Graph Compiler

PerfOp/Ultra

Andy Fox, Edvard Ghazaryan 06.07.2021, EFS ID 42832446

Goal

Graph Compiler

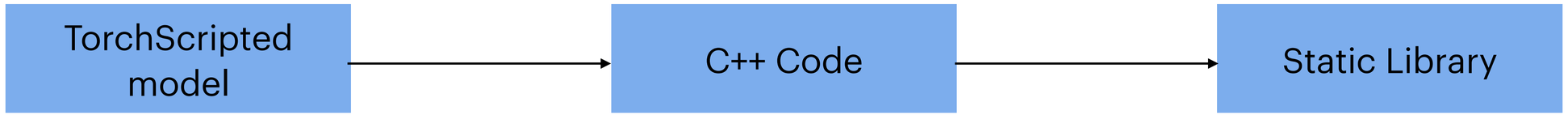
Provide the zero-overhead principle:

What you don’t use, you don’t pay for.

(Zero cost for abstractions employed by DL framework.)

Flow

Graph Compiler



Andy Fox, Edvard Ghazaryan 06.07.2021, EFS ID 42832446

Input/Output (simplified)

Graph Compiler

```
graph(%self.1 : __torch__.___torch_mangle_0.DeepAndWide,
      %ad_emb_packed.1 : Tensor,
      %user_emb.1 : Tensor,
      %wide.1 : Tensor):
  %29 : Float(51:1, 1:51) = prim::Constant[value=<Tensor>]()
  %8 : int = prim::Constant[value=-1]()
  %7 : int = prim::Constant[value=2]()
  %6 : float = prim::Constant[value=10.]()
  %5 : float = prim::Constant[value=0.]()
  %4 : int = prim::Constant[value=1]()
  %self._mu : Float(1:50, 50:1) = prim::Constant[value=<Tensor>]()
  %self._sigma : Float(1:50, 50:1) = prim::Constant[value=<Tensor>]()
  %self._fc_b : Float(1:1) = prim::Constant[value={1.70265}]()
  %wide_offset.1 : Tensor = aten::add(%wide.1, %self._mu, %4)
  %wide_normalized.1 : Tensor = aten::mul(%wide_offset.1, %self._sigma)
  %wide_preproc.1 : Tensor = aten::clamp(%wide_normalized.1, %5, %6)
  %user_emb_t.1 : Tensor = aten::transpose(%user_emb.1, %4, %7)
  %dp_unflatten.1 : Tensor = aten::bmm(%ad_emb_packed.1, %user_emb_t.1)
  %dp.1 : Tensor = aten::flatten(%dp_unflatten.1, %4, %8)
  %19 : Tensor[] = prim::ListConstruct(%dp.1, %wide_preproc.1)
  %input.1 : Tensor = aten::cat(%19, %4)
  %fc1.1 : Tensor = aten::addmm(%self._fc_b, %input.1, %29, %4, %4)
  %23 : Tensor = aten::sigmoid(%fc1.1)
```

```
synthetic_forward (Tensor& gad_emb_p1,
                   Tensor& gemb_1,
                   Tensor& gwide_1)
{
  gwide_o1 = native::add(gwide_1, gself__mu, g4);
  gwide_n1 = native::mul(gwide_o1, gself__sigma);
  gwide_preproc_1 = native::clamp(gwide_n1, g5, g6);
  gemb_t_1 = native::transpose(gemb_1, g4, g7);
  gdp_u1 = native::bmm_cpu(gad_emb_p1, gemb_t_1);
  gdp_1 = native::flatten(gdp_u1, g4, g8);
  g19 = {gdp_1, gwide_preproc_1};
  gi_1 = native::cat(g19, g4);
  gfc1_1 = native::addmm_cpu(gfc_b, gi_1,
                             g29, g4, g4);
  g23 = native::sigmoid(gfc1_1);
  g24 = {g23};
  return g24;
```

Results

Graph Compiler

Machine:
Processor 3.2 GHz 6-Core Intel Core i7
Memory 32 GB 2667 MHz DDR4

Model	Ultra vs Pytorch Interpreter	Notes
LSTM	55x	Run predictor 5000 time
DeepAndWide	40x	Run predictor 5000 time
LLD6	20x	Run predictor 5000 time
ResNet18	6x	Run predictor 50 time
ResNet50	6x	Run predictor 50 time

Andy Fox, Edvard Ghazaryan 06.07.2021, EFS ID 42832446