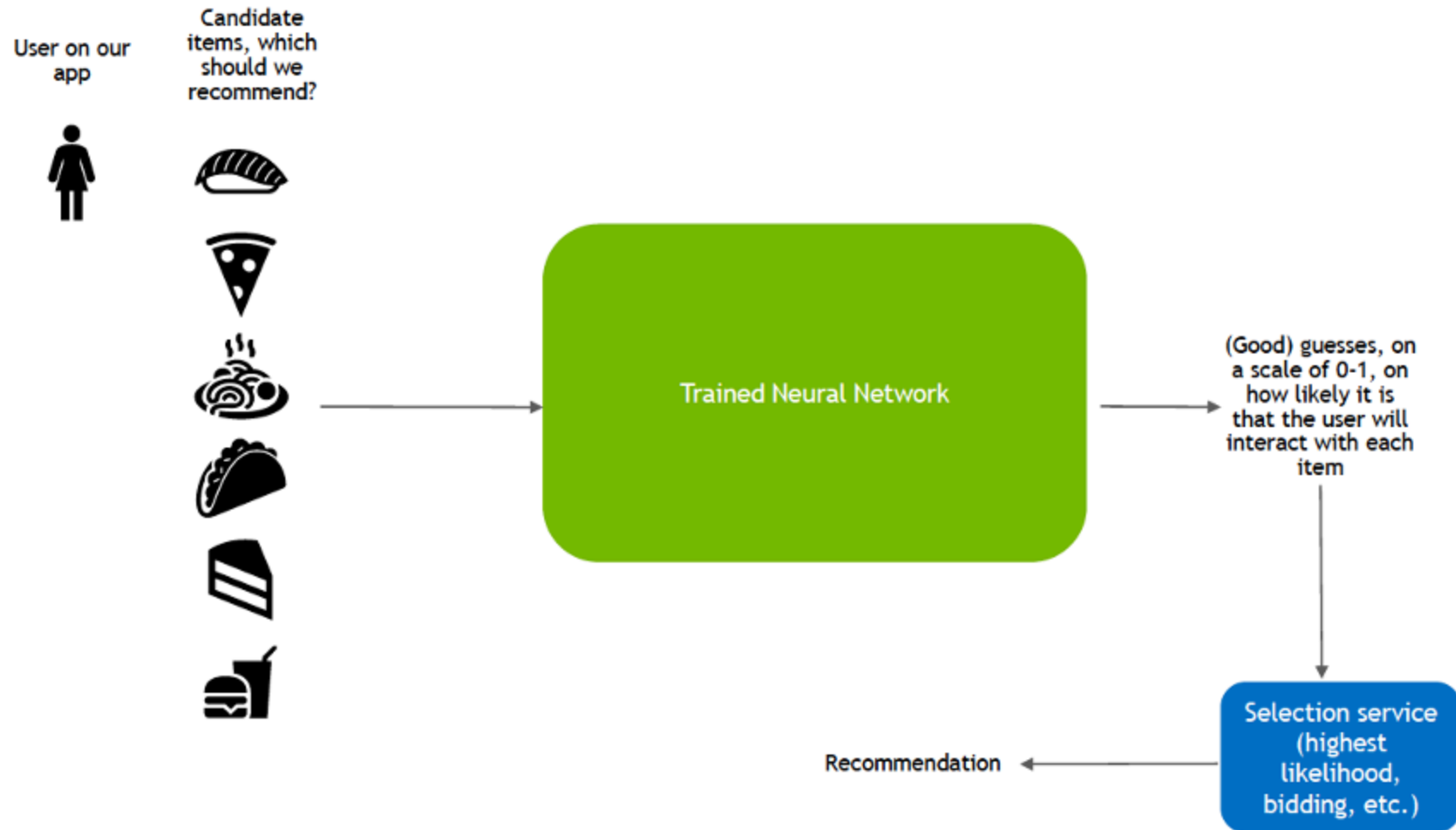


Фаза 2 • Неделя 4 • Вторник

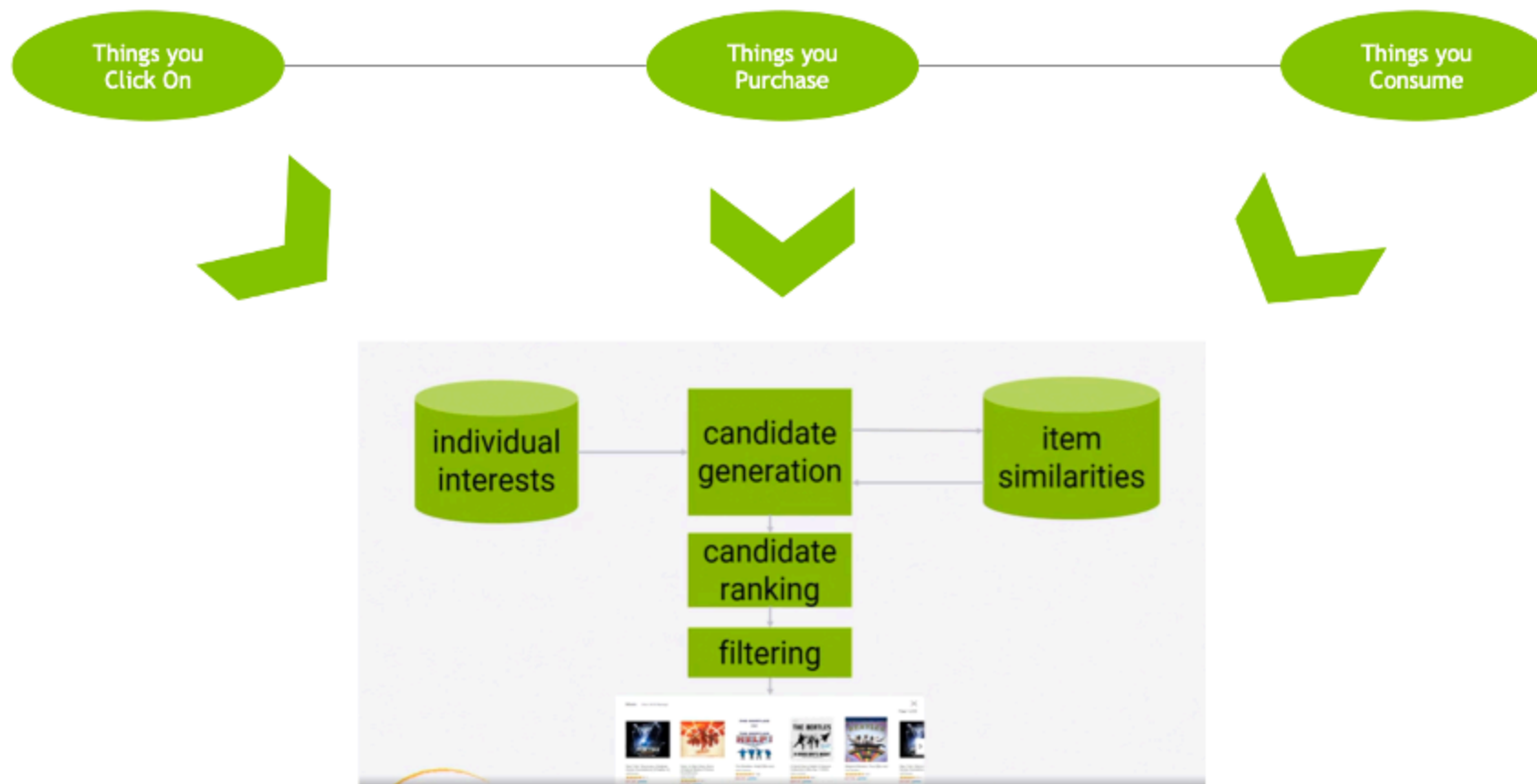
Рекомендательные системы • RecSys

- узнаем, как можно использовать нейросетевой подход в рекомендательных системах
- разберем популярные архитектуры
- реализуем базовый нейросетевой рекомендательный алгоритм
- немного о метриках

# Идея



# Цель

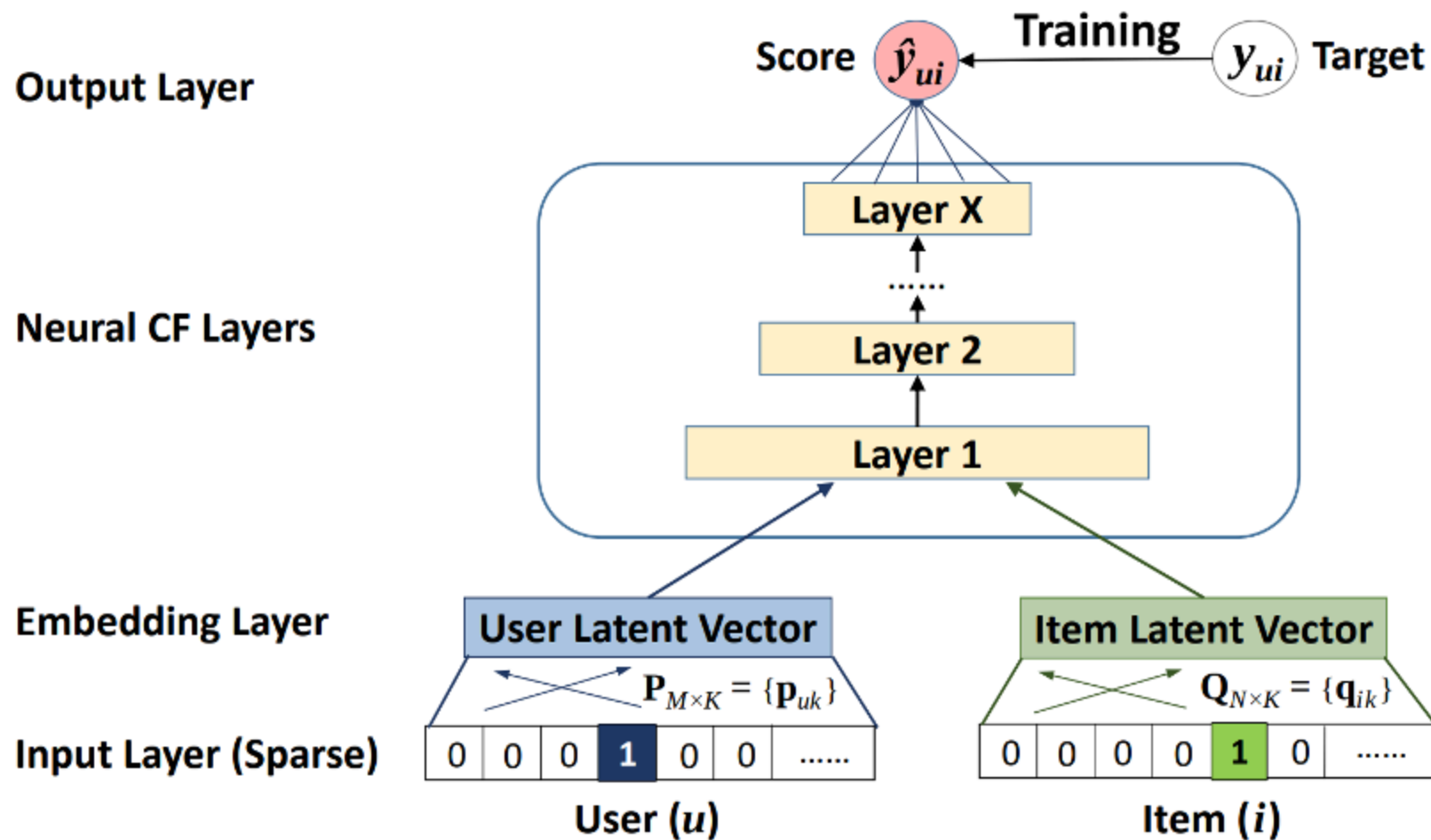


Предположим, необходимо предсказать рейтинг  $r \in [1, 5]$ . Как можно поступить?

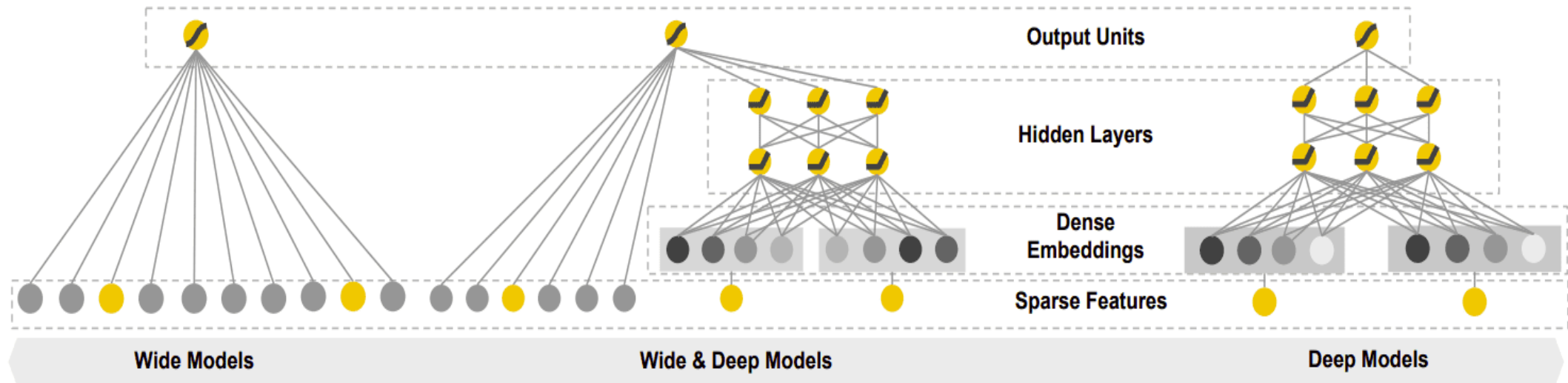
Решить как обычную задачу регрессии с ограничением на диапазон выходных данных.

Так вы сегодня и поступите!

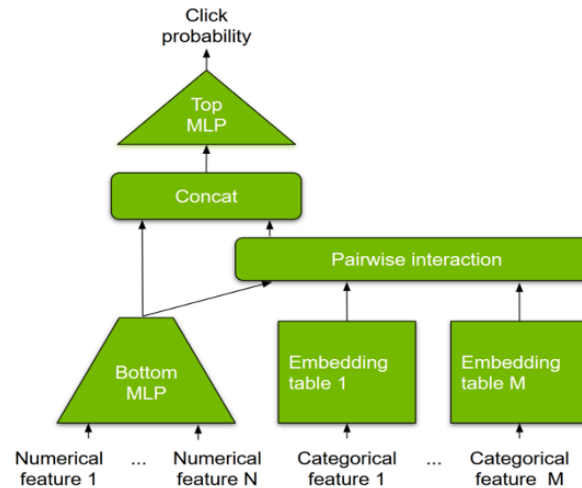
# Neural Collaborative Filtering • NCF



# Wide and Deep



# DLRM • Deep Learning Recommendation Model



Let the users and products be described by many continuous and categorical features. To process the categorical features, each categorical feature will be represented by an embedding vector of the same dimension, generalizing the concept of latent factors used in matrix factorization (3). To handle the continuous features, the continuous features will be transformed by an MLP (which we call the *bottom* or *dense* MLP) which will yield a dense representation of the same length as the embedding vectors (5).

We will compute second-order interaction of different features explicitly, following the intuition for handling sparse data provided in FMs (4), optionally passing them through MLPs. This is done by taking the dot product between all pairs of embedding vectors and processed dense features. These dot products are concatenated with the original processed dense features and post-processed with another MLP (the *top* or *output* MLP) (5), and fed into a sigmoid function to give a probability.



# Метрики качества рекомендаций

- True positive – порекомендовали и пользователь действительно оценил ВЫСОКО
- False positive – порекомендовали, но пользователь оценил НИЗКО
- False negative – не порекомендовали, но пользователь оценил ВЫСОКО








# Рекомендованные и релевантные объекты




item_id	user rating	pred rating
item1	4	2.3
item2	2	3.6
item3	3	3.5
item4	-	4.3
item5	5	4.5
item6	-	2.3

**Релевантные** объекты – те, которые интересны пользователю (мы это знаем из выборки)

**Рекомендованные** объекты – те, которые модель сочла интересными для пользователя (прогноз модели)

# Precision@k








item_id	user rating	pred rating
item1	4 	<del>2.3</del>
item2	4 	3.6 
item3	3	3.5 
item4	2	4.3 
item5	5 	4.5 
item6	1	<del>2.3</del>




-  relevant
-  recommended
-  out of range if k=3

- рекомендованными считаем те, у которых прогнозный рейтинг  $\hat{y} \geq 3.5$
- смотрим долю релевантных элементов среди первых  $k$  рекомендованных элементов
- если  $k = 3$ , то среди первых трех рекомендаций (item2,4,5) релевантных две: item2 и item5:

$$\text{precision@3} = \frac{2}{3} = 0.66$$


# Recall@k

item_id	user rating	pred rating
item1	4 	<del>2.3</del>
item2	4 	3.6 
item3	3	3.5 
item4	2	4.3 
item5	5 	4.5 
item6	1	<del>2.3</del>

-  relevant
-  recommended
-  out of range if k=3

- рекомендованными считаем те, у которых прогнозный рейтинг  $\hat{y} \geq 3.5$
- смотрим долю релевантных элементов среди всех релевантных элементов, присутствующих в первых  $k$  рекомендациях
- релевантных элементов среди первых  $k = 3$  рекомендаций 2 (item2 , item5)
- всего релевантных элементов 3

$$\text{recall}@k = \frac{2}{3} = 0.66$$

- построение рекомендательной системы всегда очень сильно зависит от формата данных
- можно использовать эмбединги, если данные статичны
- если есть последовательности, то можно использовать рекуррентные слои (LSTM, GRU)
- можно использовать attention слои
- существуют архитектуры на основе энкодеров/декодеров трансформеров
-  Метрики качества ранжирования