

Github Link: <https://github.com/DanielKim512/Intro2DL.git>

1. a) You would mainly need 4 hyperparameters: K = Number of filters = 3, F = Spatial extent = 5, S = Stride = 1, and P = amount of zero padding = 0 (Valid) to produce $W_2 \times H_2 \times D_2 = 508 \times 508 \times 3$, where $W_2 = (W_1 - F + 2P) / S + 1 = 508$, $H_2 = (H_1 - F + 2P) / S + 1 = 508$, and $D_2 = K = 3$. Parameter sharing introduces $F \times F \times D_1$. Can compute $(F \times F \times D_1)$ weight and (K) bias here.
b) Fully connected layer would have more parameters due to all neurons being connected to every other neuron. To begin, the parameters needed would be W = Number of weights connected to a Conv layer = $O^2 \times K \times F = 508^2 \times 3 \times 4096$ (AlexNet) = 3171090432, B = Number of biases connected to a Conv layer = 4096, O = output size of previous Conv layer, K = Number of filters from previous Conv layer and F = Number of Neurons = 4096. So, the total number of parameters is $W + B = 3171094528$.
2. When starting out a convolutional neural network, we attempt to match the features to every possible position. The match is the filter which we are provided with being a downward and upward diagonal streak. We would use convolution math to calculate the match of a feature to a certain part of the image. Convolution would help us identify whether there exists a strong match or no match. This would be done repeatedly for each of the filters resulting in layers. We would then want to use pooling which would be implementing the argmax, from a small window across an image. This is to preserve the best fits of each feature within the window and help with computational load. After ReLU, we apply fully connected layers acting as a multilayered perceptron that applies linear transformation.
3. Translation invariance can be gained using pooling layers. Because max pooling will ignore everything except the highest activation input, if the high activation pixel remains inside the input feature map, the result of maxing over the entire feature map would still be the same high probability prediction. This demonstrates how convolution is translation invariant.
4. Paper 2: The ImageNet LSVRC-2010 contest gave 1.2 million high resolution images with 1000 different classes. Previously, models were created with much smaller datasets. This article discusses their methods in using techniques like dropout to reduce overfitting and applying larger datasets by implementing multiple GPUs to achieve the lowest error rate. Eventually, they entered the ILSVRC-2012 competition with a similar model to the paper and reduced the 2nd place's error rate in the competition by nearly half. Due to the random nature of the training process in convolutional neural networks. Because the neural net is trained on stochastic gradient descent, batch size parameter should be thoughtfully picked. In addition, overfitting may be a cause due to dropout method being applied to only the first two fc layers. Suggestions to potentially address the problem are with more attention to data augmentation, decreasing learning rate, and more epochs.