[Paper Review] Improving Neural Architecture Search Image Classifiers via Ensemble Learning (1)

세 번째 논문 리뷰! 오늘은 앙상블 모델을 활용해 모델의 성능을 높이는 예제를 다룬 논문을 읽어보았다. 이번 리뷰는 원문을 번역해 이해했기 때문에 약간의 오류(?)가 있을 수 있다. 원문은 아래 링크에 첨부한다.

목차

NAS(Network Architecture Search)로 최고의 신경망 구조를 찾는 과제는 연구자들로 하여금 상당한 시간, 자원을 필요로 한다. 일단 우수한 신경망 블록을 알아내면, 사전에 정의된 매개변수 예산제약 조건 하에서 직접 블록들을 조합하는 작업이 필요하다. 이를 대체할 수 있는 기술로 본 논문에서는 앙상블을 소개한다. 특히 여러 개의 소규모 신경망들을 자동으로 조합해 최종 모델을 구성하는 'AdaNAS' 알고리즘을 다룬다. 또한, 이전에 학습한 앙상블 모델을 기반으로 소규모 신경망을 반복적으로 생성하는 'distillation' 기술도 다룬다. 최종적으로 모델의 파라미터 수를 늘리지 않으면서 정확도를 개선하는 실험을 하고, CIFAR-10과 CIFAR-100 데이터셋을 비교한다.

서로 다른 예산 제약을 갖는 애플리케이션에 적용하기 위해서는 각각의 예산제약에 맞는 모델을 설계해야 한다. 이러한 각각의 작은 모델들을 조합해 모델(tower?)를 쌓는 방법으로 앙상블이 있는데, 앙상블 모델은 개별 하위 모델들보다 훨씬 강력하다. 또한, 각각의 조립품들은 모델의 출력이 병렬로 계산되기 때문에 문제상황별로 독립적으로 훈련될 수 있다. 본 논문에서는 동일한 매개변수 개수로 단일 대형모델보다 앙상블 모델이 더 나은 성능을 보이는지 확인하고, 앙상블의 하위 신경망을 순차적으로 학습하는 과정에서 이전 학습으로부터의 정보를 활용할 수 있는지 확인해본다. 뿐만 아니라, 앙상블 기술을 통해 각각의 블록으로부터 완전히 자동으로 tower를 구축할 수 있는지 확인한다.
특히, 앙상블의 자동 생성에 있어, 이전 학습 정보를 사용하여 성능 좋은 하위 모델을 반복적으로 자동생성하는 'AdaNAS' 알고리즘을 다루며, BAN (Born Again Networs)을 활용한 아이디어를 살펴보고, 이를 확장 적용한 AKD (Adaptive Knowledge Distillation)의 결과를 비교한다. 마지막으로는 ShakeDrop과 같은 데이터 증강 및 정규화 작업을 사용하지 않을 때에 비해 CIFAR-100의 성능을 비교한다.

이전에 이루어진 (Dutt et al., 2018) 연구와 앙상블 모델의 차이점은 averaging layer를 소프트맥스 계층 앞에 두어 branche들을 더 촘촘하게 만드는 것이다. 이

러한 작업을 통해 앙상블 모델은 타 모델에 비해 높은 성능을 낼 수 있다.
이 두 모델의 차이는 대표적으로 다음과 같이 구분할 수 있다. 1) 하위 신경망을 순차적으로 train 한다. 이 과정을 통해 knowledge distillation (하위 모델에서 큰 모델로 지식을 전달하는 과정) 2) average layer를 여러 branch 들의 가중평균으로 계산한다. 이를 통해 순차적으로 train된 각 하위 네트워크에 매번 새로운 가중치가 적용될 수 있다. 3) 3개 이상의 하위 신경망에 대해 더 나은 정확도를 보인다.

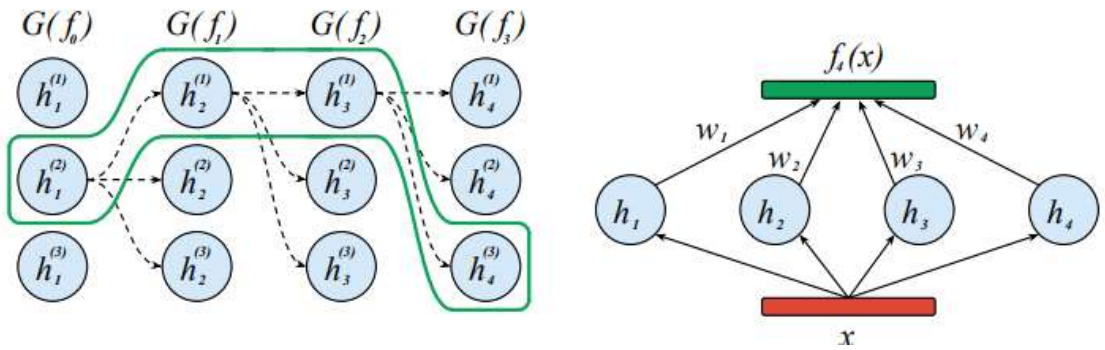
신경망 설계에 대해 기술한 최신 연구인 (Zoph & Le, 2016; Zoph et al., 2017)에서는 작은 데이터셋 (ex. CIFAR-10)에서 모델 block을 search한 후 Imagenet 같은 큰 데이터셋에 적용한다. 이러한 작업은 이전보다 향상된 cell 구조를 전달할 수 있는 새로운 search space (i.e., NASNet)를 만든다는 점에서 이롭다.
모델 압축 (Model Compression)과 지식 증류 (Knoledge Distillation)은 큰 앙상블 모델에서 작은 서브 모델로 예측력을 전달하는 기법이다. 작은 student 모델은 큰 teacher 모델의 예측력과 더불어 선택적으로 Ground Truth model에 대해 학습하는 것을 목표로 한다.

3. AdaNAS algoritm

3번째 섹션에서는 모델들을 조합하는 알고리즘을 다룬다.

3.1 General algorithm

왼쪽 그림에서는 총 3개의 검색 공간 (search space)로 이루어진 구조가 4번 반복된다. 여기에서 최적의 subnetwork로 이루어진 최종 앙상블이 만들어지며 이는 오른쪽 그림과 같다. 각각의 subnetwork들은 그들 각각의 가중치가 스케일링되어 출력을 구성한다.



앙상블 매커니즘

여기에서는 S = {(x1, y1), ..., (xm, ym)}과 같은 i.i.d.에 따라 도출된 훈련/검증 데이터셋으로 지도학습 시나리오를 설계한다. 빈 앙상블 f0로 시작하는 반복 알고리즘의 I 번째 단계에서 subnetworks h1, h2, ..., hI를 생성해 앙상블 fI를 구성한다. 이와 같이 각 반복에서 하위 네트워크의 후보군을 정한다.
생성자 G는 이전 반복 단계에서의 앙상블(H_i = G(F_(i-1))을 기반으로 하위 네트워크의 후보를 제안하는 역할을 한다. 이때, 모델은 후보 subnetwork의 매개변수 p_i,j를 독립적으로 훈련한다. 훈련 알고리즘은 이전 앙상블로부터의 distillation loss 뿐 아니라, classification loss를 최소화하는 θ를 찾는다.

이전 앙상블과 subnetwork들은 새로운 후보 subnetwork의 훈련 결과에 영향을 받지 않는다. 이렇게 공통의 연산 그래프를 공유함으로써, 단일 GPU로도 효율적으로 후보 subnetworks를 훈련시킬 수 있다.
모든 후보 subnetwork가 훈련되면, 그 결과를 f-1단계의 앙상블에 추가한다. 이때, 선택되는 값은 전체 앙상블의 loss를 최소화하는 값이다. 이렇게 최적의 subnetwork인 h(*)_i가 h_i에 추가된다. 아래와 같이 훈련된 하위 subnetwork에 가중치를 적용한 값들의 summation으로 f_i를 구성한다.

$$f_i = \sum_{k=1}^{i} w_k \cdot h_k, \qquad (1)$$

loss of generality가 없는 경우 h_k를 k번째 반복에서 선택된 subnetwork의 logit으로 정의한다. 여기서 혼합 가중치는 다중 클래스 분류 문제에서의 소프트맥스 활성화함수와 같은 방식으로 각 subnetwork의 output에 적용된다. 다음 실험 섹션에서는 위의 f_i가 이전 단계에서의 subnetwork logit의 평균으로 계산된 단순한 알고리즘으로 훈련한다.

혼합 가중치 벡터(w^(K))는 최종 앙상블의 classification loss를 최소화하도록 훈련된다. 훈련 동안 이 혼합 가중치 벡터와 후보 subnetwork((h_i)^(k))를 번갈아가며 최적화한다. 모델은 i번째 subnetwork가 선택되면 다음 반복으로 넘어가는 방식으로 진행되는 알고리즘을 거쳐 최종 앙상블 fI를 출력한다.

```
# AdaNAS algorithm (pseudo code)
# Input: S, I, G

f_0 <- 0
for i = 1 to I do
        H_i <- G(f_(i-1))
    for h_i(j) in H_i do
        theta(j) <- argmin_theta L_h(S, h_i(j)(theta)) +
                                                    L_KD(S, f_(i-1), h_i(j)*theta)
        w(j) <- argmin_w L_f(S, sum(w_k(j))*h_k + w_i(j)*h_i(j))
    end for
    j* <- argmin_j L_f(S, sum(w_k(j)*h_k + w_i(j) * h_i(j)*theta(j))
    h_i <- h_i(j*)*(theta(j*))
    w <- w(j*)
    f_i <- sum(w_k*h_k)

end for
return f_I
```