UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

Νευρο-Ασαφής Υπολογιστική
Χειμερινό Εξάμηνο 2018-2019
Δημήτριος Κατσαρός

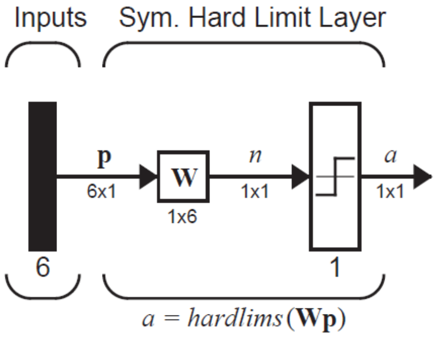
Σειρά προβλημάτων: 3^{η}: **ΟΜΑΔΙΚΕΣ** (2-ΑΤΟΜΩΝ) **ΕΡΓΑΣΙΕΣ**

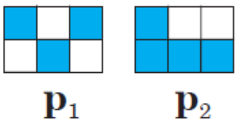
Ημέρα ανακοίνωσης: Monday, January 14, 2019
Προθεσμία παράδοσης: Κυριακή, Φεβρουάριος 03, 2019

SECTION 1: Hebbian and Competitive learning

Problem-01

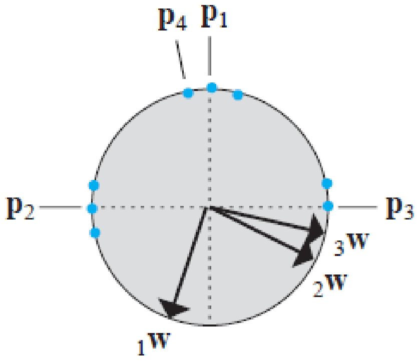
Use the Hebb rule to determine the weight matrix for a perceptron network to recognize the patterns shown to the right.





Problem-02

Consider the input vectors and initial weights shown in the figure below.



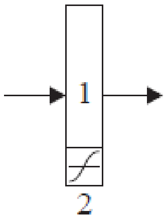
1. Draw the diagram of a competitive network that could classify the data above so that each of the three clusters of vectors would have its own class.
2. Train the network graphically (using the initial weights shown) by presenting the labeled vectors in the following order: p_1, p_2, p_3, p_4. [Recall that the competitive transfer function chooses the neuron with the lowest index to win if more than one neuron has the same.]

SECTION 2: Recurrent neural networks

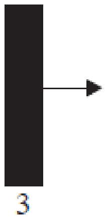
Problem-03

Before stating the problem, let's first introduce some notation that will allow us to efficiently represent dynamic networks:

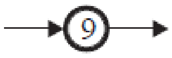
Layer #1 with 2 tansig neurons.



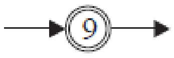
Input vector with 3 elements.



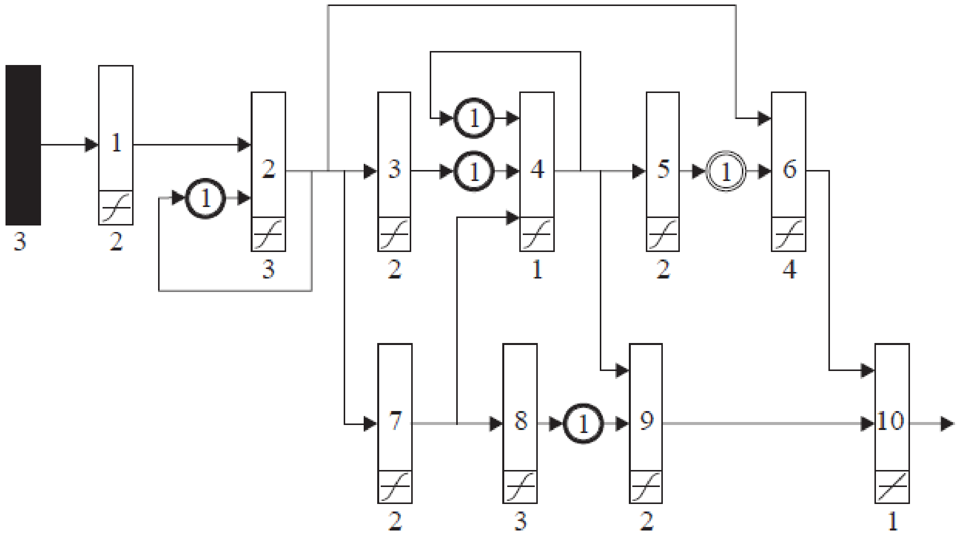
Tapped delay from 1 to 9.



Tapped delay from 0 to 9.



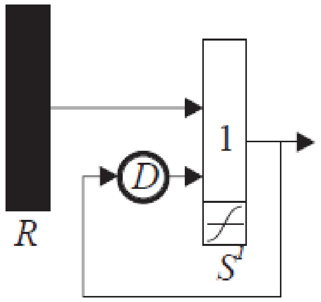
Using this notation, consider the following network:



Define the network architecture, by showing U, X, I_m, $DI_{m,1}$, $DL_{m,l}$, L^f_m, L^b_m, $E^U_{LW}(x)$, $E^U_{LW}(u)$. Also select a simulation order and indicate the dimension of each matrix.

Problem-04

Find the computational complexity for the BPTT algorithm applied to the sample network in figure below as a function of the number of neurons in Layer 1 (S^1), the number of delays in the tapped delay line (D) and the length of the training sequence (Q).



SECTION 3: Convolutional neural networks

Problem-05

Consider a part of a large CNN illustrated in the figure below, which depicts a 3x3 input layer, a 2x2 kernel, and the resulting 2x2 activation map. Let the performance function be denoted as *L*. Write down the equation of $\partial L/\partial W_{ij}$ for each *i,j* as a function of $\partial L/\partial h_{ij}$ (which has been computed in the earlier stage of backpropagation).

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}

W_{11}	W_{12}
W_{21}	W_{22}

h_{11}	h_{12}
h_{21}	h_{22}

Problem-06

Consider a convolutional neural network that is used to classify images into two classes. The structure of the network is as follows:

- INPUT: 100x100 grayscale images.
- LAYER 1: Convolutional layer with 100 5x5 convolutional filters.
- LAYER 2: Convolutional layer with 100 5x5 convolutional filters.
- LAYER 3: A max pooling layer that down-samples Layer 2 by a factor of 4 (from 100x100 → 50x50)
- LAYER 4: Dense layer with 100 units
- LAYER 5: Dense layer with 100 units
- LAYER 6: Single output unit

How many weights does this network have?

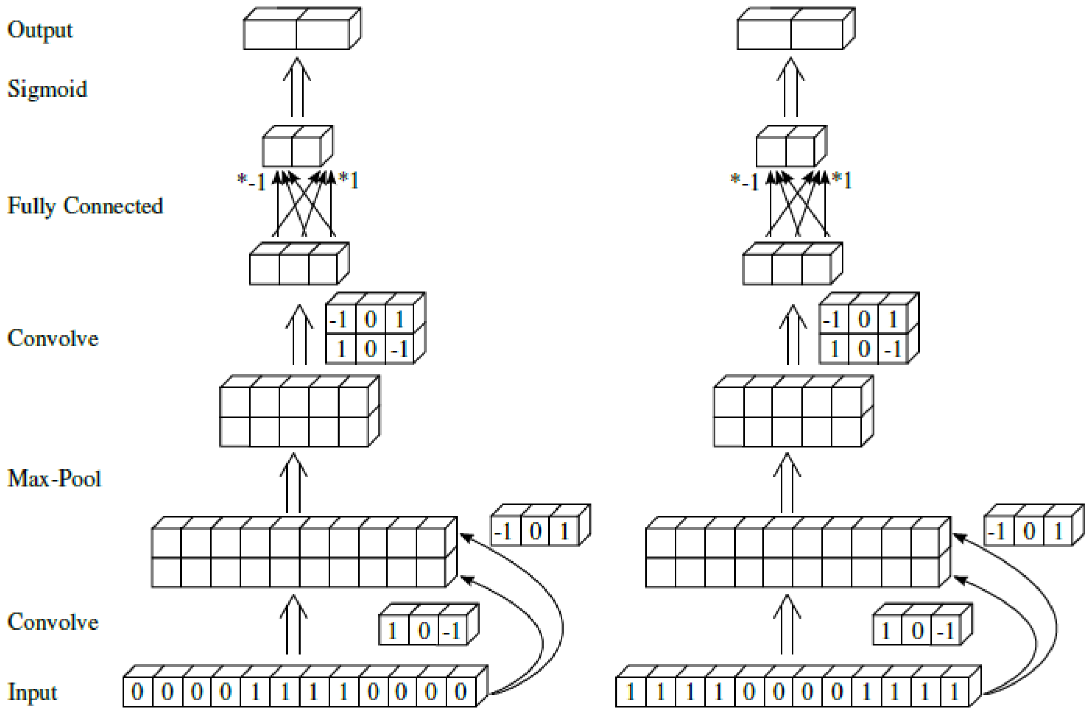
Problem-07

We address a convolutional neural network (CNN) with one-dimensional input. While two-dimensional CNNs can be used for example for grayscale images, one-dimensional CNNs could be used for time-series such as temperature or humidity readings. Concepts for the 1D-case are equivalent to 2D networks. We interpret data in our network as three-dimensional arrays where a row denotes a feature map, a column denotes a single dimension of the observation, and the depth of the array represents different observations. As we will only work with a single input vector, the depth will always be one.

Let the following CNN be given:

- Input I: Matrix of size 1x12x1. We therefore have an input with twelve dimensions consisting of a single feature map.
- First convolutional layer with filters $F^1_0 = (-1,0,1)$ and $F^1_1 = (1,0,-1)$ that generates two output feature maps from a single input feature map. Use *valid* mode for convolutions.
- Max-pooling layer with stride 2 and filter size 2. Note that max-pooling pools each feature map separately.
- Convolutional layer with convolutional kernel $F^2_0 = ((-1,0,1),(1,0,-1))$ of size 2x3x1.
- Fully connected layer that maps all inputs to two outputs. The first output is calculated as the negative sum of all its inputs, and the second layer is calculated as the positive sum of all its inputs.
- Sigmoidal activation function

Calculate the response of the CNN for the two inputs (0,0,0,0,1,1,1,1,0,0,0,0) and (0,0,0,0,1,1,1,1,0,0,0,0).



SECTION 4: Fuzzy logic

Problem-08

Consider the following reference set: {A, B, C, D, E, F, G}, and the fuzzy subsets

$\mathbf{A}$= {(A|0), (B|0.3), (C|0.7), (D|1), (E|0), (F|0.2), (G|0.6)},

$\mathbf{B}$= {(A|0.3), (B|1), (C|0.5), (D|0.8), (E|1), (F|0.5), (G|0.6)},

$\mathbf{C}$= {(A|1), (B|0.5), (C|0.5), (D|0.2), (E|0), (F|0.2), (G|0.9)}.

Calculate the following:

1. $\mathbf{A} \cap \mathbf{B}$
2. $\mathbf{A} \cup \mathbf{B}$
3. $\mathbf{A} \cap \mathbf{B}^c$
4. $(\mathbf{A} \cup \mathbf{B}^c) \cap \mathbf{C}$
5. $(\mathbf{A} \cap \mathbf{B})^c \cup \mathbf{C}^c$
6. $(\mathbf{A} \cap \mathbf{A}^c) \cup \mathbf{A}$

Problem-09

Prove the fuzzy DeMorgan laws:

1. $X \cap Y = (X^c \cup Y^c)^c$
2. $X \cup Y = (X^c \cap Y^c)^c$

Problem-10

Show that $N_\alpha(N_\alpha(x)) = x$ for the generalized negation operator

$$N_\alpha(x) = \frac{1-x}{1+\alpha x} \qquad \alpha>1,\ 0\le x<1.$$

Χρηστικές πληροφορίες:

Η προθεσμία παράδοσης είναι αυστηρή. Είναι δυνατή η παροχή παράτασης (μέχρι 3 ημέρες), αλλά μόνο αφού δώσει ο διδάσκων την έγκρισή του και αυτή η παράταση στοιχίζει 10% ποινή στον τελικό βαθμό της συγκεκριμένης Σειράς Προβλημάτων. Η παράδοση γίνεται με email (στο dkatsar@e-ce.uth.gr) του αρχείου λύσεων σε μορφή pdf (ιδανικά typeset σε LaTeX). Θέμα του μηνύματος πρέπει να είναι το: CE418-Problem set 03: AEM1-AEM2

Ερμηνεία συμβόλων:

Δεν απαιτεί την χρήση υπολογιστή ή/και την ανάπτυξη κώδικα.

Απαιτεί την χρήση του Web για ανεύρεση πληροφοριών ή διεξαγωγή πειράματος.

Απαιτεί την ανάπτυξη κώδικα σε όποια γλώσσα προγραμμαστιμού ή Matlab. Το παραδοτέο θα περιέχει:

- ❖ Την λύση της άσκησης
- ❖ Τον πηγαίο κώδικα υλοποίησης