Representation of “The Hashing Trick” layer as 2D matrices

Introduction

A standard fully-connected layer is written as:

$$o = g(z) = g(aW)$$

Where

- o is the output vector, length M.
- a is the input vector, size N
- W is $N \times M$.

With a hashed layer, we want to have a similar representation $o = g(z) = g((aH)W)$ with H determined by the hash, and W a matrix of adjustable weights. We will see how these matrices can be constructed. To force a 2D representation, we will need to use very long vectors and matrices, with sparsity, repititions, or both.

Developing the H matrix

In simple terms, this is the mechansim proposed in the paper:

- Inputs to the algorithm: parameter K
- Create (via hashing), M splits of the input vector a. Each split assigns an element a_j into one of K groups. A “good” hash function will ensure that:
    - For each split, the probablity of a particular element a_j to fall in any particular group is equal between the groups (and therefore equals $1/K$)
    - The group assignments in each split are “as independent as possible” from the other splits (pairwise-independence or better)

This can be expressed as follows:

(a number in brackets $[L]$ denotes the set of natural numbers from 1 to L):

- $h_i : [N] \rightarrow [K],\ i \in [M]$
- $a'_{i,k} = \sum_{j : h_i(j) = k} a_j.$

a' needs to be double indexed, hence acquiring a 2D, or matrix, form. But we want to avoid this since this breaks the ordinary notation where the neurons in a layer are represented as a vector. For this we introduce a new index letter $q \in [MK]$ so that

$$a'_q = \sum_{j : h_{\lfloor q/K \rfloor}(j) = q \mod K} a_j$$

We are now ready to describe the matrix H:

- $H \in \{0, 1\}^{N \times MK}$.
- $H_{j,q} = 1 \Longleftrightarrow h_{\lfloor q/K \rfloor}(j) = q \mod K.$

So the intermediate layer created by the hash, is actually much larger than both N and M and “codes” M splits of the integers 1 to N. If we take just the first K columns of H, we will have the value 1 exactly once in each row. This also holds for the second group of K columns, third, and so forth. The probability of a 1 in H is $\frac{N}{NK} = \frac{1}{K}$.

This completes our analysis of H. we now need to understand the structure of W.

The structure of W

$W \in \mathbb{R}^{MK \times M}$, since it takes a vector of lengh MK as inputs and spits out a vector of length M. It includes only K unique values. The first column has the unique, nonzero values running from row 1 to K. The rest of the column is filled with zeros. The 2nd column starts with K zeros, then the K unique values, then zeros all the way down. In the 3rd column, the nonzero values start in position $2K + 1$ and run up to position $3K$.