Regularizing CNN Transfer Learning With Randomised Regression (2020)

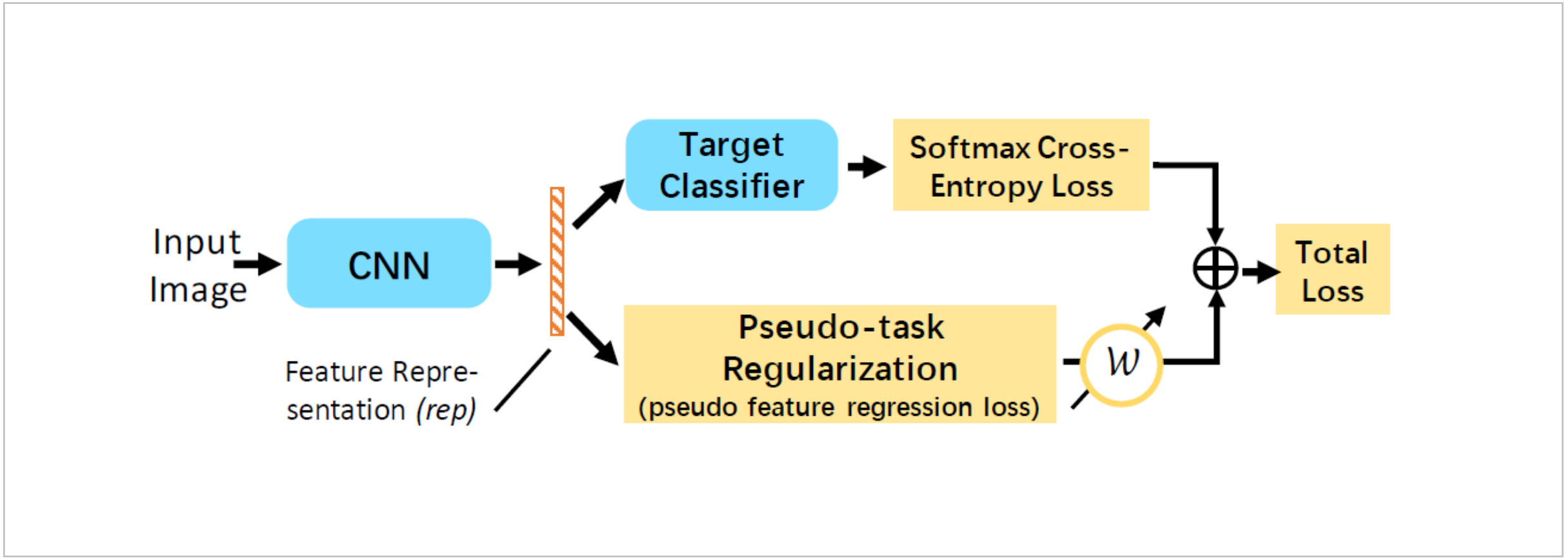
Zhong, Y., & Maki, A. KIn Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13637-13646).

IIE8557-01 동적계획법과 강화학습

경영과학연구실 이태헌

Pseudo-task regularization

- Pseudo-task regularization은 모델을 학습시킬 때 주로 사용되는 기술로서, 모델이 주요 작업에 대해 더 잘 일반화될 수 있도록 도와 줌
- 주된 작업(target task) 외에도 부가적인 가짜 작업(pseudo-task)을 도입하여 네트워크가 다양한 특징을 학습할 수 있게 하는 방식이 pseudo-task regularization
- 모델이 목표 작업에 과적합 되지 않도록 방해하며, 이를 통해 모델이 보다 일반적이고 다양한 표현을 학습할 수 있게 도와 일반화 성능 향상



효율적 정규화 향상 방법 필요성

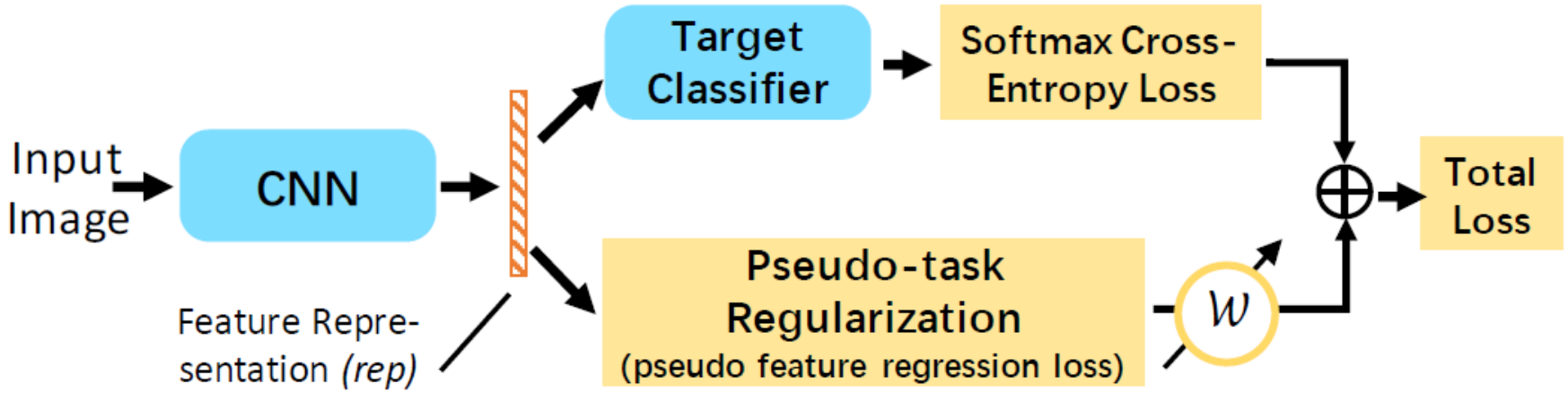
- Fine-tuning은 target task에 대해 전체 네트워크를 최적화하는 것을 목표로 함. 이는 제한된 양의 도메인 데이터를 가진 CNN transfer learning에서 많이 사용됨
- Fine-tuning 초기에는 Source model의 deep network가 small-scale target task에 대해 과도하게 매개변수화 되어 있기 때문에 과적합 방지 위해 Source model을 조정할 필요가 있음
- Fine-tuning 주요 challenge 중 하나는 제한된 training sample로 과도하게 매개변수화 된 모델에 대한 네트워크 정규화를 달성하는 것
- 정규화 향상은 using a complex network architecture during training, recalling the source model 등 resource-dependent cost가 발생함
- Fine-tuning 중 효율적 정규화 향상 방법 필요

‘Pseudo-task Regularization 방법으로 Randomised Regression을 사용하여 CNN Transfer Learning 정규화 성능을 높이고자 함’

- **Randomised Regression을 Pseudo-task Regularization 방법으로 활용**

  - pseudo-task를 통해 정규화에 유용한 gradient가 생성됨
  - target task를 방해함으로써 모델의 일반화 성능 향상 시킴
  - 간단한 방법으로 CNN Transfer learning 정규화 성능을 올리고자 함

An overview of the proposed Pseudo-task Regularization(PtR)



- 두 개의 Training 목적을 활용하는 multi-task framework
- Target task 분류를 위한 Cross-Entropy Loss, PtR task 통한 pseudo feature regression loss 사용

Pseudo-task Regularization

Algorithm 1: Training with Pseudo-task Regularization

Source: a) Off-the-shelf net; b) Labeled data in target domain

Procedure:

for *iteration (batch)* i **do**

Compute cross-entropy loss $L_{ce}^{(i)}$.

if $L_{ce}^{(i)}$ *far from minimum* **then**

Back propagate $L_{ce}^{(i)}$ only;

else

First, perform the following calculations:

1. $L_{PtR}^{(i)}$: the pseudo-regression task loss, $L_{PtR}^{(i)} = f_{reg}(rep^{(i)}, t^{(i)})$ w.r.t. the regression target $t^{(i)}$ generated on-line;
2. $G_{ce}^{(i)}$ and $G_{PtR}^{(i)}$: the gradient norms of $L_{ce}^{(i)}$ and $L_{PtR}^{(i)}$ w.r.t. $rep^{(i)}$. $rep^{(i)}$ stands for the image representations of the batch;
3. $\overline{G}_{ce}^{(i)}$ and $\overline{G}_{PtR}^{(i)}$: the average of $G_{ce}^{(i)}$ and $G_{PtR}^{(i)}$ over the batch;
4. Weight w: $w = \frac{\overline{G}_{ce}^{(i)}}{\overline{G}_{PtR}^{(i)} \cdot R}$ for a target ratio R.

Then, back propagate $L_{ce}^{(i)} + w \cdot L_{PtR}^{(i)}$.

end

end

1. 시작 조건

a) Training 된 네트워크, b) target domain의 레이블 데이터

2. 각 반복(배치)에 대해

1) cross-entropy loss $L_{ce}^{(i)}$ 계산

2) 만약, $L_{ce}^{(i)}$가 최소값에서 멀리 떨어져 있다면 cross-entropy loss만으로 Back propagate

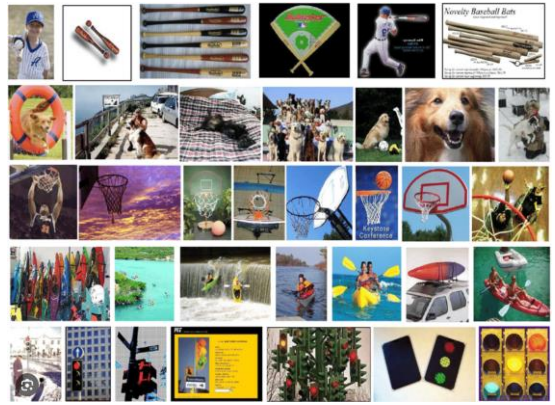
3) 그렇지 않다면, 아래 계산 수행

- $L_{PtR}^{(i)}$: 알고리즘 실행 중 생성된 회귀 대상 t^i 에 대한 pseudo-regression task loss 계산
- $G_{ce}^{(i)}$, $G_{PtR}^{(i)}$: 배치의 이미지 표현 rep^i 에 대한 $L_{ce}^{(i)}, L_{PtR}^{(i)}$ 에 대한 gradient norm 계산
- 가중치를 동적으로 조절하여 $L_{ce}^{(i)} + w * L_{PtR}^{(i)}$ 으로 Back propagate

Datasets

- Flower102, Cube200-2011, MIT67, Caltech256 dataset 사용

- 사용된 3개의 datasets (Flower102, Cube200-2011, MIT67)은 각 클래스별 샘플이 상대적으로 적음. Caltech256 dataset은 각 클래스별 샘플이 상대적으로 많음
- 클래스 별 데이터가 적은 상황에서는 모델 과적합 가능성이 커지므로, 정규화 방법에 따른 차이 확인
- 또한 클래스 별 데이터가 많은 상황에서도 제안한 정규화 방법의 효과성 검증



Pseudo-task Regularization (PtR), vanilla fine-tuning 모델 분류 성능 비교

- VGG-16을 Baseline 모델로 사용함
- Regularization Gain : Pseudo-task Regularization (PtR) 방법을 사용할 때, 두 가지 회귀 함수 SML1과 L2를 사용한 정규화 기법이 얼마나 성능 향상을 가져왔는지를 나타냄
- Error Rate Reduction : SML1, L2 두 정규화 기법을 사용함으로써 얻어진 오류율의 감소를 나타냄

	Baseline	Regularization Gain		Error Rate Reduction	
		SML1	L2	SML1	L2
Flower102	83.92% (0.36)	2.38% (0.32)	2.61% (0.42)	14.80%	16.23%
CUB200	75.07% (0.26)	3.05% (0.39)	2.84% (0.37)	12.23%	11.39%
MIT67	71.55% (0.38)	1.42% (0.58)	1.39% (0.40)	4.99%	4.89%
Stanford40	76.99% (0.19)	2.50% (0.09)	2.21% (0.16)	10.86%	9.60%
WebFace500	77.54% (0.52)	0.95% (0.56)	0.83% (0.47)	4.23%	3.70%

SML1 (Smooth L1 Loss) : L1 Loss와 L2 Loss의 조합으로, 이 loss function은 예측 오차가 작을 때는 L2 Loss처럼 동작하고, 오차가 클 때는 L1 Loss처럼 동작함

다른 일반화 기법들과 분류 성능 비교 (CUB200, Flower102 dataset)

- JointTrain: 여러 작업 동시 학습 전략
- Learning without Forgetting (LwF) : 기존 정보를 유지하면서 새로운 작업을 학습
- Pair-wise Confusion (PC): 다양한 작업이나 도메인 간의 특징을 혼동시키는 전략
- Feature Norm Penalty (FNP) : feature norm에 penalty를 부여하여 복잡도 제한, 과적합 방지함

CUB200 dataset 결과 비교

Method	Baseline	Acc.	Gain
JointTrain (VGG-16)	72.1	74.6	2.5
LwF (VGG-16)	72.1	72.3	0.2
PC(VGG-16)	73.3	76.5	**3.2**
PtR (VGG-16)	75.1	78.1	3.0
PC (ResNet-50)	78.2	80.3	**2.1**
FNP (ResNet-50)	80.3	80.6	0.3
PtR (ResNet-50)	80.3	81.9	1.6
PtR (ResNet-50, w/o WD)	81.0	82.0	1.0

Flower102 dataset 결과 비교

Method	Baseline	Acc.	Gain
JointTrain (VGG-16)	72.1	74.6	2.5
LwF (VGG-16)	72.1	72.3	0.2
PC(VGG-16)	73.3	76.5	**3.2**
PtR (VGG-16)	75.1	78.1	3.0
PC (ResNet-50)	78.2	80.3	**2.1**
FNP (ResNet-50)	80.3	80.6	0.3
PtR (ResNet-50)	80.3	81.9	1.6
PtR (ResNet-50, w/o WD)	81.0	82.0	1.0

다른 일반화 기법들과 분류 성능 비교 (MIT67, Caltech256 dataset)

- Borrowing Treasures from the Wealthy (BTfW) : 큰 데이터셋의 정보를 작은 데이터셋에 이용
- Inductive Bias (Ind.Bias) : 모델에 일반화를 도와주는 가정이나 제약 조건을 추가

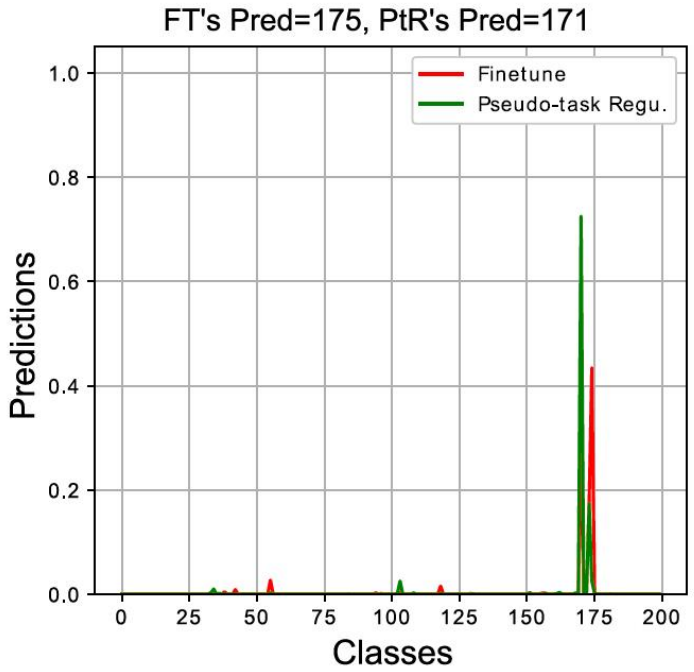
MIT67 dataset 결과 비교

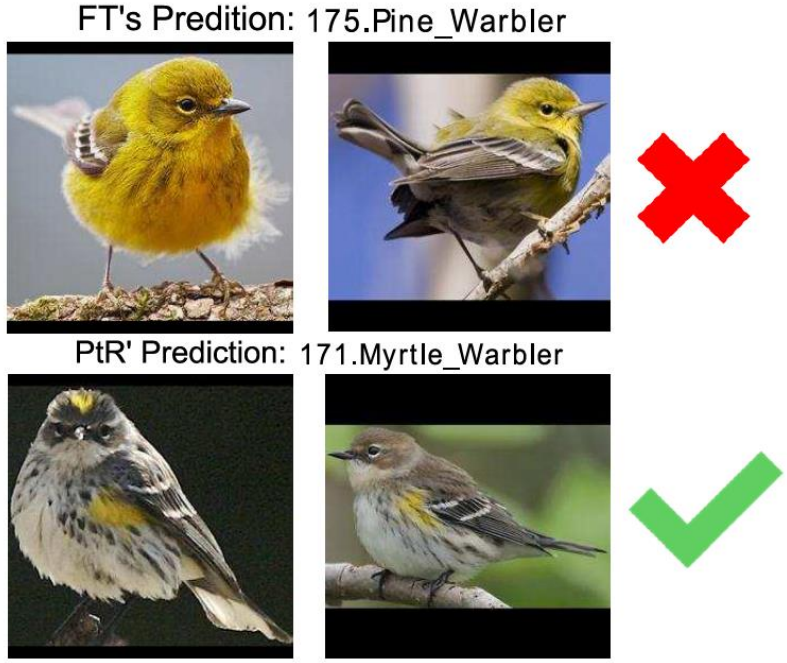
Method	Baseline	Acc.	Gain
JointTrain (VGG-16)	74	75.5	**1.5**
LwF (VGG-16)	74	74.7	0.7
PtR (VGG-16)	71.6	73.0	1.4
BTfW (ResNet-152)	81.7	82.8	**1.1**
Ind.Bias (ResNet-101)	77.5	78	0.5
FNP (ResNet-50)	77.4	78.0	0.6
PtR (ResNet-50)	77.4	77.9	0.5
PtR (ResNet-101)	78.7(78.7)	79.2(79.2)	0.5(0.5)

Caltech256 dataset 결과 비교

	Caltech256-30			Caltech256-60		
Method	Bsln.	Acc.	Gain	Bsln.	Acc.	Gain
BTfW	81.2	83.8	**2.6**	86.4	89.1	**2.7**
Ind.Bias	81.5	83.5	2.0	85.3	86.4	1.1
FNP	84.0	83.8	-0.2	86.8	86.9	0.1
PtR	84.0	84.5	0.5	86.8	87.2	0.4
PtR,w/o WD	84.0	84.5	0.5	86.9	87.2	0.3

Sample from the validation set of CUB200 that PtR correctly rectified mis-classification caused in the vanilla fine-tuning





Conclusions

- Pseudo-task Regularization (PtR) 통해 제한된 데이터 샘플로 전송 학습의 정규화를 향상시키기 위해 pseudo-regression 작업에 의한 유용한 간섭을 생성하는 multi-task learning framework 활용
- PtR로부터 정규화 효과는 target task와 pseudo-regression gradient norm을 기반으로 정규화 강도를 동적으로 조절. 간단한 PtR 방법으로 robust한 성능 향상을 보임

Q & A