Google AI

Sharp Students - Dull Teachers

Tricks of the Trade for Neural Machine Translation

Orhan Firat
Research Scientist
Mountain View - CA

Sept 4th, 2018

MTM'18 Prague

Let’s read some Theano

[US] | https://github.com/lisa-groundhog/GroundHog/blob/master/groundhog/trainer/SGD_adadelta.py#L114

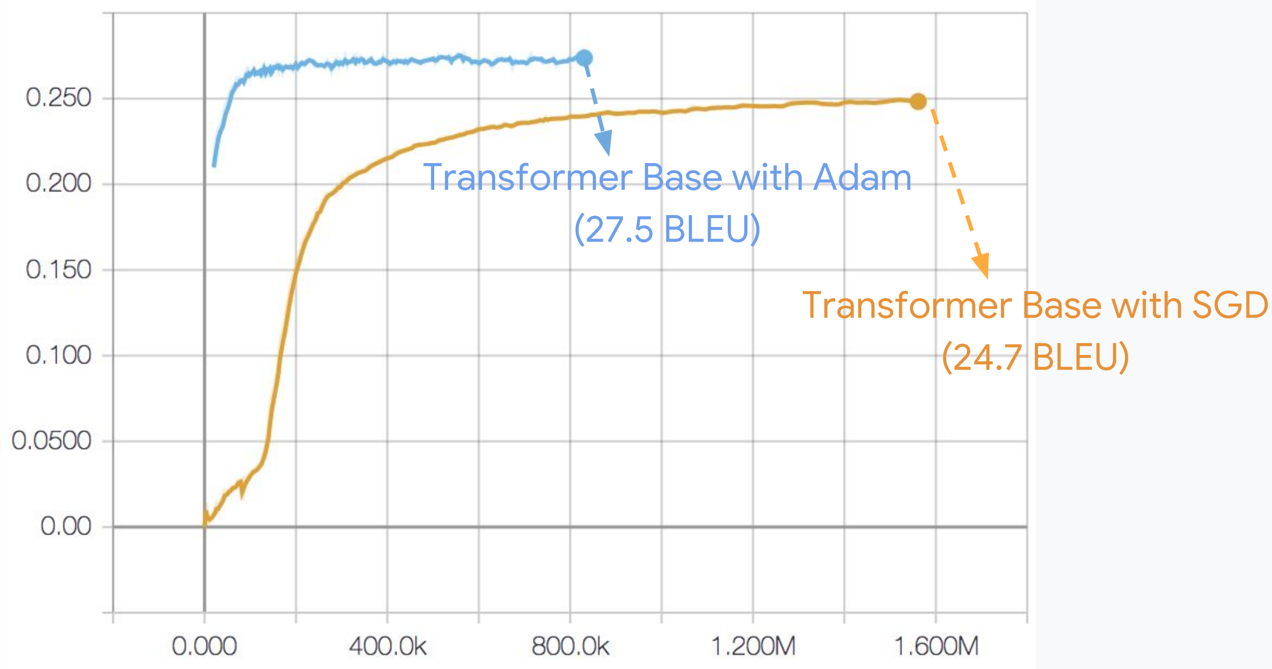
```python

        norm_gs = TT.sqrt(sum(TT.sum(x**2)
            for x,p in zip(gs, self.model.params) if p not in self.model.exclude_params_for_norm))
        if 'cutoff' in state and state['cutoff'] > 0:
            c = numpy.float32(state['cutoff'])
            if state['cutoff_rescale_length']:
                c = c * TT.cast(loc_data[0].shape[0], 'float32')

            notfinite = TT.or_(TT.isnan(norm_gs), TT.isinf(norm_gs))
            _gs = []
            for g,p in zip(gs,self.model.params):
                if p not in self.model.exclude_params_for_norm:
                    tmpg = TT.switch(TT.ge(norm_gs, c), g*c/norm_gs, g)
                    _gs.append(
                        TT.switch(notfinite, numpy.float32(.1)*p, tmpg))
                else:
                    _gs.append(g)
            gs = _gs
        store_gs = [(s,g) for s,g in zip(self.gs, gs)]
        updates = store_gs + [(s[0], r) for s,r in zip(model.updates, rules)]

```

Shiniest Hammer

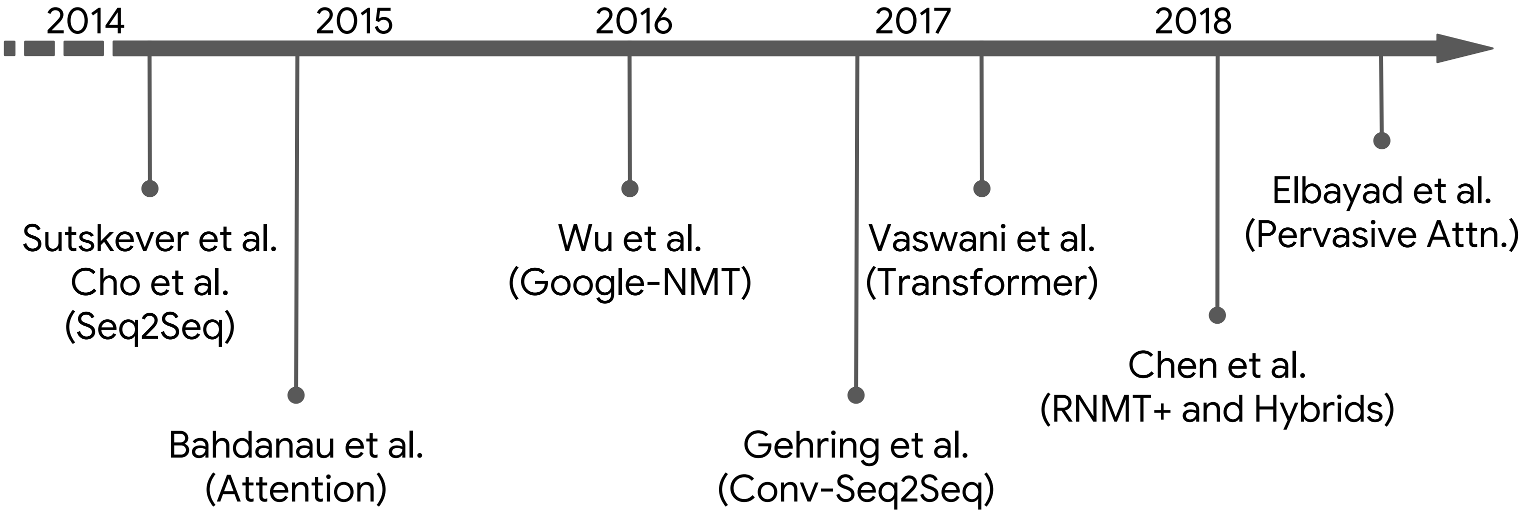


Conclusion

1. Each model comes with a set of additional techniques that might also be applicable to the others.
   (tricks of the trade for NMT)

2. Gains from training (optimization) might be larger than the other ingredients for improvement.
   (sharp students - dull teachers)

A Brief History of NMT Models



$$quality = f(X, \theta, \mu)$$

X : Data
θ : Model
μ : Hyperparameters

The Best of Both Worlds - I
(Chen et al. 2018)

Every new approach is:
- accompanied by a set of modeling and training techniques.

Goal:
1. Tease apart architectures and their accompanying techniques.
2. Identify key *modeling* and *training* techniques.
3. Apply them on RNN based Seq2Seq → **RNMT+**

Conclusion:
- **RNMT+** outperforms all previous three approaches.

The Best of Both Worlds - II

(Chen et al. 2018)

Also, each new approach has:

- a fundamental architecture (signature wiring of neural network).

Goal:

1. Analyse properties of each architecture.
2. Combine their strengths.
3. Devise new hybrid architectures → **Hybrids**

Conclusion:

- **Hybrids** obtain further improvements over all the others.

Building Blocks

- RNN Based NMT - **RNMT**
- Convolutional NMT - **ConvS2S**
- Conditional Transformation Based NMT - **Transformer**

GNMT - Wu et al.

- Core Components:
  - RNNs
  - Attention (Additive)
  - biLSTM + uniLSTM
  - Deep residuals
  - Async Training

- Pros:
  - De facto standard
  - Modelling state space

- Cons:
  - Temporal dependence
  - Not enough gradients

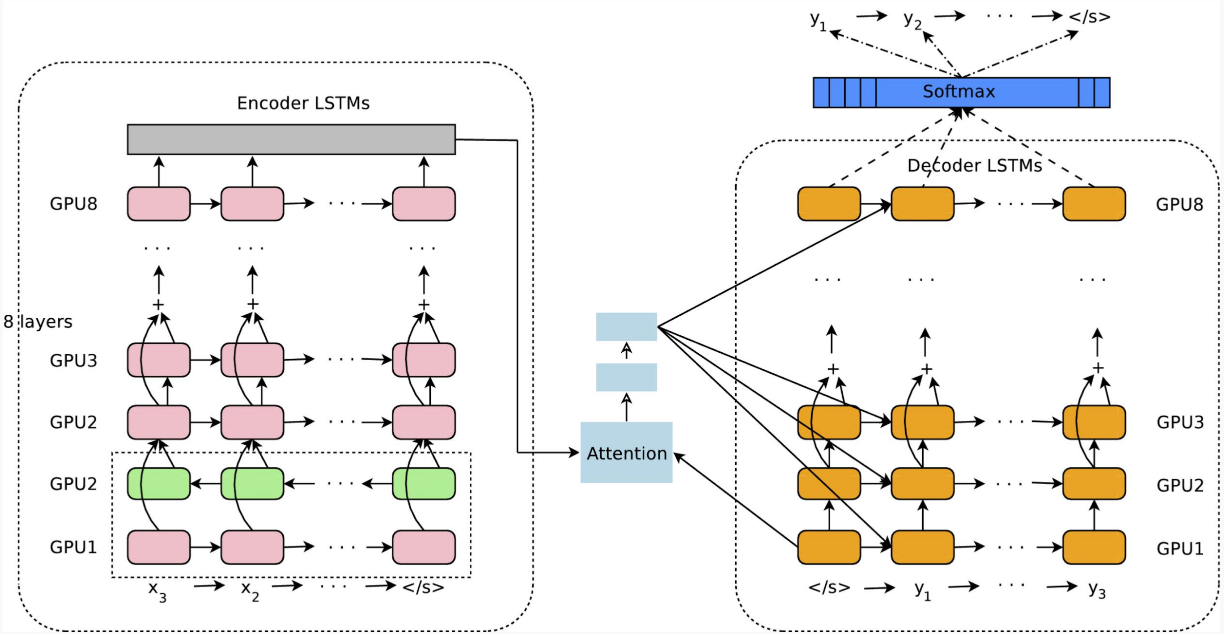
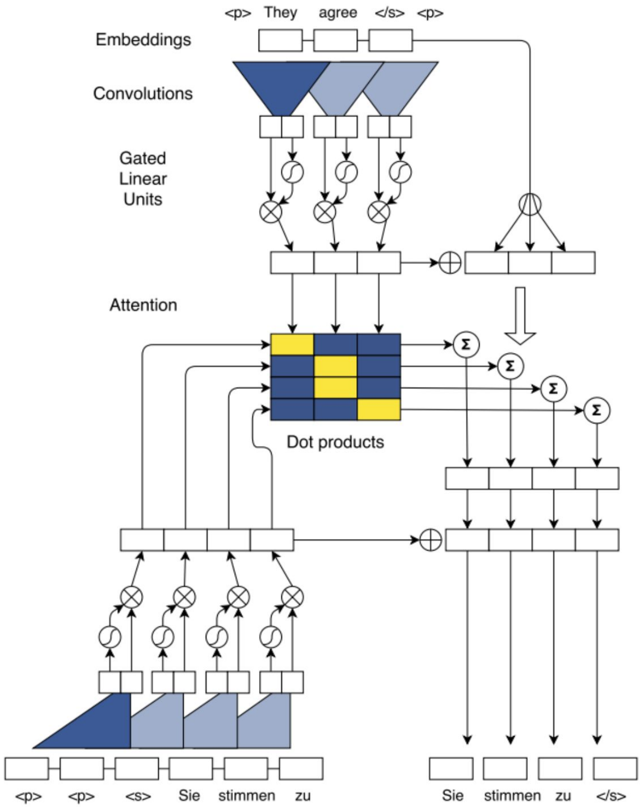


*Figure from "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation" Wu et al. 2016

ConvS2S - Gehring et al.



- Core Components:
  - Convolution - GLUs
  - Multi-hop attention
  - Positional embeddings
  - Careful initialization
  - Careful normalization
  - Sync Training
- Pros:
  - No temporal dependence
  - More interpretable than RNN
- Cons:
  - Need to stack more to increase the receptive field

*Figure from “Convolutional Sequence to Sequence Learning” Gehring et al. 2017

Transformer - Vaswani et al.

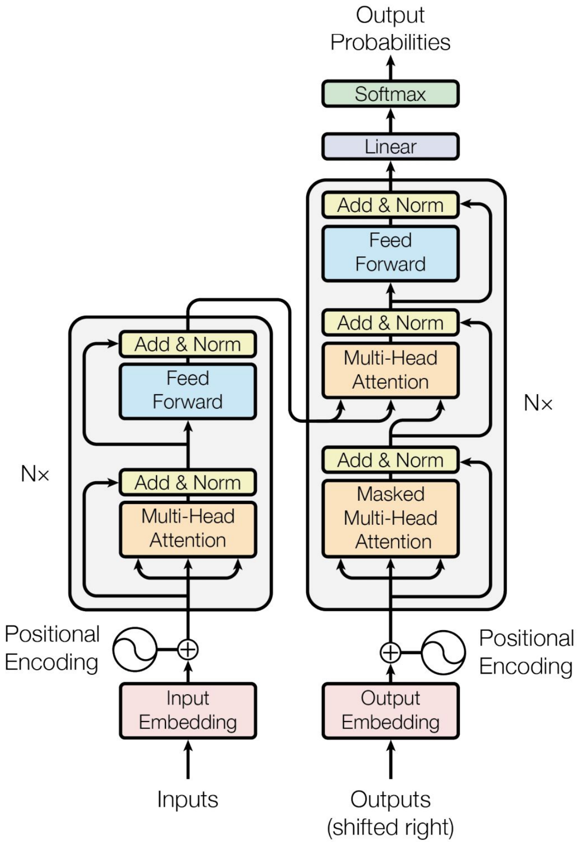


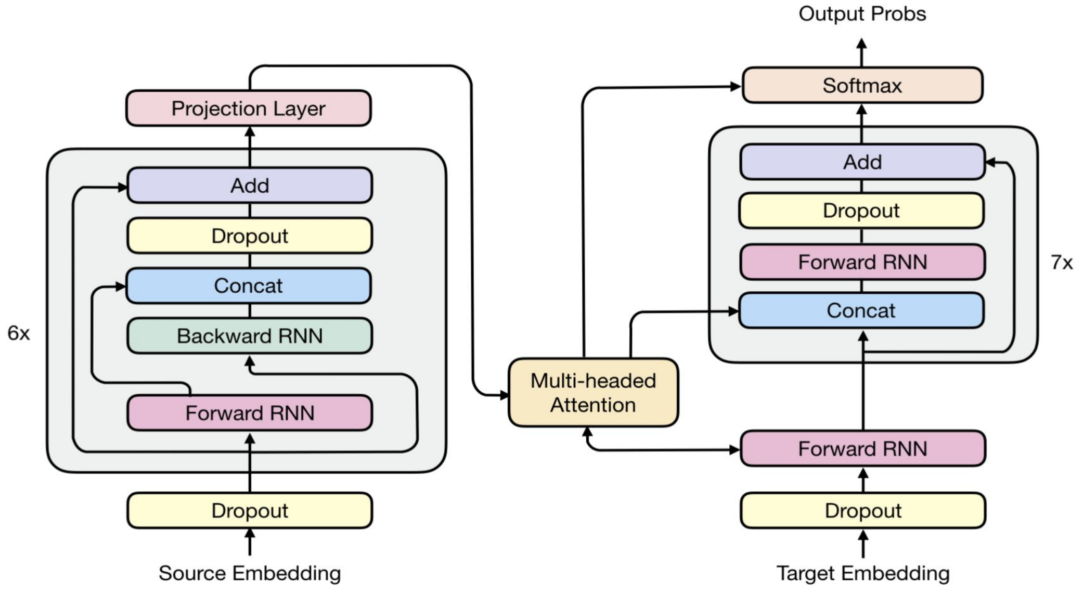
*Figure from "Attention is All You Need" Vaswani et al. 2017

- Core Components:
  - Self-Attention
  - Multi-headed attention
  - Layout: N→ f() → D→ R
  - Careful normalization
  - Careful batching
  - Sync training
  - Label Smoothing
  - Per-token loss
  - Learning rate schedule
  - Checkpoint Averaging

- Pros:
  - Gradients everywhere - faster optimization
  - Parallel encoding both training/inference

- Cons:
  - Combines many advances at once
  - Fragile

The Best of Both Worlds - I: RNMT+



- The Architecture:
  - Bi-directional encoder 6 x LSTM
  - Uni-directional decoder 8 x LSTM
  - Layer normalized LSTM cell
    - Per-gate normalization
  - Multi-head attention
    - 4 heads
    - Additive (Bahdanau) attention

Model Comparison - I : BLEU Scores

WMT'14 En-Fr
(35M sentence pairs)

Model	Test BLEU	Epochs	Training Time
GNMT	38.95	-	-
ConvS2S [7]	39.49 ± 0.11	62.2	438h
Trans. Base	39.43 ± 0.17	20.7	90h
Trans. Big [8]	40.73 ± 0.19	8.3	120h
RNMT+	41.00 ± 0.05	8.5	120h

WMT'14 En-De
(4.5M sentence pairs)

Model	Test BLEU	Epochs	Training Time
GNMT	24.67	-	-
ConvS2S	25.01 ± 0.17	38	20h
Trans. Base	27.26 ± 0.15	38	17h
Trans. Big	27.94 ± 0.18	26.9	48h
RNMT+	28.49 ± 0.05	24.6	40h

- RNMT+/ConvS2S: 32 GPUs, 4096 sentence pairs/batch.
- Transformer Base/Big: 16 GPUs, 65536 tokens/batch.

Model Comparison - II : Speed and Size

WMT'14 En-Fr
(35M sentence pairs)

Model	Test BLEU	Epochs	Training Time
GNMT	38.95	-	-
ConvS2S [7]	39.49 ± 0.11	62.2	438h
Trans. Base	39.43 ± 0.17	20.7	90h
Trans. Big [8]	40.73 ± 0.19	8.3	120h
RNMT+	41.00 ± 0.05	8.5	120h

Model	Examples/s	FLOPs	Params
ConvS2S	80	15.7B	263.4M
Trans. Base	160	6.2B	93.3M
Trans. Big	50	31.2B	375.4M
RNMT+	30	28.1B	378.9M

WMT'14 En-De
(4.5M sentence pairs)

Model	Test BLEU	Epochs	Training Time
GNMT	24.67	-	-
ConvS2S	25.01 ± 0.17	38	20h
Trans. Base	27.26 ± 0.15	38	17h
Trans. Big	27.94 ± 0.18	26.9	48h
RNMT+	28.49 ± 0.05	24.6	40h

- RNMT+/ConvS2S: 32 GPUs, 4096 sentence pairs/batch.
- Transformer Base/Big: 16 GPUs, 65536 tokens/batch.

“oh, it is just a better tuned model”

Well... no!

Stability: Ablations

WMT'14 En-Fr

Model	RNMT+	Trans. Big
Baseline	41.00	40.73
- Label Smoothing	40.33	40.49
- Multi-head Attention	40.44	39.83
- Layer Norm.	*	*
- Sync. Training	39.68	*

* Indicates an unstable training run

Evaluate importance of four key techniques:

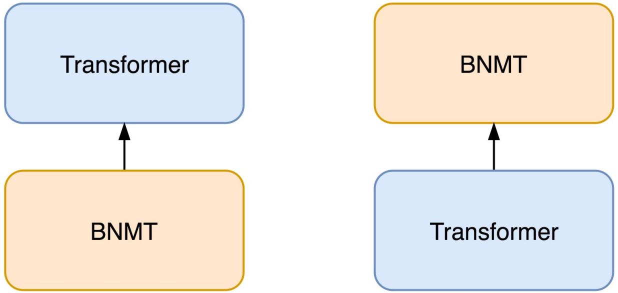
1. Label smoothing
   - Significant for both
2. Multi-head attention
   - Significant for both
3. Layer Normalization
   - Critical to stabilize training (especially with multi-head attention)
4. Synchronous training
   - Critical for Transformer
   - Significant quality drop for RNMT+
   - Successful only with a tailored learning-rate schedule

The Best of Both Worlds - II: Hybrids

Strengths of each architecture:

- **RNMT+**
  - Highly expressive - continuous state space representation.
- **Transformer**
  - Full receptive field - powerful feature extractor.
- Combining individual architecture strengths:
  - Capture complementary information - “Best of Both Worlds”.
- Trainability - important concern with hybrids
  - **Connections between different types of layers need to be carefully designed.**

Encoder - Decoder Hybrids



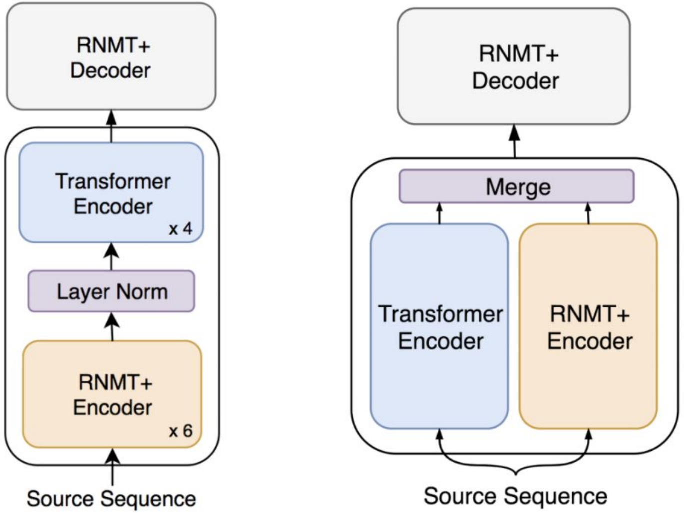
Encoder	Decoder	En→Fr Test BLEU
Trans. Big	Trans. Big	40.73 ± 0.19
RNMT+	RNMT+	41.00 ± 0.05
Trans. Big	RNMT+	$\mathbf{41.12 \pm 0.16}$
RNMT+	Trans. Big	39.92 ± 0.21

Separation of roles:

- Decoder - conditional LM
- Encoder - build feature representations

→ Designed to contrast the roles.
(last two rows)

Encoder Layer Hybrids



(a) Cascaded Encoder (b) Multi-Column Encoder

Improved feature extraction:

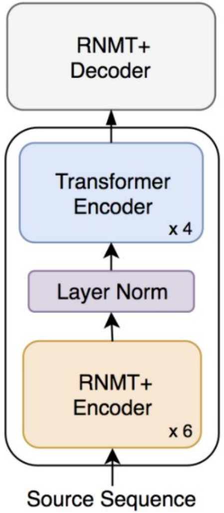
- Enrich stateful representations with global self-attention
- Increased capacity

Details:

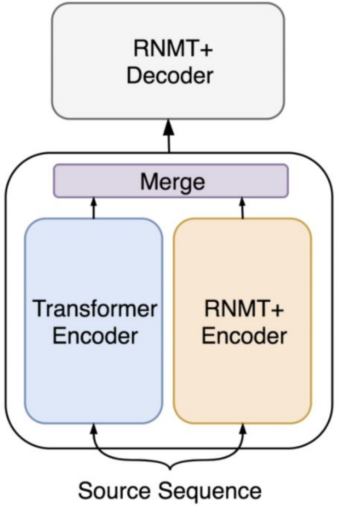
- Pre-trained components to improve trainability
- Layer normalization at layer boundaries

Cascaded Hybrid - **vertical** combination
Multi-Column Hybrid - **horizontal** combination

Encoder Layer Hybrids



(a) Cascaded Encoder



(b) Multi-Column Encoder

Model	En→Fr BLEU	En→De BLEU
Trans. Big	40.73 ± 0.19	27.94 ± 0.18
RNMT+	41.00 ± 0.05	28.59 ± 0.05
Cascaded	$\mathbf{41.67} \pm \mathbf{0.11}$	28.62 ± 0.06
MultiCol	41.66 ± 0.11	$\mathbf{28.84} \pm \mathbf{0.06}$

Lessons Learnt

Need to separate other improvements from the architecture itself:

- Your good ol' architecture may shine with new modelling and training techniques
- **Stronger baselines** (Denkowski and Neubig, 2017)

Dull Teachers - Smart Students

- "A model with a sufficiently advanced lr-schedule is indistinguishable from magic."

$$\mathit{expressivity} \not\propto \mathit{trainability}$$

Understanding and Criticism

- Hybrids have the potential, more than duct taping.
- Game is on for the next generation of NMT architectures

$$quality = f(X, \theta, \mu)$$

Google AI

Sharp Students
Dull Teachers

Machine Translation is a .... problem.

Expressivity and Trainability

What computations can this model perform?

How easy is it to fit a model to the data?

* Great blog: https://blog.evjang.com/2017/11/exp-train-gen.html

$$quality = f(X, \theta, \mu)$$

Modelling (expressivity)

Optimization (trainability)

X : Data

θ : Model

μ : Hyperparameters

Aiding the Model

Model enhancements that eases the training:

- Residuals
- Normalizations (layer, batch, spectral)
- **Transparent attention** (Bapna et al. 2018)
- **Parameter Sharing**
  (Press and Wolf 2016, Jean et al. 2018, Dehghani et al. 2018)

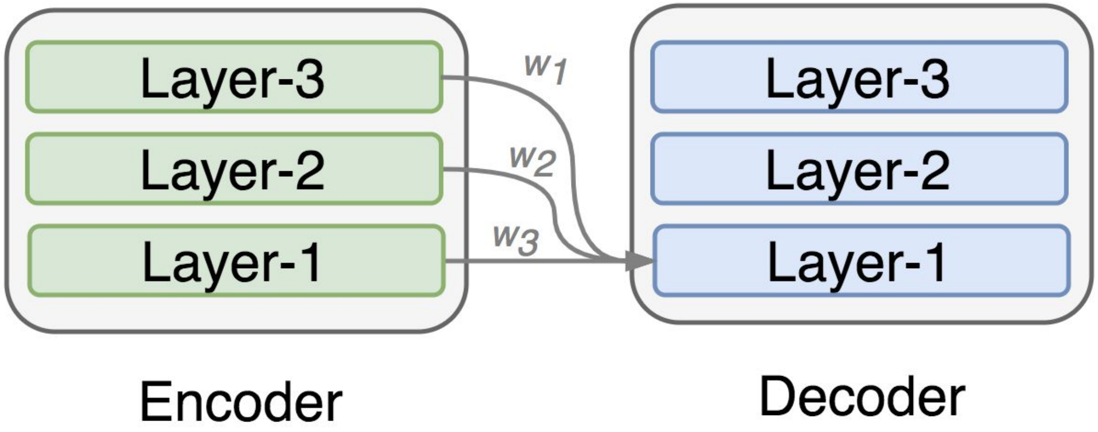
Aiding the Optimizer

Step rule enhancements that eases the training:

- Sync-training
- **Grad-norm tracker** (Chen et al. 2018)
- **Large batches**
  (Goyal et al. 2017, Ott et al. 2018)
- **Learning Rate Schedules** (Bengio 2012)
- **New step rules**
  (Shazeer and Stern 2018, Gupta et al. 2018)

Transparent Attention or Encoder -I

(Bapna et al. 2018- Training Deeper NMT Models with Transparent Attention)



Transparent Attention or Encoder -II

(Bapna et al. 2018- Training Deeper NMT Models with Transparent Attention)

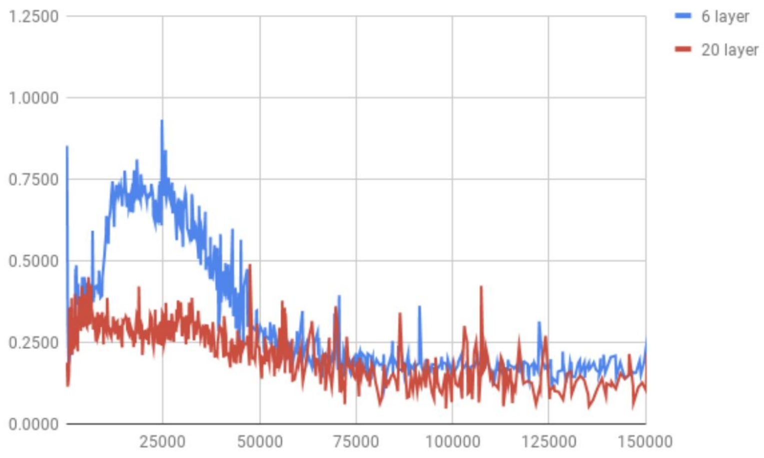


Figure 1: Grad-norm ratio (r_t) vs training step (t) comparison for a 6 layer (blue) and 20 layer (red) Transformer trained on WMT 14 En→De.

$$r_t = \left(\|\nabla_{h_1} L^{(t)}\| / \|\nabla_{h_N} L^{(t)}\| \right)$$

Indicator of a healthy training (Raghu et al. 2017)

- Lower layers converge quickly
- Topmost layers take longer

Expect large grad-norm ratio at the early stages of the training, then flatten.

Transparent Attention or Encoder -III

(Bapna et al. 2018- Training Deeper NMT Models with Transparent Attention)

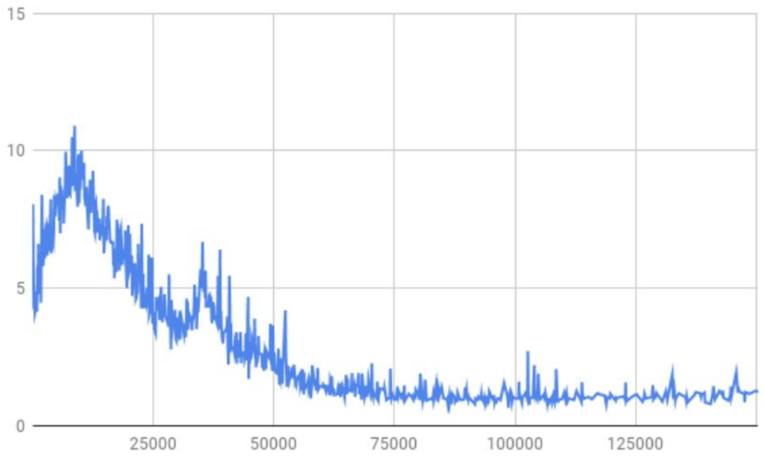


Figure 3: Grad-norm ratio (r_t) vs training step for 20 layer Transformer with transparent attention.

$$r_t = \left(\|\nabla_{h_1} L^{(t)}\| \Big/ \|\nabla_{h_N} L^{(t)}\| \right)$$

Indicator of a healthy training (Raghu et al. 2017)

- Lower layers converge quickly
- Topmost layers take longer

Expect large grad-norm ratio at the early stages of the training, then flatten.

Transparent Attention or Encoder -IV

(Bapna et al. 2018- Training Deeper NMT Models with Transparent Attention)

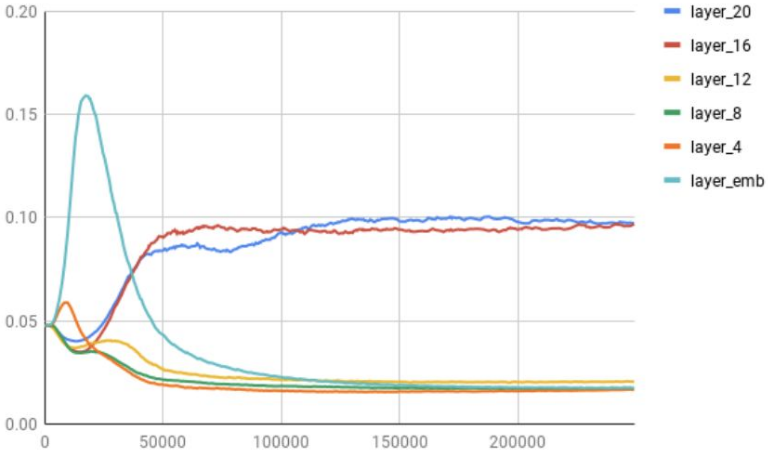


Figure 4: Plot illustrating the variations in the learned attention weights $s_{i,6}$ for the 20 layer Transformer encoder over the training process.

En→De WMT 14	Transformer (Base)				(Big)
Encoder layers	6	12	16	20	6
Num. Parameters	94M	120M	137M	154M	375M
Baseline	27.26	*	*	*	27.94
Baseline - residuals	*	6.00	*	*	N/A
Transparent	27.52	27.79	**28.04**	27.96	N/A

Training dynamics:

- Raghu et al. 2017

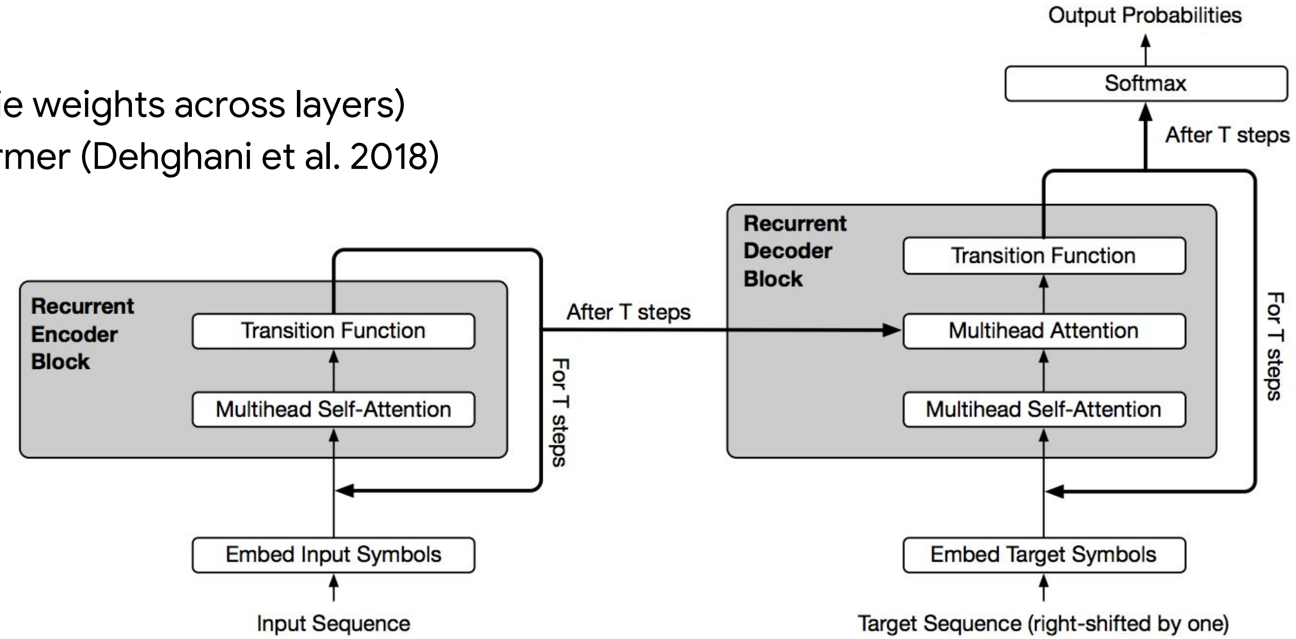
Caveats:

- Residuals & Skip-connections → Shallowness

Parameter Sharing

Reuse the same layer (tie weights across layers)

- Universal Transformer (Dehghani et al. 2018)



- Short-cuts for the credit assignment.
- Improve SOTA further

Gradient Norm Tracker - II

(Chen et al. 2018- The Best of Both Worlds)

All gradients are equal, but some gradients are more equal:

- Identifying pathological error signal dynamically.
- What to discard, when to discard?

Adaptive gradient clipping:

- Keep track of the log of the gradient norm:
    - Exponential moving average
    - Exponential moving standard deviation
- Abort step (skip update completely) when:
    - moving average exceeds 4 std

Large (very large) Batches
(Goyal et al. 2017, Ott et al. 2018)

What is gradient?

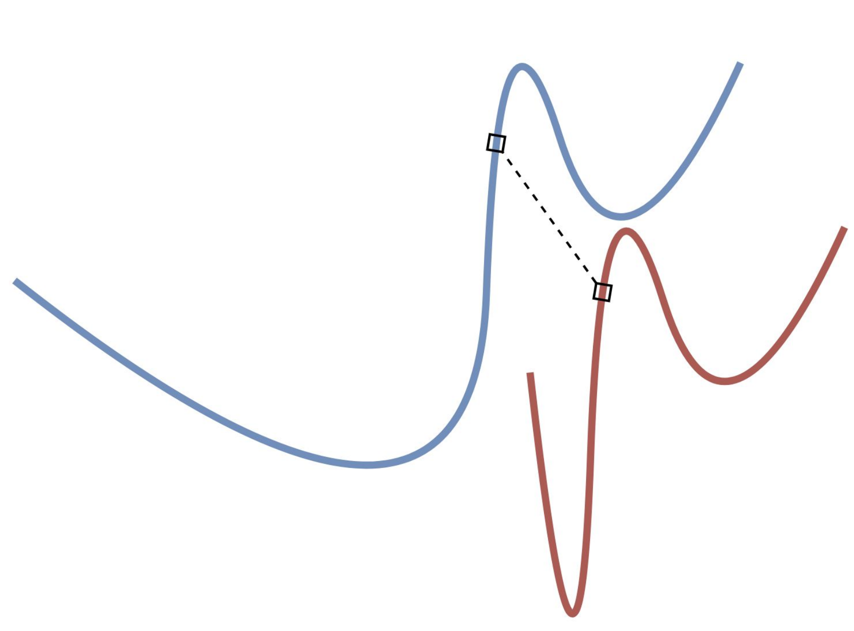
- The vector of first order partial derivatives.

What is gradient descent?

- Use local information to find a minimum.

What does it mean to increase the batch size?

- Better estimate of this first order approximation.



The Realm of Learning Rates

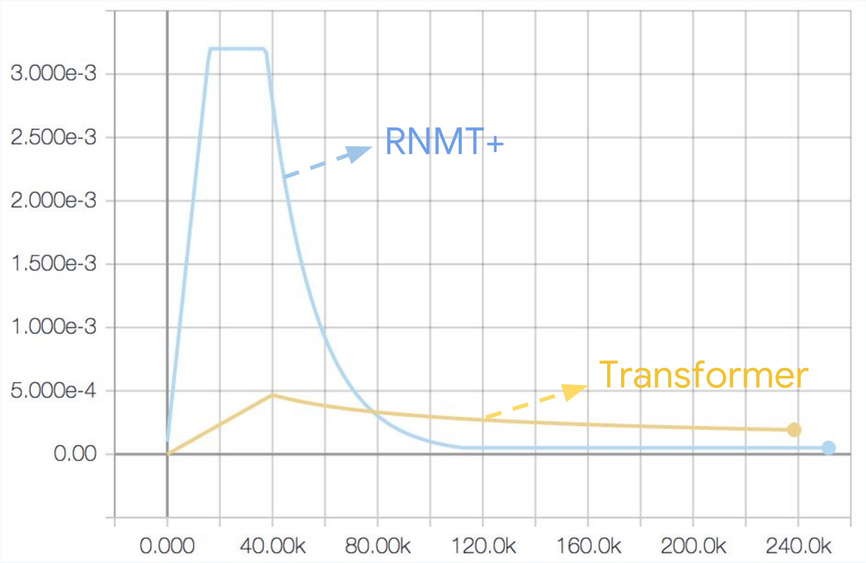
“Often the single most important hyper-parameter”

Practical recommendations for gradient-based training of deep architectures, Bengio 2012

Should always be tuned.

The Learning Rate Schedules - I

- Warm-up
  - Stabilize
  - Necessary for sync training
- Plateau
  - Memorize/Explore/Drift (Shwartz-Ziv and Tishby, 2017)
  - Danger zone if too long
- Decay
  - Compress/Exploit/Diffusion (Shwartz-Ziv and Tishby, 2017)
  - When to end is critical for quality



Better Step Rules

Adafactor (Adam++)

Shazeer and Stern, 2018

Algorithm 1 Adam (Kingma & Ba, 2015)

Inputs: initial point x_0, step sizes $\{\alpha_t\}_{t=1}^T$, first moment decay β_1, second moment decay β_2, regularization constant ϵ

Initialize $m_0 = 0$ and $v_0 = 0$

for $t = 1$ **to** T **do**

$\quad g_t = \nabla f_t(x_{t-1})$

$\quad m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$

$\quad \boxed{v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2}$

$\quad \hat{m}_t = m_t / (1 - \beta_1^t)$

$\quad \hat{v}_t = v_t / (1 - \beta_2^t)$

$\quad x_t = x_{t-1} - \alpha_t \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$

end for

Algorithm 4 Adafactor for weight matrices.

Inputs: initial point $X_0 \in \mathbb{R}^{n \times m}$, relative step sizes $\{\rho_t\}_{t=1}^T$, second moment decay $\{\hat{\beta}_{2t}\}_{t=1}^T$ such that $\hat{\beta}_{21} = 0$, regularization constants ϵ_1 and ϵ_2, clipping threshold d

for $t = 1$ **to** T **do**

$\quad \alpha_t = \max(\epsilon_2, \mathrm{RMS}(X_{t-1})) \rho_t$

$\quad G_t = \nabla f_t(X_{t-1})$

$\quad R_t = \hat{\beta}_{2t} R_{t-1} + (1 - \hat{\beta}_{2t})(G_t^2 + \epsilon_1 1_n 1_m^\top) 1_m$

$\quad C_t = \hat{\beta}_{2t} C_{t-1} + (1 - \hat{\beta}_{2t}) 1_n^\top (G_t^2 + \epsilon_1 1_n 1_m^\top)$

$\quad \hat{V}_t = R_t C_t / 1_n^\top R_t$

$\quad U_t = G_t / \sqrt{\hat{V}_t}$

$\quad \hat{U}_t = U_t / \max(1, \mathrm{RMS}(U_t)/d)$

$\quad X_t = X_{t-1} - \alpha_t \hat{U}_t$

end for

Recipe to make a better teacher

1. Increase the batch size to its maximum
   - Synchronous training
   - Accumulate gradients (gradient checkpointing)
   - Parameter sharing

2. Identify instabilities
   - Normalization
   - Gradient tracking

3. Work on your optimizer
   - Learning-rate schedules
   - Better step-rules

More Practical Tips - I

- Gradient norm can be misleading
  - look at the norm of the actual step (update)

- Denominator or the decaying squared sum of gradients that you normalize by can shrink and become really close to zero (towards the end of the training)
  - may increase the step size too much and prevent you to converge
  - will keep oscillating around a local minima.

- Another summary: (variable norm / norm of the update) ~ [1e-2 to 1e-3]

More Practical Tips - II

- Linear Scaling Rule: (Krizhevsky, 2014, Goyal et al. 2017)
  - When a batch size is multiplied by k, multiply the learning rate by k.
  - Pick as high a learning rate as possible (cannot exceed a certain value)
  - Reduce Beta2 of Adam

- Warmup: (He et al. 2016, Goyal et al. 2017)
  - Initial learning phase (network changes rapidly)
  - Increase warm-up if the model is unstable.

From *Firat and Cho* MTM'16 Talk: Conclusion

What Lies Ahead?

Perhaps, we've only scratched the surface!

- Language barrier, surpassing human level quality.

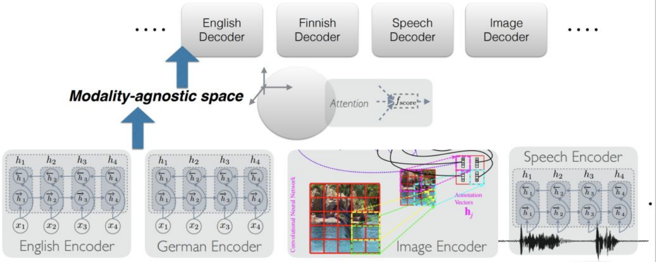
Revisiting the new territory:

Character-level Larger-Context Multilingual

Neural Machine Translation

using,

- Multiple modalities
- Better error signals
- and better GPUs 😎



How many neurons are there in the largest artificial neural network?

Google AI

Thank You

Open source implementations coming very soon!

https://ai.google/research/join-us/

https://ai.google/research/join-us/ai-residency/