# COMP4211 Programming Assignment 2

**Hangyul Son**
**20537267**
**hson@connect.ust.hk**

**Q1.**
Smoothness: The ReLU function is not differentiable at x=0, which can cause problems when computing gradients during backpropagation. The ELU function, on the other hand, is differentiable everywhere, which makes it a smoother function and can help with gradient computations.

Saturation: The ReLU function can suffer from the "dying ReLU" problem, where neurons can become permanently inactive if their output value is consistently zero or negative. The ELU function can help alleviate this problem by allowing negative values and saturating less when the input is negative.

Computation: The ELU function is slightly more computationally expensive than the ReLU function due to its use of exponential functions.

Source: ChatGPT

**Q2.**
"valid" means no padding. "same" results in padding with zeros evenly to the left/right or up/down of the input. When padding="same" and strides=1, the output has the same size as the input.
Source: Tensorflow official website

Given that padding = ’same’, strides = 1, kernel size = 5, filters = Nfilters, in order to keep the output the same size as the input, B*H*W*C, the padding would be applied to make the size of, B*(H+4)*(W+4)*C. As the official documentation says, the same amount of padding would be evenly applied to the input's left, right, up, and down, in this case, 2 on each side. After the convolutional operation, the output would be the same size as the input, B*H*W*C.

**Q3.**
The default padding parameter in DownRightMovedConv2d is ‘valid,’ which means no padding. The padding is applied manually with the tf.pad() function. The padding amount depends on the size of the kernel. For example, if the size of the kernel is [2,2], then the padding of size 1 is applied to left and above of the input so to ensure that the output tensor only depends on the previous pixels. In the example, what we call ‘previous pixels’ would be the manually applied padding of zeros. After the first input has convolved to produce the first output, the steps above would repeat with the output that has been produced and the extra paddings.

**Q4.**
$(2*2*2Cin + 1) * N^{filter.} + (2*2*Cin + 1) * N^{filter}$

**Q5.**
A residual connection is a technique used in deep neural networks where the input to a layer is added to the output of that same layer or a previous layer. Mathematically, a residual connection can be represented as follows:

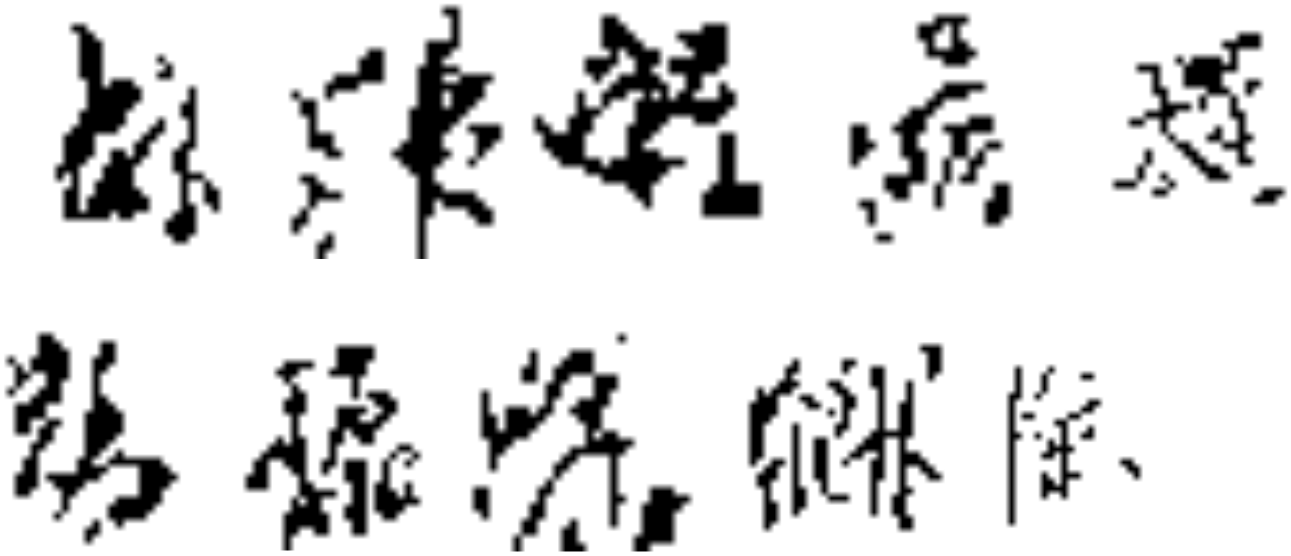
y = F(x) + x

where y is the output of the layer, F(x) is the transformation applied to the input x, and the addition of x is the residual connection.

Residual connection helps tackle the vanishing gradients problem. In deep networks, gradients can become very small as they propagate through multiple layers, which can cause the network to converge very slowly or not at all. By using residual connections, the gradients can propagate directly from the output to the input of the layer, bypassing some of the layers in between. This can help to ensure that the gradients remain sufficiently large and enable the network to learn more effectively.

Source: ChatGPT

**Q6.**



**Q7.**
**Before pretrained model**

```
329/329 [==============================] - 46s 131ms/step - loss: 0.6916
```

**After pretrained model and model.fit()**

```
329/329 [==============================] - 43s 132ms/step - loss: 0.1848
```

**Q8.**
We should set the from_logits parameter as True if we want result of model.predict function to be interpreted as a logit value. We set it to false if we want the it to be interpreted as a probability. In our case we want the predicted values to be in the range of 0 to 1, and therefore, from_logits = False.