Transformer Model

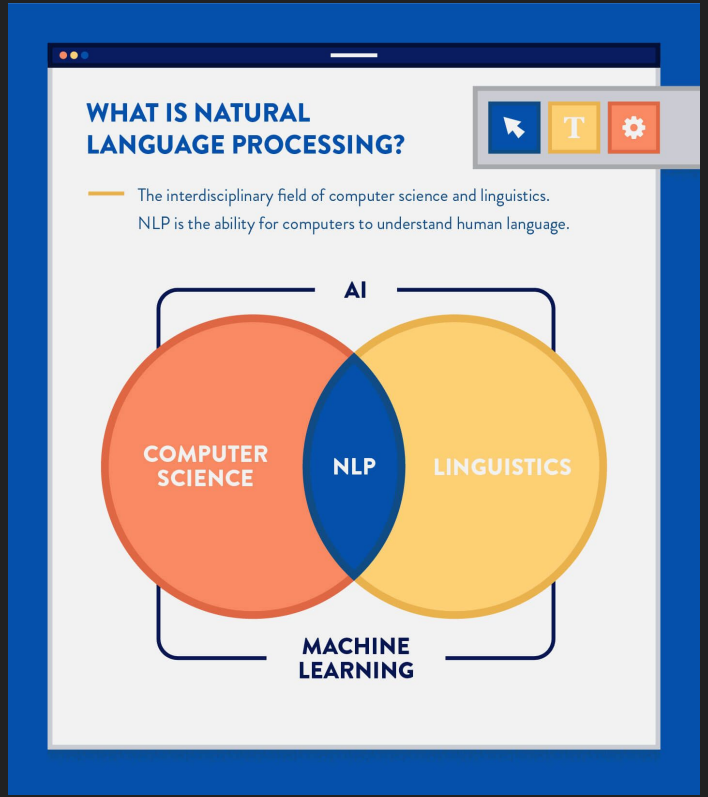
Matthew Graham, Liam Propst, Tyler Lewis, Lucas Gaudet

Introduction

- Natural Language Processing
- Sentiment Analysis
- Attention is All You Need
- The Transformer Architecture
- Our Project

Why Natural Language Processing?``````

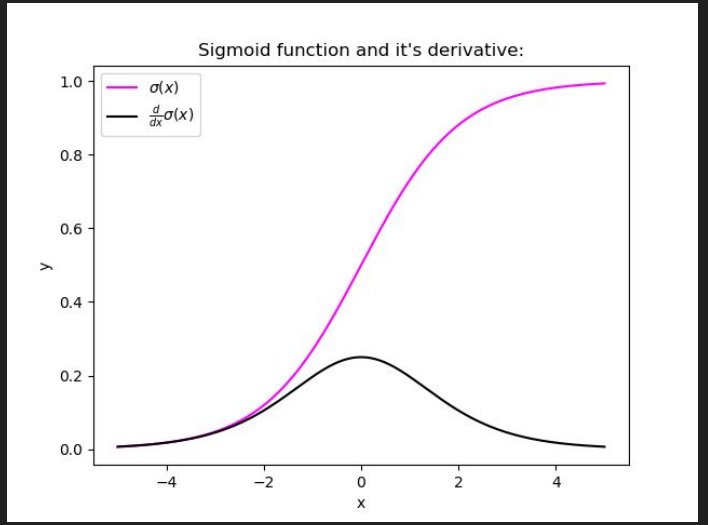
- Human to Machine Communication
- Sentiment Analysis
  - Identifying trends and patterns on contextual information
  - Ability to analyze and quantify the unquantifiable
- Applications
  - Research
  - Document Summarization
  - Automation
  - Translation



Literature Review: Attention is All You Need

Proposing Solution to Recurrent Neural Networks (RNNs)

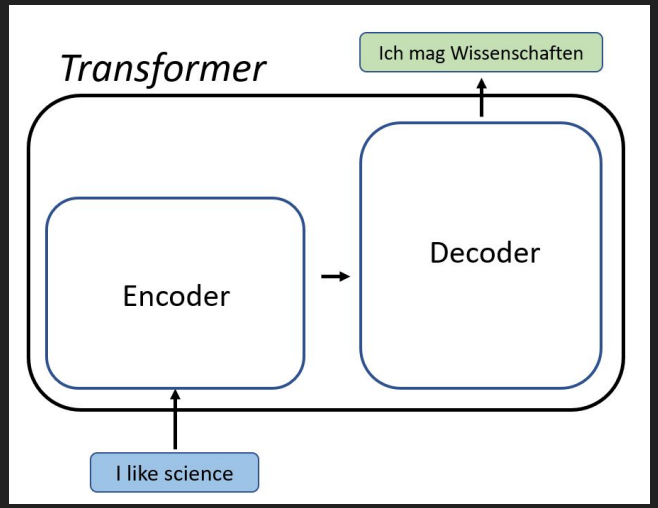
- Vanishing Gradient Problem
  - Gradients
    - Vector of partial derivatives for updating weights of information
    - Pivotal to machine learning algorithms
  - Gradients become infinitely near zero as moving through abundance of layers
  - Slows training process
- Introduction of the Transformer Model



Literature Review: Attention is All You Need

Solving the Vanishing Gradient Problem

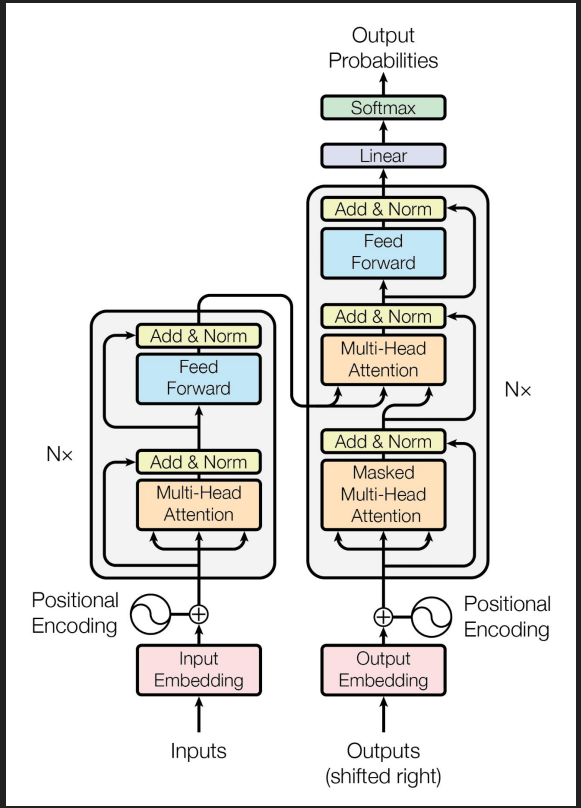
- Transformer Replaces RNNs with Self Attention Mechanisms
- Parallel Processing of Input Sequences
- Better Captures Long-Range Dependencies in Sequential Data
  - Natural Language Text
  - Pronoun Referencing
  - Subject-Verb Agreement
  - Named Entities



Literature Review: Attention is All You Need

Proposes the Transformer Model

- Encoder
- Decoder
- Multi-head self-attention mechanism
- Multi-head attention mechanism
- Position-wise feedforward network



Literature Review: Attention is All You Need

Applications of Attention in transformer architecture:

- Language modeling and prediction
- Language generation
- Speech recognition

Projects utilizing Attention in transformer architecture:

- BERT: Bidirectional Encoder Representations from Transformers (Google)
- Transformer-XL (Google)
- T5 Text to Text Transformer (Google)
- GPT-3 (OpenAI)

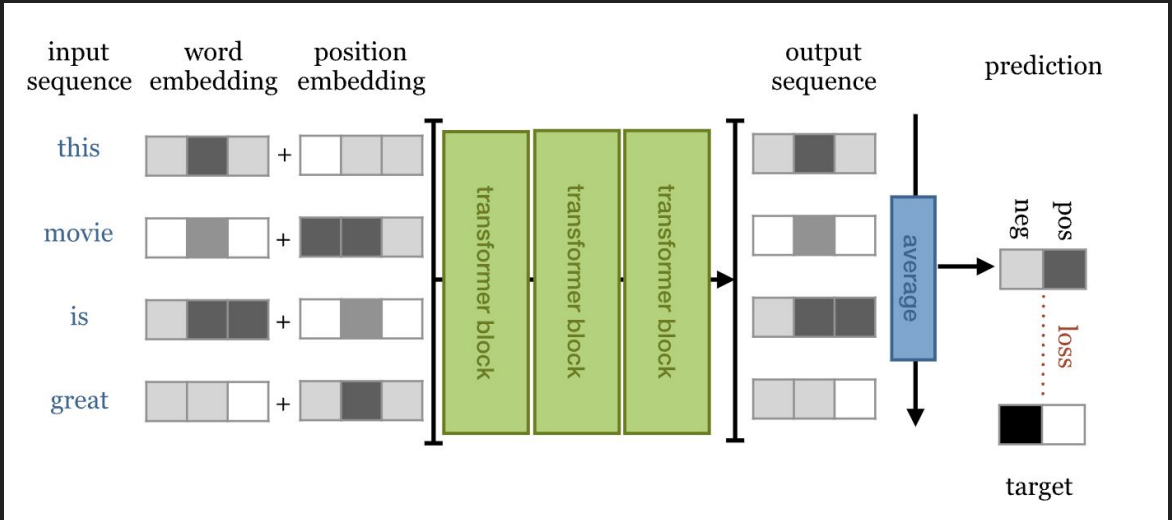
What is Self Attention?

- Allows the model to weigh the importance of different input elements (e.g., words in a sentence)
- Key steps:
  - Input embeddings: The input sequence is first converted into a set of continuous vector representations, or embeddings.
  - Linear transformations: For each input embedding, three linear transformations are applied to generate three different vectors: a Query vector (Q), a Key vector (K), and a Value vector (V).
  - Scaled dot-product attention
  - Value vector weighting
  - Output vector

Our Project Architecture

- The Transformer uses self-attention mechanism for generating output sequence by attending to different parts of input sequence.
- The architecture consists of two main parts: an encoder and decoder, both contain multiple Transformer blocks.
- Our code defines different modules: self-attention, Transformer block, encoder, decoder, and whole Transformer model.
- We then used these components to build a Sentiment Classifier model

Our Implementation: Sentiment Classifier




```
[liampropst@Liams-MBP Projects % python3 Sentiment.py                          ]
 Enter an input: This movie was really amazing. The plot moved so well and the charac
 ter development was stellar
 1: positive
 Enter an input: I didn't like how this movie was made
 0: negative
 Enter an input: Kurz is really awesome
 1: positive
 Enter an input:
```

Complications

- Long training times
  - Training of a transformer is a time-intensive task, and requires sophisticated GPU hardware
  - Limited training time/epoch runs to reasonably train our model
- Overfitting
  - Model learns the training data “too well” and leads to inaccuracy when given test samples
  - weight on positive sentiment

Future Development

- Running with larger data sets
  - Hardware and Software Considerations
  - Improving Accuracy
- Fine-tune the model
  - Updating parameters within the model
- Adapt transformer to other applications
  - Generic transformer applied to specific classifier model
  - Straightforward to adapt to other models and applications
- Compare and contrast with other models
  - Utilize the models already created

Responsibilities

- Matthew - Transformer Model Research, Presentation
- Lucas - Lead Researcher, Lead Transformer Developer
- Tyler - Documentation, run on Nvidia Docker server, Testing Loop
- Liam - Self Attention and Algorithm Research, Unit Testing

Resources Utilized

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems* (p./pp. 5998--6008),
- Transformers from Scratch | Peterbloem.Nl. https://peterbloem.nl/blog/transformers?fbclid=IwAR2chHBjhd6atKng0aawWUF4oIk5MwquYC9F7P85FQXMvVzrJnIOpVt3Hq4. Accessed 8 May 2023.
- https://github.com/cmparlettpelleriti/CPSC393ParlettPelleriti/blob/main/Lectures/TransformersII.pdf?fbclid=IwAR39CrHMit30BGyUT_LRDI0E-kzSRg87UTsTT2jzv8JL9ECr980rv9dopWA