LLAMA 2 IMPLEMENTATION REPORT
Nhóm 5

Thành viên
Phạm Thành Long
Nguyễn Đức Minh
Trần Tiến Nam

1. Rotary Positional Embeddings (RoPE)

Tensor query và key được chuyển đổi thành dạng số phức bằng cách chia chiều cuối cùng của chúng (head dim) thành hai phần: thành phần thực (real) và thành phần ảo (imaginary)

Ta tính tần số góc quay theo công thức:

freqs = θ^(-i / head_dim)

với i là các chỉ số chẵn trong khoảng [0, head dim]

Ta tính angles (góc) bằng torch.outer giữa vị trí token trong chuỗi (torch.arange(seqlen)) và freqs. Từ đó ta được freqs_cis chứa giá trị sine và cosine của các góc.

Trước khi tính phần thực và phần ảo ta cần reshape hai tensor sine và cosine để khớp với các tensor phần thực và phần ảo tạo bởi query và key.

Ta áp dụng công thức quay cho phần thực và phần ảo

```
rot_query_real = query_real * cos_vals - query_imag * sin_vals
rot_query_imag = query_real * sin_vals + query_imag * cos_vals
rot_key_real = key_real * cos_vals - key_imag * sin_vals
rot_key_imag = key_real * sin_vals + key_imag * cos_vals
```

Cuối cùng ta kết hợp phần thực và phần ảo về với shape ban đầu của query và key

2. Root Mean Square Norm (RMS-Norm)

Chuẩn trung bình bình phương (**Root Mean Square Norm - RMS Norm**) là một phương pháp đo lường độ lớn trung bình của một tập hợp số liệu hoặc vector. Nó được tính bằng cách lấy căn bậc hai của trung bình cộng các bình phương của các phần tử trong tập hợp đó.

$$||x||_{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2 + \epsilon}$$

Trong đó:

- x là tensor đầu vào
- n là số phần tử của tensor x
- x_i là từng phần tử của tensor x

3. Forward của LlamaLayer

- Đầu tiên normalization lớp input đầu vào với hàm RMSNorm để đảm bảo tính ổn định và hiệu suất của model.
- Sau khi chuẩn hóa đầu vào sẽ đi qua lớp Attention để tính mối quan hệ giữa các phần tử trong chuỗi đầu vào.
- Residual connection: Đầu ra của lớp Attention sẽ được cộng với đầu vào ban đầu để duy trì được thông tin cũ và cải thiện khả năng truyền ngược gradient trong quá trình training.
- Tiếp tục đưa kết quả qua một lớp layer normalization để đảm bảo tính ổn định trước khi đưa vào một mạng nơ-ron truyền thẳng.
- Cuối cùng đầu ra của mạng nơ-ron truyền thẳng sẽ được cộng với đầu ra chưa chuẩn hóa của Residual connection giúp duy trì những thông tin quan trọng và tăng tính ổn định của model.

4. AdamW

AdamW là một biến thể của Adam optimizer. AdamW đã khắc phục một số nhược điểm của Adam bằng cách tách biệt hoàn toàn việc cập nhật trọng số (weight update) và suy giảm trọng số (weight decay)

- Cập nhật gradient như Adam

$$g_t = \nabla_\theta L(\theta_t)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t},\ \widehat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon}$$

- Áp dụng Weight Decay (tách riêng)

$$\theta_{t+1} = \theta_{t+1} - \eta\lambda\theta_t$$

5. Attention

Cài đặt attention theo công thức sau:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Trong đó:

- Q (Query): Truy vấn - đại diện cho phần cần tìm kiếm thông tin.
- K (Key): Khóa - đại diện cho các thông tin có sẵn.
- V (Value): Giá trị - chứa thông tin cần trích xuất.
- d_k là số chiều của vector Key (head_dim), dùng để chuẩn hóa giá trị Attention Scores.

6. Generate function:

Hàm generate thực hiện sinh văn bản sử dụng temperature sampling. Nó nhận đầu vào là một chuỗi từ đã được tham chiếu trên vocab của model, và mở rộng nó bằng dự đoán từ thông tin của hidden state cuối cùng.

Trong mỗi bước:

- Hàm cắt chuỗi nếu vượt quá độ dài tối đa.
- Dự đoán từ tiếp theo dựa trên logits của mô hình.
- Nếu temperature = 0.0 thì model chọn từ có xác suất cao nhất
- Nếu temperature > 0.0, mô hình thực hiện temperature sampling, giúp tăng tính ngẫu nhiên bằng cách chia logits cho nhiệt độ và lấy mẫu từ phân phối softmax.
- Từ được chọn sẽ được nối vào chuỗi input và chuỗi mới này sẽ làm input tiếp theo. Model sẽ ngừng sinh khi đạt số lượng từ mới được sinh tối đa

7. Kết quả huấn luyện

```
$ python rope_test.py
Rotary embedding test passed!
```

```
$ python sanity_check.py
Your Llama implementation is correct!
```

```
$ python optimizer_test.py
Optimizer test passed!
```

Setting 1 (Text cotinuation)

```
Temperature is 0.0
I have wanted to see this thriller for a while, and it didn't disappoint. Keanu Reeves, playing the hero John Wick, is this day. He was playing with his toy car, driving it around th
e living room. Suddenly, he heard a loud crash. He had broken the car and was very sad.
John was angry and he shouted at his little brother. He was only three years old and he was only three. He was only three years old. He was very ups
---------------
Wrote generated sentence to outputs/generated-sentence-temp-0.txt.
load model from stories42M.pt
Temperature is 1.0
I have wanted to see this thriller for a while, and it didn't disappoint. Keanu Reeves, playing the hero John Wick, is it!" As John toddled up to the sweet aroma, he held his mum's h
and tight, hoping he would finally reach the top step.
But as he stepped, he felt so heavy, like an apple. He had made a mistake! His mum apologisedDen handlely leaving, so he started to cry.
His m
---------------
Wrote generated sentence to outputs/generated-sentence-temp-1.txt.
```

Setting 2 (zero shot prompting):

- Với dataset SST-5

```
eval: 100%|████████████████████████████████████████████████████████████| 111/111 [00:02<00:00, 44.40it/s]
eval: 100%|████████████████████████████████████████████████████████████| 221/221 [00:04<00:00, 51.16it/s]
dev acc :: 0.215
test acc :: 0.223
```

- Với dataset CFIMDB

```
eval: 100%|████████████████████████████████████████████████████████████| 25/25 [00:02<00:00, 10.64it/s]
eval: 100%|████████████████████████████████████████████████████████████| 49/49 [00:04<00:00, 12.18it/s]
dev acc :: 0.502
test acc :: 0.213
```

Setting 3 (task-specific classification finetuning):

- Với dataset SST-5

```
eval: 100%|████████████████████████████████████████████████████████████| 14/14 [00:01<00:00, 12.59it/s
eval: 100%|████████████████████████████████████████████████████████████| 28/28 [00:02<00:00, 12.59it/s
dev acc :: 0.410
test acc :: 0.401
```

- Với dataset CFIMDB

```
eval: 100%|████████████████████████████████████████████████████████████| 25/25 [00:03<00:00,  7.97it/s]
eval: 100%|████████████████████████████████████████████████████████████| 49/49 [00:05<00:00,  9.14it/s]
dev acc :: 0.857
test acc :: 0.471
```

Sau khi thay đổi các siêu tham số dropout, n_layers, n_kv_heads

```
eval: 100%|████████████████████████████████████████████████████████████| 14/14 [00:01<00:00, 13.03it/s]
eval: 100%|████████████████████████████████████████████████████████████| 28/28 [00:02<00:00, 12.61it/s]
dev acc :: 0.269
test acc :: 0.288
```