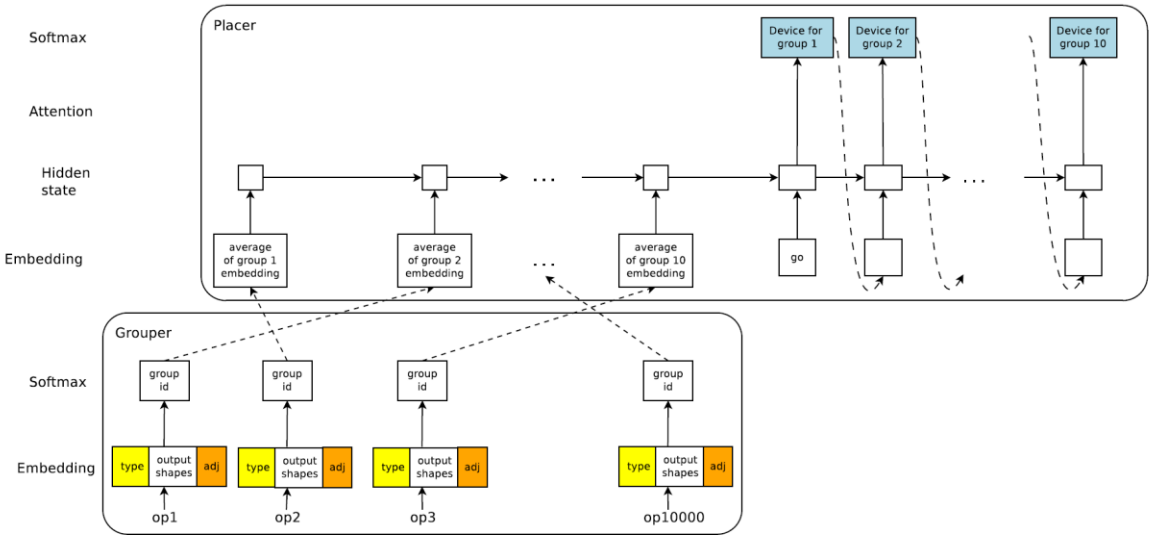
A HIERARCHICAL MODEL FOR DEVICE PLACEMENT

1. A hierarchical model for efficient placement of computational graphs onto hardware devices, especially in heterogeneous environments with a mixture of CPUs, GPUs, and other computational devices.



2. The grouping and device allocation are learned jointly.

3. The model consists of two sun-networks i) assigns operations to the group ii) Placer that assigns groups to target devices.

4. **Main objective** is to predict a placement that speeds up the training of neural network graphs. The runtime we are optimizing for is the time taken to conduct one forward pass, one back-propagation pass, and one parameter update on the target neural network.

5. **Steps to implement**

   a. The Grouper is a feed forward model followed by a softmax layer with an output size equal to the number of groups. The Placer is a sequence-to-sequence model with Long Short-Term Memory and a content-based attention mechanism.

b. First generate operation embedding’s , each has three vectors 1) operation type information, 2) output sizes and number of outputs, 3) adjacency information

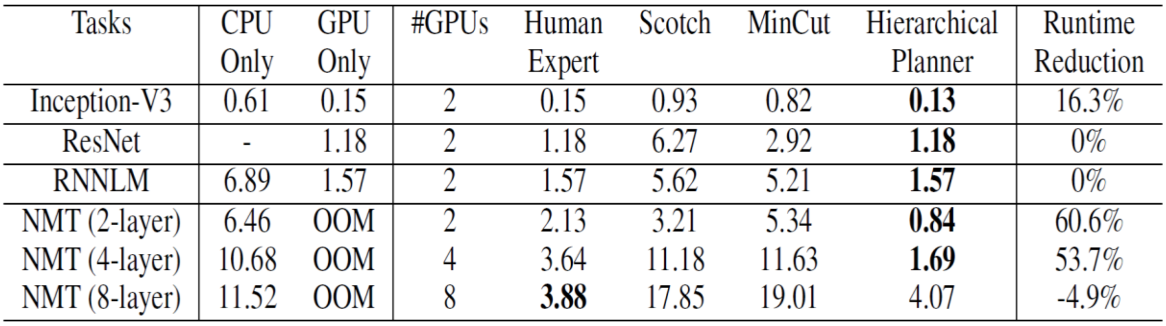
c. To generate input for the Placer, 1) Count of each operation type in the group. 2) Count the total number of output shapes of all the operations in that group. Concatenating all the operation output shape embedding’s described. 3) contains group adjacency information, its i-th value is 1 if the group has edges to the i-th group and 0 otherwise.

d. The decoder uses an attention mechanism to attend over the encoder states.

$$d_t \sim softmax(Ctanh(\tfrac{l_t}{T})) \qquad (1)$$

Where temperature=T, lt= activations, constant =C

6. Experiment: Hierarchical Planner to widely used machine learning models in computer vision and natural language processing. Compare our results to heuristic and RL-based graph optimization baselines and demonstrate. Two simpler alternatives 1) no grouping, 2) Random grouping
7. Models: four widely used deep neural networks 1) Inception-V3, 2) ResNet, 3) RNNLM, 4) NMT
8. Baselines: CPU Only, GPU Only, Scotch, MinCut, Human Expert, ColocRL.
9. **Devices and Software:** Our experiments are run on machines with 1 Intel Haswell 2300 CPU and up to 8 Nvidia Tesla K40 GPUs. We use TensorFlow r1.3 to run our experiments.



Tasks	CPU Only	GPU Only	#GPUs	Human Expert	Scotch	MinCut	Hierarchical Planner	Runtime Reduction
Inception-V3	0.61	0.15	2	0.15	0.93	0.82	**0.13**	16.3%
ResNet	-	1.18	2	1.18	6.27	2.92	**1.18**	0%
RNNLM	6.89	1.57	2	1.57	5.62	5.21	**1.57**	0%
NMT (2-layer)	6.46	OOM	2	2.13	3.21	5.34	**0.84**	60.6%
NMT (4-layer)	10.68	OOM	4	3.64	11.18	11.63	**1.69**	53.7%
NMT (8-layer)	11.52	OOM	8	**3.88**	17.85	19.01	4.07	-4.9%

10. Results Compared with Graph Partitioning Heuristics