NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Master in Science (Artificial Intelligence)

(MSAI)

AI-6126

Homework 2

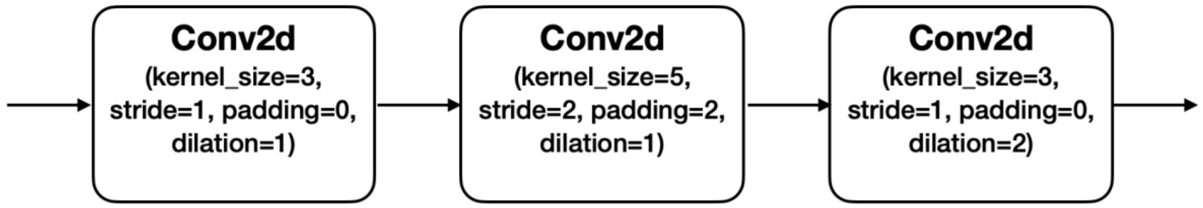
Name of Student:
Teo Lim Fong

Matriculation No: G2101964G

Question 1

i) Semantics segmentation does not differentiate different instances in the same class. For example, if there are two dogs and one cat, the two dogs will have the same colour and the cat will have different colour as label. Instance segmentation differentiate different instances in the same class. For example, if there are two dogs and one cat, all three animals will have three different colours as label.

ii)


Conv2d
(kernel_size=3, stride=1, padding=0, dilation=1)
Conv2d
(kernel_size=5, stride=2, padding=2, dilation=1)
Conv2d
(kernel_size=3, stride=1, padding=0, dilation=2)

The formula for finding receptive field is:

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$
$$j_{out} = j_{in} * s$$
$$r_{out} = r_{in} + (k-1) * j_{in}$$

For dilation, the formula is:

$k' = d(k\text{-}1) + 1$, where d is the dilation factor and k is the kernel size

At layer 0,

$r_0 = 1$

$j_0 = 1$

$n_0 =$ image size (not given)

In conv2d layer 1,

k = 3

s = 1

p = 0

d = 1

At layer 1,

$j_1 = 1 * 1 = 1$

$k' = d(k\text{-}1) + 1 = 1(3\text{-}1) + 1 = 3$

$\boldsymbol{r_1 = r_0 + (k' - 1) j_0}$ **= 1+ (3-1) (1) = 3**

In conv2d layer 2,

k = 5

s = 2

p = 2

d = 1

At layer 2,
$j_2 = 1 * 2 = 2$
$k' = d(k\text{-}1) + 1$ = 1(5-1) + 1 = 5
$\boldsymbol{r_2} = \boldsymbol{r_1} + (\boldsymbol{k'} - \mathbf{1})\boldsymbol{j_1}$ **= 3+ (5-1) (1) = 7**

In conv2d layer 3,
k = 3
s = 1
p = 0
d = 2

At layer 3,
$j_3 = 2 * 1 = 2$
$k' = d(k\text{-}1) + 1$ = 2(3-1) + 1 = 5
$\boldsymbol{r_3} = \boldsymbol{r_2} + (\boldsymbol{k'} - \mathbf{1})\boldsymbol{j_2}$ **= 7 + (5-1) (2) = 15**

Therefore, the receptive field for layer 1 is 3, layer 2 is 7 and layer 3 is 15.

iii) Dilated Convolution, Pyramid Pooling Module, Skip Connection, Markov Random Field

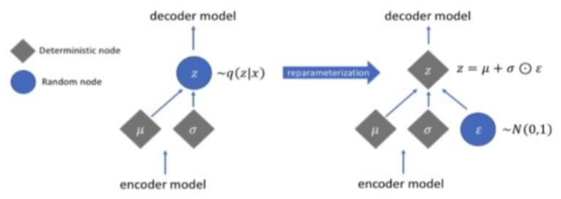
iv)
$$\begin{bmatrix}1&2&3&0\\2&3&4&0\\3&4&5&0\\0&0&0&0\end{bmatrix}+\begin{bmatrix}0&2&4&6\\0&4&6&8\\0&6&8&10\\0&0&0&0\end{bmatrix}+\begin{bmatrix}0&0&0&0\\3&6&9&0\\6&9&12&0\\9&12&15&0\end{bmatrix}+\begin{bmatrix}0&0&0&0\\0&4&8&12\\0&8&12&16\\0&12&16&20\end{bmatrix}$$

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

Question 2

In VAE, the bottle neck is replaced by two components – mean vector and standard deviation vector instead of just a latent vector. Then these vectors are feed into the decoder network by taking a sample from the distribution. The problem is that we cannot run backpropagation through a sampling node (non-deterministic). Therefore, in order to run the backpropagation, we need to apply reparameterization trick.

The sampled latent vector can actually be seen as a sum of fixed μ with σ multiplied by ϵ, where the μ and σ is trainable parameter and ϵ always have N (0, 1). In the diagram below shown that the latent vectors z is a non-deterministic which indicate that it is indifferentiable. However, after using the reparameterization trick, the latent z is now a deterministic which mean we can apply backpropagation.



Both the μ and the σ is deterministic because these parameters are trainable and ϵ is non-deterministic because it does not require any training.

Question 3

In GANs, there is no control of which data is going to generate. However, in Conditional GANs, there is control of which data is going to generate by adding an additional conditional possibility y in the network.

Question 4

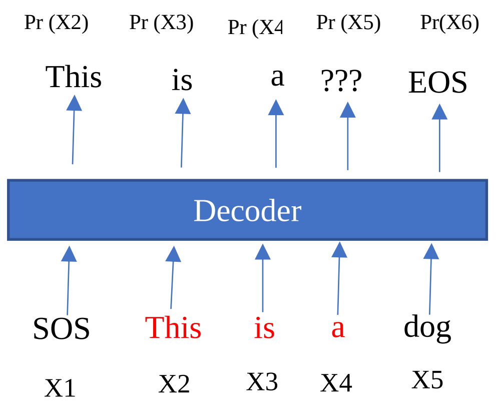
a) A – Position Encoding
B – Multi-Head Attention
C – Feed Forward
D – Masked Multi-Head Attention
E – Multi-Head Attention
F – Feed Forward

b) It is to retain the correct position or sequence of the sentence. In position encoding, it uses sine and cosine curves to find the correct sequence.

c) The value and keys are from the encoder of the source sentence (NLP) and the query are the output from the target sentence (NLP). Or I also can rephrase it as ‘The value and keys are from the output of the encoder and the query are the output from the decoder.’

d) Masked self-attention is needed in order to predict the next token (example ‘x3’) using the current tokens (‘x1’ and ‘x2’) in a sentence. If we do not mask the tokens, it is considered as ‘cheating’ in the decoder during training.

Let’s say we have this sentence ‘This is a dog’ and we masked the word ‘dog’. The decoder has to predict the next token by using the last few words (highlighted in red). *SOS = Start of the sentence and EOS = End of the sentence



Example of creating a masked self-attention for decoder using code.

```
def create_mask(size):

    return torch.triu(torch.ones(size, size) * float('-inf'), diagonal=1)
```

```
masked_selfattention = create_mask(3) #let assume the size is 3
print(masked_selfattention)
```

```
tensor([[0., -inf, -inf],
        [0., 0., -inf],
        [0., 0., 0.]])
```

Question 4: The Transformer architecture proposed by Vaswani et al. in "Attention is All You Need" NIPS 2017 is shown in Figure Q1.

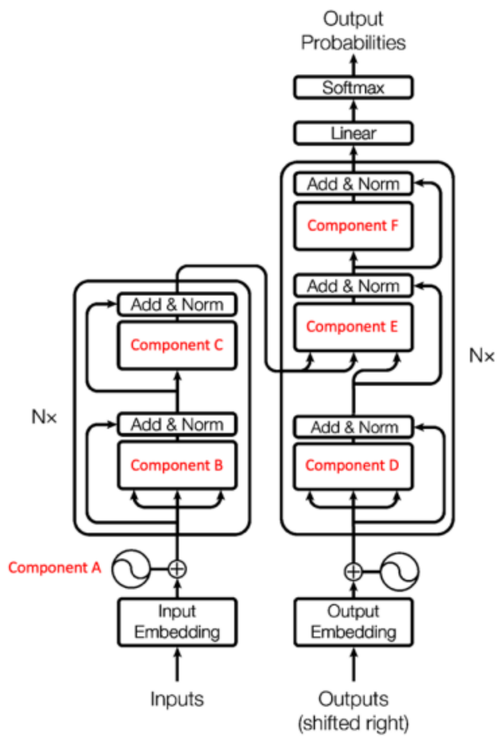


Figure Q1

Answer the following questions:

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

(3 marks)

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

(1 mark)

c) What are the inputs of Component E?

(1 mark)

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

(3 marks)

[END]