_module(layer_norm, dropout, x, use_cuda, train, seed):
    if use_cuda:
        device = torch.device('cuda:0')
    else:
        device = torch.device('cpu')
    set_random_seed(seed, use_cuda=use_cuda)
    layers = []
    if layer_norm:
        layers.append(torch.nn.LayerNorm(normalized_shape=(x.shape[-1],), eps
    if dropout:
        layers.append(torch.nn.Dropout(0.5, inplace=False))

    if not layers:
        raise ValueError('set `layer_norm` and/or `dropout` to True')

    model = torch.nn.Sequential(*layers)
    model.to(device)
    if train:
        model.zero_grad()
        model.train()
    else:
        model.eval()
    x = x.clone().to(device)
    if train:
        out = model(x)
    else:
        with torch.no_grad():
            out = model(x)
    return out


def run_layer_norm_module(seed):
    set_random_seed(seed, use_cuda=False)
    x = torch.rand(3, 3)
    return {
        'x': x,
        'cpu_t': _run_module(layer_norm=True, dropout=False, x=x, use_cuda=Fa
        'cpu_e': _run_module(layer_norm=True, dropout=False, x=x, use_cuda=Fa
        'gpu_t': _run_module(layer_norm=True, dropout=False, x=x, use_cuda=Tr
        'gpu_e': _run_module(layer_norm=True, dropout=False, x=x, use_cuda=Tr
    }



def run_dropout_module(seed):
    set_random_seed(seed, use_cuda=False)
    x = torch.rand(4, 256, 256)
    return {
        'x': x,
        'cpu_t': _run_module(layer_norm=False, dropout=True, x=x, use_cuda=Fa
        'gpu_t': _run_module(layer_norm=False, dropout=True, x=x, use_cuda=Tr
    }
```

In [10]:

```
runs = [run_dropout_module(seed=42) for i in range(5)]

for i in range(1, len(runs)):
    for k in runs[0]:
        assert torch.equal(runs[0][k], runs[i][k]), (i, k)

for k in runs[0]:
    if k != 'x':
        print(k, (runs[0][k].cpu() == 0.0).sum(axis=-1).tolist())
        print('=' * 100)
```

```
cpu_t [[119, 120, 120, 127, 126, 116, 131, 124, 125, 120, 123, 128, 130, 132,
134, 124, 133, 135, 130, 128, 119, 138, 125, 123, 132, 134, 137, 142, 114, 12
0, 131, 136, 124, 125, 155, 118, 144, 125, 129, 135, 124, 126, 134, 121, 121,
136, 127, 130, 124, 117, 106, 124, 132, 143, 124, 128, 136, 120, 119, 123, 11
6, 127, 138, 117, 124, 137, 128, 119, 123, 124, 138, 128, 118, 127, 134, 136,
133, 123, 123, 131, 119, 123, 136, 128, 123, 138, 127, 125, 133, 116, 125, 13
```

```
5, 130, 117, 127, 131, 125, 137, 123, 137, 137, 128, 122, 126, 132, 139, 141,
127, 112, 145, 148, 143, 130, 118, 129, 143, 136, 122, 121, 124, 140, 127, 11
6, 125, 126, 133, 124, 132, 138, 136, 135, 126, 134, 132, 123, 135, 119, 131,
121, 126, 128, 125, 121, 131, 126, 131, 126, 142, 130, 119, 135, 128, 118, 13
6, 132, 137, 135, 119, 121, 129, 132, 124, 114, 137, 138, 117, 128, 127, 139,
132, 126, 134, 129, 131, 127, 118, 123, 130, 133, 136, 138, 140, 128, 119, 13
0, 134, 129, 131, 122, 124, 138, 125, 127, 123, 127, 124, 142, 122, 132, 126,
117, 122, 123, 136, 118, 130, 122, 135, 125, 132, 119, 129, 137, 122, 130, 12
6, 144, 130, 143, 122, 119, 131, 123, 122, 117, 121, 127, 120, 137, 127, 132,
127, 134, 124, 139, 142, 122, 129, 116, 120, 121, 135, 123, 125, 129, 131, 12
6, 131, 121, 120, 121, 134, 120, 128, 152, 114], [126, 136, 128, 128, 126, 13
5, 120, 130, 121, 119, 124, 138, 122, 122, 134, 127, 127, 130, 118, 128, 124,
128, 125, 137, 134, 127, 125, 117, 121, 128, 117, 121, 116, 116, 144, 131, 12
8, 126, 124, 117, 143, 131, 135, 145, 133, 122, 122, 129, 127, 126, 119, 122,
120, 127, 132, 134, 132, 119, 132, 132, 141, 128, 132, 133, 132, 118, 132, 12
9, 121, 136, 131, 127, 135, 120, 128, 124, 137, 127, 121, 130, 133, 135, 129,
130, 131, 131, 118, 129, 119, 127, 140, 134, 127, 136, 138, 125, 125, 133, 12
1, 111, 124, 120, 133, 133, 134, 136, 143, 130, 126, 122, 137, 139, 147, 126,
138, 134, 117, 130, 122, 134, 138, 120, 128, 117, 132, 120, 138, 131, 136, 12
7, 124, 131, 124, 112, 142, 118, 121, 129, 117, 116, 136, 137, 120, 134, 138,
132, 128, 123, 137, 132, 123, 131, 122, 117, 125, 108, 123, 127, 136, 125, 12
6, 130, 123, 118, 135, 110, 135, 122, 114, 136, 125, 128, 124, 128, 131, 129,
119, 130, 124, 134, 132, 121, 133, 138, 129, 116, 116, 125, 142, 107, 110, 13
5, 130, 128, 119, 142, 119, 124, 137, 129, 132, 115, 117, 125, 118, 141, 131,
117, 138, 125, 120, 130, 126, 125, 123, 136, 137, 127, 122, 135, 138, 114, 12
9, 142, 119, 129, 113, 135, 133, 135, 130, 137, 141, 143, 131, 128, 137, 137,
122, 123, 123, 141, 125, 141, 109, 136, 133, 116, 112, 126, 128, 115, 125, 13
2, 149, 128], [125, 138, 126, 128, 136, 128, 127, 126, 124, 129, 133, 138, 12
3, 128, 130, 124, 133, 123, 125, 128, 128, 139, 137, 136, 130, 130, 132, 134,
118, 108, 125, 126, 125, 122, 130, 133, 137, 135, 133, 126, 132, 134, 134, 11
6, 135, 129, 135, 129, 120, 131, 120, 132, 121, 108, 132, 128, 130, 125, 137,
117, 128, 139, 123, 138, 128, 111, 122, 112, 133, 124, 126, 137, 130, 143, 12
1, 130, 124, 125, 130, 116, 126, 129, 134, 129, 138, 121, 133, 124, 135, 126,
129, 132, 131, 118, 133, 135, 118, 132, 112, 119, 125, 119, 140, 129, 132, 13
8, 120, 130, 129, 125, 123, 111, 129, 121, 129, 126, 126, 142, 130, 132, 133,
113, 121, 133, 136, 125, 126, 127, 140, 137, 120, 135, 134, 133, 131, 125, 12
0, 136, 129, 121, 126, 134, 136, 130, 132, 132, 125, 127, 116, 140, 133, 125,
130, 135, 134, 119, 120, 115, 131, 127, 121, 133, 119, 124, 129, 122, 129, 11
4, 121, 117, 119, 119, 119, 133, 150, 131, 127, 117, 117, 139, 130, 131, 129,
128, 128, 120, 130, 137, 132, 129, 114, 134, 127, 116, 119, 128, 112, 126, 11
0, 122, 133, 124, 132, 141, 128, 113, 126, 120, 124, 135, 129, 117, 129, 125,
138, 125, 129, 129, 124, 136, 127, 135, 128, 128, 141, 128, 133, 128, 117, 13
4, 116, 132, 130, 122, 127, 127, 137, 132, 137, 115, 119, 125, 123, 134, 116,
120, 127, 133, 122, 125, 123, 117, 139, 134, 120, 131], [131, 118, 138, 120,
148, 111, 122, 120, 123, 130, 145, 119, 125, 141, 124, 126, 116, 125, 135, 13
0, 114, 134, 129, 132, 115, 124, 129, 127, 116, 122, 146, 136, 115, 129, 135,
123, 119, 132, 149, 143, 129, 120, 119, 122, 119, 128, 134, 136, 133, 123, 13
6, 117, 127, 126, 135, 122, 130, 129, 122, 139, 143, 133, 125, 128, 125, 138,
114, 136, 131, 129, 134, 120, 122, 127, 118, 121, 124, 141, 126, 126, 130, 13
1, 121, 116, 146, 130, 138, 121, 139, 115, 119, 125, 134, 120, 139, 144, 126,
128, 121, 123, 129, 128, 137, 126, 128, 129, 131, 107, 136, 135, 118, 126, 13
4, 129, 126, 142, 137, 125, 128, 143, 125, 135, 133, 120, 112, 123, 141, 126,
123, 140, 130, 125, 120, 138, 126, 123, 133, 143, 123, 118, 134, 128, 144, 12
5, 127, 115, 139, 121, 133, 128, 119, 131, 126, 124, 149, 138, 132, 137, 136,
118, 132, 138, 136, 123, 134, 118, 119, 136, 124, 127, 123, 124, 123, 121, 13
6, 132, 130, 114, 139, 132, 130, 123, 130, 132, 105, 136, 126, 125, 125, 124,
127, 115, 129, 131, 136, 120, 128, 131, 136, 118, 132, 120, 130, 116, 136, 12
9, 129, 132, 129, 134, 113, 140, 117, 134, 118, 133, 117, 148, 134, 129, 129,
128, 139, 128, 126, 126, 121, 130, 134, 133, 124, 145, 137, 126, 120, 134, 12
9, 124, 129, 147, 132, 108, 127, 138, 132, 113, 113, 140, 119, 118, 127, 130,
117, 118, 133, 126]]
=============================================================================
=======================
gpu_t [[119, 135, 138, 132, 123, 125, 128, 123, 120, 127, 118, 119, 139, 133,
122, 127, 132, 146, 124, 142, 133, 133, 135, 137, 130, 141, 134, 145, 121, 12
0, 138, 126, 115, 128, 132, 131, 122, 120, 113, 121, 129, 141, 128, 120, 122,
136, 126, 105, 137, 133, 116, 134, 130, 120, 125, 121, 138, 126, 127, 124, 12
7, 131, 132, 135, 131, 138, 119, 130, 128, 139, 134, 134, 125, 143, 132, 126,
135, 135, 120, 142, 132, 130, 141, 126, 134, 125, 138, 116, 129, 131, 112, 13
```

```
6, 131, 134, 124, 117, 120, 124, 130, 134, 123, 126, 115, 115, 127, 141, 130,
133, 116, 132, 135, 136, 108, 122, 119, 136, 133, 128, 129, 129, 129, 115, 14
0, 135, 117, 134, 135, 122, 126, 142, 126, 143, 114, 124, 137, 116, 135, 125,
126, 127, 113, 122, 126, 115, 135, 147, 133, 128, 124, 126, 115, 141, 145, 13
9, 135, 129, 122, 122, 111, 133, 120, 130, 143, 137, 128, 124, 118, 118, 134,
127, 131, 140, 113, 139, 139, 123, 116, 119, 130, 130, 124, 125, 132, 142, 13
1, 119, 122, 127, 120, 128, 125, 128, 132, 136, 137, 114, 148, 116, 126, 114,
124, 134, 129, 137, 124, 132, 135, 139, 141, 114, 127, 132, 117, 125, 115, 12
1, 133, 128, 132, 136, 125, 135, 125, 143, 128, 117, 129, 127, 131, 124, 124,
148, 134, 135, 108, 115, 129, 112, 112, 132, 126, 129, 140, 135, 138, 114, 12
6, 128, 118, 113, 125, 135, 133, 128, 137, 129], [136, 143, 129, 131, 128, 11
7, 134, 127, 133, 122, 125, 127, 117, 131, 130, 129, 132, 135, 123, 133, 133,
132, 117, 119, 125, 119, 137, 136, 139, 125, 134, 134, 140, 140, 132, 123, 14
5, 142, 130, 133, 138, 120, 130, 132, 117, 135, 120, 142, 135, 133, 123, 127,
131, 111, 135, 147, 133, 127, 132, 119, 130, 140, 120, 119, 117, 137, 139, 11
7, 115, 121, 138, 134, 119, 128, 124, 123, 113, 125, 106, 119, 130, 124, 135,
129, 130, 118, 119, 128, 129, 137, 119, 126, 122, 134, 143, 133, 129, 132, 12
6, 128, 137, 120, 129, 121, 128, 147, 120, 129, 142, 127, 127, 127, 130, 124,
125, 115, 123, 111, 129, 120, 141, 126, 135, 122, 127, 121, 121, 144, 130, 12
5, 122, 141, 126, 131, 125, 123, 139, 129, 118, 129, 137, 144, 140, 124, 138,
118, 129, 135, 131, 128, 138, 137, 124, 152, 114, 137, 119, 139, 115, 131, 11
6, 136, 122, 132, 132, 114, 113, 124, 126, 129, 122, 134, 148, 132, 130, 113,
126, 129, 126, 133, 127, 123, 141, 126, 126, 126, 134, 123, 123, 131, 125, 12
9, 140, 129, 123, 126, 135, 120, 130, 122, 139, 125, 134, 130, 129, 122, 133,
113, 127, 125, 127, 117, 130, 109, 136, 123, 132, 128, 125, 129, 137, 132, 13
1, 137, 121, 128, 131, 127, 128, 128, 121, 134, 128, 136, 132, 121, 129, 143,
137, 137, 134, 136, 125, 127, 128, 131, 114, 137, 127, 126, 144, 129, 131, 12
6, 147, 119], [123, 121, 108, 126, 130, 131, 127, 137, 121, 138, 120, 142, 11
6, 131, 118, 143, 132, 106, 143, 145, 137, 128, 134, 102, 132, 125, 129, 131,
130, 126, 124, 131, 138, 131, 126, 138, 126, 134, 125, 126, 142, 132, 130, 13
0, 132, 116, 135, 124, 113, 123, 127, 132, 135, 125, 127, 134, 130, 134, 116,
119, 134, 135, 134, 126, 122, 122, 123, 122, 122, 123, 137, 126, 147, 124, 11
4, 137, 129, 127, 140, 131, 115, 122, 131, 119, 131, 135, 148, 127, 146, 124,
132, 122, 127, 130, 144, 124, 127, 147, 129, 126, 136, 115, 124, 115, 137, 14
0, 133, 119, 128, 127, 131, 120, 129, 139, 130, 146, 120, 120, 137, 118, 119,
131, 138, 115, 129, 138, 138, 136, 116, 125, 120, 128, 130, 136, 121, 122, 13
4, 134, 127, 108, 142, 114, 129, 143, 120, 120, 134, 118, 134, 136, 141, 125,
131, 129, 126, 125, 135, 133, 133, 134, 139, 142, 131, 128, 126, 143, 129, 12
0, 126, 132, 127, 141, 124, 132, 131, 135, 127, 123, 135, 133, 132, 134, 129,
135, 136, 133, 130, 130, 117, 115, 139, 118, 115, 141, 129, 136, 138, 123, 12
8, 145, 127, 130, 130, 123, 131, 123, 109, 131, 137, 135, 140, 133, 119, 145,
140, 130, 123, 131, 126, 126, 139, 145, 133, 125, 133, 129, 136, 127, 122, 12
3, 120, 130, 125, 136, 127, 120, 139, 132, 124, 128, 144, 132, 108, 132, 114,
131, 130, 137, 132, 115, 128, 127, 129, 132, 126, 123], [126, 146, 132, 130,
120, 122, 138, 123, 130, 116, 132, 130, 114, 136, 125, 117, 130, 126, 127, 12
9, 131, 124, 130, 134, 123, 119, 137, 120, 124, 138, 121, 123, 136, 118, 136,
133, 121, 136, 131, 121, 119, 130, 121, 113, 126, 134, 126, 132, 123, 135, 13
3, 117, 139, 138, 125, 123, 133, 129, 133, 139, 133, 117, 116, 130, 124, 123,
131, 117, 124, 119, 127, 131, 120, 120, 115, 136, 134, 140, 133, 126, 139, 11
6, 129, 131, 134, 141, 115, 134, 119, 125, 129, 117, 126, 125, 128, 115, 129,
119, 124, 131, 115, 121, 135, 137, 129, 138, 136, 132, 146, 147, 127, 144, 12
3, 114, 140, 136, 126, 111, 125, 136, 120, 121, 139, 124, 139, 132, 129, 118,
148, 120, 137, 130, 137, 121, 141, 122, 130, 123, 126, 117, 129, 128, 122, 13
6, 135, 127, 153, 147, 134, 141, 122, 119, 128, 131, 126, 130, 120, 126, 136,
128, 132, 133, 135, 114, 119, 128, 121, 120, 131, 129, 141, 134, 128, 121, 14
2, 128, 133, 141, 123, 125, 127, 128, 126, 140, 128, 111, 107, 132, 125, 126,
127, 131, 127, 124, 120, 114, 134, 132, 120, 124, 136, 120, 121, 133, 121, 12
4, 137, 133, 131, 125, 130, 126, 128, 135, 123, 137, 130, 147, 139, 136, 133,
138, 117, 126, 126, 111, 121, 107, 129, 120, 126, 120, 120, 130, 126, 129, 12
3, 116, 116, 126, 130, 127, 130, 131, 132, 131, 141, 137, 112, 115, 125, 120,
122, 144, 146, 130]]
================================================================================
======================
```

In [11]:

```
runs = [run_layer_norm_module(seed=42) for i in range(5)]

for i in range(1, len(runs)):
    for k in runs[0]:
        assert torch.equal(runs[0][k], runs[i][k]), (i, k)

runs[0]
```

```
Out[11]: {'x': tensor([[0.88226926327, 0.91500395536, 0.38286375999],
                  [0.95930564404, 0.39044821262, 0.60089534521],
                  [0.25657248497, 0.79364132881, 0.94077146053]]),
          'cpu_t': tensor([[ 0.63882547617,  0.77325701714, -1.41208255291],
                  [ 1.31617474556, -1.10615468025, -0.21002103388],
                  [-1.38439559937,  0.44202369452,  0.94237166643]],
                 grad_fn=<NativeLayerNormBackward>),
          'cpu_e': tensor([[ 0.63882547617,  0.77325701714, -1.41208255291],
                  [ 1.31617474556, -1.10615468025, -0.21002103388],
                  [-1.38439559937,  0.44202369452,  0.94237166643]]),
          'gpu_t': tensor([[ 0.63882541656,  0.77325695753, -1.41208267212],
                  [ 1.31617438793, -1.10615372658, -0.21002060175],
                  [-1.38439559937,  0.44202369452,  0.94237166643]], device='cuda:0',
                 grad_fn=<NativeLayerNormBackward>),
          'gpu_e': tensor([[ 0.63882541656,  0.77325695753, -1.41208267212],
                  [ 1.31617438793, -1.10615372658, -0.21002060175],
                  [-1.38439559937,  0.44202369452,  0.94237166643]], device='cuda:0')}
```