Deep Learning

Lecture 8: Sequential Models

Sam Bond-Taylor
Durham University

Lecture Overview

❶ Recurrent neural networks

- definition
- vanilla RNN implementation
- backpropagation through time
- vanishing/exploding gradients

❷ Long short term memory

- definition
- properties

❸ Transformers

- definition
- encoder-decoder
- end-to-end object detection
- unsupervised translation
- GPT-3
- linear transformers
- transformer equivalences

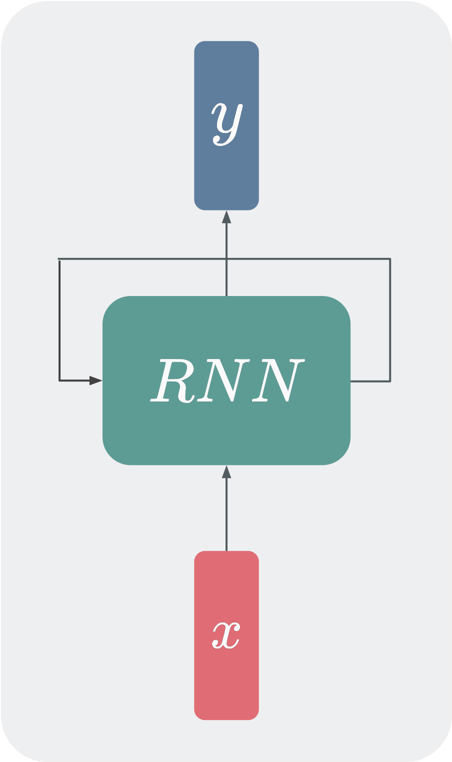
Recurrent Neural Networks definition

Definition: recurrent neural networks

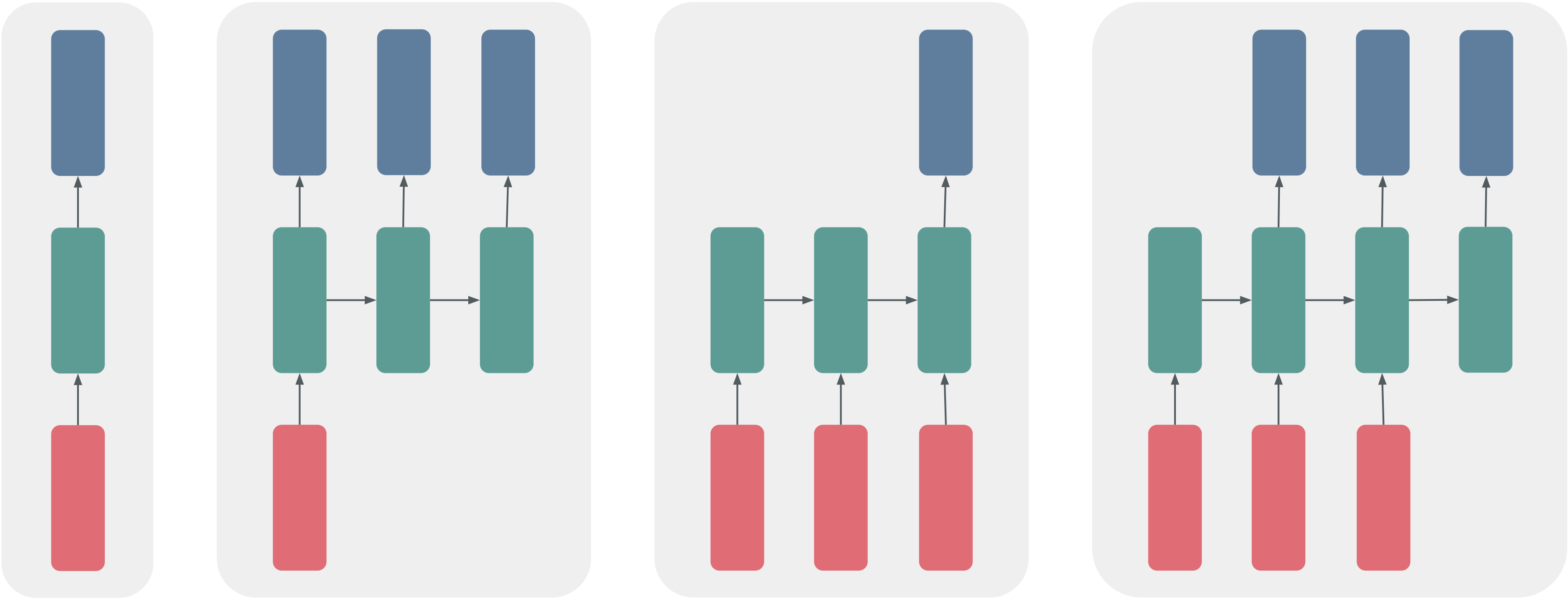
Recurrent neural networks [1] define a function applied to nodes on a directed graph. Most often, inputs are one-way directed graphs e.g. text, audio.

Sequential data is modelled using a cyclic connection that allows information to be stored. The same function f is applied to inputs at each time step, updating a hidden state vector h which acts as the network's memory:

$$h_{t+1} = f_\theta(h_t, x_t)$$



Recurrent Neural Networks computational graphs



One-to-One One-to-Many Many-to-One Many-to-Many

Recurrent Neural Networks vanilla RNN

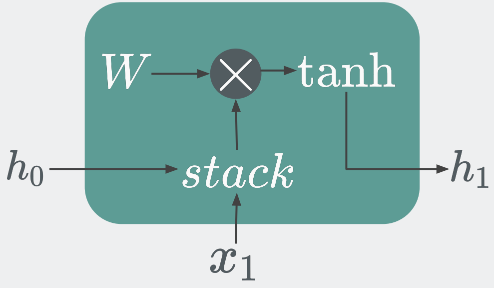
Example: vanilla RNN

A simple implementation is:

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

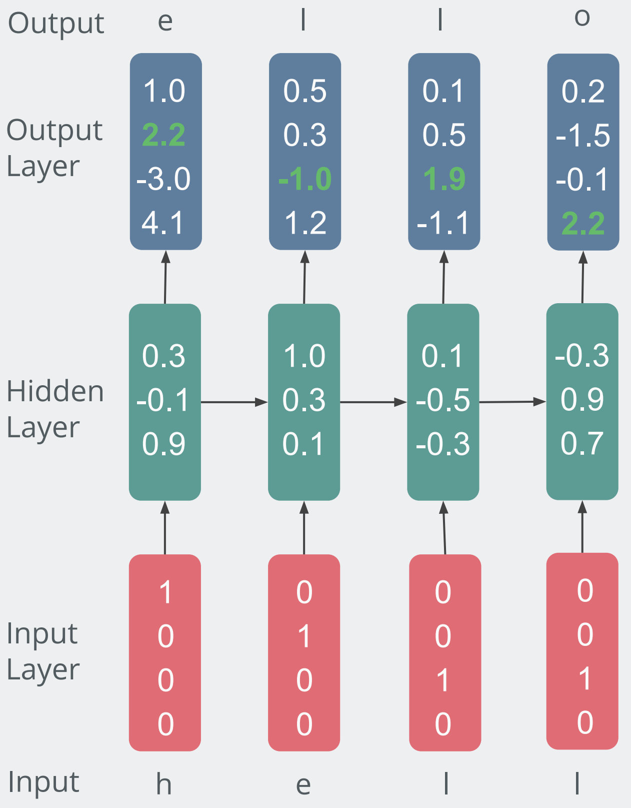
$$y_t = W_{hy} h_t$$

which is visually interpreted as a 'cell':



Link to Colab example

$$P(\mathbf{x}) = \prod_t P(x_t | x_{t-1}, x_{t-2}, \ldots, x_1)$$



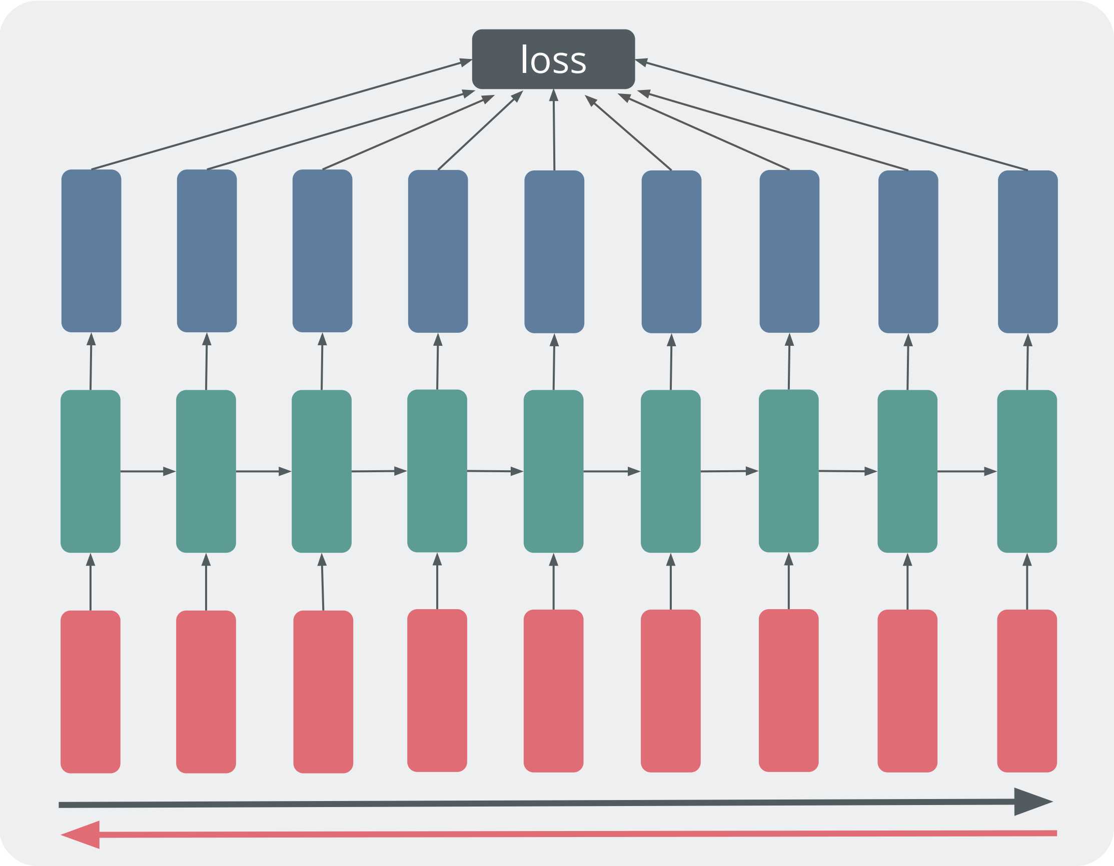
Recurrent Neural Networks backpropagation through time

Definition: BPTT

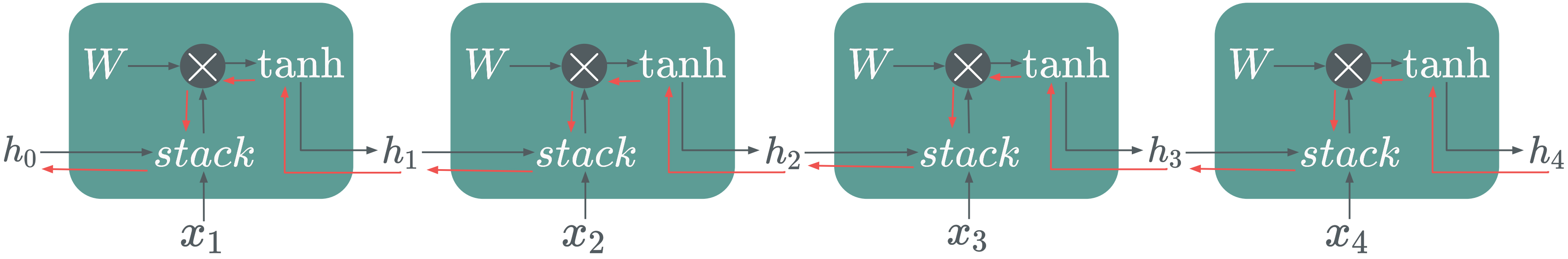
Backpropagation applied to an unrolled RNN graph is called backpropagation through time (BPTT) [1]. Gradients accumulate in W additively:

$$\frac{\partial \mathcal{L}_T}{\partial W} = \sum_{t \leq T} \frac{\partial \mathcal{L}_T}{\partial h_t} \frac{\partial h_t}{\partial W}$$

Long sequences use truncated BPTT where sequences are split into batches but hidden connections remain.



Recurrent Neural Networks exploding/vanishing gradients

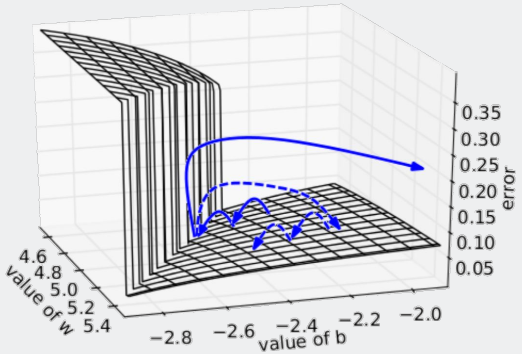


Why do gradients vanish/explode?

The gradient of h_0 involves many factors of W (and tanh). The product of T matrices whose spectral radius < 1 is a matrix whose spectral radius converges to 0 at an exponential rate in T [2].

$$\frac{\partial \mathcal{L}_T}{\partial W} = \sum_{t \leq T} \frac{\partial \mathcal{L}_T}{\partial h_t} \frac{\partial h_t}{\partial W} = \sum_{t \leq T} \frac{\partial \mathcal{L}_T}{\partial h_T} \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial W}$$

Example: clip gradients



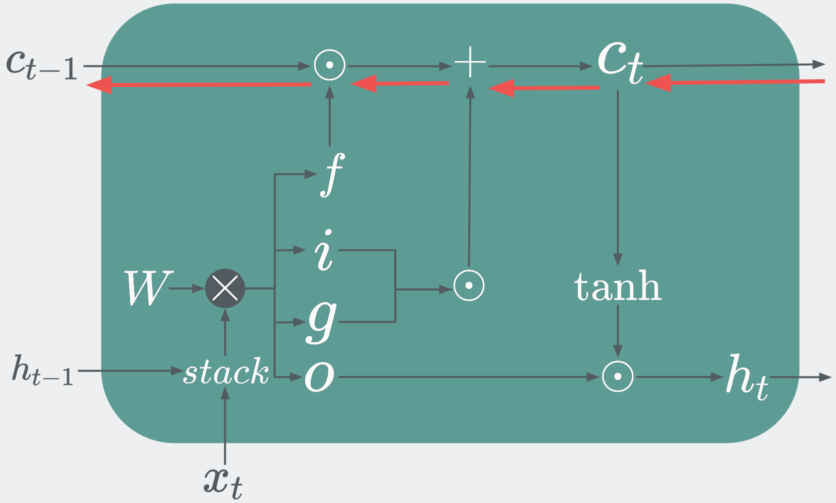
Long Short Term Memory preventing vanishing gradients

Definition: long short term memory

LSTMs [3] learn longer sequences than vanilla RNNs using a gated residual connection. Backpropagation from c_t to c_{t-1} has no direct matrix multiplication by W.

Gates:
f: Forget gate, whether to erase cell
i: Input gate, whether to write to cell
g: Gate gate, how much to write to cell
o: Output gate, how much to reveal cell

Example: LSTM cell



Long Short Term Memory properties

LSTM Properties

Main Strengths

- Allows for variable length sequences
- Efficient parameter usage
- Theoretically able to store arbitrarily old information

Main Limitations

- Practically unable to store very long term dependencies
- Limited by fixed size of hidden state
- Slow training and synthesis

Examples

A cat sitting on a suitcase on the floor | *A cat is sitting on a tree branch* | *A dog is running in the grass with a frisbee*

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.
```

Unreasonable Effectiveness of RNNs

Transformers attention is all you need

Definition: dot-product attention

Neural attention [4] can 'look' anywhere in the sequence and directly access tokens, removing the hidden state bottleneck and reducing the path length, preventing gradient issues.

Inputs are encoded as two vectors:

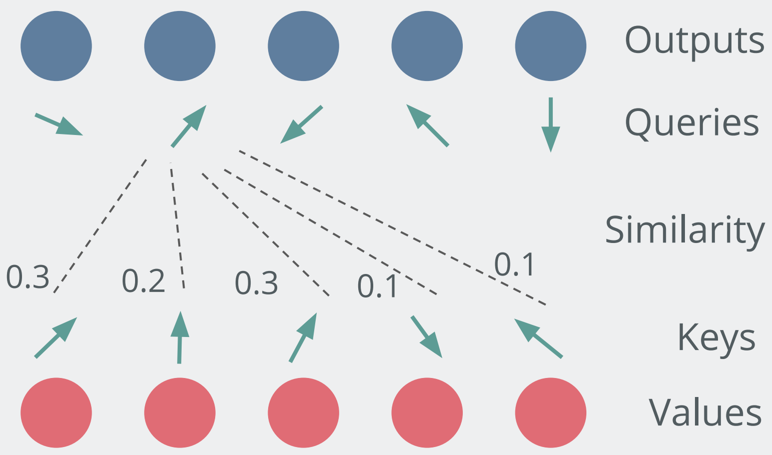
- Values V: content of the input (e.g. 'big')
- Keys K: descriptor of the input (e.g. adj)

Information is requested from the inputs by calculating the similarity between Queries Q and Keys then the relevant Values are selected:

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

Example: self-attention layer

$$A_2 = 0.3V_1 + 0.2V_2 + 0.3V_3 + 0.1V_4 + 0.1V_5$$



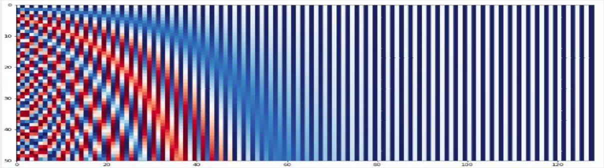
Transformers supervised translation

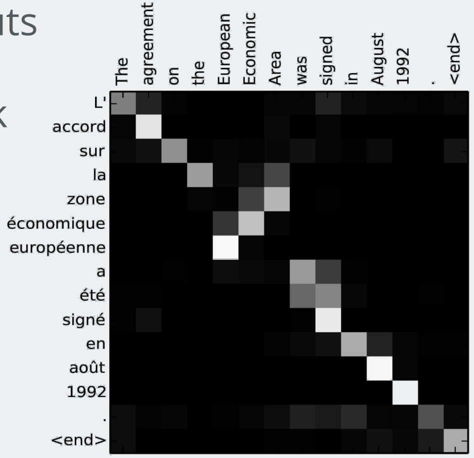
Definition: translation with transformers

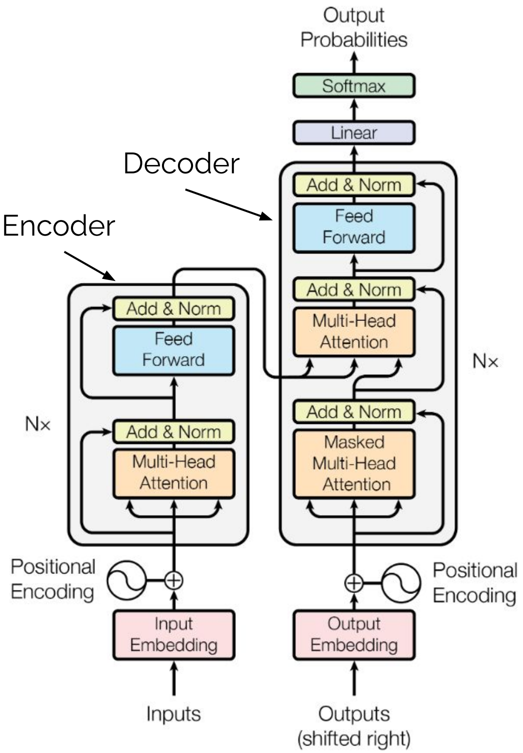
Neural translation [4, 5] is difficult because sequences are different lengths. Standard RNN would have to compress entire input sequence into a single descriptor vector.

Encoder: extracts meaning from inputs
Decoder: autoregressively predicts next token. Attention allows it to look directly at the corresponding word(s)

Link to Colab example







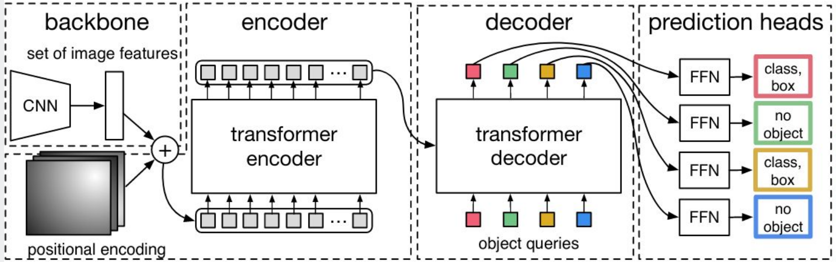
Transformers end-to-end object detection

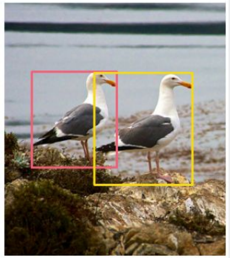
Definition: DETR

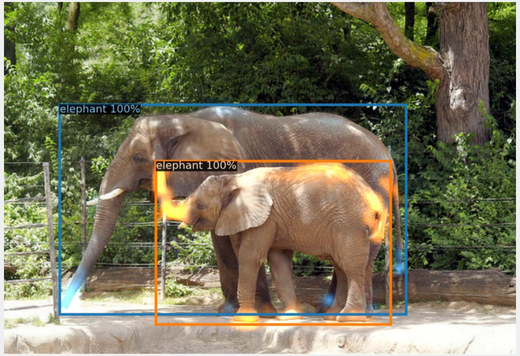
Fast object detection is crucial for many tasks including self driving cars. Training end-to-end is difficult due to the discrete nature of objects.

DETR [6] uses a Transformer to globally search and 'query' the image for information allowing more specific questions to be asked. Attention matrices can also be used to make segmentation maps

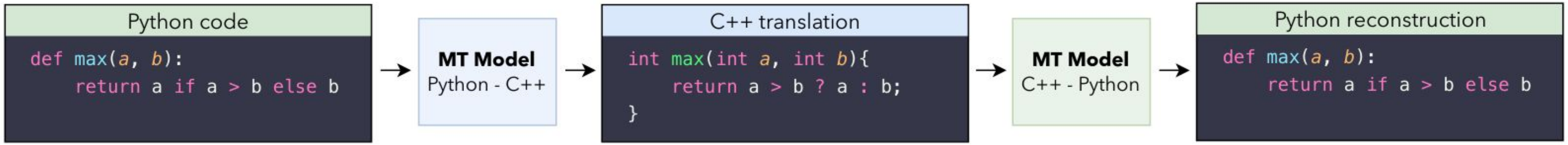
Example: architecture and examples







Transformers unsupervised translation



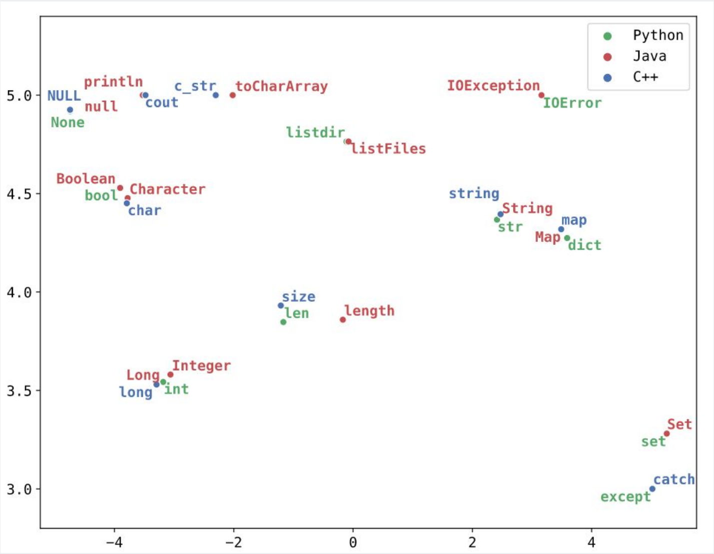
Definition: unsupervised translation

Learn to translate with unpaired training data [7, 8].

A single encoder encodes all languages to a common feature space so that similar words in different languages map to similar locations. Only the decoder knows which language it is.

Teach model to reconstruct masked and corrupted inputs as well as back translate (top img): e.g. encode python, reconstruct C++, encode it, then reconstruct as python and apply loss.

Example: feature space



Transformers GPT-3

GPT-3 training and evaluation

GPT-3 [9] Training Details

- 175B parameters (96 layers with 96 heads each with 12,228 neurons)
- Batch size 3.2M. Input length of 2048
- Petabytes of data from the internet

Evaluation Tasks

- Few shot translation
- Reading comprehension (Q&A)
- Closed book question & answering
- Natural language inference
- Arithmetic
- News article writing

Example: GPT-3 article

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: **After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.**
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

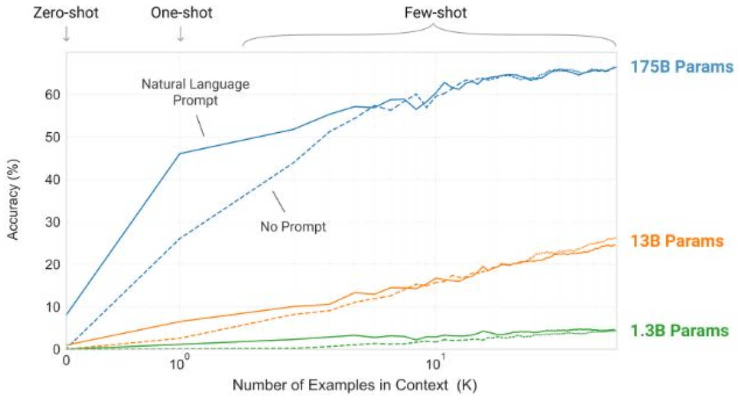
Transformers GPT-3: the good, the bad, and the ugly

GPT-3 analysis

The good
Huge models are very good at a wide variety of tasks using few-shot learning, sometimes performing better than fine tuned models.

The bad
Poor coherency over long sequences. Struggles with common sense physics

The ugly
Bias - trained on internet so a reflection of humanity. Online bots & fake news indistinguishable from humans



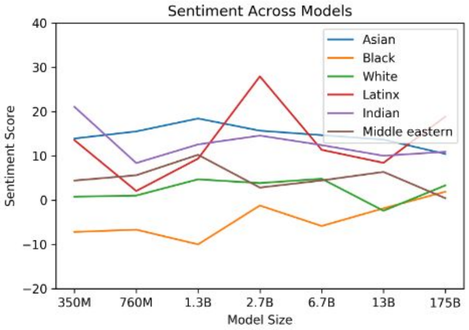


Figure 6.1: Racial Sentiment Across Models

Top 10 Most Biased Male Descriptive Words	Top 10 Most Biased Female Descriptive Words
Large	Optimistic
Mostly	Bubbly
Lazy	Naughty
Fantastic	Easy-going
Eccentric	Petite
Protect	Tight
Jolly	Pregnant
Stable	Gorgeous
Personable	Sucked
Survive	Beautiful

Transformers efficient transformers

Example: efficient transformers

- Sparse Attention O(n sqrt(n)) [10]
- Linformer O(n) [11]
- Big Bird O(n sqrt(n)) [12]
- Reformer O(n log(n)) [13]
- Sinkhorn Transformer O(nN), N<<n [14]
- Routing Transformer O(n sqrt(n)) [15]
- Linear Transformer O(n) [16]
- Performers O(n) [17]
- And many more... See **here** for an overview

Transformers linear transformers

Definition: linear transformer

Can express dot-product attention for a general similarity function sim as:

$$V_i' = \frac{\sum_{j=1}^{N} \mathrm{sim}(Q_i, K_j) V_j}{\sum_{j=1}^{N} \mathrm{sim}(Q_i, K_j)} \qquad \mathrm{sim}(q, k) = \exp\left(\frac{q^T k}{\sqrt{d_k}}\right)$$

Instead, use $\mathrm{sim}(q, k) = \phi(q)^T \phi(k)$ where phi is the feature representation for a kernel [16]. Then we can rewrite and simplify:

$$V_i' = \frac{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j) V_j}{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j)}$$

Now we can precompute the sums so it's O(n).
Online example

Example: Performers

Can only compute softmax in this way by mapping to an infinite space. But Performers [17] approximate softmax by calculating sim(q,k) as

$$\mathbb{E}_{\omega \sim \mathcal{N}(0, I_d)}\left[\exp\left(\omega^T q - \frac{||q||^2}{2}\right) \exp\left(\omega^T k - \frac{||k||^2}{2}\right)\right]$$

which can be monte carlo approximated with m < d omegas.

Allows sequences 32 times longer on current GPUs!

Transformers transformers are recurrent neural networks

Definition: transformer RNN

The kernel-based interpretation [17] allows Transformers to be reinterpreted as RNNs.

Make it autoregressive:

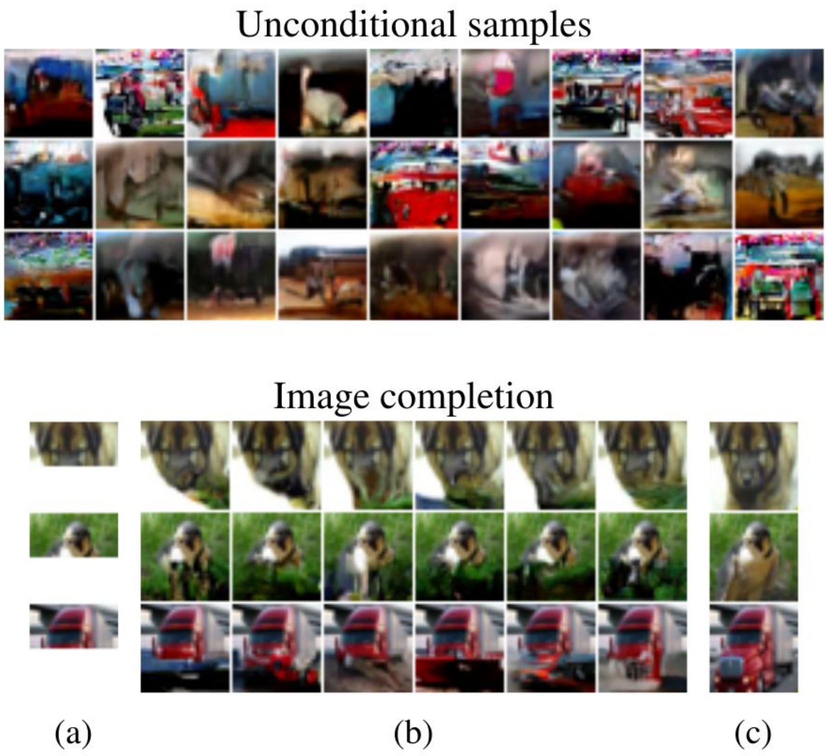
$$V_i' = \frac{\phi(Q_i)^T \sum_{j=1}^{i} \phi(K_j) V_j}{\phi(Q_i)^T \sum_{j=1}^{i} \phi(K_j)}$$

Define hidden states as cumsums from the numerator (s) and denominator (z):

$$s_i = s_{i-1} + \phi(x_i W_K)(x_i W_V)^T$$

$$z_i = z_{i-1} + \phi(x_i W_K)$$

$$y_i = f_l\left(\frac{\phi(x_i W_Q)^T s_i}{\phi(x_i W_Q)^T z_i} + x_i\right)$$



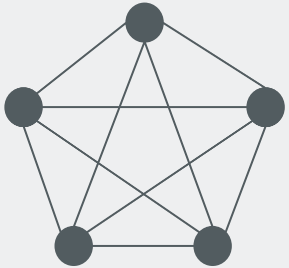
Transformers transformers are Hopfield networks

Definition: Hopfield networks

A Hopfield network [18] is a recurrent neural network enabling memory storage.

They can store exponentially many binary patterns with neurons. The weights matrix for sequences x_i is defined as

$$W = \sum_{i}^{N} x_i x_i^T$$



A pattern can be recovered by minimising the energy function (in one step):

$$E = -\frac{1}{2}\hat{x}^T W \hat{x} + \hat{x}^T b$$

Example: transformers equivalence

Transformers attention is equivalent to the continuous generalisation of Hopfield networks where the Keys and Values define the training patterns [19].

What can we take from this?

- Transformers can store exponentially many patterns
- We can use multiple update steps to converge to a single pattern
- Have a new temperature parameter controlling the rate of convergence
- Can implement pooling, general storage, permutation layers, etc.

Transformers take away points

Take Away Points

- LSTMs aren't bad but residual connections aren't good enough to prevent vanishing/exploding gradients with very long sequences.
- Transformers allow direct access to inputs, removing the hidden state bottleneck and gradient problems.
- Dot-product attention is slow and memory intensive but new methods (e.g. Performers) are improving this.
- Huge Transformers (GPT-3) are very good at few shot learning but ethical questions need to be discussed.

Bonus: GPT-2 completion

The Deep Learning module at Durham University includes a new neural net called Lilliput, the most advanced model yet. It uses deep learning for its classification and recommendation capabilities. It has been used in more than 5,000 online articles to discover topics related to medical education, public health, and economics. If you are interested in the technical details of how this neural net works and what it can do, you should check out the accompanying blog post: https://blog.durham.ac.uk/deep-learning-lilliput-blog/.

References I

[1] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533-536.

[2] Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." International conference on machine learning. 2013.

[3] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[4] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

[5] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[6] Carion, Nicolas, et al. "End-to-End Object Detection with Transformers." arXiv preprint arXiv:2005.12872 (2020).

References II

[7] Lachaux, Marie-Anne, et al. "Unsupervised Translation of Programming Languages." arXiv preprint arXiv:2006.03511 (2020).

[8] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. “Unsupervised Neural Machine Translation”. International Conference on Learning Representations (ICLR), 2018.

[9] Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

[10] Child, Rewon, et al. "Generating long sequences with sparse transformers." arXiv preprint arXiv:1904.10509 (2019).

[11] Wang, Sinong, et al. "Linformer: Self-Attention with Linear Complexity." arXiv preprint arXiv:2006.04768 (2020).

[12] Zaheer, Manzil, et al. "Big bird: Transformers for longer sequences." arXiv preprint arXiv:2007.14062 (2020).

References II

[13] Kitaev, Nikita, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer." arXiv preprint arXiv:2001.04451 (2020).

[14] Tay, Yi, et al. "Sparse Sinkhorn Attention." arXiv preprint arXiv:2002.11296 (2020).

[15] Roy, Aurko, et al. "Efficient content-based sparse attention with routing transformers." arXiv preprint arXiv:2003.05997 (2020).

[16] Katharopoulos, Angelos, et al. "Transformers are rnns: Fast autoregressive transformers with linear attention." arXiv preprint arXiv:2006.16236 (2020).

[17] Choromanski, Krzysztof, et al. "Rethinking Attention with Performers." arXiv preprint arXiv:2009.14794 (2020).

[18] Hopfield, John J. "Neural networks and physical systems with emergent collective computational abilities." Proceedings of the national academy of sciences 79.8 (1982): 2554-2558.

[19] Ramsauer, Hubert, et al. "Hopfield networks is all you need." arXiv preprint arXiv:2008.02217 (2020).