Suggestions - Problem Set 6

1. The stated goal in this problem is to form a standard feed-forward neural network whose input layer has the 101 sampled values of the polynomial in [0,1], in the form $P(i/100)$ for $i = 0, 1, \ldots, 100$. The final (output) layer should ideally have 4 neurons with activations $a_1, 2a_2, 3a_3, 4a_4$. You may imagine several (shallow or deep) networks that might accomplish this. Any network is acceptable. Additionally, you might consider whether your network inrorporates the concept of derivative in a natural way. For example, you might consider a network whose first layer has the indicated samples $P(i/100)$ of the polynomial itself, while the second layer has samples of the derivative $P'(i/100)$. How would you accomplish this? Can you iterate this process to get the second derivative into the third layer? Once this is done, how can you extract a_1 from the network? It may be that you need to 'carry' this value forward from the first neuron in the second layer (why?). Do you need a separate neuron in each layer to carry this value forward?

3. (b) First, the operation that replaces each pixel activation x_i by that of the pixel j just to the left of i should initially be represented by a 3×3 matrix like W above. What is the schematic form of the matrix W corresponding to this operation? Now we want a $p \times p$ matrix (here p is the *total* number of pixels) A which maps our *linear* feature vector representation $\boldsymbol{x} = (x_1, \ldots, x_p)$ of the image. In this representation the vector $A\boldsymbol{x}$ replaces every activation x_i by the activation x_j corresponding to the pixel j that appears directly to the left of i in the *original* two dimensional array. Notice that i and j may not be simply related in this linear representation of the image. We assume we have specified the mapping M_1 which takes i to $j = M_1(i)$. Specify the form of A_1 in terms of this mapping. How can we then specify A_2, A_3, A_4?

(d) Here, you may need define some additional matrices to represent diagonal shifts. How can you write the full filter matrix A_W in terms of these?

4. (a) Why would vertical edges tend to be detected by the above matrix W?
(b) Note that (normalizing the above matrix W) we have

$$\frac{1}{4}W = \begin{bmatrix} 1/4 \\ 1/2 \\ 1/4 \end{bmatrix} [\,-1/2 \quad 0 \quad 1/2\,]. \tag{2}$$

Show that this matrix computes the horizontal derivative at the center pixel, and also at the two pixels directly above and below the center pixel. Why is the weighting [1/4, 1/2, 1/4]^T for these three derivatives reasonable (which of these three is the best approximation to the horizontal derivative at the center)? More specifically, show that the filter $\frac{1}{4}W$ above represents a horizontal derivative at the center pixel with weight $1/2$, a horizontal derivative at the pixel above the center pixel with weight 1/4, and a horizontal derivative at the pixel below the center pixel with weight 1/4. Explain how you can read off this information easily from the outer product form (2) of the matrix W

(or $\frac{1}{4}W$). This easy way of extracting the derivative information is the primary purpose of writing $\frac{1}{4}W$ as the outer product (2).

(c) With regard to the specific entries in W, why might the form in (b) be preferable to the form $W' = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = 6 \cdot \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix}$?

(f) For the case of diagonal derivatives, there may be several choices of specific weights in W that are reasonable. You can identify at least one of these choices and explain what it does.