Part 1: Deep Learning-Based Recommendation

Research Paper: Wide & Deep Learning for Recommender Systems

The paper by Heng-Tze Cheng et al. presents a method called Wide & Deep learning, which combines the benefits of memorization and generalization in recommender systems. The authors argue that while generalized linear models with nonlinear feature transformations are effective for large-scale regression and classification problems with sparse inputs, they require more feature engineering effort for generalization. On the other hand, deep neural networks can generalize better to unseen feature combinations through low-dimensional dense embeddings learned for sparse features, but they can over-generalize and recommend less relevant items when user-item interactions are sparse and high-rank. To address these issues, the authors propose a Wide & Deep learning framework that jointly trains wide linear models and deep neural networks. The wide component is a generalized linear model with feature transformations, while the deep component is a feed-forward neural network with embeddings. The two components are combined using a weighted sum of their output log odds, which is then fed to one common logistic loss function for joint training. The main contributions of the paper include:

The Wide & Deep learning framework for jointly training feed-forward neural networks with embeddings and linear model with feature transformations for generic recommender systems with sparse inputs. The implementation and evaluation of the Wide & Deep recommender system productionized on Google Play, a mobile app store with over one billion active users and over one million apps. The authors show that the Wide & Deep framework significantly improves the app acquisition rate on the mobile app store while satisfying the training and serving speed requirements. They also discuss the limitations of cross-product transformations and the challenges of learning effective low-dimensional representations for queries and items when the underlying query-item matrix is sparse and high-rank.

The key components of the Wide & Deep learning model are:

1. Wide Component: A generalized linear model that includes raw input features and transformed features. The most important transformation is the cross-product transformation, which captures interactions between binary features and adds nonlinearity to the model. The wide component is trained using a logistic regression model.
2. Deep Component: A feed-forward neural network that converts sparse, high-dimensional categorical features into low-dimensional and dense real-valued vectors, known as embedding vectors. These embedding vectors are then fed into the hidden layers of the neural network. The deep component is trained using a neural network with ReLU activation functions.
3. Joint Training: The wide and deep components are combined using a weighted sum of their output log odds, which is then fed to one common logistic loss function for joint training. This allows the model to optimize both the wide and deep parts simultaneously, achieving both memorization and generalization.
4. Model Structure: The model structure includes an input layer that takes in training data and vocabularies, a wide component with cross-product transformations, a deep part with embedding vectors, and a logistic output unit.

**Pytorch Widedeep Library:**

The pytorch-widedeep library is a flexible package for building multimodal deep learning models that can combine tabular data, text, and images using wide and deep architectures. It is based on the Wide and Deep algorithm proposed by Google

**Key Features:**

1. **Modular Design:** The library provides separate model components for handling different data types - wide, deeptabular, deeptext, and deepimage. These can be combined in various ways to build complex multimodal models.
2. **Deeptabular Models:** The library offers three deeptabular models of increasing complexity - TabMLP, TabResNet, and TabTransformer. These allow applying deep learning to tabular data
3. **Text and Image Support:** The library supports integrating text and image data through the deeptext and deepimage components, which can use various neural network architectures like RNNs and transformers

In our wide-deep based recommendation system, the wide columns include : {"Book-Title", "Book-Author", "Publisher", "Year-Of-Publication", "City", "Age" }, the crossed columns include : {("Book-Title", "Book-Author"), ("City", "Book-Author")}. In our dataset we embed the following columns {"Publisher", "Book-Title", "Book-Rating", "Book-Author", "Year-Of-Publication", "City", "Country"}. We have {“Age”} as continuous column and book ratings are the target feature.

We use TabMLP model which involves embeddings representing categorical features concatenated with continuous features, which are then passed through a Multi-Layer Perceptron (MLP)

