UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

Νευρο-Ασαφής Υπολογιστική
Χειμερινό Εξάμηνο 2020-2021
Δημήτριος Κατσαρός

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

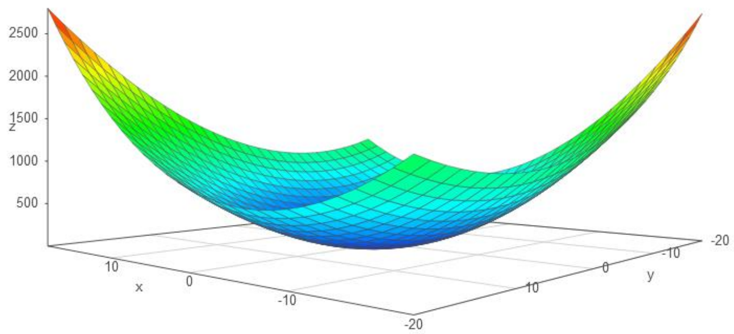
Ημέρα ανακοίνωσης (updated **Prob.07**): Monday, November 16, 2020
Προθεσμία παράδοσης: Κυριακή, Δεκέμβριος 20, 2020

SECTION 1: Working with second-order minimizers: Conjugate Gradient and Newton

Problem-01

Find the minimum of the (quadratic) two-dimensional function:

$$F(w) = w^T \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} w = 3w_1^2 + 2w_2^2 + 2w_1 w_2$$

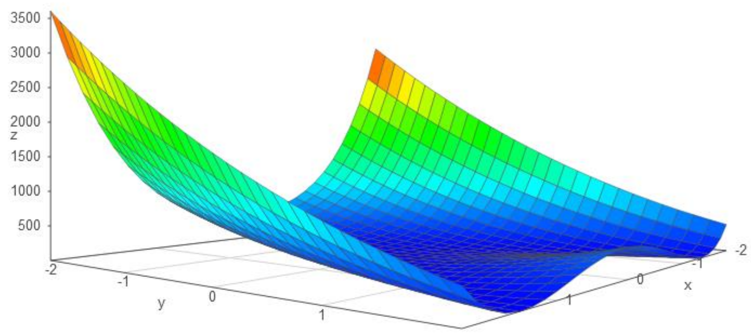


with the Conjugate Gradient (Fletcher-Reeves), with initial guess $w^{(0)}= [1.5\ -0.75]^T$. Then, apply the Gradient Descent method (accuracy: three decimal points) on the same function and unit step movement. For each method show your analytic calculations.

Problem-02

Starting from $x^{(0)}= [-0.5\quad 0.5]^T$, find the minimum of the Rosenbrock function (https://en.wikipedia.org/wiki/Rosenbrock_function):

$$F(w) = 100(w_2 - w_1^2)^2 + (1 - w_1)^2$$

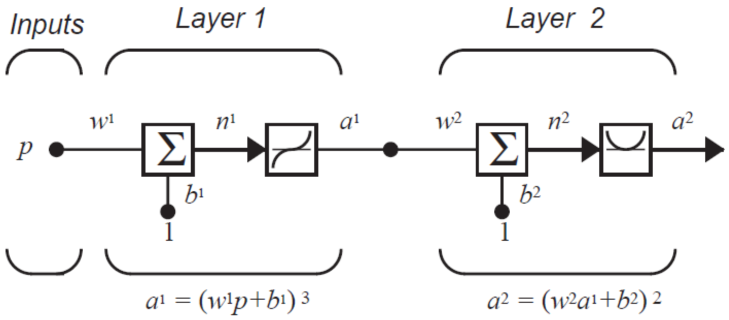


using Newton's method and fixed step $\lambda_k=1$ (even though this is not a quadratic form).

SECTION 2: Working with multilayer perceptrons and standard backpropagation

Problem-03

For the network shown below



the initial weights and biases are chosen to be

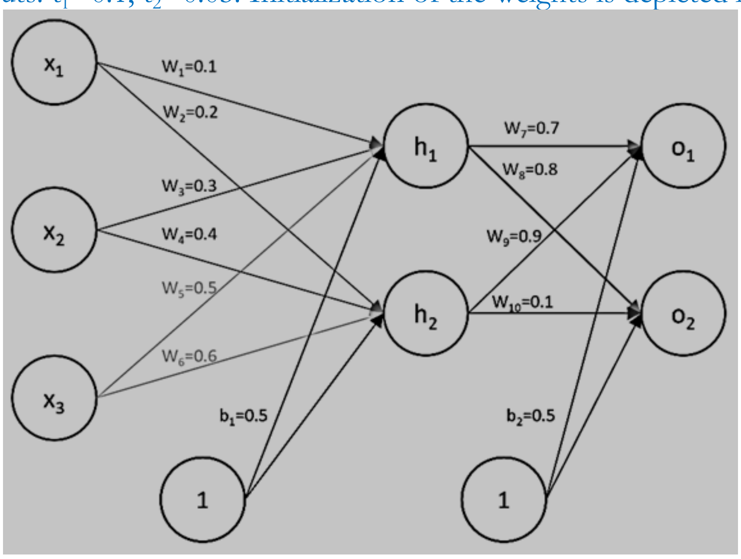
$$w^1(0) = -2,\ b^1(0) = 1,\ \ w^2(0) = 1,\ b^2(0) = -2.$$

An input/target pair is given to be $\{p=1, t=0\}$.

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

Problem-04

Perform one iteration of backpropagation (use matrix operations) for the following neural network. Activation function is the sigmoid function $\sigma(x) = 1/(1+e^{-x})$. Use a learning rate $\alpha=0.01$. We provide the following inputs: $x_1=1$, $x_2=4$, $x_3=5$ and require the following outputs: $t_1=0.1$, $t_2=0.05$. Initialization of the weights is depicted in the figure.



Problem-05

Suppose we have a simple network of the shape (linear => sigmoid => linear => sigmoid=> . . . => linear). Write out the chain rule for computing the derivative of the final outputs of this network with respect to the parameters of the first linear layer. What can you say about the vanishing gradient problem using this expression?

Problem-06

Write a (MATLAB/python/…) program to implement the backpropagation algorithm for a $1\text{-}S^1\text{-}1$ 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:

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

Use S^1= 3 and S^1= 15. 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

In the setting with S^1= 15 and learning rate α=0.15 of the previous exercise, apply the dropout technique (https://jmlr.org/papers/v15/srivastava14a.html) with dropout probability θ of hidden-layer neurons equal to θ=0.1, then with θ=0.25, and then with θ=0.5. Apply the dropout during the training phase only, and only on the hidden layer units (not of course to the input neuron). Discuss the convergence properties of the algorithm, as well as its accuracy and contrast your findings with those of the previous exercise. Perform any additional experiments to figure out the operation of dropout as a generalization technique.

SECTION 3: Working with variations of backpropagation and modern minimizers

Problem-08

Consider the following quadratic function:

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

Perform two iterations of the VLBP algorithm, with initial guess $\mathbf{x}_0$=[-2 -1.5]^T. Use the algorithm parameters: α=0.5, γ=0.1, η=1.5, ϱ=0.5, and ζ=4%. Plot the algorithm trajectory on a contour plot of $F(\mathbf{x})$.

Problem-09

Determine under what conditions on its parameters' values, Adam corresponds to AdaGrad?

SECTION 4: Working with Convolutional neural networks

Problem-10

You are given the following description of a convolutional neural network:

Layer information

#	name	size
0	input	1x28x28
1	conv2d1	32x24x24
2	maxpool1	32x12x12
3	conv2d2	32x10x10
4	maxpool2	32x5x5
5	dense	256
6	output	10

For each layer of this network, calculate the number of:

1. weights per neuron in that layer (including bias)
2. neurons in that layer
3. connections to neurons in that layer
4. independent parameters in that layer

Assume that the input images are gray-scale, there is no padding (i.e., valid padding), and there are 10 valid outputs. The size of each filter in each convolutional layer can be determined by the reduction in the size of the image.

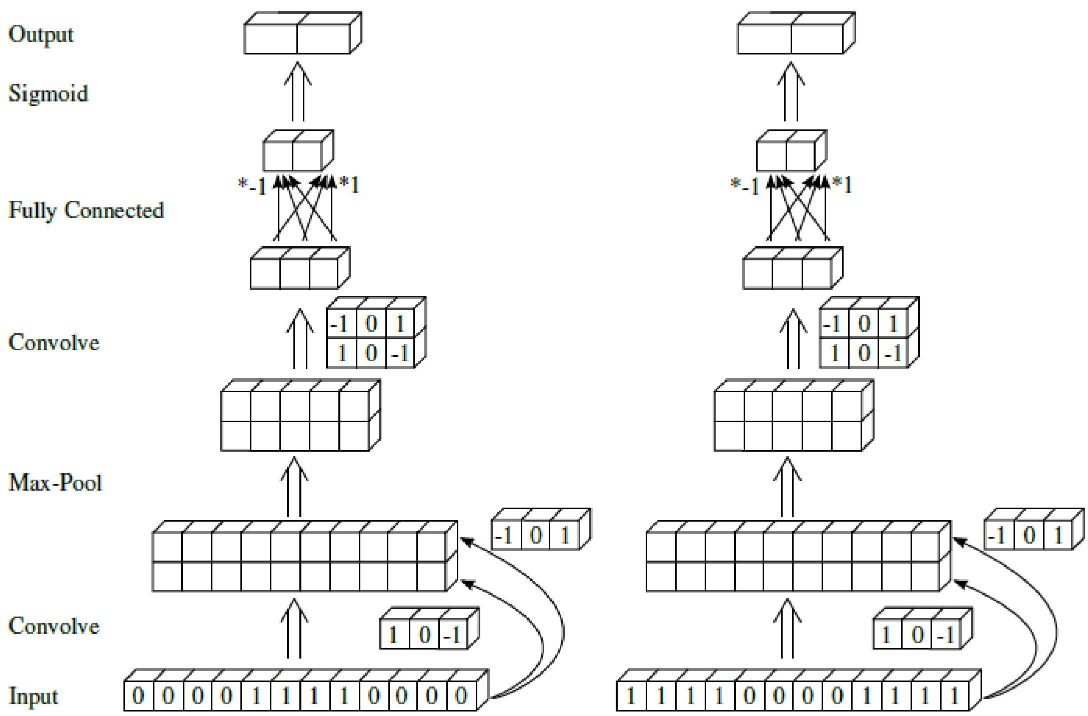
Problem-11

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 (1,1,1,1,0,0,0,0,1,1,1,1).



Problem-12

How can we compute the gradient of convolution layer when convolution stride is greater than 1?

Problem-13

Compute the gradient of max pooling with overlapping regions.

<u>Χρηστικές πληροφορίες</u>:

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, αλλιώς με υψηλής ποιότητας scanning/photo χειρογράφου). Θέμα του μηνύματος πρέπει να είναι το: CE418-Problem set 02: AEM1-AEM2

<u>Ερμηνεία συμβόλων</u>:

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