Transformers

Transformers are a type of neural network architecture that transforms or changes an input sequence into an output sequence

Word embedding – creating vectors, which represents each word in numerical form it, can be integer or float or binary representation.

Static embedding - These are static representation of the words. These embedding retain the same meaning for the word in all the sentences even the sentence has different meaning as whole.

Contextual embedding – creating vectors which represents contextual meaning of the words. These are dynamic representation of the words. Unlike static embedding, which assign a fixed vector to each word, contextual embedding generate different vectors for the same word depending on its context, improving NLP tasks like sentiment analysis, more precise language translation etc.

Transformers architecture – There are two parts in the generic transformer architecture:-

1. Encoding
2. Decoding

In a standard Transformer architecture, the encoding process involves converting input sequences into a contextualized, numerical representation. The decoder to generate the output sequence then uses this representation. Here is a more detailed breakdown:

1. **Input**: The encoding process starts with an input sequence, which could be text, images, or other data formats.

2. **Embedding**: The input is first converted into numerical representations called embedding. This involves mapping each element of the input sequence (e.g., each word in a sentence) into a vector.

3. **Positional Encoding**: Since Transformers lack the sequential processing capability of RNNs; positional encoding is used to inject information about the order of elements in the input sequence. This helps the model understand the context of each element within the sequence.

4. **Encoder Layers**: The encoded input then passes through multiple identical encoder layers. Each encoder layer consists of two main components: Multi-Head Attention: This mechanism allows the model to attend to different parts of the input sequence simultaneously, capturing relationships and dependencies between elements. Feed-Forward Network: This network further processes the output of the attention mechanism, refining the representation.

5. **Contextualized Representation**: The output of the encoder layers is a contextualized representation of the input sequence, capturing its meaning and relationships between elements. This representation is then passed to the decoder for generating the output. In essence, the encoding part of a Transformer model transforms the input into a richer, contextualized numerical representation that the decoder can use to generate the desired output.

Here is a more detailed breakdown of the decoder's working:

1. **Input Embedding**: The decoder starts with an embedding layer that converts input tokens into numerical representations (embedding).

2. **Positional Encoding**: Positional encodings are added to the embedding to preserve the order of the input sequence.

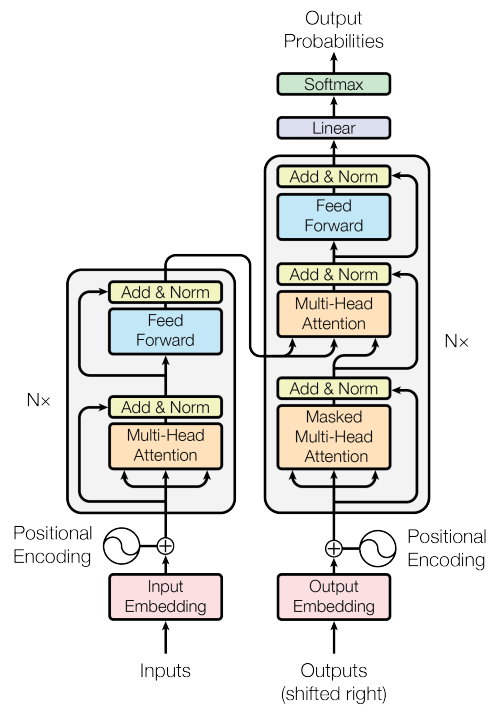
3. **Masked Self-Attention**: The decoder uses a masked self-attention mechanism to attend to previous tokens in the target sequence, preventing it from looking at future tokens.

4. **Encoder-Decoder Attention**: The decoder attends to the encoder's output, which allows it to incorporate information from the input sequence.

5. **Feed-Forward Network**: A feed-forward network processes each embedding vector independently, further refining the output.

6. **Output Prediction**: The decoder outputs a probability distribution over the vocabulary, and the token with the highest probability is selected as the next token in the output sequence. The decoder continues this process iteratively, generating one token at a time, until it reaches an end-of-sequence token. The decoder's ability to attend to different parts of the input and its own previous outputs enables it to generate coherent and contextually relevant sequences, making it a powerful tool for tasks like machine translation and text generation.

Mathematical Representation of Transforms generic architecture



**1. Encoder:**

The encoder processes the input sequence and converts it into a context-rich representation. The encoder itself is made up of a stack of identical layers, each containing two main components:

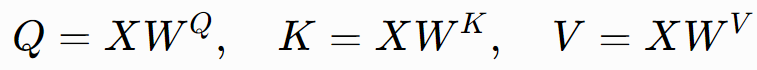
**a) Multi-Head Self-Attention Mechanism:**

The self-attention mechanism calculates attention scores for each word in a sequence concerning all other words in the sequence. This allows the model to weigh the importance of each word relative to the others.

Creation of Query, Key, and Value:

Each word (or token) in the input sequence is first embedded into a continuous vector space. These embedding are then linearly transformed to generate three different vectors: Query ((Q)), Key ((K)), and Value ((V)).

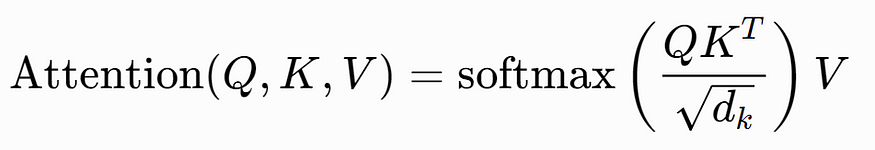
Mathematically, this can be represented as:



where:

* (X) is the input embedding matrix,
* (W^Q, W^K, W^V) are weight matrices learned during the training process.

The self-attention mechanism operates as follows:



where:

* (Q) (Query) is the matrix of query vectors,
* (K) (Key) is the matrix of key vectors,
* (V) (Value) is the matrix of value vectors,
* (d\_k) is the dimension of the key vectors.

**b) Position-Wise Feed-Forward Network:**

This component processes the output of the self-attention mechanism for each word independently through a series of linear transformations and activation functions. This helps to introduce non-linearity and further refine the word representations.

**2. Decoder:**

The decoder generates the output sequence by attending (focusing) on the relevant parts of the input sequence (using the encoded representations). Like the encoder, it has a stack of identical layers, each containing:

**a) Masked Multi-Head Self-Attention Mechanism:**

This mechanism is similar to the encoder’s self-attention but is masked to prevent the decoder from “cheating” by looking at the future words in the sequence during training.

**b) Encoder-Decoder Attention:**

This layer allows the decoder to attend to the encoder’s output representations, effectively aligning the input and output sequences.

**c) Position-Wise Feed-Forward Network:**

Similar to the encoder, this network further processes the attention outputs.

10 Projects on Transformers

For beginners looking to work with transformer models, 10 great projects include **sentiment analysis, text summarization, chatbot development, language translation**, and more. These projects leverage the power of transformers for natural language processing tasks, offering hands-on experience with this popular neural network architecture. [[1](https://dev.to/lokesh_singh/top-20-machine-learning-projects-to-boost-your-resume-18k6#:~:text=Chatbot%20Development%20%F0%9F%92%AC%20%2D%20Difficulty:%207/10.%20Create,to%20adjust%20responses%20based%20on%20user%20mood!), [2](https://www.projectpro.io/article/artificial-intelligence-project-ideas/461#:~:text=Translator%20App.%20If%20you%20are%20interested%20in,importance%20of%20each%20word%20in%20a%20sentence.), [3](https://www.neurond.com/blog/automatic-text-summarization-system-using-transformers), [4](https://colab.research.google.com/github/jalammar/jalammar.github.io/blob/master/notebooks/Trax_TransformerLM_Intro.ipynb#:~:text=TransformerLM%20Quick%20Start%20and%20Guide%20Language%20models,(e.g.%20machine%20translation%2C%20sentiment%20analysis%2C%20chatbots%2C%20summarization).), [5](https://fall-2023-python-programming-for-data-science.readthedocs.io/en/latest/Lectures/Theme_3-Model_Engineering/Lecture_20-Transformer_Networks/Lecture_20-Transformer_Networks.html#:~:text=Consequently%2C%20Transformers%20(%20Transformer%20Model%20)%20are,Network%20(%20Artificial%20Neural%20Networks%20)%20architecture.), [6](https://datasciencedojo.com/blog/transformer-models/#:~:text=Transformer%20models%20are%20a%20type%20of%20deep,machine%20translation%2C%20text%20summarization%2C%20and%20question%20answering.)]

**Here's a breakdown of 10 beginner-friendly transformer projects:**

1. **Sentiment Analysis:** Analyze the sentiment (positive, negative, or neutral) of text data, such as movie reviews or social media posts. This project introduces basic text processing and transformer models for classification. ProjectPro [[7](https://chintanonweb.medium.com/from-beginner-to-expert-10-must-try-programming-projects-with-step-by-step-examples-68af9fe7d230#:~:text=Difficulty%20Level:%20Intermediate%20Dive%20into%20the%20exciting,neutral)%20of%20a%20given%20sentence%20or%20document.), [8](https://www.appliedaicourse.com/blog/data-science-projects/#:~:text=3.%20Sentiment%20Analysis%20Project%20Objective:%20Analyze%20textual,reviews%20or%20social%20media%20posts.%20Key%20Points:), [9](https://link.springer.com/chapter/10.1007/978-981-97-4533-3_3#:~:text=Text%20Classification:%20Transformer%20models%20can%20be%20employed,label%20the%20text%20into%20various%20sentiment%20categories.), [10](https://www.youtube.com/watch?v=5tloWG2Jelo#:~:text=Explore%20the%20fundamentals%20of%20text%20classification%20using,analysis%2C%20spam%20detection%2C%20and%20more.%20This%20content)]
2. **Text Summarization:** Create a model that can automatically generate concise summaries of text documents. This involves using transformer models for sequence-to-sequence tasks and extracting the most important information. ProjectPro [[11](https://timespro.com/blog/top-25-data-science-project-ideas-and-topics-for-beginners-and-experts#:~:text=Text%20Summarisation:%20Create%20a%20model%20that%20automatically,to%20extract%20key%20information%20quickly%20and%20efficiently.), [12](https://link.springer.com/chapter/10.1007/978-981-97-8019-8_6#:~:text=Text%20summarization%20is%20a%20crucial%20task%20in,tools%20for%20both%20extractive%20and%20abstractive%20summarizations.), [13](https://www.skillshare.com/en/classes/hugging-face-course-for-beginners/1793198890/projects#:~:text=The%20text%20to%20text%20generation%20class%20is,translation%20using%20the%20Hugging%20Phase%20Transformers%20library.), [14](https://link.springer.com/article/10.1007/s11042-023-17275-9#:~:text=The%20Transformer%20employs%20self%2Dattention%20mechanisms%2C%20which%20can,relevant%20information%20from%20the%20entire%20text%20simultaneously.)]
3. **Chatbot Development:** Build a conversational chatbot that can engage in natural-sounding conversations with users. This project utilizes transformers for question answering and generating responses in a chatbot context. Simplilearn.com [[15](https://www.kaggle.com/datasets/govindaramsriram/chatbot-dataset-for-transformers#:~:text=Description%20The%20ChatBot%20Dataset%20for%20Transformers%20is,text%20format%2C%20focusing%20on%20natural%2C%20human%2Dlike%20responses.), [16](https://www.upgrad.com/blog/exciting-deep-learning-project-ideas-for-beginners/#:~:text=Building%20a%20ChatBot%20This%20project%20focuses%20on,like%20ambiguity%2C%20multilingual%20queries%2C%20and%20system%20scalability.), [17](https://www.linkedin.com/pulse/unveiling-diversity-transformer-based-language-models-swagat-panda-x4fec#:~:text=Dialogue%20Generation:%20Transformer%2Dbased%20decoder%20models%20are%20utilized,relevant%20responses%20to%20user%20queries%20or%20prompts.)]
4. **Language Translation:** Develop a model capable of translating text from one language to another. This project involves using transformer models for machine translation, a well-established NLP application. Analytics Vidhya [[2](https://www.projectpro.io/article/artificial-intelligence-project-ideas/461#:~:text=Translator%20App.%20If%20you%20are%20interested%20in,importance%20of%20each%20word%20in%20a%20sentence.), [18](https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-project-ideas#:~:text=A%20Language%20Translation%20Model%20project%20aims%20to,natural%20language%20processing%20and%20machine%20translation%20techniques.), [19](https://www.analyticsvidhya.com/blog/2023/08/nlp-projects/#:~:text=The%20machine%20translation%20system%20will%20be%20able,information%20more%20accessible%20to%20a%20worldwide%20audience.), [20](https://medium.com/nerd-for-tech/easy-guide-to-transformer-models-6b15c103bfcf#:~:text=It%20(%20The%20Transformer%20model%20)%20has,tasks%2C%20including%20machine%20translation%20and%20sentiment%20analysis.), [21](https://www.rapidinnovation.io/post/what-is-a-transformer-model-331d1#:~:text=To%20explore%20more%20about%20output%20generation%20techniques,studies%20on%20various%20NLP%20tasks%20and%20models.)]
5. **Fake News Detection:** Identify and classify fake news articles based on their content. This project utilizes transformer models for text classification and pattern recognition in potentially misleading text. Simplilearn.com [[22](https://medium.com/@programmingAi/exploring-the-most-fascinating-python-machine-learning-projects-45fd83e3bbac), [23](https://medium.com/towards-data-science/5-solved-end-to-end-data-science-projects-in-python-acdc347f36d0), [24](https://ieeexplore.ieee.org/document/9558906/#:~:text=In%20this%20article%2C%20we%20propose%20an%20improved,resource%20and%20used%20to%20train%20the%20models.), [25](https://www.gptechblog.com/generative-ai-models-transformers-diffusion-models/#:~:text=Hallucination%20%2D%20When%20used%20to%20generate%20text%2C,sometimes%20toxic)%20responses%20in%20a%20convincing%20manner.), [26](https://www.themoonlight.io/en/review/how-can-transformers-predict-pseudo-random-numbers#:~:text=The%20authors%20explore%20whether%20Transformers%20(%20transformer,of%20LCGs%20with%20varying%20parameters%20and%20moduli.)]
6. **Grammar Correction:** Create a system that can automatically correct grammatical errors in text. This project utilizes transformer models for language modeling and error detection in text. Intellipaat [[27](https://intellipaat.com/blog/nlp-projects-for-beginners/#:~:text=Grammar%20Autocorrector.%20The%20grammar%20autocorrector%20project%20actively,parsing%2C%20part%2Dof%2Dspeech%20tagging%2C%20and%20error%20correction%20algorithms.), [27](https://intellipaat.com/blog/nlp-projects-for-beginners/#:~:text=Grammar%20Autocorrector.%20The%20grammar%20autocorrector%20project%20actively,parsing%2C%20part%2Dof%2Dspeech%20tagging%2C%20and%20error%20correction%20algorithms.)]
7. **Question Answering:** Build a system that can answer questions based on a given context. This project involves using transformer models for understanding context and generating relevant answers. DEV Community [[2](https://www.projectpro.io/article/artificial-intelligence-project-ideas/461#:~:text=Translator%20App.%20If%20you%20are%20interested%20in,importance%20of%20each%20word%20in%20a%20sentence.), [28](https://bayramblog.medium.com/hugging-face-and-dify-ai-a-friendly-guide-to-ai-magic-80ffb1d52a5d#:~:text=Question%2DAnswering:%20Build%20systems%20that%20can%20answer%20questions%20based%20on%20context.), [29](https://www.fabricatedknowledge.com/p/ai-foundations-part-1-transformers#:~:text=Users%20can%20adapt%20the%20model's%20deep%20and,in%20the%20question%20to%20get%20better%20performance.), [30](https://www.stratascratch.com/blog/30-project-ideas-to-showcase-your-machine-learning-skills/#:~:text=Machine%20Learning%20Project%20Idea%20%2324:%20Question%2DAnswering%20In,given%20a%20textual%20question%20and%20the%20context.)]
8. **Spam Email Detection:** Develop a model that can identify and classify spam emails. This project utilizes transformer models for text classification and pattern recognition in spam content. Simplilearn.com [[31](https://medium.com/@kattilaxman4/simple-and-practical-data-science-topics-for-your-next-academic-thesis-4064e3f863ec#:~:text=The%20Objective:%20The%20objective%20of%20this%20topic,and%20is%20able%20to%20detect%20spam%20emails.), [32](https://medium.com/pythoneers/lets-build-a-transformer-with-tensorflow-part-two-528ef7068cc6#:~:text=Building%20a%20transformer%20using%20TensorFlow%20equips%20you,like%20classifying%20emails%20as%20spam%20or%20not.), [33](https://medium.com/pythoneers/lets-build-a-transformer-with-tensorflow-part-two-528ef7068cc6#:~:text=Building%20a%20transformer%20using%20TensorFlow%20equips%20you,like%20classifying%20emails%20as%20spam%20or%20not.)]
9. **Product Review Analysis:** Analyze product reviews to identify key themes, customer sentiment, and potential issues. This project leverages transformer models for sentiment analysis and topic modeling in text. Simplilearn.com [[34](https://link.springer.com/chapter/10.1007/978-981-97-4533-3_3#:~:text=This%20research%20paper%20provides%20a%20comprehensive%20survey,translation%2C%20document%20classification%2C%20question%2Danswering%2C%20and%20many%20more.)]
10. **Code Generation:** Use transformers to generate code from natural language descriptions. This is a more advanced project but can be a good challenge for those looking to push their skills. ProjectPro [[35](https://www.novelvista.com/blogs/generative-ai/generative-ai-projects-for-beginners-complete-guide)]

These projects offer a range of difficulty and can be scaled to suit different skill levels. They provide hands-on experience with transformer models and can be a great way to learn and grow in the field of NLP and deep learning. ProjectPro [[1](https://dev.to/lokesh_singh/top-20-machine-learning-projects-to-boost-your-resume-18k6#:~:text=Chatbot%20Development%20%F0%9F%92%AC%20%2D%20Difficulty:%207/10.%20Create,to%20adjust%20responses%20based%20on%20user%20mood!), [36](https://clslearn.com/articles/python-project/#:~:text=Projects%20can%20be%20tailored%20to%20different%20skill,levels%20as%20you%20tackle%20increasingly%20complex%20challenges.), [37](https://scholar.harvard.edu/binxuw/classes/machine-learning-scratch/materials/transformers#:~:text=We%20prepared%20this%20series%20of%20jupyter%20notebooks,their%20architecture%20to%20the%20training%20and%20usage.), [38](https://www.lyzr.ai/glossaries/transformer-models/#:~:text=Common%20Uses%20and%20Applications%20of%20Transformer%20Models,by%20enabling%20unprecedented%20performance%20in%20various%20applications.)]

This video will guide you through coding a transformer from scratch on PyTorch: <https://www.youtube.com/watch?v=ISNdQcPhsts>

*Generative AI is experimental.*