

Problem Set 1

To run and solve this assignment, one must have a working IPython Notebook installation. The easiest way to set it up for both Windows and Linux is to install [Anaconda](#). Then save this file to your computer, run Anaconda and choose this file in Anaconda's file explorer. Use `Python 3` version. Below statements assume that you have already followed these instructions. If you are new to Python or its scientific library, Numpy, there are some nice tutorials [here](#) and [here](#).

To run code in a cell or to render **Markdown+LaTeX** press `Ctrl+Enter` or `[↵]` (like "play") button above. To edit any code or text cell (double)click on its content. To change cell type, choose "Markdown" or "Code" in the drop-down menu above.

If a certain output is given for some cells, that means that you are expected to get similar results in order to receive full points (small deviations are fine). For some parts we have already written the code for you. You should read it closely and understand what it does.

Total: 85 points.

[30pts] Problem 1: Backprop in a simple MLP

This problem asks you to derive all the steps of the backpropagation algorithm for a simple classification network. Consider a fully-connected neural network, also known as a multi-layer perceptron (MLP), with a single hidden layer and a one-node output layer. The hidden and output nodes use an elementwise sigmoid activation function and the loss layer uses cross-entropy loss:

$$f(z) = \frac{1}{1 + \exp(-z)}$$
$$\mathcal{L}(\hat{y}, y) = -y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y})$$

The computation graph for an example network is shown below. Note that it has an equal number of nodes in the input and hidden layer (3 each), but, in general, they need not be equal. Also, to make the application of backprop easier, we show the *computation graph* which shows the dot product and activation functions at their own nodes, rather than the usual graph showing a single node for both. </p>
</div>
<div data-bbox=