ONNX Runtime 图优化 Pass 总结

1. 常量/算子折叠与清理类

Pass	作用	应用场景	注意事项
eliminate_identity	删除恒等节点	避免无效节点计算	无风险
eliminate_nop_transpose	删除无效 Transpose	数据维度未变化时	注意动态 shape
eliminate_nop_pad	删除 Pad 无效节点	Pad 为 0 时	无风险
eliminate_nop_flatten	删除 Flatten 一维恒等操作	展平张量无需改变	无风险
eliminate_deadend	删除无输出或不可达节点	清理孤立节点	确保无调试输出依赖
eliminate_unused_initializer	删除未被节点引用的常量	减少模型大小	无风险
eliminate_duplicate_initializer	合并重复常量	减少模型大小	确保值完全相同
eliminate_shape_op	折叠静态 Shape 节点	静态输入 reshape	动态 shape 无法折叠
eliminate_shape_gather	折叠 Shape+Gather 常量	获取固定维度	动态 shape 不可折叠
eliminate_slice_after_shape	折叠 Shape→Slice 链	静态输入 reshape	同上
eliminate_nop_reshape	删除恒等 Reshape	shape 不变	无风险
eliminate_nop_with_unit	删除乘 1、加 0 节点	常量折叠	无风险
eliminate_common_subexpression	合并重复计算	节点重复	需确保语义一致

2. 算子融合类

Pass	作用	应用场景	注意事项
fuse_consecutive_transposes	合并连续 Transpose	避免多次转置	动态 shape 注意
fuse_consecutive_concats	合并连续 Concat	提高效率	Concat 维度需匹配

Pass	作用	应用场景	注意事项
fuse_consecutive_squeezes	合并连续 Squeeze	避免冗余	无风险
fuse_consecutive_unsqueezes	合并连续 Unsqueeze	避免冗余	无风险
fuse_consecutive_slices	合并连续 Slice	减少中间 tensor	需静态索引
fuse_consecutive_reduce_unsqueeze	合并 Reduce+Unsqueeze	避免中间节点	无风险
fuse_consecutive_log_softmax	合并连续 LogSoftmax	模型导出重复	需保持 axis 一致
fuse_matmul_add_bias_into_gemm	MatMul+Add → Gemm	高效矩阵运算	Bias 可广播
fuse_pad_into_conv	Pad+Conv → Conv	卷积前填充	动态 padding 注意
fuse_pad_into_pool	Pad+Pool → Pool	池化前填充	同上
fuse_transpose_into_gemm	Transpose+Gemm → Gemm	统一矩阵布局	axis 对齐
fuse_qkv	Q/K/V 切片+MatMul → 一步完成	Transformer 优化	Slice 必须可合并
fuse_concat_into_reshape	Concat+Reshape → Reshape	构建静态形状	有维度限制
adjust_slice_and_matmul	Slice+MatMul → 优化索引	Q/K/V 分片	索引必须静态

3. 输入/输出类型重写类

Pass	作用	应用场景	注意事项
rewrite_input_dtype	重写输入 dtype	量化/硬件适配	精度变化
rewrite_output_dtype	重写输出 dtype	量化/硬件适配	精度变化

4. 其他特殊优化

Pass	作用	应用场景	注意事项
replace_einsum_with_matmul	Einsum → MatMul	简化矩阵运算	仅支持可降为 MatMul 的 Einsum
lift_lexical_references	子图优化	子图节点提升	高级场景
split_init / split_predict	分割计算	编译器优化	需要保持数据依赖

Pass	作用	应用场景	注意事项
eliminate_nop_concat	删除恒等 Concat	单一输入 Concat	无风险
eliminate_nop_expand	删除恒等 Expand	Expand 不改变 shape	无风险

5. 量化场景结合示意

优化前

```
graph.input (float32)
      │
      ▼
  Cast(to=float16)
      │
      ▼
  QuantizeLinear(scale, zero_point)
      │
      ▼
  Conv_int8 / MatMul_int8
      │
      ▼
  DequantizeLinear(scale, zero_point)
      │
      ▼
  Cast(to=float32)
      │
      ▼
graph.output (float32)
```

优化后（rewrite_input_dtype + rewrite_output_dtype）

```
graph.input (uint8)
      │
      ▼
  Conv_int8 / MatMul_int8
      │
      ▼
graph.output (uint8)
```

- 输入直接量化，输出直接返回量化值
- 删除冗余 Cast/Quantize/Dequantize 节点
- 减少中间 tensor，提升推理性能

---

6. 总结

- **常量/算子折叠**：消除无用节点和常量，提高图清洁度
- **算子融合**：合并连续算子或 Slice/MatMul 优化，减少中间计算和内存访问
- **输入/输出 dtype 重写**：量化或硬件适配，删除冗余 Cast/Quantize/Dequantize
- **其他特殊优化**：Einsum 转 MatMul、子图提升、分割计算等，提高图可部署性和性能