Course 4 - Week 1

1. Why is feed forward NN a bad idea for computer vision tasks?
    a. For an n x n image, we need n x n parameters for the first layer, and it is easy to have millions of parameters in a feed forward NN. It's hard to find sufficient training set to prevent **overfitting**.
    b. Memory cost is expensive
2. Why do we use CNN for computer vision tasks?
    a. Parameter sharing: A filter which is useful for one part of the image might also be useful for another part of the image
    b. Sparse connection: each output only depends a *portion* of inputs in the previous feature map
3. Why do we use padding in a CNN?
    a. A sequence of convolution operations will dramatically reduce the size of the feature maps and much information is lost
    b. For pixels in the middle of a feature map, they will be included in the convolution operation many times while for pixels at the corners, they will only be included in the convolution operation once. In this way, we are losing information at the corners.
4. Given previous feature map size n x n, filter size f, padding p, stride s, if we use same padding *for a convolution operation*, how do we choose filter size f?
    a. First output size = floor((n + 2p - f)/s + 1); here we can think output size as the number of moves of a filter in either horizontal or vertical direction.
    b. More on floor: we only compute convolution when the filter is **inside** the feature map
    c. now we want floor((n + 2p - f)/s + 1) = n and solve for n
5. Given previous feature map size n x n, filter size f, padding p, stride s, if we use same padding *for a pooling operation*, how do we choose filter size f?
    a. Same as last question. Like convolution, every pooling operation also produces only a scalar value.