TASK-AWARE NEURAL ARCHITECTURE SEARCH

Cat P. Le, Mohammadreza Soltani, Robert Ravier, Vahid Tarokh

Department of Electrical and Computer Engineering, Duke University

2021 IEEE International Conference on Acoustics, Speech and Signal Processing

Duke

Motivation

- The design of handcrafted neural networks for a task requires a lot of time and resources.
- Current neural architecture search techniques require domain knowledge to define the search space.
- The goal is to utilize the knowledge of previous (base) task to design a suitable search space for the incoming (target) task.

Approach

- Given a dictionary of previous task-data pairs.
- For any incoming target task-data pair, our goal is to find an architecture for achieving high performance on the target task.
- TA-NAS works as follows:
  1. **Task Similarity:** Given an incoming task-data set pair, TA-NAS finds the most related task-data set pairs in the dictionary.
  2. **Search Space**: TA-NAS defines a suitable search space for the target task-data set pair, based on the related pairs.
  3. **Search Algorithm**: TA-NAS searches to discover an optimal architecture in term of performance for the target task-data set pair on the search space.

Task Similarity

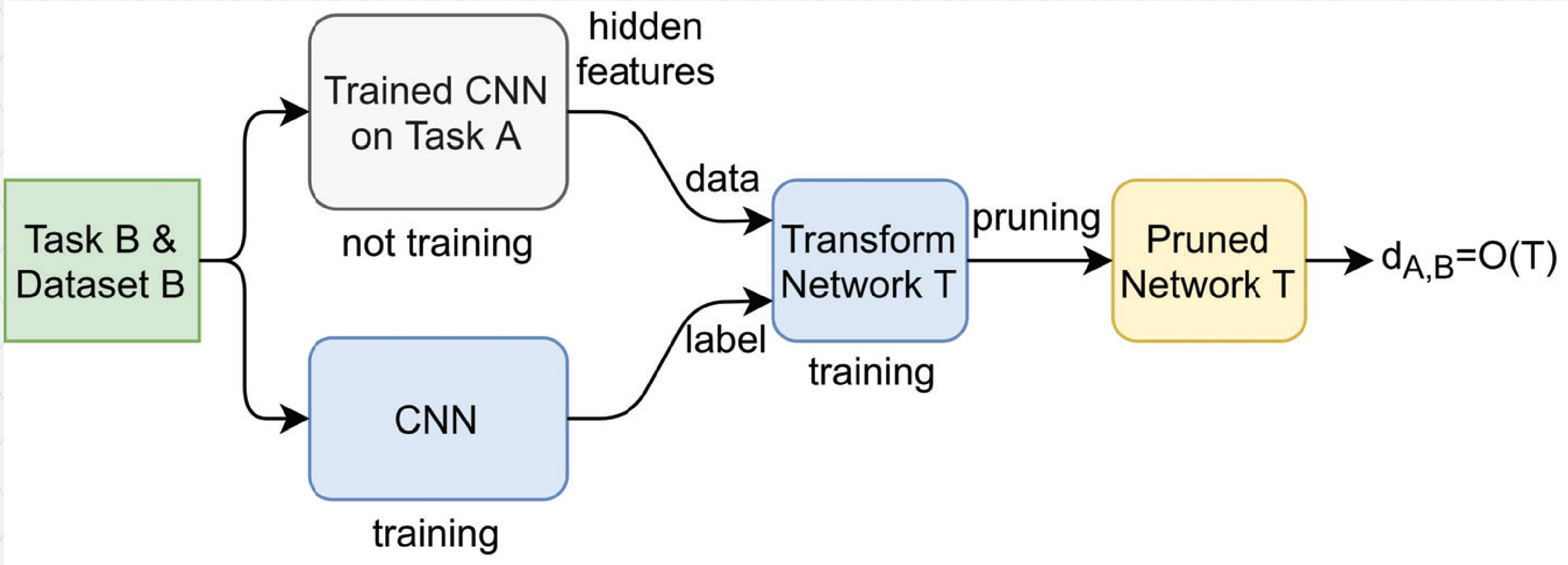
- We represent a task by a sufficiently trained neural network.
- Let A = (T_A, X_A) and B = (T_B, X_B) be two task-data set pairs, where N_A and N_B are two trained architectures that are ε-representative for A and B, respectively.
- We can define a dissimilarity measure between A and B as follows:

$$d_{A,B}^{\epsilon} = \min_{N_t \in S_t :\, \mathcal{L}_B(N_t \circ N_A) \geq 1-\epsilon} O(N_t)$$

where S_t is a given transform network search space, and O() is a general measure of complexity, and N_t is the network that take the last-layer hidden features of N_A and transform them into N_B's.

Duke

Task Similarity



Search Space

- The search space is defined by the structures of cell and skeleton.
- A cell is a densely connected directed-acyclic graph of nodes, where all nodes are connected by operations.
- The skeleton is often predefined.
- Here, we construct the search space of the target task by combining the skeletons, cells, and operations from only the most similar pairs in the dictionary.

Fusion Search (FUSE)

- Fusion Search (FUSE) is a search algorithm that considers the network candidates as a whole and performs the optimization using gradient descent. For any set of **C** candidates, we relax the outputs by exponential weights:

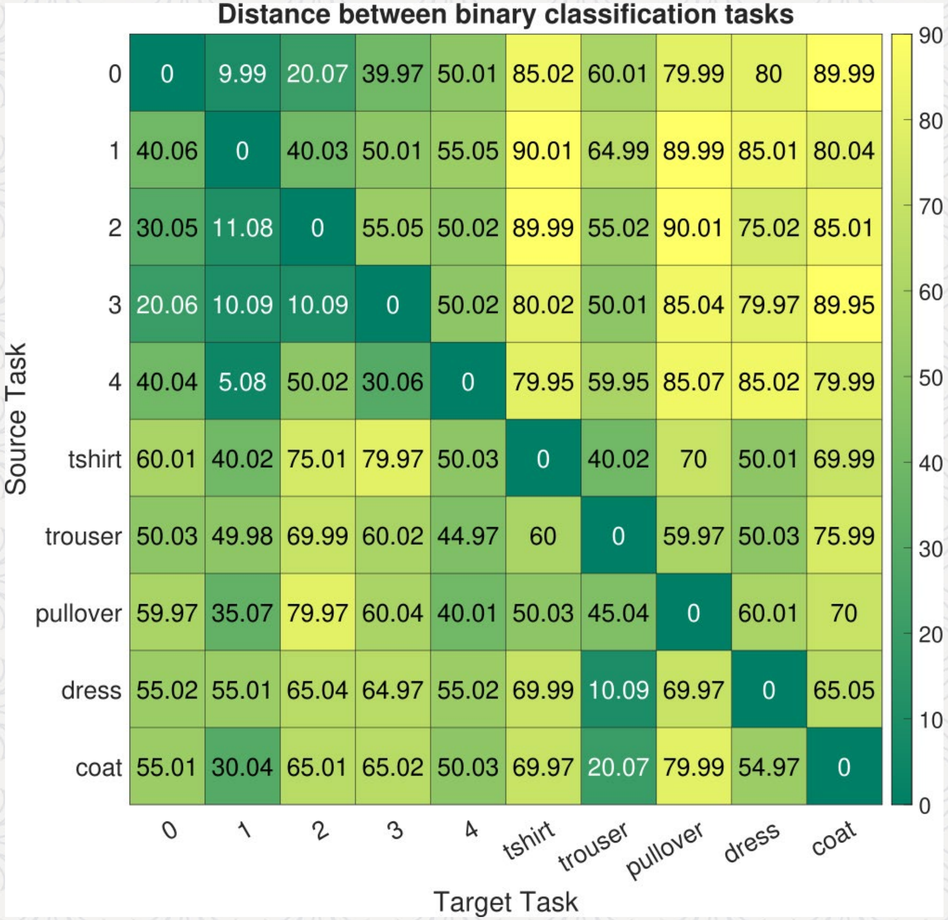
$$\bar{c}(X) = \sum_{c \in \boldsymbol{C}} \frac{\exp(\alpha_c)}{\sum_{c' \in \boldsymbol{C}} \exp(\alpha_{c'})} c(X)$$

- The training procedure is based on alternative minimization and can be divided into:
    1. freeze α, train network's weights: $\min_{w} \mathcal{L}(w; \alpha, \bar{c}, X_{train})$
    2. freeze network's weights, update α: $\min_{w} \mathcal{L}(\alpha; w, \bar{c}, X_{val})$

Result

- For our experiment, we initialize with a set of base binary classification tasks consisting of finding specific digits in MNIST and specific objects in Fashion-MNIST.
- Let the target task be the binary classification task from Quick, Draw! data set. Tasks from the same data set are more similar than tasks from different data sets.

Result



Architecture	Error (%)	Param (M)	GPU days
ResNet-18	1.42	11.44	-
ResNet-34	1.2	21.54	-
DenseNet-161	1.17	27.6	-
Random Search	1.33	2.55	4
FUSE w. standard space	1.21	2.89	2
FUSE w. task-aware space	1.18	2.72	2

Conclusion

- We proposed TA-NAS to address the Neural Architecture Search problem.
- By introducing the task similarity, we can create a restricted search space and quickly evaluate candidates using the FUSE search algorithm.
- This search algorithm can be applied to find the best way to grow or to compress the current network.