# Assignment 5

CS5370: Deep Learning for Vision/AI5100: Deep Learning/AI2100: Deep Learning
IIT-Hyderabad
Jan-Apr 2021

**Max Marks:** 50
**Due:** 12th April 2021 11:59 pm

## Instructions

- Please use Google Classroom to upload your submission by the deadline mentioned above. Your submission should comprise of a single ZIP file, named `<Your_Roll_No>_Assign5`, with all your solutions, including code.
- For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due. Note that each student begins the course with 10 grace days for late submission of assignments, of which upto 4 grace days can be used for a single assignment. Late submissions will automatically use your grace days balance, if you have any left. You can see your balance on the Marks and Grace Days document, soon to be shared under the course Google drive.
- You have to use PYTHON for the programming questions.
- Please read the department plagiarism policy. Do not engage in any form of cheating - strict penalties will be imposed for both givers and takers. Please talk to instructor or TA if you have concerns.

## 1 Theory (15 marks)

*You can submit your response as a PDF document, which can be typed out in LaTeX/Word, or handwritten and scanned. If handwritten, please ensure legibility of answers.*

1. *(2 + 1 + 1 + 1 = 5 marks)* One of the biggest advantages of transformers over RNNs is "**Parallelism**".

   (a) Let t, l and n be defined as t = sequence length, l = number of layers and n = number of neurons at each layer respectively. Contrast RNN model with transformer model in terms of time complexity and space complexity, both at train time and test time. Express it in terms of t, l and n.

   (b) What happens to the performance of parallelism if n (number of neurons at each layer) is smaller than the sequence length t?

   (c) Self-attention layer looks across the tokens of a given input sequence. Isn't this a bottleneck for parallelism? Explain.

   (d) Does the feed forward network and layer norm look across the tokens? Explain how they support parallelism.

2. *(2 + 2 = 4 marks)* You learned in the attention lecture about $q$ (query), $v$ (value) and $k$ (key) vectors. Let us look at the case of single-head attention. Let us say $q$, $k$, $v$ vectors are from $\mathbb{R}^d$ and there are $m$ value vectors and $m$ key vectors.

   Then attention vector and attention weights can be defined as:

$$z = \sum_{i=1}^{m} (v_i \alpha_i)$$

$$\alpha_i = \frac{\exp(k_i^T q)}{\sum_{i=1}^{m} \exp(k_i^T q)}$$

   (a) Let us say $z$ is evaluated to $v_j$ for some j. What does this mean? Explain using query, key and value vectors.

   (b) Now, take an orthogonal set of key vectors $\{k_1, \cdots, k_m\}$ where all key vectors are orthogonal, that is $k_i \perp k_j$ for all $i \neq j$. Let $||k_i|| = 1$ for all $i$. Let $v_a$, $v_b \in \{v_1, \ldots, v_m\}$ be two of the value vectors from a set of m arbitrary vectors. Express query vector q such that the output z is roughly equal to the average of $v_a$ and $v_b$, that is, 1/2 * ($v_a$ + $v_b$).

3. *(2 marks)* The Variational Autoencoder represents the standard variational lower bound as the combination of a reconstruction term and a regularization term. Starting with the variational lower bound below, derive these two terms, specifying which is the reconstruction term and which is the regularization term.

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

4. *(2 + 1 + 1 = 4 marks)* You may know the minimax problem of generative adversarial networks (GAN). Let us take a simpler example to understand how difficult is GAN minmax problem. Consider a function f(p, q) = pq. What does $min_p max_q$f(p,q) evaluate to? (Clue: minmax problem minimizes the maximum value possible)

   Let us start with the point k and evaluate the function for n steps, by alternating gradient (update first q and then p) with step size 1. Writing out the update step in terms of $p_t, q_t, p_{t+1}, q_{t+1}$ will be helpful.

   (a) As you iterate with each step, enter the values for $(p_t, q_t)$ in the table below. (Assume k = (1,1) and n = 6)

| $q_0$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| $p_0$ | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ |
| 1 | | | | | | |

   (b) By using the above approach, is it possible to reach the optimal value? Why or why not?

   (c) What is the equilibrium point?

# 2 Programming (35 marks)

- The programming questions are shared in "Assignment_5.zip". Please follow the instructions in the notebook. Turn-in the notebook via Google Classroom once you finish your work.
- Please check the corresponding Notebook for Marks breakdown details for each questions.

**Question-1 : Image Captioning** *(15 marks)*

Find out the caption generated by the model when an input image ("image3.jpg") is being fed to the encoder?

(a) Implementation of Encoder CNN module: 6 marks

(b) Implementation of Decoder LSTM module: 4 marks

(c) Input pre-processing, Loading Vocabulary and Parameters of Encoder and Decoder module in order to perform inference with the given input image : 5 marks

**Question-2 : VAE** *(13 marks)*

Given an input image $x$ and parameters of an Encoder and Decoder of a Variational AutoEncoder(VAE) , Compute the variational Lower bound of $p(x)$, i.e. probability of $x$?

Hint: $\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathrm{KL}\left[q_\phi(z|x)||p(z)\right]$ Note: Use torch.exp() library function in order to compute exponential.

(a) Define Encoder, Decoder and VAE module and load the given pretrained model weight : 5 marks

(b) Given an input image, Do the Forward pass through the encoder module, Apply Re-parameterization trick to sample a latent using encoder output, and finally take the forward pass through the decoder in order to reconstruct the input image from the latent : 5 marks

(c) Compute Variational Lower bound (ELBO): 3 marks

**Question-3 : Transformer** *(7 marks)*

Implement Custom Encoder layer using Multi-Head-Attention for Transformer network as given in question 3 in the notebook and finally report the validation loss on the given setup.

(a) Implement the Custom Encoder layers of Transformer as asked in question: 4 marks

(b) Implement the "forward" function of Custom Encoder layers as asked in question: 3 marks

**[Optional] Question-4 : Normalizing Flow** *(ungraded)*

Compute "Transformed" Probability Density Function (pdf) of $\mathbf{y}$, i.e. $q_1(\mathbf{y})$. Where, $\mathbf{y}$ is obtained from $\mathbf{z}$ via an invertible transformation $f$ as shown below:

Let $\mathbf{z} \sim q_0(\mathbf{z})$ where $q_0(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. Let $f(\mathbf{z})$ be an invertible transformation given by

$$\mathbf{y} = f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^\top \mathbf{z} + b)$$

The pdf of $\mathbf{y}$ is given by $q_1(\mathbf{y})$

Find out the mean and standard deviation of $q_1(\mathbf{y})$; given the values of $q_0(\mathbf{z})$ , $\mathbf{f}$ and all the parameters required to compute $f(\mathbf{z})$?

(a) Implement functions required to compute the transformed distribution $q1(y)$: 2 marks

(b) Define function to compute $y = f(z)$ and $|det(\frac{df}{dz})|$ as given in the question : 4 marks

(c) Finally, compute the Transformed Probability Density Function of y , i.e. $q1(y) = q0/|det(\frac{df}{dz})|$ as given in the question : 1 mark