Introd. to Neural Networks

Dr. S. Recksiegel — WS 2022/23

Exercise Sheet 4 — L. Rickler, E. Eberhard, J. Mayer-Steudte, L. Wicke

November 9, 2022 — Moodle: https://www.moodle.tum.de/course/view.php?id=79979

---

1. Root finding / Nullstellensuche

We want to write a program to find the roots of

$$x^2 - 7x + 10$$

using the Newton-Raphson method. To find out where to start looking for a root, you would usually have to either plot the functions or implement e.g. the bisection method to find an appropriate starting point, but here you should be able to do without this.

Which precision (in terms of ϵ_m) can you expect? Make sure that your program terminates before reaching the point where additional steps worsen the result.

What happens when you try to find the root of (the standard branch of) the arc tan function using $x_0 = 2$ as the initial guess? Why?

2. χ^2 fit of a non-linear function (Breit-Wigner)

In this exercise we are going to fit a non-linear function. The dataset

```
double datax[9] = {0,25,50,75,100,125,150,175,200};
double datay[9] = {10.6,16.0,45.0,83.5,52.8,19.9,10.8,8.25,4.7};
double datasigma[9] ={2.3, 3.5, 4.5, 6.4, 4.4, 3.4, 2.1, 1.6, 1.1};
```

can be copied and pasted from the moodle.

To fit a theory function $y(x, \vec{a})$ depending on model parameters $\vec{a} = a_1, \ldots, a_m$ to data points (x_i, y_i), $i = 1, \ldots, n$ with errors σ_i, we have to minimise

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{y_i - y(x_i, \vec{a})}{\sigma_i} \right)^2$$

Let $a_1 = f_r$, $a_2 = E_R$, $a_3 = \Gamma^4/4$ and $x = E$. With these parameters the Breit-Wigner resonance formula reads

$$y(x) = \frac{a_1}{(a_2 - x)^2 + a_3} \,. \tag{1}$$

Finding the minimum of χ^2 ($\partial\chi^2/\partial a_i = 0$ for all i) comes down to multidimensional root-finding, $f_i = 0$ with $i = 1, 2, 3$.

- Newton-Raphson in principle works but only converges close to the minimum. Implement Newton-Raphson, using

  $$\beta_k = -\frac{1}{2}\frac{\partial\chi^2}{\partial a_k} \quad \text{and} \quad \alpha_{kl} = \frac{1}{2}\frac{\partial^2\chi^2}{\partial a_k \partial a_l}$$

  (the factors of 1/2 are purely conventional). This way, you have to calculate

  $$\delta\vec{a} = (\alpha_{kl})^{-1} \cdot \beta$$

  for each Newton-Raphson step. As starting values, use $\vec{a} = (60000, 70, 55)$ and find the $\vec{a}$ for which β vanishes, i.e. the $\vec{a}$ for which χ^2 is minimal.

- Try Levenberg-Marquardt for less well-informed starting values. Instead of the NR-step from the previous section, use

  $$\delta\vec{a} = (\alpha_{kl} + \lambda \cdot \mathbb{1})^{-1} \cdot \beta$$

  (Levenberg, converges, but often rather slowly) or

  $$\delta\vec{a} = (\alpha_{kl} + \lambda\, \alpha_{kk})^{-1} \cdot \beta$$

  (Levenberg-Marquardt, faster convergence). The parameter λ is initialised to 1 and increased (algorithm behaves more like gradient descent) if $\delta\vec{a}$ would make the fit worse or decreased (algorithm behaves more like NR) if $\delta\vec{a}$ makes the fit better. Only in the latter case is $\delta\vec{a}$ added to $\vec{a}$.

You can look at the Mathematica Notebook provided in the Moodle (or the chapter from Numerical Recipies) for guidance.