**Deep Learning in Question and Answering**

Deep Learning is a subset domain under machine learning which involves neural networks. It includes the input layer to the hidden layers and finally the output layer. This field has numerous applications like Image recognition and classification tasks, language detection, and translation through speech or text inputs, Recommendation systems, Generative tasks as like text generation with the help of Large Language Models (LLMs), and many more.

A diagram of machine learning

Description automatically generated

Let us discuss various approaches to how deep learning was implemented for question and answer. A question-answering system can be achieved by a query response-based system dialog system or chatbot. The world’s first chatbot was Eliza, developed during 1960s an MIT researcher named Joseph Weizenbaum when Neural networks and Neural Information Processing systems were in the experimentation was conducted by IBM and Georgetown University with the ambition of creating a system of machine translation.

A screen shot of a computer screen

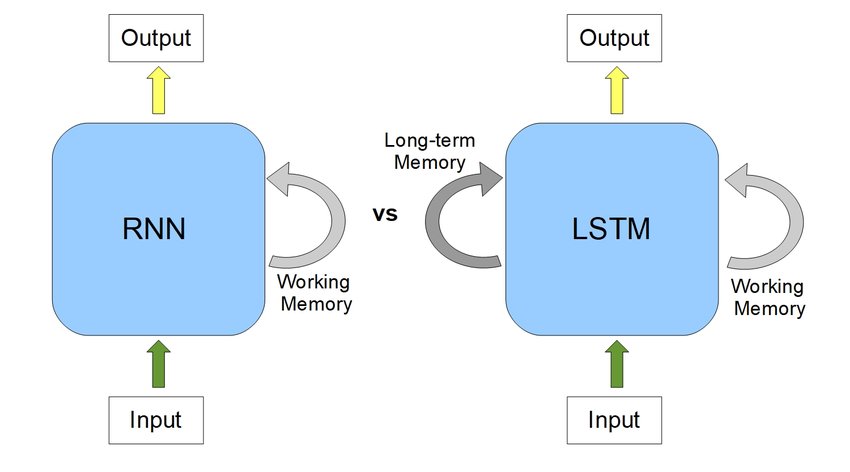
Description automatically generated

**Neural Networks :**

The Neural Networks are capable of text classification, and gathering features but in what the best way can it be expressed in simple sentences which is how to produce a simple sequence of words understandable to users? How can a Neural Network work in terms of different languages and their slang? The neural networks learn the patterns of input data by encoding the text information to numerical embeddings, these are fed upon the neural layers of the model.

Recurrent Neural Networks (RNNs) can capture sequential related data, unlike feed-forward neural networks in sequential tasks. It is possible due to the maintenance of a hidden state unit that captures information and updates based on the previous step. After training through embeddings, the final hidden layer includes all the condensed contexts that are used for Q&A. Thus, RNNs were preferred for text processing-related tasks, but what about capturing greater lengths of textual information and sequence prediction-related tasks? The RNNs cannot handle such lengthy texts due to vanishing gradient problems (larger weight adjustments leading to unstable training) as RNNs are designed to handle sequential data.

LSTMs or Long Short-Term Memory networks were used as sequence-to-sequence machine translation. It includes a memory unit and gating mechanism that can store and retrieve long information. The drawback is that this Neural Network is much more complex than RNN. Even though it solves long-range capturing, it requires larger computational power and resources to perform such tasks.



Is it necessary to capture all the available context to answer the question, how about capturing only the relevant ones or the contextually important ones? At such an assumption, the “Attention” mechanism was introduced to overcome problems like gradient vanishing, capturing long-range textual information, and providing optimal solutions to save computational resources.

**Attention mechanism:**

In 2017, at the Neural Information Processing Systems (NeuIPS) Conference, “Attention is All You Need” Ashish Vaswani, a computer scientist in the deep learning field and former worker at Google Brain introduced a paper-based transformer. The transformer architecture is based on this term named – ‘Attention.’ Now, what is this Attention?

“Attention” defines the measurement of the selection of important tokens among different input data elements which can impact the network predictions the output or simply the task. The selection plays a vital role in improving network performance.

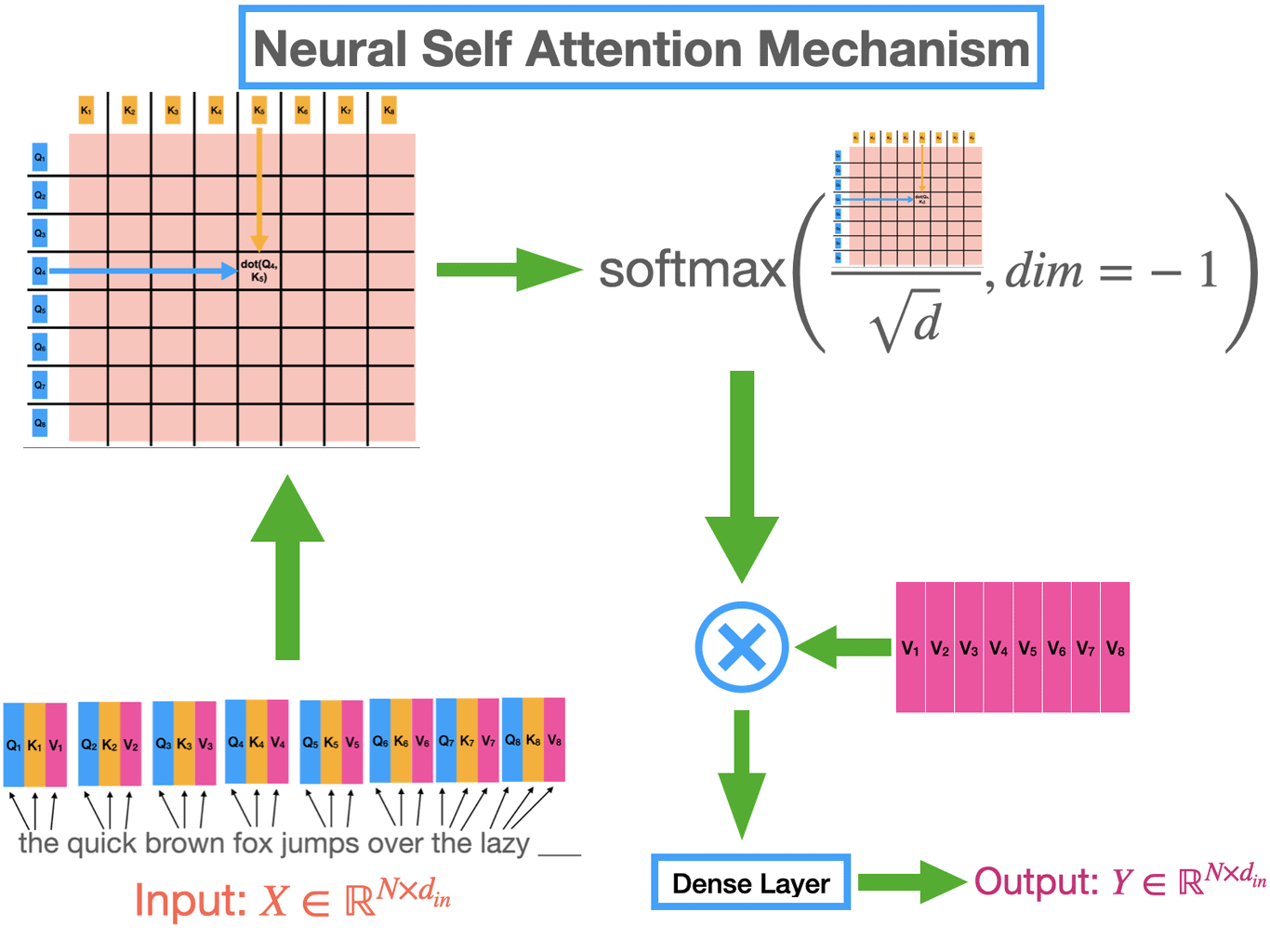
A close-up of a computer screen

Description automatically generated

The model selectively weighs around the various parts of the input data based on the importance of words with a given task which the model has either to decide, or we can provide the pre-assumed importance representations in the form of weights. The 2017 “Attention” paper introduced the attention mechanism which can be done in a parallel manner across the layers in the learning model.

How does it work then? The mechanism is carried out by calculating the attention score i.e., the similarity between a query and a key. Here the query defines the attention or the context to be focused upon, the key defines the embedded word tokens or the data that we had fed the model upon which are compared to the embedded query form inside the model. Then the control, checks the similarity between these two, as a part of the prediction of upcoming words an activation value is calculated which fires the prediction to be one or zero which is seen in Perceptron, an old friend of the neural network.

The associated weights to this activation at the output layers are normalized by an activation function- SoftMax (preferred) was used in the ‘Attention’ paper. This also improves the memory management utilized by the model.



This ‘Attention’ overcomes the inability to capture long-range sequences which the RNN models failed to capture, in proportion to other problems like numerical overflow (larger datatype constraint as computer memory cannot handle it) or gradient vanishing problem loss.

Overall, attention helps the learning model identify the keywords present in our prompt, by assigning relevance score among other words in the query. After determining the keyword, the one with the highest score, it searches from the available source or the knowledge that could best explain this context, in that takes the closest answer and the output generated according to the prompt template the expertise device.

With the introduction of ‘Attention’, the transformer-based deep learning model was introduced which contains an encoder (positional based), Attention layer, feed-forward network, and decoder. The transformers are pre-trained over a large scale of data thus the large language model (LLM) was given. Each LLM is initially trained on a semantic, well-structured dataset from a wide range of domains like literature, medicines, etc. which learns the basic sentence formation, grammar, and the ability to generate non-repeating content upon new call, such models are called the Foundational Model. Examples of such models are: GPTs from openai, Llama from meta and many more.

Based on the training, the transformer model is trained upon large datasets and the learned weights are assumed to be in parameters, larger the parameter more the scope of learning it was exposed to. There’s a drag about large models being more in parameters ranging from few million to hundreds of billion, the response time is way slow. Also, tech giants like Microsoft, introduced Small Language Model (SML) which have a few million parameters but well trained on carefully selected, well-refined dataset giving on-par performance with LLMs, such examples are phi models, mini-llama etc.

The Language model can also be down streamed to a selected domain we are trying to tune a foundational model to only a text-to-SQL-based task which is utilized by various enterprises and small other companies as an integration to their main application. Such tasks are called fine-tuning.

But fine-tuning was not a long-term solution while dealing with multiple kinds of data like images, voice, text, etc. Currently using the principles of transfer learning, we can transfer the learned patterns from one model to another, trying to build a model useful for multiple tasks, such models are multi-models where the current research and developments are taking place.

There are many such real-world language models diversified and implemented under different use cases as you can see in below diagram.

