Junyu Yin G2101985L

Question 1

i)

Semantic segmentation aims to label each pixel in the image with a category label. It doesn't differentiate instances and only cares about pixels. For example, assuming there is more than one cat in an image, a model performing a semantic segmentation task will label all cats with the same label.

In contrast, instance segmentation further identifies the pixels that belong to each object. It separates instances instead of merging them.

ii)

Case 1 (a forward process that only generates a 1×1 map, considering padding):
Suppose there is an input image of size $N \times N$. The given network does the following things:

$$N \times N \rightarrow (N-2) \times (N-2) \rightarrow \frac{N-1}{2} \times \frac{N-1}{2} \rightarrow \frac{N-9}{2} \times \frac{N-9}{2}.$$

By solve the following equation

$$\frac{N-9}{2} = 1,$$

we get that the receptive field is 11×11.
Case 2 (a backward process that uses a 1×1 pixel in the last feature map, regardless of padding):
Suppose we use a 1×1 pixel in the last feature map. A backward inference is as follows:

$$1 \times 1 \rightarrow 5 \times 5 \rightarrow 13 \times 13 \rightarrow 15 \times 15.$$

we get that the receptive field is 15×15.

iii)

Dilated Convolution; Skip Connections; Pooling; Pyramid Scene Parsing Net; Markov Random Field; Self-attention like CCNet.

iv)

The output matrix is

$$\begin{bmatrix} 1 & 4 & 7 & 6 \\ 5 & 17 & 27 & 20 \\ 9 & 27 & 37 & 26 \\ 9 & 24 & 31 & 20 \end{bmatrix}$$

Question 2

In VAE's original form, the decoder samples z directly from the parametric model $q_\phi(z|x)$. To train this model, we need to backpropagate through this random sampling process and that is the problem because this random sampling process is not differentiable. But by using this trick, we can compute the gradients of μ and Σ in the sampling process.

In short, the reparameterization trick helps the gradients to flow through the sampling process ($z \sim q_\phi(z|x)$) during the backpropagation.

Question 3

Conditional GANs learn $p(x|y)$, while GANs learn $p(x)$. As the result, Conditional GANs can more or less control the generation process, but GANs can only generate data randomly.

Question 4

a)

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

b)

Positional Encoding injects the information about the relative or absolute position of the tokens in the sequence into input word embeddings.

It can be achieved by using the following sinusoidal positional encoding:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

where pos is the position and i is the dimension.

c)

K, V transformed from the output of the top encoder and Q from the decoder's output in the previous step.

d)

In the decoder, the self-attention layer is only allowed to attend to earlier positions in the output sequence. Otherwise, the decoder can cheat during the training by just attending to the current position when predicting the current token.

This can be done by masking future positions (setting them to -inf) before the softmax step in the self-attention calculation, that is, by adding the following matrix to QK^T.

$$\begin{pmatrix} 0 & -inf & \dots & -inf \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -inf \\ 0 & 0 & \dots & 0 \end{pmatrix}$$