1 Finding the minimal change that will get rid of the network's WaterMark

Given a watermarked trained neural network as described here. We tested what is the minimal change to the network last layer in order to "remove" some watermarks from the network.

1.1 Defining the problem

Neural network decision for an input v is defined as the coordinate with the maximal value, if the network output is the vector $\overrightarrow{out}$ Given a watermarked network N with a set of K watermarks $\{w_1, \cdots, w_K\}$ we'll mark the network last layer L so L is a $m \times n$ matrix were n is the layer's number of neurons and m is the network output size. The change to the last layer will be a matrix with the same dimension as L we'll mark as ε, so $\varepsilon_{i,j}$ is the change to the last layer matrix entry $L_{i,j}$. The overall change to the layer will be $\|\varepsilon\|_1$. For a certain input v we're only interested in the input to the last layer we'll mark the input to the last layer u. u is a $n \times 1$ vector. So our new changed network output is $(L+\varepsilon)u$. For a single watermark w we need to find the minimal ε so that the $argmax$

1.2 Defining the problem

$$\begin{aligned}
\sum_{x \neq y} \|x-y\|_q^q &= \sum_{x \neq y} \sum_{l=1}^{k} (x)_l - (y)_l{}^q \\
&= \sum_{l=1}^{k} \sum_{x \neq y} (x)_l - (y)_l{}^q \\
&= \sum_{i=1}^{k} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (x_i)_l - (x_j)_l{}^q \\
\text{(assume } (x_1)_l \geq (x_2)_l \geq \cdots \geq (x_n)_l) &= \sum_{i=1}^{k} \sum_{i=1}^{n} \sum_{j=i+1}^{n} ((x_i)_l - (x_j)_l)^q
\end{aligned}$$

*Note that if q is an even number we don't need the sort.

Assuming $x \geq y$:

$$x - y^q = (x-y)^q = \sum_{i=0}^{q} \binom{q}{i} x^i y^{q-i}$$

So we get:

$$\sum_{x \neq y} \|x-y\|_q^q = \sum_{i=1}^{k} \sum_{i=1}^{n} \sum_{j=i+1}^{n} ((x_i)_l - (x_j)_l)^q$$