مشروع نهاية الفصل 6

مقرر Deep Learning — Convolutional Neural Networks — الـ convolution بفلاتر ثابتة

الهدف: تطبّق convolution بإيدك بفلاتر ثابتة (حواف) وتستخدم النتيجة كـ features.
الأدوات: Python مع numpy وscikit-learn، وتقدر تشغّل الكود في Google Colab المجاني.

الداتا: sklearn.datasets.load_digits (١٧٩٧ صورة ٨×٨، ١٠ أرقام). مصدرها UCI. مدمجة في scikit-learn.

الأسئلة العملية

1 جهّز (Prepare)

- قسّم digits ٨٠/٢٠.

2 Convolution (Convolution)

- اكتب دالة تطبّق فلتر ٣×٣ على صورة ٨×٨ (valid، خرج ٦×٦).

3 فلاتر (Filters)

- استخدم فلتر حواف أفقي وفلتر حواف رأسي وخد |الناتج|.

4 Classifier (Classifier)

- درّب LogisticRegression على الـ features الجديدة وعلى الـ pixels الخام وقارن الدقتين.

بونس

- بونس: أضف max pooling ٢×٢.
- بونس: جرّب فلتر blur.

أسئلة تفكير (اكتب إجابة في سطرين)

1. ليه الـ convolution بيستخدم نفس الأوزان في كل مكان؟
2. إيه فايدة الـ pooling؟

ماذا تسلّم؟

- ملف الكود (أو notebook) بيشتغل من الأول للآخر.
- فقرة قصيرة (٥ أسطر) بتلخّص النتيجة وإيه اللي اتعلمته.

الحل النموذجي

جرّب بنفسك الأول، وبعدين قارن. الأرقام الحقيقية في النتيجة تحت.

الكود

```
# dataset ships inside scikit-learn (no download needed)
import numpy as np
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split

X, y = load_digits(return_X_y=True)
X = X / 16.0
Xtr, Xte, ytr, yte = train_test_split(X, y, test_size=0.2, random_state=0)
Ytr = np.eye(10)[ytr]
from sklearn.linear_model import LogisticRegression

Kh = np.array([[1, 1, 1], [0, 0, 0], [-1, -1, -1]], float)
Kv = Kh.T

def conv(img, K):
    out = np.zeros((6, 6))
    for i in range(6):
        for j in range(6):
            out[i, j] = (img[i:i + 3, j:j + 3] * K).sum()
    return out

def feats(A):
    return np.array([np.concatenate([np.abs(conv(a.reshape(8, 8), Kh)).ravel(),
np.abs(conv(a.reshape(8, 8), Kv)).ravel()]) for a in A])

Ftr, Fte = feats(Xtr), feats(Xte)
print('edge features:', round(LogisticRegression(max_iter=2000).fit(Ftr, ytr).score(Fte,
yte), 3))
print('raw pixels   :', round(LogisticRegression(max_iter=2000).fit(Xtr, ytr).score(Xte,
yte), 3))
```

النتيجة

```
edge features: 0.964
raw pixels   : 0.964
```

ملاحظات على الإجابات

- مشاركة الأوزان بتقلّل عدد الـ parameters وبتخلّي الفلتر يلاقي النمط في أي مكان.

- الـ pooling بيصغّر الخريطة وبيدّي ثبات صغير قدام الإزاحة. الفلاتر هنا ثابتة مش متعلّمة، فمتوقعش إنها تتفوق على الـ pixels الخام دايماً.

Credits: topic follows the matching chapter of Dive into Deep Learning (d2l.ai, CC BY-SA 4.0). Questions, code and explanations are original.