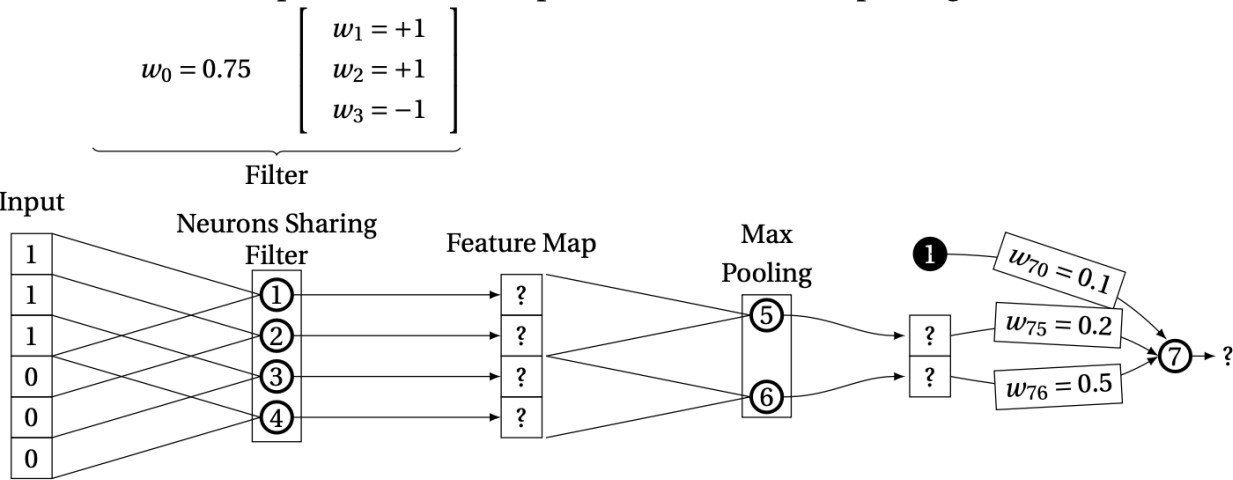
Assignment 4 - Extra Credit

1. The figure below illustrates a layer of a convolutional neural network that is processing a one-dimensional input. For ease of reference each of the neurons in the network has been labeled: 1, 2, 3, 4, 5, 6, 7. The architecture of the network consists of ReLUs that share a filter (Neurons 1, 2, 3, 4), followed by a sub-sampling layer containing two max pooling units (Neurons 5, 6), and then a fully connected layer containing a single ReLU (Neuron 7). The ReLU in the first layer has a 3-by-1 receptive field, and there is a stride of 1 used in this layer. The max pooling units have a receptive field of 2-by-1, and there is no overlap between the receptive fields of the max pooling units.



What value will this network output [show your work — provide the weighted sum (z) and activation (a) for each neuron]?

Z1 = .75 +(1\*1)+(1\*1) +(-1\*1) = 1.75

Z2 = .75 + (1\*2)+(1\*1) + (-1\*1) = 2.75

Z3 = .75 + (1\*2)+(1\*1)+(-1\*2) = 1.75

Z4 = .75 + (1\*1) + (1\*1) + (-1\*2) = .75

A1 = Max(0,1.75) = 1.75

A2 = Max(0,2.75) = 2.75

A3 = Max(0,1.75) = 1.75

A4 = Max(0,.75) = .75

A5 = Max(A1,A2) = 2.75

A6 = Max(A3, A4) = 1.75

Z7 = .1 + (.2\*2.75) + (.5\*1.75) = 1.525

A7 = Max(0,1.525)

1. How does the length of the sequence affect RNN performance? Specifically, what challenges do RNNs face as sequence lengths increase in size?

The length overall causes it to run longer than as it goes through the sequences, some challenges that they face include the vanishing gradient problem when it gets backpropagated. It loses a lot of information as time goes on because of the vanishing gradient. The exploding gradient is another issue that could occur if the weights are updated too much and make the whole model unstable. Some other things could include actual hardware issues running out of memory if the sequence is too large or not perfectly capturing older info. Overfitting for large data and long sequences could also occur.