[week1] End-to-End Multi-Task Learning with Attention

End-to-End Multi-Task Learning with Attention

Abstract

💡 task-specific feature level attention을 학습할 수 있는 멀티 task learning 아키텍처
e2e 훈련 가능
어떤 feed-forward 신경망에 대해서도 간단하게 구현 가능
멀티태스크 학습 부분에서 SOTA 달성
multi task loss 함수의 가중치 스키마에 대해 덜 민감함

Introduction

일반적인 single task learning에 비해 multi task learning이 필요로하는 2가지

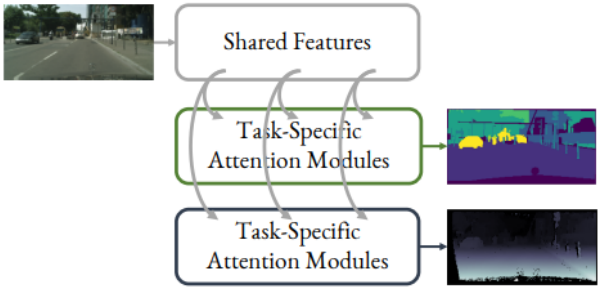

Figure 1: Overview of our proposal MTAN. The shared network takes input data and learns task-shared features, whilst each attention network learns task-specific features, by applying attention modules to the shared network.

1) 네트워크 아키텍처 (어떻게 공유?)

: MTL은 task-shared와 task-specific에 대해 모두 표현 가능해야한다.

→ 일반적인 representation을 학습해야한다 (over fitting 방지)

→ 각 task에 대한 feature들을 학습해야한다 (under fitting 방지)

2) Loss Function (task 별 균형 유지?)

: MTL의 손실함수는 모든 task가 동등하게 학습될 수 있도록 해야함

가중치를 자동으로 학습하거나 다른 가중치에 대해 robust한 네트워크 필요

⇒ 기존의 멀티태스크 learning은 두 가지 중 하나를 달성하는 것에 집중했음

그러나 본 모델은 두가지 챌린지를 모두 달성할 수 있도록 함

[1] 공유된 task와 특정 task 모두 자동적으로 학습 가능하도록

[2] loss weighting scheme의 선택에 대해 robust 하도록

Multi-task Attention Network

MTAN의 아키텍처는 어떤 feed-forward network라도 포함 가능해야한다

- **Architecture Design**

  2가지 component로 구성됨

  → single shared 네트워크 : 특정 task를 기반으로 설계 가능

→ K task specific attention 네트워크 : attention 모듈들로 구성 (shared network에 link)

⇒ 각 attention 모듈은 task-specific feature를 학습하기 위해 shared network의 특정 레이어에 soft attention mask를 적용한다.

(attention mask는 shared network에서 feature selector 역할을 함 → e2e에서 자동적으로 학습)

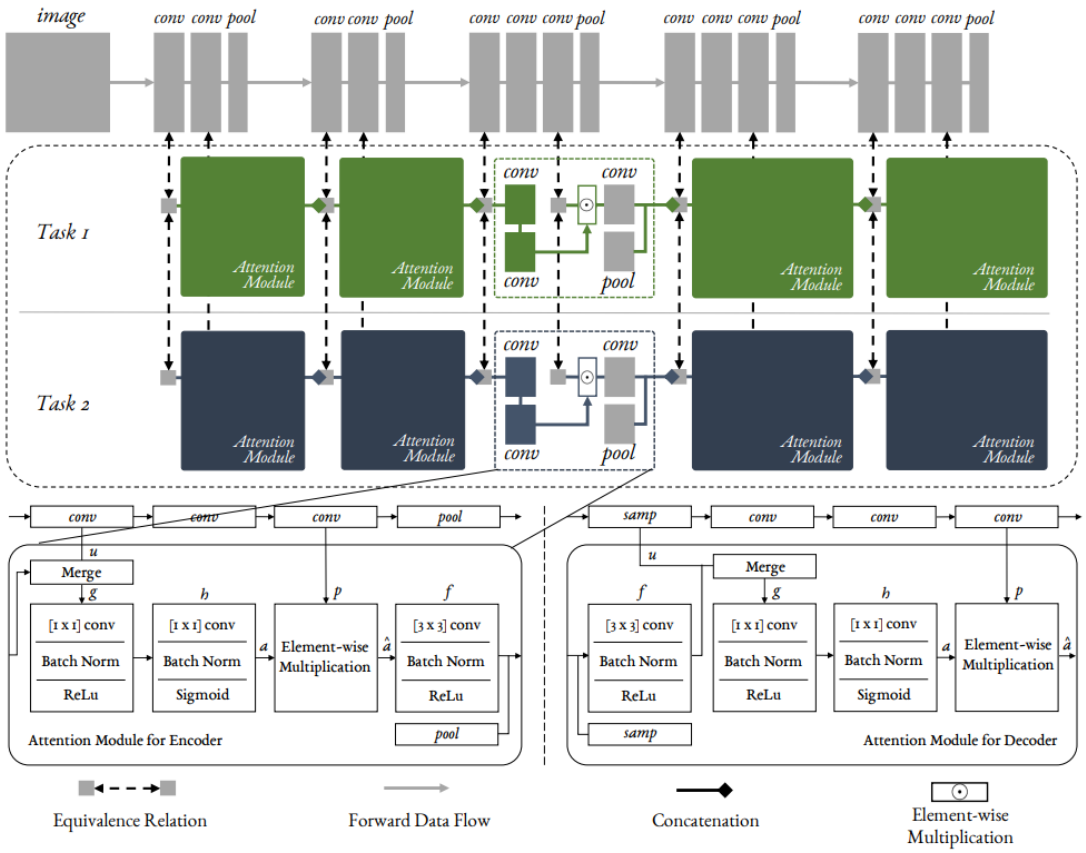

Figure 2: Visualisation of MTAN based on VGG-16, showing the encoder half of SegNet (with the decoder half being symmetrical to the encoder). Task one (green) and task two (blue) have their own set of attention modules, which link with the shared network (grey). The middle attention module has its structure exposed for visualisation, which is further expanded in the bottom section of the figure, showing both the encoder and decoder versions of the module. All attention modules have the same design, although their weights are individually learned.

→ 각각의 attention 모듈은 soft attention mask를 학습 (soft attention mask는 shared network의 해당하는 레이어의 feature에 의존)

⇒ 공유 네트워크와 soft attention mask는 multiple task에 거쳐 일반적인 공유 feature 학습이 최대화 될 수 있도록 함

- **Task Specific Attention Module**

  attention 모듈은 task-related feature들을 학습할 수 있도록 설계

  (공유 네트워크 feature에 soft attention을 적용한 것)

$$\hat{a}_i^{(j)} = a_i^{(j)} \odot p^{(j)} ,$$

** p(j) : 공유 네트워크의 j번째 블록

** a_i(j) : task i의 layer를 위한 attention mask

** a_i(j)_hat : task-specific feature

$$a_i^{(j)} = h_i^{(j)} \left(g_i^{(j)} \left(\left[u^{(j)}; f^{(j)} \left(\hat{a}_i^{(j-1)} \right) \right] \right) \right), j \geq 2 \quad (2)$$

(f(j), g_i(j), h_i(j)는 batchnormalization을 적용한 convolution layer)

→ g_i(j), h_i(j)는 [1x1] 커널로 구성 (i번째 특정 task의 attention mask 역할)

→ f(j)는 [3x3] 커널 (다른 attention 모듈로 전달하기 위한 공유 feature 추출 역할)

attention mask는 sigmoid activation에 태워 [0,1] 범위를 갖도록 함

→ 이후 self-supervised 방식으로 학습

- **The Model Objective**

$$\mathcal{L}_{tot}(\mathbf{X}, \mathbf{Y}_{1:K}) = \sum_{i=1}^{K} \lambda_i \mathcal{L}_i(\mathbf{X}, \mathbf{Y}_i).$$

일반적인 multi task (K task)의 경우 loss function

X: input / Y_i: task specific labels

(가중치 람다를 가지는 loss L들의 선형 결합으로 구성)

Experiments

Image-to-Image 예측 task를 위한 SegNet에 대해 MTAN 설계 → 성능 평가

1) one-to-many

2) many-to-many

- **Dynamic weight average**

  대부분의 multi task 는 여러 task 간의 균형을 찾지 못하면 여러 task를 훈련하기 어려움

  → GradNorm에 영향을 받은 DWA 방법론 차용

$$\lambda_k(t) := \frac{K \exp(w_k(t-1)/T)}{\sum_i \exp(w_i(t-1)/T)}, w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)}, \tag{7}$$

  (각 task에 대한 loss 변화율을 고려하여 시간에 따른 task 별 가중치의 평균을 학습)

  (GradNorm과 달리 수치적인 task loss만 고려하면 되어서 간단)

- **Result**

#P.	Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err
2	One Task	n.a.	51.09	90.69	0.0158	34.17
3.04	STAN	n.a.	51.90	90.87	0.0145	27.46
1.75	Split, Wide	Equal Weights	50.17	90.63	0.0167	44.73
		Uncert. Weights [14]	**51.21**	**90.72**	**0.0158**	44.01
		DWA, $T = 2$	50.39	90.45	0.0164	**43.93**
2	Split, Deep	Equal Weights	**49.85**	88.69	0.0180	43.86
		Uncert. Weights [14]	48.12	88.68	**0.0169**	**39.73**
		DWA, $T = 2$	49.67	**88.81**	0.0182	46.63
3.63	Dense	Equal Weights	**51.91**	90.89	0.0138	27.21
		Uncert. Weights [14]	51.89	**91.22**	**0.0134**	**25.36**
		DWA, $T = 2$	51.78	90.88	0.0137	26.67
≈2	Cross-Stitch [20]	Equal Weights	50.08	90.33	0.0154	34.49
		Uncert. Weights [14]	50.31	90.43	**0.0152**	**31.36**
		DWA, $T = 2$	**50.33**	**90.55**	0.0153	33.37
1.65	MTAN (Ours)	Equal Weights	53.04	**91.11**	**0.0144**	**33.63**
		Uncert. Weights [14]	**53.86**	91.10	0.0144	35.72
		DWA, $T = 2$	53.29	91.09	0.0144	34.14

  → single shared feature를 자동적으로 attention mask를 통해 학습하면서 추가적인 파라미터 없이 좋은 성능을 보임 (더 적은 파라미터 수)

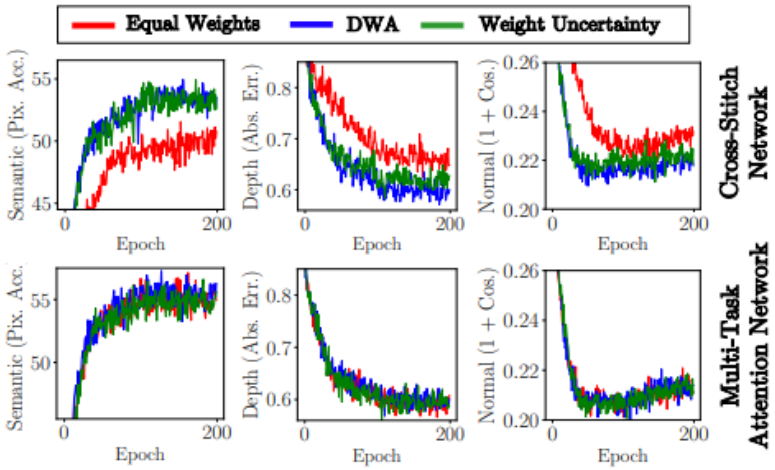

Figure 3: Validation performance curves on the NYUv2 dataset, across all three tasks (semantics, depth, normals, from left to right), showing robustness to loss function weighting schemes on the Cross-Stitch Network [20] (top) and our Multi-task Attention Network (bottom).

→ 다양한 loss function 가중치 스키마에 걸쳐 높은 성능 유지 (다른 방법론보다 가중치 스키마에 대해 robust) ← learning curve 그래프를 통해 확인 가능

- **Effect of Task Complexitiy**

Type	#P.	Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err	Surface Normal Angle Distance (Lower Better) Mean	Surface Normal Angle Distance (Lower Better) Median	Surface Normal Within t° (Higher Better) 11.25	Surface Normal Within t° (Higher Better) 22.5	Surface Normal Within t° (Higher Better) 30
Single Task	3	One Task	n.a.	15.10	51.54	0.7508	0.3266	31.76	25.51	22.12	45.33	57.13
	4.56	STAN	n.a.	15.73	52.89	0.6935	0.2891	32.09	26.32	21.49	44.38	56.51
Multi Task	1.75	Split, Wide	Equal Weights	15.89	51.19	0.6494	0.2804	33.69	28.91	18.54	39.91	52.02
			Uncert. Weights [14]	15.86	51.12	**0.6040**	0.2570	**32.33**	**26.62**	**21.68**	**43.59**	**55.36**
			DWA, $T = 2$	**16.92**	**53.72**	0.6125	**0.2546**	32.34	27.10	20.69	42.73	54.74
	2	Split, Deep	Equal Weights	13.03	41.47	0.7836	0.3326	38.28	36.55	9.50	27.11	39.63
			Uncert. Weights [14]	**14.53**	43.69	0.7705	0.3340	**35.14**	**32.13**	**14.69**	**34.52**	**46.94**
			DWA, $T = 2$	13.63	**44.41**	**0.7581**	**0.3227**	36.41	34.12	12.82	31.12	43.48
	4.95	Dense	Equal Weights	16.06	52.73	0.6488	0.2871	33.58	28.01	20.07	41.50	53.35
			Uncert. Weights [14]	**16.48**	**54.40**	0.6282	0.2761	**31.68**	**25.68**	**21.73**	**44.58**	**56.65**
			DWA, $T = 2$	16.15	54.35	**0.6059**	**0.2593**	32.44	27.40	20.53	42.76	54.27
	≈3	Cross-Stitch [20]	Equal Weights	14.71	50.23	0.6481	0.2871	33.56	28.58	20.08	40.54	51.97
			Uncert. Weights [14]	15.69	52.60	0.6277	0.2702	32.69	27.26	21.63	42.84	54.45
			DWA, $T = 2$	**16.11**	**53.19**	**0.5922**	**0.2611**	**32.34**	**26.91**	**21.81**	**43.14**	**54.92**
	1.77	MTAN (Ours)	Equal Weights	**17.72**	55.32	**0.5906**	0.2577	31.44	**25.37**	**23.17**	45.65	57.48
			Uncert. Weights [14]	17.67	**55.61**	0.5927	0.2592	**31.25**	25.57	22.99	**45.83**	**57.67**
			DWA, $T = 2$	17.15	54.97	0.5956	**0.2569**	31.60	25.46	22.48	44.86	57.24

Table 3: 13-class semantic segmentation, depth estimation, and surface normal prediction results on the NYUv2 validation dataset. #P shows the number of network parameters, and the best performing combination of multi-task architecture and weighting is highlighted in bold. The top validation scores for each task are annotated with boxes.

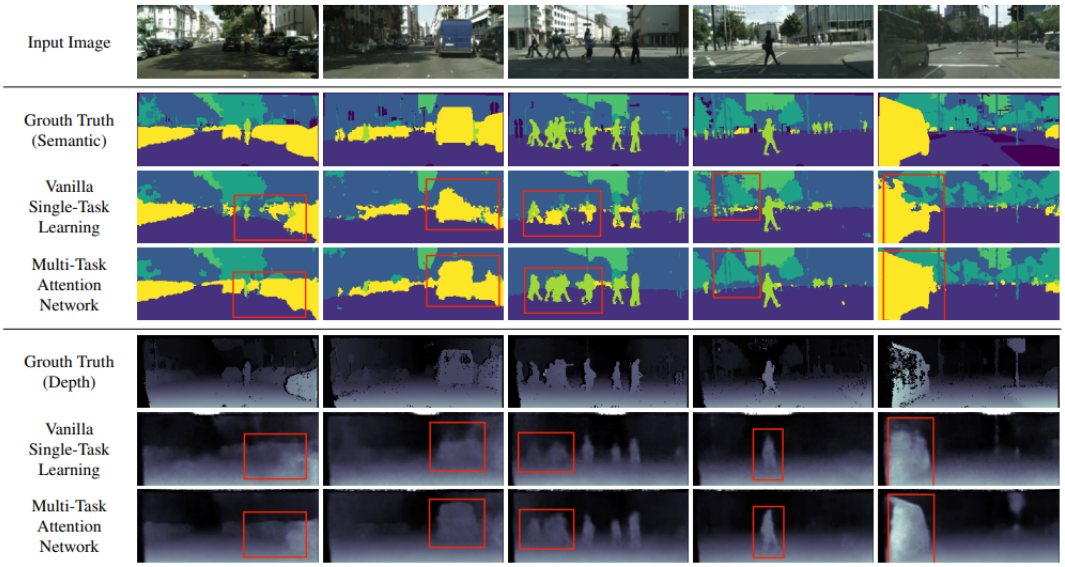

Figure 4: CityScapes validation results on 7-class semantic labelling and depth estimation, trained with equal weighting. The original images are cropped to avoid invalid points for better visualisation. The red boxes are regions of interest, showing the effectiveness of the results provided from our method and single task method.

⇒ MTAN이 simple task에 대해 simple parameter, complex task에 대해 효율적인 개수의 parameter로 좋은 성능을 내는 것을 확인할 수 있다.

- **Attention Masks as feature selectors**

각 mask가 feature selector로 작용

shared feature의 정보가 없는 파트를 masking하고 각 task에 유용한 부분에 초점을 맞추는 기능 수행

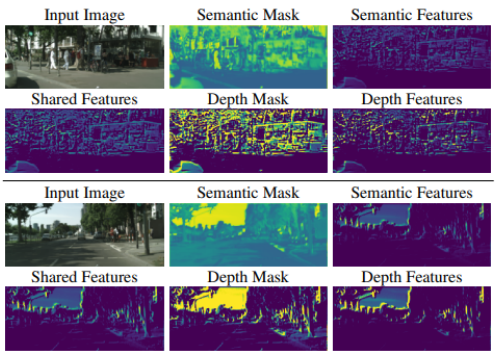

Figure 5: Visualisation of the first layer of 7-class semantic and depth attention features of our proposed network. The colours for each image are rescaled to fit the data.

→ 해당 작업에 이점을 얻을 수 있음

Conclusion

multi task 학습을 위해 Multi-Task Attention Network (MTAN) 제시

→ 각 task에 대한 attention 모듈과 global feature pool로 구성

⇒ task-shared와 task-specific에 대해 자동적으로 end-to-end 방식의 학습이 가능

손실함수에 사용되는 가중치 스키마에 대해 강건함을 가짐

attention mask를 통해 가중치를 공유하면서 해당 방법론은 SOTA를 달성하며 높은 파라미터 효율성을 지니게 됨.