Parametric ReLU 层

- 初始示例代码

初始示例代码

```
import numpy as np
from cuda import cudart
import tensorrt as trt

nIn, cIn, hIn, wIn = 1, 3, 3, 3  # 输入张量 NCHW
data = np.tile(np.arange(-4, 5, dtype=np.float32).reshape(1, hIn, wIn), (cIn, 1, 1))  # 输入数据

np.set_printoptions(precision=8, linewidth=200, suppress=True)
cudart.cudaDeviceSynchronize()

logger = trt.Logger(trt.Logger.ERROR)
builder = trt.Builder(logger)
network = builder.create_network(1 << int(trt.NetworkDefinitionCreationFlag.EXPLICIT_BATCH))
config = builder.create_builder_config()
inputT0 = network.add_input('inputT0', trt.DataType.FLOAT, (nIn, cIn, hIn, wIn))
#------------------------------------------------------------ --------------------# 替换部分
slopeLayer = network.add_constant((1, cIn, 1, 1), np.array([0.5, 1, 2], dtype=np.float32))  # 斜率张量，可广播到与 inputT0 相同大小即可，可以控制在全局、单维度、单元素的水平上的斜率
parsmetricReLULayer = network.add_parametric_relu(inputT0, slopeLayer.get_output(0))
#------------------------------------------------------------ --------------------# 替换部分
network.mark_output(parsmetricReLULayer.get_output(0))
engineString = builder.build_serialized_network(network, config)
engine = trt.Runtime(logger).deserialize_cuda_engine(engineString)
context = engine.create_execution_context()
_, stream = cudart.cudaStreamCreate()

inputH0 = np.ascontiguousarray(data.reshape(-1))
outputH0 = np.empty(context.get_binding_shape(1), dtype=trt.nptype(engine.get_binding_dtype(1)))
_, inputD0 = cudart.cudaMallocAsync(inputH0.nbytes, stream)
_, outputD0 = cudart.cudaMallocAsync(outputH0.nbytes, stream)

cudart.cudaMemcpyAsync(inputD0, inputH0.ctypes.data, inputH0.nbytes,
cudart.cudaMemcpyKind.cudaMemcpyHostToDevice, stream)
context.execute_async_v2([int(inputD0), int(outputD0)], stream)
cudart.cudaMemcpyAsync(outputH0.ctypes.data, outputD0, outputH0.nbytes,
cudart.cudaMemcpyKind.cudaMemcpyDeviceToHost, stream)
cudart.cudaStreamSynchronize(stream)

print("inputH0 :", data.shape)
print(data)
print("outputH0:", outputH0.shape)
print(outputH0)

cudart.cudaStreamDestroy(stream)
cudart.cudaFree(inputD0)
cudart.cudaFree(outputD0)
```

- 输入张量形状 (1,3,3,3)

$$\left[\left[\left[\begin{bmatrix} -4. & -3. & -2. \\ -1. & 0. & 1. \\ 2. & 3. & 4. \end{bmatrix}\begin{bmatrix} -4. & -3. & -2. \\ -1. & 0. & 1. \\ 2. & 3. & 4. \end{bmatrix}\begin{bmatrix} -4. & -3. & -2. \\ -1. & 0. & 1. \\ 2. & 3. & 4. \end{bmatrix}\right]\right]\right]$$

- 输出张量形状 (1,3,3,3)，就是 leaky ReLU，负数部分的斜率被改变

$$\left[\left[\left[\begin{bmatrix} -2. & -1.5. & -1. \\ -0.5 & 0. & 1. \\ 2. & 3. & 4. \end{bmatrix}\begin{bmatrix} -4. & -3. & -2. \\ -1. & 0. & 1. \\ 2. & 3. & 4. \end{bmatrix}\begin{bmatrix} -8. & -6. & -4. \\ -2. & 0. & 1. \\ 2. & 3. & 4. \end{bmatrix}\right]\right]\right]$$