🧮 Let's Choose Dimensions:

Let's define:

- `k = 3` → 3 memory slots
- `d = 4` → hidden size = 4
- So:
  - `M ∈ ℝ^{3×4}` (memory matrix)
  - `h_z ∈ ℝ^{1×4}` (hidden vector from encoder)
  - `W ∈ ℝ^{3×4}`, `b ∈ ℝ^{3}`

1️⃣ Example Matrices

🔹 `h_z` : Hidden vector from encoder

plaintext Copy Edit

```
h_z = [[0.2, 0.4, 0.6, 0.8]] # shape: 1x4
```

🔸 `W` : Linear projection layer

plaintext Copy Edit

```
W = [[ 0.1, 0.0, 0.2, 0.3], # shape: 3x4 [-0.2, 0.5, -0.1, 0.4], [ 0.3, 0.2, 0.0, 0.1]]
```

🔸 `b` : Bias vector

plaintext Copy Edit

```
b = [0.1, 0.0, -0.2] # shape: 3
```

🔹 `M` : Memory matrix (latent factors)

plaintext Copy Edit

```
M = [[0.5, 0.1, 0.0, 0.3], # shape: 3x4 [0.2, 0.4, 0.1, 0.0], [0.3, 0.3, 0.3, 0.3]]
```

2 Step-by-Step Computation

Compute attention logits:

$$\text{logits} = W h_z^T + b$$

Let's compute each:

plaintext

Copy Edit

```
W × h_z.T = [ (0.1*0.2 + 0.0*0.4 + 0.2*0.6 + 0.3*0.8) = 0.02 + 0 + 0.12 + 0.24 = 0.38 (-0.2*0.2 + 0.5*0.4 + -0.1*0.6 + 0.4*0.8) = -0.04 + 0.2 - 0.06 + 0.32 = 0.42 (0.3*0.2 + 0.2*0.4 + 0.0*0.6 + 0.1*0.8) = 0.06 + 0.08 + 0 + 0.08 = 0.22 ] Add bias `b = [0.1, 0.0, -0.2]`: logits = [0.38+0.1, 0.42+0.0, 0.22-0.2] = [0.48, 0.42, 0.02]
```

Apply softmax:

$$\alpha = \text{softmax}([0.48, 0.42, 0.02]) \approx [0.375, 0.353, 0.272]$$

Compute weighted sum over memory:

$$z = \sum_{i=1}^{3} \alpha_i \cdot M_i$$

$$z = 0.375 * [0.5, 0.1, 0.0, 0.3] + 0.353 * [0.2, 0.4, 0.1, 0.0] + 0.272 * [0.3, 0.3, 0.3, 0.3]$$

Break it down:

plaintext

Copy Edit

```
= [0.1875, 0.0375, 0.000, 0.1125] + [0.0706, 0.1412, 0.0353, 0.0000] + [0.0816, 0.0816, 0.0816, 0.0816] = [0.3397, 0.2603, 0.1169, 0.1941]
```

Final Result:

$$z = [0.3397, 0.2603, 0.1169, 0.1941] \in \mathrm{R}^{1\times 4}$$

This is your **latent entailment memory representation** from memory `M`, selected using attention from `h_z`.