Summary and Highlights

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

- Word2Vec is the short form for “word to vector.” It is a group of models that produce word embeddings or vectors, which are numerical representations capturing the essence of words.
- A neural network model consists of an input layer, an embedding layer, and an output layer.
- Words are fed into an embedding layer, which interacts with an output layer to predict context words.
- The number of neurons in both input and output layers corresponds to the vocabulary size, while the embedding layer’s size, which is chosen by the user, defines the word vector dimensions.
- The continuous bag of words, or CBOW model, utilizes context words to predict a target word and generate its embedding.
- Skip-gram model predicts surrounding context words from a specific target word. It operates in contrast to the CBOW model.
- The Skip-gram model simplifies the task by predicting one context word at a time from a target word.
- The sequence generation function in the creation of the skip-gram model is identical to CBOW but with a switch in the order of the target and context. This breakdown allows you to work with the full context in smaller parts.
- While training the parameters, you define the learning rate, loss function, optimizer, and learning rate scheduler.
- Once the model is trained, you can retrieve the weights, which are the actual word embeddings.
- Word embeddings utilize Stanford pretrained GloVe or global vectors that leverage large-scale data for word embeddings.
- Sequence-to-sequence models within generative AI are used in machine translation, such as converting English phrases into French.

- Sequence-to-label tasks take multiple inputs to produce a single label, useful in document classification.
- Label-to-sequence tasks generate a full sequence from a single input, as seen in generative models for image creation.
- RNN is a type of artificial neural network that uses sequential or time series data.
- It is designed to remember past information and use it to influence future decisions.
- RNNs only remember short-term information and are challenging to train. Two popular RNN enhancements are Gated Recurrent Units (or GRUs) and Long Short-Term Memory (or LSTMs).
- In general, sequence-to-sequence models are more difficult to train than RNNs due to several factors.
- In model training, the aim is to minimize the cross-entropy loss by summing the output of the predicted outcomes with the actual labels.
- Procedure for training sequence-to-sequence models:
    - Initialize the model in training mode to activate essential layers like dropout, ensuring optimal performance during training.
    - Iterate through training data batches, assigning input (src) and target (trg) sequences to the correct device.
    - Generate predictions by output.
    - Reshape the output tensor, which aligns the rows and columns correctly for loss calculation.
    - Calculate the average loss per batch after processing all batches.
- Performing translations in sequence models requires complex functions for making predictions.
- RNNs can be used to create sequence-to-sequence models that receive one sequence as input and generate another sequence as output.
- The encoder-decoder architectures are introduced so that the sequences do not necessarily require to be of the same length.
- Encoder is a series of RNNs that process the input sequence individually, passing their hidden states to their next RNN.

- The last hidden state, context, is passed to the Decoder module.
- The decoder module, similarly, is a series of RNNs that autoregressively generates the translation as one token at a time.
- The embedding layer transforms the input token into an embedded vector, which traverses through the RNN cell to produce the hidden state.
- Perplexity is a metric and a precious tool for evaluating the efficiency of LLMs and GenAI models.
- In Perplexity, you can employ cross-entropy loss to measure the discrepancy between the predicted and actual distribution.
- Perplexity is calculated as an exponent of the loss obtained from the model.
- Perplexity provides an overall measure of model performance but doesn't capture the nuances of generated text quality.
- In machine translation, precision measures the accuracy of the generated translation, whereas recall measures the completeness of the generated translation.
- F1-Score is a harmonic mean of precision and recall that is used to judge the performance of a model based on them.
- In NLP, there are several popular libraries that provide implementations of various evaluation metrics, such as NLTK and PyTorch libraries.