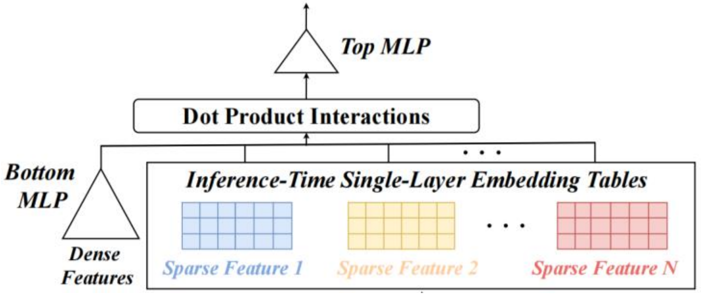
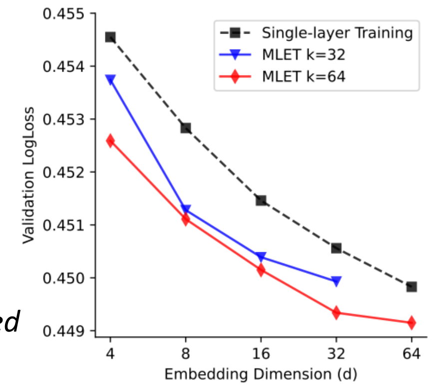
Enhancing Cross-Category Learning in Recommendation Systems with Multi-Layer Embedding Training

Zihao Deng, Benjamin Ghaemmaghami, Ashish Kumar Singh, Benjamin Cho, Leo Orshansky, Mattan Erez, Michael Orshansky

Coventional Embedding Training (Single-Layer Embedding)

larger embedding --> higher performance

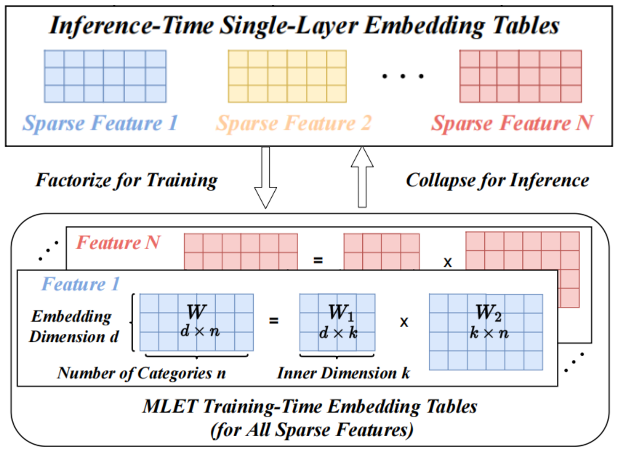




Conventional training: each sparse feature is represented by a single-layer $W \in \mathbb{R}^{d \times n}$ (n: # of embedding items)

DLRM on Criteo-Kaggle

MLET: Overparameterized Multi-Layer Embedding Training



MLET uses a two-layer architecture that factorizes the embedding table W in terms of W_1 and W_2.

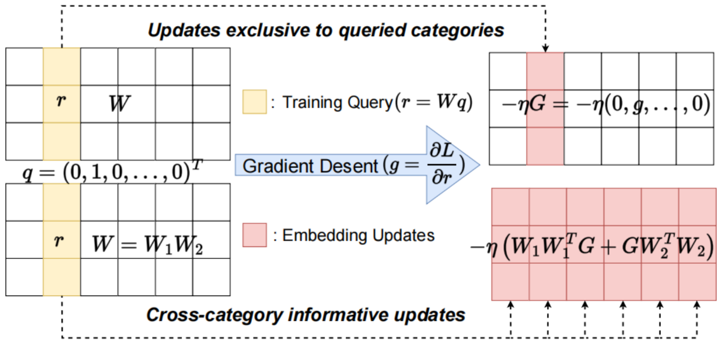
$$W = W_1 W_2$$

($W_1 W_2$ are trained jointly)

d: a hyperparameter that represents the inner dimension of factorization.

Single-layer: sparse updates. Only queried items are updated.

MLET: dense updates. Information from queried items is also used to update non-queried ones (cross-category information).



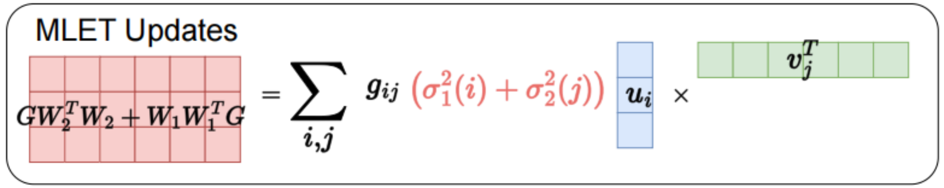
MLET Theory: Re-weighting Mechanism

MLET creates an re-weighting effect: the re-weighting factor $\sigma_1{}^2(i) + \sigma_2{}^2(j)$ boosts the update in directions proven to be important based on earlier training.

Reweighting of Single Layer Updates

$\sigma_1(i)$: i-th singular value of W_1

$\sigma_2(j)$: j-th singular value of W_2



$\sigma_1(i) / \sigma_2(j)$ reflects the importance of u_i / v_j, to the learned W_1 / W_2.

Experimental Results (Avazu)