**Parameter size:**

Each parameter (weight) in the model is stored as a floating-point number. Typically, these are stored as 32-bit (or 4 bytes) floating-point values in most deep learning frameworks.

Total parameters: From the model summary, **the model has 371,696 parameters**.

Size per parameter: If each parameter is stored as a **32-bit (4-byte) floating-point value, the size of** **each parameter is 4 bytes**.

To calculate the total size of the weights, we multiply the total number of parameters by the size of each parameter:

Toal size (bytes) = Total number of parameter x Size per parameter

Total size (KB) = Total size (bytes) / 1024

Calculation:

* Total parameters = 371,696
* Size per parameter = 4 bytes (32-bit floating point)

**Total size in bytes:**

**Total size (bytes) = 371,696 x 4 = 1,486,784 bytes**

**Total size (KB) = 1,486,784/1024 = 1,451 KB**

To further reduce the size, below techniques can be used

**Quantization**: Quantization reduces the precision of the weights and activations (from float32 to int8 or float16), thus reducing the model size and speeding up inference.

We can choose different levels of quantization (dynamic range, float16, full integer) based on the trade-off between model size and accuracy.

**Compress weights** using existing compression methods (zip, gzip)

**Weight pruning?**

Increase parameters and decrease float

Work on decreasing prediction time

Hashing