從神經網絡到 GPT

YouTube 影片

Andrej Karpathy - 讓我們從零開始，用程式碼一步步建構 GPT。

Umar Jamil - Attention is all you need (Transformer) - 模型解釋（包括數學）、推論與訓練。

StatQuest with Josh Starmer - Transformer 神經網絡，ChatGPT 的基礎，清晰解釋 !!!

Pascal Poupart - CS480/680 講座 19：注意力與 Transformer 網絡。

The A.I. Hacker - Michael Phi - 圖解 Transformer 神經網絡：逐步解釋。

我的學習方式

當我讀完《神經網絡與深度學習》這本書的一半後，我開始複製書中識別手寫數字的神經網絡範例。我在 GitHub 上創建了一個倉庫，https://github.com/lzwjava/neural-networks-and-zhiwei-learning。

這才是真正的難點。如果一個人能夠不抄襲任何程式碼，從零開始寫出來，那麼他對這個概念的理解就非常深刻了。

我的複製程式碼還缺少 `update_mini_batch` 和 `backprop` 的實現。然而，通過仔細觀察加載數據、前向傳播和評估階段的變量，我對向量、維度、矩陣和物體的形狀有了更深入的理解。

然後我開始學習 GPT 和 Transformer 的實現。通過詞嵌入和位置編碼，文本轉化為數字。從本質上講，這與識別手寫數字的簡單神經網絡沒有區別。

Andrej Karpathy 的講座《Let’s build GPT》非常棒。他解釋得很清楚。

第一個原因是這真的是從零開始。我們首先看到如何生成文本。這有點模糊和隨機。第二個原因是 Andrej 能夠非常直觀地解釋事情。Andrej 花了幾個月的時間做 nanoGPT 項目。

我剛剛有了一個新的想法來判斷講座的質量。作者是否真的能寫出這些程式碼？為什麼我不理解，作者忽略了哪些主題？除了這些優雅的圖表和動畫，它們的缺點和缺陷是什麼？

回到機器學習主題本身。正如 Andrej 提到的，dropout、殘差連接、自注意力、多頭注意力、遮罩注意力。

通過觀看更多上述影片，我開始有點理解了。

通過使用 sin 和 cos 函數進行位置編碼，我們得到一些權重。通過詞嵌入，我們將詞語轉化為數字。

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

披薩從烤箱裡拿出來，味道很好。

在這句話中，算法如何知道它指的是披薩還是烤箱？我們如何計算句子中每個詞的相似度？

我們需要一組權重。如果我們使用 Transformer 網絡來完成翻譯任務，每次輸入一個句子，它都能輸出另一種語言的對應句子。

關於這裡的點積。我們在這裡使用點積的一個原因是點積會考慮向量中的每個數字。如果我們使用平方點積呢？我們先計算數字的平方，然後讓它們做點積。如果我們做一些反向點積呢？

關於這裡的遮罩，我們將矩陣的一半數字改為負無窮大。然後我們使用 softmax 使值範圍在 0 到 1 之間。如果我們將左下角的數字改為負無窮大呢？

計劃

繼續閱讀程式碼和論文，觀看影片。只是享受樂趣，跟隨我的好奇心。

https://github.com/karpathy/nanoGPT

https://github.com/jadore801120/attention-is-all-you-need-pytorch