2. Neural Network Theory

1. Convolutional Layer:
   Input Image Size: 256×256
2. Convolutional Layer:
   Number of feature maps: 32
   Filter size: 3×3
   Stride: 1
3. Pooling Layer:
   Stride of the pooling layer: 2
   Pooling groups of size: 3×3

For the receptive field, we can work backward from the pooling layer to the convolutional layer. The pooling layer with a stride of 2 reduces the size of each dimension by a factor of 2.

Let's consider a single unit in the pooling layer. To find the region of the input image that influences the activation of that unit:

Each unit in the pooling layer is influenced by the region in the previous layer that corresponds to the pooling groups in the pooling layer. The pooling layer's stride of 2 reduces the size by half in each dimension.

So, for a single unit in the pooling layer:

- It covers a region from the previous layer's feature maps that are affected by a 3×3 pooling operation.
- Each unit in the pooling layer, after a stride of 2, corresponds to a 3×3 region in the previous layer (convolutional layer) which, in turn, is influenced by a 3×3 region from the input image.
- Thus, the receptive field for a single unit in the pooling layer, considering the convolutional layer and the pooling operation, is a 3×3 region in the input image.

3. Gradient Descent Theory

For the given target function $Y_d = w_0 + w_1x_1 + \cdots + w_nw_n$, we define the cost function $J(w)$ for a set of training examples as $J(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(Y_d^{(i)} - Y^{(i)}\right)^2$ which is also define as the MSE

Where,

m is the number of training examples.
$Y_d^{(i)}$ is the predicted output of the i^{th} training example using the target function
$Y^{(i)}$ is the actual output of the i^{th} training example

The weights are updated iteratively using the gradient descent update rule, which is derived from the gradient of the cost function with respect to the weights.

The update rule for each weight in the gradient descent algorithm is:

$$w_j := w_j - \alpha\frac{\partial J(w)}{\partial_{w_j}}$$

Where α is the learning rate.

Thus, we can calculate $\frac{\partial J(w)}{\partial_{w_j}}$ using calculus. Therefore,

$$w_j := w_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(Y_d^{(i)} - Y^{(i)})\,x_j^{(i)}$$