CS5370 Deep Learning for Vision – Assignment 5

Submitted by:
Vishal Singh Yadav
CS20MTECH01001

(1) Let t = sequence length.
l = number of layers.
n = number of neurons at each layer.

For RNN model at train time:

Time complexity $= O(t \cdot n^2 \cdot l)$

Space complexity $= O(t \cdot n \cdot l)$

There are k^2 connections for each hidden-to-hidden connection. for a total of $t \cdot k^2 \cdot l$

We need to store $t \cdot k \cdot d$ hidden units.

For RNN model at test time.

Time complexity: $O(t \cdot n^2 \cdot l)$

Space complexity: $O(n \cdot d)$

For transformer model at train time:-

Time complexity $= O(t^2 \cdot n \cdot l)$

Space complexity $= O(t^2 n^2 l)$

For transformer model at test time:-

Time complexity $= O(t^2 n l)$

Space complexity $= O(t n^2 l)$

(b) Transformers have input length longer than max. length of input sequence. This helps it to attend to anywhere in the input at any time.

In cases when number of neurons at each layer n is smaller than sequence length t, self. attention can be restricted to neighbourhood of size n.

This would increase the path length to $O(t/n)$ and the time complexity will increase as the entire sequence cannot be parallelised.

(c) Self attention does look across the tokens of a given input. This does not affect parallelism as the self attention ~~part~~ is only one part of the model ~~and the parallelism~~
Additionally the self attention module is itself parallelised where it calculates the attention for all words parallely.
At higher level in the model, the entire model is parallelised for multiple layers.

(d) layer norm does not look across tokens. ~~layer norm is col~~
layer norm takes the entire input and calculates norm across all features & all elements for each instance independently. These instances can be parallelised for parallelising layer norm.

feed forward network does look through the tokens. It consists of weights that are trained during training. The exact matrix is applied to each token position
Since It can be applied without any interference by other token positions, it can be parallelized by applying to all tokens simultaneously.

<2> <a> $z = \sum_{i=1}^{m} (v_i a_i)$

for decoder, we need the weighted combination of inputs.

Here, a is the weights for individual input and v is the input words.

In terms of Query vector q, Key vector, k, & value vector v.

z can be calculated as $\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right).V$.

~~Summing~~ This is the attention vector.

We can concatenate all z ~~vectors~~ and multiply by weight to get final z.

<b>

<3> $L(q) = \int q(z|x) \log\left(\frac{p(x|z)}{q(z|x)}\right) dz$

Applying Bayes's theorem

$$L(q) = \int q(z|x) \log\left(\frac{p(x|z)\,p(z)}{q(z|x)\,p(x)}\right) dz.$$

$$= \int q(z|x) \left[\log\left(\frac{p(x|z)\,p(z)}{q(z|x)}\right) - \log p(x)\right] dz$$

$$= \int q(z|x) \log\left(\frac{p(x|z)\,p(z)}{q(z|x)}\right) dz + \int q(z|x) \log p(x)\, dz$$

$$= q(z|x) \log\left(\frac{p(x|z)\,p(z)}{q(z|x)}\right) dz + \log p(x)$$

$$= \int q(z|x) [\log p(x|z) + \log p(z) - \log q(z|x)] dz$$

$$= E_{\sim q(z|x)}[\log p(x|z) + \log p(z) - \log q(z|x)]$$

$$= E_{\sim q(z|x)}[\log p(x,z) - \log q(z|x)]$$

In ~~this~~ reconstruction loss is:
$-E_{\sim q(z|x)} \log q(z|x)$

Reconstruction term is:

$E_{\sim q(z|x)}[\log p(x|z)]$

Regularization term is

$-E_{\sim q(z|x)}[\log q(z|x)]$

(4) $f(p,q) = pq$.

$\min_p \max_q f(p,q)$

$f(p,q) := \mathbb{E}[f_{S\varepsilon}(p,q)]$

where S is random sample from dataset.

$\varepsilon \sim N(0, I_d)$ is noise vector

$f_{S\varepsilon}$ measures how accurately descriminator $D(q;\cdot)$ distinguishes S from $G(p;\varepsilon)$

$f_{S\varepsilon}(p,q) := \log(D(q;S)) + \log(1 - D(y; G(x;\varepsilon)))$

using Gradient Descent Ascent method, updates can be made as

$$x_{i+1} = x_i - \nabla_x f(x_i, y_i)$$

$$y_{i+1} = y_i + \nabla_y f(x_i, y_i)$$

(a)

	q_0	q_1	q_2	q_3	q_4	q_5	q_6
	1	2	2	0	-4	-8	-8

p_0	p_1	p_2	p_3	p_4	p_5	p_6
1	0	-2	-4	-4	0	8

(b) By using the above approach, we will not reach an optimal value as the algorithm will not converge but will keep cycling from any start point. Since it is neccesary to find a global maximum for optimal point, this method will not find it.

(c) An equilibrium point is a saddle point. which the GAN needs to reach. A saddle point is a point on graph surface where the slopes in orthogonal directions are all zero but is not a local extremum of the function.