Device Placement Optimization with Reinforcement Learning
A Hierarchical Model for Device Placement

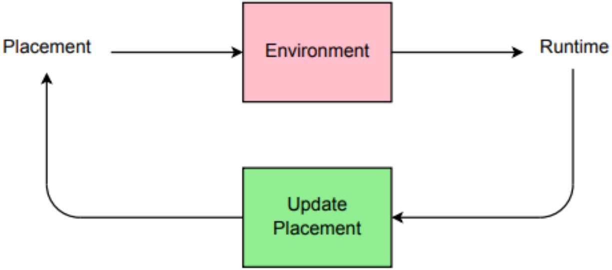
A. Mirhoseini, Hieu Pham, A. Goldie et al

November 2019

Problem Background

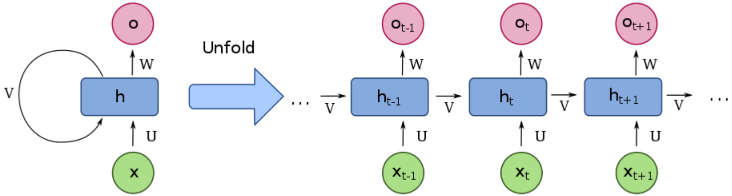
- Tensorflow allows user to place operators on different devices to take advantage of *parallelism* and *heterogeneity*
- Current solution: human experts use heuristics to place the operators as best they can
- Some simple graph-based automated approaches (e.g. Scotch) perform worse

Approach



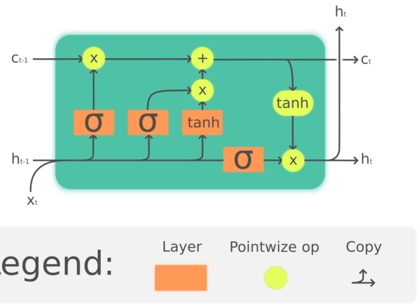
- Use reinforcement learning and neural nets to find the best placement

Background: RNNs



- RNNs model dependencies between data; they have persistence
- E.g. previous words or **previous placements of operators**

Background: LSTM and the Vanishing Gradient Problem

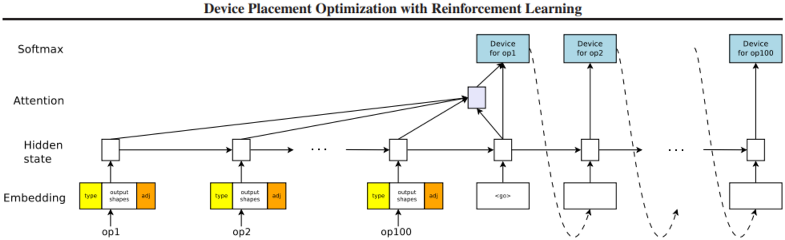


- Too many multiplications means gradient quickly diminishes to 0
- Gated structure can model long term dependencies better
- Forget, input and output gates control a hidden state

Background: Reinforcement Learning

- Traditional use of NNs is in a supervised setting with labelled training data
- Need to learn from the environment
- Want to maximise the expected reward:
  $J(\theta) = \sum_{\tau} P(\tau;\theta)R(\tau)$
- The derivative, $\nabla_{\theta} J(\theta)$ is equivalent to
  $\sum_{\tau} P(\tau;\theta)\nabla_{\theta} log(P(\tau;\theta)R(\tau)$
- This is actually an expected value, so can use monte-carlo sampling to approximate:
  $\nabla_{\theta} J(\theta) \approx \frac{1}{K}\sum_{i=1}^{K} R(x_i)\nabla_{\theta} log(P(x_i|\theta))$

Implementation: Neural network architecture



- Sequence-to-sequence model; this is two RNNs that communicate via shared state
- Input: sequence of vectors representing the **type** of each operation, output sizes, encoding of links with other operators
- Output: placements for operations

Implementation: RL

- Uses monte-carlo sampling as discussed
- Reward function is the square-root of running time
- High fixed cost for OOM on e.g. single GPUs
- Subtract a moving average from reward to decrease variance

Grouping

- Dataflow graph huge: big search space and vanishing gradient
- Solution one: Co-locate operators manually into groups that should be executed on the same device
- Solution two: Add another (feed-forward) neural network, the **grouper**
- Hierarchical approach: **grouper** and **placer**

Evaluation: Experimental setup

- Measure time for single step of several different models: RNNLM, NMT, Inception-V3, ResNet
- Run on a single machine, using CPU and 2 - 8 GPUs
- Baselines are single CPU, single GPU, using the Scotch library, expert placement

Evaluation: Results

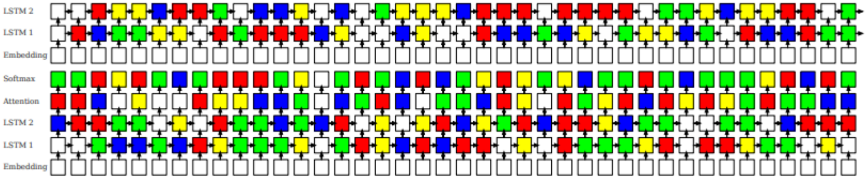
Tasks	CPU Only	GPU Only	#GPUs	Human Expert	Scotch	MinCut	Hierarchical Planner	Runtime Reduction
Inception-V3	0.61	0.15	2	0.15	0.93	0.82	**0.13**	16.3%
ResNet	-	1.18	2	1.18	6.27	2.92	**1.18**	0%
RNNLM	6.89	1.57	2	1.57	5.62	5.21	**1.57**	0%
NMT (2-layer)	6.46	OOM	2	2.13	3.21	5.34	**0.84**	60.6%
NMT (4-layer)	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%

- Only 3 hours for hierarchical model
- Performance significantly better than the manually co-located version

Evaluation: Understanding the results

- Classic tradeoff: distributing more for more parallelism, want to minimise copying costs
- Different architectures have different amounts of parallelism available to exploit

Strengths



- Hierarchical planner completely end-to-end
- Overhead of three hours is small (original paper 13-27 hours)
- Capable of finding complex placements which are beyond a human
- Sometimes very substantial improvements

Weaknesses

- First paper not reproducible: don't mention the version of Tensorflow, even original authors couldn't reproduce results
- Results mixed; often no improvement if best placement is trivial. Can this be determined by looking at the amount of parallelism in the graph?
- Will it scale? NMT 8-layer has a decrease in performance compared to human expert. Why this sudden decline?
- How many times did they run the random RL process?
- Incorporate humans to improve placements even further

Questions