Mock Exam

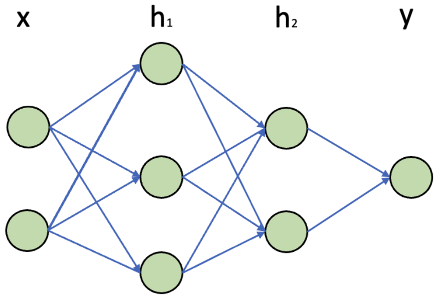

Figure 1: Multi-layer Perceptron.

Problem 1: Forward Propagation (mock) (20')

Let's consider the neural network illustrated in Figure 1.

It has the following architectures:

$$\begin{aligned} h_1 &= \sigma(W_1 * x + b_1) \\ h_2 &= \mathrm{ReLU}(W_2 * h_1 + b_2) \\ \hat{y} &= \sigma(W_3 * h_2 + b_3) \end{aligned} \tag{1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function and $\mathrm{ReLU}(x) = \max(0, x)$ is the ReLU function. Suppose:

The input $x = \begin{bmatrix} 2 \\ 6 \end{bmatrix}$,

three sets of weights $W_1 = \begin{bmatrix} 0.3 & 1.6 \\ -0.8 & 0.1 \\ 0.8 & -0.5 \end{bmatrix}$, $W_2 = \begin{bmatrix} 1.8 & 0.6 & -0.3 \\ 0.2 & -1.5 & 0.7 \end{bmatrix}$, $W_3 = \begin{bmatrix} 0.9 & 1.6 \end{bmatrix}$,

and biases $b_1 = \begin{bmatrix} 0.3 \\ -0.1 \\ 1.0 \end{bmatrix}$, $b_2 = \begin{bmatrix} 0.2 \\ -0.6 \end{bmatrix}$, $b_3 = 0.5$.

1. (5') Calculate the value of h_1. What size does h_1 have?
2. (5') Calculate the value of h_2. What size does h_2 have?
3. (5') Calculate the value of $\hat{y}$. What size does $\hat{y}$ have?
4. (5') Explain the role of sigmoid and ReLU functions in the neural network. What would happen if we don't use them?

Problem 2: Backward Propagation (mock) (20')

Let's still consider the neural network illustrated in Figure 1. Suppose:
the ground truth $y = 1$,
the loss function is $\mathcal{L} = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$.
(hint: $\sigma'(x) = \sigma(x)(1 - \sigma(x))$)

1. (5') Calculate the value of $\partial \mathcal{L} \ / \ \partial \hat{y}$.

2. (5') Calculate the value of $\partial \hat{y} \ / \ \partial h_2$.

3. (5') Calculate the value of $\partial \mathcal{L} \ / \ \partial W_3$.

Problem 3: RNN (mock) (20')

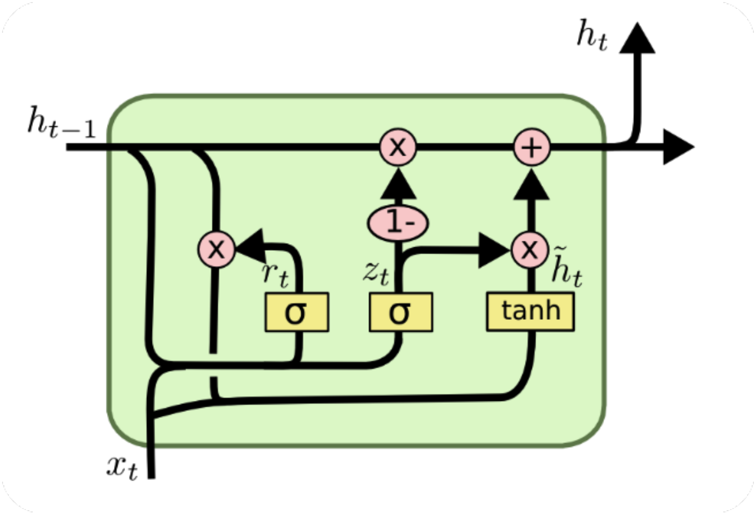

Figure 2: GRU cell

1. **[10']** Based on Figure 2, what is the update mechanism of the GRU? The bias should be token into consideration. (The weight matrix for the reset gate can be indicated by W_{rx} and W_{rh}, and the bias is b_r. Similar for other gates.)

2. **[5']** What is the main difference between LSTM and GRU?

3. **[5']** Consider the GRU layer defined by the following code snippet:

```
"""
From Pytorch Docs:
torch.nn.GRU(input_size, hidden_size, num_layers=1, bias=True,
batch_first=False, dropout=0, bidirectional=False)
"""
embed_size = 8
hidden_size = 16
gru_layer = torch.nn.GRU(embed_size, hidden_size, batch_first=True,
bidirectional=True)
output, (hn, cn) = gru_layer(batch_embeds)
```

Suppose the shape of batch_embeds is (b, l, e), where $b = 4$ is the batch size, $l = 10$ is the sequence length, and $e = embed\_size = 8$ is the embedding size. What is the output shape?

Problem 4: Transformer (mock) (20')

1. **[5']** Given the multi-head self-attention defined below, assuming MultiHead has the same shape as X, how many parameters does it have?

$$\begin{aligned}
&X \in \mathbb{R}^{k\times d};\ W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d\times d}\\
&Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V\\
&\text{Head}_i = \text{Attention}(Q_i, K_i, V_i) = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}}\right) V_i\\
&\text{MultiHead} = \text{Concat}(\text{Head}_1, \ldots, \text{Head}_h) W^O
\end{aligned}$$

2. **[5']** What are the main differences between Transformer encoder layers and decoder layers?

3. **[5']** List several pre-trained language models which are Transformer encoder-based, Transformer decoder-based respectively.

4. **[5']** What are the two pre-training tasks for BERT? Briefly descibe the tasks and their inputs and targets.

Problem 5: Regularization (mock) (10')

Assume you are training a classification model with 4 output units and the loss function J as defined below. The weight parameters, regularization parameter, expected and predicted outputs for 4 examples are given below

$$J = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i$$

$$\theta = \begin{bmatrix} 0.5 \\ -0.4 \\ 0.6 \\ -0.2 \end{bmatrix}, \lambda = 0.1,$$

$$y_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \hat{y_1} = \begin{bmatrix} 0.10 \\ 0.20 \\ 0.10 \\ 0.60 \end{bmatrix}, y_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \hat{y_2} = \begin{bmatrix} 0.30 \\ 0.20 \\ 0.45 \\ 0.05 \end{bmatrix},$$

$$y_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \hat{y_3} = \begin{bmatrix} 0.20 \\ 0.55 \\ 0.10 \\ 0.15 \end{bmatrix}, y_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \hat{y_4} = \begin{bmatrix} 0.75 \\ 0.10 \\ 0.10 \\ 0.05 \end{bmatrix}$$

Redefine your loss function J with

1. **[10']** L1 Regularization and calculate the loss.