



Νευρο-Ασφαής Υπολογιστική
Χειμερινό Εξάμηνο 2018-2019
Δημήτριος Κατσαρός

Σειρά προβλημάτων: 2^η: ΟΜΑΔΙΚΕΣ (2-ΑΤΟΜΩΝ) ΕΡΓΑΣΙΕΣ

Ημέρα ανακοίνωσης: Sunday, November 25, 2018
Προθεσμία παράδοσης: Κυριακή, Δεκέμβριος 09, 2018

SECTION 0: Warming-up with steepest descent



Problem-01

Use the MATLAB function `fminunc` to minimize $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ defined as follows:

$$\forall x \in \mathbb{R}^2, f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3).$$

You should write an MATLAB M-file to evaluate both f and ∇f . Specify that you are supplying the gradient ∇f by setting the `GradObj` option to `on` using the `optimset` function. Use the steepest descent algorithm by setting the `LargeScale` option to `off` and the `HessUpdate` option to `steppdesc` using the `optimset` function. Use initial guess $x^{(0)} = [3 \ -5]^T$. Report the number of iterations required.

SECTION 1: Working with ADALINE neural networks



Problem -02

Suppose that we have the following three reference patterns and their targets:

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, t_1 = [26] \right\}, \left\{ \mathbf{p}_2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, t_2 = [26] \right\}, \left\{ \mathbf{p}_3 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, t_3 = [-26] \right\}.$$

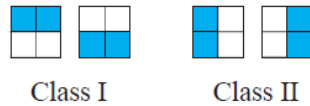
The probability of vector \mathbf{p}_1 is 0.25, the probability of vector \mathbf{p}_2 is 0.25, and the probability of vector \mathbf{p}_3 is 0.5.

- Draw the network diagram for an ADALINE network with no bias that could be trained on these patterns.
- Sketch the contour plot of the mean square error performance index.
- Show the optimal decision boundary (for the weights that minimize mean square error), and verify that it separates the patterns into the appropriate categories.
- Find the maximum stable learning rate for the LMS algorithm. If the target values are changed from 26 and -26 to 2 and -2, how would this change the maximum stable learning rate?
- Perform one iteration of the LMS algorithm, starting with all weights equal to zero, and presenting input vector \mathbf{p}_1 . Use a learning rate of $\alpha=0.5$.



Problem-03

In the following figure two classes of patterns are given.

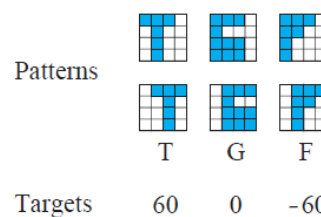


- A. Use the LMS algorithm to train an ADALINE network to distinguish between patterns belonging to class I and class II (we want the network to identify horizontal and vertical lines).
 - B. Can you explain why the ADALINE network might have difficulty with this problem?
- [Try with/without biases, different initial values, and different target values. Record your observations and findings.]



Problem-04

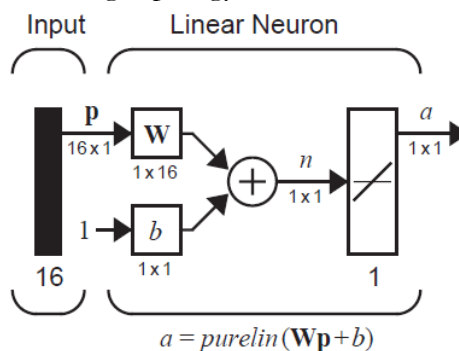
Repeat the work of Widrow and Hoff on a pattern recognition problem from their classic 1960 paper (link-4 on the course's Webpage). They wanted to design a recognition system that would classify the six patterns shown below:



These patterns represent the letters T, G and F, in an original form on the top and in a shifted form on the bottom. The targets for these letters (in their original and shifted forms) are +60, 0 and -60, respectively. The objective is to train a network so that it will classify the six patterns into the appropriate T, G or F groups.

The blue squares in the letters will be assigned the value +1, and the white squares will be assigned the value -1. First we convert each of the letters into a single 16-element vector. We choose to do this by starting at the upper left corner, going down the left column, then going down the second column, etc.

Learning rate $\alpha=0.03$. Present the training patterns in a random sequence. You should use an ADALINE of the following topology:



You are required to draw a plot of the sum square error versus training steps. Each step is defined as the presentation of one input pattern to the neural network.

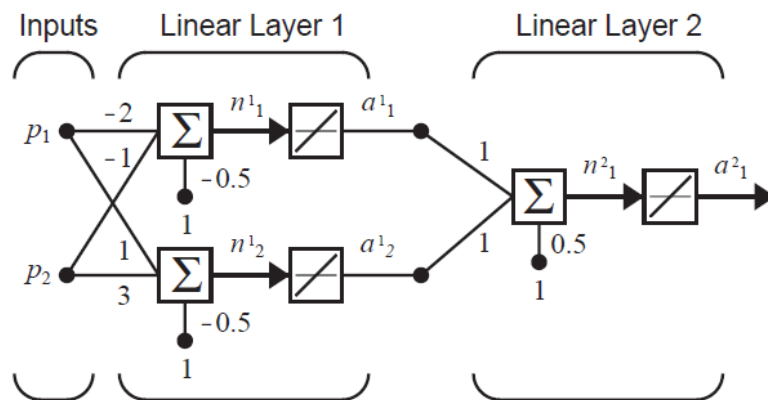
[Your plot should look similar to that in Fig.5 of Widrow-Hoff original paper.]

SECTION 2: Working with multilayer neural networks and standard backpropagation



Problem-05

Find a single-layer network that has the same input/output characteristic as the network shown below.



Problem-06

Write a (MATLAB/python) program to implement the backpropagation algorithm for a 1-S¹-1 network (logsigmoid-linear). Write the program using matrix operations, as we did in the class lecture. Choose the initial weights and biases to be random numbers uniformly distributed between -0.5 and 0.5, and train the network to approximate the function

$$g(p) = 1 + \sin[p(\pi/2)] \text{ for } -2 \leq p \leq 2.$$

Use S¹ = 2 and S² = 10. Experiment with several different values for the learning rate α , and use several different initial conditions. Discuss the convergence properties of the algorithm as the learning rate changes.



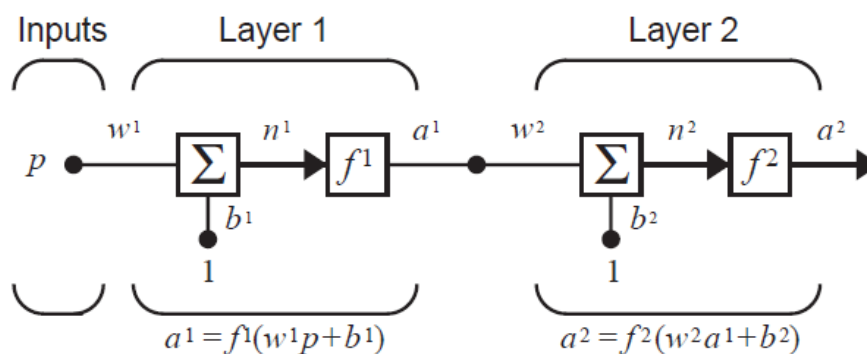
Problem-07

The standard steepest descent backpropagation algorithm, which is summarized in the slide entitled “Summary of backpropagation algorithm” in Lecture-06, was designed to minimize the performance function that was the sum of squares of the network errors, as given in the last equation of slide 17 of Lecture-06. Suppose that we want to change the performance function to the sum of the fourth powers of the errors (e^4) plus the sum of the squares of the weights and biases in the network. Show how the equations in the slide entitled “Summary of backpropagation algorithm” will change for this new performance function. (You don't need to rederive any steps which are already derived in our lectures and do not change.)



Problem-08

For the network shown below



the initial weights and biases are chosen to be

$$w^1(0) = 1, b^1(0) = -2, w^2(0) = 1, b^2(0) = 1.$$

The network transfer functions are:

$$f^1(n) = (n)^2, \quad f^2(n) = 1/n$$

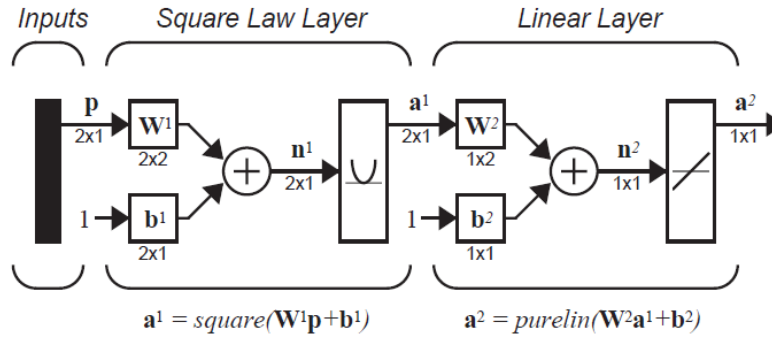
and the input/target pair is given to be $\{p=1, t=1\}$.

Perform one iteration of backpropagation with $\alpha=1$.



Problem-09

Consider the following multilayer perceptron network. (The transfer function of the hidden layer is $f(n) = n^2$.)



The initial weights and biases are:

$$\mathbf{W}^1(0) = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}, \mathbf{W}^2(0) = \begin{bmatrix} 2 & 1 \end{bmatrix}, \mathbf{b}^1(0) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{b}^2(0) = \begin{bmatrix} -1 \end{bmatrix}$$

Perform one iteration of the standard steepest descent backpropagation (use matrix operations) with learning rate $\alpha = 0.5$ for the following input/target pair:

$$\left\{ \mathbf{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, t = \begin{bmatrix} 2 \end{bmatrix} \right\}$$



Problem-10

In the multilayer network, the net input is computed as follows:

$$\mathbf{n}^{m+1} = \mathbf{W}^{m+1} \mathbf{a}^m + \mathbf{b}^{m+1} \text{ or } n_i^{m+1} = \sum_{j=1}^{S^m} w_{i,j}^{m+1} a_j^m + b_i^{m+1}$$

If the net input calculation is changed to the following equation (squared distance calculation):

$$n_i^{m+1} = \sum_{j=1}^{S^m} (w_{i,j}^{m+1} - a_j^m)^2$$

how will the sensitivity backpropagation (i.e., $\mathbf{s}^m = \mathbf{F}'^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}$) change?



Problem-11

Consider again the net input calculation, as described in **Problem-10**. If the net input calculation is changed to the following equation (multiply by the bias, instead of add),

$$n_i^{m+1} = \left(\sum_{j=1}^{S^m} w_{i,j}^{m+1} a_j^m \right) \times b_i^{m+1}$$

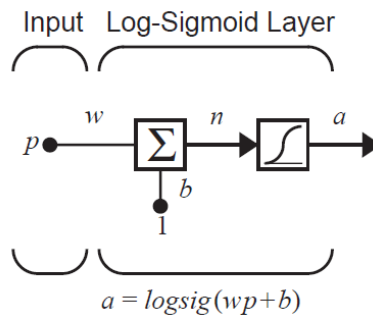
how will the sensitivity backpropagation change?

SECTION 3: Working with multilayer neural networks and variations of backpropagation



Problem-12

We want to train the network shown below on the training set $\{(\mathbf{p}_1 = [-2]), (\mathbf{t}_1 = [0.8])\}, \{(\mathbf{p}_2 = [2]), (\mathbf{t}_2 = [1])\}$, where each pair is equally likely to occur. Demonstrate the effect of batching by computing the direction of the initial step for SDBP with and without batching, starting from the initial guess: $w(0) = 0, b(0) = 0.5$. Do the steps point towards the same direction?



Problem-13

Consider the following quadratic function: $x_1^2 + 2x_2^2$
 Perform 3 iterations of the variable learning rate algorithm, with initial guess: $\mathbf{x}_0 = [0, -1]^T$.
 Use the algorithm parameters: $\alpha=1$, $\gamma=0.2$, $\eta=1.5$, $\rho=0.5$, $\zeta=5\%$.
 (Count an iteration each time the function is evaluated after the initial guess.)



Problem-14

Consider the following quadratic function:

$$F(x) = \frac{1}{2} x^T \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix} x + [4 \quad 4]x.$$

We want to use the steepest descent algorithm with momentum to minimize this function.

- Suppose that the learning rate is $\alpha=0.2$. Find a value for the momentum coefficient γ for which the algorithm will be stable. [Wait for the exercise in the class.]
- Suppose that the learning rate is $\alpha=20$. Find a value for the momentum coefficient γ for which the algorithm will be stable. [Wait for the exercise in the class.]
- Write a MATLAB program to plot the trajectories of the algorithm for the α and γ values of both part (A) and part (B) on the contour plot of $F(x)$, starting from the initial guess: $\mathbf{x}_0 = [-1 \quad -2.5]^T$.



Problem-15

Consider the following quadratic function:

$$F(x) = \frac{1}{2} x^T \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} x + [4 \quad -4]x.$$

We want to use the steepest descent algorithm with momentum to minimize this function.

- Perform two iterations (finding \mathbf{x}_1 and \mathbf{x}_2) of steepest descent with momentum, starting from the initial condition $\mathbf{x}_0 = [0 \ 0]^T$. Use a learning rate of $\alpha=1$ and a momentum coefficient of $\gamma=0.75$.
- Is the algorithm stable with this learning rate and this momentum? [Wait for the exercise in the class.]
- Would the algorithm be stable with this learning rate, if the momentum were zero?

SECTION 4: Working with radial basis neural networks

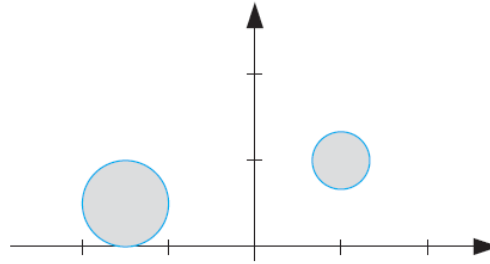


Problem-16

Consider an RBF network with the weights and biases in the first layer fixed. Show how the LMS algorithm of ADALINE could be modified for learning the second layer weights and biases.

**Problem-17**

Design an RBF network to perform the classification illustrated in the following figure. The network should produce a positive output whenever the input vector is in the shaded region and a negative output otherwise.

**Problem-18**

Consider a 1-2-1 RBF network (two neurons in the hidden layer and one output neuron). The first layer weights and biases are fixed as follows:

$$\mathbf{W}^1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \mathbf{b}^1 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

Assume that the bias in the second layer is fixed at 0 ($b^2=0$). The training set has the following input/target pairs: $\{p_1=1, t_1=-1\}$, $\{p_2=0, t_2=0\}$, $\{p_3=-1, t_3=1\}$.

- Use linear least squares to solve for the second layer weights, assuming that the parameter $\rho=0$.
- Plot the contour plot for the sum squared error. Recall that it will be a quadratic function.
- Write a MATLAB/python program to check your answers to parts (A) and (B).
- Repeat tasks (A) to (C) with $\rho=4$. Plot again the squared error.

**Problem-19**

Write a MATLAB/python program to implement the steepest descent algorithm for the 1-S¹-1 RBF network. Train the network to approximate the function:

$$g(p) = 1 + \sin(p\pi/8) \text{ for } -2 \leq p \leq 2.$$

- Select 10 data points at random from the interval $-2 \leq p \leq 2$.
- Initialize all parameters (weights and biases in both layers) as small random numbers, and then train the network to convergence. (Experiment with the learning rate α , to determine a stable value.) Plot the network response for $-2 \leq p \leq 2$, and show the training points on the same plot. Compute the sum squared error over the training set. Use 2, 4 and 8 centers. Try different sets of initial weights.

Χρηστικές πληροφορίες:

Η προθεσμία παράδοσης είναι ανυστερή. Είναι δυνατή η παροχή παράτασης (μέχρι 2 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 02: AEM1-AEM2

Εομηνεία 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 προγραμματισμού ή Matlab. Το παραδοτέο θα περιέχει:

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