Homework II

Li Hantao, G2101725H, MSAI, hli038@e.ntu.edu.sg

I. QUESTION I

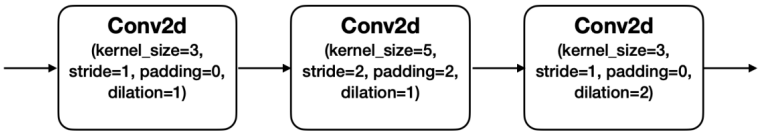
The following questions are related to image segmentation.

i) What is the difference between semantic segmentation and instance segmentation?

Answer:

Semantic segmentation will assign a class to each pixel in the image, but objects in the same category will not be distinguished. Instance segmentation only classifies specific objects we are interested instead of all the pixel in the image but assign a different label to different objects in the same category. For example, in a picture of a crowd, semantic segmentation will label all people as "person" and instance segmentation will label them as "person1", "person2".....

ii) Given the following network, calculate the receptive field.



Answer:

We can assume that the third Conv2d layer with dilation=2 has a bigger kernel size to reship it to kernel_size=5 and dilation=1. In this way, we can calculate the receptive easily:

$$R_1 = 1 + (F_1 - 1) \cdot S_0 = 1 + 2 = 3$$

$$R_2 = R_1 + (F_2 - 1) \cdot S_0 \cdot S_1 = 3 + 4 = 7$$

$$R_3 = R_2 + (F_3 - 1) \cdot S_0 \cdot S_1 \cdot S_2 = 7 + 4 \cdot 2 = 15$$

iii) Spatial context is particularly important for segmentation tasks. List at least four techniques that improve semantic segmentation in terms of spatial context.

Answer:

1. GCN (Global Convolutional Network) directly enlarges the size of kernel size [1], which can generate a larger receptive field and get more information about the spatial context.
2. ParseNet [2], utilizing global pooling to calculate a global feature as context information. It aims to enlarge the receptive field of each pixel to obtain richer context information.
3. PSPNet (Pyramid Scene Parsing Net) to get information from different scales [3]. It integrates the features of four different scales and extracts global context information through different region-based context aggregation.
4. ASPP (Atrous Spatial Pyramid Pooling) in DeepLab [4-5], using three groups of parallel dilated convolution operations with different rates to calculate the context information of each location, and the subsequent V3 additionally introduces the global average pooling operation to enhance the context information of each location.
5. Using graph convolution to model the dependency between different regions or categories [6], using 1x1 convolution for spatial projection and graph convolution for information diffusion. The dependence of different nodes in graph volume products also depends on 1x1 convolution learning.

6. Using self-attention to establish the spatial-wise relationship, like OCNet [7]. Because the receptive field will degenerate, even using global pooling cannot bring the receptive field to the whole picture. Therefore, establishing a long-distance context relationship has become a key point. OCNet establishes an object context map for each pixel according to the feature similarity between pixels. In the implementation, flatten h * w * C to n * C, and calculate the affinity between pixels.
7. Markov Random Field.

iv) Given a transposed convolution kernel as follow, whose stride=1, padding=0, dilation=1,

$$\boldsymbol{Kernel} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \qquad \boldsymbol{Input} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Answer:

In the normal definition of transposed convolution, we use the kernel and input to get the answer:

$$\boldsymbol{Output} = \begin{pmatrix} 5 & 14 & 11 & 6 \\ 19 & 43 & 33 & 16 \\ 15 & 33 & 23 & 10 \\ 9 & 18 & 11 & 4 \end{pmatrix}$$

In the processing of BP, which utilizes the transposed convolution to speed up the calculation, the 'transposed convolution' will flip the kernel first, such as in Pytorch. In this way, the answer will be:

$$\boldsymbol{Output} = \begin{pmatrix} 1 & 4 & 7 & 6 \\ 5 & 17 & 27 & 10 \\ 9 & 27 & 37 & 26 \\ 9 & 24 & 31 & 20 \end{pmatrix}$$

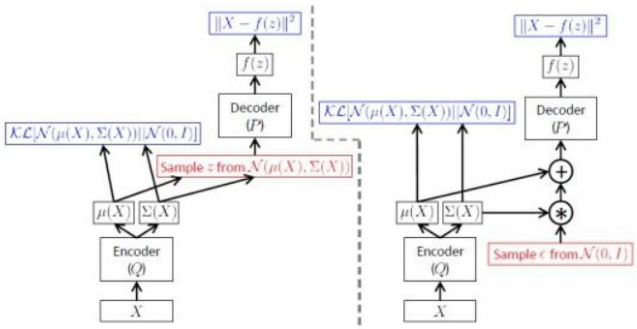
II. QUESTION II

What is the purpose of the reparameterization trick in VAE?

Answer:

Backpropagation cannot flow through a random node because we cannot do differentiate for sampling. The reparameterization trick separates the uncertainty of random variables so that the intermediate nodes that cannot do BP can be derived. More specifically, we want to train the VAE by finding the maximum variational lower bound (the formula in the slides is shown below). When we sample Z, we did a non-differentiable calculation, which would make the gradient between encoder and decoder unable to be transferred (like the figure on the left). However, in our assumption, Z must depend on encoder Q. Thus, instead of sampling Z directly from the distribution obtained by Q, we first sample a value from a standard normal distribution ε, then we transform this value through μ and Σ into the distribution obtained by Q (like the figure on the right). In this way, the whole network is differentiable.

$$E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)$$



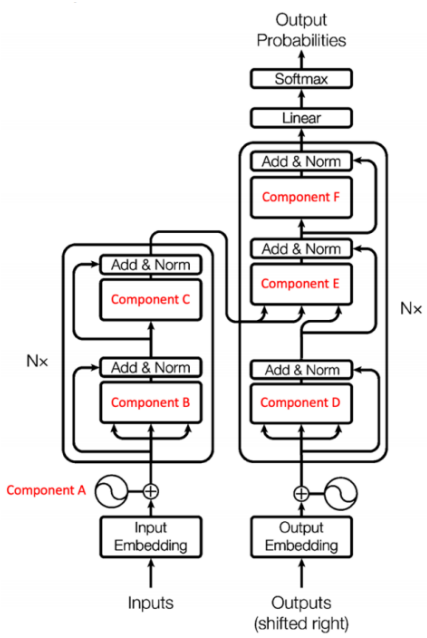
III. QUESTION III

What is the difference between GANs and Conditional GANs?

Answer:
Conditional GANs are conditional models based on GANs. If the generator and discriminator are applicable to some additional conditions, such as class labels, the data generation process can be guided by adding them to the input layer and inputting it into the generator and discriminator for adjustment. If the condition is class label y, Conditional GANs can be considered an improvement in transforming the unsupervised GANs model into a supervised model. In this way, Conditional GANs learn p(x|y) instead of p(x), making the generator and discriminator both take label y as an additional input.

IV. QUESTION IV

The Transformer architecture proposed by Vaswani et al. in "Attention is All You Need" NIPS 2017 is shown in Figure.



i) Write the name of components A, B, C, D, E, and F.

Answer:

Component A: Positional Encoding.	Component D: Masked Multi-Head Attention.
Component B: Multi-Head Attention.	Component E: Multi-Head Attention.
Component C: Feed Forward.	Component F: Feed Forward.

ii) Explain the role of Component A and suggest a way to generate the output encoding of this component.

Answer:

Without the positional encoding implementation, the self-attention network cannot get the information on the position/order of the input vectors. For example, when we are doing the POS tagging, the order of input words is super important. Thus, we add an additional vector, named positional encoding, to give information about the relative or absolute position of the tokens in the sequence.

To generate the positional encoding, this paper utilizes the Sinusoidal positional encoding, which is shown below:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

where pos is the position, and i is the dimension. This makes each dimension contain some position information, and the position coding of each position character is different. Moreover, we can implement learnable encodings, such as RNN and FLOATER.

iii) What are the inputs of Component E?

Answer:

The encoder-decoder attention layer (or cross-attention layer) has an input similar to the standard multi-head self-attention layer. However, in this layer, we use the Q matrix gained from the decoder layer below while using the K, V matrix from the output of the encoder stack, which can help the decoder focus on appropriate places in the input sequence.

iv) Explain why masked self-attention is needed in the decoder of the Transformer. Suggest a way to achieve masked self-attention in practice.

Answer:

Masked self-attention means only allowing to attend to earlier positions in the output sequence. This is done by masking future positions. The reasons are as follows:

In the training process, the input of the lowest decoder layer is the ground truth. When we calculate loss, if we use normal self-attention, the output contains the information on the right (especially the information of the next word we want to predict), which uses the GT information, so that the model is cheating. We cannot know the future information in advance in the actual reasoning process.

In the inference process, we need to keep the prediction of repeated words consistent, which means that the attention value of each step must be kept unchanged, so we need to mask the future words. This also makes the inference process of the model consistent with the training stage.

In Transformers' code [8], the masked self-attention is achieved using a 0-1 (False-True) matrix. Specifically, the value of matmul of Query and Key after self-attention is set to the minimum value. A lower triangular matrix completes the implementation with 1 at the bottom left and 0 at the top right, which can mask the value of the Attention matrix (matmul of the Q and K).

```
def subsequent_mask(size):
    "Mask out subsequent positions."
    attn_shape = (1, size, size)
    subsequent_mask = np.triu(np.ones(attn_shape), k=1).astype('uint8')
    return torch.from_numpy(subsequent_mask) == 0
```

[1] Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2017). Large kernel matters--improve semantic segmentation by global convolutional network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4353-4361).
[2] Liu, W., Rabinovich, A., & Berg, A. C. (2015). Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579.
[3] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2881-2890).
[4] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848.
[5] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587
[6] Chen, Y., Rohrbach, M., Yan, Z., Shuicheng, Y., Feng, J., & Kalantidis, Y. (2019). Graph-based global reasoning networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 433-442).
[7] Yuan, Y., Huang, L., Guo, J., Zhang, C., Chen, X., & Wang, J. (2018). Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916.
[8] https://github.com/harvardnlp/annotated-transformer