Boosting Share Routing for Multi-task Learning

Xiaokai Chen
Tencent PCG
China
dzhchxk@126.com

Xiaoguang Gu
Tencent PCG
China
ryanxggu@tencent.com

Libo Fu
Tencent PCG
China
derekfu@tencent.com

Introduction

- Multi-task learning (MTL) aims to make full use of the knowledge contained in multi-task labels to improve the overall performance.

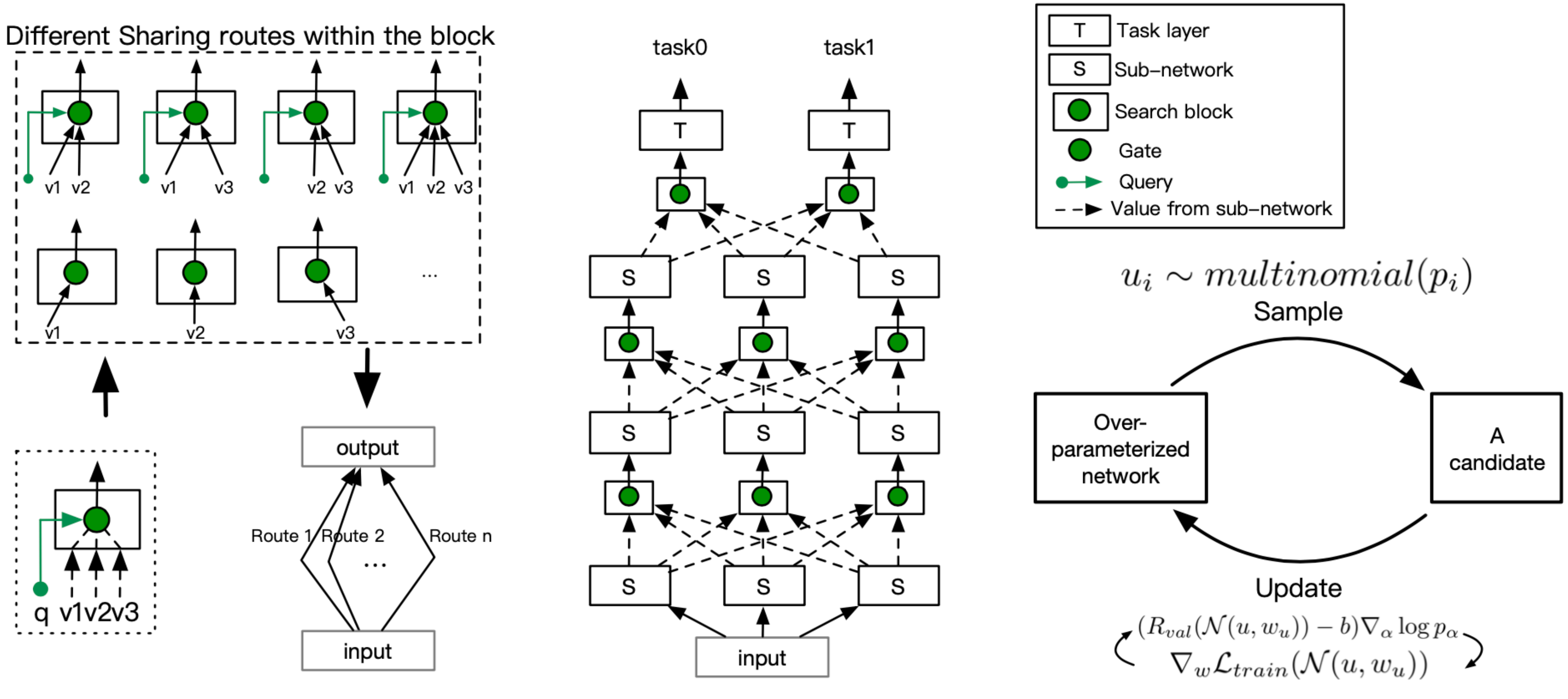
- Compared with learning tasks separately MTL benefits a lot:
  - It not only reduces maintenance cost of online systems but also could achieve better performance.

Introduction

- Negative transfer
  - Suitable sharing mechanism is hard to design as the relationship among tasks is complicated.

Introduction

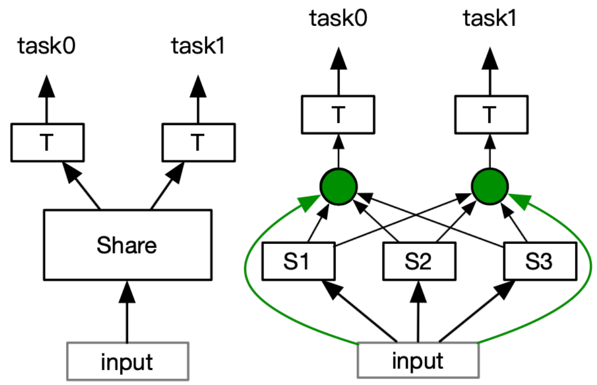
- MTNAS
  - **M**ulti-**T**ask **N**eural **A**rchitecture **S**earch
  - It can efficiently find a suitable sharing route for a given MTL problem.



(a) Local routes in a search block (b) Over–parameterized network (c) Searing process

Related work

- Parameter sharing in Multi-task learning.
  - Two typical approaches:

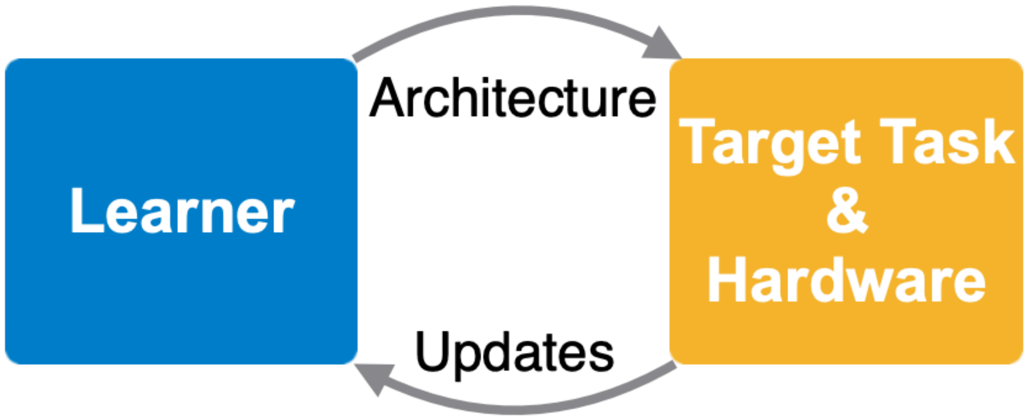


(a) Share-Bottom (b) MMoE

- The limits of MMoE:
  - It forces all experts to contribute to all tasks, which limits the flexibility of the sharing route.
  - Gating, can hardly help learning a sparse connection although theoretically possible.

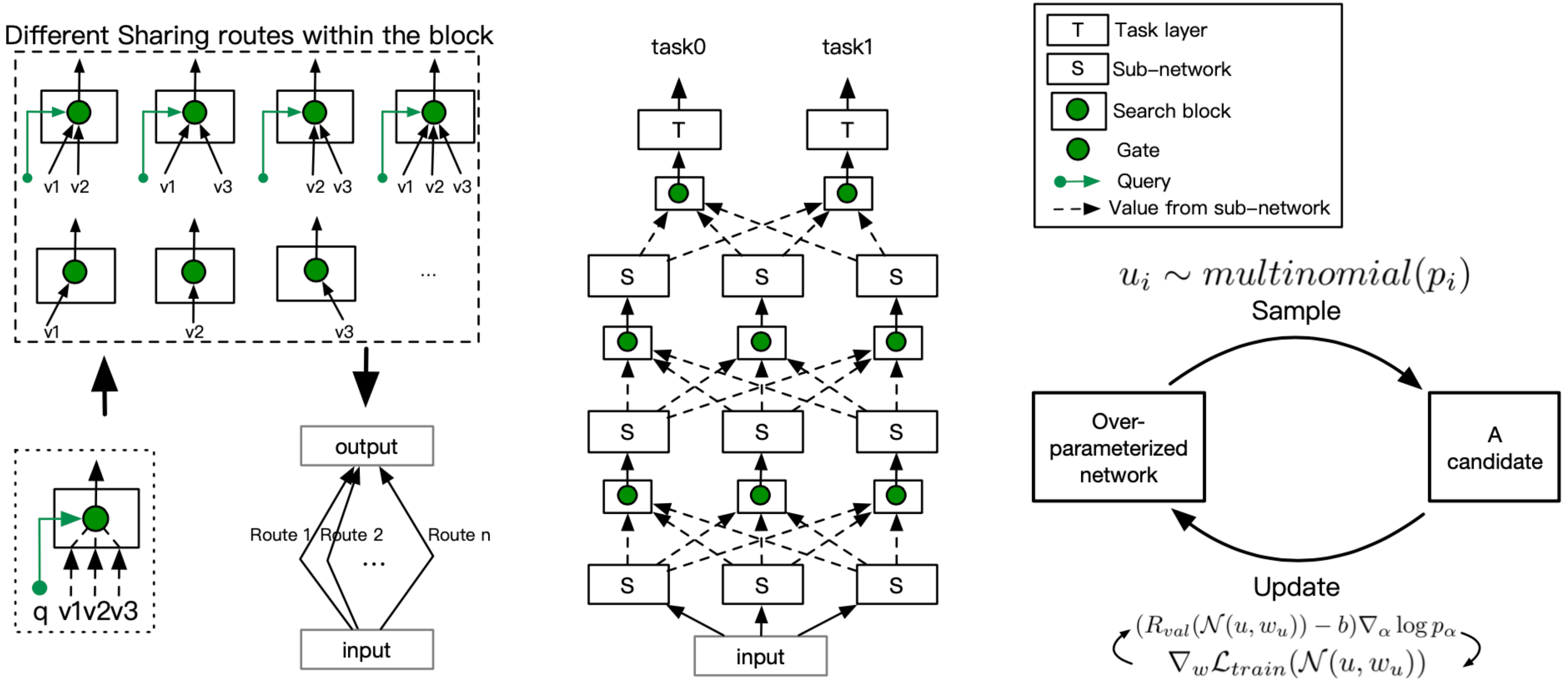
Related work

- Neural architecture search(NAS)
  - Reinforcement learning based methods
  - Evolutionary algorithm based methods
  - Gradient based methods
    - **Proxyless**



Multi-Task Neural Architecture Search

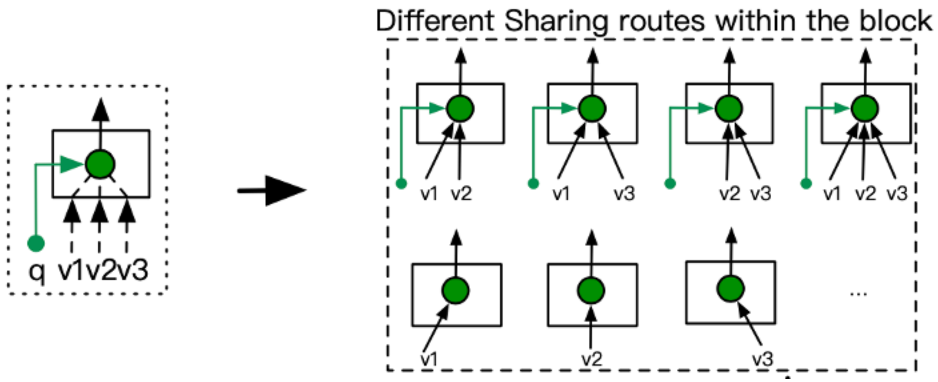
- Framework overview



(a) Local routes in a search block (b) Over-parameterized network (c) Searing process

Multi-Task Neural Architecture Search

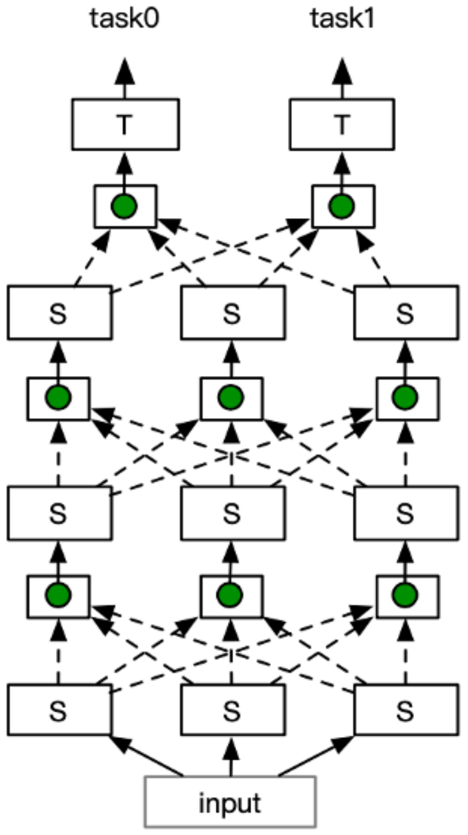
- Framework overview
  - A search block



(a) Local routes in a search block

Multi-Task Neural Architecture Search

- Framework overview
  - The whole search space



(b) Over–parameterized network

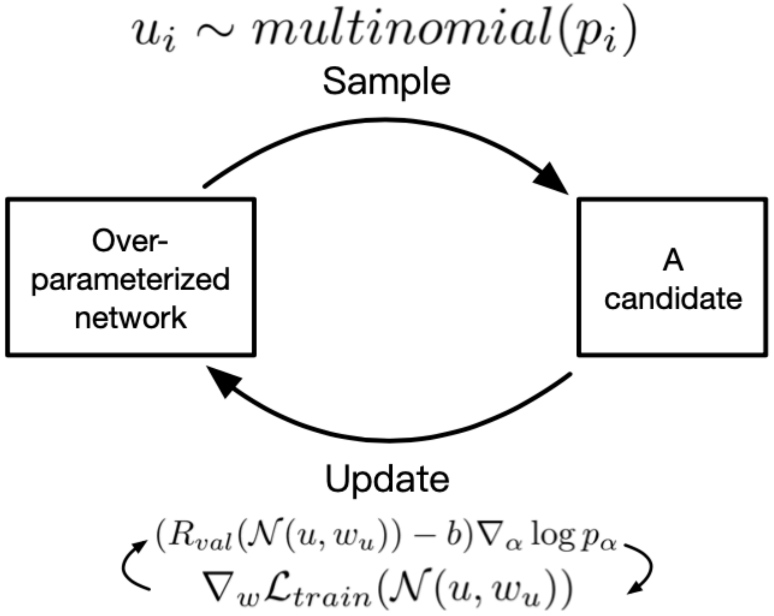
Multi-Task Neural Architecture Search

- Search a promising sharing route
  - Introduce architecture parameters: α_i
    - Sampling probability: $p_i = softmax(\alpha_i)$
  - Utilizing REINFORCE to optimize α_i

$$J(\alpha) = E_{u \sim p(\alpha)} R_{val}(\mathcal{N}(u, w_u))$$

$$\nabla_\alpha J(\alpha) = (R_{val}(\mathcal{N}(u, w_u)) - b)\nabla_\alpha \log p_\alpha$$

  - The architecture parameters and the weights of the network are optimized **alternately**.



(c) Searing process

Experiment

- Compared with single task model as well as typical multi-task approaches

Table 2: Results on BookCrossing and Tiktok (higher AUC is better).

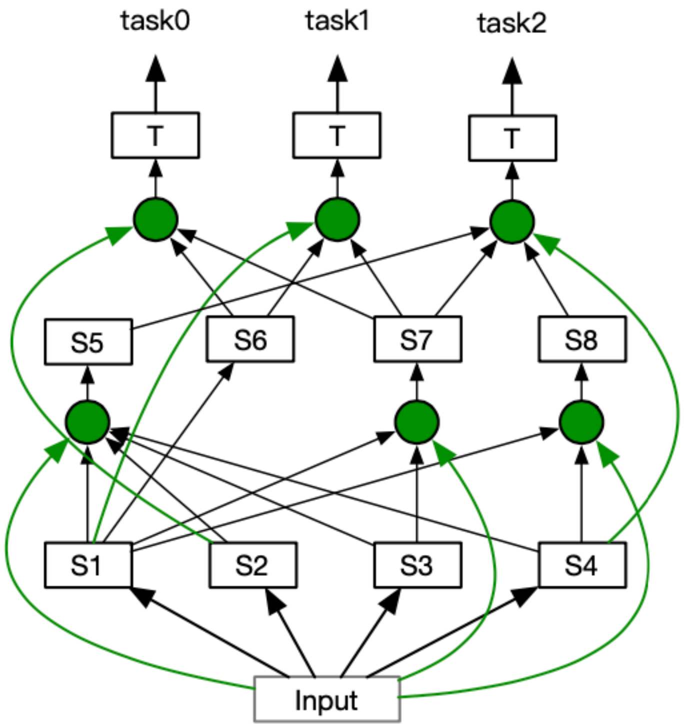
Method	BookCrossing			Tiktok		
	AUC0	AUC1	MTL-Loss	AUC0	AUC1	MTL Loss
Single	0.7842	-	-	0.7485	-	-
	-	0.7984	-	-	0.9428	-
Share-Bottom	0.7834	0.8014	0.7322	0.7478	0.9415	0.6020
MMoE	0.7885	0.8022	0.7302	0.7488	0.9425	0.6006
ML-MMoE	0.7884	0.8051	0.7299	0.7487	0.9421	0.6011
AutoMTL(Ours)	**0.7907**	**0.8086**	**0.7247**	**0.7507**	**0.9467**	**0.5930**

Table 3: Results on GoodRead (higher AUC is better).

Method	AUC0	AUC1	AUC2	MTL Loss
Single	0.8104	-	-	-
	-	0.7752	-	-
	-	-	0.8209	-
Share-Bottom	0.8250	0.7748	0.8415	1.0726
MMOE	0.8255	0.7761	0.8441	1.0716
ML-MMOE	0.8244	0.7743	0.8443	1.0720
AutoMTL(ours)	**0.8281**	**0.7771**	**0.8460**	**1.0656**

Experiment

- The learned sharing route on GoodReads dataset.



(c) Architecture on GoodReads

Table 4: The Pearson correlation (PCC) of tasks.

Dataset	t0&t1	t1&t2	t0&t2
GoodReads	0.493	0.124	0.245

Experiment

- Analysis on synthetic data

Figure 4: Comparison of Share-Bottom, MMoE and MTNAS on synthetic data with two unrelated tasks. The plot show total loss over steps.

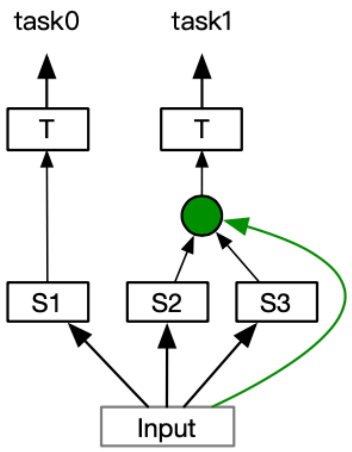


Figure 5: The learned sharing route with L=1,H=3.

Conclusion

- MTNAS
  - can efficiently find a suitable sparse sharing route for MTL
  - consistently outperforms single-task model and typical multi-task approaches on three real-world datasets
  - Experiments on synthetic data further demonstrates that, by allowing sparse connection among shared sub-networks, MTNAS is able to find a sparse route that can effectively alleviate negative transfer when tasks are less related.

Thanks!