
Google AI

Google AI Berlin

Learning Representations

For this talk: representations of text / sentences

You have a string, but you want activations

All layers in a deep neural net processing text

RNNs/LSTMs Are Everywhere

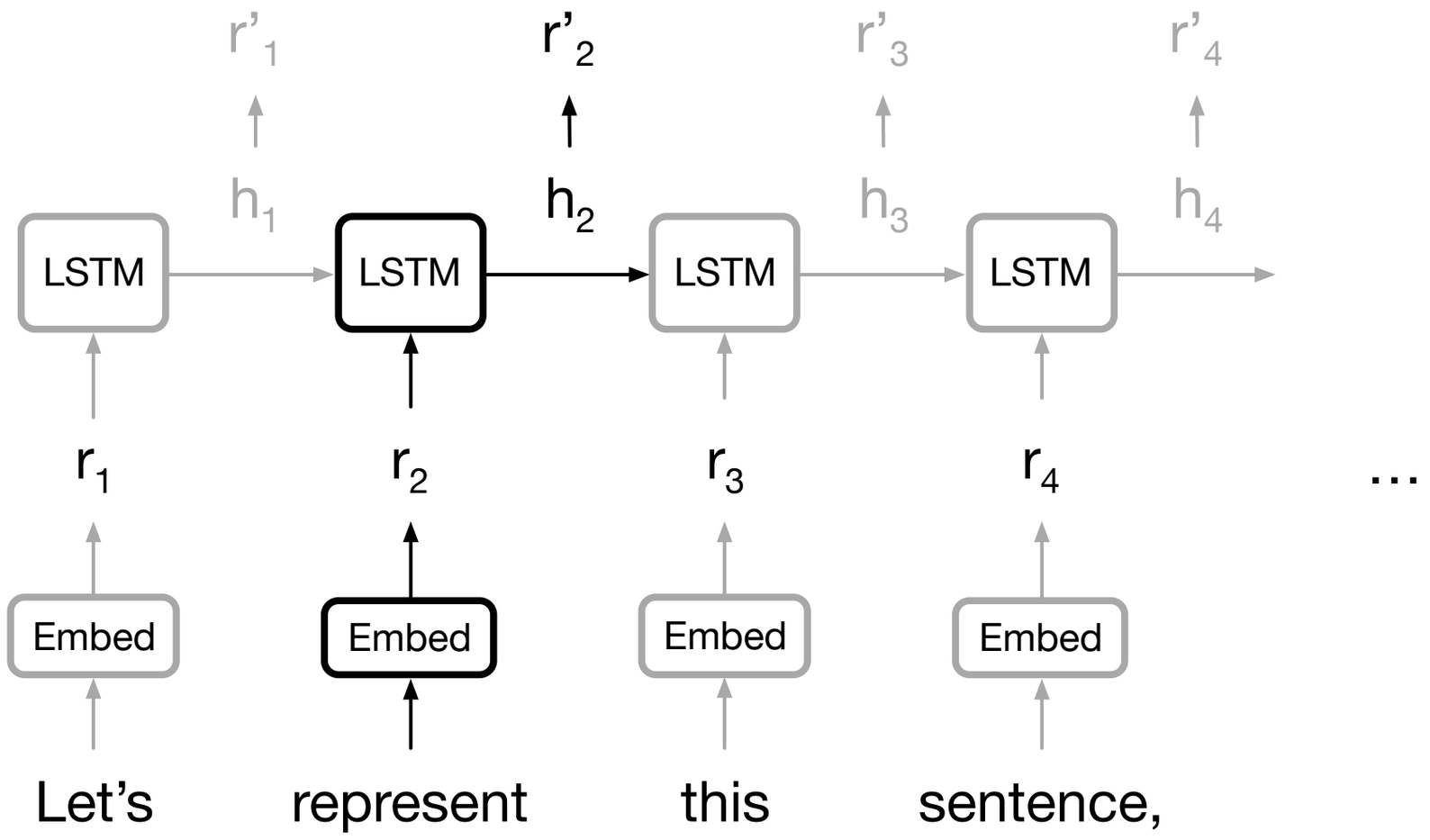
Very successful representing variable-length data

Sequences, e.g. language, time-series, …

Gating for error propagation (LSTM, GRU, …)

At the core of seq2seq

RNNs/LSTMs Are Everywhere



But...

Sequentiality prohibits intra-instance parallelization

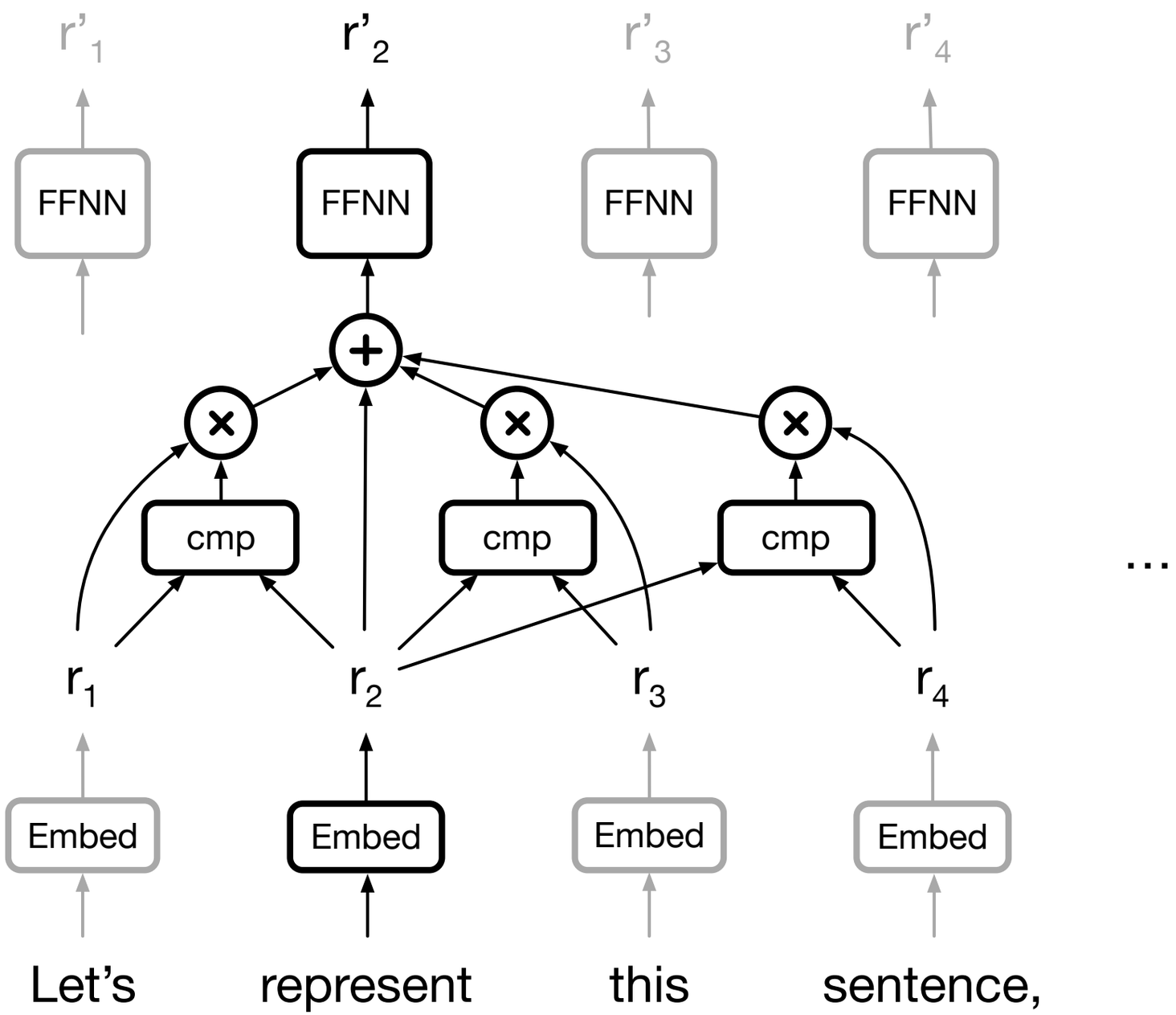
Long-range dependencies are tricky, despite gates

Many modalities are hierarchical-ish (e.g. language)

RNNs (w/ sequence-aligned states) seem wasteful!


We drew trees for decades
and now it's all sequential?!

Self-Attention

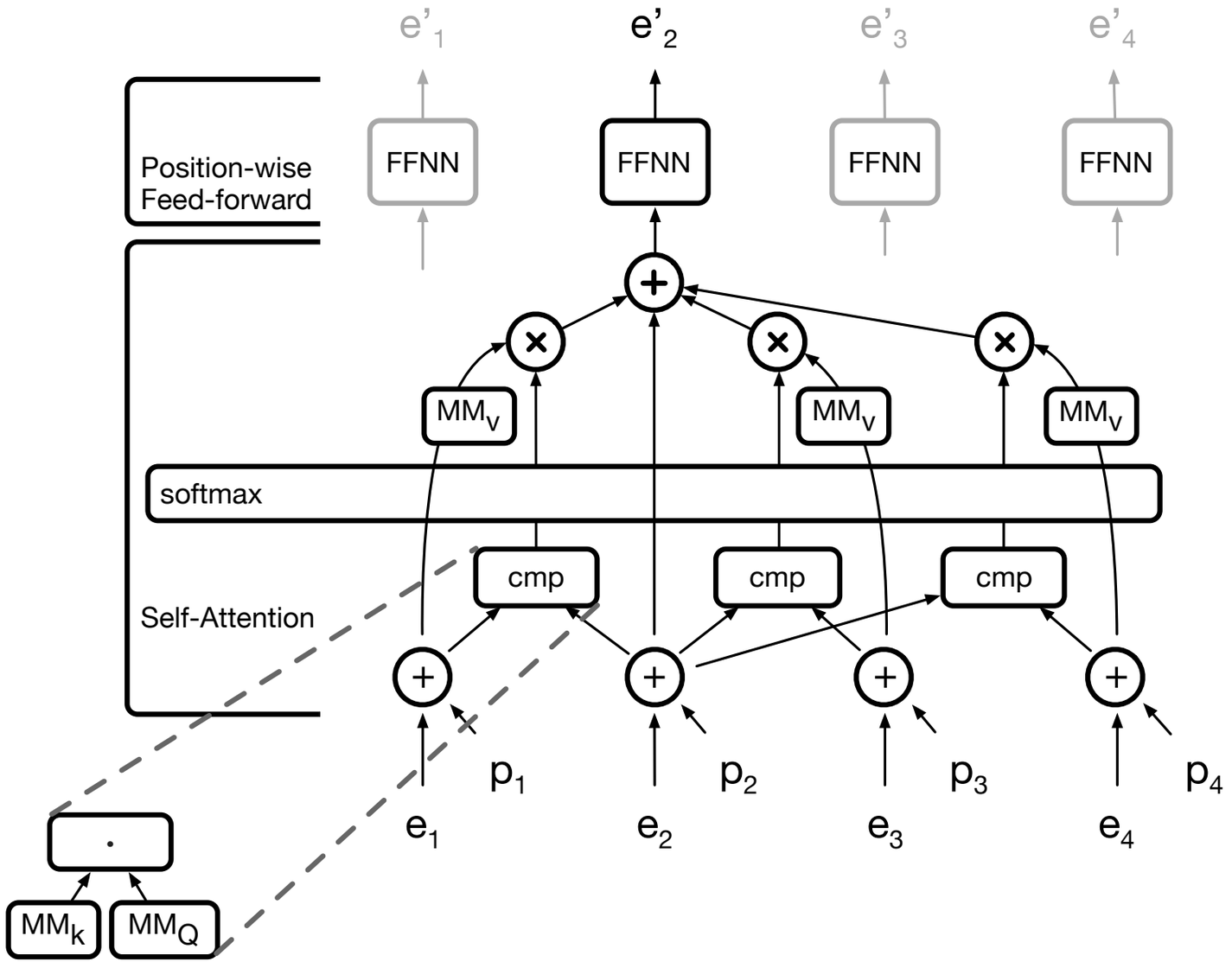


Scaled Dot-Product Attention

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Self-attention when Q, K and V are functions of input

The Transformer



Attention Is All You Need

Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin NIPS 2017

All positions interact directly with all other positions

Quadratic in length, but linear in depth (cf RNNs)

Applications

Machine Translation

Language Modeling

Question Answering

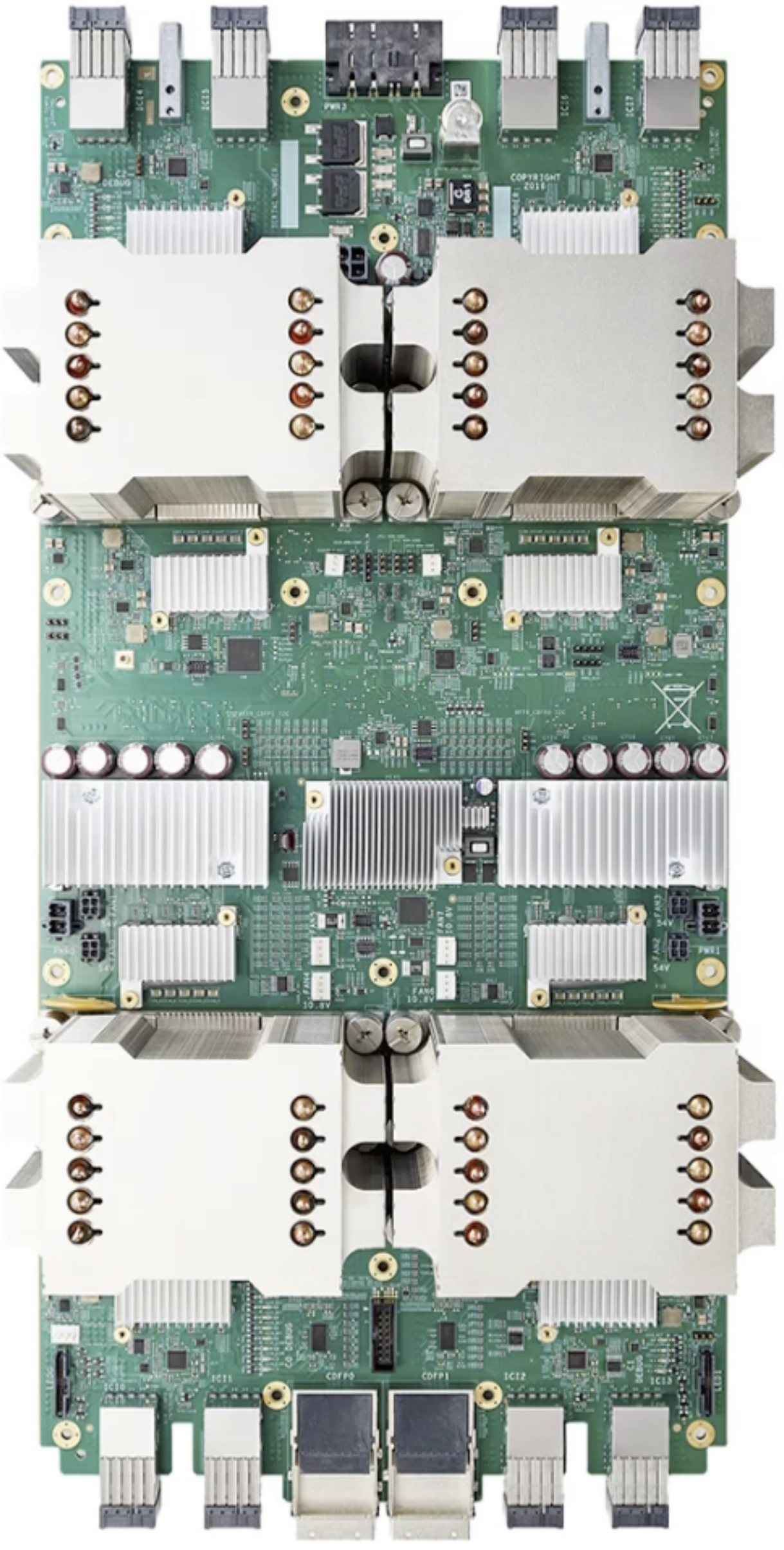
Image Generation

Video Understanding

Gene Sequencing

…

GANs


COPYRIGHT
2016
PWR2
PWR1
FAN2
FAN3
FAN4
FAN5
FAN6
FAN7
54V
10.8V
CDFP0
CDFP1

Generative Pre-Training & BERT

Self-supervised language representation learning

Sentence representations from Transformer LM

Scales better than BiLSTMs in ELMo, especially on TPUs

Transformer efficient for reconstruction objectives

BERT (denoising autoencoder, noise is masking of words)

Improving Language Understanding by Generative Pre-Training
Radford, Narasimhan, Salimans, Sutskever

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Devlin, Chang, Lee, Toutanova

Open Questions

What are good self-supervision/proxy tasks?

Can we learn them?

Are word-indexed representations the right thing?

Variable-length vs. fixed-length representations

Recurrence?

Universal Must Be Better

Neural GPU, LSTM, NTM are computationally universal

Can scale the number of operations for the same input size, with the same parameters (e.g. loops)

#Ops need not be fixed (linear) function of length

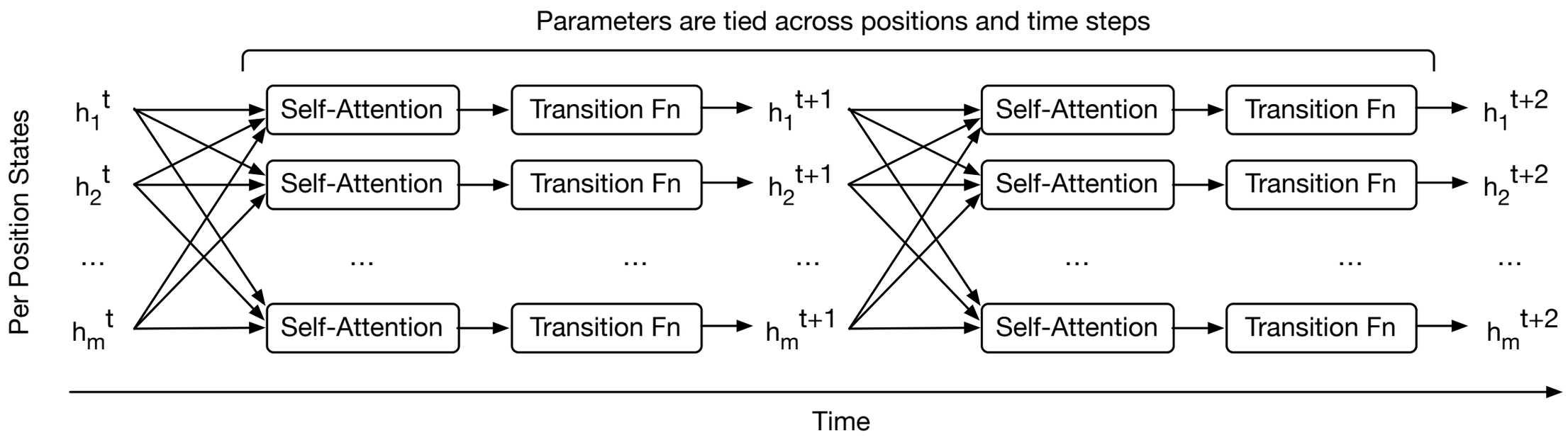
(Though LSTMs typically had that)

Inductive Bias (hand-wavy)

Recurrence can provide generally useful inductive bias

For instance, aids generalization to unseen input lengths

Universal Transformer



Universal Transformers
Dehghani, Gouws, Vinyals, Uszkoreit, Kaiser. ICLR 2019

But actually...

Just a Transformer with tied parameters across layers

Extend position representations with current time-step

Changes the ratio of operations per parameter

Reintroduces inductive bias of a kind of recurrence

Empirically better at syntactic and algorithmic tasks

Mandatory Table With Numbers

Model	LM Perplexity & (Accuracy)			RC Accuracy		
	control	dev	test	control	dev	test
Neural Cache [10]	**129**	139	-	-	-	-
Dhingra et al. [7]	-	-	-	-	-	0.5569
Transformer	154 (0.14)	5336 (0.0)	9725 (0.0)	0.4102	0.4401	0.3988
LSTM	138 (0.23)	4966 (0.0)	5174 (0.0)	0.1103	0.2316	0.2007
Universal Transformer	131(0.32)	279 (0.18)	319 (0.17)	**0.4801**	0.5422	0.5216
Adaptive Universal Transformer	130 (0.32)	**135** (0.22)	**142** (0.19)	0.4603	**0.5831**	**0.5625**

Table 3: LAMBADA language modeling (LM) perplexity (lower better) with accuracy in parentheses (higher better), and Reading Comprehension (RC) accuracy results (higher better). '-' indicates no reported results in that setting.

Now What?

Represent and generate huge things (e.g. books, videos)

The typical Transformer still generates left-to-right

But it can learn “in-filling” (e.g. BERT)

Can we learn to generate that way?

With some parallelism?

Thank you for your attention

Google AI Berlin