Reyhane Shahrokhian 99521361

HomeWork5 of DeepLearning Course

Dr. DavoodAbadi

Question 1:

1-1:

It's suitable for second and third options. The many-to-one RNN architecture means that the network processes a sequence of inputs and produces a single output.

In the first option, the output is a sequence of data so the many-to-one RNN is not useful.

In the second option, the input is actually a sequence of words and the output is 0 or 1 so the many-to-one RNN suits this task.

In the third one, like the second the input is a sequence and the output is a label which is a single one.

1-2:

Option 3 is correct because the behavior of the cat is influenced by the current weather and the weather of the past days, and future information is not crucial, so a simple RNN should be considered as a suitable choice.

1-3:

Option 3 is correct. In time step t of a RNN, the network computes $P(y_t|y_1,..., y_{t-1})$. This reflects the sequential nature of RNNs, where the prediction at each time step depends on the previous time steps in the sequence.

Question 2:

2-1:

We know :

$$J_t =- \sum_{i=1}^{2} y_{t,i} log \widehat{y}_{t,i} , \widehat{y}_t = \sigma(o_t)$$

So :

$$\frac{\partial J_t}{\partial o_t} = \frac{\partial J_t}{\partial \widehat{y}_t} \times \frac{\partial \widehat{y}_t}{\partial o_t} =- \sum_{i=1}^{2} \frac{y_{t,i}}{\widehat{y}_{t,i}} \times \sigma'(o_t) =- \sum_{i=1}^{2} y_{t,i} \times (1 - \widehat{y}_{t,i})$$

2-2:

We know :

$$g_{ot} = \frac{\partial J_t}{\partial o_t} , o_t = w_{yh} h_t , h_t = \psi(z_t) , z_t = w_{hh} h_{t-1} + w_{hx} x_t$$

So:

$$\frac{\partial J_t}{\partial h_i} = \frac{\partial J_t}{\partial \widehat{y}_{t,i}} \times \frac{\partial \widehat{y}_{t,i}}{\partial o_t} \times \frac{\partial o_t}{\partial h_t} \times \prod_{k=0}^{t-i-1} \frac{\partial h_{t-k}}{\partial h_{t-k-1}}$$

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial z_t} \times \frac{\partial z_t}{\partial h_{t-1}} = \psi'(z_t) \times w_{hh}$$

$$\frac{\partial J_t}{\partial h_i} = g_{ot} \times w_{yh} \times w_{hh}^{t-i} \times \prod_{k=0}^{t-i-1} \psi'(z_t)$$

As the network is not bidirectional so the $\frac{\partial J_t}{\partial h_i}$ is valid if $i \leq 3$.

For example i = 3:

$$\frac{\partial J_t}{\partial h_3} = g_{ot} \times w_{yh}$$

2-3:

We know:

$g_{ht} = \frac{\partial J_t}{\partial h_i}$, $o_t = w_{yh} h_t$, $h_t = \psi(z_t)$, $z_t = w_{hh} h_{t-1} + w_{hx} x_t$

So:

$$\frac{\partial J_t}{\partial w_{hh}} = \sum_{i=1}^{t} \frac{\partial J_t}{\partial h_i} \times \frac{\partial h_i}{\partial w_{hh}}$$

$$\frac{\partial h_t}{\partial w_{hh}} = \frac{\partial h_t}{\partial z_t} \times \frac{\partial z_t}{\partial w_{hh}} = \psi'(z_t) \times h_{t-1}$$

$$\frac{\partial J_t}{\partial w_{hh}} = \sum_{i=1}^{t} g_{ht} \times \psi'(z_i) \times h_{t-1}$$

2-4:

We know:

$$gw_{hh,t} = \frac{\partial J_t}{\partial w_{hh}}$$

$$\frac{\partial J}{\partial w_{hh}} = \sum_{t=1}^{3} \frac{\partial J_t}{\partial w_{hh}} = \sum_{t=1}^{3} gw_{hh,t}$$

Question 3:

3-1:

First of all q and keys should be multiplied together and then the argmax function should be applied on them to get the index:

Argmax(q.keys = [-2, 3, 0, 6]) = 3

So the index 3 of values is the answer:

Output = [6, 1, 2]

3-2:

Using argmax as attention in training models can have certain drawbacks. Argmax is a non-differentiable operation, meaning it doesn't have a well-defined gradient. In the context of attention mechanisms, the argmax operation is often used to select the most relevant element in a sequence, but it poses challenges during backpropagation because gradients cannot flow through it. This makes it difficult to compute gradients with respect to the parameters of the attention mechanism, which are crucial for training the model.

When using argmax as an attention mechanism, it becomes challenging to directly improve the queries or keys during training because argmax is a non-differentiable operation. This means that the gradients necessary for updating the parameters of the queries or keys cannot be computed through the argmax operation, making it difficult to perform gradient-based optimization.

Argmax also has some advantages. It's a simple and computationally efficient method for selecting the most relevant information from the input sequence. It involves choosing the element with the maximum attention weight, making it easy to implement and understand.

The attention weights obtained through argmax are easy to interpret. The model essentially focuses on the position in the input sequence with the highest relevance, providing insights into the decision-making process.

Question 4:

As the TA mentioned the explanation is not needed and the notebook file is attached.