UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

Νευρο-Ασαφής Υπολογιστική
Χειμερινό Εξάμηνο 2023-2024
Δημήτριος Κατσαρός

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

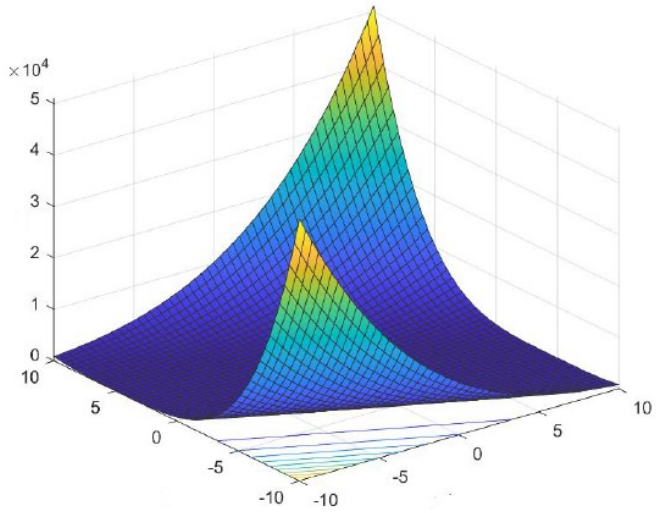
Ημέρα ανακοίνωσης: Thursday, January 18, 2024
Προθεσμία παράδοσης: Κυριακή, Φεβρουάριος 11, 2024

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

Problem-01

Find the minimum of the two-dimensional function:

$$F(\mathbf{w}) = w_1^2 + w_2^2 + (0.5w_1 + w_2)^2 + (0.5w_1 + w_2)^4$$



with the Conjugate Gradient (Fletcher-Reeves), with initial guess w(0)= $[3, 3]^T$. Perform five iterations (if not finding the minimum earlier). Observe and comment on the slow/fast convergence towards the minimum $\mathbf{w}=(0, 0)^T$. Then, apply the Gradient Descent method (accuracy: three decimal points) on the same function and unit step movement. Perform ten iterations (if not finding the minimum earlier). For each method show your analytic calculations.

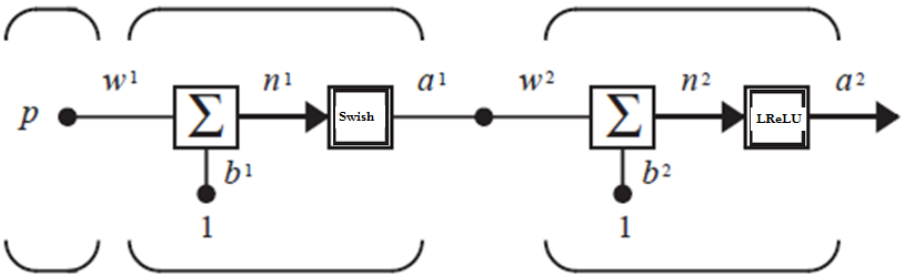
Problem-02

Find the minimum of the previous function using the Newton method and fixed step λ_k=1 (even though this is not a quadratic form).

SECTION 2: Working with multilayer perceptrons and standard backpropagation

Problem-03

For the neural network shown below



the initial weights and biases are chosen to be

$$w^1(0)= -3,\ b^1(0)= 2,\ \ w^2(0)= -1,\ b^2(0)= -1.$$

An input/target pair is given to be {p=1, t=0}, and the parameter of LReLU is equal to 0.001. Perform two iterations of backpropagation with learning rate $\alpha=1$.

Problem-04

Write a (MATLAB/python/Keras/…) program to implement the backpropagation algorithm for a 1-S^1-1 network (logsig-ReLU). 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:

$$g(p)=1+\sin[p(3\pi/8)] \text{ for } -2 \le p \le 2.$$

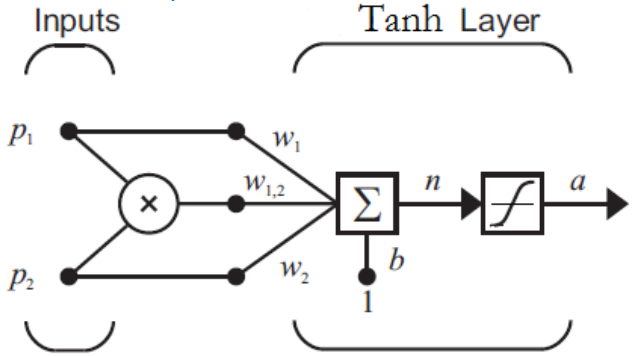
Use $S^1= 2$, $S^1= 8$, and $S^1= 12$. Experiment with several different values for the learning rate α (make sure you experiment with $\alpha=0.1$), and use several different initial conditions. Discuss the convergence properties and the accuracy of the algorithm as the learning rate changes, and as the capacity (in terms of number of hidden neurons) increases.

Problem-05

In the setting with $S^1= 12$ and learning rate $\alpha=0.1$ 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 $\theta=0.15$, then with $\theta=0.25$, and then with $\theta=0.35$. 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.

Problem-06

Consider the network shown in the figure below, where the inputs to the neuron involve both the original inputs and their product.



A. Find a learning rule for the network parameters, using the steepest descent algorithm (as we have done in the class for backpropagation).

B. For the following initial parameter values, inputs and target, perform one iteration of your learning rule with learning rate $a = 1$:

$$w_1= 1,\ w_2= -1,\ w_{1,2}= 0.5,\ b_1 = 1, \text{ and } p_1= 0,\ p_2= 1,\ t= 0.75.$$

Problem-07

Show that an MLP using only ReLU (or pReLU) constructs a continuous piecewise linear function.

SECTION 3: Working with variations of backpropagation and modern optimizers

Problem-08

Consider the following function:

$$F(w) = 0.1w_1^2 + 2w_2^2.$$

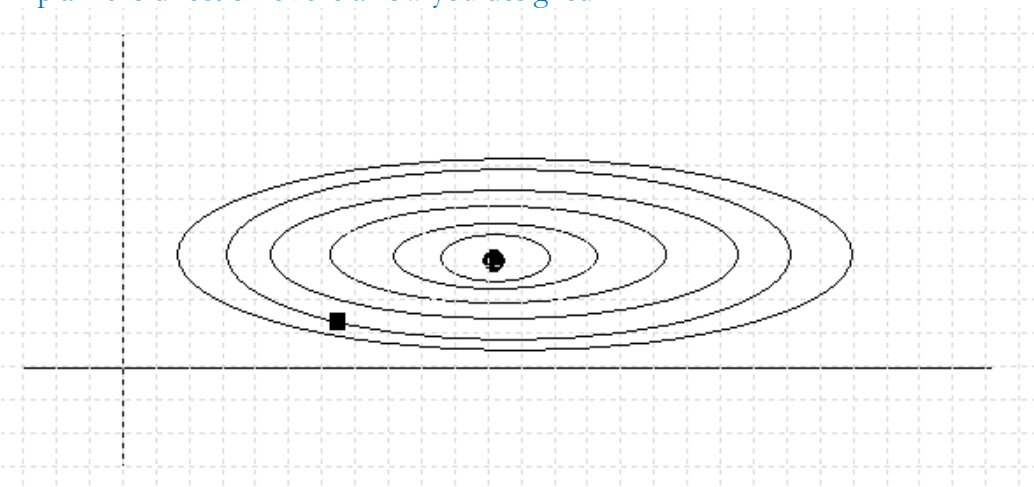
1. Find the minimum of the function implementing the Adadelta optimizer (instead of gradient descent) with learning rate α=0.4. Plot the algorithm's trajectory on a contour plot of $F(\mathbf{x})$.
2. Change the learning rate to α=3, and repeat the same task.
3. Try out Adadelta for the same objective function rotated by 45degrees, i.e., $F(w) = 0.1(w_1 + w_2)^2 + 2(w_1 - w_2)^2$. Does it behave differently?

Problem-09

On the plot below, show **one gradient step** (with an arrow) for each of the methods mentioned below. The minimum is the black circle, whereas your position is the black square. Make sure you label the three arrows in your answer.

- Standard gradient
- Natural gradient (or Newton's method)
- Adagrad or RMSprop (assume they have run for a while to accumulate gradient information)

Explain the direction of the arrow you designed.



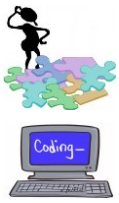
Problem-10

You are given the description of the following optimization method, where a, β, $\nu \in \boldsymbol{R}$, (a is the learning rate):

$$g_{t+1} \leftarrow \beta \cdot g_t + (1 - \beta) \cdot \nabla \hat{L}_t(\theta_t)$$

$$\theta_{t+1} \leftarrow \theta_t - \alpha \left[(1 - \nu) \cdot \nabla \hat{L}_t(\theta_t) + \nu \cdot g_{t+1} \right]$$

For which values of β and ν you get optimization methods familiar to you (described in our lectures)? Name them, and write the respective equations.



Problem-11

Consider the following input image:

$$\mathbf{I} = \begin{pmatrix} 20 & 35 & 35 & 35 & 35 & 20 \\ 29 & 46 & 44 & 42 & 42 & 27 \\ 16 & 25 & 21 & 19 & 19 & 12 \\ 66 & 120 & 116 & 154 & 114 & 62 \\ 74 & 216 & 174 & 252 & 172 & 112 \\ 70 & 210 & 170 & 250 & 170 & 110 \end{pmatrix}$$

A. What is the output provided by a convolution layer with the following properties:
stride=(1, 1), and

$$\text{kernel} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

B. Take the output from A. and apply a max pooling layer with the following properties:
stride=(2,2)
window shape=(2, 2).

C. For the following kernels, describe what kind of feature they extract from the image:

$$\mathbf{F1} = \begin{pmatrix} -10 & -10 & -10 \\ 5 & 5 & 5 \\ -10 & -10 & -10 \end{pmatrix}$$

$$\mathbf{F2} = \begin{pmatrix} 2 & 2 & 2 \\ 2 & -12 & 2 \\ 2 & 2 & 2 \end{pmatrix}$$

$$\mathbf{F3} = \begin{pmatrix} -20 & -10 & 0 & 5 & 10 \\ -10 & 0 & 5 & 10 & 5 \\ 0 & 5 & 10 & 5 & 0 \\ 5 & 10 & 5 & 0 & -10 \\ 10 & 5 & 0 & -10 & -20 \end{pmatrix}$$

Problem-12

Consider a 1D convolutional network where the input has three channels. The first hidden layer is computed using a kernel size of three and has four channels. The second hidden layer is computed using a kernel size of five and has ten channels. How many biases and how many weights are needed for each of these two convolutional layers?

Problem-13

Max-pooling can be accomplished using ReLU operations.

A. Express max(a, b) by using only ReLU operations.

B. Use this to implement max-pooling by means of convolutions and ReLU layers.

C. How many channels and layers do you need for a 2x2 convolution? How many for a 3x3 convolution?

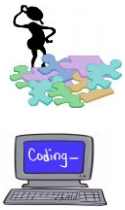
Problem-14

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 lters.
- LAYER 2: Convolutional layer with 100 5x5 convolutional lters.
- 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-15

Your task is to implement fast convolutions with a kxk kernel. One of the algorithm candidates is to scan horizontally across the source, reading a k-wide strip and computing the 1-wide output strip one value at a time. The alternative is to read a k + Δ wide strip and compute a Δ-wide output strip.

A. Why is the latter preferable? Is there a limit to how large you should choose Δ?

B. Create a sample image of 228x228 and a series of 3x3, 7x7 and 11x11 kernels, and measure the execution time of the two alternatives. Comment on your results.

Χρηστικές πληροφορίες:

Η προθεσμία παράδοσης είναι αυστηρή. Είναι δυνατή η παροχή παράτασης (μέχρι 4 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@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

Ερμηνεία συμβόλων:

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

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

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