

Báo cáo tiến độ thực hiện Khoá luận tốt nghiệp

Những điều đã thực hiện được:

- Thêm các đặc trưng phụ (category_id, parent_category, timestamp) vào dataset

	user_id	item_id	event	category_id	parent_category	timestamp
0	257597	179333	view	1173	805	0.218006
1	992329	125263	view	1117	14	0.218248
2	111016	160653	view	1117	14	0.218062
3	483717	127563	view	1117	14	0.218059
4	951259	185159	view	1117	14	0.218007
5	972639	11314	view	1117	14	0.218237
6	810725	223144	view	1098	897	0.218056
7	794181	221212	view	1542	745	0.218171
8	824915	215967	view	646	1606	0.217970
9	339335	41688	view	1117	14	0.217995

- Kích thước dataset khoảng (2883449, 6)
- Chuẩn hoá dữ liệu interaction (view:0, addtocart:1, transaction:2) thành interaction_value để xây dựng mô hình LightFM

Train Shape: (2018414, 7)

Validation Shape: (288345, 7)

Test Shape: (576690, 7)

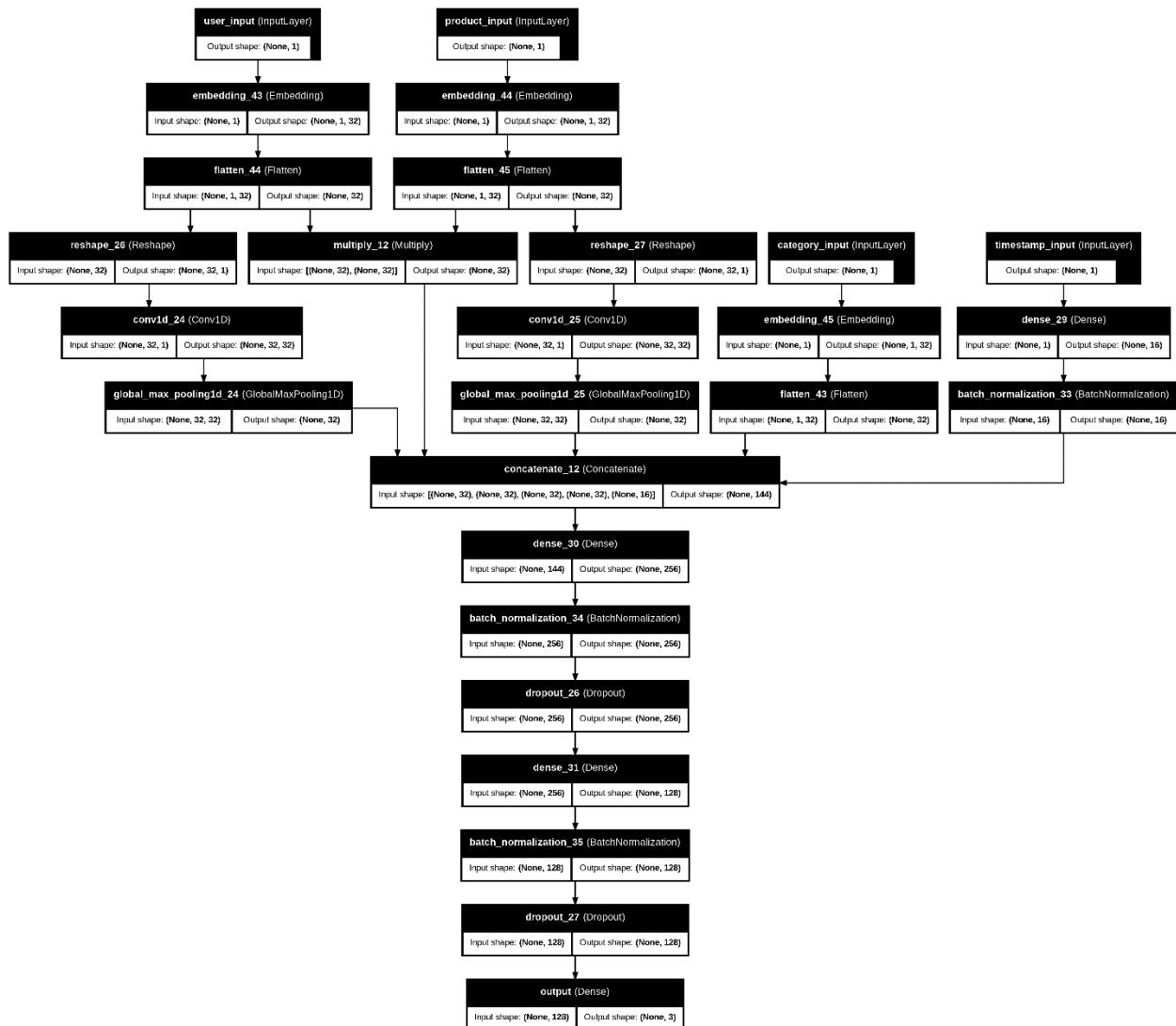
Example data:

	user_id	item_id	event	category_id	parent_category	timestamp
1038379	956036	111212	view	1522	1009	0.809669
1832336	1328004	192051	view	1117	14	0.107441
589703	795354	106903	view	857	1272	0.412567
2830548	1293358	50868	view	1117	14	0.635474
597362	1226016	53537	view	1677	896	0.420086

	interaction_value
1038379	0.0
1832336	0.0
589703	0.0
2830548	0.0
597362	0.0

- Về xây dựng mô hình LightFM:
 - o Trích xuất đặc trưng giữa **user_id** và **item_id** so với **interaction_value** → thu được **Shape of user embedding: (1407580, 32)** và **Shape of item embedding: (235061, 32)** vector có 32 chiều, Dùng để tính toán độ tương đồng giữa user và item
- Về xây dựng mô hình LightFM + CNN:

- Dùng LightFM embeddings cho user & product.
- Học đặc trưng category thông qua embedding trainable
- Dùng CNN để trích xuất thông tin quan trọng từ embeddings.
- Áp dụng Attention giữa user và product để tạo sự liên kết.
- Chuẩn hóa & học đặc trưng timestamp để thêm thông tin thời gian.
- Kết hợp tất cả đặc trưng để dự đoán hành vi người dùng (view, addtocart, transaction).



- Đánh giá mô hình:

```
val_loss, val_accuracy = model_LightFM_CNN.evaluate(X_valid, y_valid)
print(f"Validation Loss: {val_loss:.4f}, Validation Accuracy: {val_accuracy:.4f}")
```

9011/9011 ————— 15s 2ms/step - accuracy: 0.9665 - loss: 0.1636
 Validation Loss: 0.1628, Validation Accuracy: 0.9668

```
test_loss, test_accuracy = model_LightFM_CNN.evaluate(X_test, y_test)
print(f"Test Loss: {test_loss:.4f}, Test Accuracy: {test_accuracy:.4f}")
```

18022/18022 ————— **26s** 1ms/step - accuracy: 0.9665 - loss: 0.1637
Test Loss: 0.1634, Test Accuracy: 0.9667

```
train_loss, train_accuracy = model_LightFM_CNN.evaluate(X_train, y_train)
print(f"Train Loss: {train_loss:.4f}, Train Accuracy: {train_accuracy:.4f}")
```

63076/63076 ————— **91s** 1ms/step - accuracy: 0.9667 - loss: 0.1628
Train Loss: 0.1626, Train Accuracy: 0.9667

○ Thử nghiệm mô hình:

1/1 ————— **0s** 448ms/step
User 126845 -> Product 147118 | Category 1273 | Timestamp 0.19690634174193633 => Predicted Class = 0
User 875062 -> Product 97382 | Category 394 | Timestamp 0.9632482456947571 => Predicted Class = 0
User 1091948 -> Product 8897 | Category 1117 | Timestamp 0.6065823650396567 => Predicted Class = 0
User 547669 -> Product 6358 | Category 1117 | Timestamp 0.8434560558404343 => Predicted Class = 0
User 1306622 -> Product 208679 | Category 1595 | Timestamp 0.23711632382098458 => Predicted Class = 0

Những điều chưa thực hiện:

- Xây dựng mô hình học tăng cường (Deep Q learning)
- So sánh ưu và nhược điểm với những mô hình khác
- Thử nghiệm tập dữ liệu khác