ANLY-590 Assignment 2

September 2019

1 Feedforward: Building a ReLU neural network

Consider the rectified linear activation function : $h_j = \max(0, a_j)$.

1. Draw a network with:
   - 2 inputs
   - 1 hidden layers with 3 hidden units and a
   - 1-class output (for binary classification)
2. Write out the mathematical equation for the output of this network (feel free to break the input-output relationship into multiple equations).
3. Write out the forward-pass function in python, call it `ff_nn__ReLu(...)`
4. Suppose you have the following set of weight matrices:

$$W^{(1)} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & .5 \end{bmatrix} \qquad b^{(1)} = [0, 0, 1]^T \tag{1}$$

$$V = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \qquad c = [1] \tag{2}$$

$$\tag{3}$$

   and a few inputs:

$$X = \begin{bmatrix} 1 & -1 \\ 0 & -1 \\ 0 & 1 \end{bmatrix}$$

   what are the class probabilities associated with the forward pass of each sample?

2 Gradient Descent

Consider a simple non-convex function of two variables:

$$f(x, y) = (1 - x^3) + 100 * (y^2 - x)^2$$

1. What are the partial derivatives of f with respect to x and to y?

2. Create a visualization of the contours of the Rosenbrock function.

3. Write a Gradient Descent algorithm for finding the minimum of the function. Visualize your results with a few different learning rates.

4. Write a Gradient Descent With Momentum algorithm for finding the minimum. Visualize your results with a few different settings of the algorithm's hyperparameters.

3 Backprop

1. For the same network as in Question 1, derive expressions of the gradient of the Loss function with respect to each of the model parameters.

2. Write a function `grad_f(...)` that takes in a weights vector and returns the gradient of the Loss at that location.

3. Generate a synthetic dataset like the XOR pattern (see below).

4. Fit your network using Gradient Descent. Keep track of the total Loss at each iteration and plot the result.

5. Repeat the exercise above using Momentum. Comment on whether your algorithm seems to converge more efficiently.

6. Plot a visualization of the final decision boundary that your model has learned. Overlay the datapoints in this plot.