مشروع نهاية الفصل 9

مقرر Deep Learning — Attention and Transformers — Self-attention والمواضع

الهدف: تطبّق self-attention على صفوف الأرقام وتشوف أثر إضافة معلومة الموضع.
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. كل صورة ٨ tokens.

2 Attention (Attention)

- Q و K و V من مصفوفات عشوائية ثابتة، و weights = softmax(Q K^T / sqrt(d)).

3 بدون مواضع (Without positions)

- افرد المخرجات ودرّب LogisticRegression.

4 مع مواضع (With positions)

- أضف vector موضع لكل صف قبل الـ attention وقارن.

بونس

- بونس: استخدم head ثاني وضمّ المخرجات.
- بونس: اطبع مصفوفة الأوزان لصورة واحدة.

أسئلة تفكير (اكتب إجابة في سطرين)

1. ليه الـ attention لوحدها بتتجاهل الترتيب؟
2. ليه بنقسم على sqrt(d)؟

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
from sklearn.linear_model import LogisticRegression

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)
rng = np.random.default_rng(0)
d = 16
Wq, Wk, Wv = [rng.normal(0, 0.5, (8, d)) for _ in range(3)]
pos = rng.normal(0, 0.5, (8, 8))

def softmax(s):
    e = np.exp(s - s.max(-1, keepdims=True))
    return e / e.sum(-1, keepdims=True)

def attention(Z, use_pos):
    S = Z.reshape(-1, 8, 8)
    if use_pos: S = S + pos
    Q, K, V = S @ Wq, S @ Wk, S @ Wv
    A = softmax(Q @ K.transpose(0, 2, 1) / np.sqrt(d))
    return (A @ V).reshape(len(Z), -1)

def score(f):
    m = LogisticRegression(max_iter=2000).fit(f(Xtr), ytr)
    return m.score(f(Xte), yte)

print('no positions  :', round(score(lambda Z: attention(Z, False)), 3))
print('with positions:', round(score(lambda Z: attention(Z, True)), 3))
```

النتيجة

```
no positions  : 0.769
with positions: 0.931
```

ملاحظات على الإجابات

- الـ attention بتحسب علاقات بين الـ tokens بغض النظر عن مكانها، فبنحتاج نضيف الموضع بنفسنا.
- القسمة على sqrt(d) بتمنع السكورات تكبر مع d فالـ softmax ما يبقاش حاد زيادة. vectors المواضع هنا عشوائية ثابتة مش متعلّمة.

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