MNIST digit recognizer NN

The MNIST dataset consists of a 42000 x 785 matrix, where each row represents an entry. The first column is the actual number, and the subsequent 784 columns are pixel values. To normalize grayscale values, each pixel column is divided by 255, mapping the range between 0 and 1.

Separate the training and dev data. With m = 42000 cases, the training data corresponds to columns [1000:m] transposed. The resulting matrix is 785 x (m - 1000). The first row represents Y_train (correct answers), having dimensions 1 x (m - 1000), while the remaining 784 rows form X_train (pixel information) with dimensions 784 x (m - 1000).

The input layer processes X_train, applying a linear transformation followed by the ReLU activation function.

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$

$$A^{[1]} = \text{ReLU}(Z^{[1]})$$

$$\text{ReLU(x)} = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{else} \end{cases}$$

Note that ReLU can operate on a matrix as well acting as an element by element operator.

The second layer is given by, and transformed using the softmax activation function.

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = \text{softmax}(Z^{[2]})$$

$$\text{softmax}(\vec{z}) = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}}$$

The back propagation is given by

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

$$dB^{[2]} = \frac{1}{m} \sum dZ^{[2]}$$

$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * \text{ReLU'}(z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]T}$$

$$dB^{[1]} = \frac{1}{m} \sum dZ^{[1]}$$

Then, according to a given hyperparameter α, the parameters are updated as follows:

$$W^{[2]} = W^{[2]} - \alpha dW^{[2]}$$

$$b^{[2]} = b^{[2]} - \alpha db^{[2]}$$

$$W^{[1]} = W^{[1]} - \alpha dW^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha db^{[1]}$$

For $\alpha = 0.1$ and 500 iterations, we achieve an accuracy of around 0.86. This value is the cap. In order to improve this NN, we could add more layers or perhaps choose different activation functions.