

Summary and Highlights

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

- One-hot encoding converts categorical data into feature vectors.
- The bag-of-words representation portrays a document as the aggregate or average of one-hot encoded vectors.
- When you feed a bag-of-words vector to a neural network's hidden layer, the output is the sum of the embeddings.
- The Embedding and EmbeddingBag classes are used to implement embedding and embedding bags in PyTorch.
- A document classifier seamlessly categorizes articles by analyzing the text content.
- A neural network is a mathematical function consisting of a sequence of matrix multiplications with a variety of other functions.
- The Argmax function identifies the index of the highest logit value, corresponding to the most likely class.
- Hyperparameters are externally set configurations of a neural network.
- The prediction function works on real text that starts by taking in tokenized text. It processes the text through the pipeline, and the model predicts the category.
- A neural network functions via matrix and vector operations, called learnable parameters.
- In neural network training, learnable parameters are fine-tuned to enhance model performance. This process is steered by the loss function, which serves as a measure of accuracy.
- The prediction function works on real text that starts by taking in tokenized text. It processes the text through the pipeline, and the model predicts the category.
- Cross-entropy is used to find the best parameters.
- For unknown distribution, estimate it by averaging the function applied to a set of samples. This technique is known as Monte Carlo sampling.
- Optimization is used to minimize the loss.

- Generally, the data set should be partitioned into three subsets: training data for learning, validation data for hyperparameter tuning, and test data to evaluate real-world performance.
- The training data is split into training and validation, and then data loaders are set up for training, validation, and testing.
- Batch size specifies the sample count for gradient approximation, and shuffling the data promotes better optimization.
- When you define your model, `init_weights` helps with optimization.
- To train your loop:
 - Iterate over each epoch
 - Set the model to training mode
 - Calculate the total loss
 - Divide the data set into batches
 - Perform gradient descent
 - Update the loss after each batch is processed