University of Thessaly

Neuro-Fuzzy Computing

ECE447

2nd Problem Set

Alexandra Gianni
ID: 3382

Nikos Stylianou
ID: 2917

January 19, 2024

Problem 1

In this exercise we need to find the minimum of the given 2-dimensional function:

$$F(\mathbf{w}) = w_1^2 + w_2^2 + (0.5w_1 + w_2)^2 + (0.5w_1 + w_2)^4 \tag{1}$$

with the Conjugate Gradient (Fletcher-Reeves) method.

Initially, we can conclude that the function $F(w)$ is not in quadratic form because of the term $(0.5w_1+w_2)^4$. A function is said to be in quadratic form if it can be expressed as a second-degree polynomial where all the terms are either squared terms or cross-products of the variables. The presence of the fourth-degree term $(0.5w_1 + w_2)^4$. makes this function a higher-degree polynomial, specifically a quartic function with respect to $(0.5w_1 + w_2)$, which means it cannot be classified as quadratic.
Also, the independent values in this function are w_1, w_2, because only with them we can manipulate the $F(w)$.

As an initial guess we have $w\left(0\right) = \left[3, 3\right]^T$.
The steps we have to use are specific for each iteration

FIRST ITERATION k = 0

Step1: Calculate the Gradient at $w\left(k\right)$

$$\nabla f(w_1, w_2) = \begin{pmatrix} \dfrac{\partial f}{\partial w_1} \\ \dfrac{\partial f}{\partial w_2} \end{pmatrix} = \begin{pmatrix} 2w_1 + (0.5w_1 + w_2) + 2(0.5w_1 + w_2)^3 \\ 2w_2 + 2(0.5w_1 + w_2) + 4(0.5w_1 + w_2)^3 \end{pmatrix} = \begin{pmatrix} 2.5w_1 + w_2 + 2(0.5w_1 + w_2)^3 \\ w_1 + 4w_2 + 4(0.5w_1 + w_2)^3 \end{pmatrix}$$

where at the point $w\left(0\right) = \left[3, 3\right]^T$ we have $\nabla f(x) = \begin{pmatrix} -53 \\ -19 \end{pmatrix}$

Problem 3

For the given neural network, we have:

- learning rate $LR = 1$,
- $w^1\left(0\right) = -3,\ w^2\left(0\right) = -1$,
- $b^1\left(0\right) = 2,\ b^2\left(0\right) = -1$ and
- input/target pair $\{p = 1,\ t = 0\}$

FIRST ITERATION

Step 1: Calculate first layer's output

$$n^1 = w^1 p + b^1 = (-3)(1) + 2 = -1$$

$$a^1 = Swish\left(n^1\right) = Swish\left(-1\right) = \frac{n^1}{1 + e^{-n^1}} = \frac{-1}{1 + e} = -0.2689$$

<u>Step 2: Calculate second layer's output</u>

$$n^2 = w^2 a^1 + b^2 = (-1)(-0.2689) + (-1) = -0.7311$$
$$a^2 = LReLU\left(n^2\right) = LReLU\left(-0.7311\right) = -0.000731$$

<u>Step 3: Calculate error</u>

$$e = t - a^2 = (1 - (-0.000731)) = 1.000731$$

<u>Step 4: Calculate sensitivity on second layer</u>

$$s^2 = -2\ LReLU^{'}\left(n^2\right)\left(t - a^2\right) = -2\,(0.001)\,(1.000731) = -0.002001$$

LReLU's derivative is 1 for $x > 0$ and 0.001 for $x < 0$.

<u>Step 5: Calculate sensitivity on first layer using back-propagation</u>

$$s^1 = Swish^{'}\left(n^1\right)\left(W^2\right)^T s^2 = Swish^{'}\left(-1\right)\left(-1\right)\left(-0.002001\right) = 0.0723 \cdot (-1) \cdot (-0.002001)$$
$$s^1 = 0.000145$$

<u>Step 6: Update wheights and biases</u>