从神经网络到 GPT

YouTube 视频

Andrej Karpathy - Let’s build GPT: 从零开始，逐行编写代码，详细解释。

Umar Jamil - Attention is all you need (Transformer) - 模型解释（包括数学推导），推理和训练

StatQuest with Josh Starmer - Transformer Neural Networks, ChatGPT 的基础，清晰解释!!!

Pascal Poupart - CS480/680 Lecture 19: Attention and Transformer Networks

The A.I. Hacker - Michael Phi - Illustrated Guide to Transformers Neural Network: A step-by-step explanation

我的学习方式

当我读完《神经网络与深度学习》一书的一半时，我开始复制识别手写数字的神经网络示例。我在 GitHub 上创建了一个仓库，https://github.com/lzwjava/neural-networks-and-zhiwei-learning。

这是最困难的部分。如果一个人能从零开始编写代码而不抄袭任何代码，那么他就能很好地理解这个过程。

我复制的代码仍然缺乏 update_mini_batch 和 backprop 的实现。然而，通过仔细观察加载数据、前向传播和评估阶段的变量，我对向量、维度、矩阵和对象的形状有了更好的理解。

我开始学习 GPT 和 Transformer 的实现。通过词嵌入和位置编码，文本被转换为数字。本质上，它与识别手写数字的简单神经网络没有区别。

Andrej Karpathy 的讲座“Let’s build GPT”非常好。他解释得很清楚。

第一个原因是它确实是从零开始的。我们首先看到如何生成文本，它有点模糊和随机。第二个原因是 Andrej 能够直观地解释事情。Andrej 花了几个月时间做 nanoGPT 项目。

我刚刚有了一个评判讲座质量的新想法。作者真的能写出这些代码吗？为什么我不明白，作者遗漏了哪些主题？除了这些优雅的图表和动画，它们有哪些缺点和不足？

回到机器学习话题本身。正如 Andrej 提到的，dropout、残差连接、自注意力、多头注意力、掩码注意力。

通过观看上述更多视频，我开始理解一些东西。

通过 sin 和 cos 函数进行位置编码，我们得到一些权重。通过词嵌入，我们将单词转换为数字。

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

披萨从烤箱里出来，味道很好。

在这句话中，算法如何知道“它”指的是披萨还是烤箱？我们如何计算句子中每个单词的相似性？

我们需要一组权重。如果我们使用 Transformer 网络来做翻译任务，每次我们输入一个句子，它可以输出另一种语言的对应句子。

关于这里的点积。我们在这里使用点积的一个原因是点积会考虑向量中的每个数字。如果我们使用平方点积怎么办？我们先计算数字的平方，然后让它们进行点积。如果我们做一些反向点积呢？

关于这里的掩码，我们将矩阵一半的数字更改为负无穷大。然后我们使用 softmax 将值范围调整到 0 到 1。如果我们将左下角的数字更改为负无穷大呢？

计划

继续阅读代码和论文，观看视频。只要玩得开心，跟随我的好奇心。

https://github.com/karpathy/nanoGPT

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