Google’s Neural Machine Translation System

Bridging the Gap between Human and Machine Translation

Presented by Anthony Alvarez and GwonJae Cho

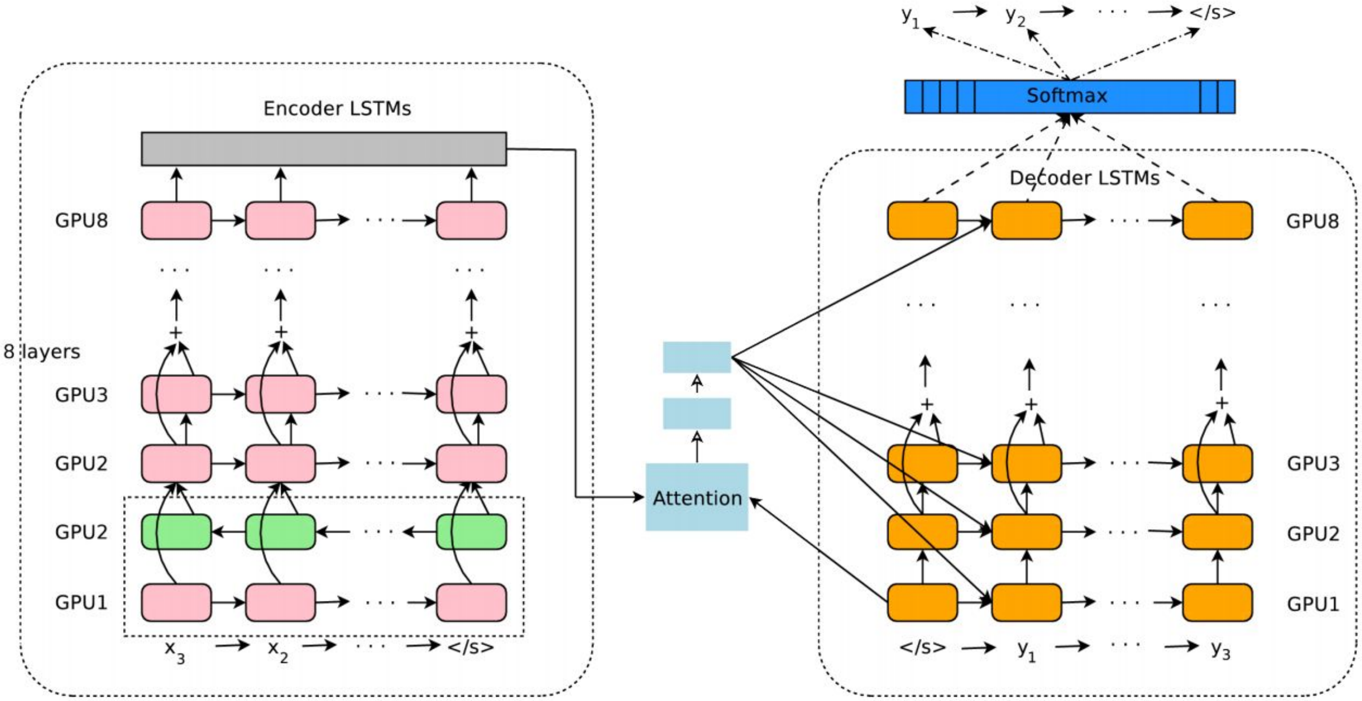
Introduction

- Neural Machine Translation
  - Ability to learn directly, end-to-end fashion
  - Consists of two recurrent neural networks and often accompanied by an attention mechanism
  - Worse in accuracy when training large-scale datasets
    - Slower training and inference speed
    - Ineffectiveness in dealing with rare words
    - Sentence coverage
- In Google's Neural Machine Translation,
  - Used LSTM RNN with residual connections between layers
  - Connected attention from the bottom layer of the decoder to the top layer of the encoder
  - Low precision arithmetic for inference
  - Used sub-word units

Related Work

- Prior to NMT, Statistical Machine Translation was dominant paradigm with some success
- Attention mechanism to deal with rare words, a character encoder, a character decoder, sentence level loss minimization
- However, systematic comparison with large scale, production quality phrase-based translation systems has been lacking.

Model Architecture



Model Architecture

$$\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_M} = EncoderRNN(x_1, x_2, x_3, ..., x_M) \quad (1)$$

$$\begin{aligned} P(Y|X) &= P(Y|\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ..., \mathbf{x_M}) \\ &= \prod_{i=1}^{N} P(y_i|y_0, y_1, y_2, ..., y_{i-1}; \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ..., \mathbf{x_M}) \end{aligned} \quad (2)$$

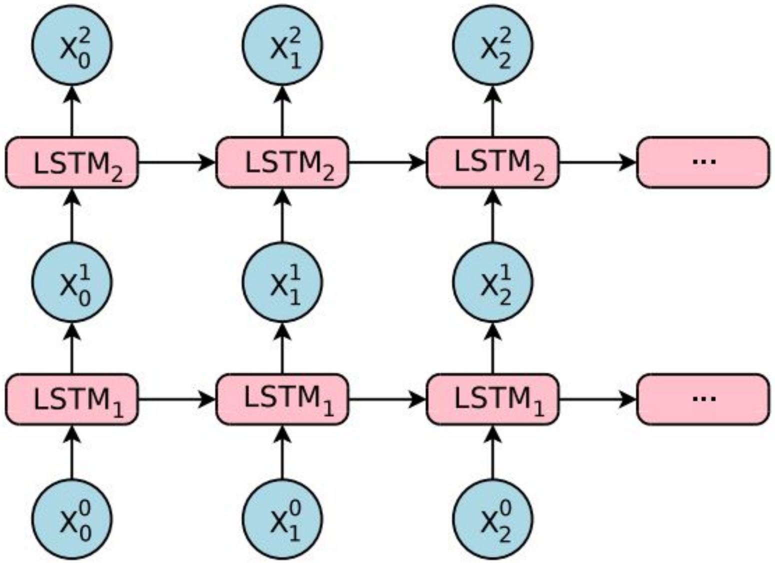
- Decoder : RNN + softmax layer
- Attention

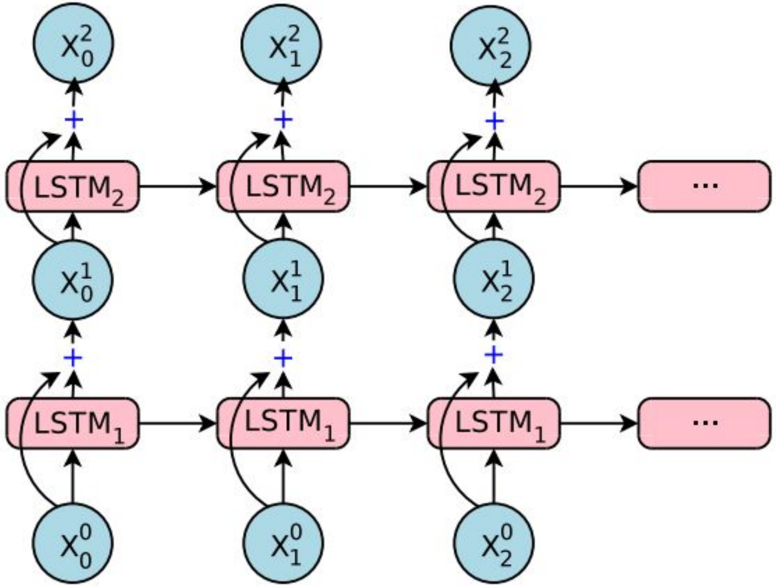
$$s_t = AttentionFunction(\mathbf{y}_{i-1}, \mathbf{x}_t) \quad \forall t, \quad 1 \le t \le M$$

$$p_t = \exp(s_t) / \sum_{t=1}^{M} \exp(s_t) \qquad \forall t, \quad 1 \le t \le M$$

$$\mathbf{a}_i = \sum_{t=1}^{M} p_t.\mathbf{x}_t$$

Residual Connections





Residual Connections

$$\mathbf{c}_t^i, \mathbf{m}_t^i = \mathrm{LSTM}_i(\mathbf{c}_{t-1}^i, \mathbf{m}_{t-1}^i, \mathbf{x}_t^{i-1}; \mathbf{W}^i)$$
$$\mathbf{x}_t^i = \mathbf{m}_t^i$$
$$\mathbf{c}_t^{i+1}, \mathbf{m}_t^{i+1} = \mathrm{LSTM}_{i+1}(\mathbf{c}_{t-1}^{i+1}, \mathbf{m}_{t-1}^{i+1}, \mathbf{x}_t^i; \mathbf{W}^{i+1})$$

$$\downarrow$$

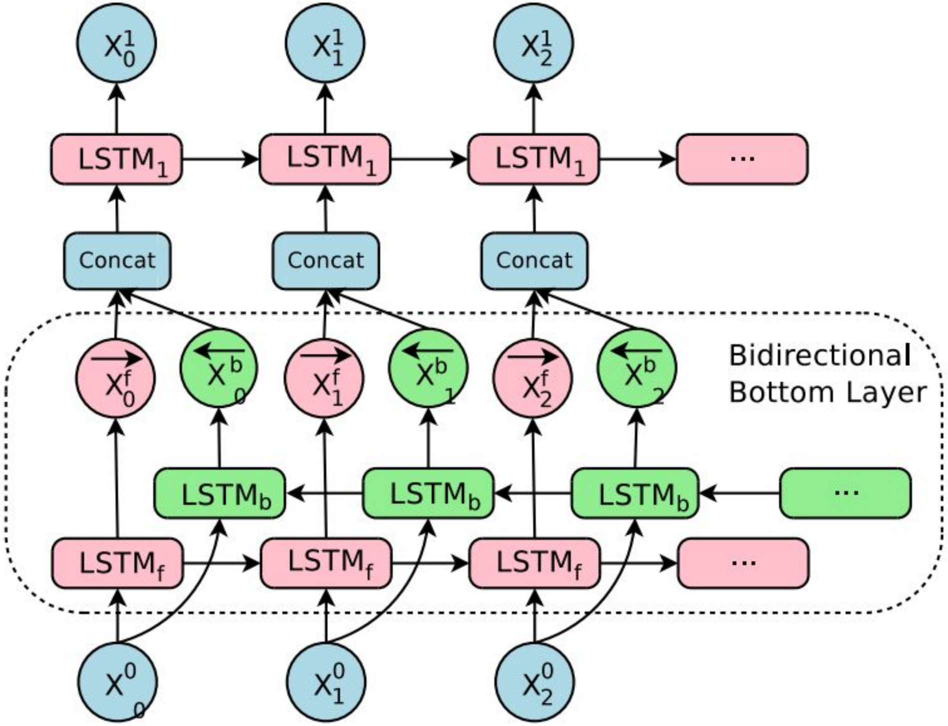
$$\mathbf{c}_t^i, \mathbf{m}_t^i = \mathrm{LSTM}_i(\mathbf{c}_{t-1}^i, \mathbf{m}_{t-1}^i, \mathbf{x}_t^{i-1}; \mathbf{W}^i)$$
$$\mathbf{x}_t^i = \mathbf{m}_t^i + \mathbf{x}_t^{i-1}$$
$$\mathbf{c}_t^{i+1}, \mathbf{m}_t^{i+1} = \mathrm{LSTM}_{i+1}(\mathbf{c}_{t-1}^{i+1}, \mathbf{m}_{t-1}^{i+1}, \mathbf{x}_t^i; \mathbf{W}^{i+1})$$

Result : Improve the gradient flow

Bidirectional First layer

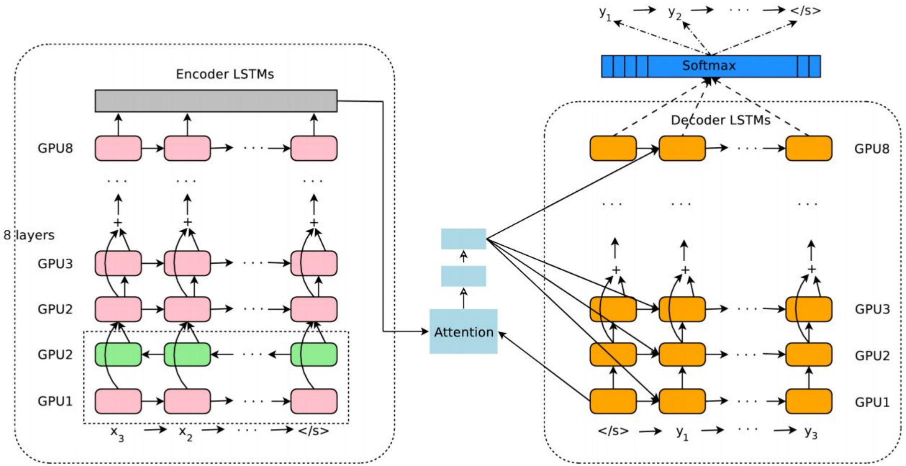
- The information required to translate certain words on the output side can appear anywhere on the source side.
- Depending on the language pair, the information for a particular output word can be distributed
- Bidirectional RNN for the encoder

Bidirectional First layer



Model Parallelism

- Data Parallelism
  - Train n model replicas concurrently using a Downpour SGD algorithm
  - n replicas all share one copy of model parameters
- Model Parallelism
  - The encoder and decoder networks are partitioned along the depth dimension and are placed on multiple GPUs



Segmentation Approaches

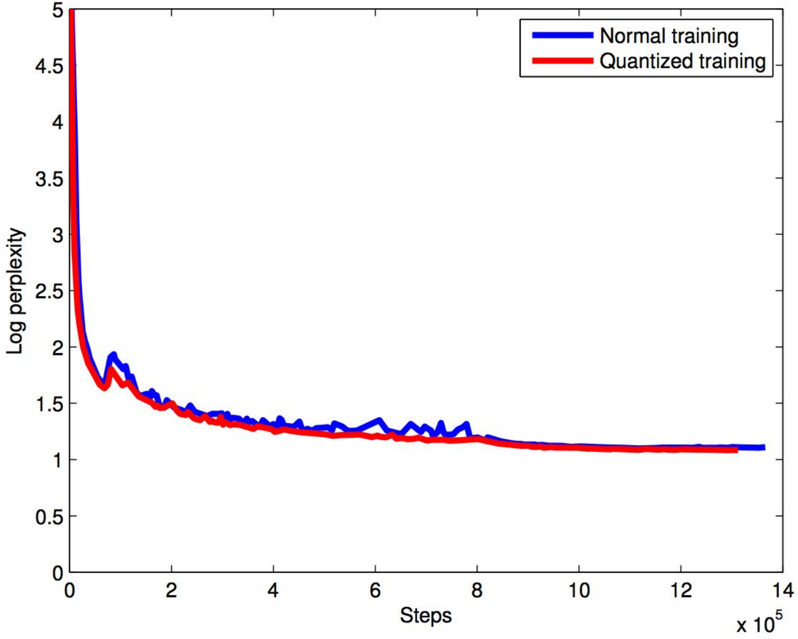
- Wordpiece(Sub-word Units)
    1. break words into wordpieces given a trained wordpiece model
    2. produces a wordpiece sequence, which is then converted into the corresponding word sequence.

Word: Jet makers feud over seat width with big orders at stake

wordpieces: _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

Quantizable Model and Quantized Inference

Speed up network by reducing accuracy



$$\mathbf{s}_i = \max(\text{abs}(\mathbf{W}[i,:]))$$

$$\mathbf{WQ}[i,j] = \text{round}(\mathbf{W}[i,j]/\mathbf{s}_i \times 127.0)$$

	BLEU	Log Perplexity	Decoding time (s)
CPU	31.20	1.4553	1322
GPU	31.20	1.4553	3028
TPU	31.21	1.4626	384

Decoder

Few new features to speed decoding

- Length normalization lp() helps avoid penalizing long sentences
- $p_{i,j}$ is the attention probabiliyt of the target word y_j on the source word x_i
- At each step only consider tokens that have local scores close to best token for that step
- Limit number of hypotheses to 8-12
- After each batch eliminate hypothesis more than 'beamsize' worse than best hypothesis

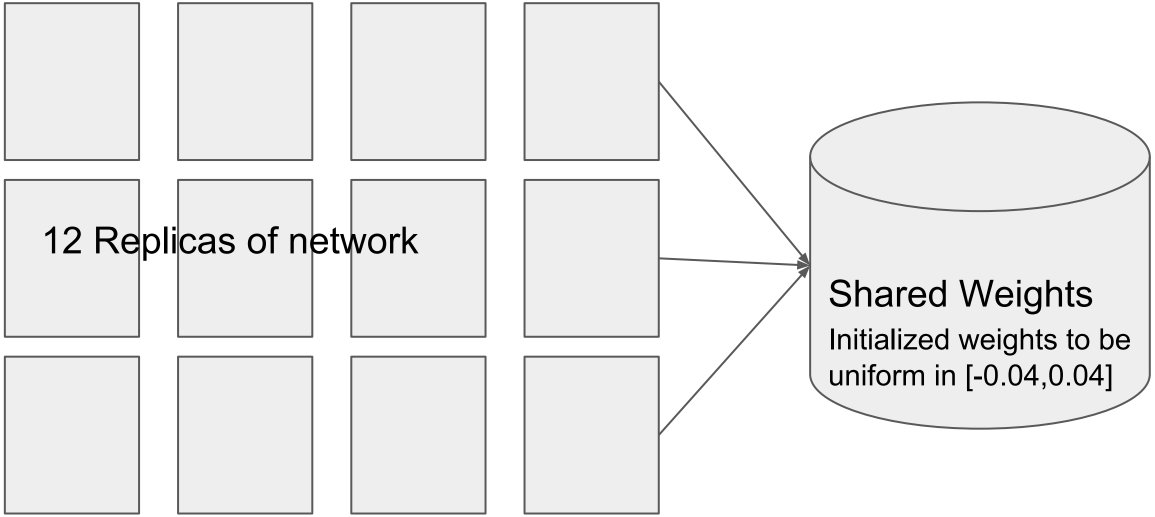
BLEU		α					
		0.0	0.2	0.4	0.6	0.8	1.0
β	0.0	30.3	30.7	30.9	31.1	31.2	31.1
	0.2	31.4	31.4	31.4	31.3	30.8	30.3
	0.4	31.4	31.4	31.4	31.1	30.5	29.6
	0.6	31.4	31.4	31.3	30.9	30.1	28.9
	0.8	31.4	31.4	31.2	30.8	29.8	28.1
	1.0	31.4	31.3	31.2	30.6	29.4	27.2

$$s(Y,X) = \log(P(Y|X))/lp(Y) + cp(X;Y)$$

$$lp(Y) = \frac{(5+|Y|)^\alpha}{(5+1)^\alpha}$$

$$cp(X;Y) = \beta * \sum_{i=1}^{|X|} \log(\min(\sum_{j=1}^{|Y|} p_{i,j}, 1.0)),$$

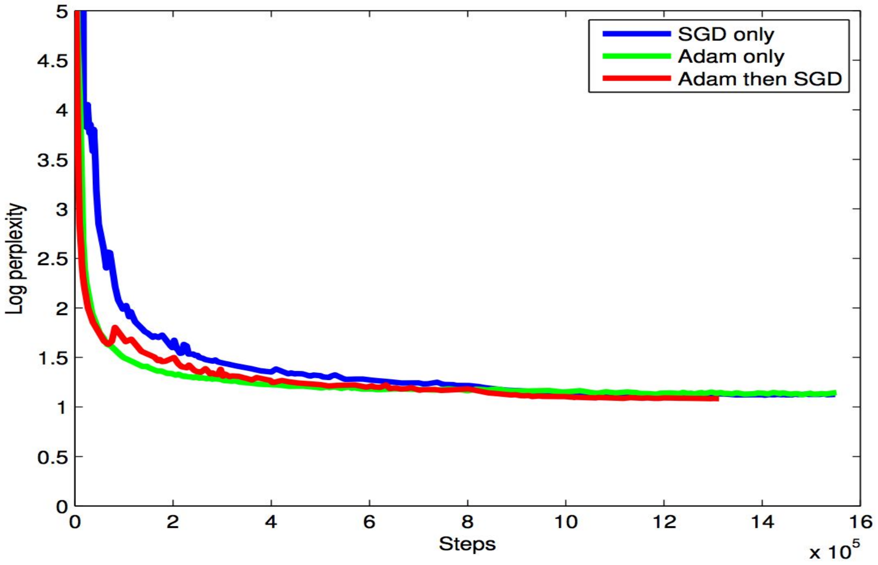
Training Procedure



All gradients are trimmed to be less than 5
Drop out in training prevents overfitting; Dropout set to between 0.2 and 0.3

Results after ML training

- Learning rate is set to be high for first 1.2 million steps then gradually brought down over next 800k steps
- Once ML alone has converged its is further optimized using reinforcement learning.
- On large Google proprietary datasets dropout is not used.



More ML and RL results

Table 4: Single model results on WMT En→Fr (newstest2014)

Model	BLEU	CPU decoding time per sentence (s)
Word	37.90	0.2226
Character	38.01	1.0530
WPM-8K	38.27	0.1919
WPM-16K	37.60	0.1874
WPM-32K	38.95	0.2118
Mixed Word/Character	38.39	0.2774
PBMT [15]	37.0	
LSTM (6 layers) [31]	31.5	
LSTM (6 layers + PosUnk) [31]	33.1	
Deep-Att [45]	37.7	
Deep-Att + PosUnk [45]	39.2	

Table 6: Single model test BLEU scores, averaged over 8 runs, on WMT En→Fr and En→De

Dataset	Trained with log-likelihood	Refined with RL
En→Fr	38.95	39.92
En→De	24.67	24.60

Best models vs Human Evaluation

- Ensemble models using best networks show that RL improves BLEU
- Humans seem to be unable to distinguish ML and ML+RL methods
- Human data set was only 500 side by side examples so not definitive dataset.

Model	BLEU
WPM-32K (8 models)	40.35
RL-refined WPM-32K (8 models)	41.16
LSTM (6 layers) [31]	35.6
LSTM (6 layers + PosUnk) [31]	37.5
Deep-Att + PosUnk (8 models) [45]	40.4

Model	BLEU	Side-by-side averaged score
PBMT [15]	37.0	3.87
NMT before RL	40.35	4.46
NMT after RL	41.16	4.44
Human		4.82

Improvement on Production Google Data

Table 10: Mean of side-by-side scores on production data

	PBMT	GNMT	Human	Relative Improvement
English → Spanish	4.885	5.428	5.504	87%
English → French	4.932	5.295	5.496	64%
English → Chinese	4.035	4.594	4.987	58%
Spanish → English	4.872	5.187	5.372	63%
French → English	5.046	5.343	5.404	83%
Chinese → English	3.694	4.263	4.636	60%

Improvement on Production Google Data