Assertion 层

- 初始示例代码
- message

初始示例代码

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

nIn0, cIn0, hIn0, wIn0 = 1, 3, 4, 5  # 输入张量 NCHW
data0 = np.arange(nIn0 * cIn0 * hIn0 * wIn0, dtype=np.float32).reshape(nIn0, cIn0, hIn0, wIn0)  # 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))
profile = builder.create_optimization_profile()
config = builder.create_builder_config()
config.max_workspace_size = 1 << 30
inputT0 = network.add_input('inputT0', trt.DataType.FLOAT, (-1, -1, -1, 5))
profile.set_shape(inputT0.name, (1, 1, 1, 5), (1, 3, 4, 5), (2, 6, 8, 5))
config.add_optimization_profile(profile)
#----------------------------------------------------------------------------# 替换部分
# assert(not bool(inputT0.shape[2]-4)), 检测 inputT0 的次高维是否为 4
_H1 = network.add_shape(inputT0)

_H2 = network.add_slice(_H1.get_output(0), [2], [1], [1])
_C1 = network.add_constant([1], np.array([4], dtype=np.int32))

_H3 = network.add_elementwise(_H2.get_output(0), _C1.get_output(0), trt.ElementWiseOperation.SUB)

_H4 = network.add_identity(_H3.get_output(0))
_H4.get_output(0).dtype = trt.DataType.BOOL
'''
# 使用该分支会有报错
# [TRT] [E] 9: [graph.cpp::computeInputExecutionUses::549] Error Code 9: Internal Error ((Unnamed Layer* 5) [Unary]: IUnaryLayer cannot be used to compute a shape tensor)
_H5 = network.add_unary(_H4.get_output(0),trt.UnaryOperation.NOT)
_H5.get_output(0).dtype = trt.DataType.BOOL
_H6 = network.add_identity(_H5.get_output(0)) # 输出被检测表达式的值
_H6.get_output(0).dtype = trt.DataType.INT32
'''
'''
# 添加 add_assertion 时，使用 add_unary 会有报错：
# [TRT] [E] 9: [graph.cpp::computeInputExecutionUses::549] Error Code 9: Internal Error ((Unnamed Layer* 5) [Unary]: IUnaryLayer cannot be used to compute a shape tensor)
#_H5 = network.add_unary(_H4.get_output(0),trt.UnaryOperation.NOT)
#_H5.get_output(0).dtype = trt.DataType.BOOL
'''
```

```
'''
# 使用该分支总是报 assert 的警告，就算输入全为 True
_H0 = network.add_shape(inputT0)
_H1 = network.add_identity(_H0.get_output(0))
_H1.get_output(0).dtype = trt.DataType.BOOL
_H2 = network.add_assertion(_H1.get_output(0),"Assert infomation!")
_H3 = network.add_identity(_H1.get_output(0))
_H3.get_output(0).dtype = trt.DataType.INT32
'''
_H6 = network.add_identity(_H4.get_output(0))  # 注意相比上面分支少了 NOT 操作，检测表达式变成了 assert(bool(inputT0.shape[2]-4))
_H6.get_output(0).dtype = trt.DataType.INT32
_H7 = network.add_assertion(_H4.get_output(0), "inputT0.shape[1] is not 4!")  # assertion 层接受一个张量输入，没有张量输出
#------------------------------------------------------------------------------# 替换部分
network.mark_output(_H6.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()

context.set_binding_shape(0, data0.shape)

inputH0 = np.ascontiguousarray(data0.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 :", data0.shape)
print(data0)
print("outputH0:", outputH0.shape, outputH0.dtype)
print(outputH0)

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

- 输入张量形状 (1,3,4,5)

$$\left[\left[\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 5. & 6. & 7. & 8. & 9. \\ 10. & 11. & 12. & 13. & 14. \\ 15. & 16. & 17. & 18. & 19. \end{bmatrix} \begin{bmatrix} 20. & 21. & 22. & 23. & 24. \\ 25. & 26. & 27. & 28. & 29. \\ 30. & 31. & 32. & 33. & 34. \\ 35. & 36. & 37. & 38. & 39. \end{bmatrix} \begin{bmatrix} 40. & 41. & 42. & 43. & 44. \\ 45. & 46. & 47. & 48. & 49. \\ 50. & 51. & 52. & 53. & 54. \\ 55. & 56. & 57. & 58. & 59. \end{bmatrix}\right]\right]$$

- 遗留问题：
  - 无论 hIn0 是不是 4，都不会报错，且 _H6 的输出值恒为 1
  - 怎样利用 tensor 的形状做 assert?

---

message

```
    # 在初始示例代码中 _H7 定义的下面添加一行
    _H7.message = "edited message!"
```

- 输出结果同初始示例代码，改变了警告信息的内容