Institut for Theoretical Physics
University Leipzig

Prof. Dr. B. Rosenow
A. Afanah , M. Staats

Statistical Mechanics of Deep Learning - Problem set 1

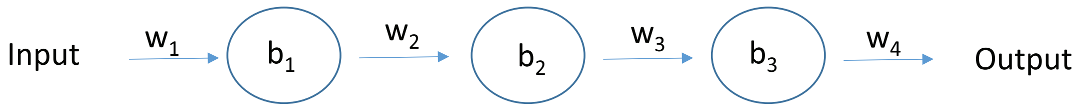
Winter Term 2023/24

Hand in: Thursday, 19.10 at 11:59 pm, you can upload your solutions to the course webpage on Moodle platform.

1. The vanishing gradient problem

2+1+4+3 Points

Consider a simple neural network with one unit per hidden layer as depicted in the figure below. The output of the network is denoted by $f_w(x) = w_4 g(z_3)$ with the sigmoidal activation function $g(z) = \dfrac{1}{1+e^{-z}}$. The preactivations z_i are computed recursively in the following way: $z_1 = w_1 x + b_1$, and $z_i = w_i g(z_{i-1}) + b_i$. The deviation between the desired output $a(x)$ and the actual network output is quantified by the quadratic cost function $C = \frac{1}{2}[f_w(x) - a(x)]^2$.



Deep neural networks are known to suffer from the so-called vanishing gradient problem. The goal of this exercise is to invistigate the cause of this problem, and then to explore suitable restrictions to be imposed on the range of weights and activations to avoid this problem.

(a) Find the derivative $\dfrac{\partial C}{\partial b_1}$, and comment on the value of the derivative in the case $g'(z) < 1$.

(b) Consider the product $|wg'(wz+b)|$ and assume that $|wg'(wz+b)| \geq 1$. Argue that this can only ever occur if $|w| \geq 4$.

(c) Supposing that $|w| \geq 4$, consider the set of preactivations $\{z\}$ for which $|wg'(wz+b)| \geq 1$. Show that the set of $\{z\}$ satisfying that constraint can range over an interval no greater in width than

$$\frac{2}{|w|} \ln\left(\frac{|w|(1+\sqrt{1-4/|w|})}{2} - 1\right)$$

(d) Show numerically that the above expression bounding the width of the range is greatest at $|w| \approx 6.9$, where it takes a value ≈ 0.45. So given that everything lines up just perfectly, we still have a fairly narrow range of input activations which can avoid the vanishing gradient problem.

2. Learning Python *8 Points*

The goal of this exercise is to make you familiar with Python and some of its libarys.

(a) Write a Python programm that generates random integer numbers in the range of [1, 200] until the number 100 is randomly drawn or more than 200 numbers in total where drawn. The generated numbers should be stored in a list.

(b) Make a numpy array out of the list and sort the array in ascending order using numpy functions.

(c) Loop through all elements of the array. If the element is odd substract 1, if it is even add 1.

(d) Create an numpy array of even length using numpy.linspace and plot the generated data with plt.plot(data_here). The parameters of numpy linspace are your choice.