ML Problem Solving

Gradient Descent + One-Hot Encoding

Using a calculator and/or a python+numpy environment is recommended for this PSS. Make sure you write your intermediate results, for instance, write down the matrices/tensors with all values for the initial calculations and on iterative steps, write down only the tensors where the values have changed.

Given is the following dataset (same as previous PSS), where each row is a sample represented by a 4-pixel image and the measured output is the last column.

D_{train} = [[255, 128, 128, 0, 0], [55, 128, 128, 128, 1], [192, 128, 128, 0, 0],

[100, 128, 128, 100, 1], [30, 64, 128, 30, 2], [20, 64, 128, 0, 2]]

Consider a neural network (NN) composed of 3 linear activations (a 3-class fully-connected layer), followed by softmax, parameterized by weights and biases B_{1x3}, initialized with the values 0.

We will use cross-entropy loss L. Recall that:

$$\Delta w_{i,j} = -\alpha \frac{\partial L}{\partial w_{i,j}} = \alpha\ x_i \begin{cases} (1 - p_j), & \text{if answer } a = j; \\ -p_j, & \text{otherwise.} \end{cases}$$

$$\Delta b_j = -\alpha \frac{\partial L}{\partial b_j} = \alpha \begin{cases} (1 - p_j), & \text{if } a = j; \\ -p_j, & \text{otherwise.} \end{cases}$$

where a is the ground-truth class, p_j is the softmax of $l_j = b_j + x \cdot w_j$.

1. Write a feature matrix as X'_{6x5} incorporating the bias such that $X' \cdot W' = X \cdot W + B$.

2. Write down Y_{6x3} as a one-hot encoded ground-truth (target) matrix.

3. Now let's try to write a vector-form of the update equation $W'_{5x3} = W_{5x3} + D_{5x3}$ for this loss (taking advantage of the one-hot encoding). Call the logit output matrix Z and the softmax output $S = softmax(Z)$. Hint: your logit output matrix Z should match the shape of your one-hot encoded groud-truth matrix Y. Write the update for **1 sample** (1 example in X). Optional: write down an equation for all samples.