المحور الرابع: الشبكات العصبية الالتفافية(CNN)

I. مقدمة حول الشبكات العصبية الالتفافية(CNN)

تُعد الشبكات العصبية الالتفافية (Convolutional Neural Networks – CNN) من أهم النماذج المستخدمة في مجالات الرؤية الحاسوبية، مثل:

- التعرف على الصور
- كشف الأجسام
- تصنيف الصور
- تحليل النصوص والصوت

تتميز CNN بقدرتها على **استخراج الأنماط والخصائص (Features)** من الصور بطريقة فعّالة، وهي مصممة لمعالجة البيانات ذات البنية الشبكية مثل **الصور (2D)** أو **الإشارات الصوتية(1D)**

II. الفكرة الأساسية لCNN

الصورة تحتوي على **بنية مكانيةSpatial Structure** ، أي أن البكسلات المتجاورة تشترك في معلومات مهمة.
على عكس الشبكات التقليدية(Fully Connected) ، فإنCNN *لا تتعامل مع كل بكسل بشكل مستقل* أي أنها تأخذ في الاعتبار **الجوار المحلي للبكسلات** لاستخراج الميزات.

تقوم CNN بعملية تسمى **الالتفاف(convolution)** ، وهي عبارة عن تمرير فلتر صغير على الصورة لاستخراج خصائص مثل:

- الحواف
- الأشكال
- الأنسجة
- الأنماط المتكررة

III. الطبقات الأساسية فيCNN

تتكون الشبكات الالتفافية من عدة أنواع من الطبقات، لكل منها دور محدد:

1. **طبقة الالتفاف(Convolution Layer)**

هي الطبقة الأساسية في.CNN

دورها:

- استخراج **Features**من الصورة باستخدام **مرشِّحات/فلاتر.Filters**
- كل فلتر يتعلم خاصية معينة:
  - فلتر للحواف الأفقية
  - فلتر للحواف العمودية
  - فلتر للإنحناءات
  - فلتر للقوامtextures

ما الذي تفعله؟

تمرر فلتر صغير (مثلاً 3×3) على كامل الصورة وتنتج خريطة خصائصFeature Map

2. طبقة التجميع(Pooling Layer)

تُعرف أيضًا بـ.**Downsampling**

دورها:

- تقليل حجم البيانات مع الحفاظ على أهم الخصائص.
- تقليل عدد المعاملات ⟵ تسريع التدريب.
- جعل النموذج أكثر قدرة على التعميم.(Generalization)

أشهر الأنواع:

- **Max Pooling:** تأخذ أكبر قيمة في كل منطقة.
- **Average Pooling:** تأخذ متوسط القيم.

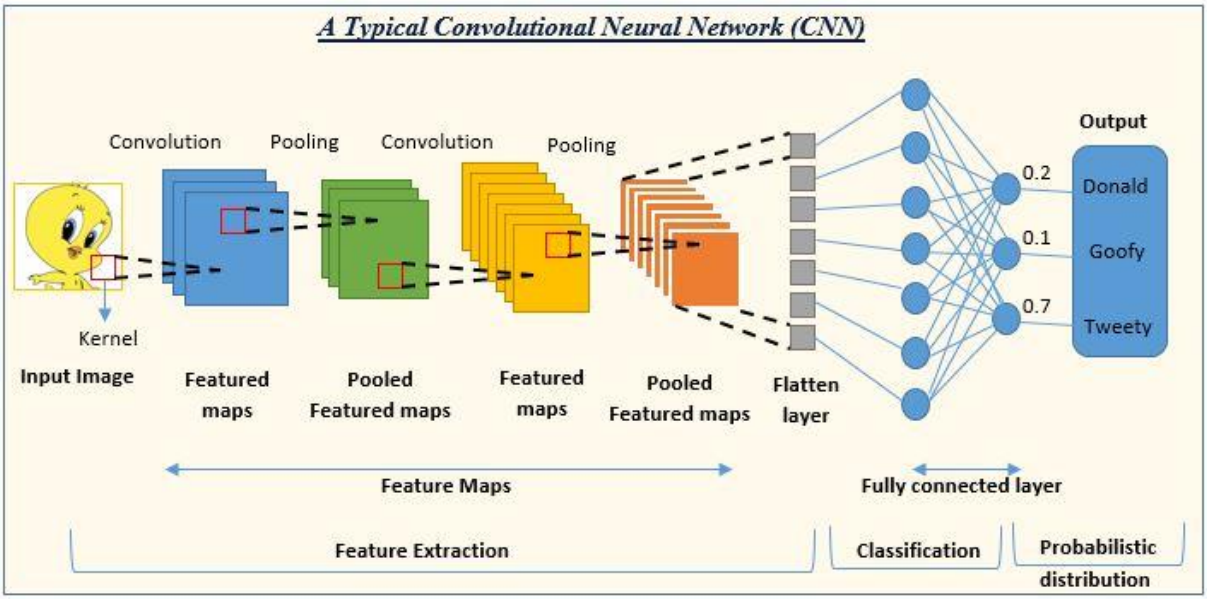
3. طبقةFlatten

بعد الانتهاء من طبقات الالتفاف والتجميع، يتم تحويل (تفريغ) المصفوفات إلى **متجه واحد** لتمريره إلى الطبقات الكاملة Fully Connected.

4. طبقات(Fully Connected (Dense Layers

تُستخدم في النهاية لإنجاز مهمة:

- التصنيف(Classification)
- التنبؤ(Prediction)



IV. مثال كامل على سلسلة العمليات في الشبكات الالتفافية

(الالتفاف ⟵ التجميع ⟵ الالتفاف ⟵ التجميع ⟵ التفريغ)

نستخدم في هذا المثال صورة صغيرة ثنائية الأبعاد بحجم: **4×4**

$$X = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 4 & 5 & 6 & 1 \\ 7 & 8 & 9 & 2 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

والعمليات التي سنطبقها هي:

1. الطبقة الالتفافية الأولى
2. طبقة التجميع الأولى
3. الطبقة الالتفافية الثانية
4. طبقة التجميع الثانية
5. طبقةFlatten

نحسب جميع المراحل خطوة بخطوة.

❖ **الطبقة الالتفافية الأولى(Convolution #1)**

نستخدم فلترًا بحجم :2×2

$$F_1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

بما أنّ حجم الصورة 4×4 وحجم الفلتر 2×2، فالناتج سيكون مصفوفة بحجم .3×3

نقوم بتمرير الفلتر على الصورة وحساب الالتفاف في كل موضع.

- **حساب الصف الأول من الناتج**

الموضع(1،1)

$$\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \Rightarrow 1(1) + 2(0) + 4(-1) + 5(1)$$

$$= 1 - 4 + 5 = 2$$

الموضع(2,1)

$$\begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix} \Rightarrow 2 - 5 + 6 = 3$$

الموضع(3,1)

$$\begin{bmatrix} 3 & 0 \\ 6 & 1 \end{bmatrix} \Rightarrow 3 - 6 + 1 = -2$$

- **حساب الصف الثاني من الناتج**

الموضع(1,2)

$$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix} \Rightarrow 4 - 7 + 8 = 5$$

الموضع(2،2)

$$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix} \Rightarrow 5 - 8 + 9 = 6$$

الموضع(3,2)

$$\begin{bmatrix} 6 & 1 \\ 9 & 2 \end{bmatrix} \Rightarrow 6 - 9 + 2 = -1$$

- حساب الصف الثالث من الناتج

الموضع(1,3)

$$\begin{bmatrix} 7 & 8 \\ 1 & 2 \end{bmatrix} \Rightarrow 7 - 1 + 2 = 8$$

الموضع(2,3)

$$\begin{bmatrix} 8 & 9 \\ 2 & 3 \end{bmatrix} \Rightarrow 8 - 2 + 3 = 9$$

الموضع(3،3)

$$\begin{bmatrix} 9 & 2 \\ 3 & 4 \end{bmatrix} \Rightarrow 9 - 3 + 4 = 10$$

✓ ناتج الطبقة الالتفافية الأولى

$$C_1 = \begin{bmatrix} 2 & 3 & -2 \\ 5 & 6 & -1 \\ 8 & 9 & 10 \end{bmatrix}$$

❖ طبقة التجميع الأولى(MaxPooling 2×2)

نستخدم نافذة تجميع حجمها 2×2 وخطوة (stride) تساوي 1

النافذة الأولى:

$$\begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix} \Rightarrow \max = 6$$

النافذة الثانية :

$$\begin{bmatrix} 3 & -2 \\ 6 & -1 \end{bmatrix} \Rightarrow \max = 6$$

النافذة الثالثة :

$$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix} \Rightarrow \max = 9$$

النافذة الر ابعة:

$$\begin{bmatrix} 6 & -1 \\ 9 & 10 \end{bmatrix} \Rightarrow \max = 10$$

✓ **ناتج التجميع الأول**

$$P_1 = \begin{bmatrix} 6 & 6 \\ 9 & 10 \end{bmatrix}$$

❖ **الطبقة الالتفافية الثانية(Convolution #2)**

نستخدم فلترًا جديدًا:

$$F_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

بما أنّ حجم الإدخال 2×2 وحجم الفلتر 2×2، فالناتج سيكون قيمة واحدة فقط:

الحساب:

$$\begin{bmatrix} 6 & 6 \\ 9 & 10 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = 6 \cdot 0 + 6 \cdot 1 + 9 \cdot 1 + 10 \cdot 0 = 0 + 6 + 9 + 0 = 15$$

✓ **ناتج الالتفاف الثاني**

$$C_2 = [15]$$

❖ **طبقة التجميع الثانية**

بما أنّ الإدخال عبارة عن عنصر واحد فقط، يبقى كما هو:

$$P_2 = [15]$$

❖ **طبقةFlatten**

يقوم Flatten بتحويل المصفوفة إلى متجه:

$$\text{Flatten} = [15]$$

الناتج النهائي للمثال كاملاً

$$\boxed{[15]}$$

❖ **مثال عملي باستخدامKeras**

في هذا المثال، سنبني شبكة **CNN بسيطة** لتصنيف بيانات منطق **AND**.

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten, 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 CNN MODEL
# ============================
model = Sequential()


# First convolution
model.add(Conv1D(16, kernel_size=2, padding="same",
          activation='relu', input_shape=(2,1)))
model.add(MaxPooling1D(pool_size=2))


# Second convolution
model.add(Conv1D(32, kernel_size=2, padding="same", activation='relu'))
model.add(MaxPooling1D(pool_size=1))


# Flatten
model.add(Flatten())
```

```
# Output layer
model.add(Dense(1, activation='sigmoid'))


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


model.summary()
model.fit(X, y, epochs=200, verbose=1)


# ============================
# 4. TEST
# ============================
print("\nPredictions:")
print(model.predict(X))
```

V. الخاتمة

الشبكات العصبية الالتفافية(CNN) من أهم النماذج المستخدمة في مجالات الرؤية الحاسوبية والتعرف على الأنماط في البيانات ذات البنية الشبكية مثل الصور والإشارات الصوتية.

CNN تتميز بالقدرة على:

- **استخراج الخصائص (Features)** من البيانات بطريقة تلقائية وفعّالة، باستخدام طبقات الالتفاف.

- **تقليل الأبعاد** والحفاظ على أهم المعلومات من خلال طبقات التجميع(Pooling)
- **تحويل المخرجات إلى متجهات قابلة للتصنيف** عبر طبقة Flatten والطبقات الكاملة الاتصال (Fully Connected)

- كل فلتر في طبقة الالتفاف يتعلم ميزة مختلفة من البيانات.
- كل خطوة في الشبكة تساعد على بناء **تمثيل غني للبيانات** قبل عملية التصنيف النهائية.
- Poolingيساعد على تقليل عدد المعاملات وزيادة قدرة النموذج على التعميم.

وباختصار، فإن CNN تمثل أداة قوية لمعالجة البيانات المصفوفية، وتعد الأساس للعديد من التطبيقات الحديثة في التعرف على الصور والفيديوهات وتحليل الإشارات