Research Talk

Anna Goldie

Overview

- Natural Language Processing
  - **Conversational Modeling (Best Paper Award at ICML Language Generation Workshop, EMNLP 2017)**
  - Open-Source tf-seq2seq framework (4000+ stars, 1000+ forks), and exploration of NMT architectures (EMNLP 2017, 100+ citations)
- Deep Dive: ML for Systems
  - Device Placement with Deep Reinforcement Learning (ICLR 2018)

Tell me a story about a bear...

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Tell me a story about a bear...

a. "I don't know."

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Tell me a story about a bear...

a. "I don't know."

b. "A bear walks into a bar to get a drink, then another bear comes and sits in his room with the bear thought he was a wolf."

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Motivation: Generate Informative and Coherent Responses

- Address shortcomings of sequence-to-sequence models
  - Short/generic responses with high MLE in virtually any context
    - “I don’t know.”
  - Incoherent and redundant responses when forced to elaborate through explicit length promoting heuristics
    - “I live in the center of the sun in the center of the sun in the center of the sun…”

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Method Overview

- Generate segment by segment
  - Inject diversity early in generation process
  - Computationally efficient form of target-side attention
- Stochastic beam search
  - Rerank segments using negative sampling

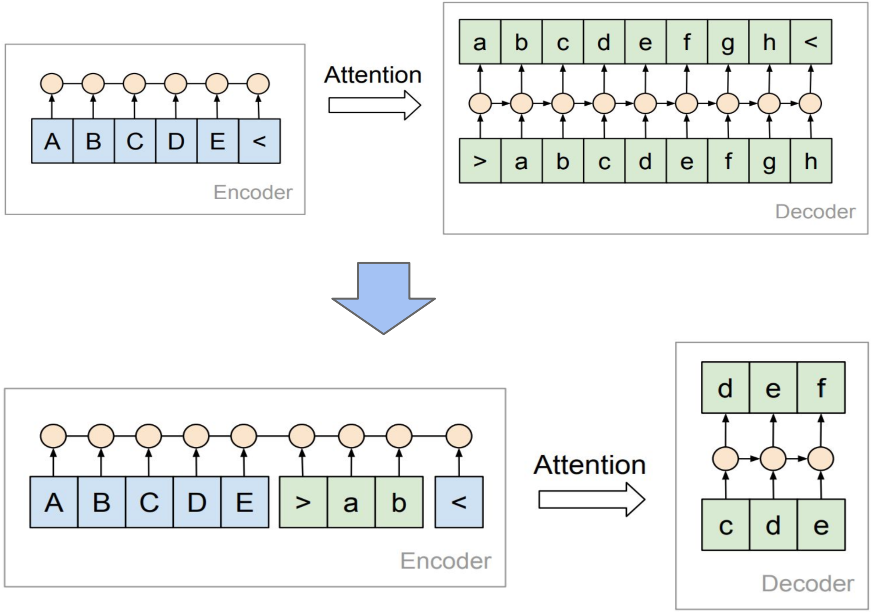
Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Self Attention for Coherence

- Glimpse Model: Computationally efficient form of Self Attention

- Memory capacity of the decoder LSTM is a bottleneck

- So, let decoder also attend to the previously generated text



Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Stochastic Beam Search with Segment Reranking

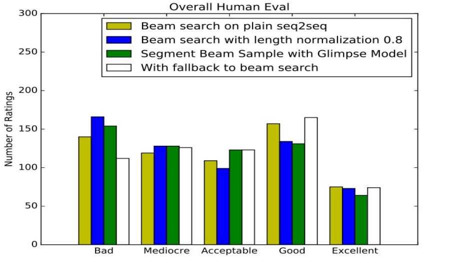
$$\text{Tell me a story.} \rightarrow \left\{\begin{array}{ccc} \text{Once} & \text{upon} & \text{a} \\ \text{A} & \text{bear} & \text{walks} \\ \text{Sparky} & \text{the} & \text{wonder} \end{array}\right.$$

$$S\left(\mathbf{y}_k|\mathbf{x},\mathbf{y}_{1:k-1}\right) = \frac{P\left(\mathbf{y}_k|\mathbf{x},\mathbf{y}_{1:k-1}\right)}{\sum_{\mathbf{x}'\in\Phi} P\left(\mathbf{y}_k|\mathbf{x}',\mathbf{y}_{1:k-1}\right)}$$

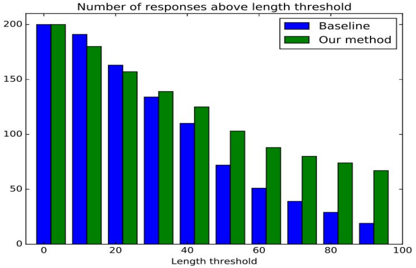
$$\text{Tell me a story.} \rightarrow \text{A bear walks} \left\{\begin{array}{ccc} \text{around} & \text{like} & \text{a} \\ \text{into} & \text{a} & \text{bar} \\ \text{in} & \text{a} & \text{hurry} \end{array}\right.$$

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

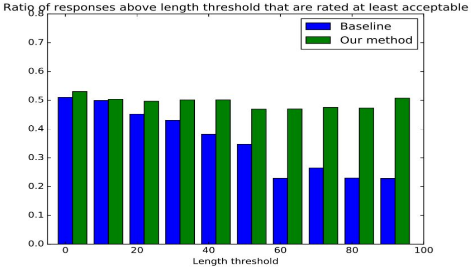
Evaluation



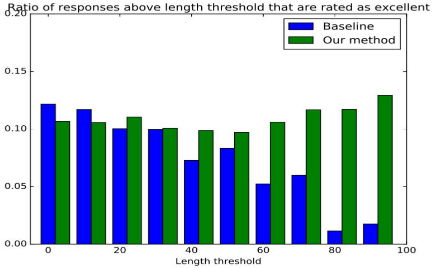
(a)



(b)



(c)



(d)

N for computing "N-choose-1"	**50**	**10**	**2**
No Normalization	0.047	0.15	0.56
Normalize by Marginal	0.44	0.65	0.91
Normalize by Random Prompts (our heuristics)	**0.61**	**0.78**	**0.97**

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Sample Conversation Responses

PROMPT:	RESPONSE FROM BASELINE (A) vs OUR METHOD (B)
what do you have planned tonight	**A:** I have no idea what I want to do with my life. I have no idea what I want to (Bad, Mediocre, Bad) **B:** I was planning on doing a few of these, but I'm not sure what to choose. I need to find a place to camp, and I don't have a lot of time to make a plan. (Excellent, Good, Mediocre)
I'd like to read a good science fiction novel. Haven't read scifi in a while.	**A:** If you like sci fi you should check out The Name of the Wind by Patrick Rothfuss (Excellent, Excellent, Excellent) **B:** If you want to know what sci-fi is, read the first book by Neil Gaiman. It's a sci-fi sci-fi sci-fi novel. It is really good and I recommend it to all of you. It has a great premise. (Acceptable, Excellent, Acceptable)
Got any big weekend plans?	**A:** I have plans for the weekend but I don't know if I will be able to make it to the (Bad, Mediocre, Bad) **B:** I don't really have much to do on Saturday. But I plan on going to the movies at the mall and going to the show. (Good, Good, Bad)
Sometimes flying messes my ears, because of the air pressure. Especially if I have a cold or something, flying gets me messed up.	**A:** I don't know why, but when I was a kid I used to do this all the time when I was younger. (Good, Bad, Bad) **B:** I have no idea why, but I feel exactly the same way. I don't know why, but my brain is so sensitive to air that it makes my nose twitch. (Excellent, Mediocre, Good)
Yeah. I have this friend, and we get bored just hanging out here sometimes, so we take little trips to places. We went to Seattle a couple of months ago.	**A:** What kind of things do you like to do? (Excellent, Excellent, Excellent) **B:** That sounds like a great time. I have a lot of friends that go to Seattle, so I was wondering if you could share any of the locations? (Excellent, Excellent, Excellent)
have you seen Suicide Squad?	**A:** I have not, but I will check it out. (Good, Good, Good) **B:** I saw it in a theater once and it was great. It was very... I don't know, a little dark. I like the psychological effects and the way it portrays the characters in the movie. (Excellent, Excellent, Excellent)
Actually I like the snow too, except for today. It's all slushy and it's all over the sidewalks and it's gross.	**A:** I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. I hate the snow. (Acceptable, Good, Good) **B:** I'm in the same boat as you. I hate it when it's snowy and I don't like to shovel it all day. :(So this is a perfect excuse to not get out of bed and go somewhere else. (Excellent, Good, Acceptable)
What do you hope to see in the future?	**A:** I want to be able to go back in time to when I was born. I want to be (Mediocre, Bad, Mediocre) **B:** I want to see an actual movie where the characters have the ability to hide the fact that they are actually in the movie. (Good, Acceptable, Acceptable)
what dangers could happen with the singularity?	**A:** I don't think it would be that big of a deal. (Acceptable, Mediocre, Acceptable) **B:** If you look at it from the perspective of an alien, you can't say that there aren't any alternative ways to deal with it. (Acceptable, Mediocre, Acceptable)
I think you're a little crazy.	**A:** I don't think you know what that word means. (Acceptable, Good, Good) **B:** I am, actually. I don't think I'm crazy though. I certainly am. (Acceptable, Mediocre, Acceptable)

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

1st workshop on Learning to Generate Natural Language
@ICML 2017, Sydney Australia

Best Paper Award

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models

Louis Shao*, Stephan Gouws*, Anna Goldie *, Denny Britz, Brian Strope, Ray Kurzweil

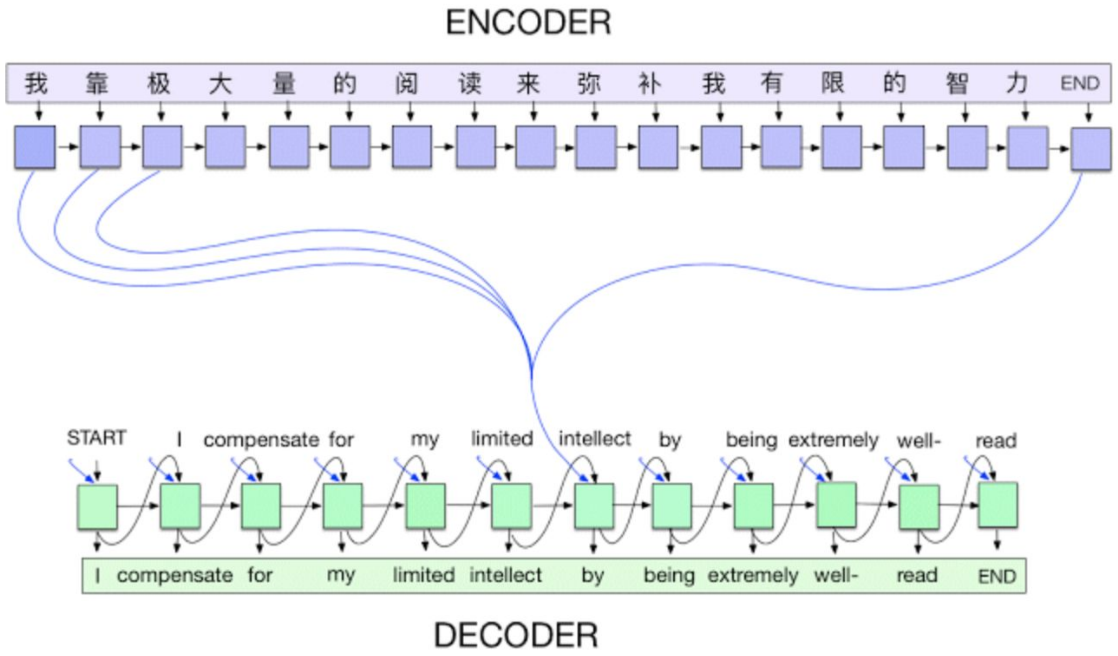
Sponsors DeepMind Bloomberg Maluuba

Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models, EMNLP 2017

Overview

- Natural Language Processing
  - Conversational Modeling (Best Paper Award at ICML Language Generation Workshop, EMNLP 2017)
  - **Open-Source tf-seq2seq framework (4000+ stars, 1000+ forks), and exploration of NMT architectures (EMNLP 2017, 100+ citations)**
- Deep Dive: ML for Systems
  - Device Placement with Deep Reinforcement Learning (ICLR 2018)

tf-seq2seq: A general-purpose encoder-decoder framework for Tensorflow



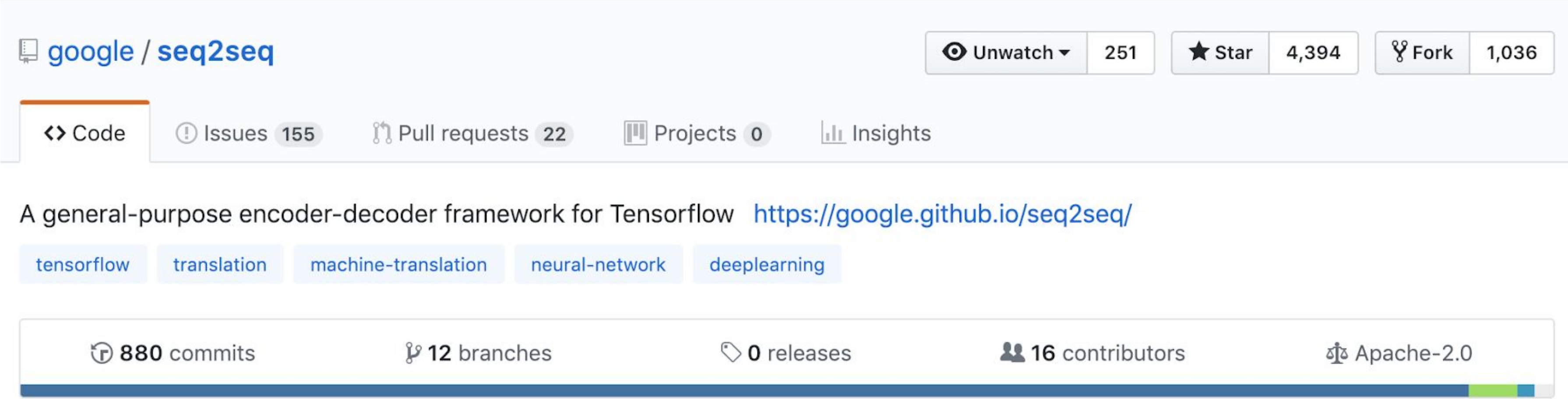
Massive Exploration of Neural Machine Translation Architectures, EMNLP 2017

Goals for the Framework

- **Generality**: Machine Translation, Summarization, Conversational Modeling, Image Captioning, and more!
- **Usability**: Train a model with a single command. Several types of input data are supported, including standard raw text.
- **Reproducibility**: Training pipelines and models configured using YAML files
- **Extensibility**: Code is modular and easy to build upon
  - E.g., adding a new type of attention mechanism or encoder architecture requires only minimal code changes.
- **Documentation**:
  - All code is documented using standard Python docstrings
  - Guides to help you get started with common tasks.
- **Performance**:
  - Fast enough to cover almost all production and research use cases
  - Supports distributed training

Massive Exploration of Neural Machine Translation Architectures, EMNLP 2017

Reception



- Featured in AMTA Panel on “Deploying Open Source Neural Machine Translation (NMT) in the Enterprise”
- Used in dozens of papers from top industry and academic labs

Massive Exploration of Neural Machine Translation Architectures, EMNLP 2017

Massive Exploration of Neural Machine Translation Architectures

Denny Britz*, Anna Goldie*, Minh-Thang Luong, Quoc V. Le
{agoldie,thangluong,qvl}@google.com

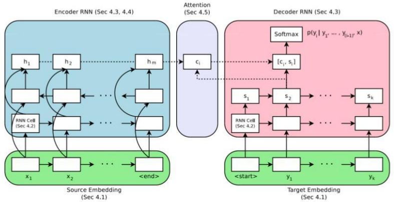
Abstract

One major drawback of current Neural Machine Translation (NMT) architectures is that they are expensive to train, typically requiring days to weeks of GPU time to converge. This makes exhaustive hyperparameter search, as is commonly done with other neural network architectures, prohibitively expensive. In this work, we present the first large-scale analysis of NMT architecture hyperparameters. We report empirical results and variance numbers for several hundred experimental runs, corresponding to over 250,000 GPU hours on the standard WMT English to German translation task. Our experiments lead to novel insights and practical advice for building and extending NMT architectures.

Open Source Framework: **tf-seq2seq**

- We ran all experiments on **tf-seq2seq**, our own open source framework in TensorFlow that makes it easy to experiment with seq2seq models and achieve state-of-the-art results
- **tf-seq2seq** supports various configurations of the standard seq2seq model, such as depth of the encoder/decoder, attention mechanism, RNN cell type, and beam size
- https://google.github.io/seq2seq/ has tutorials and source code

Network Architecture: Sequence to Sequence Model



Results

Embedding Dimensionality

Dim	newstest2013	Params
128	21.50 ± 0.16 (21.66)	36.13M
256	21.73 ± 0.09 (21.85)	46.20M
512	21.78 ± 0.05 (21.83)	66.32M
1024	21.36 ± 0.27 (21.67)	106.58M
2048	**21.86** ± 0.17 (22.08)	187.09M

Results

RNN Cell Variant

Cell	newstest2013	Params
LSTM	**22.22** ± 0.08 (22.33)	68.95M
GRU	21.78 ± 0.05 (21.83)	66.32M
Vanilla-Dec	15.38 ± 0.28 (15.73)	63.18M

Encoder and Decoder Depth and Type of Residual Connections

Depth	newstest2013	Params
Enc-2	21.78 ± 0.05 (21.83)	66.32M
Enc-4	**21.85** ± 0.32 (22.23)	69.47M
Enc-8	21.32 ± 0.14 (21.51)	75.77M
Enc-8-Res	19.23 ± 1.96 (21.97)	75.77M
Enc-8-ResD	17.30 ± 2.64 (21.03)	75.77M
Dec-1	21.76 ± 0.12 (21.93)	64.75M
Dec-2	21.78 ± 0.05 (21.83)	66.32M
Dec-4	**22.37** ± 0.10 (22.51)	69.47M
Dec-4-Res	17.48 ± 0.25 (17.82)	68.69M
Dec-4-ResD	21.10 ± 0.24 (21.43)	68.69M
Dec-8	01.42 ± 0.23 (1.66)	75.77M
Dec-8-Res	16.99 ± 0.42 (17.47)	75.77M
Dec-8-ResD	20.97 ± 0.34 (21.42)	75.77M

Unidirectional vs Bidirectional Encoders

Cell	newstest2013	Params
Bidi-2	**21.78** ± 0.05 (21.83)	66.32M
Uni-1	20.54 ± 0.16 (20.73)	63.44M
Uni-1R	21.16 ± 0.35 (21.64)	63.44M
Uni-2	20.98 ± 0.10 (21.07)	65.01M
Uni-2R	21.76 ± 0.21 (21.93)	65.01M
Uni-4	21.47 ± 0.22 (21.70)	68.16M
Uni-4R	21.32 ± 0.42 (21.89)	68.16M

"R" suffix indicates a reversed source sequence.

Attention Mechanism

Attention	newstest2013	Params
Mul-128	22.03 ± 0.08 (22.14)	65.73M
Mul-256	22.33 ± 0.28 (22.64)	65.93M
Mul-512	21.78 ± 0.05 (21.83)	66.32M
Mul-1024	18.22 ± 0.03 (18.26)	67.11M
Add-128	22.23 ± 0.11 (22.38)	65.73M
Add-256	22.33 ± 0.04 (22.39)	65.93M
Add-512	**22.47** ± 0.27 (22.79)	66.33M
Add-1028	22.10 ± 0.18 (22.36)	67.11M
None-State	9.98 ± 0.28 (10.25)	64.23M
None-Input	11.57 ± 0.30 (11.85)	64.49M

Results

Beam Search Strategies

Beam	newstest2013	Params
B1	20.66 ± 0.31 (21.08)	66.32M
B3	21.55 ± 0.26 (21.94)	66.32M
B5	21.60 ± 0.28 (22.03)	66.32M
B10	21.57 ± 0.26 (21.91)	66.32M
B25	21.47 ± 0.30 (21.77)	66.32M
B100	21.10 ± 0.31 (21.39)	66.32M
B10-LP-0.5	21.71 ± 0.25 (22.04)	66.32M
B10-LP-1.0	**21.80** ± 0.25 (22.16)	66.32M

Varying the beam width and adding length penalties (LP).

Final System Comparison

Hyperparameter	Value
embedding dim	512
rnn cell variant	LSTMCell
encoder depth	4
decoder depth	4
attention dim	512
attention type	Bahdanau
encoder	bidirectional
beam size	10
length penalty	1.0

Model	newstest14	newstest15
Ours (experimental)	22.03	24.75
Ours (combined)	22.19	25.23
OpenNMT	19.34	-
Luong	20.9	-
BPE-Char	21.5	23.9
BPE	-	20.5
RNNSearch-LV	19.4	-
RNNSearch	-	16.5
Deep-Att*	20.6	-
GNMT*	24.61	-
Deep-Conv*	-	24.3

Systems with an * do not have a public implementation.

Conclusions

- Large embeddings with 2048 dimensions achieved the best results, but only by a small margin. Even small embeddings with 128 dimensions seem to have sufficient capacity to capture most of the necessary semantic information.
- LSTM Cells consistently outperformed GRU Cells.
- Bidirectional encoders with 2 to 4 layers performed best. Deeper encoders were significantly more unstable to train, but show potential if they can be optimized well.
- Deep 4-layer decoders slightly outperformed shallower decoders. Residual connections were necessary to train decoders with 8 layers and dense residual connections offer additional robustness.
- Parameterized additive attention yielded the overall best results.
- A well-tuned beam search with length penalty is crucial. Beam widths of 5 to 10 together with a length penalty of 1.0 seemed to work well.

Massive Exploration of Neural Machine Translation Architectures, EMNLP 2017

Takeaways

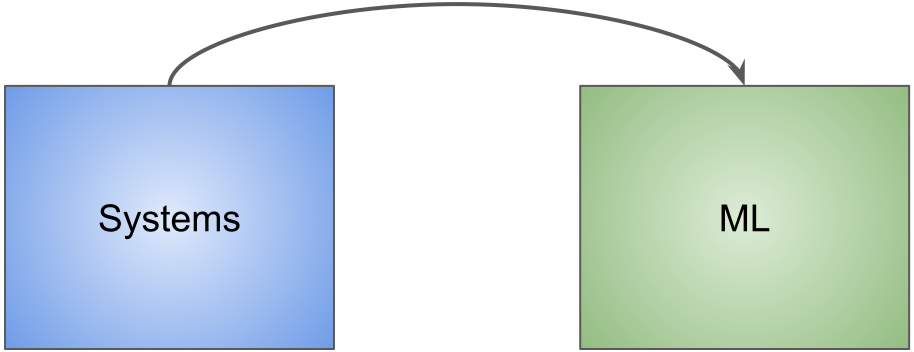
- LSTM Cells consistently outperformed GRU Cells.
- Parameterized additive attention outperformed multiplicative attention.
- Large embeddings with 2048 dimensions achieved the best results, but only by a small margin.
- A well-tuned beam search with length penalty is crucial. Beam widths of 5 to 10 together with a length penalty of 1.0 seemed to work well.

Massive Exploration of Neural Machine Translation Architectures, EMNLP 2017

Overview

- Natural Language Processing
  - Conversational Modeling (Best Paper Award at ICML Language Generation Workshop, EMNLP 2017)
  - Open-Source tf-seq2seq framework (4000+ stars, 1000+ forks), and exploration of NMT architectures (EMNLP 2017, 100+ citations)
- Deep Dive: ML for Systems
  - **Device Placement with Deep Reinforcement Learning (ICLR 2018)**

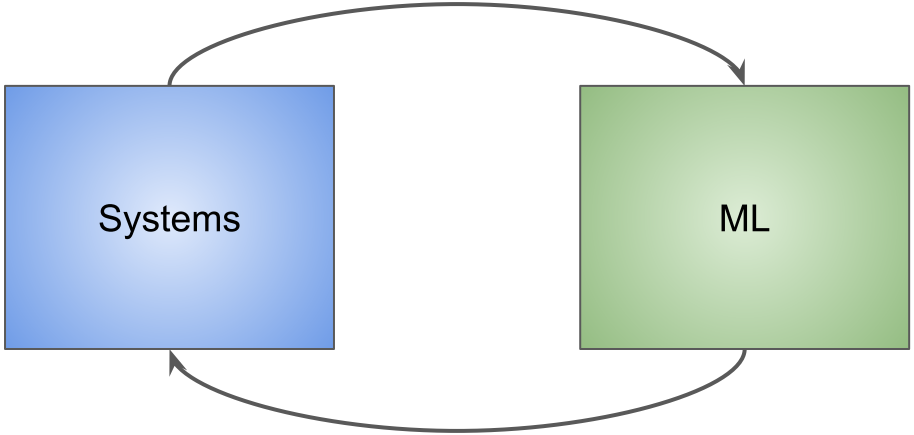
In the past decade, systems and hardware have transformed ML.



A Hierarchical Model For Device Placement, ICLR 2018

In the past decade, systems and hardware have transformed ML.

Now, it’s time for ML to transform systems.



A Hierarchical Model For Device Placement, ICLR 2018

Problems in computer systems

Design

- Computer architecture exploration
  - Architectural specification tuning
  - MatMul tiling optimization
- ML engines like TensorFlow
- Chip design
  - Verification
  - Logic synthesis
  - Placement
  - Manufacturing

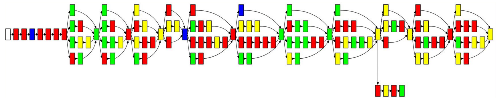
Operation

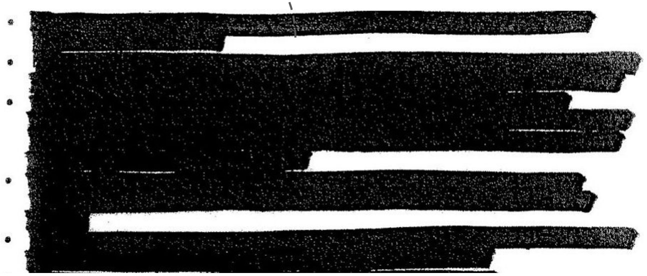
- Resource allocation
  - Model parallelism (e.g. TPU Pods)
  - Compiler register allocation
- Resource provisioning
  - Network demand forecasting
  - Memory forecasting
- Scheduling
  - TensorFlow op scheduling
  - Compiler instruction scheduling

A Hierarchical Model For Device Placement, ICLR 2018

ML for Systems Brain Moonshot

Device Placement





A Hierarchical Model For Device Placement, ICLR 2018

Hierarchical Learning for Device Placement

Azalia Mirhoseini*, Anna Goldie*, Hieu Pham, Benoit Steiner, Quoc V. Le, Jeff Dean

(*): Equal contribution

SUMMARY

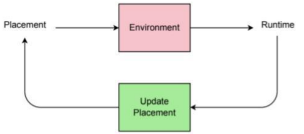
We propose a Reinforcement Learning algorithm that learns to automatically design model parallelism for TensorFlow graphs.

PROBLEM

- Given:
  - TensorFlow computational graph G with N ops
  - List of computing devices D (GPUs, CPUs, etc.)
- Find:
  - Placement $P = \{p_1, p_2, ..., p_N\}$, with $p_i \in D$
  - Minimizes the running time of G

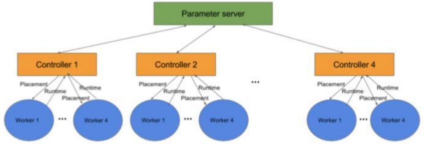
A REINFORCEMENT LEARNING APPROACH

- Using policy gradient to learn a policy π that:
  - Proposes placement and then measures runtime
  - Minimizes expected runtime $J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d]$



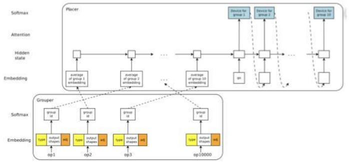
DISTRIBUTED TRAINING

- N controllers share a parameter server.
- Each controller sends placements to its children.
- Each child executes its placement.
- Each controller receives runtimes and updates the policy asynchronously.



MODEL

A two-level hierarchical network, consisting of a Grouper (which partitions the graph into groups) and a Placer (which places those groups onto devices)



TRAINING WITH REINFORCE

The goal is to minimize the expectation of runtime:

$$J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d] = \sum_{g \sim \pi_g} \sum_{d \sim \pi_d} p(g;\theta_g) p(d|g;\theta_d) R_d$$

$$\nabla_{\theta g} J(\theta_g, \theta_d) = \sum_{g \sim \pi_g} \nabla_{\theta g} p(g;\theta_g) \sum_{d \sim \pi_d} p(d|g;\theta_d) R_d$$

$$\approx \frac{1}{m} \sum_{g_i \sim \pi_g}^{1 \le i \le m} \nabla_{\theta g} \log p(g_i;\theta_g) . \frac{1}{k} (\sum_{d_j \sim \pi_d}^{1 \le j \le k} R_{d_j})$$

$$\nabla_{\theta d} J(\theta_g, \theta_d) = \sum_{d \sim \pi_d} \sum_{g \sim \pi_g} p(g;\theta_g) \nabla_{\theta d} p(d|g;\theta_d) R_d$$

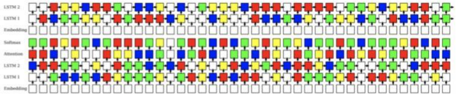
$$\approx \frac{1}{k} \sum_{d_j \sim \pi_d}^{1 \le j \le k} \frac{1}{m} (\sum_{g_i \sim \pi_g}^{1 \le i \le m} \nabla_{\theta d} \log p(d_j|g_i;\theta_d) R_{d_j})$$

RESULTS

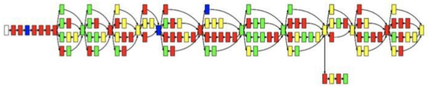
Tasks	CPU Only	GPU Only	#GPUs	Human Expert	Scotch	MinCut	Hierarchical Planner	Runtime Reduction
Inception-V3	0.61	0.15	2	0.15	0.93	0.82	**0.13**	16.3%
ResNet	-	1.18	2	1.18	6.27	2.92	**1.18**	0%
RNNLM	6.89	1.57	2	1.57	5.62	5.21	**1.57**	0%
NMT (2-layer)	6.46	OOM	2	2.13	3.21	5.34	**0.84**	60.6%
NMT (4-layer)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

EXAMPLE PLACEMENTS

- Each color is a GPU; transparent is the CPU.
- Neural Machine Translation with 2 layers

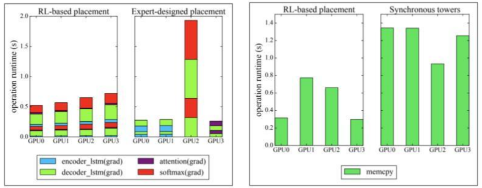


- Inception-V3



UNDERSTANDING THE PLACEMENTS

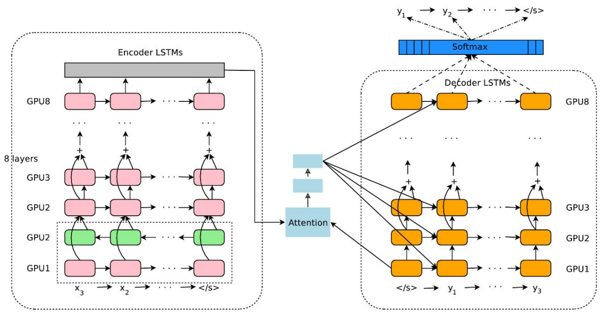
- Our method learns to optimize for different objectives for different models.
  - For RNNLM: learns that it is best to put all ops on a single GPU.
  - For NMT: learns to balance computation across devices.
  - For Inception-V3: learns to mitigate the time spent on inter-device memory copy.



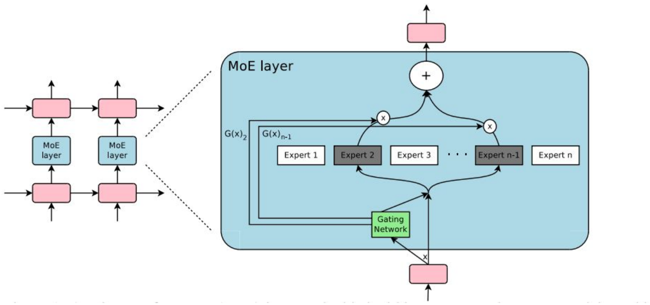
On the left, we show the computational load profiling of NMT model for RL-based and expert-designed placements. Smaller blocks of each color correspond to forward pass and same-color upper blocks correspond to back-propagation. On the right, we show memory copy time profiling. All memory copy activities in Synchronous tower are between a GPU and a CPU, which are in general slower than GPU copies that take place in the RL-based placement.

What is device placement and why is it important?

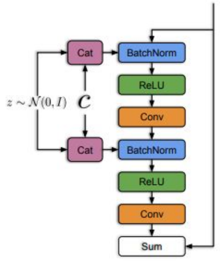
Trend towards many-device training, bigger models, larger batch sizes



Google neural machine translation’16
300 million parameters,
trained on 128 GPUs



Sparsely gated mixture of experts’17
130 billion parameters,
trained on 128 GPUs



BigGAN’18
355 million parameters,
trained on 512 TPU cores

A Hierarchical Model For Device Placement, ICLR 2018

Standard practice for device placement

- Often based on greedy heuristics
- Requires deep understanding of devices: nonlinear FLOPs, bandwidth, latency behavior
- Requires modeling parallelism and pipelining
- Does not generalize well

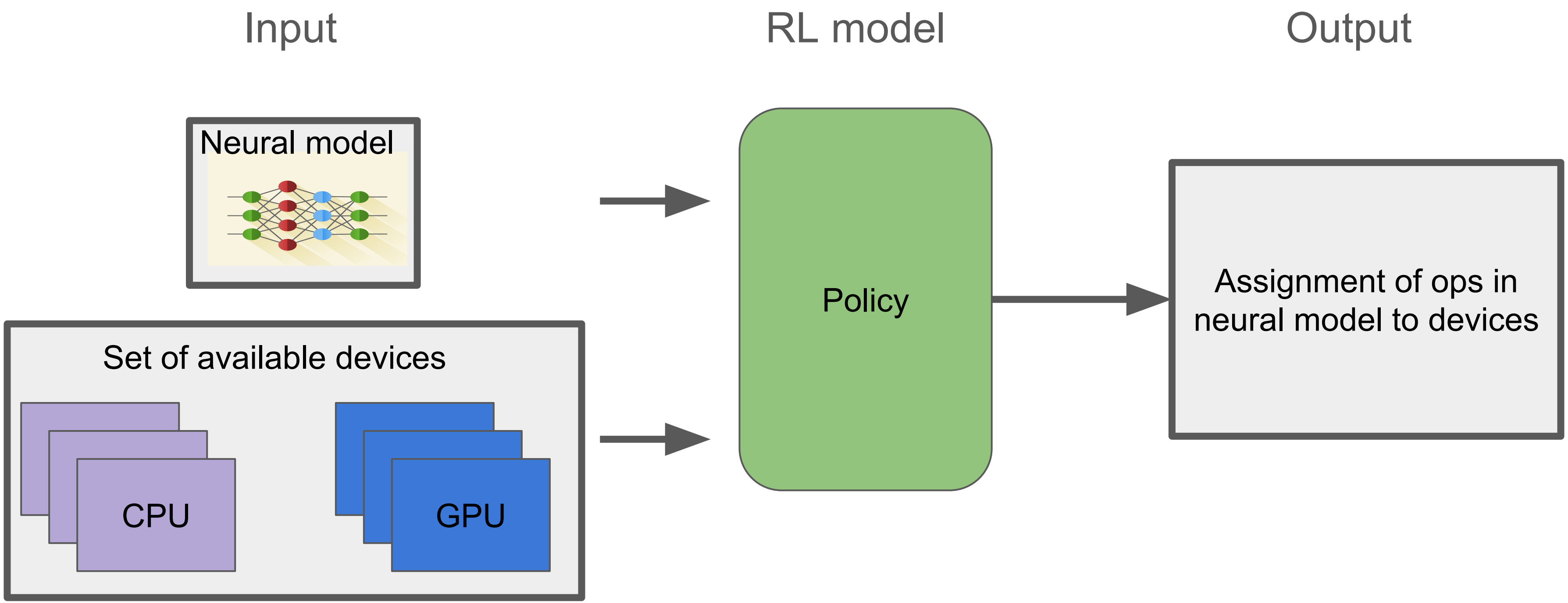
A Hierarchical Model For Device Placement, ICLR 2018

ML for device placement

- ML is repeatedly replacing rule based heuristics
- We show how RL can be applied to device placement
    - Effective search across large state and action spaces to find optimal solutions
    - Automated learning from underlying environment only based on reward function (e.g. runtime of a program)

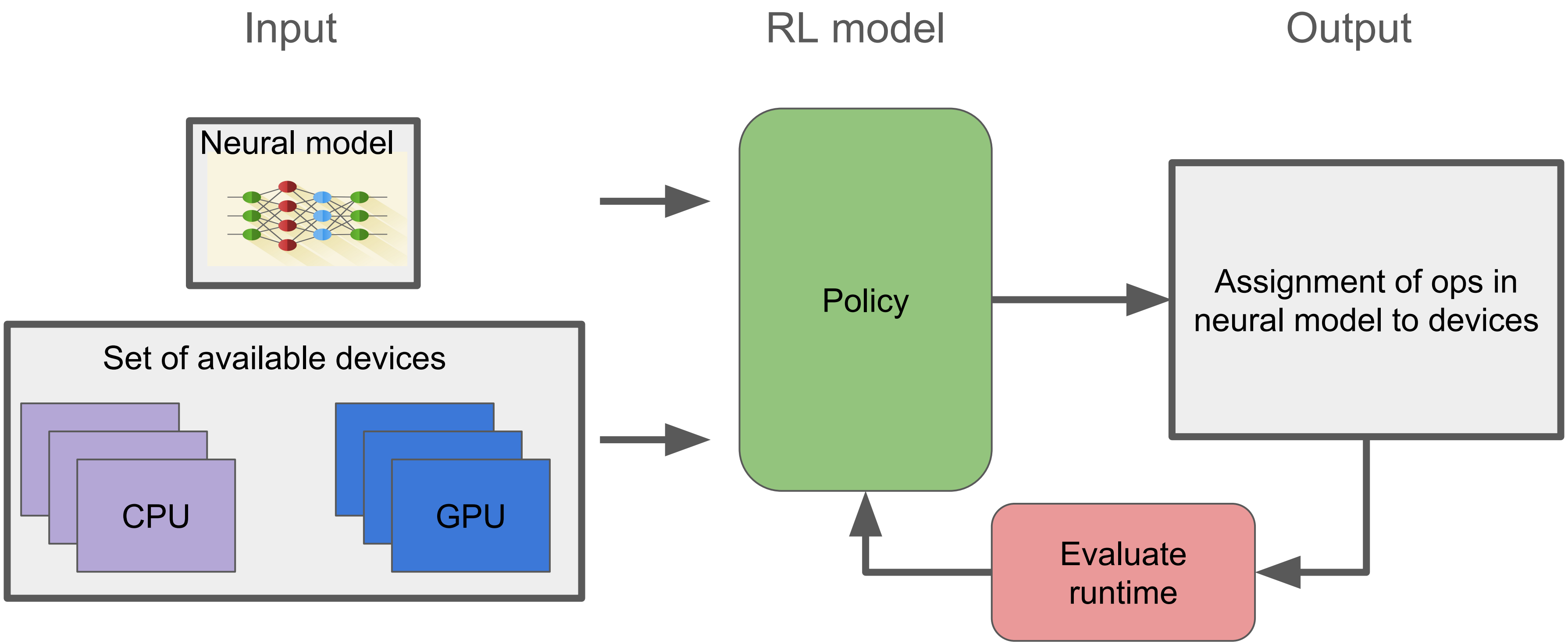
A Hierarchical Model For Device Placement, ICLR 2018

Posing device placement as an RL problem



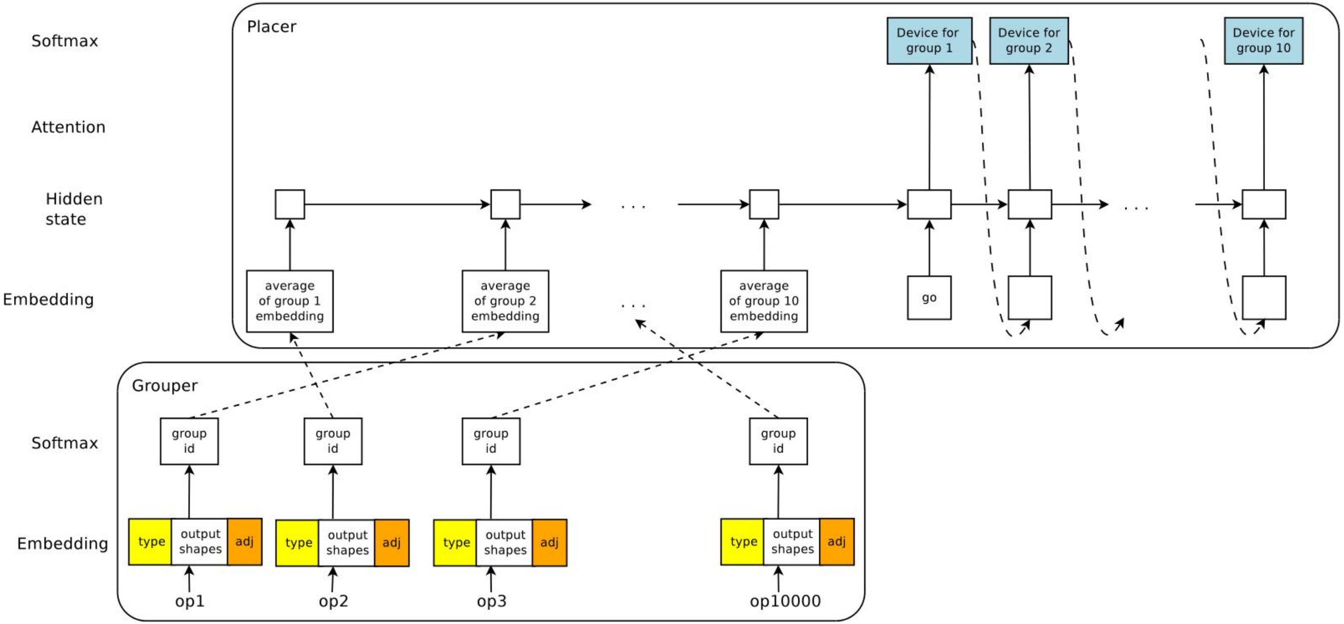
A Hierarchical Model For Device Placement, ICLR 2018

Posing device placement as an RL problem



A Hierarchical Model For Device Placement, ICLR 2018

An end-to-end hierarchical placement model



A Hierarchical Model For Device Placement, ICLR 2018

Training with REINFORCE

Objective: Minimize expected runtime for predicted placement d

$$J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d] = \sum_{g \sim \pi_g} \sum_{d \sim \pi_d} p(g;\theta_g) p(d|g;\theta_d) R_d$$

$J(\theta_g, \theta_d)$: expected runtime

θ_g: trainable parameters of Grouper

θ_d: trainable parameters of Placer

R_d: runtime for placement d

A Hierarchical Model For Device Placement, ICLR 2018

Training with REINFORCE

Objective: Minimize expected runtime for predicted placement d

$$J(\theta_g, \theta_d) = \boxed{\mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d]} = \sum_{g\sim\pi_g} \sum_{d\sim\pi_d} p(g;\theta_g)p(d|g;\theta_d)R_d$$

$J(\theta_g, \theta_d)$: expected runtime

θ_g: trainable parameters of Grouper

θ_d: trainable parameters of Placer

R_d: runtime for placement d

A Hierarchical Model For Device Placement, ICLR 2018

Training with REINFORCE

$$J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d] = \sum_{g\sim\pi_g}\sum_{d\sim\pi_d}\boxed{p(g;\theta_g)}p(d|g;\theta_d)R_d$$

Probability of predicted group assignment of operations

A Hierarchical Model For Device Placement, ICLR 2018

Training with REINFORCE

$$J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d] = \sum_{g \sim \pi_g} \sum_{d \sim \pi_d} p(g;\theta_g)\boxed{p(d|g;\theta_d)}R_d$$

Probability of predicted device placement
conditioned on grouping results

A Hierarchical Model For Device Placement, ICLR 2018

Gradient update for Grouper

$$J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d] = \sum_{g\sim\pi_g} \sum_{d\sim\pi_d} p(g;\theta_g)p(d|g;\theta_d)R_d$$

$$\boxed{\nabla_{\theta g} J(\theta_g, \theta_d)} = \sum_{g\sim\pi_g} \nabla_{\theta g} p(g;\theta_g) \sum_{d\sim\pi_d} p(d|g;\theta_d)R_d$$

$$\approx \frac{1}{m} \sum_{g_i\sim\pi_g}^{1\leq i\leq m} \nabla_{\theta g} \log p(g_i;\theta_g) . \frac{1}{k} (\sum_{d_j\sim\pi_d}^{1\leq j\leq k} R_{d_j})$$

Derivative w.r.t. parameters of Grouper

A Hierarchical Model For Device Placement, ICLR 2018

Gradient update for Placer

$$J(\theta_g, \theta_d) = \mathbf{E}_{\mathbf{P}(\mathbf{d};\theta_\mathbf{g},\theta_\mathbf{d})}[R_d] = \sum_{g\sim\pi_g}\sum_{d\sim\pi_d} p(g;\theta_g)p(d|g;\theta_d)R_d$$

$$\boxed{\nabla_{\theta d} J(\theta_g, \theta_d)} = \sum_{d\sim\pi_d}\sum_{g\sim\pi_g} p(g;\theta_g)\nabla_{\theta d} p(d|g;\theta_d)R_d$$

$$\approx \frac{1}{k}\sum_{d_j\sim\pi_d}^{1\leq j\leq k}\frac{1}{m}\left(\sum_{g_i\sim\pi_g}^{1\leq i\leq m}\nabla_{\theta d}\log p(d_j|g_i;\theta_d)R_{d_j}\right)$$

Derivative w.r.t. parameters of Placer

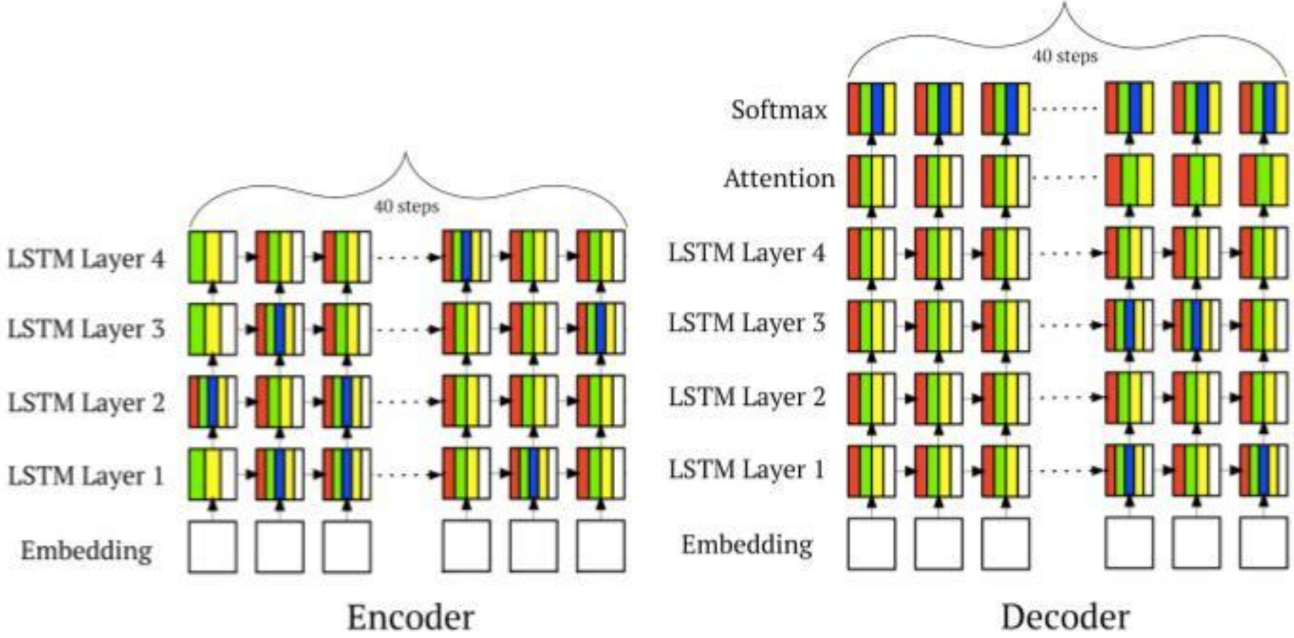
A Hierarchical Model For Device Placement, ICLR 2018

Results (runtime in seconds)

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%

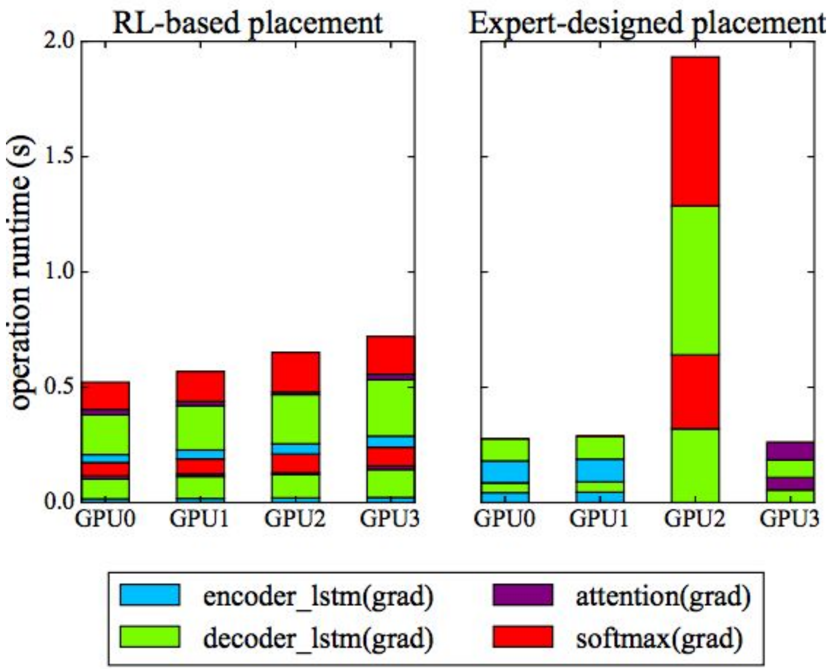
A Hierarchical Model For Device Placement, ICLR 2018

Learned placements on NMT



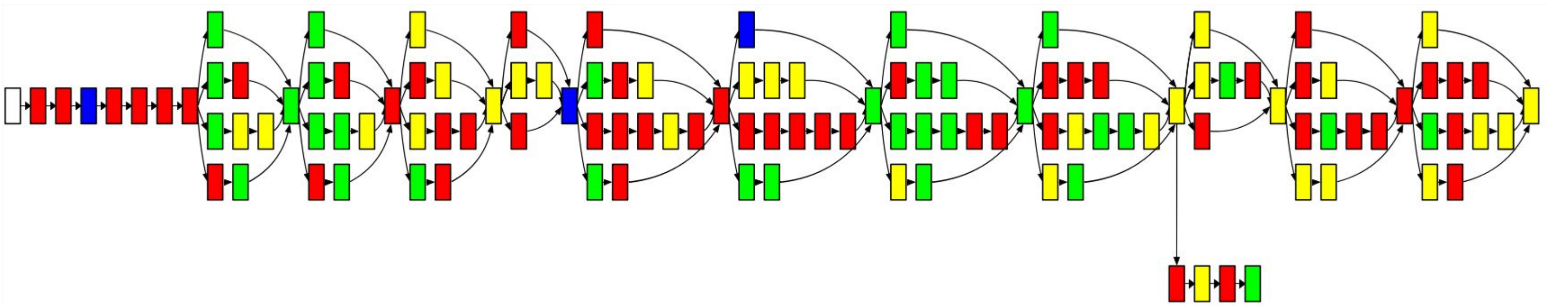
A Hierarchical Model For Device Placement, ICLR 2018

Profiling placement on NMT



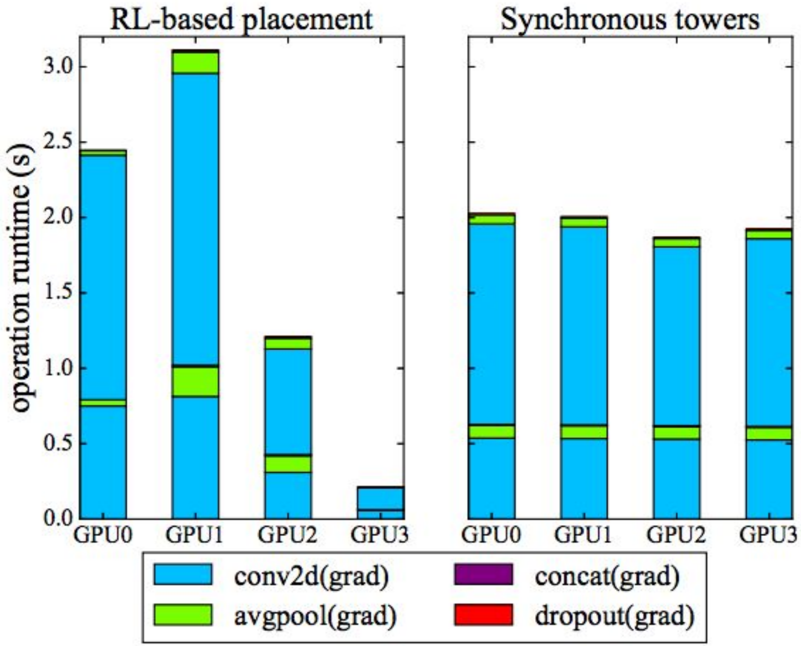
A Hierarchical Model For Device Placement, ICLR 2018

Learned placement on Inception-V3



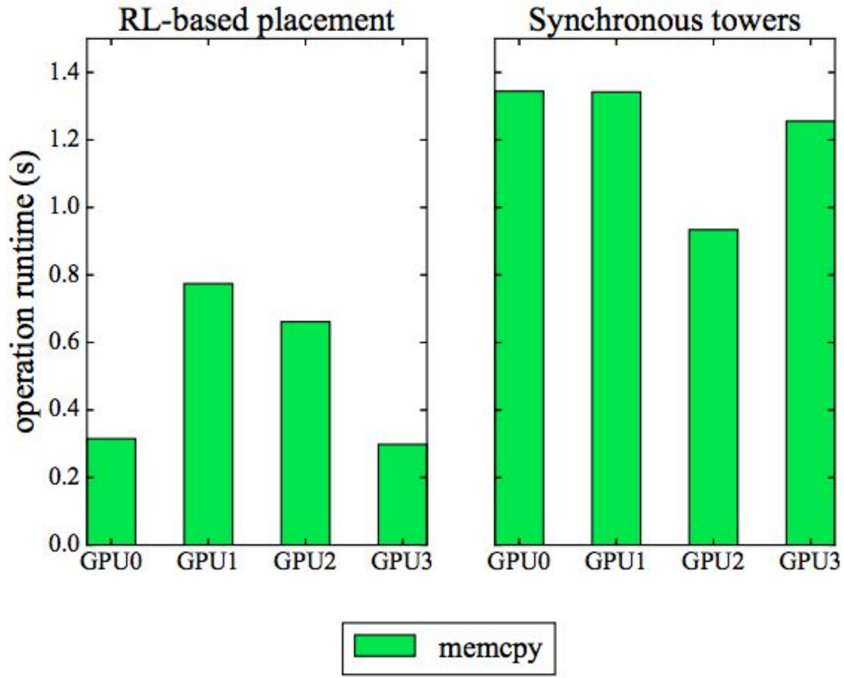
A Hierarchical Model For Device Placement, ICLR 2018

Profiling placement on Inception-V3



A Hierarchical Model For Device Placement, ICLR 2018

Profiling placement on Inception-V3



A Hierarchical Model For Device Placement, ICLR 2018

Overview

- Natural Language Processing
  - Conversational Modeling (Best Paper Award at ICML Language Generation Workshop, EMNLP 2017)
  - Open-Source tf-seq2seq framework (4000+ stars, 1000+ forks), and exploration of NMT architectures (EMNLP 2017, 100+ citations)
- Deep Dive: ML for Systems
  - Device Placement with Deep Reinforcement Learning (ICLR 2018)

Questions?