BERT (Bi-directional Encoder Representations)

Kolluru Sai Keshav,
PhD Scholar

Image Credits: Google

Problems with LSTM

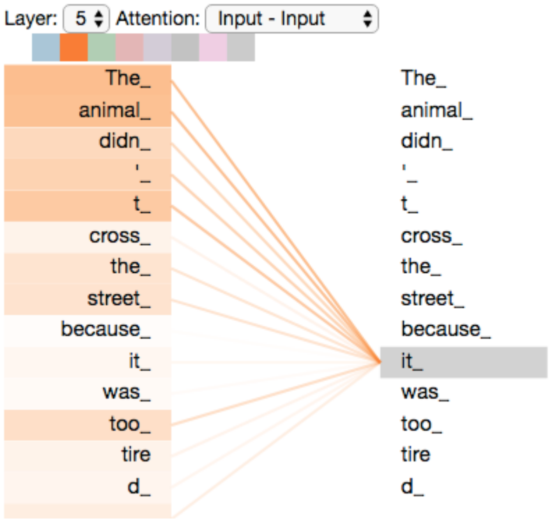
- They are slow
  - Sequential nature of computation makes it tough to optimize operations on GPUs
  - In contrast to CNNs, where convolutions are completely parallelizable
- They are not deep
  - They suffer from vanishing gradient problems, which is aggravated for deeper networks
  - Lack of depth constrains the compositionality power of the network
  - Deepest LSTM networks are 8 layered, in-contrast to 50-layered Resnets
- They don’t transfer well
  - Networks trained on one task, do not generalize well to even other datasets in the same task, not to speak about other tasks
  - ImageNet-trained ResNet is fine-tuned on many other datasets to get

Self-Attention

- Attention:
  - Weighted sum of vectors
  - Seq2Seq: The weight is computed between the current decoder state and the input vectors
  - Memory Networks: The weight is computed between the query vector and the memory vectors
- Self-Attention:
  - Embedding of each token is a weighted sum of embedding of other tokens

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

Self-Attention



Self-Attention

$$\alpha_{ts} = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \qquad \text{[Attention weights]}$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \qquad \text{[Context vector]}$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t; \boldsymbol{h}_t]) \qquad \text{[Attention vector]}$$

General-Attention

Image Credits: http://jalammar.github.io/illustrated-transformer/

Self-Attention

- Attention:
  - Weighted sum of vectors
  - Seq2Seq: The weight is computed between the current decoder state and the input vectors
  - Memory Networks: The weight is computed between the query vector and the memory vectors
- Self-Attention:
  - Embedding of each token is a weighted sum of embedding of other tokens

- Benefits:
  - Highly Parallelizable - Solves the issue with speed
  - Architecture use does not suffer from vanishing gradient,
    hence, can be made arbitrarily deep (upto 30 layers - as of.. yesterday)

Pretraining - Masked Language Modelling

- In NLP, we are interested in solving a variety of end tasks - Question Answering, Search, etc.
- One approach - train neural models from scratch
- Issue - This involves two things
  - Modelling of Syntax and Semantics of the language
  - Modelling of the end-task
- Pretraining - Learns the modelling of syntax and semantics - through another task
- So the model can focus exclusively on modelling of end-task

Pretraining - Masked Language Modelling

- Pretraining - Learns the modelling of syntax and semantics - through another task
- So the model can focus exclusively on modelling of end-task

- Which base task to choose:
    - Must have abundant data available
    - Must require learning of syntax and semantics
- Language Modelling
    - Does not require human annotated labels - abundance of sentences
    - Requires understanding of both syntax and semantics to predict the next word in sentence

Masked Language Modelling

- Issue with Language modelling - Unidirectional
- Cannot train model on bidirectional context - required for many end tasks
- Solution: Masked Language Modelling
  - Randomly mask a word in the sentence - train the model to predict it
  - Similar to the problem in A2

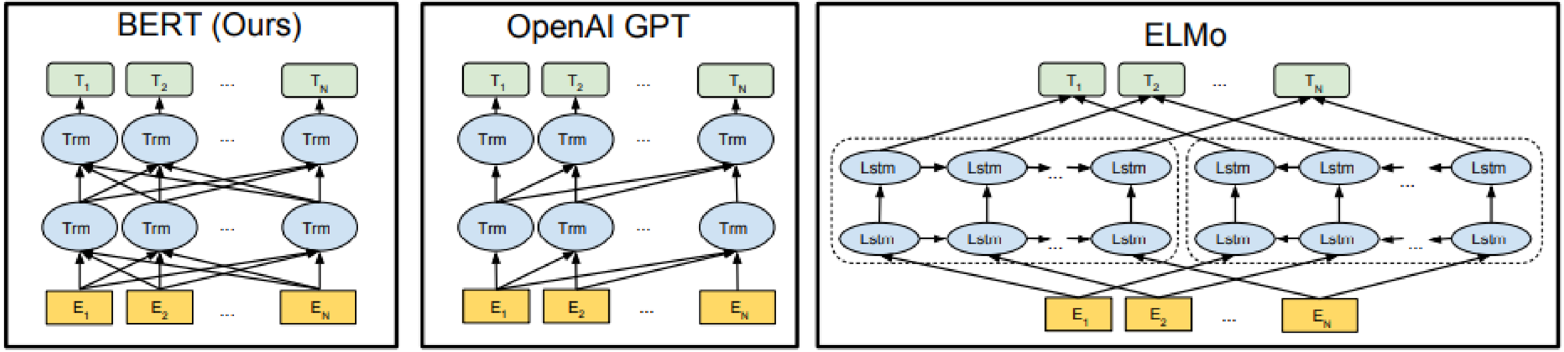


*Image Credits: [1]

The Transformer Architecture

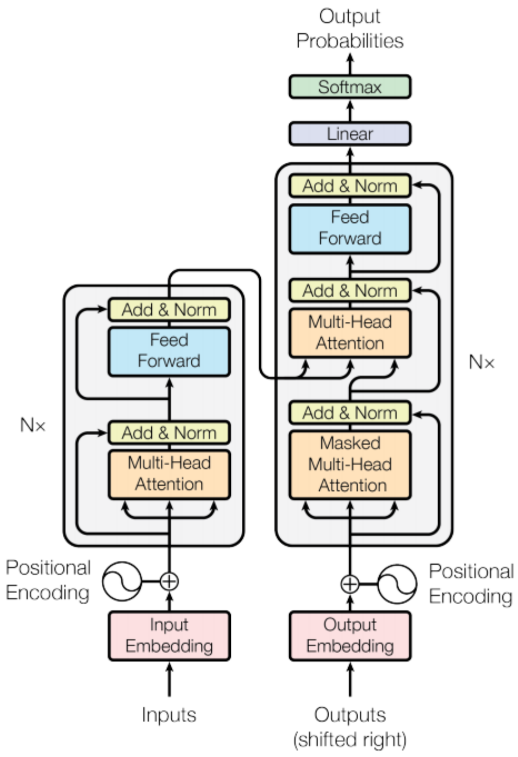
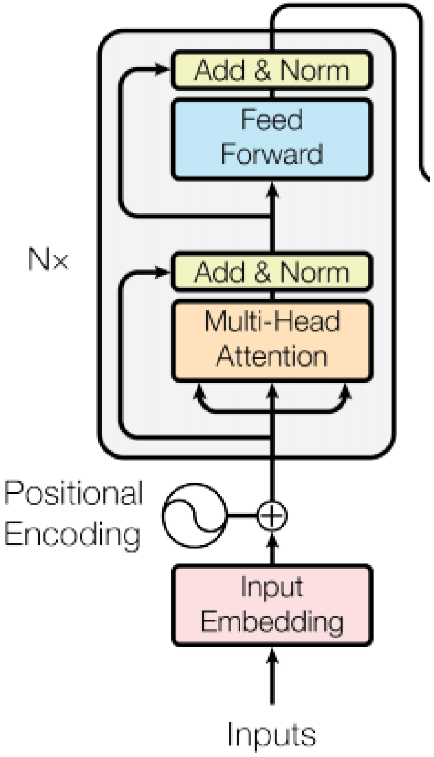


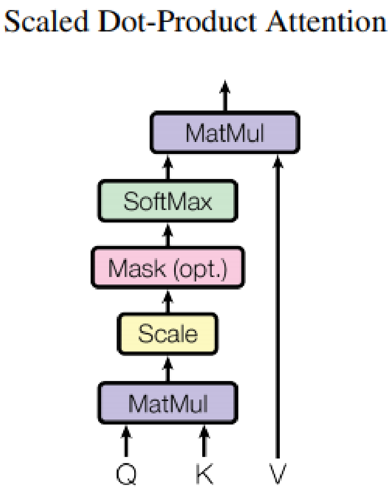
Figure 1: The Transformer - model architecture.

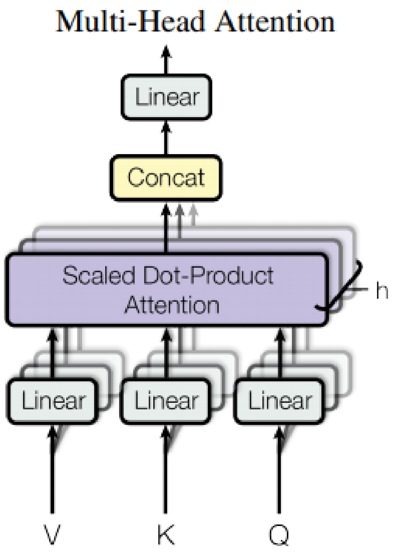


Encoder

*Image Credits: [2]

Multi-Head Attention





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

*Image Credits: [2]

Word-Piece tokenizer

- Middle ground between character level and word level representations
- tweeting → tweet + ##ing
- xanax → xa + ##nax
- Technique originally taken from paper for Japanese and Korean languages

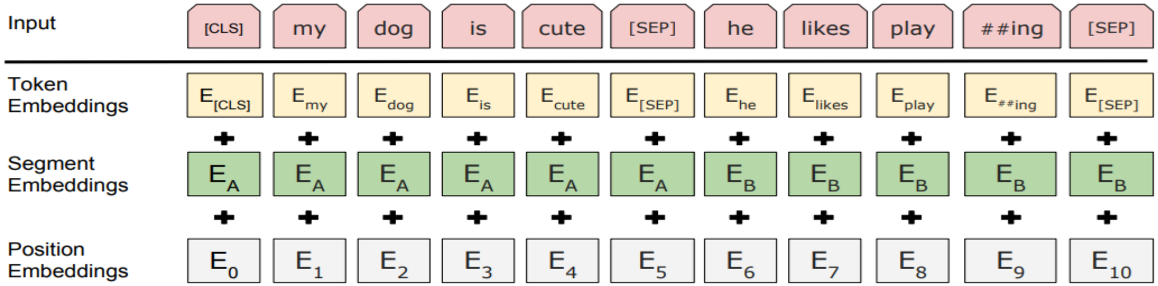
- Given a training corpus and a number of desired tokens D, the optimization problem is to select D wordpieces such that the resulting corpus is minimal in the number of wordpieces when segmented according to the chosen wordpiece model.

Schuster, Mike, and Kaisuke Nakajima. "Japanese and korean voice search." *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012.

Positional Encoding

- Originally used by CNNs for capturing the sequential nature of the input
- Type of positional encoding:
  - Learned (Use by BERT)
  - Fixed: Using a generative function

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$



- This is the only way sequential information is maintained in the model
  - Dream experiment: Remove positional encoding and train the model
  - The extent of reduction in accuracy will show the importance of sequential nature of the current nlp tasks

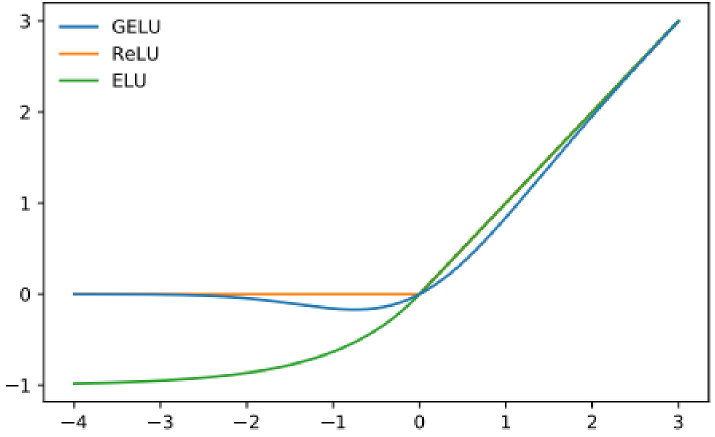
*Image Credits: [1]

Misc Details

- Uses an activation function called GeLU - a continuous version of ReLU
- Multiplies the input with a stochastic one-zero map (in the expectation)

$$\text{GELU}(x) = xP(X \leq x) = x\Phi(x).$$

$$0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$$



- Optimizer: A variant of the Adam optimizer where the learning rate first increases (Warm-up phase) and is then decayed

*Image Credits: [3]

Practical Tips

- Proper modelling of input for BERT is extremely important
  - Question Answering: [CLS] Query [SEP] Passage [SEP]
  - Natural Language Inference: [CLS] Sent1 [SEP] Sent2 [SEP]
  - BERT cannot be used a feature extractor
  - Embedding of query, embedding of passage and take their dot product
- Maximum input length is limited to 512. Truncation strategies have to be adopted
- Number of hyper-parameters are actually few:
  - Batch Size: 16, 32
  - Learning Rate: 3e-6, 1e-5, 3e-5, 5e-5
- BERT-Large model requires random restarts to work
- Always PRE-TRAIN, on related task - will improve accuracy
- Super-optimized for TPUs, not so much for GPUs

References

[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*(2018).

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

[3] Hendrycks, Dan, and Kevin Gimpel. "Gaussian Error Linear Units (GELUs)." *arXiv preprint arXiv:1606.08415* (2016).