Training Cost Model for Neural Web

Witold Warchoł

September 22, 2025

1 Overview

This document provides a symbolic formula for estimating the training cost of the custom brain-inspired neural network architecture. The model integrates neuron updates, hierarchical memory, meta-cognition, imagination, social/emotional layers, and validation overhead.

2 Training Cost Formula

The effective training time is given by:

$$T_{\text{train}} = \kappa \cdot \frac{D \cdot E \cdot C_{\text{step}}}{H} \tag{1}$$

where:

- D = dataset size (number of samples)
- E = number of epochs
- C_{step} = compute cost per training step (operations)
- H = effective hardware throughput (FLOPs/s)
- κ = inefficiency multiplier due to kernel optimization and system overhead

3 Step Cost Expansion

The per-step compute cost can be decomposed as:

$$\begin{aligned} C_{\text{step}} \approx\ & N \cdot C_{\text{neuron}} \\ & + M \cdot C_{\text{mem}} \\ & + S \cdot (N \cdot C_{\text{neuron}}) \\ & + F \cdot (C_{\text{meta}} + C_{\text{identity}} + C_{\text{emotion}} + C_{\text{social}}) && (2) \\ C_{\text{step,eff}} =\ & V \cdot C_{\text{step}} && (3) \end{aligned}$$

where:

- N = total neuron count (proportional to parameter count, e.g. 3×10^8)
- C_{neuron} = operations per neuron update (typically ~ 20)
- M = number of active memory vectors
- C_{mem} = operations per memory update (50–200)
- S = number of imagination scenarios simulated per step
- F = frequency of meta/identity/emotion updates per input
- V = validation overhead factor (1.05–1.2)

4 Example Calculation

For a network with:

- $N = 3 \times 10^8$
- $M = 10^5$
- $C_{\text{neuron}} = 20$
- $C_{\text{mem}} = 100$
- $S = 3$, $F = 0.2$, $V = 1.1$
- $D = 10^6$ samples, $E = 3$

we obtain:

$$\begin{aligned} C_{\text{step}} &\approx (3 \times 10^8)(20) + (10^5)(100) + 3(3 \times 10^8)(20) && (4) \\ &\approx 2.4 \times 10^{10} \text{ ops} && (5) \end{aligned}$$

Total training compute:

$$C_{\text{train}} = D \cdot E \cdot C_{\text{step}} \approx 7.2 \times 10^{16} \text{ ops} \quad (6)$$

On an RTX 3060 ($H \approx 1.3 \times 10^{13}$ FLOPs/s):

$$T_{\text{train}} \approx \kappa \cdot \frac{7.2 \times 10^{16}}{1.3 \times 10^{13}} \approx \kappa \cdot 5.5 \times 10^3 \text{ seconds} \quad (7)$$

With $\kappa = 5$, this gives roughly 7.5 hours.

5 Interpretation

- Consumer GPUs (RTX 3060–3090) yield training times in the **hours** range.
- High-end GPUs (RTX 4090, A100) can reduce training to **tens of minutes**.
- Inefficiency factor κ depends strongly on kernel design and memory locality.