Summary and Highlights: Transformers in Keras

Congratulations! You have completed this module. At this point in the course, you know:

- The transformer model consists of two main parts: the encoder and the decoder.
- Both the encoder and decoder are composed of layers that include self-attention mechanisms and feedforward neural networks.
- Transformers have become a cornerstone in deep learning, especially in natural language processing.
- Understanding and implementing transformers will enable you to build powerful models for various tasks.
- Sequential data is characterized by its order and the dependency of each element on previous elements.
- Transformers address the limitations of recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) by using self-attention mechanisms, which allow the model to attend to all positions in the input sequence simultaneously.
- Transformers' versatile architecture makes them applicable to a wide range of domains, including computer vision, speech recognition, and even reinforcement learning.
- Vision transformers have shown that self-attention mechanisms can be applied to image data.
- By converting audio signals into spectrograms, transformers can process the sequential nature of speech data.
- Transformers have found applications in reinforcement learning, where they can be used to model complex dependencies in sequences of states and actions.
- Time series data is a sequence of data points collected or recorded at successive points in time.
- By leveraging the self-attention mechanism, transformers can effectively capture long-term dependencies in time series data, making them a powerful tool for forecasting.
- The key components of the transformer model include an embedding layer, multiple transformer blocks, and a final dense layer for output prediction.

- Sequential data is characterized by its temporal or sequential nature, meaning that the order in which data points appear is important.
- TensorFlow provides several layers and tools specifically designed for sequential data. These include:
    - RNNs
    - LSTMs
    - Gated recurrent units
    - Convolutional layers for sequence data (Conv1D)
- Text data requires specific preprocessing steps, such as tokenization and padding.
- TensorFlow's TextVectorization layer helps in converting text data into numerical format suitable for model training.