All-in-one Deep Neural Network for Autonomous Vehicles

Trong-Lanh Nguyen[12], Guillaume Magniez[1] and Thierry Chateau[2]

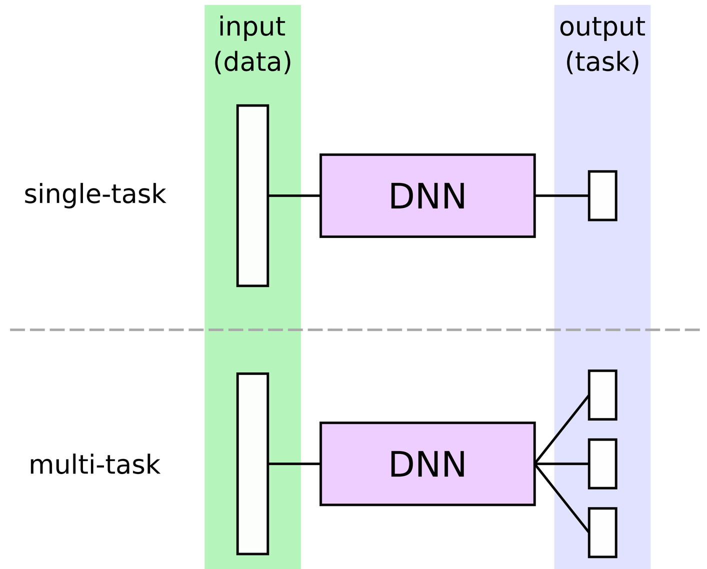
Safran Electronics & Defence [1]

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal [2]

Objectives

1. Develop a **multi-task** Neural Network for autonomous vehicle **visual perception** (monocular video).
2. Devise a way to add or remove tasks on the fly without training the full network all over again.
3. Extend the application to **multi-sensor** input.

Introduction



- Multi-Task Learning (MTL) [1] encompasses the idea that similar computational tasks in deep learning should at one point share some kind of common latent representation, in a more or less pronounced fashion, depending on their relatedness.
- Learning such common representation in a single network can offer several advantages :
  - It can reduce the total size of the networks as well as the computational cost, as a portion of the weights is shared.
  - Various features can be useful to some tasks while being easier to learn by others.
  - The learnt latent representation generalise better.
  - Tasks can be specifically added to incite the network to focus on a wanted data feature.
- In autonomous driving, vision tasks such as obstacle and traffic signs detection and tracking, lane marking detection, scene segmentation, etc., occur in the same space. It is intuitively clear that they are related to some extent.

Problematic

- How to select which tasks to share, and at which point ?
- Which architecture would be able to adapt to additional tasks and enable to train only a part of the network each time ?

Dataset

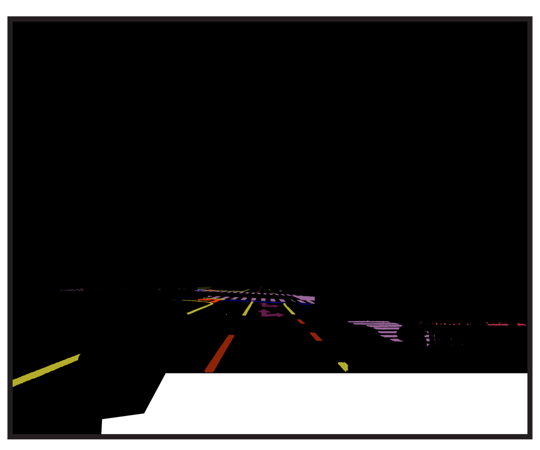
- Trials will mainly be done over Baidu's **ApolloScape** dataset, that was chosen because it exhibits some uncommon and valuable characteristics compared with other existing collections :
  - It is densely annotated, both spatially and temporally
    - Per pixel labels are given for each frames of video sequences
  - labels cover several interesting tasks :
    - Semantic segmentation
    - Instance segmentation
    - Lane marking detection
    - Depth estimation
    - Pose estimation


Color image


Semantic segmentation

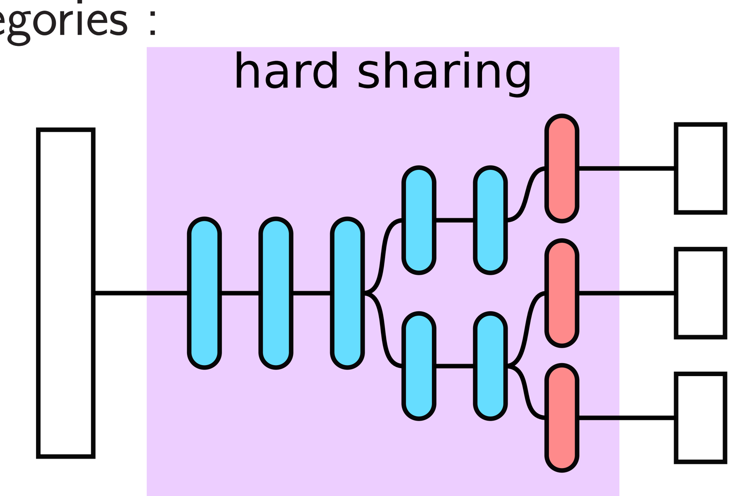

Depth map


Instance segmentation

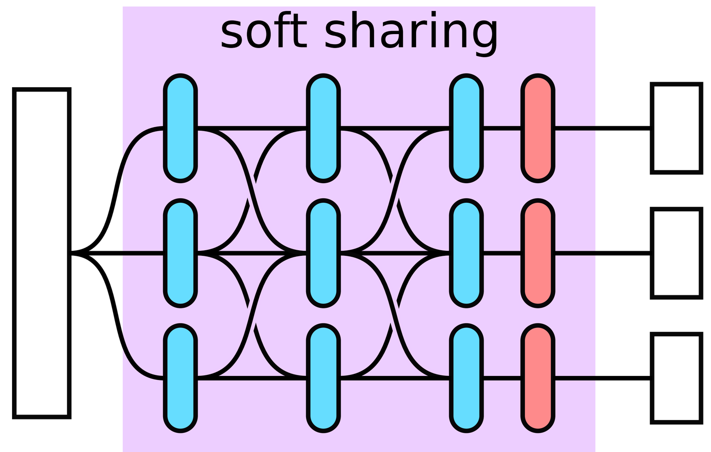

Lane markings

Methods

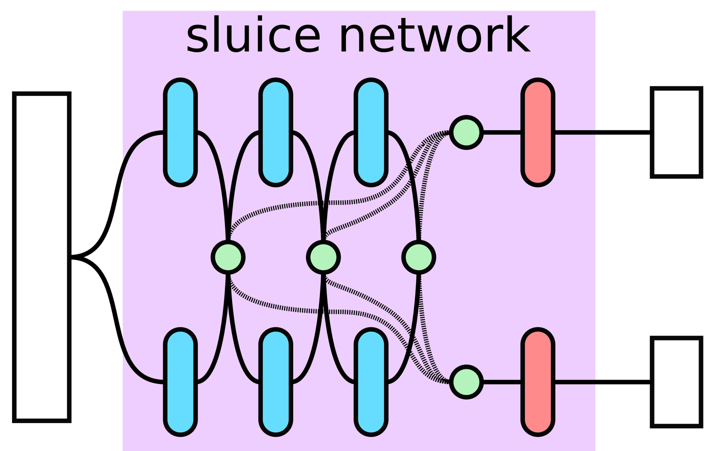
- MTL methods generally fit into two categories :
  - **Hard parameter sharing**
    It is the most straightforward and obvious approach to MTL, starting from a common core and branching progressively to each task. the problem is to find where to split.



  - **Soft parameter sharing**
    In this kind of configuration, each task retain its own specific network, but sharing happens between layers. While loosing the sparing of weights, it is more flexible than hard sharing. The challenge is to define how the sharing is made.



- Traditionally, tasks are shared following an hand-crafted scheme, chosen *ad hoc* for a peculiar task set. Thus, some work has been done toward the automation and optimisation of multi-task network architectures. The difficulty lies primarily in the very large space of structures to explore.
  - **Sluice networks** [2] are an attempt at unifying previous approaches with a meta-architecture, adding linear combinations with learnable coefficients of every task at each layer in a soft sharing structure, and showing how other methods can emerge depending on the learnt coefficients, including hard parameter sharing.



  - **Task taxonomy** methods [3] use a computational metric to assess tasks affinity and cluster them in a hard parameter sharing fashion.
  - **Soft layer ordering** [4] tries to explore the possibilities that arise when breaking the assumption that all tasks must use the shared layers in the same order, by allowing each task to learn and use its own sequence.

Conclusion

- One of the thesis' objectives being to have a modular network, hard parameter sharing approaches are quite impractical, since there is very little space for learning when adding a new task. On the other hand, soft sharing techniques linearly grow in size with the number of tasks and can become an hassle when embedding the network in a mobile platform.
- The state of the art seems devoid of modular approaches to multi-task neural networks, Therefore it could be a future contribution if it is achieved.
- Extending the method to use multiple sources of information as input (multi-sensor) will be the last goal.

References

[1] R. Caruana. "Multitask Learning". In: *Machine Learning* 28 (July 1997). DOI: 10.1023/A:1007379606734.

[2] S. Ruder et al. "Latent Multi-task Architecture Learning". In: *arXiv:1705.08142 [cs, stat]* (Nov. 2018).

[3] S. Vandenhende et al. "Branched Multi-Task Networks: Deciding What Layers To Share". In: *arXiv:1904.02920 [cs]* (Apr. 2019).

[4] E. Meyerson and R. Miikkulainen. "Beyond Shared Hierarchies: Deep Multitask Learning through Soft Layer Ordering". In: *arXiv:1711.00108 [cs, stat]* (Feb. 2018).

Contact

- trong-lanh.nguyen@safrangroup.com
- t-lanh.nguyen@etu.uca.fr

Journée Scientifique de l'École Doctorale des Sciences Pour l'Ingénieur 16 mai 2019