Constant 层

- 初始示例代码
- weights & shape

初始示例代码

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

nIn, cIn, hIn, wIn = 1, 3, 4, 5  # 输入张量 NCHW
data = np.arange(nIn * cIn * hIn * wIn, dtype=np.float32).reshape(nIn, cIn, hIn, wIn)  # 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()
#---------------------------------------------------------- --------------------# 替换部分
constantLayer = network.add_constant(data.shape, data)
#---------------------------------------------------------- --------------------# 替换部分
network.mark_output(constantLayer.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()

outputH0 = np.empty(context.get_binding_shape(0), dtype=trt.nptype(engine.get_binding_dtype(0)))
_, outputD0 = cudart.cudaMallocAsync(outputH0.nbytes, stream)

context.execute_async_v2([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(outputD0)
```

- 输出张量形状 (1,3,4,5)

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

weights & shape

```
constantLayer = network.add_constant([1], np.array([1], dtype=np.float32))
constantLayer.weights = data  # 重设常量数据
constantLayer.shape = data.shape  # 重设常量形状
```

- 输出张量形状 (1,3,4,5)，结果与初始示例代码相同
- Constant 层不支持 bool 数据类型