1 The function c

Semantically, the function $c(v_1, \ldots, v_d)$ computes the L2 norm without square root:

$$
\begin{aligned}
c &: \mathbb{R}^d \to \mathbb{R} \\
c(v) &= \sum_{i \in \{1 \ldots d\}} v_i^2
\end{aligned}
$$

Our validity condition is that $c(v) \leq 1$.

This is because v is the vector of gradients a client computed using its own (private) dataset. The gradient vectors of all clients $\{v_x : \mathbb{R}^d | x \in \text{clients}\}$ should be aggregated to the sum $v_{\text{agg}} : \mathbb{R}^d$. To make this differentially private, noise can be added to the shares of v_{agg} homomorphically by each server, this way nobody has access to the unnoised aggregate. The problem is that privacy can only be guaranteed if the gradient vector v_x of a given client x has norm ≤ 1, hence our validity predicate.

2 Encoding c

Because of the validity predicate, the individual gradient entries of a client $v_{i,x} : \mathbb{R}$ can only lie in the range $[-1, 1]_{\mathbb{R}}$. This allows us to interpret them as fixed point numbers, and encode them as integers in $\left[-2^{n-1}, 2^{n-1}\right)_{\mathbb{Z}}$ for a fixed resolution n. Since we cannot directly use negative integers with prio, we translate this range and get $v_{i,x} : [0, 2^n)_{\mathbb{N}}$.

Thus, from the point of view of prio, the function c is as follows:

$$
\begin{aligned}
c &: [0, 2^n)^d \to \left[0, d \cdot 2^{2n+1}\right) \\
c(v) &= \sum_{i \in \{1 \ldots d\}} v_i^2 + 2^{2n-2} - 2^n v_i
\end{aligned}
$$

The term is such that for each i the summand should never become negative. The validity condition is that $c(v) \leq 2^{2n-2} - 1$.

3 Other

It might be interesting to add that d might be relatively large: for a small neural network it would be something like 250.