RaggedSoftMax 层

初始示例代码

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

np.random.seed(97)
nIn, cIn, hIn, wIn = 1, 3, 4, 5  # 输入张量 NCHW
data0 = np.ones(cIn * hIn * wIn, dtype=np.float32).reshape(cIn, hIn, wIn)  # 输入数据
data1 = np.tile(2 * np.arange(hIn, dtype=np.int32), (cIn, 1)).reshape(cIn, hIn, 1)

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, (cIn, hIn, wIn))  # 两个张量都只要 3 维
inputT1 = network.add_input('inputT1', trt.DataType.INT32, (cIn, hIn, 1))
#---------------------------------------------------------- --------------------# 替换部分
raggedSoftMaxLayer = network.add_ragged_softmax(inputT0, inputT1)
#---------------------------------------------------------- --------------------# 替换部分
network.mark_output(raggedSoftMaxLayer.get_output(0))
#engine          = builder.build_engine(network,config)
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(data0.reshape(-1))
inputH1 = np.ascontiguousarray(data1.reshape(-1))
outputH0 = np.empty(context.get_binding_shape(2), dtype=trt.nptype(engine.get_binding_dtype(2)))
_, inputD0 = cudart.cudaMallocAsync(inputH0.nbytes, stream)
_, inputD1 = cudart.cudaMallocAsync(inputH1.nbytes, stream)
_, outputD0 = cudart.cudaMallocAsync(outputH0.nbytes, stream)

cudart.cudaMemcpyAsync(inputD0, inputH0.ctypes.data, inputH0.nbytes,
cudart.cudaMemcpyKind.cudaMemcpyHostToDevice, stream)
cudart.cudaMemcpyAsync(inputD1, inputH1.ctypes.data, inputH1.nbytes,
cudart.cudaMemcpyKind.cudaMemcpyHostToDevice, stream)
context.execute_async_v2([int(inputD0), int(inputD1), int(outputD0)], stream)
cudart.cudaMemcpyAsync(outputH0.ctypes.data, outputD0, outputH0.nbytes,
cudart.cudaMemcpyKind.cudaMemcpyDeviceToHost, stream)
cudart.cudaStreamSynchronize(stream)

print("inputH0 :", data0.shape)
print(data0)
print("inputH1 :", data1.shape)
print(data1)
print("outputH0:", outputH0.shape)
print(outputH0)

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

- 输入张量 0 形状 (3,4,5)

$$\begin{bmatrix}\begin{bmatrix}1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\end{bmatrix} \begin{bmatrix}1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\end{bmatrix} \begin{bmatrix}1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\\ 1. & 1. & 1. & 1. & 1.\end{bmatrix}\end{bmatrix}$$

- 输入张量 1 形状 (1,3,4,1)

$$\begin{bmatrix}\begin{bmatrix}0\\2\\4\\6\end{bmatrix} \begin{bmatrix}0\\2\\4\\6\end{bmatrix} \begin{bmatrix}0\\2\\4\\6\end{bmatrix}\end{bmatrix}$$

- 输出张量形状 (3,4,5)，每个 batch 都在指定长度 (1,2,3,4) 上计算了 Soft Max，其余元素变成 0
- 计算长度为 0 时输出值全为 0（每 batch 第一行），计算长度大于输入张量 1 的宽度时，存在访存越界（第 2 batch 最后一行红色数字），结果随机
- 这里只是恰好 $0.1862933 = \frac{e^1}{5e^1+e^0}$

$$\begin{bmatrix}\begin{bmatrix}0. & 0. & 0. & 0. & 0.\\ 0.5 & 0.5 & 0. & 0. & 0.\\ 0.25 & 0.25 & 0.25 & 0.25 & 0.\\ 0.167 & 0.167 & 0.167 & 0.167 & 0.167\end{bmatrix}\\ \begin{bmatrix}0. & 0. & 0. & 0. & 0.\\ 0.5 & 0.5 & 0. & 0. & 0.\\ 0.25 & 0.25 & 0.25 & 0.25 & 0.\\ 0.167 & 0.167 & 0.167 & 0.167 & 0.167\end{bmatrix}\\ \begin{bmatrix}0. & 0. & 0. & 0. & 0.\\ 0.5 & 0.5 & 0. & 0. & 0.\\ 0.25 & 0.25 & 0.25 & 0.25 & 0.\\ {\color{red}0.186} & {\color{red}0.186} & {\color{red}0.186} & {\color{red}0.186} & {\color{red}0.186}\end{bmatrix}\end{bmatrix}$$

- 该层两个输入张量只接受 3 维张量，否则报错：

```
[TRT] [E] 4: [raggedSoftMaxNode.cpp::computeOutputExtents::13] Error Code 4: Internal Error ((Unnamed
Layer* 0) [Ragged SoftMax]: Input tensor must have exactly 3 dimensions)
[TRT] [E] 4: (Unnamed Layer* 0) [Ragged SoftMax]: input tensor must have 2 non batch dimensions
[TRT] [E] 4: [network.cpp::validate::2871] Error Code 4: Internal Error (Layer (Unnamed Layer* 0)
[Ragged SoftMax] failed validation)
```

- 两个输入的维度要一致，否则报错：

```
[TRT] [E] 3: [network.cpp::addRaggedSoftMax::1294] Error Code 3: API Usage Error (Parameter check failed
at: optimizer/api/network.cpp::addRaggedSoftMax::1294, condition: input.getDimensions().nbDims ==
bounds.getDimensions().nbDims
```