# COMP4332/RMBI4310

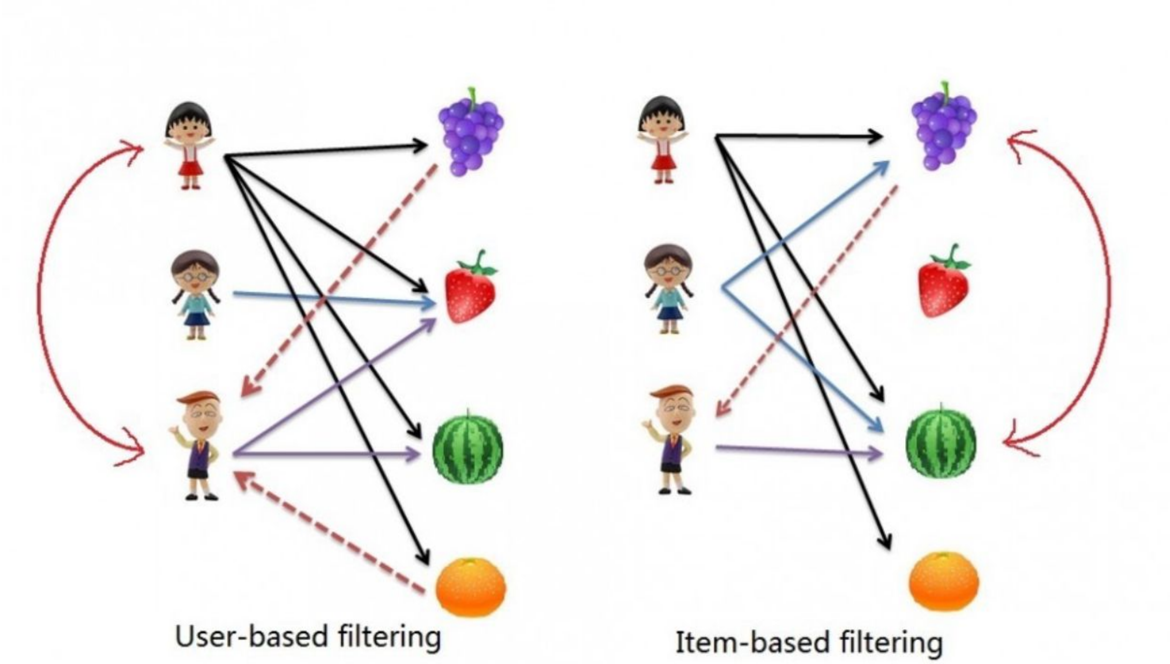
## Big Data Mining and Management
## Advanced Data Mining for Risk Management and Business Intelligence (2025 Spring)

## Tutorial 4: Neural CF and Wide & Deep Learning

TA: Chunyang LI (cliei@connect.ust.hk)

# 1 Neural CF

# Recap: user CF / item CF



# Limitations of traditional matrix factorization

- Linear Assumption
  - MF models user-item interactions as a dot product of latent vectors.
  - Fails to capture complex, non-linear patterns in real-world data.
- Fixed Interaction Function
  - Dot product restricts flexibility.
- Shallow Representation Learning
  - Performs poorly with sparse data.
  - Latent vectors are learned via linear decomposition, limiting expressiveness.

# Neural Collaborative Filtering (Neural CF) [1]

Core:

- Replace dot product in MF with a neural network to learn arbitrary interaction functions.
- Key Components:
  - User/Item Embeddings: convert IDs into dense vectors.
  - Interaction Layer: Neural network (e.g., MLP) learns non-linear user-item relationships.

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.S. (2017). Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 173–182). International World Wide Web Conferences Steering Committee.

# Model Architecture

Input Layer:

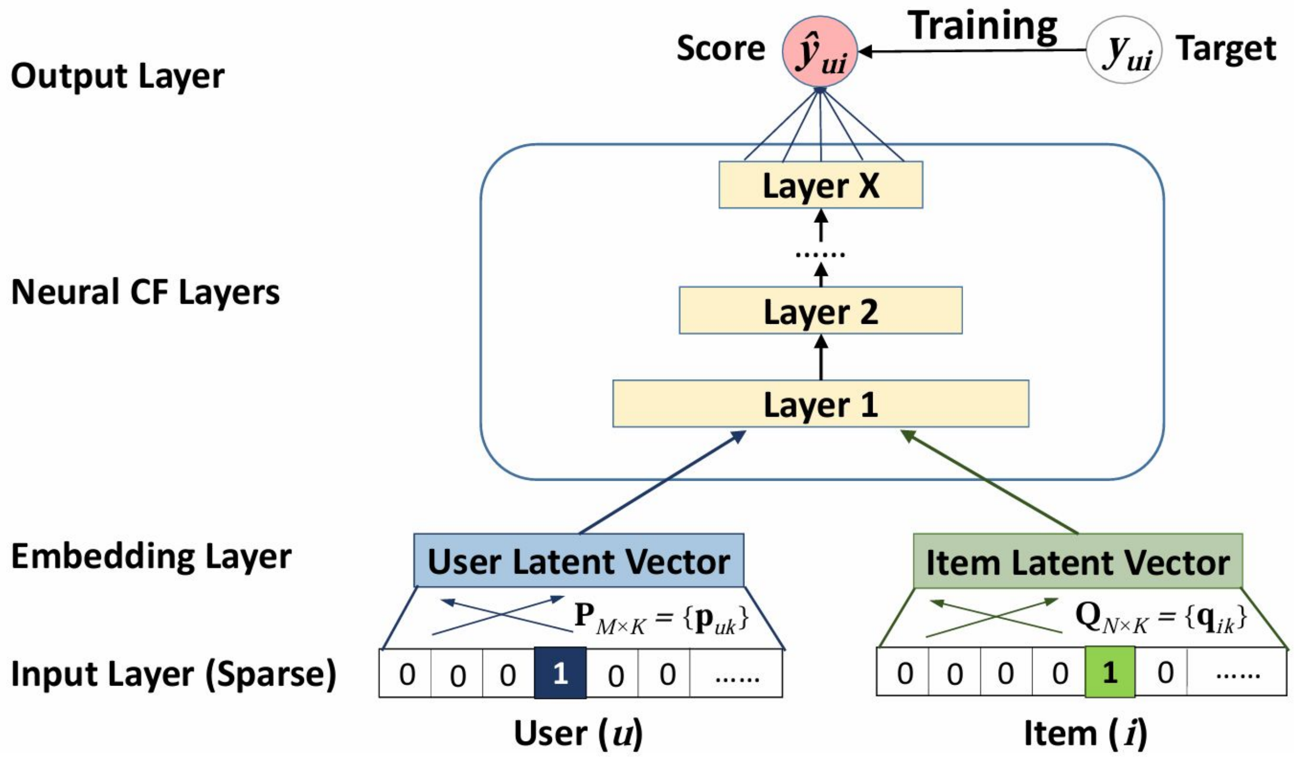
- User ID to Embedding layer
- Item ID to Embedding layer

Neural Interaction Layers:

- Multi-Layer Perceptron (MLP)

Loss Function: Mean Square Loss

# Model Architecture

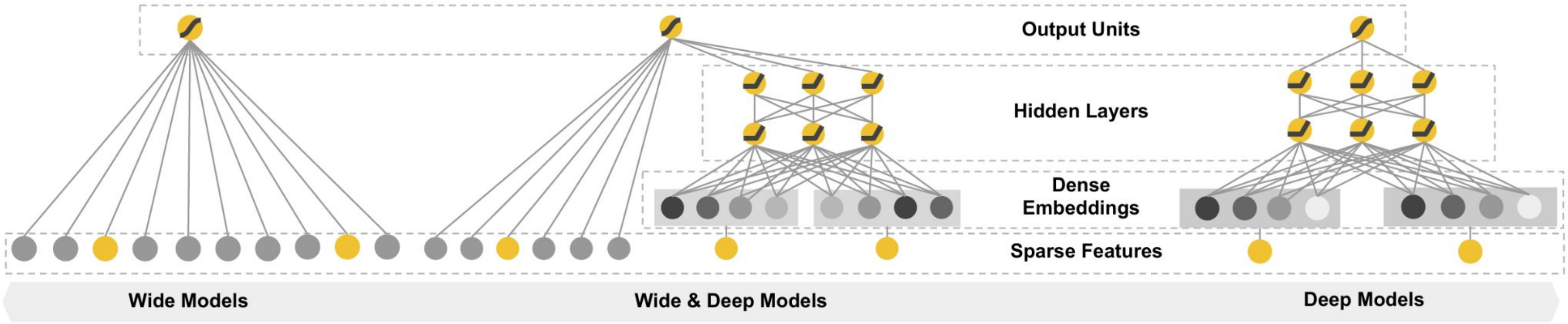


# Deep Learning Workflow (Pytorch)

1. Preparing data
    a. Prepare dataset (we use custom datasets in this tutorial).
    b. Split data into training and test sets.
2. Building model
    a. Creating a model by subclassing **nn.Module** (usually) and define the **forward()** method.
3. Fitting the model (training)
    a. Loss function
    b. Optimizer
4. Making predictions and evaluation (inference)
    a. Evaluation metrics
5. Saving and loading model

# 2 Wide & Deep Learning

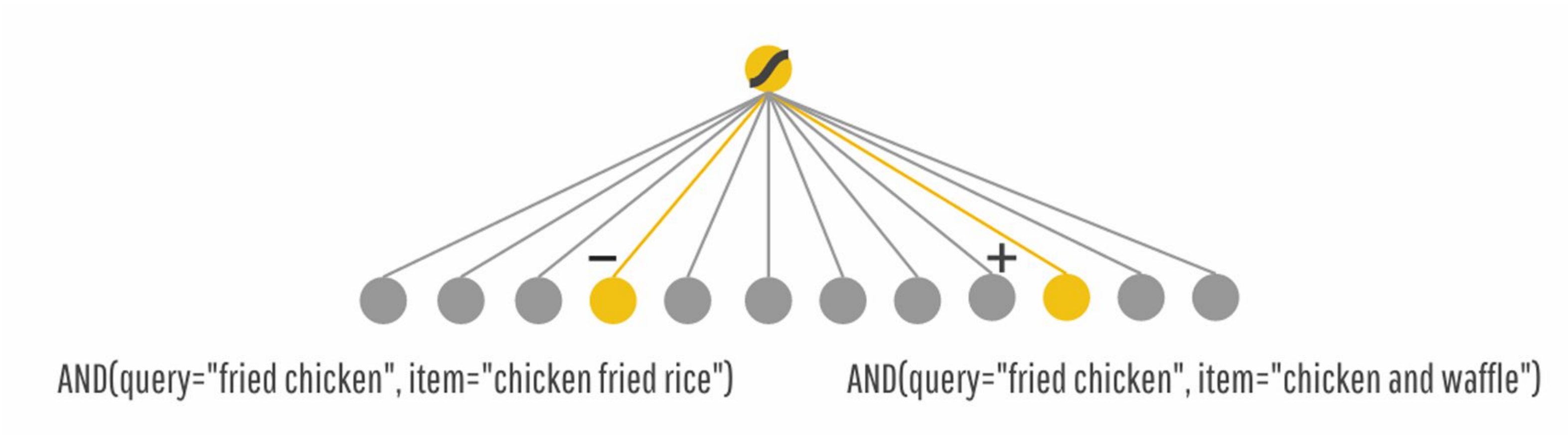
# Wide & Deep Learning




**Memorization** **Generalization**

Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H.B., Anderson, G., Corrado, G.S., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). Wide & Deep Learning for Recommender Systems. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*.

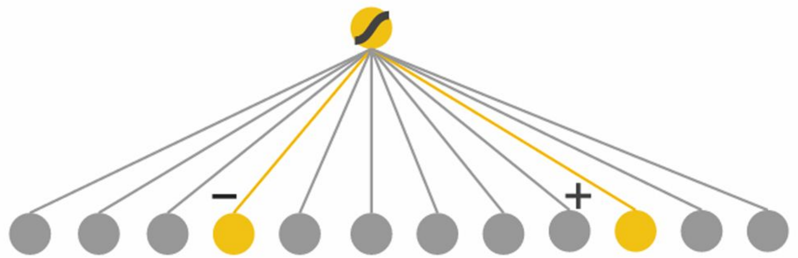
# The Wide Model




**Memorization**

Exploiting the correlation available in the historical data

# The Wide Model



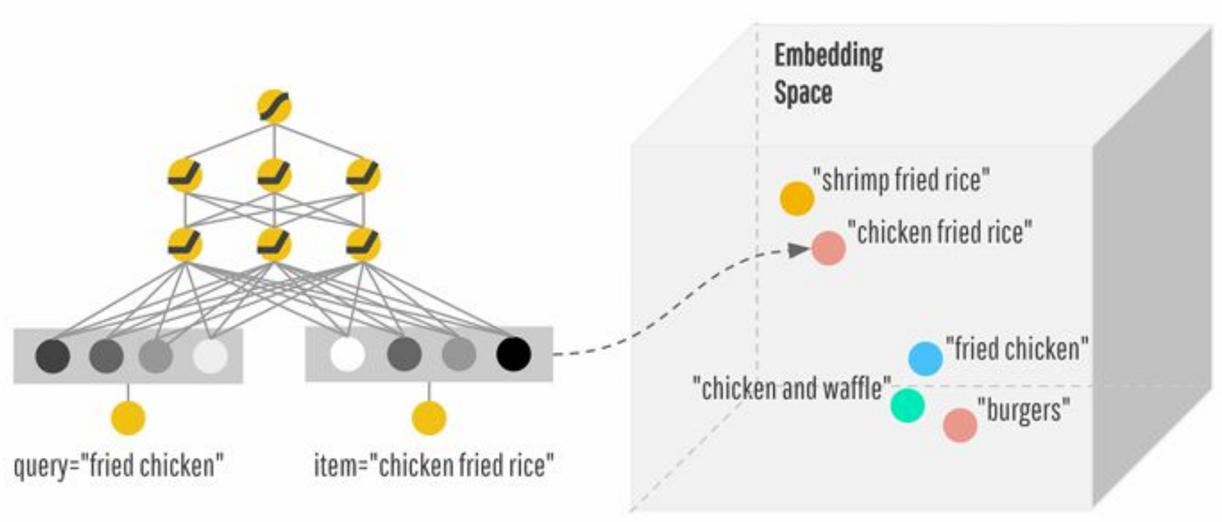

**Cross-product transformation:**

$\phi_k$ = {fried chicken, chicken and waffle}

$x^{(1)}$ = {..., fried chicken, chicken and waffle, ...} $\qquad \phi_k\left(x^{(1)}\right) = 1$

$x^{(2)}$ = {..., fried chicken, chicken fried rice, ...} $\qquad \phi_k\left(x^{(2)}\right) = 0$

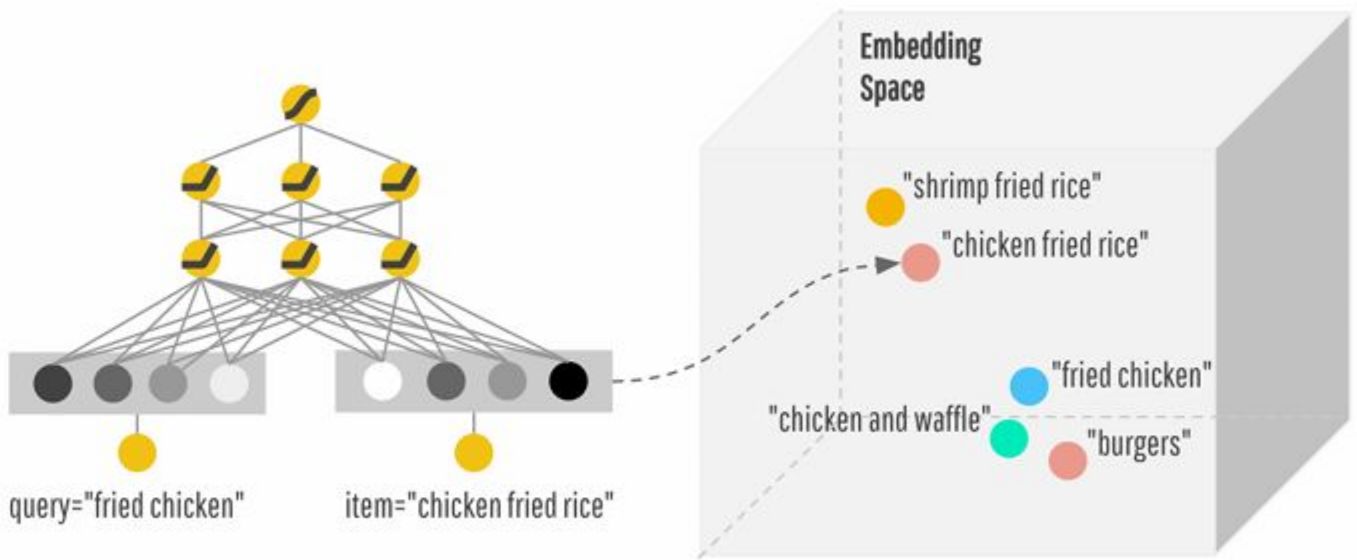
# The Deep Model




**Generalization**

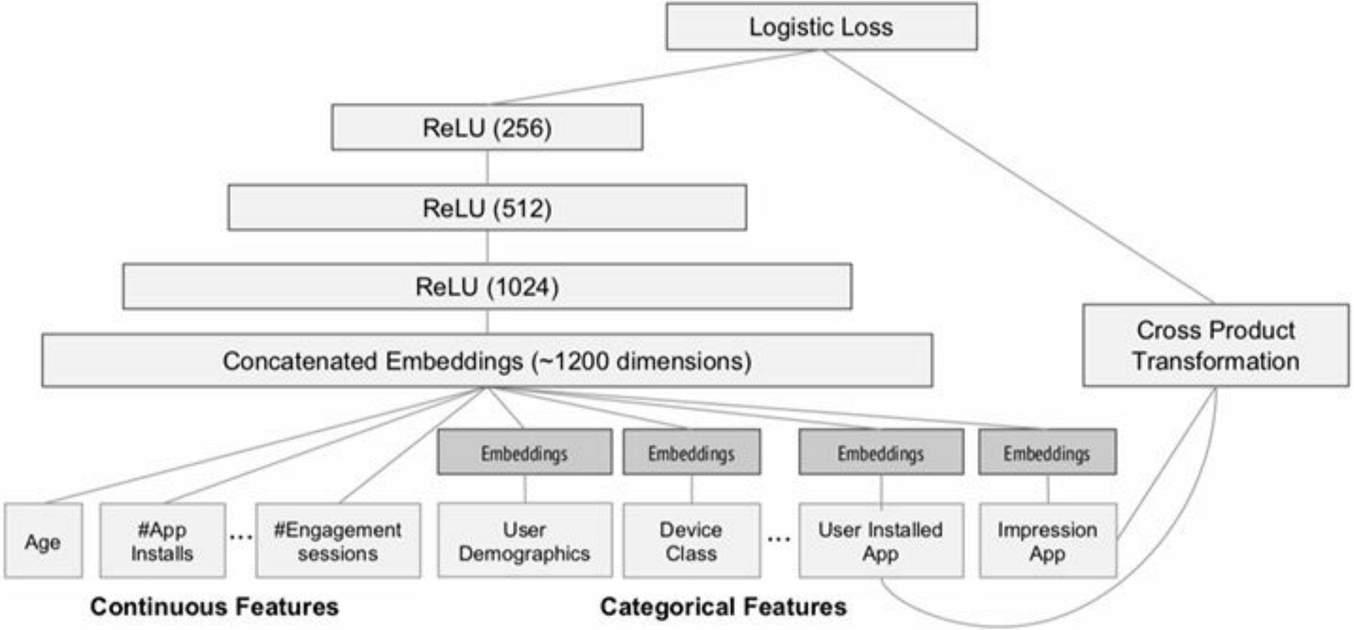
New feature combinations that have never or rarely occurred in the past

# The deep model




$$a^{(l+1)} = f(W^{(l)}a^{(l)} + b^{(l)})$$

# The Wide & Deep Model




$$P(Y=1|\mathbf{x}) = \sigma(\mathbf{w}_{wide}^T[\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}_{deep}^T a^{(l_f)} + b)$$

# Continuous Features

Example feature name: “average_stars”

Example feature value: 3.63

# Deep Categorical Features

Example feature name: “item_city”

Example feature value: “Phoenix”

# Wide Features

```
print(selected_categories[:50])
```

```
['Restaurants', 'Food', 'Nightlife', 'Bars', 'American (Traditional)', 'American (New)', 'Breakfast & Brunch', 'Sandwiches', 'Italian', 'Mexican', 'Event Planning & Services', 'Pizza', 'Burgers', 'Seafood', 'Japanese', 'Arts & Entertainment', 'Coffee & Tea', 'Sushi Bars', 'Desserts', 'Salad', 'Chinese', 'Asian Fusion', 'Steakhouses', 'Beauty & Spas', 'Cafes', 'Shopping', 'Specialty Food', 'Hotels & Travel', 'Cocktail Bars', 'Beer', 'Wine & Spirits', 'Sports Bars', 'Vegetarian', 'Wine Bars', 'Barbeque', 'Bakeries', 'Pubs', 'Automotive', 'Fast Food', 'Mediterranean', 'Thai', 'Caterers', 'Lounges', 'Active Life', 'Vegan', 'Chicken Wings', 'Hotels', 'Venues & Event Spaces', 'Diners', 'Juice Bars & Smoothies']
```

```
get_top_k_p_combinations(tr_df, 2, 10, output_freq=True)
```

```
[(('Food', 'Restaurants'), 12331),
 (('Bars', 'Nightlife'), 12134),
 (('Restaurants', 'Food'), 11543),
 (('Nightlife', 'Bars'), 11480),
 (('Restaurants', 'Nightlife'), 11290),
 (('Nightlife', 'Restaurants'), 11215),
 (('Bars', 'Restaurants'), 10992),
 (('Restaurants', 'Bars'), 10822),
 (('American (New)', 'Restaurants'), 8816),
 (('Restaurants', 'American (New)'), 8424)]
```

# Implementation of the Rating Prediction Task

- See the Jupyter Notebook.

Thank you