Resource efficient performance improvement method using Knowledge Distillation and ViT

이 름 이재정 지도교수 유종빈

연구배경

임베디드 환경과 같은 컴퓨터 리소스가 한정되어 있는 환경에서 딥러닝 모델이 돌아갈 수 있도록 해야 하는 문제 이전에, 딥러닝 모델을 학습시키기 위해선 GPU라는 값 비싼 장비가 필요하다. 이러한 이유로 대부분의 상황에서 GPU라는 컴퓨터 리소스는 개인에게 있어서 한정되어 있으며, 주어진 한정된 GPU를 효율적으로 활용하면서 모델의 성능을 최대한 향상시키는 것은 주요한 문제이다.

따라서 본 연구에서는 teacher 모델의 gradient가 필요없어 GPU 메모리의 효율적인 knowledge distillation 방법과 inductive bias가 낮아 학습을 계속해서 시킬 수록 성능이 향상될 수 있는 가능성을 가지고 있는 vision transformer 아키텍처를 활용하여 지속적인 knowledge distillation을 통한 학습 싸이클을 만들었을 시에 vision transformer 아키텍처 모델의 성능을 극한으로 끌어올릴 수 있는지 검증해 본다.

연구진행과정

*** 학습 데이터 셋 선정** : cifar-100

*** 학습 모델 선정**

pretrained resnet50

pretrained DeiT(Data-efficient image Transformer)

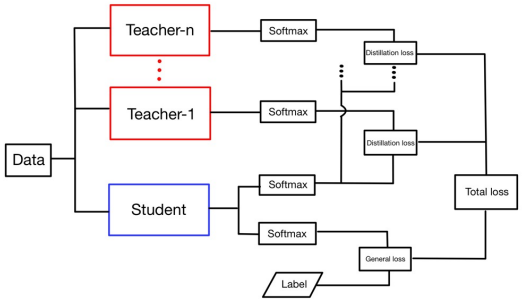
컴퓨터 리소스가 제한된 환경에서 최대한 모델의 성능을 끌어올리는 것을 목표로 했으므로 Imagenet에 pretrain된 모델들을 선정했으며, vision transformer 모델 또한 파라미터 수가 적은 DeiT 모델을 사용했다. DeiT 모델은 classification을 위한 class token과 함께 distillation token이 있어 vision transformer 모델 중 knowledge distillation 시 student로 활용 하기에도 적절하기 때문이다. 추가로 vision transformer와는 다른 knowledge 형식을 담고 있을 CNN 아키텍처인 resnet50도 teacher로 활용해 보았다.

*** Knowledge distillation 방식**

response-based knowledge distillation 사용.

teacher model의 output값으로 knowledge distillation.

distillation loss function은 soft와 hard를 모두 사용하여 비교하였으며, teacher를 늘리기 위해 distillation loss function을 n개의 teacher가 될 수 있도록 변경.



- n개의 teacher를 위해 변경한 soft distillation function

$$L_{soft} = (1-\alpha)L_{CE}(\psi(Z_s), y) + \frac{\alpha\tau^2}{n}\sum_{i=1}^{n} KL(\psi(Z_s/\tau), \psi(Z_{t_i}/\tau))$$

KL divergence로 student와 teacher의 결과 값들 간의 분포를 고려한 distillation loss function.

- n개의 teacher를 위해 변경한 hard distillation function

$$L_{hard} = (1-\alpha)L_{CE}(\psi(Z_s), y) + \frac{\alpha}{n}\sum_{i=1}^{n} L_{CE}(\psi(Z_s/\tau), \psi(Z_{t_i}/\tau))$$

student와 teacher의 결과 값들 간의 cross entropy loss 후 산술평균한 distillation loss function.

*** knowledge distillation 학습 싸이클**

pretrained resnet50(1)의 classifier를 바꿔 fine tuning

pretrained DeiT(2)의 classifier을 바꿔 fine tuning

resnet50(1)을 teacher, DeiT(2)를 student로 하여 DeiT(2)학습

Knowledge distillation을 한 DeiT(2)와 resnet50(1) 두 모델을 teacher로 하고, 새로운 DeiT(3)를 student로 하여 학습

Knowledge distillation을 한 DeiT(3)과 resnet50(1) 두모델을 teacher로 하고, 새로운 DeiT(4)를 student로 하여 학습

⋮

결과 및 분석

- error 값들은 3번 학습 시켰을 때의 평균값이다.
- soft는 soft distillation loss function을 hard는 hard distillation loss function을 사용함을 의미한다.

Model	# params	top-1 error	top-5 error
resnet50 fine tuning	23M	13.2	1.8
DeiT fine tuning	22M	13.7	2.3
DeiT-soft	22M	14.0	1.8
DeiT-hard	22M	12.3	1.7

한 번의 distillation 학습 싸이클이 돌았을 경우 hard knowledge distillation에서 일관된 좋은 성능을 얻었다.

- 모델 명 뒤 형식: distillation loss/사용된 teacher 모델의 base architecture/사용된 teacher 모델의 수

Model	# params	top-1 error	top-5 error
DeiT-soft/CNN/1	22M	14.0	1.8
DeiT-hard/CNN/1	22M	12.3	1.7
DeiT-soft/DeiT/1	22M	12.4	1.6
DeiT-hard/DeiT/1	22M	11.9	1.7
DeiT-soft/DeiT.CNN/2	22M	12.5	1.5
DeiT-hard/DeiT.CNN/2	22M	12.4	1.7
DeiT-soft/CNN.CNN/2	22M	12.7	1.5
DeiT-hard/CNN.CNN/2	22M	12.3	1.7

두 번 이상의 distillation 학습 싸이클이 돌았을 경우에는 일관된 좋은 성능을 얻을 수 없었다.

*** 결과 분석**: 한 번의 distillation 학습 싸이클에서 효과가 있었지만 두 번 이상의 distillation 학습 싸이클부터 효과가 없었던 이유는 DeiT의 distillation token을 늘리지 않고, distillation loss function만을 바꿔서 일 것으로 해석된다. Distillation token의 개수를 늘려 DeiT 모델이 knowledge를 수용할 수 있는 그릇을 늘려 실험해볼 가치가 있다. 하지만 이럴 경우 학습 모델의 parameter 증가로 GPU 리소스에 efficient하지 않을 수 있으므로 한 번의 distillation 학습 싸이클이 효율적인 모델 성능향상에 있어서 가장 효과적인 방법일 것으로 판단된다.