**Concat 텐서 그룹수와 보존율 분석 보고서**

1. **서론**
   1. **배경**
      1. **[2.1.1 Bottleneck 반복수와 보존율 분석 보고서]**의 후속 연구: concat의 입력 텐서 그룹수가 많아질수록 채널마다 min/max scale의 차이가 커질 확률이 크기 때문에 넓어진 분포를 양자화 과정에서 제대로 표현하지 못해 정보 손실이 증가할 수 있다. 또한 concat된 텐서 값들이 특정 범위에 치우치거나, 값들의 분포가 비선형적인 경우에 양자화 과정에서 작은 값들의 정보가 손실되거나 큰 값이 clipping 되어 정보 왜곡이 발생할 수 있다.  
          따라서 직전 연구에서는 C3 모듈에 대한 Bottleneck 반복 수에 따른 보존율 상관관계를 알아보았기 때문에, 본 실험에서는 C2f에서 concat의 입력 텐서 그룹수와 보존율의 상관 관계를 중점적으로 다룬다.
   2. **실험 준비** 
      1. 실험 모델
         1. v8n\_1111
         2. v8n\_1221
         3. v8n\_2222
         4. v8n\_2332
   3. **목적**

\* 보존율 = (NmAP / GmAP) \* 100 (GPU에서의 성능 대비 NPU에서 재현율)

* + 1. 실험 1 : concat 텐서 그룹 수에 따른 보존율 변화 분석  
       : [모델 1]을 기준으로 concat의 입력 텐서 그룹수와 Bottleneck 모듈 반복 수가 증가시킨 [모델 2, 3, 4]를 비교하여 상관 관계를 분석한다.
  1. **평가 지표**
     1. 보존율
     2. NPU 성능

1. **본론**
   1. **실험 설계**
      1. 데이터셋
         1. 각 모델들이 학습에 사용된 데이터와 테스트에 사용된 데이터는 동일
            1. 학습 데이터 : 제공받은 train set과 자체적으로 수집한 data set을 병합
            2. 테스트 데이터 : 제공받은 test set
      2. 모델 설계

\* 1111 ~ 2332는 Backbone에서 C2f 모듈 내의 Bottleneck 반복 횟수

* + - 1. v8n\_1111: Backbone에서의 Bottleneck 반복 횟수를 (1, 1, 1, 1)로 설정한 모델
      2. v8n\_1221: Backbone에서의 Bottleneck 반복 횟수를 (1, 2, 2, 1)로 설정한 모델
      3. v8n\_2222: Backbone에서의 Bottleneck 반복 횟수를 (2, 2, 2, 2)로 설정한 모델
      4. v8n\_2332: Backbone에서의 Bottleneck 반복 횟수를 (2, 3, 3, 2)로 설정한 모델
  1. **실험 결과 및 분석**
     1. 실험 1 : Concat 텐서 그룹수에 따른 연산 보존율 비교
        1. 실험 항목
           1. 대조군1 : v8n\_1111  
              실험군1 : v8n\_1221, 실험군2 : v8n\_2222, 실험군3 : v8n\_2332  
              : v8n 모델에 대해 동일 조건으로 Backbone의 Bottleneck 반복 횟수만 변경하여 Concat 텐서 그룹 수를 제외한 변인을 통제
        2. **가설** **– Concat 텐서 그룹수가 증가할수록 보존율이 감소할 것이다.**  
           : Concatenation 과정에서 텐서 그룹수가 증가하면 데이터 값들의 분포가 커진다. 이러한 현상으로 양자화 과정에서 8bit 연산 환경에서 범위가 넓어진 데이터 분포 값으로 인해 정보 손실이 크게 발생할 수 있다. 또한 데이터 분포에 outlier로 인한 악영향을 받거나 큰 값들이 clipping되어 정보가 왜곡될 수 있다. 따라서 v8n\_1111 모델에서 Concatenation 텐서 그룹수가 증가할수록 양자화 손실 폭이 증가하여 보존율이 낮아질 것이다.
        3. 결과  
           - 평가 기준: 모델 성능

|  |  |
| --- | --- |
| [그래프 1] | [그래프 2] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | v8n\_1111 | v8n\_1221 | v8n\_2222 | **v8n\_2332** |
| NPU mAP (%) | 29.42 | 33.20 | 32.66 | **34.42** |
| NPU/GPU (%) | 52.16 | 58.37 | 57.72 | **59.59** |

* + - 1. 분석 :

: 반복 수 1111을 기준으로 1221로 증가했을 때 모델의 보존율이 52.16%에서 58.37%로 약 6.2% 증가했다.  
 반복 수 1221을 기준으로 2222로 증가했을 때 모델의 보존율이 58.37%에서 57.72%로 약 0.65% 감소했다.

반복 수 2222을 기준으로 2332로 증가했을 때 모델의 보존율이 57.72%에서 59.59%로 약 1.87% 증가했다.  
실험 결과를 통해 **[그래프 1]**과 **[그래프 2]**의 동일한 성능 증감 경향성을 볼 수 있듯 가설과는 다르게 Concat 텐서 그룹수가 증가함에 따라 보존율 성능이 증가하는 것을 확인했다. **[1.3 Backbone 구조 변경 분석 보고서]**의 분석 내용으로 Bottleneck 반복 횟수를 균일하게 조정할 경우 성능이 증가할 수 있다는 가능성이 본 보고서의 내용에도 공존할 수 있기 때문에 가설과는 다른 실험 결과가 도출된 것으로 보인다.

1. **결론**
   1. **주요 결론**
      1. 실험 1 : Concat 텐서 그룹수에 따른 연산 보존율 비교

: 해당 실험의 결과는 기존의 가설과 반대의 결과가 나타났다. 직전 연구에서 Bottleneck의 반복수가 늘어나면 보존율이 감소하는 것을 확인했는데, 이 결과를 같이 고려했을 때 C2f의 경우에는 반복 수가 늘어날 경우 보존율이 증가하므로 영향을 미치는 다른 요인이 있다는 결론을 얻을 수 있었다.

* 1. **향후 연구 방향**
     1. 실험 계획  
        : C2f 모듈에서는 예상과 다른 결과를 얻었으며 추가적인 요인에 대한 분석이 필요한데, **[1.4 P2 적용 및 Backbone 구조 변경 분석 보고서]**의 향후 연구인 v8s\_P2\_2221 모델에 대한 경량화 방법으로 C2f, C3, C3G 모듈을 비교하여 적용할 것이기 때문에 이 과정에서 C2f 모듈이 가지는 의외의 특성을 함께 고려하며 실험을 진행할 필요가 있다.
     2. 필요 연구  
        : 앞서 말한 것과 같이 기존의 가설과 다른 실험 결과가 나오게 한 요인에 대한 추가 연구가 필요하다.