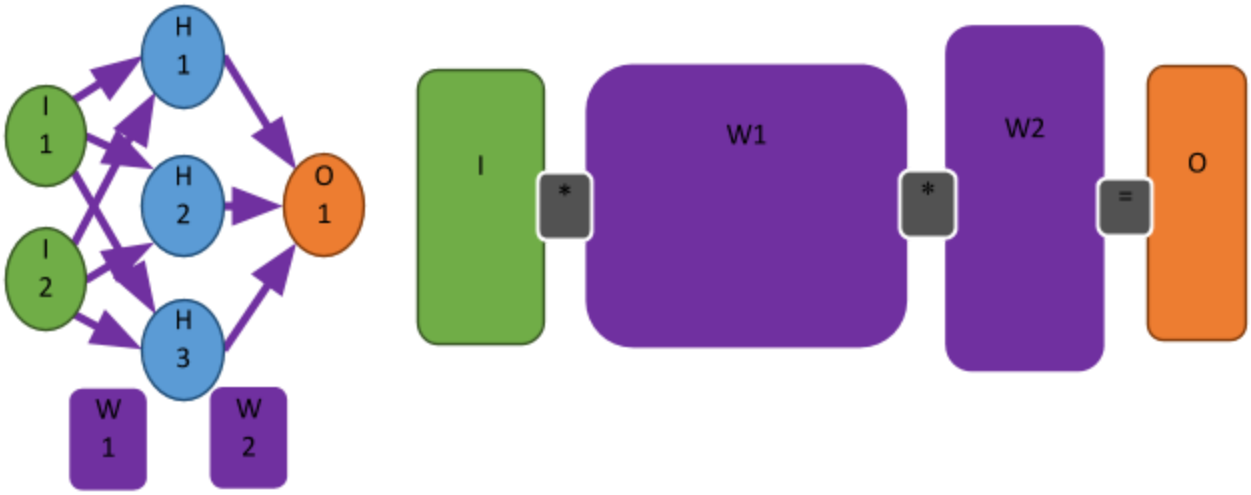
Welcome to the first Mooqita challenge for the Advanced Ruby on Rails course. Please keep in mind it is still optional and won't count to your credentials in this Ruby class. The usual challenges are taken from real world tasks and interview questions. For this specific challenge series we first need to build up some basic skills and knowledge. The challenges after this will lead to more realistic and practical exercises.

This challenge is also a little more complex than the previous ones and will contain a kind of tutorial as well. We will give you an introduction into one of the current industry hot topics, namely machine learning using neural networks. In this challenge you will build your very own implementation of a neural network. To simplify the task we will provide basic source code to start from and a data set to test your implementation. Our simple example implementation of the neural network achieved an average accuracy on this set of M = 0.97 (SD = 0.01) with 450 training iterations (more on that later). Can you beat it? There are some really good tutorials out there on neural networks, e.g. https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/. We recommend building an easy one such as a feed forward network. If you are not familiar with machine learning and neural networks just start the challenge. We will explain the basic concepts and ideas throughout this challenge.

Imagine you want to predict if the next homework will be hard or easy. Two parameters are important: teacher mood (1 grumpy to 5 happy) and class behavior (1 mean to 5 lovely). You have some samples from previous homework assignments from the same teacher. You can use a neural network to predict the next homework based on this data, similar to the sketch below. We use two nodes in the input layer (green) one for each parameter (teacher mood I1, and class behavior I2), a number of hidden nodes (H1-H3) in the second layer (blue), and a single node (O1) in the output layer (orange).



Imagine arrows between these nodes just as in the figure. The information flows from the input nodes along these arrows to the output node. This is also called forward propagation. In reality these arrows are matrices of weights. The first matrix has as many rows as the previous layer nodes and as many columns as the next layer has nodes. This means that each node in the first layer is fully connected to the next layer. For W1 this means we have a matrix of the form 2 by 3. The second matrix w2 has the form 3 by 1. Each entry in this matrix is a weight. This weight defines how much of the value of the previous node is propagated to the next node. This process is also called forward propagation.

In our example the first column in the input matrix I is “teacher mood’ the second row is “class behavior”. After applying our matrices we want our final matrix O to give us a correct answer to whether the homework was hard or easy. As the weight matrices are initialized randomly in most nets that is very unlikely. To get the answers we expect we have to train our network. The process we use for the training is called backpropagation. For this process to work we let our network in its untrained state predict the outcome and calculate the difference between the prediction of our network and the actual observed results. This gives us an error term. With this error term we adapt the weights of each layer and then predict again until we achieve the results we expect.

This is a very crude explanation of how a neural network works but it gives you (hopefully) a slight understanding and a starting point to now investigate how to implement such a model.

Challenge:

- Make yourself familiar with basic concepts and ideas of machine learning and neural networks. Use google, read tutorials, etc. What is back and forward propagation? What is a learning function? This is for yourself. Don’t submit this as part of the solution.
- Using the provided code base, implement two functions for back and forward propagation.
- Based on these functions implement a learning function
- If not existent yet, sign up for github or a similar service and create your own public code repository. We recommend git since it is the standard used in modern software development.
- Submit your code with extensive comments to a public git repository. Submit the link together with the version number as your solution. This is to make sure your fellow students review the version you submitted as solution not later corrections.
- Review the code of others by checking out the revision number of the code they sent as solution.