Institución Universitaria
Reacreditada en Alta Calidad

Quantization – Cuantificación

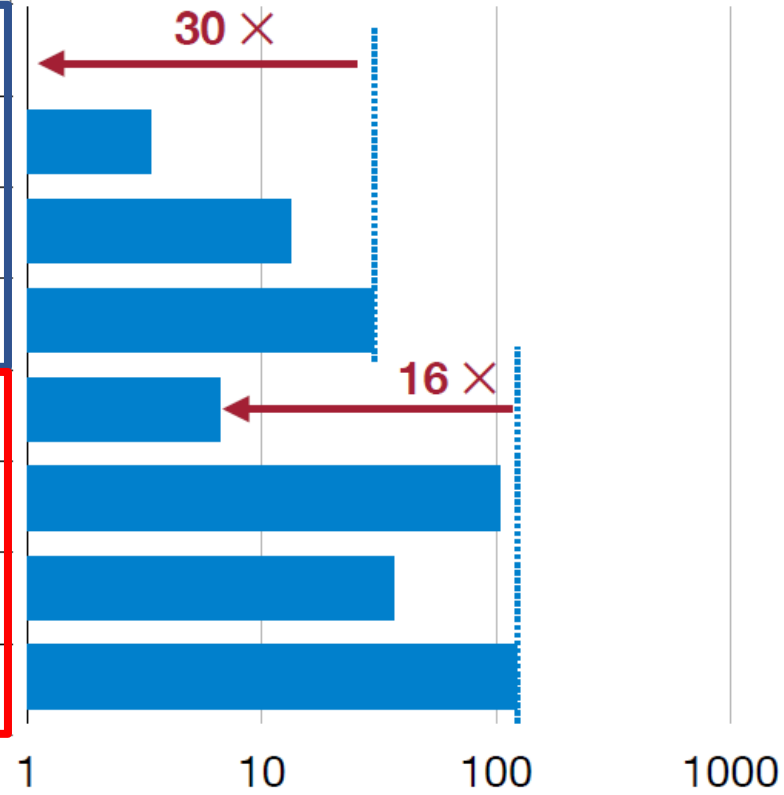
INTEGRACIÓN DE ML EN EMBEBIDOS Y EDGE COMPUTING

Contenido

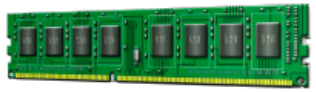
1. Costos de operaciones
2. Cuantificación
3. Cuantificación basada en K-means

Costos de Operaciones con bits

Operation	Energy [pJ]
8 bit int ADD	0.03
32 bit int ADD	0.1
16 bit float ADD	0.4
32 bit float ADD	0.9
8 bit int MULT	0.2
32 bit int MULT	3.1
16 bit float MULT	1.1
32 bit float MULT	3.7



Rough Energy Cost For Various Operations in 45nm 0.9V

1  = 200 ×+

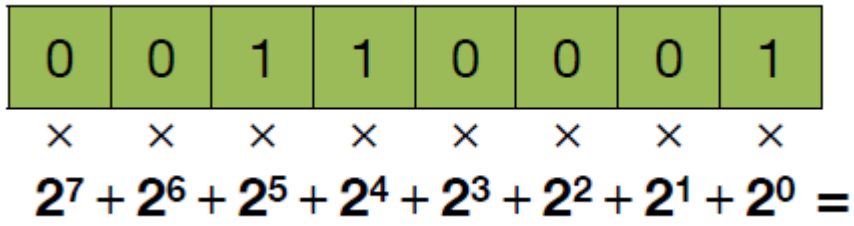
Computing's Energy Problem (and What We Can Do About it) [Horowitz, M., IEEE ISSCC 2014]

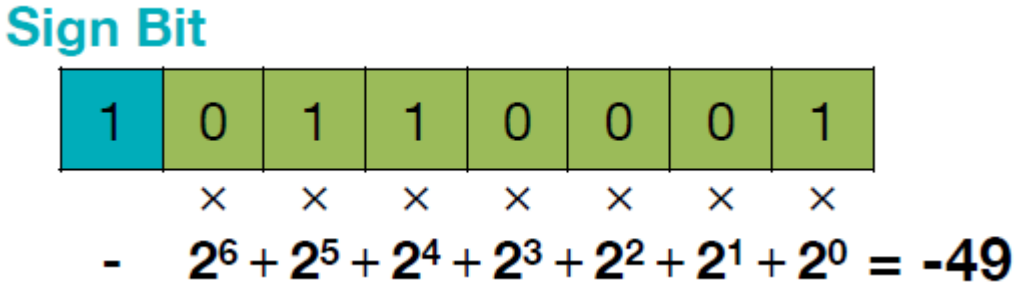
Enteros

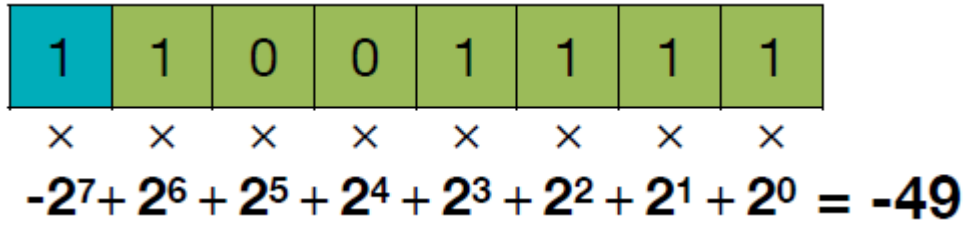
Unsigned Integer
• *n*-bit Range: $[0,2^n-1]$

Signed Integer
• Sign-Magnitude Representation
  • *n*-bit Range: $[-2^{n-1}-1, 2^{n-1}-1]$
  • Both 000…00 and 100…00 represent 0

• Two's Complement Representation
  • *n*-bit Range:
  • 000…00 represents 0
  • 100…00 represents -2^{n-1}





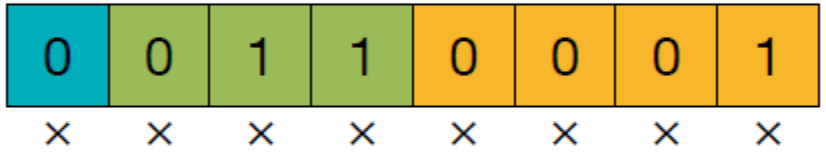


Computing's Energy Problem (and What We Can Do About it) [Horowitz, M., IEEE ISSCC 2014]

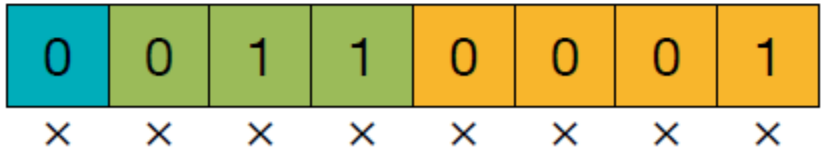
Fixed-Point Number

Integer . Fraction

"Decimal" Point

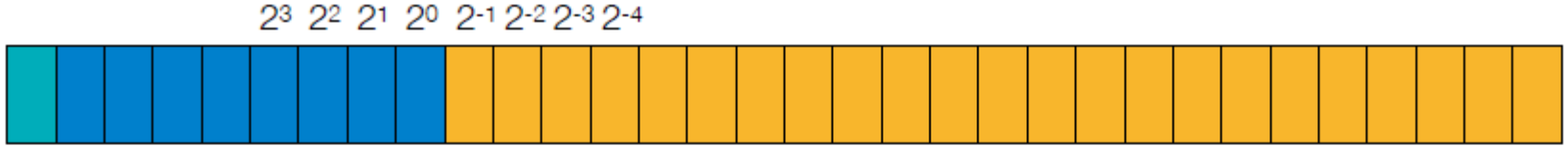


$$-2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4} = 3.0625$$



$$(\ -2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0\ ) \times 2^{-4} = 49 \times 0.0625 = 3.0625$$

Floating-Point Number

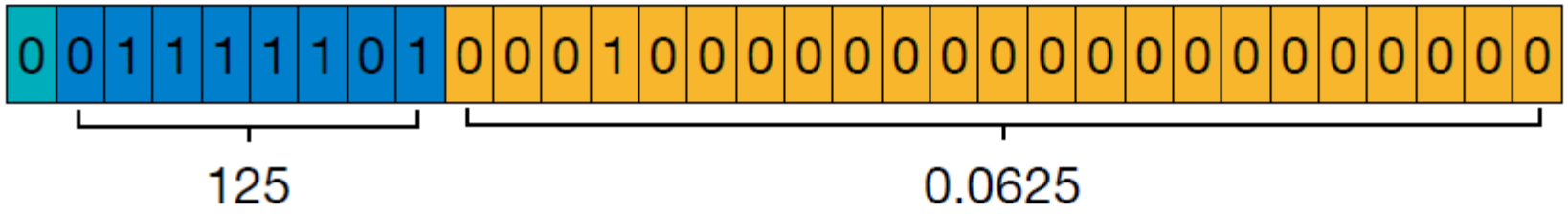


Sign 8 bit Exponent **23 bit Fraction**

$$(-1)^{sign} \times (1 + \textbf{Fraction}) \times 2^{\textbf{Exponent}-127}$$ ← Exponent Bias = 127 = $2^{8-1}-1$

(significant / mantissa)

$$0.265625 = 1.0625 \times 2^{-2} = (1 + \underline{0.0625}) \times 2^{\underline{125}-127}$$



Floating-Point Number

Exponent Width → Range; Fraction Width → Precision

IEEE 754 Single Precision 32-bit Float (IEEE FP32)

IEEE 754 Half Precision 16-bit Float (IEEE FP16)

Google Brain Float (BF16)

Exponent (bits)	Fraction (bits)	Total (bits)
8	23	32
5	10	16
8	7	16

Floating-Point Number

Exponent Width → Range; Fraction Width → Precision

IEEE 754 Single Precision 32-bit Float (IEEE FP32)

IEEE 754 Half Precision 16-bit Float (IEEE FP16)

Google Brain Float (BF16)

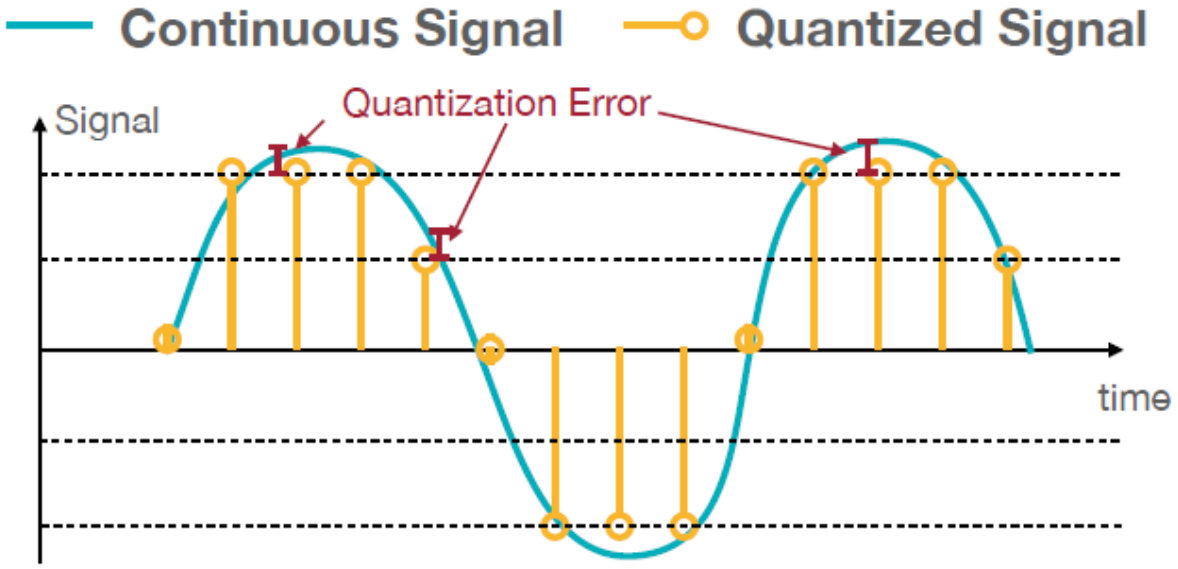
Exponent (bits)	Fraction (bits)	Total (bits)
8	23	32
5	10	16
8	7	16

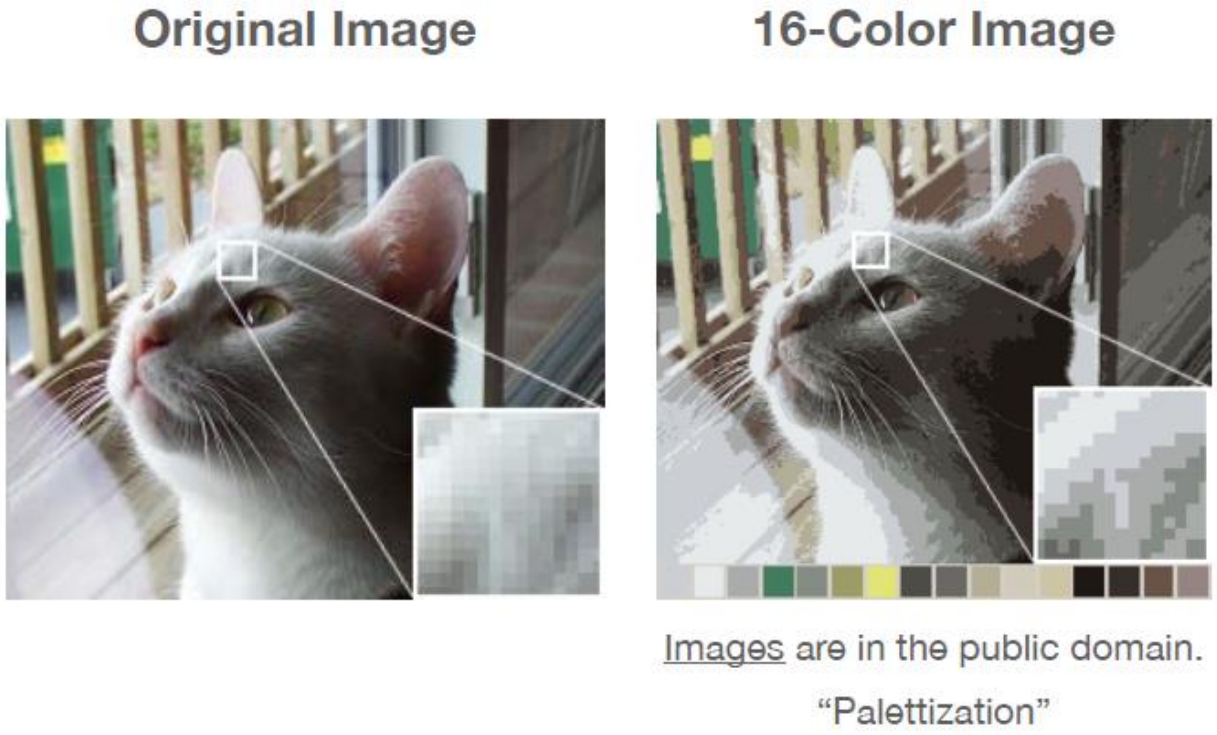
Contenido

1. Costos de operaciones
2. Cuantificación
3. Cuantificación basada en K-means

Que es cuantificación?

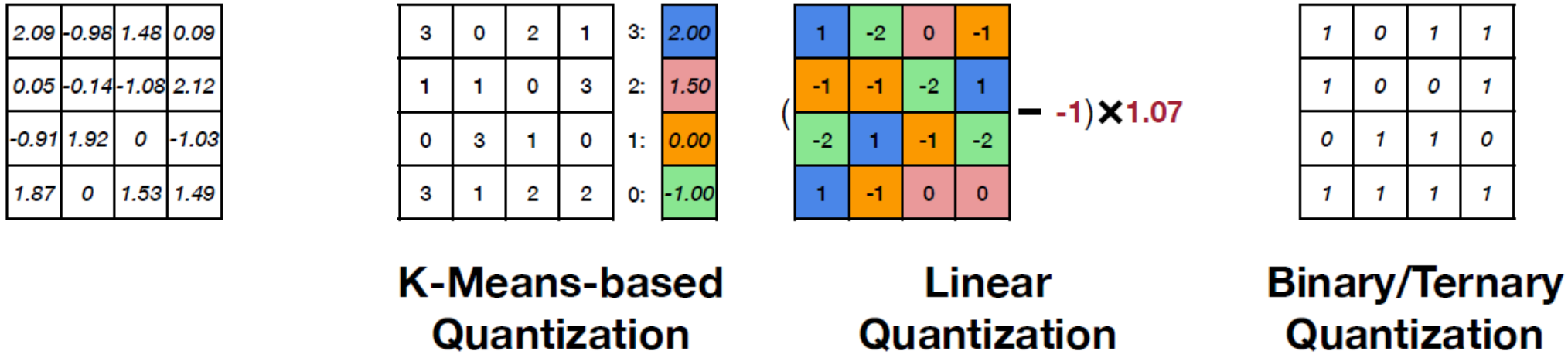
La cuantificación es el proceso de restringir una entrada desde un conjunto continuo o con valores grandes a un conjunto discreto.





https://en.wikipedia.org/wiki/Quantization_(signal_processing)

Neural Network Quantization



Almacenamiento	**Pesos de punto flotante**	**Pesos Enteros; Libro de código de punto flotante**	**Pesos Enteros**	**Pesos binarios/ternarios (-1,0,+1)**
Calculo	Aritmética de punto flotante	Aritmética de punto flotante	Aritmética Entera	Operaciones de bits

Contenido

1. Costos de operaciones
2. Cuantificación
3. Cuantificación basada en K-means

Weight Quantization

weights
(32-bit float)

2.09	*-0.98*	*1.48*	*0.09*
0.05	*-0.14*	*-1.08*	*2.12*
-0.91	*1.92*	*0*	*-1.03*
1.87	*0*	*1.53*	*1.49*

~~2.09, 2.12, 1.92, 1.87~~

2.0

K-means Weight Quantization

reconstructed weights
(32-bit float)

2.00	-1.00	1.50	0.00
0.00	0.00	-1.00	2.00
-1.00	2.00	0.00	-1.00
2.00	0.00	1.50	1.50

quantization error

0.09	0.02	-0.02	0.09
0.05	-0.14	-0.08	0.12
0.09	-0.08	0	-0.03
-0.13	0	0.03	-0.01

storage

32 bit × 16 = 512 bit = 64 B

2 bit × 16 = 32 bit = 4 B + 32 bit × 4 = 128 bit = 16 B = 20 B

3.2 × smaller

Assume N-bit quantization, and #parameters = $M >> 2^N$.

32 bit × M = $32M$ bit

N bit × M = NM bit

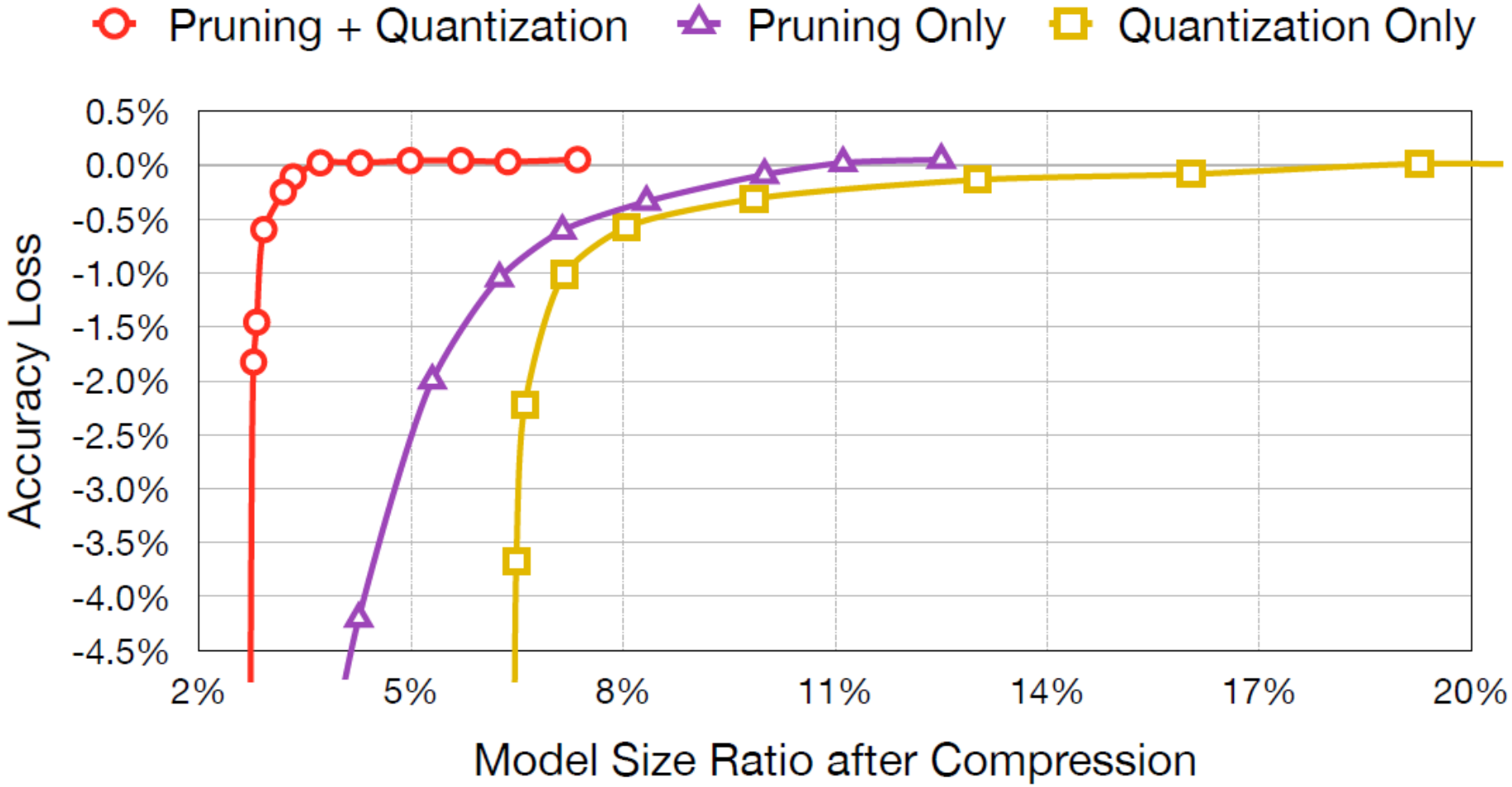
~~32 bit × 2^N = 2^{N+5} bit~~

$32/N$ × smaller

Deep Compression [Han *et al.*, ICLR 2016]

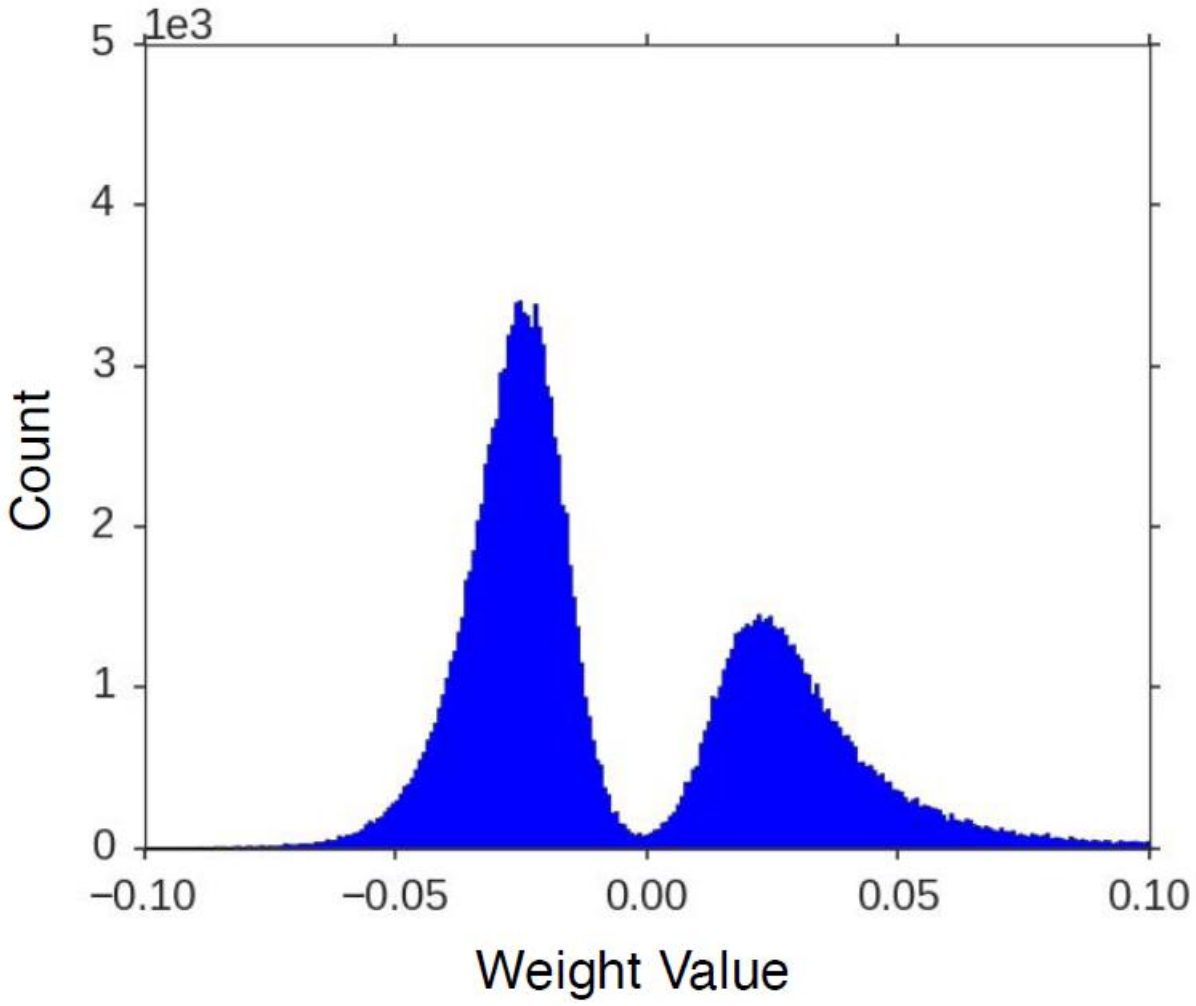
K-means Weight Quantization

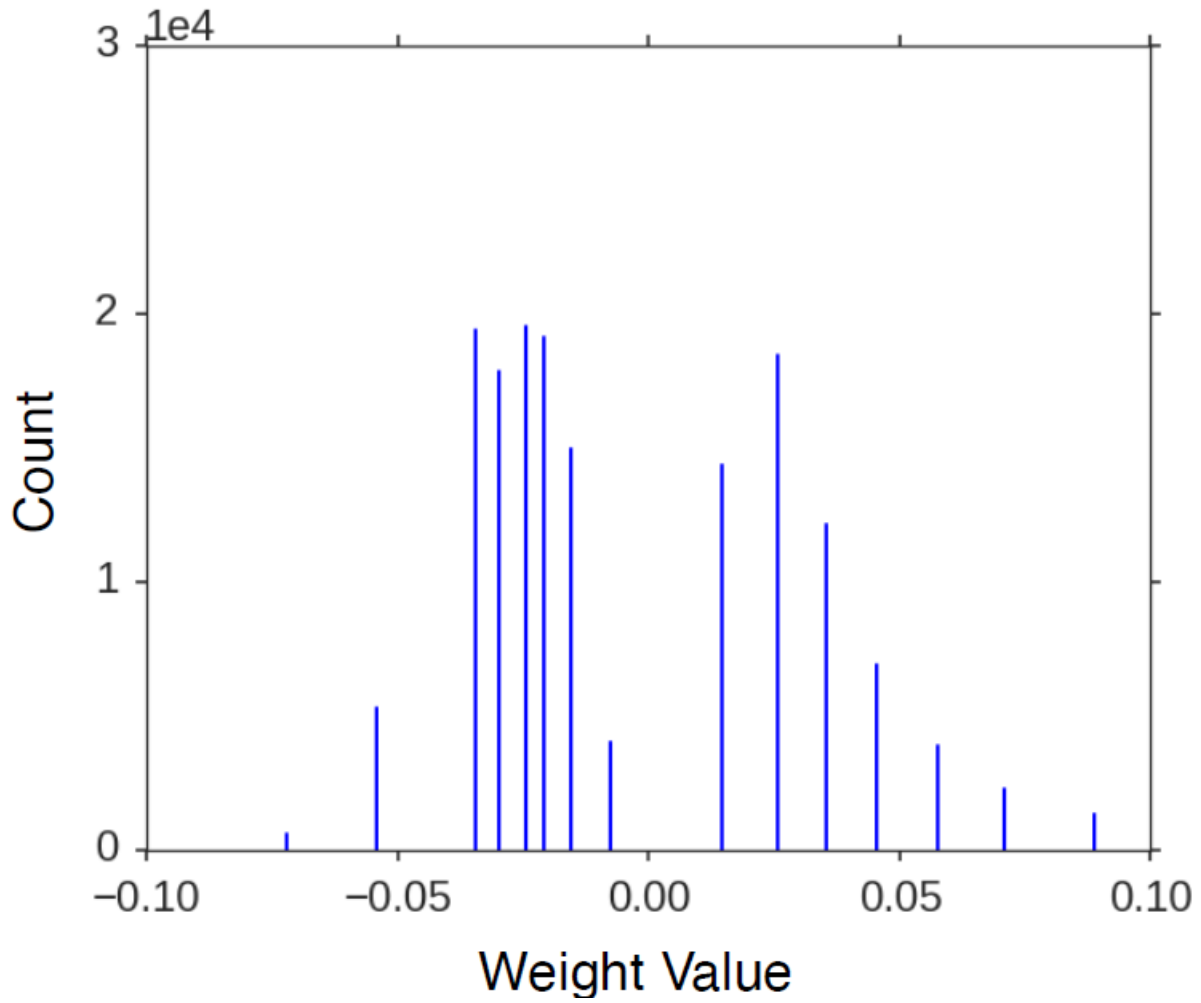
Accuracy vs. compression rate for AlexNet on ImageNet dataset



Deep Compression [Han *et al.*, ICLR 2016]

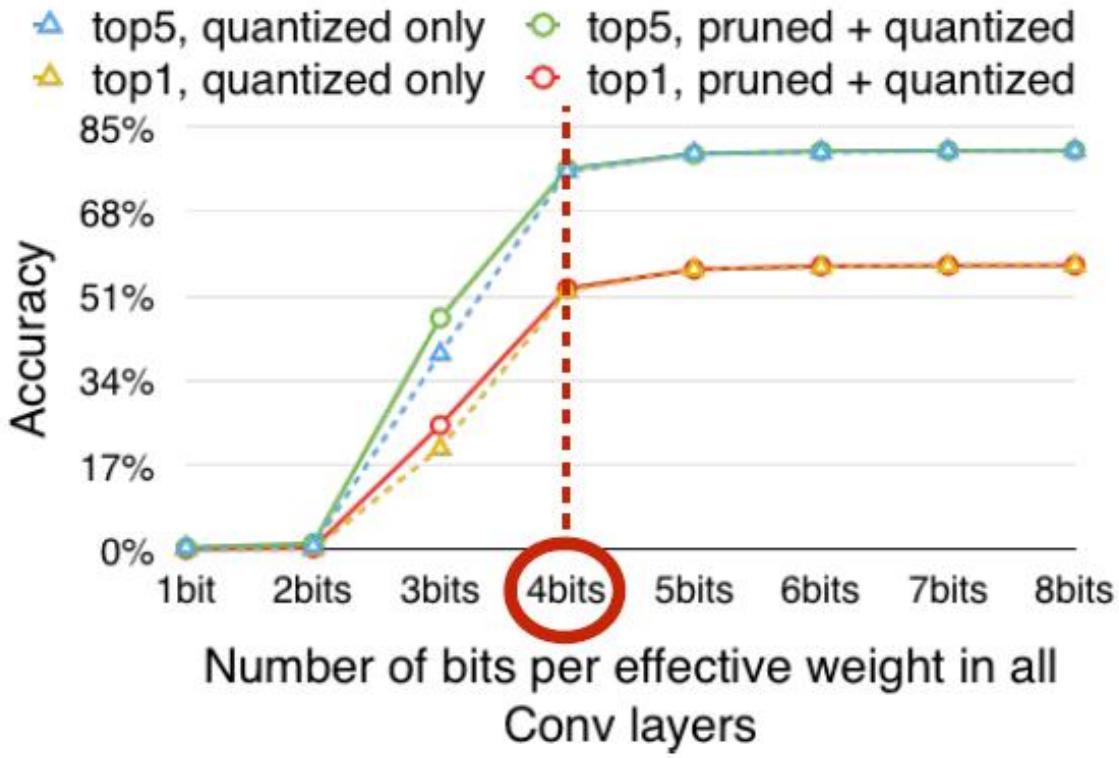
K-means Weight Quantization

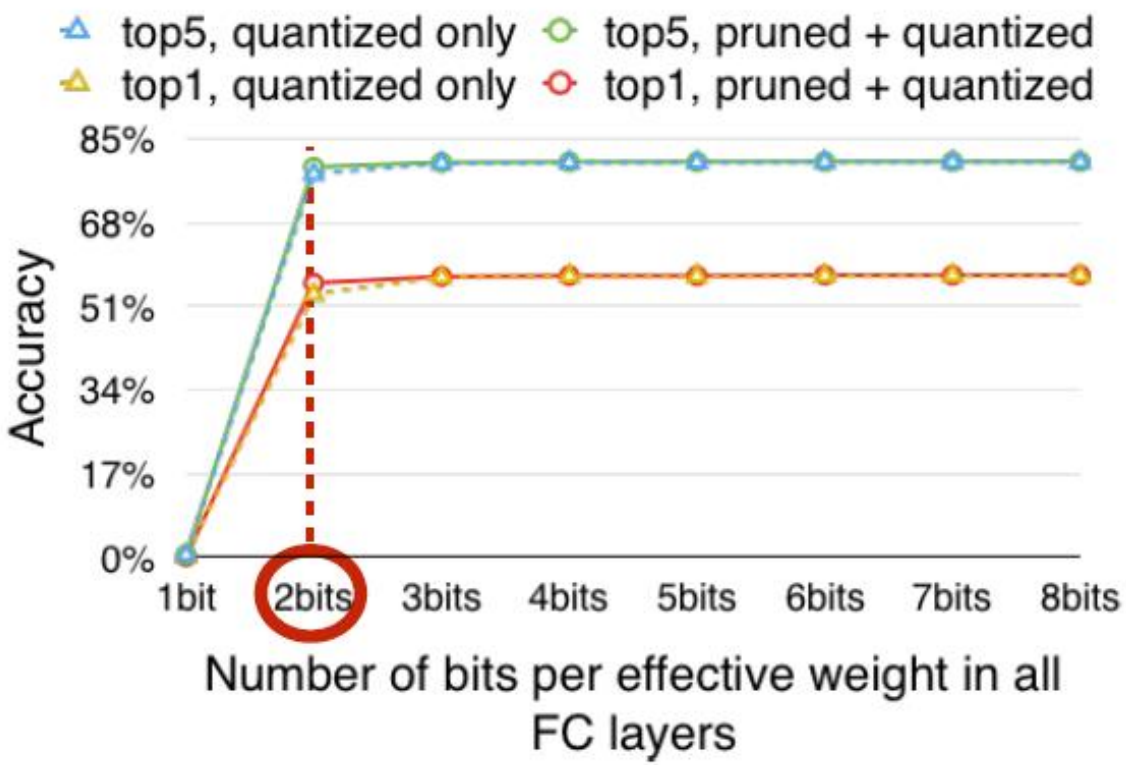




Deep Compression [Han *et al.*, ICLR 2016]

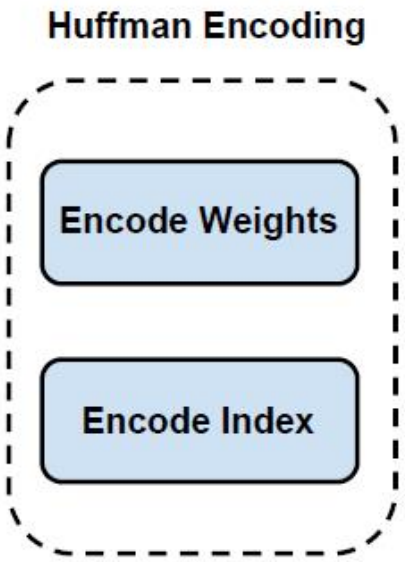
Cuantos bits se necesitan?

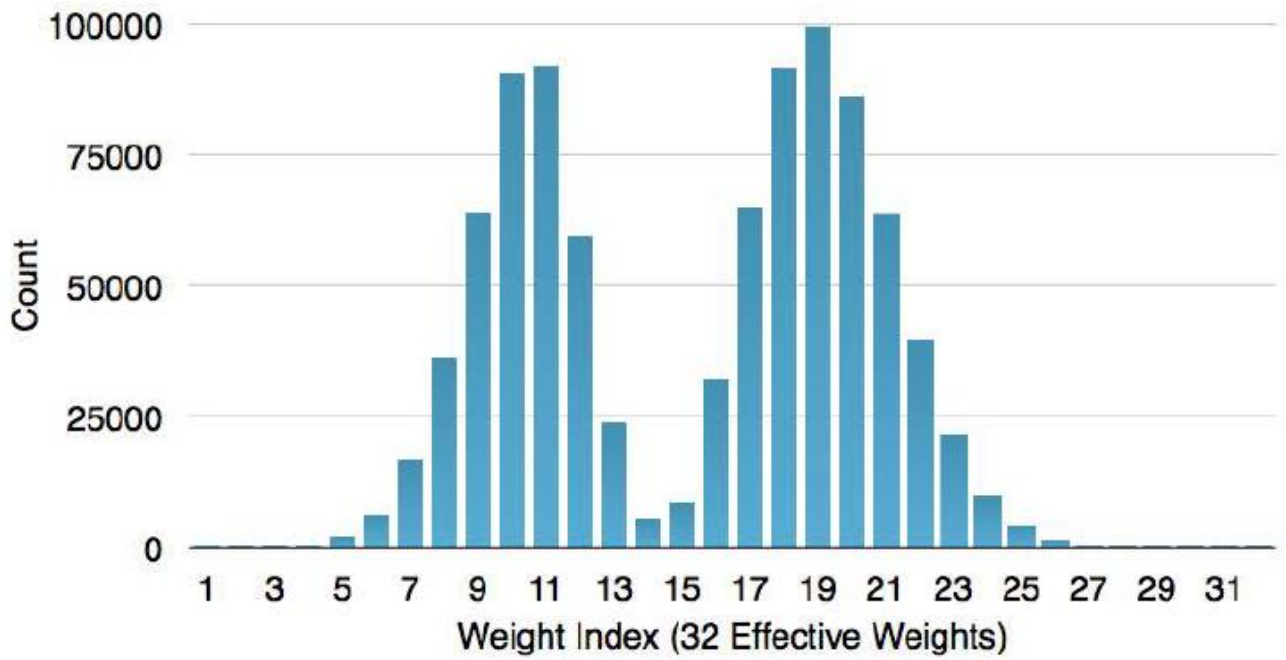




Deep Compression [Han *et al.*, ICLR 2016]

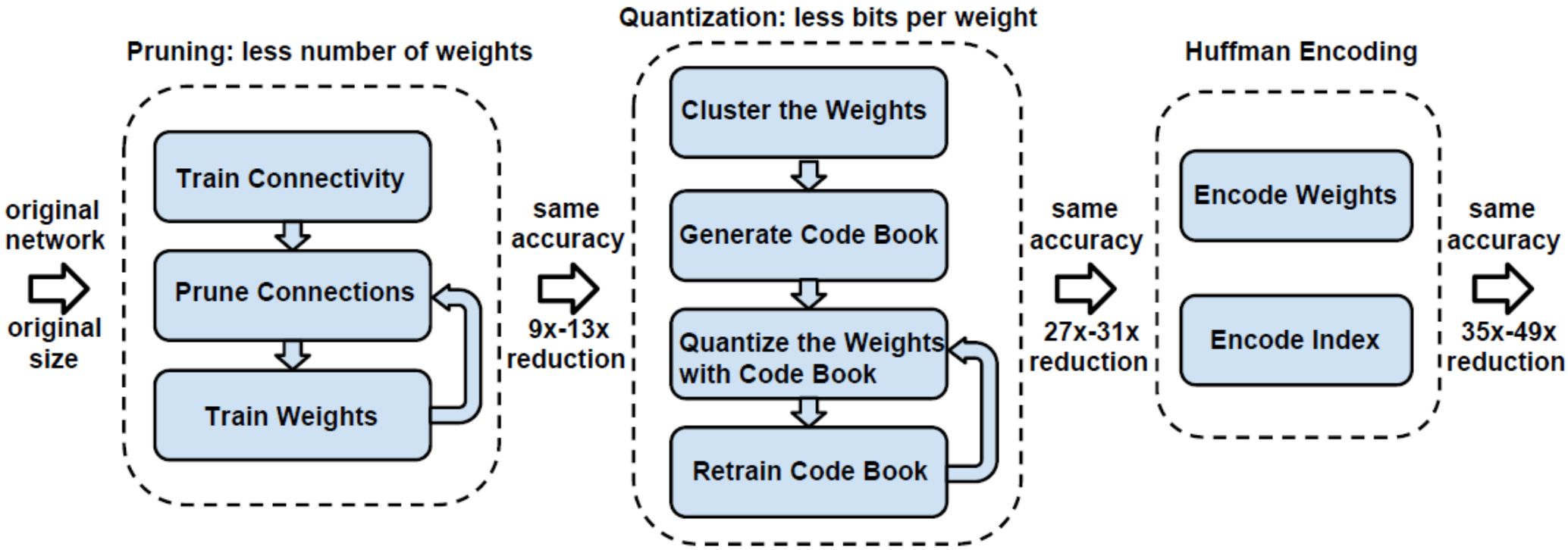
Huffman Coding





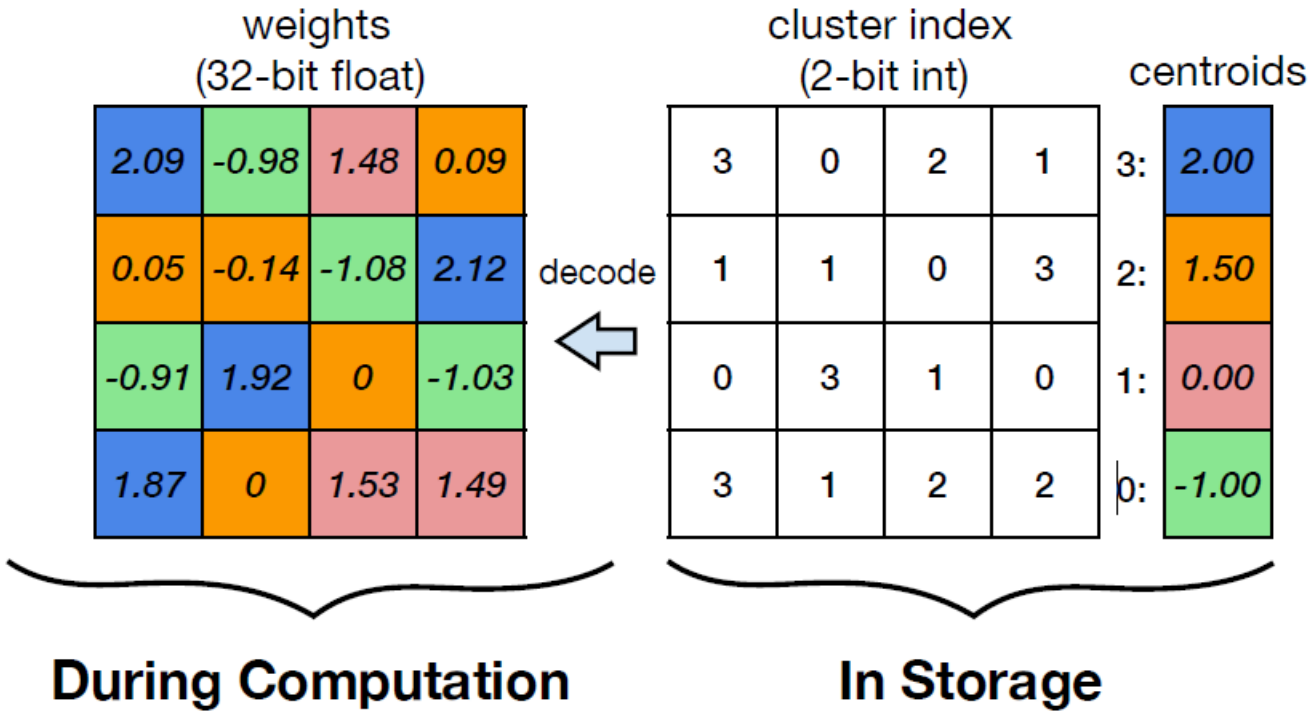
Deep Compression [Han *et al.*, ICLR 2016]

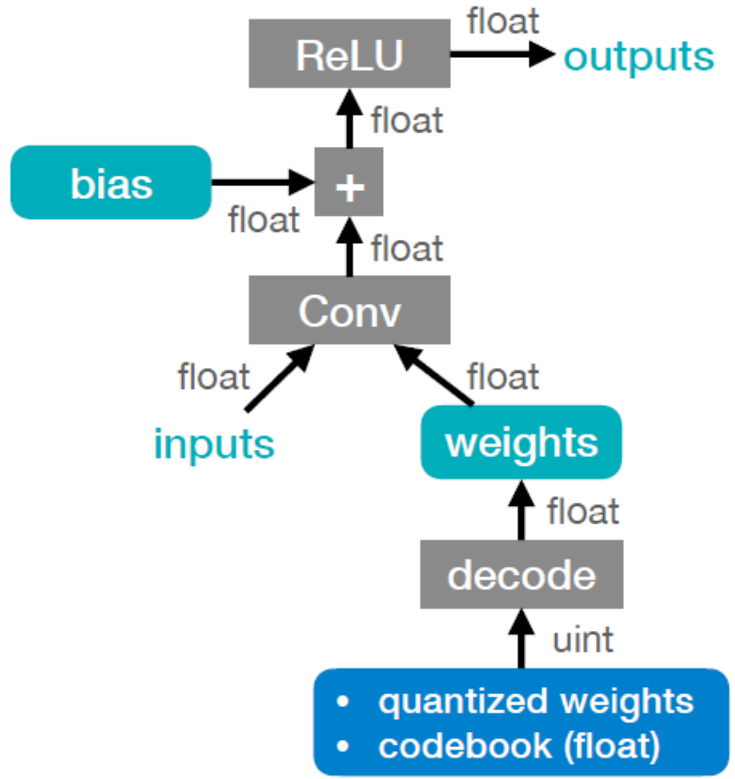
Resumen de Deep Compression



Deep Compression [Han *et al.*, ICLR 2016]

K-means-based Weight Quantization



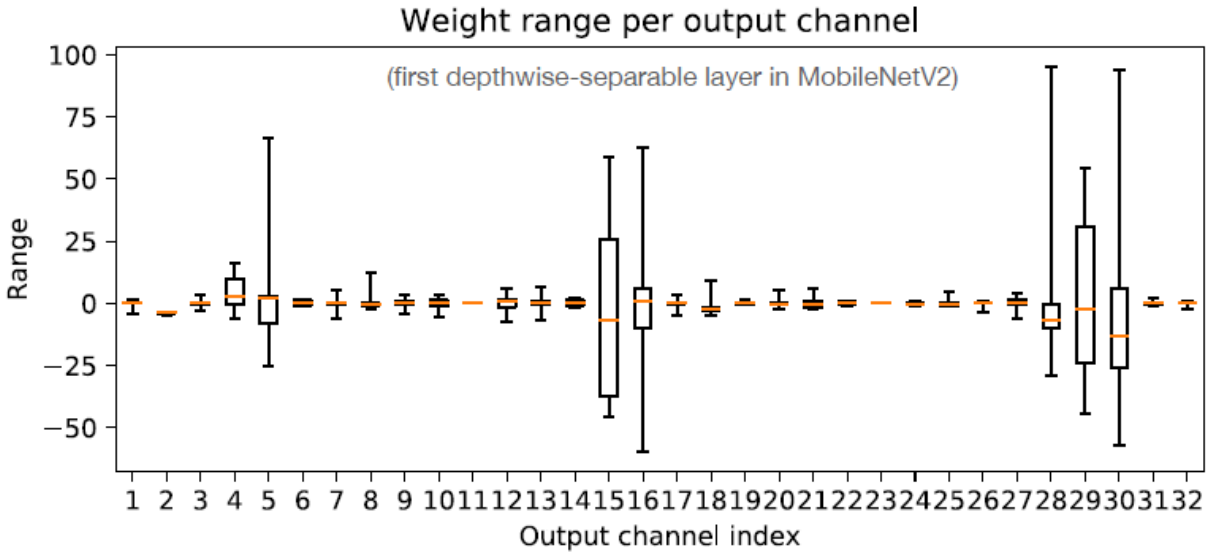


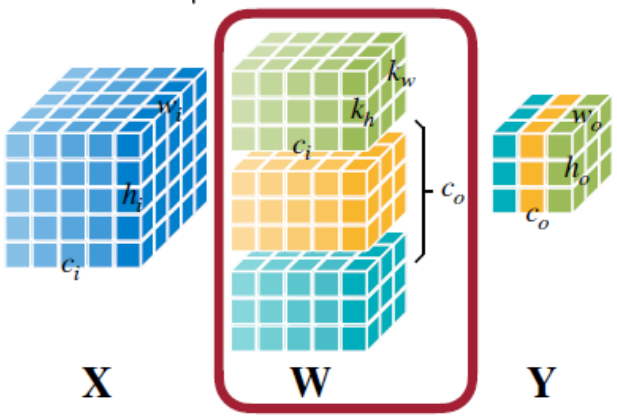
Los pesos son descomprimidos usando una lookup table (por ejemplo codebook) durante la inferencia.
La cuantificación de pesos basada en K-Means sólo ahorra costes de almacenamiento de un modelo de red neuronal.
- Todo el computo y accesos de memoria son aun de punto flotante.

Contenido

1. Costos de operaciones
2. Cuantificación
3. Cuantificación basada en K-means
4. Post-training quantization

Post-training Quantization





- $|r|_{\max} = |\boldsymbol{W}|_{\max}$
- Usando una sola escala S para todos los pesos del tensor (**Per-Tensor Quantization**)
  Funciona bien para modelos grandes.
  El acierto cae para modelos pequeños.

- Falla en casos como
  Diferencias muy grandes (mas de 100x) en los rangos de los pesos para diferentes canales de salida – pesos atípicos.

- Solución: **Per-channel Quantization**

Post-training Quantization

Per-Channel Quantization

oc \ ic				
	2.09	-0.98	1.48	0.09
	0.05	-0.14	-1.08	2.12
	-0.91	1.92	0	-1.03
	1.87	0	1.53	1.49

$$|r|_{\max} = 2.09 \qquad S_0 = 2.09$$

$$|r|_{\max} = 2.12 \qquad S_1 = 2.12$$

$$|r|_{\max} = 1.92 \qquad S_2 = 1.92$$

$$|r|_{\max} = 1.87 \qquad S_3 = 1.87$$

Quantized			
1	0	1	0
0	0	-1	1
0	1	0	-1
1	0	1	1

Reconstructed			
2.09	0	2.09	0
0	0	-2.12	2.12
0	1.92	0	-1.92
1.87	0	1.87	1.87

$$\|\mathbf{W} - \mathbf{S} \odot \mathbf{q}_\mathbf{W}\|_F = 2.08$$

Per-Tensor Quantization

$$|r|_{\max} = 2.12$$

$$S = \frac{|r|_{\max}}{q_{\max}} = \frac{2.12}{2^{2-1} - 1} = 2.12$$

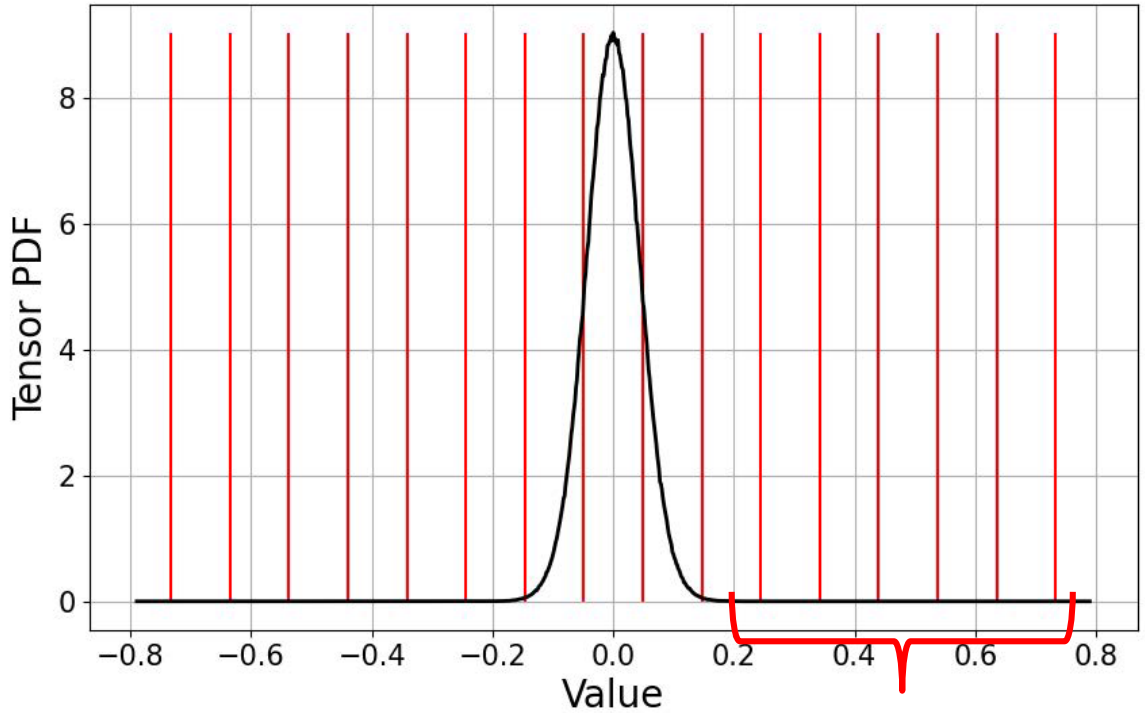
Quantized			
1	0	1	0
0	0	-1	1
0	1	0	0
1	0	1	1

Reconstructed			
2.12	0	2.12	0
0	0	-2.12	2.12
0	2.12	0	0
2.12	0	2.12	2.12

$$\|\mathbf{W} - S\mathbf{q}_\mathbf{W}\|_F = 2.28$$

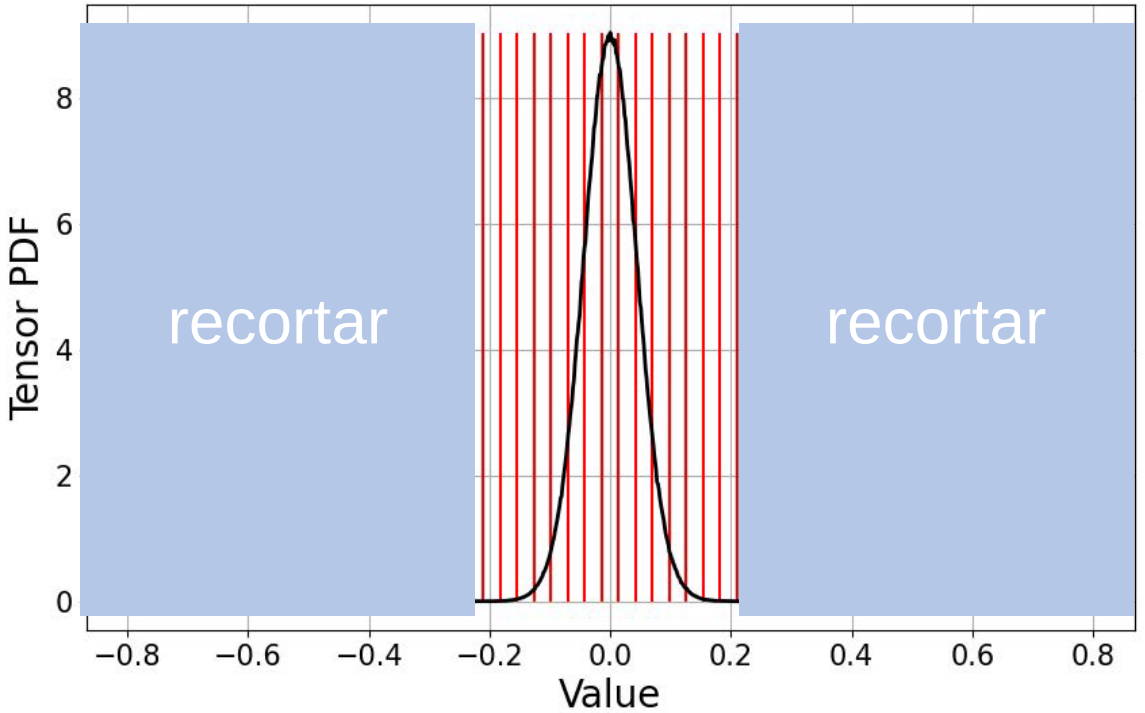
Cuantificación de rango dinámico

Cuantificación de toda la escala



Valores de baja densidad

Cuantificación recortada



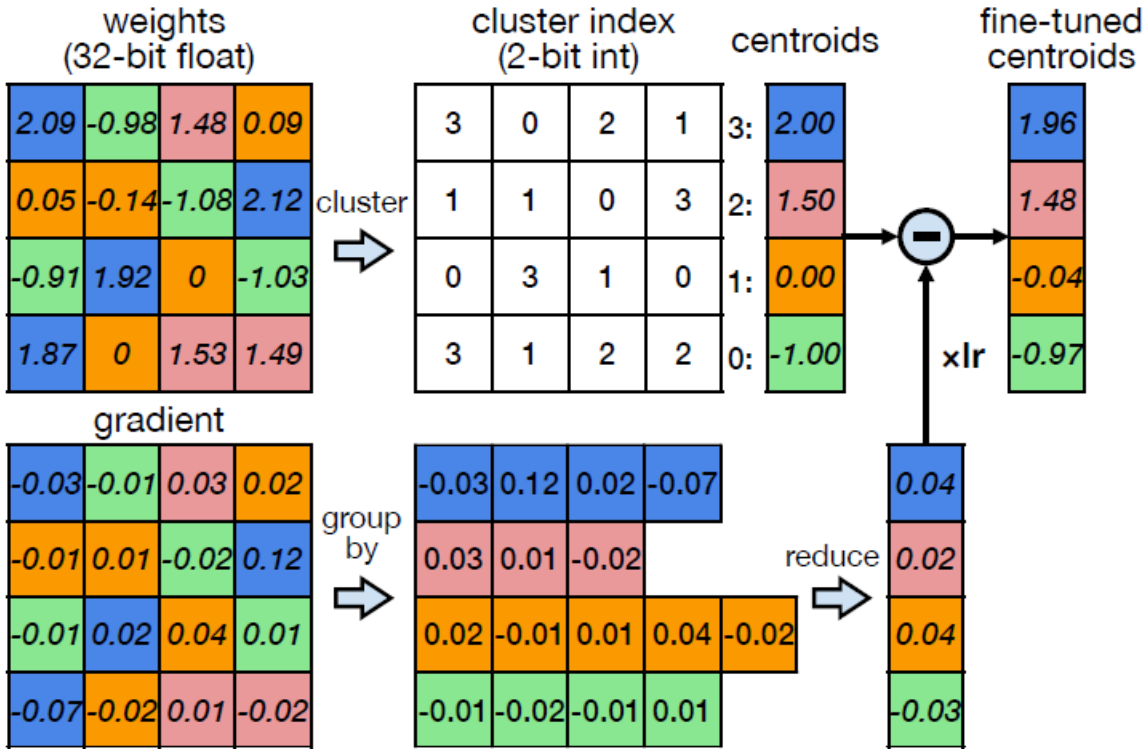
Contenido

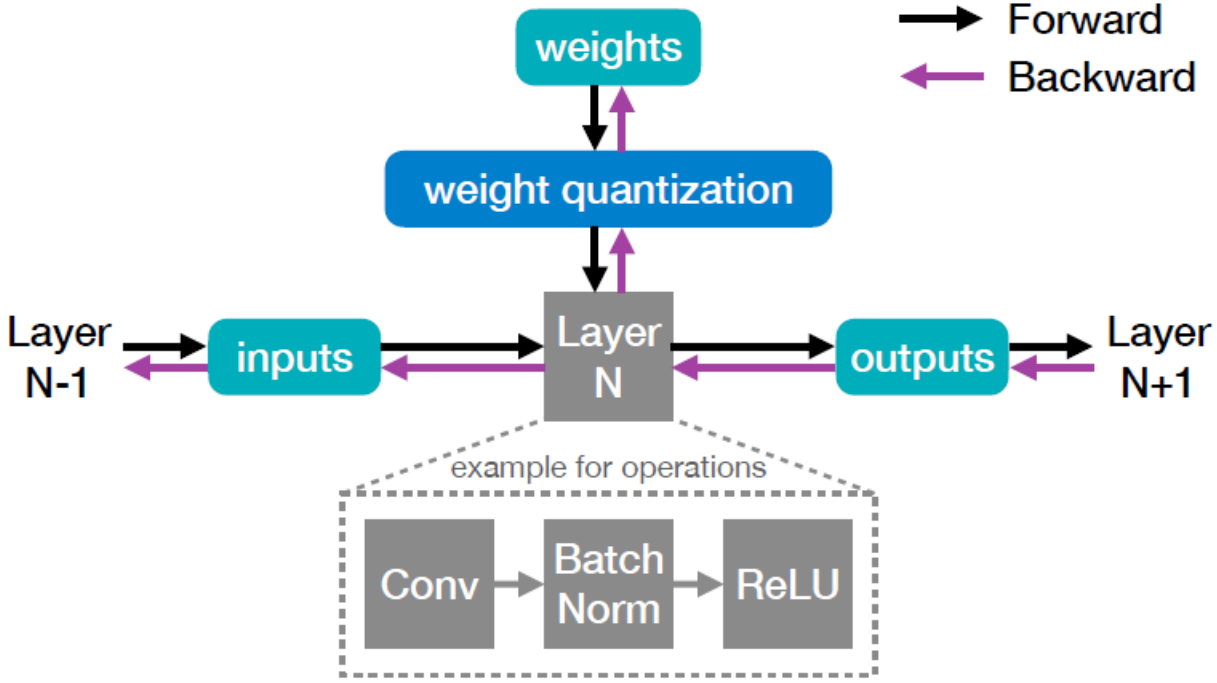
1. Costos de operaciones
2. Cuantificación
3. Cuantificación basada en K-means
4. Post-training quantization
5. Quantization-Aware Training

Quantization-Aware Training

Entrenar el modelo considerando la cuantificacion.

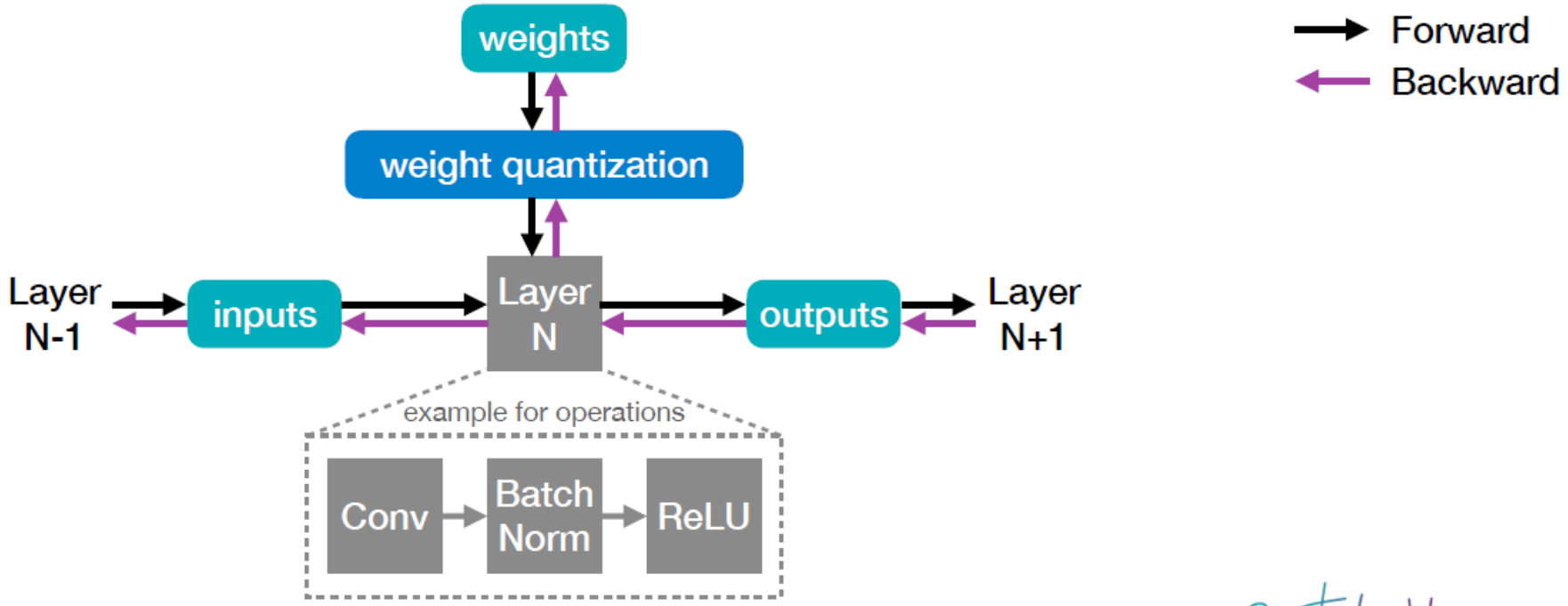
- Minimizar la perdida del acierto, especialmente cuantificaciones agresivas con 4 bits o menos.
- Usalmente, fine-tuning sobre un modelo pre-entrenado con punto flotante prove un major acierto que entrenando desde cero.





Quantization-Aware Training

- Una copia precisa de los pesos se mantiene durante el entrenamiento.
- Los pequeños gradientes son acumulados sin pérdida de precisión.
- Una vez el modelo es entrenado, solo los pesos cuantificados son usados para la inferencia.

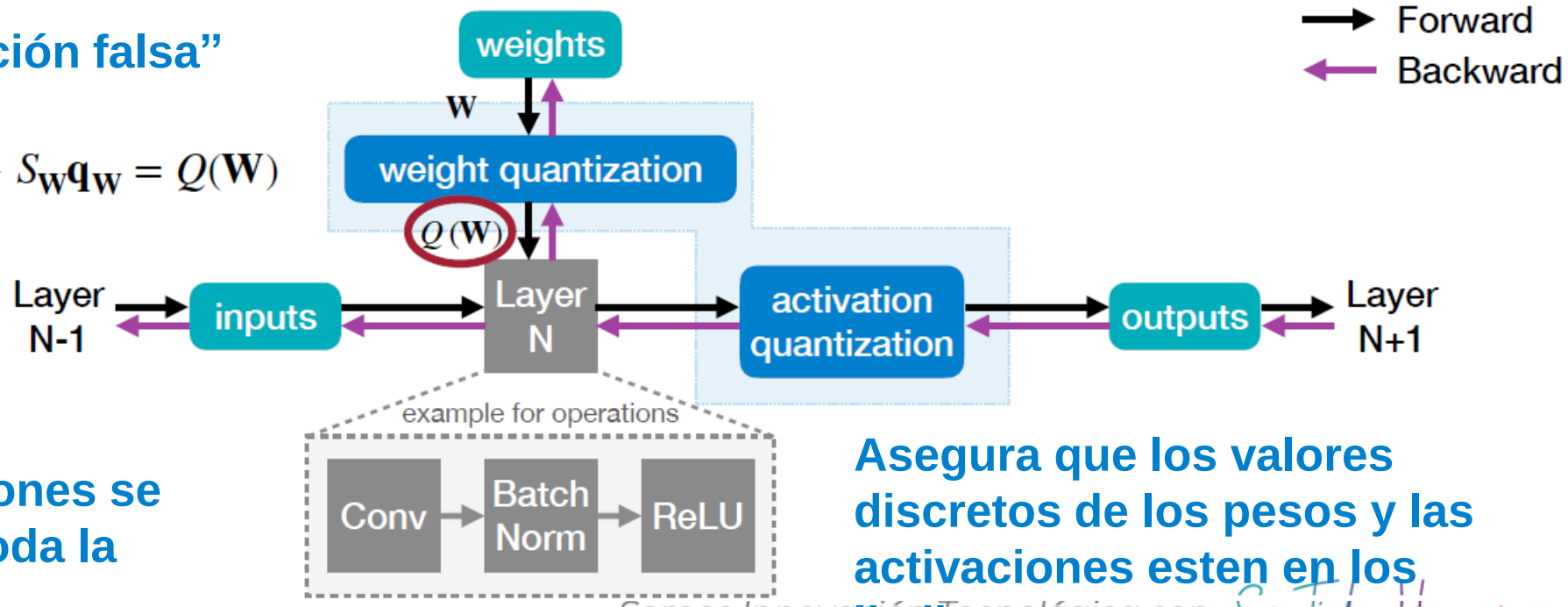


Quantization-Aware Training

- Una copia precisa de los pesos se mantiene durante el entrenamiento.
- Los pequeños gradientes son acumulados sin pérdida de precisión.
- Una vez el modelo es entrenado, solo los pesos cuantificados son usados para la inferencia.

“Simulada/Cuantificación falsa”

$$\mathbf{W} \rightarrow S_{\mathbf{W}}\mathbf{q}_{\mathbf{W}} = Q(\mathbf{W})$$



Estas operaciones se realizan con toda la precision

Asegura que los valores discretos de los pesos y las activaciones esten en los limites

Institución Universitaria
Reacreditada en Alta Calidad

Bibliografía

- Han, Efficient Deep Learning - Lecture 5, Quantization.

Institución Universitaria
Reacreditada en Alta Calidad

¡Gracias!

Somos Innovación Tecnológica con Sentido Humano

Alcaldía de Medellín