

# VANAS: 신경망 구조 탐색을 위한 시각적 분석 시스템

## VANAS: A Visual Analytics System for Neural Architecture Search

최지원  
Jiwon Choi  
성균관대학교  
Sungkyunkwan University  
jasonchoi3@g.skku.edu

홍권  
Gwon Hong  
성균관대학교  
Sungkyunkwan University  
hk98121@skku.edu

조재민  
Jaemin Jo  
성균관대학교  
Sungkyunkwan University  
jmjo@skku.edu

### 요약문

본 논문에서는 신경망 구조 탐색(Neural Architecture Search, NAS) 알고리즘의 탐색 결과를 분석하고 시각화하는 시스템인 VANAS를 제안한다. 방대한 탐색 결과를 효율적으로 전달하기 위해 간선의 중요도 및 기여도를 정량화하는 알고리즘을 고안하고 이를 기반으로 사용자에게 구조를 추천하는 알고리즘을 구현하였다. 이를 기반으로 탐색 공간에 대한 전반적인 특징을 확인할 수 있는 오버뷰와 심층 신경망 구조를 설계하고 분석할 수 있는 사용자 인터페이스를 디자인하였다. VANAS를 통하여 NAS 분야의 표준 벤치마크 데이터 중 하나인 NAS-Bench-101을 분석한 결과, 신경망을 구성하는 정점 및 간선의 중요도를 확인하고, 시스템이 제안하는 간선을 추가하여 정확도를 점진적으로 개선할 수 있음을 확인하였다.

### 주제어

인공지능, 정보시각화, 딥러닝, 신경망 구조 탐색, 인간-컴퓨터 상호작용

## 1. 서론

딥러닝 모델의 성능은 모델의 신경망 구조를 이루는 연산자들의 구성과 배치 순서에 따라 크게 달라진다는 것이 알려져 있다. 이에 자동으로 효과적인 신경망 구조를 탐색하는 신경망 구조 탐색(Neural Architecture Search, NAS) 알고리즘들이 ‘자동화된 머신러닝’(Automated Machine Learning, AutoML) 분야의 일환으로 등장하였다 [3, 4, 5]. 강화 학습 또는 One-shot 학습 등 여러 방법을 토대로 한 NAS 알고리즘들이 제안되었다. 이러한 알고리즘들은 공통적으로 주어진 예산(시간 또는 GPU 성능 등)내에서 다양한 구조의 신경망을 학습시키고 가장 성능이 좋은 구조를 찾는 데 목적을 둔다.

그러나, NAS 알고리즘이 탐색하는 공간의 방대한 크기로 인해 어떤 신경망 구조가 탐색되었는지, 신경망

구조의 어떤 특징이 정확도를 변화시키는지 인간이 이해하기 어렵다는 한계가 있다. 딥러닝 모델의 구조 또는 하이퍼파라미터를 시각화하는 선행 연구가 존재하나[6, 7, 8, 9] 고정된 하나의 구조만을 탐색할 수 있다는 한계를 지닌다. NAS 알고리즘의 탐색 결과를 그래프 형태로 보여주는 선행 연구[2]가 존재하나 단순한 노드-링크 다이어그램을 적용하고 있어 다각적인 분석은 불가능하다.

본 연구에서는 NAS 과정에서 산출되는 다양한 신경망의 구조와 데이터(정확도, 학습시간 등)에 시각적 분석을 수행할 수 있는 정보시각화 시스템인 VANAS를 제안한다. NAS 분야의 표준 벤치마크 데이터 중 하나인 NAS-Bench-101을 VANAS를 통하여 분석한 결과, 신경망을 구성하는 정점 및 간선의 중요도를 확인하고, 시스템이 제안하는 간선을 추가하여 정확도를 점진적으로 개선할 수 있음을 확인하였다.

## 2. 문제 정의

### 2.1 데이터 추상화

신경망 구조 탐색 알고리즘이 산출하는 데이터를 추상화하기 위해 약 42만 3천 개의 신경망 구조에 대한 탐색 결과를 가지고 있는 NAS-Bench-101 데이터 세트를 사용하였다 [1]. 해당 데이터 세트의 각 행이 포함하는 데이터는 아래와 같다.

- **사용된 정점 개수 및 종류:** 정점의 종류는 INPUT, OUTPUT,  $1 \times 1$  CONV,  $3 \times 3$  CONV,  $3 \times 3$  MAXPOOL 이며 하나의 구조는 최대 7개의 정점을 포함할 수 있다.
- **정점 간 연결 정보(간선):** 상기한 정점간 연결 정보로써, INPUT 정점으로부터 OUTPUT 정점으로서의 텐서의 흐름을 나타낸다. 간선에는 방향성이 존재하며, 싸이클은 존재하지 않는다. 하나의 구조는 최대 9개의 간선을 포함할 수 있다.

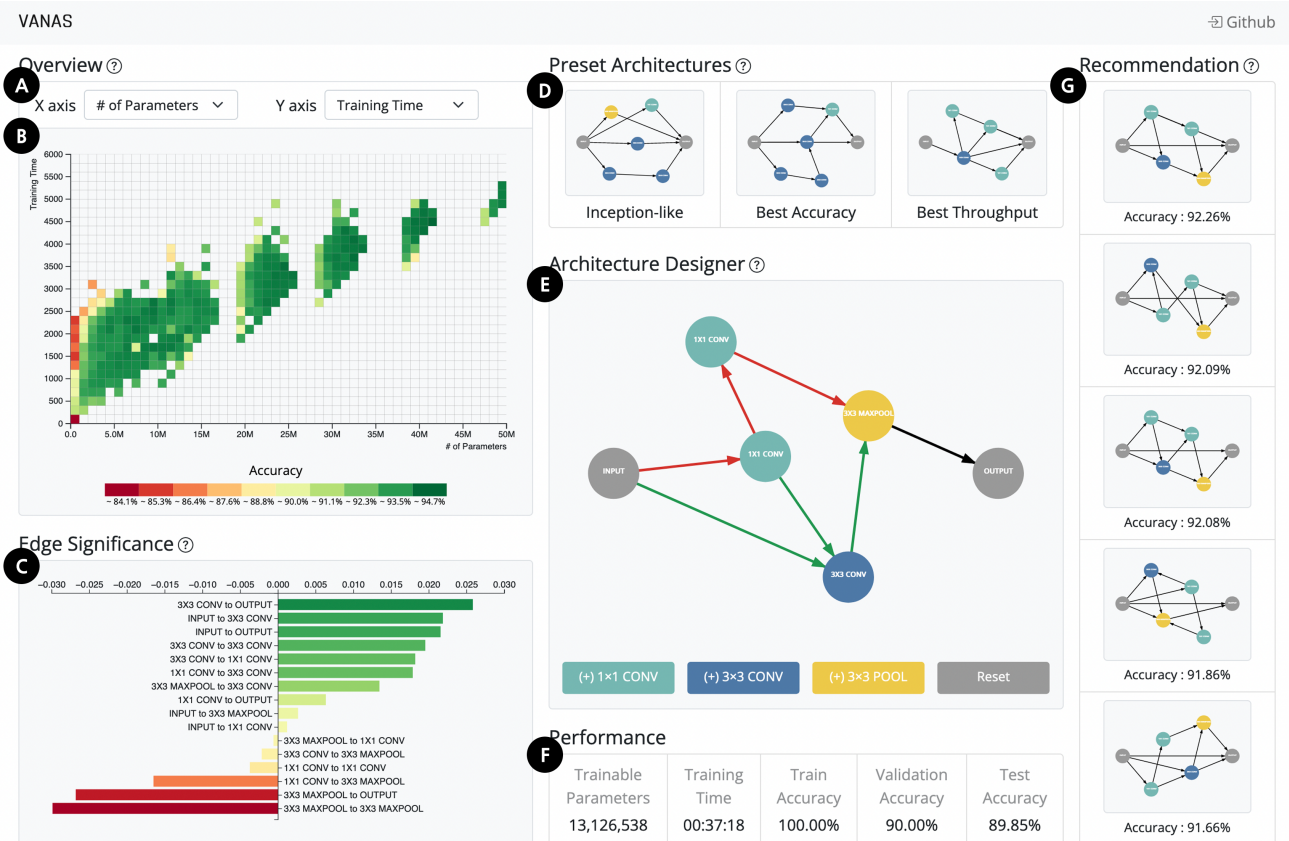


그림 1: VANAS의 인터페이스.

- 학습/검증/평가 정확도(%): 신경망에 CIFAR-10 데이터를 학습시켰을 때 얻어진 학습/검증/평가 정확도이다.
- 학습 시간(초): 학습에 소요된 총 시간이다.
- 총 가중치의 수(개): 상기한 정점과 간선을 가지고 신경망을 구성했을 때 신경망에 포함되는 학습 가능한 가중치의 개수를 의미한다.

## 2.2 태스크 추상화

본 연구에서는 NAS 결과 분석에 대해 아래 네 개의 사용자 태스크를 정의하고 이를 기반으로 디자인을 도출하였다.

1. 전반적인 결과 탐색 (태스크 1): 사용자는 학습 시간/총 가중치의 수/검증 정확도/평가 정확도 등 NAS 과정에서 산출되는 정량적인 결과에 대해 전반적으로 점검하고 이해한다.
2. 신경망 구조 설계 및 결과 질의 (태스크 2): 사용자는 신경망 구조에 포함되는 정점과 간선을 인터랙티브하게 설계하고 해당 구조의 성능을 확인한다.
3. 간선 중요도/기여도 확인 (태스크 3): 사용자는 정점과 간선이 정확도에 끼치는 영향을 파악하기

위해 특정 종류의 두 정점을 잇는 간선의 중요도와 기여도를 확인한다.

4. 점진적 개선 (태스크 4): 사용자는 정점과 간선을 추가하여 기존 신경망 구조의 성능을 개선한다.

## 3. 중요도/기여도 정량화 및 구조 추천 알고리즘

본 연구에서는 방대한 탐색 결과를 효율적으로 전달하기 위해 간선의 중요도 및 기여도를 정량화하는 알고리즘을 고안하고 이를 기반으로 사용자에게 구조를 추천하는 알고리즘을 구현하였다.

간선의 중요도란 특정 종류의 두 정점(가령, 3×3 CONV to 1×1 CONV)을 잇는 간선이 신경망 구조에 존재할 때 정확도에 얼마나 영향을 끼치는지를 정량화한 수치이다. 간선의 기여도란 특정 신경망 구조가 주어질 때 두 정점을 잇는 간선이 평균적으로 정확도에 얼마나 기여했는지를 정량화한 수치이다.

구조 추천 알고리즘은 사용자가 선택한 정점들에 대해 최대의 정확도를 얻을 수 있는 간선의 배치를 제안한다. 사용자는 정점의 종류뿐만 아니라 정점 사이에 반드시 존재해야 하는 간선을 입력으로 제공할 수도 있다. 이 경우, 추천 알고리즘은 해당 간선들은 유지한 채 추가적인 간선을 삽입하여 정확도를 개선할 수 있는지 확인한다.

### 3.1 간선의 중요도 계산

간선의 종류(예를 들어 3×3 CONV 에서 3×3 CONV)에 따른 중요도는 아래와 같이 계산된다.

$S$  : 탐색 공간

$S_E$  : 탐색 공간 내 모든 간선의 종류

$$G = (V, E) \in S, e_i \in S_E$$

$eval(x)$  : 신경망 구조  $x$ 의 평가 정확도

$$S(e_i) = \frac{\sum_{G' \in \{G' | e_i \in E(G')\}} eval(G')}{|\{G' | e_i \in E(G')\}|} - \frac{\sum_{G' \in \{G' | e_i \notin E(G')\}} eval(G')}{|\{G' | e_i \notin E(G')\}|}$$

### 3.2 간선의 기여도 계산

특정 신경망 구조를 이루는 개별 간선들의 기여도는 아래와 같이 계산된다.

$$C(e_j, G) = \frac{\sum_{G' \in \{E(G_{sub}) | e_j \in E(G_{sub}) \subset E(G)\}} eval(G')}{|\{E(G_{sub}) | e_j \in E(G_{sub}) \subset E(G)\}|} - \frac{\sum_{G' \in \{E(G_{sub}) | e_j \notin E(G_{sub}) \subset E(G)\}} eval(G')}{|\{E(G_{sub}) | e_j \notin E(G_{sub}) \subset E(G)\}|}$$

만약 해당 간선을 제외한 부분 그래프가 올바른 신경망 구조가 아니라면, 해당 간선의 기여도는 계산하지 않았다.

### 3.3 신경망 구조 추천 알고리즘

사용자의 입력을 바탕으로 신경망 구조를 추천하기 위해 사용자가 입력한 정점들과 정점의 종류와 개수가 같고 사용자가 입력한 간선의 개수보다 간선의 개수가 같거나 많은 신경망들을 대상으로 완전 탐색을 수행해 사용자가 입력한 정점과 간선을 모두 포함하는 신경망들을 선별한다. 선별된 신경망들은 평가 정확도를 기준으로 정렬 후 사용자에게 추천된다.

## 4. VANAS 디자인

Shneiderman 의 시각화 만트라[10]를 따라 VANAS 인터페이스를 디자인하였다. 그림 1 은 크게 3 개의 뷰로 구성된 VANAS 인터페이스를 보여준다. VANAS 인터페이스는 1) 탐색 결과의 특징을 전반적으로 확인할 수 있는 탐색 오버뷰(A, B)와 간선 중요도 뷰(C), 2) 인터랙티브하게 구조를 편집하고 간선의 기여도를 확인할 수 있는 구조 디자이너(E)와 구조 설계에 참고하면 좋을 특징적인 신경망 구조를 보여주는 사전 설정 뷰(D), 구조 디자이너의 상태를 바탕으로 완성된 구조를 추천해주는 구조 추천 뷰(G), 3) 구조의 성능을 확인할 수 있는 성능 뷰(F)로 구성된다.

### 4.1 탐색 오버뷰(A, B) 및 간선 중요도 뷰(C)

탐색 오버뷰와 간선 중요도 뷰를 통해 사용자는 탐색 결과의 특징을 전반적으로 확인해 탐색 공간에 대한 이해를 높이고 필요에 따라 신경망 구조를 선별할 수 있다.

Overview ②



그림 2: x, y 축에 표현할 결과 변수를 선택하여 변수 간 관계를 확인할 수 있다.

탐색 오버뷰(A, B)는 전체 탐색 공간에서 학습 시간, 총 가중치의 수, 평가 정확도 사이의 관계를 히트맵을 통해 시각화한다. 대상 데이터의 행 수가 수십만 행이기 때문에 확장성을 고려하여 산점도(scatterplot)가 아닌 히트맵을 사용하였다. 양 축에 비닝을 적용하기 때문에, 하나의 칸은 다수의 행을 포함할 수 있으며 이 중 평가 정확도가 가장 높은 구조의 정확도를 색깔을 통해 표현하였다.

그림 2 와 같이 탐색 오버뷰 설정(A)에서 X 축과 Y 축을 사용자의 탐색 조건에 맞게 설정할 수 있다. 또한, 각 칸에 커서를 올려 해당 칸의 속하는 구조들의 최대 평가 정확도를 확인할 수 있다. 마지막으로, 칸을 클릭하여 해당 칸의 신경망 구조를 구조 디자이너와 성능 뷰를 통해 조회할 수 있다.

간선 중요도 뷰(C)는 3.1 절의 간선 중요도를 가로 막대 그래프로 시각화하였다. 간선의 종류에 해당하는 막대를 클릭하여 구조 디자이너에 해당 간선을 추가할 수 있다.

### 4.2 구조 디자이너(E) 및 구조 추천 뷰(G)

구조 디자이너와 구조 추천 뷰를 통해 사용자는 탐색 오버뷰와 간선 중요도 뷰를 통해 얻은 탐색 공간에 대한 이해를 바탕으로 구체적인 구조를 설계할 수 있다.

구조 디자이너(E)는 초기에 INPUT 정점과 OUTPUT 정점이 표시된다. 정점 추가 버튼을 통해 원하는



연산자를 구조 디자이너에 추가하고 추가된 정점을 하단으로 드래그해 정점을 삭제할 수 있다. 출발 정점과 도착 정점을 연달아 클릭해 간선을 추가하고, 추가한 간선을 클릭해 삭제할 수 있다.

Architecture Designer ②

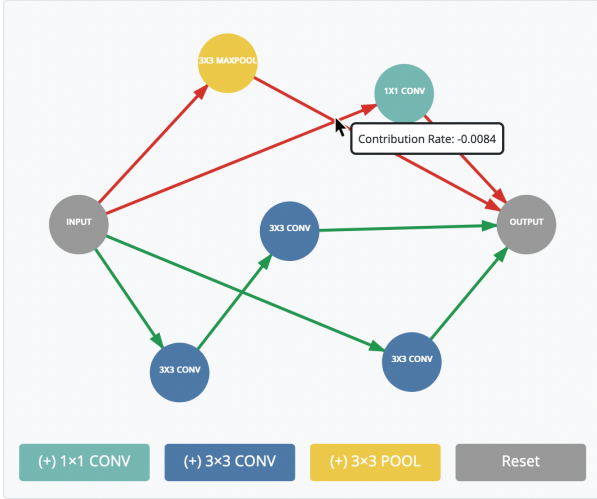


그림 3: 완성된 신경망의 간선에 마우스를 올려 해당 간선의 기여도를 확인할 수 있다.

신경망 구조를 완성한 이후 그림 3 과 같이 간선의 기여도를 확인할 수 있다. 양의 기여도를 가진 간선은 초록색으로, 음의 기여도를 가진 간선은 빨간색으로, 기여도를 계산할 수 없는 간선은 검은색으로 표시된다. 간선에 마우스를 올려 정확한 기여도를 확인할 수 있다.

신경망 구조는 INPUT 정점에서 OUTPUT 정점까지의 경로가 존재하는 유향 비순환 그래프이다. 타당하지 않은 신경망 구조의 입력은 탐색 공간에 대응되지 않으므로 깊이 우선 탐색을 이용해 1) INPUT 정점과 OUTPUT 사이에 경로가 존재하는지, 2) 그래프에 사이클이 존재하는지, 3) 사용자가 입력한 정점들과 간선들이 INPUT 정점과 OUTPUT 정점 사이의 경로에 존재하는지 검증한다. 1)과 2)의 경우 경고 메시지를 출력하고, 3)의 경우 경로에 존재하지 않는 정점들과 간선들을 투명하게 표시하여 사용자가 해당 정점과 간선들을 경로 상에 있게끔 수정할 수 있도록 유도한다.

사전 설정 뷰(D)를 통해 신경망 구조 설계에 참고하면 좋은 특징적인 신경망 구조를 보여준다. 이를 클릭하여 사전 설정을 구조 디자이너로 불러올 수 있다.

구조 추천 뷰(G)는 구조 디자이너의 사용자 입력을 바탕으로 신경망 구조를 추천한다. 간선이 추가되어있지 않다면 사용자가 추가한 정점들에 대해 최대의 평가 정확도를 얻을 수 있는 간선의 배치를 추천한다. 간선을 추가로 입력한다면 해당 간선들은 유지한 채 추가적인 간선을 삽입하여 만들 수 있는 신경망 구조 중 평가

정확도가 높은 신경망 구조를 추천한다. 추천받은 신경망 구조를 클릭하여 구조 디자이너로 불러올 수 있다.

### 4.3 성능 뷰(F)

사용자가 성능을 확인하고 싶은 신경망 구조가 있다면 성능 뷰를 통해 해당 신경망 결과의 성능을 확인할 수 있다.

### 4.4 디자인 고려사항

VANAS 를 이루는 각 뷰들은 Shneiderman 의 시각화 만트라[10]와 일관성을 고려하여 디자인되었다. VANAS 의 각 뷰(4.1, 4.2, 4.3 절)는 Shneiderman 의 시각화 만트라[10]의 세 단계에 대응된다. 이들은 인간의 정보 인지 방향[11]을 고려하여 좌상단에서 우하단으로 배치되어 사용자로 하여금 자연스럽게 시각화 만트라 단계 밟을 수 있도록 하였다.

각 뷰와 뷰에서의 상호작용은 일관성을 고려하여 디자인되었다. 가령, 정확도에 대해 일관적인 색깔 인코딩을 사용하였으며, 모든 뷰의 각 요소에 커서를 올렸을 때 툴팁을 제공하고, 클릭을 통해 세부 정보를 확인할 수 있도록 하였다.

## 5. 사례 탐구

본 연구에서 제안한 VANAS 시스템을 통하여 NAS-Bench-101 데이터를 분석하였다.

전반적인 탐색 결과를 확인하기 위해 탐색 오버뷰에서 ‘총 가중치 수’와 ‘학습 시간’에 대한 히트맵을 그려보았다. 예상했던 것처럼 가중치 수가 많을수록 학습 시간이 대체로 오래 걸린다는 것을 확인할 수 있었다. 다음으로 ‘총 가중치 수’와 ‘평가 정확도’에 대한 히트맵을 그려보았고, 가중치 수가 적으면서 평가 정확도가 현저히 낮은 신경망 구조들을 확인할 수 있었다. 특이점으로, 가중치 수와는 관계없이 정확도가 90% 이상으로 높은 구조가 존재함을 확인하였다. 이를 통해 NAS-Bench-101 에서 제공하는 최소한의 가중치 수로도 구조를 잘 디자인하면 벤치마크 시나리오에서 사용된 CIFAR-10 데이터를 높은 정확도로 분류할 수 있다는 것을 확인하였다.

간선 중요도 뷰에서는 3x3 CONV to OUTPUT 간선의 중요도가 가장 높음을 확인할 수 있었다. 다음으로 중요도가 높은 간선은 INPUT to 3x3 CONV 간선으로 두 간선 모두 3x3 CONV 정점을 포함하고 있었다. 또한 3x3 MAXPOOL to 3x3 MAXPOOL 간선은 중요도가 가장 낮았고 이외에도 3x3 MAXPOOL to 3x3 CONV 간선을 제외한 모든 3x3



MAXPOOL 이 포함된 간선은 중요도가 낮음을 확인할 수 있었다.

다음으로는 프리셋의 ‘Best Accuracy’ 를 클릭하여 가장 큰 평가 정확도를 가진 신경망 구조를 구조 디자이너로 불러와 성능을 확인해보았다. 3×3 CONV 정점이 4 개로 매우 많고, 간선의 수 또한 9 개로 최대 개수임을 확인할 수 있었다. 이는 앞서 간선 중요도 뷰에서 3×3 CONV 정점이 중요도가 높은 간선들에 많이 포함되어 있다는 발견과 일치하였다. 히트맵에서 ‘Best Accuracy’ 구조를 확인했을 때 총 가중치 수와 학습 시간 모두 큰 편에 속했으나, 예상외로 가장 큰 신경망 구조는 아니었다. 오히려 가중치 수와 학습시간이 가장 큰 구조는 정확도가 ‘Best Accuracy’ 구조보다 더 낮음을 확인할 수 있었다.

마지막으로, 구조 디자이너를 INPUT 및 OUTPUT 정점만 남도록 초기화한 후 새로운 구조를 디자인해 보았다. 앞서 간선 중요도 뷰를 살펴보면서 3×3 CONV 가 평균적으로 좋은 영향을 미친 것을 토대로 3×3 CONV 정점을 두 개 추가하고 구조 추천 뷰의 구조 중 가장 정확도가 높은 구조를 불러왔다. 성능 뷰를 통해 확인한 정확도는 93.81%로 만족스러우나 총 가중치 수와 학습 시간이 각각 약 20M, 40 분으로 크다고 판단되어 3×3 CONV 를 제거한 후 1×1 CONV 를 추가하고 추천 뷰에서 새 신경망 구조를 불러왔다. 정확도는 93.30%로 소폭 줄어들었으나, 총 가중치 수가 약 12M, 30 분으로 큰 폭으로 개선된 것을 확인할 수 있었다.

## 6. 구현

시스템에 대한 접근성을 높이기 위해 웹 기반 서버리스 앱 형태로 구현하였다. NAS-Bench-101 의 원본 데이터가 수 기가바이트이기 때문에, 웹 브라우저에서 실시간으로 분석하기 위해서는 경량화가 필요하다. 본 연구에서는 중복으로 저장되는 데이터 키 삭제, 희소 행렬 비트 마스킹 등의 방법으로 주어진 데이터 세트를 수십 메가바이트의 경량화된 JSON 파일로 변환하여 처리하였다. VANAS 시스템과 가공된 NAS-Bench-101 데이터 세트는 <https://github.com/jason-choi-skku/vanas> 에서 오픈 소스로 확인할 수 있다.

## 7. 결론 및 향후 연구

본 논문에서는 신경망 구조 탐색 알고리즘의 탐색 결과를 분석하고 시각화하는 시스템 VANAS 를 제안하였다. VANAS 는 Shneiderman 의 시각화 만트라[10]에 따라 NAS 알고리즘의 탐색 결과를 분석할 수 있는 사용자 인터페이스를 제공한다. 또한, 탐색

결과를 이해할 수 있는 여러 정량화된 수치와 알고리즘을 설계하였다. VANAS를 통해 NAS 알고리즘 결과의 표준 데이터 세트를 분석한 결과 탐색 공간을 분석하고 시스템의 제안을 바탕으로 정확도를 점진적으로 개선하여 신경망을 설계할 수 있음을 확인하였다.

향후 연구로 NAS 에 관심이 있는 사용자에게 대한 보다 심도 있는 사용자 실험을 진행하여 인터페이스를 평가하고 개선할 수 있을 것이다. NAS-Bench-101 데이터는 주어진 조건하에 가능한 모든 구조에 대한 평가 데이터를 포함하고 있으나, 탐색 공간이 더욱 방대한 경우에는 모든 구조에 대해 평가를 미리 해두는 것이 힘들다. 이 경우에는 사용자가 설계한 구조를 실시간으로 실험 및 평가할 수 있도록 한다.

## 사사의 글

본 연구는 “성균관대 SW 중심대학사업단 심화 R&D 산학프로젝트”의 지원을 받아서 수행되었다.

## 참고 문헌

1. Ying, Chris, et al. "Nas-bench-101: Towards reproducible neural architecture search." *International Conference on Machine Learning*. PMLR, 2019.
2. Neural Network Intelligence NAS Visualization, <https://nni.readthedocs.io>
3. Elsken, Thomas, Jan Hendrik Metzen, and Frank Hutter. "Neural architecture search: A survey." *The Journal of Machine Learning Research* 20.1 (2019): 1997–2017.
4. Liu, Chenxi, et al. "Progressive neural architecture search." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
5. Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." *arXiv preprint arXiv:1611.01578* (2016).
6. Samek, Wojciech, Thomas Wiegand, and Klaus-Robert Müller. "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models." *arXiv preprint arXiv:1708.08296* (2017).
7. Hohman, Fred, et al. "Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations." *IEEE transactions on visualization and computer graphics* 26.1 (2019): 1096–1106.
8. Wang, Zijie J., et al. "CNN explainer: Learning convolutional neural networks with interactive visualization." *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2020): 1396–1406.
9. Wang, Qianwen, et al. "Atmseer: Increasing transparency and controllability in automated machine learning." *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019.
10. Shneiderman, Ben. "The eyes have it: A task by data type taxonomy for information visualizations." *The craft of information visualization*. Morgan Kaufmann, 2003. 364–371.
11. Cooper, Alan, et al. "About face 4 : 인터랙션 디자인의 본질" (최윤석 외 3명 역). 에이콘, 2014. 485