Review #15A

==========

Overall merit

---

2. Weak reject

Reviewer expertise

---

2. Some familiarity

Is this paper exciting and thought-provoking?

---

2. Low risk idea or predictable hypothesis (but quite well executed)

Paper summary

The authors present Antler, an approach that calculates task affinity, generates compact task graphs, and optimizes the order of executing the tasks for multi-task learning in low-resource systems.

Strengths

S1. The authors consider the similarities and dependencies of tasks. Although a multi-task neural architecture with a shared block and separated blocks for downstream tasks is common, the optimization of architectures for lower energy consumption and higher accuracy is novel and interesting.

S2. The authors prove that ordering tasks in multi-task learning is NP-complete and then provide a solution to it, which might provide some useful insights to other researchers in the community.

S3. Analysis and discussion on both public datasets and real-world deployment are shown.

S4. The paper is well-written.

Weaknesses

---

1. [A1] It is not clear why the authors get the representations as Step 1 in Task Affinity shows. Why do they use Pearson correlation coefficient in Step 1 and Spearman's correlation coefficient in Step 2? Justifications are needed.

   [Response] An explanation is provided in Section 3.1 Step 2 "The reason for using Spearman correlation is to capture the nonlinear relationship between data representations."

2. [A2] The authors calculate task affinity with outputs of networks. It would be interesting to know if the architectures of networks make a difference.

   [Response] A detailed discussion is presented in Section 5.2 "Effect of Network Structure."

3. [A3] In figure 2, it is hard to understand why tasks with high variety share more blocks than tasks with low variety.

   [Response] Detailed explanation is provided in the caption of Figure 2.

4. [A4] Although Section 6 is based on public datasets, it is still necessary to show the tasks performed and the final task graphs as shown in 7.2 and 7.3

   [Response] As shown in Figure 16, three different final task graphs are provided.

5. [A5] The complexity of the genetic algorithm solver is unknown. How to ensure the solver will converge in a low resource setting? There's no details on this.

[Response] The algorithm complexities of the two different solvers are provided in Section 4.5. These algorithms are not run on a low-resource system. We clarified that they run on a high-end machine during the offline stage.

Involves human subject research

---

2. Research involving human and animal subjects, and ethical review/approval is mentioned in the paper

* *

Review #15B

==========

Overall merit

---

2. Weak reject

Reviewer expertise

---

3. Knowledgeable

Is this paper exciting and thought-provoking?

---

2. Low risk idea or predictable hypothesis (but quite well executed)

Paper summary

This paper presented Antler, a multitask inference framework that exploits the affinity between all pairs of tasks in a multitask inference system to construct a compact graph representation of the task set and finds an optimal order of execution of the tasks such that the end-to-end time and energy cost of inference is reduced while the accuracy remains similar to the state-of-the-art. To evaluate the performance of Antler, the authors conducted experiments on nine datasets across a set of neural network architectures and demonstrated the proposed Antler outperforms state of the art baselines.

Strengths

+ The work focuses on a very important but challenging problem: multitask inference on embedded systems.

+ The affinity analysis of multiple tasks is quite interesting. + The implementation of system prototypes is impressive.

Weaknesses

- [B1] My primary concern is how multitask learning is conducted. In multitask learning literature, different tasks usually share the chunk of the neural networks but will have different heads for different tasks. Such a single multitask learning network is trained once using the labels of different tasks. According to Figure 1, I am not sure if the proposed Antler follows the multitask learning framework. If this is true, I am concerned about the training quality of the multitask model.

[Response] A clarification is provided in the caption of Figure 1 (b) “The task graph is restrained following standard multitask learning training practices.” Section 3.3 Step 4 of the algorithm also mentions this.

- [B2] I am also concerned about how the multitask formulation was conducted in section 7.1 and 7.2. It seems to me that those tasks are not quite correlated/similar and it does not make too much sense to combine them under the multitask framework. This is because if the tasks are not correlated or similar, the combination of those tasks may hurt the accuracy for every single task.

[Response] Detailed explanation is provided in Section 6.3 “Task Decomposition and Grouping” subsection. The various levels of task relations are discussed and presented with experimental results. We clarified that Antler does not require the tasks to be related; if they are related, then we will see them being grouped together and a compact task graph will be formed by Antler. If the tasks are not related, then Antler will put them in separate groups and the task graph will be less compact.

Involves human subject research

---

1. Does not involve human and animal subjects

* *

Review #15C

==========

Overall merit

---

3. Weak accept

Reviewer expertise

---

3. Knowledgeable

Is this paper exciting and thought-provoking?

---

2. Low risk idea or predictable hypothesis (but quite well executed)

Paper summary

Antler proposes to efficiently design neural network architectures for multi-task learning on resource-constrained systems. The main idea is that many tasks have multiple common building blocks, which may be combined in order to reduce the resources required to perform inferences. Finding these overlapping subtasks, however, may be difficult. Antler measures the affinity between task graphs under different model budgets and then optimizes their execution order depending on the cost of switching between any two specific tasks. Individual tasks are moreover trained using multi-task learning on the same architecture, leveraging their similarities. Experiments with a variety of audio and image processing tasks indicate that Antler has consistently lower execution times and energy consumption, while maintaining comparable model accuracies, as baselines.

Strengths

+ Antler exhibits good performance in practice. It’s especially impressive that the model accuracy remains similar to that of baselines, even though Antler uses multi-task learning with the same architecture for all tasks.

+ The idea of exploiting cross-task affinity is interesting. Antler discovers these affinities in a relatively automated way that exploits known methods for multi-task learning.

Weaknesses

- [C1] Antler assumes that all inference tasks are known a priori, but this may not be the case in practice. Is there an online version of Antler? How difficult is it to modify the designed networks if the set of relevant tasks changes?

  [Response] Detailed explanation and experiment are provided in Section 5.2 “Effect of New Data Points” subsection. The convergence of task affinity is shown by using different ratios of the dataset examples.

- [C2] The distinction between subtasks and task graphs is not very clear in the paper. I initially thought that Antler would explicitly take advantage of common processing subtasks and design their overlap, but it seems that Antler instead looks at

similarities in the entire task graphs. Since the variety score is already being computed at various points in each task graph, would a finer-grained sharing of model information lead to further improvement?

[Response] Clarifications are presented in Section 2.1 “Tasks and Subtasks” subsection; Section 2.2 “Task Graph, Block, and Path” subsection. Figure 9 shows the effect of various branch point numbers.

Involves human subject research

---

1. Does not involve human and animal subjects

My detailed comments for authors

[C3] What is a “block” within a task network architecture? Is it a neuron in the neural network?

[Response] Please refer to C2.

[C4] How would a sample dataset be obtained to compute the variety metric? Wouldn’t this require knowing the distribution of the inference dataset? Or is it sufficient to simply know the range of possible data (I would think the full distribution would be needed, as one might care more about accuracy on more frequently occurring data points)?

[Response] Please refer to C1.

[C5] What is the complexity of the genetic algorithm solver that can be used to solve for the optimal task order? How close to the optimal are these solutions?

[Response] The complexity of two solvers are mentioned in Section 4.5, the corresponding optimality evaluation is provided in Table 3.

[C6] Why choose the task score at which the variety score and execution cost intersect? Wouldn’t it be better to try to optimize the sum of the two, or perhaps to optimize one subject to a constraint on the other? It’s also not clear whether the variety score and execution cost are of the same units, which makes the choice of their intersection, even if normalized, especially puzzling.

[Response] A unified analysis is provided in Section 3.2 “Empirical Tradeoff Curve”.

* *

Review #15D

==========

Overall merit

---

3. Weak accept

Reviewer expertise

---

2. Some familiarity

Is this paper exciting and thought-provoking?

---

3. Refreshing/daring idea or bold hypothesis (but execution has drawbacks)

Paper summary

The paper proposes Antler, a frameworks that exploits affinity and variety between pairs of tasks to design an effective computational graph representation and task scheduling for multi-task inference. The paper explains the theory behind the proposed approach, evaluates it on a read hardware and compares to the state-of-the-art approaches showing superior performance.

Strengths

---

* Well-motivated work and interesting idea how to generate an efficient task graph for multiple models. Comparison to the state-of-the-art on multiple datasets and architectures, and tests on a real hardware platform.

Weaknesses

---

* Justification of the proposed approach is somewhat lacking. Several claims are not well-supported through experiments /need further analysis and justification

Involves human subject research

---

1. Does not involve human and animal subjects

My detailed comments for authors

I enjoyed reading the paper, in particular the task graph generation part. The definitions of affinity and variety scores are sound, although could be better supported through experiments (Fig. 3 is not sufficient to understand the presented relationship for different datasets / architectures).

The points below summarize the questions I have on the paper.

* [D1] When looking at Fig. 1(b), can the generated graph have merged branches? Please provide justification.

[Response] We clarified in Section 2.2 that the task graph is a tree-like structure, hence, the branches do not merge or create any cycle. Because all networks start from the same root (input node), merging two branches after they branch out would mean that one of the branches is redundant.

* [D2] It would be helpful if the authors could compare their work to [18], which seems quite close to Antler.

[Response] The layer-wise multi–task Zipping cannot reuse intermediate results as Antler. Any time there are non-shared neurons in a layer, the results after that layers have to be recomputed.

* [D3] "In Antler, all tasks have the same network architecture." -- Given the different nature of tasks, it is completely unclear by how much this assumption limits the choice of the applications input to the Antler framework.

[Response] We have modified Antler to support pre-trained neural networks that may have different architectures as the input. A clarification is provided in Section 2.1 “Preprocessing”. A further investigation is provided in Section 5.3 “Knowledge Distillation” subsection.

* [D4] Sec. 3.1 argues that computing an affinity score can be expensive. However, this step is performed offline, before the models are deployed. How severe is the high computational cost of affinity in this case?

[Response] The high computational cost does not lie in computing affinity score itself, but lies in the fact that the number of possible decomposition trees that could be derived from n tasks is a Bell number problem (B_n). For each branch point, we have B_n ways of partitioning n tasks. If we have D branch points, the total computational complexity would be (B_n)^D. B_n has exponential computational complexity in terms of n.

* [D5] "Task graphs with low variety scores are generally desired as variety score tends to inversely correlate with inference accuracy". Such statements need justification.

[Response] This is a known result. We have included a citation in Section 3.2 as the justification. (K. Dwivedi and G. Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12387–12396, 2019.)

* *

Review #15E

==========

Overall merit

---

2. Weak reject

Reviewer expertise

---

2. Some familiarity

Is this paper exciting and thought-provoking?

---

2. Low risk idea or predictable hypothesis (but quite well executed)

Paper summary

This paper proposes Anlter, a system that leverages task affinity and ordering in multi-task settings to execute multiple tasks on embedded devices. Specifically, the system starts with a tasks set that manifests some sort of affinity and creates a task graph, where different tasks fork at different depths of the same backbone, in a tree-like manner. The root of the tree is shared, leading to performance gains, while the forking point is selected balancing the gains and dissimilarity of tasks. Subsequently, an execution order between tasks is established, via a genetic algorithm, based on the switching costs and inter-task dependencies.

Results from the evaluation over nine benchmark datasets show significantly better latency and energy behavior compared to three existing baseline multi-task systems, while at the same time the system is also evaluated on realistic deployments across two different modalities (audio, vision).

Strengths

---

* The proposed method is simple to understand.
* The evaluation of the system is quite well-executed over multiple tasks, spanning different modalities and compared with various multi-task baselines.
* Antler is actually run on devices of different capabilities and also deployed on real-life tasks and shows actual gainsovers the baselines with minimal accuracy degradation.

Weaknesses

---

* Presentation at times could be better, especially when presenting the algorithmic contributions.
* The evaluation lacks error statistics and experimental setup could include additional details for reproducibility.
* The overhead of the proposed solver and its generality and scalability have not been evaluated.

Involves human subject research

---

2. Research involving human and animal subjects, and ethical review/approval is mentioned in the paper

My detailed comments for authors

Technical issues

* [E1] The authors seem to assume an end-to-end identical architecture between tasks, whereas in reality this may only manifest in part of the networks across tasks. As such, this simplification should be described as a current limitation.

[Response] We have extended Antler to support heterogeneous pre-trained networks as one form of input via knowledge distillation. A clarification is provided in Section 2.1 “Preprocessing” and in Figure 1(a). A further investigation is provided in Section 5.3 “Knowledge Distillation” subsection.

* [E2] Another assumption is also the equal weighting of all tasks, which might be undesirable in a realistic deployment, especially when tasks have dependencies. The same applies to step 1 of §3.1 where each task t uses K samples, instead of K_t.

[Response] An explanation is provided in Section 3.1 Step 1 “Each task must use the same number of samples to ensure that the RDMs have the same dimensions.”

* [E3] The overhead of constructing the task graph and retraining is not particularly well-described in the text.

[Response] Please refer to response D4 for the overhead of constructing the task graph. The overhead of retraining is the same as standard neural network training.

[E4] While the overhead may be manageable for networks of microcontroller scale, it may be intractable for larger use-cases. I believe this should be clearly laid out in a limitations section.

[Response] Please refer to D4. The total complexity is (B_n)^D where D is the number of branch points and B_n is the Bell number.

* [E5] My understanding from the text is that the task graph generation and the execution ordering are all accomplished on the server-side.

[Response] Clarifications are provided in Section 3.3 - “Given n individually trained neural networks having the same architecture, generating the task graph offline is a four-step process”; Section 4.5 - “We describe two alternative solvers which run offline on a high-end machine.“

* [E6] In Figure 2, there is also the alternative of having different tasks as early-exits on the same backbone [a,b]

[Response] Although Antler’s task graph has resemblance with early-exit capable network architectures like [a, b], a fundamental difference is that task graph is a compact representation of multiple tasks where all tasks (hence, all blocks on the task graph) have to execute, whereas in early-exit, there is only one task that may exit from any of the exits and not all blocks are required to execute always.

* [E7] The experimental setup is missing:

[Response] Detailed description of experiment setup is provided in Section 5.1.

* [E8] Information about the server hardware.

[Response] Information is provided in Section 5.1 “Evaluation Platforms” subsection - “We run all off-line experiments on a server having 12 Intel i7-7800X CPUs and 32GB RAM.”

* [E9] How energy is measured.

[Response] Information is provided in Section 5.3 “Execution Time and Energy Cost” subsection - “The energy consumption is estimated by connecting a resistor in series and then measuring the voltage across the resistor and the system separately. “

* [E10] Figure 7: What hardware is this measured on?

[Response] Description is provided in the caption of Figure 10: MSP-16bit.

* [E11] In the evaluation, energy and latency metrics are missing variance metrics across runs, as well as details about the number of runs.

[Response] A clarification is provided in Section 5.3 “Execution Time and Energy Cost” subsection - “Since inference time and energy on microcontrollers is pretty stable, the error bars are practically zero in these figures.”

* [E12] §6.2, "Performance of genetic algorithm": This part could use a rewriting, as it is missing context and what it evaluates exactly in the case in point. Also, Table 3 is unclear what it measures in terms of its reported numbers.

[Response] Description is provided in the caption of Table 3 - “Node/Pre/Cnd represent the number of nodes, precedence, and conditional constraints, respectively. The last two columns represent the cost of the optimal result and that of Antler. The lower the better.”

* [E13] In the real-world deployment of §7, have the test-data been on a previously unseen user or is the train/test partitioning done blindly to the users?

[Response] Clarification is provided in Section 6.1 “Data Collection and Network Training” subsection - “We use 80% data for training and 20% for testing from each participant.”

[a] S. Laskaridis, A. Kouris, and N. D. Lane. 2021. Adaptive Inference through Early-Exit Networks: Design, Challenges and Directions. In Proceedings of the 5th International Workshop on Embedded and Mobile Deep Learning (EMDL'21).

Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3469116.3470012 [b]

Teerapittayanon, S., McDanel, B., & Kung, H. T. (2016, December). Branchynet: Fast inference via early exiting from deep neural networks. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 2464-2469). IEEE.

Questions

* [E14] Would Antler work if the sequence of tasks was presented in a sequential way, rather than all tasks at once? If so, how?

[Response] We assume that all tasks are given as the input at the same time. If tasks are incrementally added later, the task graph has to be recomputed for optimal performance. If a new task is added to the system but a new task graph is not recomputed, Antler framework would still support that extra task and run it as if it were an isolated branch in the task graph, and in this case, the inference results would still be correct, with the only downside is that the performance in terms of time and energy will be suboptimal.

* [E15] What would happen if the original tasks did not share the same DNN backend? Could Antler somehow marry different DNNs via distillation for example?

[Response] We have extended Antler to support heterogeneous input networks via knowledge distillation. Discussion is provided in Section 5.3 “Knowledge Distillation.” subsection.

* [E16] Could Antler work with pretrained models of different tasks and no access to the original training set? If so, how?

[Response] Explanation is provided in Section 5.3 “Knowledge Distillation.” subsection

* [E17] How does Antler interact with compression methods, such as quantisation? Could some tasks be quantised while others not and still have an functional system?

[Response] Knowledge distillation is chosen and described in Section 5.3 “Knowledge Distillation.” subsection.

* [E18] Is there a limit to the number of tasks Antler could handle?

[Response] This is described in Section 5.2 “Effect of Number of Tasks” subsection.

* [E18] It is somewhat unclear to me why the cost matrix C_{n \times n } is symmetric.

[Response] No matter if we are transiting from t_i to t_j or from t_j to t_i, the difference between the two tasks, t_i and t_j, is the same so that the transit cost should be the same, which can also be seen from Figure 1.

* [E19] From the paper, I understand that there is a single multi-task model deployed across clients. Do the authors assume that the conditional constraints described in §4.3 are IID across clients? How would the authors model the case where different clients have different (and diverse) distributions of values that lead to varying execution rates of different tasks?

[Response] Antler is a generic framework that can be customized for individual clients by using client-specific dataset and task dependencies.

Presentation

* [E20] The definition of what affinity is is quite central to the main point of the paper, and thus I would recommend moving the definition from §3.1 to the introduction maybe?

[Response] Introduction introduces affinity and provides examples. A formal definition of Affinity is in Section 3.1.

* [E21] Steps in §3.1, §3.3 and §9.2 could be more succinctly presented in an algorithmic form

[Response] For space constraints, we describe the algorithms as steps as opposed to using the Algorithm environment in latex.

* [E22] Having an Appendix as part of the main manuscript is a bit odd. Maybe the authors could incorporate the Appendix Sections in the main narrative of their text.

  [Response] We have removed the appendix.

* [E23] §5.3, "Tensor Flow" --> "TensorFlow"* §5.3, "Third, All" --> "Third, all"

  [Response] Typos have been corrected.

* [E24] Figure 12 could be better presented if accuracy was wrt the baseline accuracy of "vanilla".

  [Response] We have redrawn the plot for better readability.

* [E25] Figure 16 would be more legible if a smaller y-axis range was used.

  [Response] We have redrawn the plot for better readability.

Comment @A1 by Reviewer E

The summary of the discussion during the PC meeting is the following:

The paper has merit and we liked the approach of leveraging task affinity for bringing the cost of multi-DNN execution down by leveraging multi-task (MT) affinity, but the program committee felt that the current state of the manuscript was missing substantial details about the system and its limitations.

Specifically, it was agreed that:

* [F1] The overhead of running the task graph generation and task ordering has not been quantified.

  [Response] We have clarified in Section 3.3 and 4.5 that these are offline algorithms that run on GPU-enabled high-end machines, hence, the cost of task graph generation and task ordering are not part of the execution cost of the low-resource target system. Nevertheless, the task graph generation algorithm's dominating overhead is from the retraining of the network, which varies depending on the dataset and network. We have included the algorithmic complexity of the task ordering algorithm in Section 4.5.

* [F2] The scalability aspect of Antler, with respect to the size of the DNNs and the number of tasks to be solved and the number of potential fork points, was missing.

  [Response] Detailed explorations are provided in evaluation Section 5.2. "Effect of Network Structure." and "Effect of Number of Tasks.", "Effect of Branch Points." subsections.

* [F3] The test accuracy on the solved tasks was suspiciously high, as presented in Figure 12.

  [Response] The newly measured test accuracy is presented in Figure 15.

* [F4] The current state of the system only allows tasks to have the same backbone, which can largely not be the case across MT scenarios.

  [Response] We have extended Antler by implementing knowledge distillation to support heterogeneous networks as one form of input. MTL is a broad literature. Antler is applicable to common MTL architectures that generally have some shared layers among tasks (e.g., hard parameter sharing) as well as task-specific layers.

* [F5] Another limitation was that Antler requires access to training/validation data .

  [Response] Through the implementation of knowledge distillation, Antler now supports pre-trained networks as another form of input. User-contributed training dataset is no longer the only option to train the common network architecture.

* [F6] There was also the implicit assumption that all tasks are equal, static and that deployment targets have the same frequency/importance for these tasks**.** As such, there is a limitation on static and homogeneous MT environment.

  [Response] Detailed explanations and discussions are provided in Section 5.2 "Effect of New Data Points." subsection.

Nevertheless, it was recognised that Antler presents a promising approach and we would encourage the authors to continue pursuing this line of work.