TANGO

NAS 파라미터 최적화 기술

성 명 김중헌 소 속 고려대학교

2025 제4회 COMMUNITY CONFERENCE

TANGO

Target Adaptive No-code neural network Generation and Operation framework

주 관 ETRI (TANGO)
주 최 과학기술정보통신부 IITP 정보통신기획평가원
문 의 parkjb@etri.re.kr / 042-860-5565

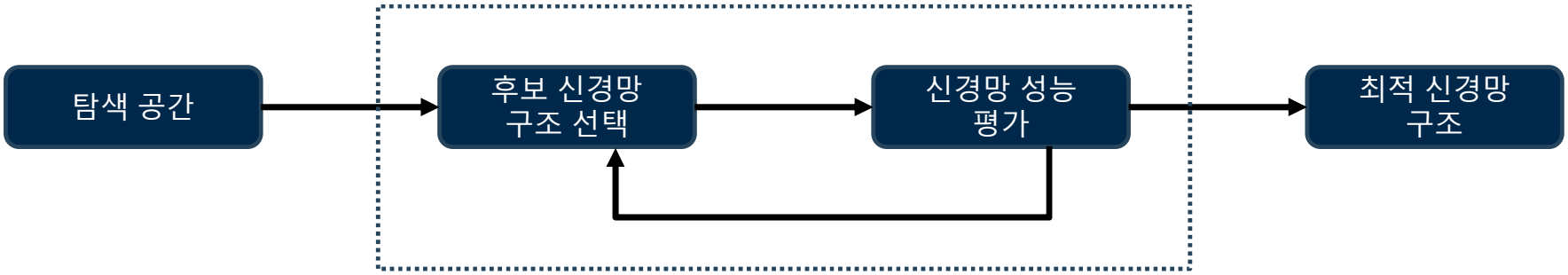
후 원 LOG labup weda tesla system (사)한국인공지능협회 SNUH 고려대학교 KOREA UNIVERSITY 홍익대학교 HONGIK UNIVERSITY 중앙대학교 RTst KETI 한국전자기술연구원 AIVN SUREDATA ACRYL 하늘소프트 TTA 한국정보통신기술협회 Telecommunications Technology Association

content

목 차

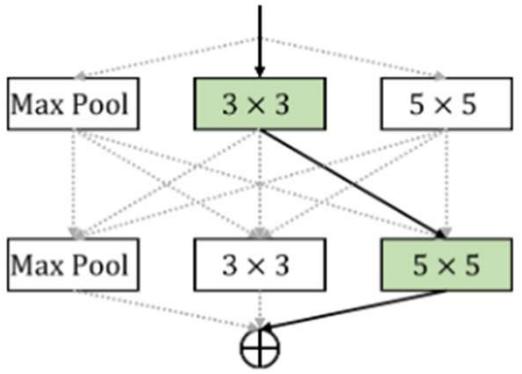
Neural Architecture Search

- **신경망 구조 탐색 기술 (Neural Architecture Search, NAS)**은 Automated Machine Learning (AutoML) 파이프라인 관점에서의 연구 분야로, 인공 신경망 구조 생성의 자동화에 대한 연구임
- 일반적인 신경망 구조 탐색 기술은 생성될 수 있는 후보 신경망 구조들의 집합을 정의하는 **탐색 공간과**, **탐색 공간 내의 후보 신경망 구조**들을 선택하고 성능을 평가하여 최적의 구조를 결정하는 방법인 탐색 전략으로 구성
- 신경망 구조 탐색은 이러한 심층 신경망 구조 설계 과정을 알고리즘으로 **자동화하여 적은 비용으로 타겟 태스크에 최적인 신경망 구조를 찾기** 위한 기술



One-Shot Neural Architecture Search

- 탐색 과정에서 필수적인 각 후보 신경망에 대한 **반복 학습**은 **막대한 시간적, 연산적 비용을 초래하여 실질적인 적용이 어려움**
- 신경망 구조 탐색 연구는 탐색 결과 신경망 구조의 **성능 개선** 뿐만 아니라, **탐색 비용을 줄이기 위한 다양한 기법**에 대한 연구 진행
- 각 후보 신경망 구조들을 독립적으로 새로 학습하지 않고 **유사한 부분에 대한 신경망 가중치를 공유하도록 하는 기법**이 등장
- 모든 후보 게이트 연산을 포함하는 **슈퍼넷 (Supernet)** 구축하여 **가중치를 공유**
- 슈퍼넷을 한 번 학습하여 다중 아키텍처 동시 평가



One-Shot NAS

One-Shot Neural Architecture Search

DARTS

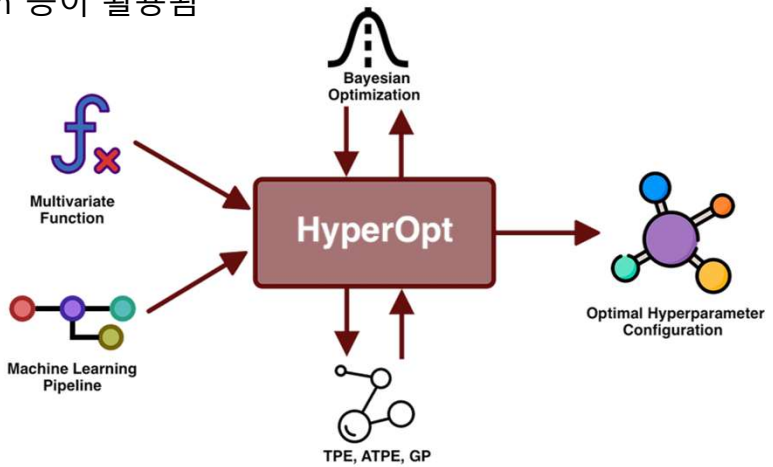
- 연속적 탐색을 통해 구조를 미분 가능하게 변환함
- 각 후보 연산에 구조 파라미터를 할당하여 경사하강법(Gradient Descent)으로 최적화함
- 모든 경로를 동시에 학습하므로, **메모리 사용량이 높고 학습 불안정성**이 발생할 수 있음
- **[장점]** 미분 기반 탐색으로 **빠른 수렴 가능**
- **[단점] 모든 경로 병렬 최적화로 인한 비효율성과 불안정성**

ProxylessNAS

- 메모리 및 하드웨어 제약을 해결하기 위해 **이진 선택 기반 접근법**을 사용함
- 여러 후보 중 이진 선택을 통해 **매 반복마다 하나의 경로만 활성화**하여 학습함
- 이를 통해 **메모리 효율성과 학습 안정성을 동시에 확보**함
- **[장점]** 이진 선택을 통한 실제 **디바이스 친화적 최적화**
- **[단점] 탐색 안정성 확보**를 위한 **파라미터 조정 필요**

Hyperparameter Optimization (HPO)

- 모델 성능에 직접적인 영향을 주는 학습 파라미터(learning rate, batch size, momentum 등)를 **자동으로 탐색**하는 과정
- 목표는 **최적의 성능(accuracy, reward 등)을 달성하면서 계산 비용을 최소화**하는 **하이퍼파라미터 조합**을 찾는 것
- 주로 Bayesian Optimization, Grid/Random Search, Evolutionary Algorithm 등이 활용됨
- 최근에는 NAS와 결합하여 공동 최적화 프레임워크로 발전함.



Hyperparameter Optimization

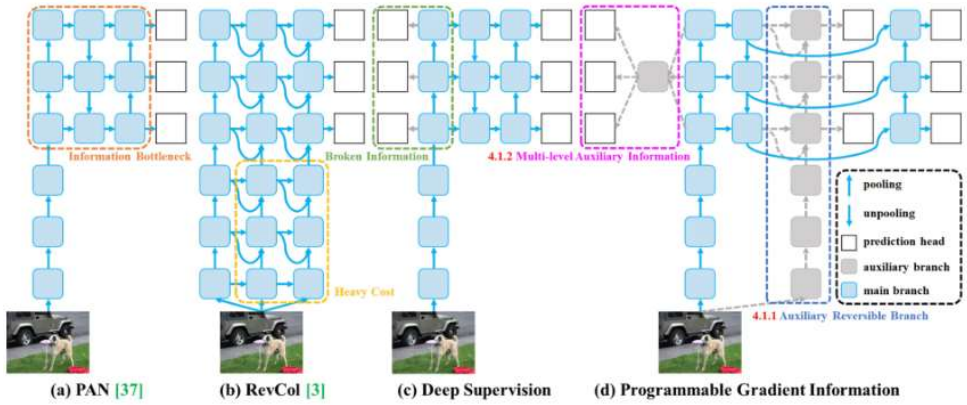
YOLO

- 실시간 객체 탐지(Real-Time Object Detection)를 위한 대표적인 딥러닝 모델
- 입력 이미지를 한 번만 통과시켜(**one forward pass**) 물체의 위치(Bounding Box)와 클래스를 동시에 예측함
- CNN 기반의 단일 네트워크로 구성되어 **속도와 정확도 간의 균형**이 뛰어남
- **고속 처리 속도:** 실시간 애플리케이션(자율주행, CCTV 등)에 적합
- **End-to-End 학습:** 단일 네트워크에서 탐지와 분류를 동시에 수행
- **적은 계산량:** 효율적인 구조로 임베디드·모바일 디바이스에도 적용 가능

구성요소	역할	설명
Backbone	특징 추출	입력 이미지로부터 시각적 특징(feature map) 생성
Neck	다중 스케일 통합	다양한 크기의 객체 탐지를 위한 feature pyramid 구성
Head	탐지 출력	Bounding Box 좌표, 객체 존재 확률, 클래스 확률을 예측

YOLOv9

- YOLO 시리즈는 v1 → v5 → v8 → v9 등으로 발전하며, **정확도 향상·경량화·NAS 통합**이 지속적으로 이루어짐
- YOLO 시리즈의 최신 버전으로, **정확도와 속도의 균형을 극대화**한 모델임
- 기존 YOLOv8의 특징 추출과 탐지 헤드 구조를 개선하여 고해상도·복잡 장면에서도 향상된 성능을 보임
- 실시간 객체 탐지(Real-time Object Detection), 자율주행, 스마트 팩토리, 드론 영상 분석 등 다양한 분야에서 활용 가능.



YOLOv9NAS

- YOLOv9의 **Neck 부분에서** NAS 적용

```
# gelan backbone
backbone:
  [
   [-1, 1, Silence, []],

   # conv down
   [-1, 1, Conv, [32, 3, 2]],  # 1-P1/2

   # conv down
   [-1, 1, Conv, [64, 3, 2]],  # 2-P2/4

   # elan-1 block
   [-1, 1, RepNCSPELAN4, [128, 128, 64, 1]],  # 3

   # avg-conv down
   [-1, 1, AConv, [240]],  # 4-P3/8

   # elan-2 block
   [-1, 1, RepNCSPELAN4, [240, 240, 120, 1]],  # 5

   # avg-conv down
   [-1, 1, AConv, [360]],  # 6-P4/16

   # elan-2 block
   [-1, 1, RepNCSPELAN4, [360, 360, 180, 1]],  # 7

   # avg-conv down
   [-1, 1, AConv, [480]],  # 8-P5/32

   # elan-2 block
   [-1, 1, RepNCSPELAN4, [480, 480, 240, 1]],  # 9
  ]
```

```
# elan head
head:
  [
   # elan-spp block
   [-1, 1, SPPELAN, [480, 240]],  # 10

   # up-concat merge
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 7], 1, Concat, [1]],  # cat backbone P4

   # elan-2 block
   [-1, 1, MRepNCSPELAN4, [360, 360, 180, 1]],  # 13

   # up-concat merge
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 5], 1, Concat, [1]],  # cat backbone P3

   # elan-2 block
   [-1, 1, MRepNCSPELAN4, [240, 240, 120, 1]],  # 16

   # avg-conv-down merge
   [-1, 1, AConv, [180]],
   [[-1, 13], 1, Concat, [1]],  # cat head P4

   # elan-2 block
   [-1, 1, MRepNCSPELAN4, [360, 360, 180, 1]],  # 19 (P4/16-medium)

   # avg-conv-down merge
   [-1, 1, AConv, [240]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5

   # elan-2 block
   [-1, 1, MRepNCSPELAN4, [480, 480, 240, 1]],  # 22 (P5/32-large)
```

```
   # routing
   [5, 1, CBLinear, [[240]]], # 23
   [7, 1, CBLinear, [[240, 360]]], # 24
   [9, 1, CBLinear, [[240, 360, 480]]], # 25

   # conv down
   [0, 1, Conv, [32, 3, 2]],  # 26-P1/2

   # conv down
   [-1, 1, Conv, [64, 3, 2]],  # 27-P2/4

   # elan-1 block
   [-1, 1, MRepNCSPELAN4, [128, 128, 64, 1]],  # 28

   # avg-conv down
   [-1, 1, AConv, [240]],  # 29-P3/8
   [[23, 24, 25, -1], 1, CBFuse, [[0, 0, 0]]], # 30

   # elan-2 block
   [-1, 1, MRepNCSPELAN4, [240, 240, 120, 1]],  # 31

   # avg-conv down
   [-1, 1, AConv, [360]],  # 32-P4/16
   [[24, 25, -1], 1, CBFuse, [[1, 1]]], # 33

   # elan-2 block
   [-1, 1, MRepNCSPELAN4, [360, 360, 180, 1]],  # 34

   # avg-conv down
   [-1, 1, AConv, [480]],  # 35-P5/32
   [[25, -1], 1, CBFuse, [[2]]], # 36

   # elan-2 block
   [-1, 1, MRepNCSPELAN4, [480, 480, 240, 1]],  # 37

   # detect
   [[31, 34, 37, 16, 19, 22], 1, DualDDetect, [nc]],  # Detect(P3, P4, P5)
  ]
```

NAS 적용 블록

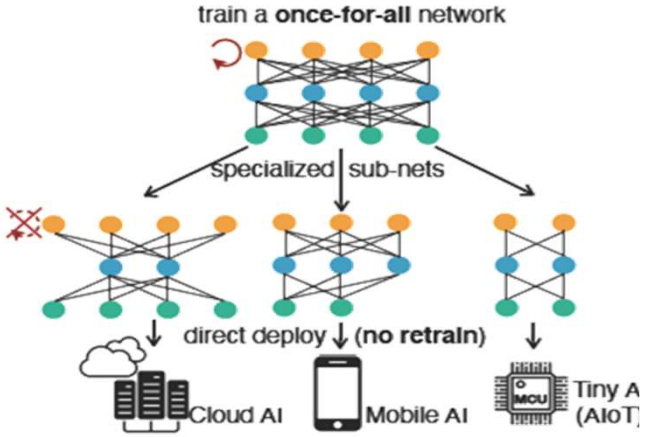
YOLOv9NAS

- 탐색 공간이 넓어질수록 최적 신경망 구조를 포함할 가능성 높아짐
- 그러나 탐색 공간의 확장은 컴퓨팅 리소스 및 시간 측면에서 높인 비용 초래
- 성능이 검증된 참조 신경망 구조, **변경 규칙**을 활용하여 **상대적으로 좁은 탐색 공간에서도 좋은 성능의 신경망 구조를 찾음**
- **변경 규칙**: **Convolutional 커널 크기 변화 : 3×3 또는 5×5, 활성화 함수 (activation) 변화 : ReLU, LeakyReLu, Mish**

```
OPS = {
    '3x3_relu_BNCSP': lambda c1, c2, c3, c4, c5:
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='relu', kernel_size=3),

    '3x3_leaky_BNCSP': lambda c1, c2, c3, c4, c5 :
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='leaky', kernel_size=3),

    '3x3_mish_BNCSP': lambda c1, c2, c3, c4, c5:
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='mish', kernel_size=3),

    '3x3_silu_BNCSP': lambda c1, c2, c3, c4, c5:
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='silu', kernel_size=3),

    '5x5_relu_BNCSP': lambda c1, c2, c3, c4, c5:
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='relu', kernel_size=5),

    '5x5_leaky_BNCSP': lambda c1, c2, c3, c4, c5:
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='leaky', kernel_size=5),

    '5x5_mish_BNCSP': lambda c1, c2, c3, c4, c5:
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='mish', kernel_size=5),

    '5x5_silu_BNCSP': lambda c1, c2, c3, c4, c5:
    MRepNCSPELAN4Layer(c1, c2, c3, c4, c5, act='silu', kernel_size=5),
}
```

타겟 적응형 NAS

- 대부분의 NAS는 고정된 하드웨어나 환경에서만 최적 구조를 탐색함
- Cloud 환경에서 학습된 모델을 Mobile이나 Edge에서 그대로 적용➜ 연산량·메모리 제약 불일치로 인해 성능 저하 발생
- 동일한 모델이라도 타깃 디바이스별 최적 구조는 다름
- 즉, **디바이스 제약(연산량·속도)에 맞춰 경량/중간/대형 네트워크 구조를 선택적 탐색** 가능해야 함



구분	기존 NAS	타겟 적응형 NAS
탐색 범위	고정된 디바이스 기준	디바이스별 맞춤형 탬색
구조 형태	단일 최적 모델	Once-for-All Supernet
효율성	재탐색 필요	Sub-net 즉시 적용
적용 대상	제한적	Cloud / Mobile / Tiny 모두 대응

타겟 적응형 YOLOv9NAS

- Once-for-All 구조의 핵심 구성 요소로, **레이어와 search_scope**에 따라 Supernet 내부에서 하드웨어 맞춤형 Sub-network를 자동 선택하도록 설계됨
- 이를 통해 추가 학습 없이 **디바이스별 최적 모델을 즉시 생성**할 수 있음

```python
def select_ops_for_layer(layer_idx: int, stage: int, search_scope: str):
    """
    레이어/스테이지/스코프에 따라 ops_mod.OPS에서 후보 함수들만 추려 반환.
    - stage: stride=2 통과 횟수로 근사.
    - search_scope: small/medium/large/full
    """
    # 스코프 상한
    scope_max = {
        'small':  {'kernels': {'3x3'},                'acts': {'relu','silu'}},
        'medium': {'kernels': {'3x3','5x5'},          'acts': {'relu','silu'}},
        'large':  {'kernels': {'3x3','5x5'},          'acts': {'relu','silu','leaky'}},
        'full':   {'kernels': {'3x3','5x5','7x7'},    'acts': {'relu','silu','leaky','mish'}},
    }
    max_k = scope_max.get(search_scope, scope_max['medium'])['kernels']
    max_a = scope_max.get(search_scope, scope_max['medium'])['acts']

    # 스테이지 선호(예시 정책)
    stage_pref = [
        {'kernels': {'3x3'},                'acts': {'relu','silu'}},              # stage 0
        {'kernels': {'3x3'},                'acts': {'relu','silu'}},              # stage 1
        {'kernels': {'3x3','5x5'},          'acts': {'relu','silu','leaky'}},      # stage 2
        {'kernels': {'3x3','5x5','7x7'},    'acts': {'relu','silu','leaky','mish'}}, # stage 3+
    ]
    pref = stage_pref[min(stage, len(stage_pref)-1)]

    allow_k = max_k & pref['kernels']
    allow_a = max_a & pref['acts']

    # 원본 OPS에서 필터링
    cands = []
    # 정렬은 보기 편하게 이름순(원하면 커널 크기/act 우선순위로 정렬 가능)
    for name, fn in sorted(ops_mod.OPS.items()):
        k, a = _parse_op_name(name)
        if k in allow_k and a in allow_a:
            cands.append(fn)
    return cands
```

타겟 적응형 YOLOv9NAS 실험 결과

- 본 연구에서는 YOLOv9-M 기반 NAS 모델을 다양한 **디바이스 환경(Small, Medium, Large, Full)에 맞추어 최적화함**
- NAS 탐색 과정에서 각 타깃 Scope별로 선택된 **최종 아키텍처 구조를 표로 제시함**

Scope	Architecture
Small	“m_13”: **1** , “m_16”: **0**, “m_19”: **0**, “m_22”: **0**, “m_28”: **0**, “m_31”: **0**, “m_34”: **2**, “m_37”: **1**
Medium	“m_13”: **2** , “m_16”: **0**, “m_19”: **2**, “m_22”: **0**, “m_28”: **0**, “m_31”: **2**, “m_34”: **2**, “m_37”: **0**
Large	“m_13”: **2** , “m_16”: **3**, “m_19”: **3**, “m_22”: **0**, “m_28”: **0**, “m_31”: **3**, “m_34”: **0**, “m_37”: **1**
Full	“m_13”: **2** , “m_16”: **0**, “m_19”: **2**, “m_22”: **0**, “m_28”: **0**, “m_31”: **0**, “m_34”: **2**, “m_37”: **0**

타겟 적응형 YOLOv9NAS 실험 결과

- 타겟 적응형 NAS 적용 결과, **소형(Small) 모델에서 Precision이 73.4%로 가장 높게 향상**됨
- 이는 불필요한 경로를 제거하여 **정확한 탐지 성능(Precision)** 을 극대화한 결과임
- 다만, **Recall(재현율)** 은 일부 감소하여 탐지 범위가 좁아지는 경향을 보임
- **mAP50 기준, 소형 모델(53.2%)에서 가장 높은 종합 성능**을 기록
- 결론적으로, **타겟 적응형 NAS는 대형 모델보다 소형·경량 모델 환경에서 효과적**임
- NAS는 전체적인 탐지 성능 향상보다, **모바일 · 엣지 환경에서 효율적 구조를 탐색하는 데 강점**을 보임

Scope	Precision (%)	Recall (%)	mAP50 (%)
Small	73.4	46.6	53.2
Medium	68.9	41.8	47.6
Large	68.5	41.3	46.9
Full	67.1	42.8	48.5

TANGO

감사합니다.

주 관 ETRI (TANGO)

주 최 과학기술정보통신부 IITP 정보통신기획평가원

문 의 parkjb@etri.re.kr / 042-860-5565

후 원 LAB labiup weda teslo System (사)한국인공지능협회 Korea Artificial Intelligence Association SNUH 서울대학교병원 고려대학교 KOREA UNIVERSITY 홍익대학교 HONGIK UNIVERSITY CAU 중앙대학교 RTst Reliable & Trustworthy

KETI 한국전자기술연구원 AIVN SUREDATA ACRYL 하늘소프트 TTA 한국정보통신기술협회 Telecommunications Technology Association