Pattern Recognition and Machine Learning: Homework 12

Qingru Hu 2020012996

2023 年 5 月 25 日

Problem 1

Give expression for t and y:

$$t = w(wu_1 + u_2) + u_3$$
$$y = w(w(wu_1 + u_2) + u_3)$$

Compute $\frac{\mathrm{d}y}{\mathrm{d}w}$ and $\frac{\partial y}{\partial p}$:

$$\frac{\mathrm{d}y}{\mathrm{d}w} = 3w^2u_1 + 2wu_2 + u_3$$
$$\frac{\partial y}{\partial p} = w^3$$

Problem 2

$$\frac{\partial L}{\partial C_{T-1}} = \frac{\partial L}{\partial h_{T-1}}\frac{\partial h_{T-1}}{\partial C_{T-1}} + \frac{\partial L}{\partial C_T}\frac{\partial C_T}{\partial C_{T-1}}$$

$$\because h_{T-1} = \tanh(C_{T-1}) \odot o_T$$
$$\therefore \frac{\partial h_{T-1}}{\partial C_{T-1}} = \mathrm{diag}(o_T)(1 - \tanh^2 C_{T-1})$$

$$\because C_T = f_T \odot C_{T-1} + i_T \odot \tilde{C}_T$$
$$\therefore \frac{\partial C_T}{\partial C_{T-1}} = \mathrm{diag}\left(f_T\right)$$

$$\frac{\partial L}{\partial C_{T-1}} = \frac{\partial L}{\partial h_{T-1}}\mathrm{diag}\left(o_T\right)\left(1 - \tanh^2 C_{T-1}\right) + \frac{\partial L}{\partial C_T}\mathrm{diag}\left(f_T\right)$$

The cell state in the LSTM is separately processed from the hidden layers and only additive updates are done in the cell state preventing gradient vanishing in that path during training. However, the use of nonlinear activation function in LSTM results in vanishing gradients in other paths than the cell state. The long term dependencies and relations are encoded in the cell state vectors and it's the cell state derivative that can prevent the LSTM gradients from vanishing.

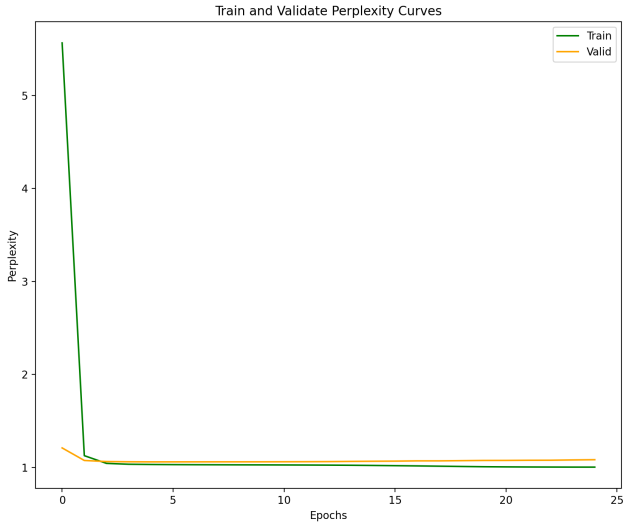
Problem 3

The model is constructed as follows. The perplexity curves is shown as below.

```python
def __init__(self, nvoc, dim = 256, nhead = 8, num_layers = 4):
    super(LMModel_transformer, self).__init__()
    self.drop = nn.Dropout(0.5)
    self.encoder = nn.Embedding(nvoc, dim)
    # WRITE CODE HERE witnin two '#' bar
    ####################################
    # Construct you Transformer model here. You can add additional parameters to the function.
    self.dim = dim
    self.transformer = nn.Transformer(d_model=dim,nhead=nhead,num_encoder_layers=num_layers,num_decoder_layers=num_layers)
    ####################################
    self.decoder = nn.Linear(dim, nvoc)
    self.init_weights()
```

```python
def forward(self, input):
    #print(input.device)
    embeddings = self.drop(self.encoder(input))

    # WRITE CODE HERE within two '#' bar
    ########################################
    # With embeddings, you can get your output here.
    # Output has the dimension of sequence_length * batch_size * number of classes
    L = embeddings.size(0)
    src_mask = torch.triu(torch.ones(L, L) * float('-inf'), diagonal=1).to(input.device.type)
    src = embeddings * math.sqrt(self.dim)
    #TODO: use your defined transformer, and use the mask.
    tar = input[1:, :]
    tar_with_zeros = torch.cat([tar, torch.zeros(1, tar.size(1), dtype=tar.dtype, device=input.device)], dim=0)
    tgt = self.encoder(tar_with_zeros) * math.sqrt(self.dim)
    output = self.transformer(src, tgt, src_mask=src_mask)
    ########################################
    output = self.drop(output)
    decoded = self.decoder(output.view(output.size(0)*output.size(1), output.size(2)))
    return decoded.view(output.size(0), output.size(1), decoded.size(1))
```



A source mask is a tensor that is used to control the attention mechanism in the transformer. It can prevent attention to certain positions or modify the attention weights. For example, a source

Problem 3 continued on next page...

mask can be used to mask out the padding tokens in a sequence or to implement causal masking for autoregressive models.

Problem 4

Text preprocessing is an essential step to prepare the corpus for modeling and directly affects the natural language processing (NLP) application results. For instance, precise tokenization increases the accuracy of part-of-speech (POS) tagging, and retaining multiword expressions improves reasoning 1.

On the other hand, image preprocessing is used to prepare images for analysis by removing noise, enhancing features, and transforming images into a format that can be easily analyzed by machine learning algorithms.

Acknowledgement: Thank Yihan Zhou (周亦涵-2020012853) for the discussion about this homework.