2G03-Project Proposal-Sarah Simionescu

Project Proposal - Sarah Simionescu

Topic

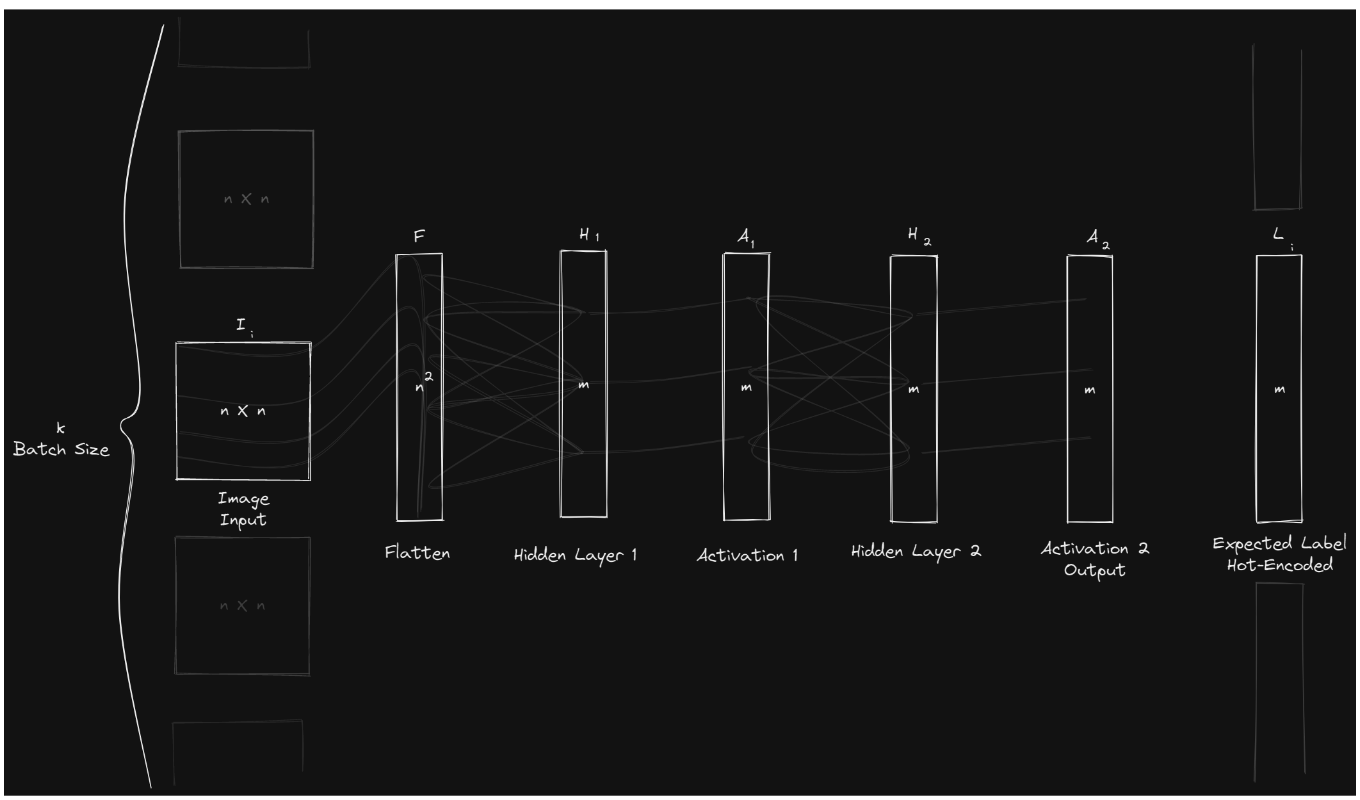
Image classification (Neural Network)

Short Description

I propose to create a Neural Network to classify objects (e.g. faces, flowers, dogs, hand-written digits). The classification will perform based on labelled training data consisting of square black and white images I paired with m possible labels.

Mathematical Description

Neural Network



Let N be a function $\mathbb{R}^{n\times n} \to \{x \in \mathbb{R} | 0 \leq x \leq 1\}^m$ where...

- Input $I \in \mathbb{R}^{n\times n}$ is the pixel values of an $n \times n$ black and white image

$$I = \begin{bmatrix} i_{11} & i_{12} & \dots & i_{1n} \\ i_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ i_{n1} & \dots & \dots & i_{nn} \end{bmatrix}$$

- Output $A_2 \in \mathbb{R}^m$ a column vector where m is the number of labels in the training data. Each label has a corresponding index from $[0, m]$. The greater the element at this index in A_2, the stronger the "likelihood" that label m is the correct label.

$$A_2 = \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_m \end{bmatrix}$$

N consists of

1. one input layer $I \in \mathbb{R}^{n \times n}$ which is simply the input
2. a flattening layer $F \in \mathbb{R}^{n^2}$ which is the columns of I rearranged into one long column vector
3. two hidden layers $H_1, H_2 \in \mathbb{R}^m$ with two corresponding activation layers $A_1, A_2 \in \mathbb{R}^m$ which compress our image into the output $A_2 \in \mathbb{R}^m$ via weights $W_1, W_2 \in \mathbb{R}^{n^2 \times m}$ and biases $B_1, B_2 \in \mathbb{R}^m$

Forward Propagation

This is the process we use to obtain predictions for a given image I i.e. the process of evaluating $N(I) = A_2$

1. $F = flatten(I)$ where $flatten$ lists all the columns of I in one n^2 column vector
2. $H_1 = W_1 \cdot F + B_1$
3. $A_1 = ReLu(H_1)$ *$ReLu$ defined in Additional Definitions
4. $H_2 = W_2 \cdot A_1 + B_1$
5. $A_2 = softmax(H_2)$ *$softmax$ defined in Additional Definitions

W_1, W_2 are referred to as the weights and B_1, B_2 are referred to as the biases. These are trainable parameters that adapt through Backward Propagation to produce increasingly accurate predictions.

Backward Propagation

This is the process we use to adjust the weights and biases to obtain increasingly more accurate predictions.

1. Take a batch of k training images $I \in \mathbb{R}^{n \times n \times k}$ and their corresponding hot-encoded training labels $L \in \{1, 0\}^{m \times k}$
2. Through Forward Propagation, obtain the output for each $i = \{1, 2, ..., k\}$ training image $N(I_i)$ and save the intermediate outputs of the layers to obtain $F, H_1, A_1, H_2, A_2 \in \mathbb{R}^{m \times k}$

For example:

$$A_2 = [N(I_0) \quad N(I_1) \quad \dots \quad N(I_k)]$$

3. Calculate the error of H_2

$$\delta H_2 = A_2 - L$$

4. Find the derivative of the loss function with the respect to the weights W_2 and biases B_2

$$\delta W_2 = \frac{1}{k} \delta H_2 A_1^T$$

$$\delta B_2 = \frac{1}{k} \sum_{i=1}^{k} \delta H_{2i}$$

5. Calculate the error of H_1

$$\delta H_1 = W_2^T \delta H_2 * ReLu'(F)$$

6. Find the derivative of the loss function with respect to the weights W_1 and biases B_1

$$\delta W_1 = \frac{1}{k} \delta H_1 F^T$$

$$\delta B_1 = \frac{1}{k} \sum_{i=1}^{k} \delta H_{1i}$$

7. Update our parameters
   $W_1 := W_1 - \alpha \delta W_1$
   $B_1 := B_1 - \alpha \delta B_1$
   $W_2 := W_2 - \alpha \delta W_2$
   $B_2 := B_2 - \alpha \delta B_2$
   where α is a hyperparameter called the learning rate

Additional Definitions

For a column vector X where $i \in \{1, 2, ..., |X|\}$

$$ReLu(X) = \begin{cases} x_i \text{ if } x_i > 0 \\ 0 \text{ if } x_i \leq 0 \end{cases}$$

$$softmax(X) = \frac{e_i^x}{\sum_{j=1}^{K} e_j^x}$$

$hot - encode(x)$ is a function $\mathbb{I} \rightarrow \{1, 0\}^m$ which returns a column vector with all of its elements set to 0, except for the element at index x, which is set to 1