المحور الخامس: الشبكات العصبية التكرارية (RNN)

I. مقدمة :

تمثّل البيانات الزمنية أحد أصعب أنواع البيانات من ناحية النمذجة، لأن معناها لا يكمن في القيم نفسها، بل في **العلاقات الاضطرادية عبر الزمن** .وخلافاً للشبكات التغذية-الأمامية (Feedforward) التي تفترض استقلالية البيانات، جاءت الشبكات العصبية التكرارية RNN لتجيب على سؤال جوهري:

كيف يمكن للشبكة العصبية أن "تتذكر" شيئاً ما حدث في الماضي أثناء معالجة الحاضر؟

هذا السؤال شكّل الأساس الفكري لظهور المعماريات التكرارية

II. الفصل الأول: الأساس المفاهيمي للشبكات التكرارية

1. مفهوم التبعية الزمنية

البيانات الزمنية لا تُقرأ كنقاط منفردة، بل كسلسلة تربط كل نقطة بسابقتها. وبذلك يصبح الهدف ليس فقط تمثيل المعلومات، بل **تمثيل المعلومات مع تاريخها**.

في الاقتصاد مثلاً:

- سعر اليوم يعتمد على اليوم السابق
- سلوك العميل اليوم يعتمد على تفاعلاته السابقة
- حجم الطلب الصناعي يتأثر بمسار الإنتاج سابقاً

من هنا ظهر نوع جديد من الشبكات:
شبكات تمتلك حالة داخلية Internal State تتطور عبر الزمن.

2. RNN كنظام ديناميكي غير خطي

يمكن النظر إلى RNN كنظام:

$$h_t = f(h_{t-1}, x_t)$$

حيث f توصف عادة بدالة تنشيط غير خطية.
هذا يجعل RNN أقرب إلى:

- نماذج التحكم Control Systems
- النماذج الديناميكية غير الخطية Nonlinear Dynamical Systems
- أو حتى ماركوف من الدرجة العليا High-order Markov Models

لكن ما يميز RNN أنها تتعلم تمثيل الاعتماد الزمني **تلقائياً** من البيانات.

3. المشكلة الجوهرية: تلاشي وتأرجح الذاكرة

أثناء التعلم، تعتمد RNN على نشر التدرجات عبر الزمن **(Backpropagation Through Time (BPTT**
لكن عند الانتقال عبر الزمن، يحدث أمران:

1) **تلاشي التدرج**

المعلومات البعيدة في الماضي تضيع تدريجياً. تصبح الشبكة "فاقدة للذاكرة" مع طول السلسلة.

2) **انفجار التدرج**

أحياناً تتضخم الإشارات بشكل يؤدي إلى عدم استقرار النموذج.

هذه مشكلة بنيوية:
الحفاظ على ذاكرة مستقرة عبر الزمن الطويل ليس ممكناً باستخدام بنية RNN البسيطة.

وهنا ظهر الحل :**الذاكرة طويلة وقصيرة الأمد LSTM .**

III. الفصل الثاني: الأساس النظري لشبكات LSTM

1. فلسفة ظهور LSTM

المشكلة الأساسية ليست في القدرة على التعلّم، بل في القدرة على **الحفاظ على المعلومات عبر الزمن دون تدهور**.

LSTM جاءت لتجيب على سؤالين:

1. كيف نضمن بقاء المعلومة طالما أنها ضرورية؟
2. كيف نتخلص من المعلومة عندما تصبح غير ذات معنى؟

إذن LSTM ليست مجرد بنية حسابية، بل **آلية للتحكّم في الذاكرة**.

2. الذاكرة الطويلة C(t): مفهوم "الممر المحمي"

- تتميز LSTM بوجود مسار مستقيم تقريباً للمعلومات عبر الزمن يسمى **Cell State** أو الذاكرة الطويلة C_t

هذا المسار يشبه "حزام ناقل" يمر عبر الزمن دون أن يتأثر كثيراً بالتدرجات.
وبالتالي:

- المعلومات لا تتبخر
- التدرجات تظل ثابتة
- العلاقات الطويلة المدى تبقى محفوظة

وهذا هو الفرق الجوهري عن RNN.

3. بوابات LSTM: آليات قرار ديناميكية

البوابات هي عناصر **قَرارية Decision Mechanisms** داخل النموذج تتحكم في ذاكرته.

1) بوابة النسيان Forget Gate

تقرر ما يجب التخلص منه من الذاكرة.
بمنطق اقتصادي:

مثل إدارة مخزون تتخلص من المنتجات منخفضة الطلب.

2) بوابة الإدخال Input Gate

تقرر ما المعلومات الجديدة التي يجب إضافتها للذاكرة.

3) بوابة الإخراج Output Gate

تقرر ما الجزء الذي يجب أن يظهر ك **تمثيل للحظة الحالية**.

هذه البوابات ليست قواعد صلبة بل:

- دوال احتمالية
- تعتمد على الانحدار اللوجستي
- تتعلم أثناء التدريب

وبذلك تتحول LSTM إلى نموذج **يكتشف بنفسه ما يجب تذكّره وما يجب نسيانه**.

IV. الفصل الثالث : GRU تبسيط النظرية دون فقدان الجوهر

1. ظهور GRU

ظهرت GRU كتطوير من نفس الفكرة ولكن بدمج البوابات، لتقليل التعقيد.
من الناحية النظرية:

- تحتفظ GRU بنفس القدرة على تمثيل العلاقات الطويلة
- تقدم نموذجاً ديناميكياً بذاكرة مضغوطة
- تُعد أقرب إلى **حل وسط بين التعقيد والكفاءة**

وهذا ما يجعلها أكثر استخداماً في التطبيقات الإدارية والاقتصادية.

2. مقارنة نظرية بين RNN و LSTM وGRU

RNN	LSTM	GRU
• تعتمد على ذاكرة وحيدة • غير مستقرة رياضياً • قدرتها على تمثيل العلاقات الطويلة ضعيفة	• إدخال نظام بوابات = استقرار • فصل الذاكرة الطويلة عن القصيرة • تعزيز قابلية التعلم • تقليل عملية تلاشي التدرج	• تبسيط لLSTM • دمج بوابتي الإدخال والنسيان • تمثيل أقل تعقيداً • أداء ممتاز لكن بمرونة نظرية أقل منLSTM

❖ مثال عملي باستخدامKeras

في هذا المثال، سنبني شبكة **LSTM(GRU) بسيطة** لتصنيف بيانات منطق **AND**.

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
# ============================
# 1. AND DATA
# ============================
X = np.array([
   [0, 0],
   [0, 1],
   [1, 0],
   [1, 1]
], dtype=float)

y = np.array([0, 0, 0, 1], dtype=float)

# reshape to (samples, features, 1)
X = X.reshape((4, 2, 1))

# ============================
```

```
# 2. BUILD LSTM MODEL

# ============================


model = Sequential()


# LSTM avec 8 unités

model.add(LSTM(8, activation='tanh', input_shape=(2, 1)))

# if we want to use GRU just change LSTM with GRU

# Couche de sortie binaire

model.add(Dense(1, activation='sigmoid'))

# ============================

# 3. TRAINING

# ============================

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])


model.summary()

model.fit(X, y, epochs=500, verbose=1)


# ============================

# 4. TEST

# ============================

print("\nPredictions:")
```

```
print(model.predict(X))
```

V. خاتمة نظرية

نستطيع تلخيص الأساس النظري لـ RNN و LSTMفي النقاط التالية:

- هي شبكات تمثل **أنظمة ديناميكية تعتمد على حالة داخلية**.
- الذاكرة ليست هدفاً بحد ذاتها، بل أداة لفهم **البنية الزمنية للسلسلة**.
- ظهرت LSTM لتجاوز عيب بنيوي في RNN (تلاشي التدرج).
- تعتمد LSTM على آليات قرار داخلية تشبه **إدارة المعرفة** :ماذا نحتفظ به؟ ماذا ننسى؟
- أصبحت LSTM و GRU أدوات نظرية قوية في تحليل السلاسل الزمنية في التسيير والاقتصاد.