Day 4: Matrix Multiplication as Linear Transformation & Transpose in backpropagationagation

Goal. Understand how matrix multiplication represents linear transformations, the role of transpose, and why transposes appear in backpropagationagation.

1. Matrix Multiplication as Linear Transformation

Let $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. Then

$$y = Ax \in \mathbb{R}^m$$

maps an n-dimensional vector to an m-dimensional vector. Each column of A describes how a basis vector of $\mathbb{R}^n$ is transformed.

Interpretation. A encodes scaling, rotation, reflection, or projection — all linear transformations.

2. Composition of Transformations

If $B \in \mathbb{R}^{p \times m}$, then applying A then B corresponds to:

$$y = B(Ax) = (BA)x.$$

Thus, matrix multiplication corresponds to composition of transformations.

3. Transpose: Geometric View

For $A \in \mathbb{R}^{m \times n}$, the transpose $A^\top \in \mathbb{R}^{n \times m}$ satisfies

$$\langle Ax, y \rangle = \langle x, A^\top y \rangle \quad \forall x \in \mathbb{R}^n, y \in \mathbb{R}^m.$$

Meaning. $A^\top$ is the adjoint: it "moves" a transformation to the other side of the inner product. This is fundamental for gradients and backpropagationagation.

4. Why Transpose Appears in backpropagation

Suppose forward pass:

$$y = Ax.$$

Loss $L = L(y)$. Gradient wrt input x:

$$\nabla_x L = A^\top \nabla_y L.$$

So backpropagation through a linear layer requires multiplying by the transpose of the weight matrix.

This generalizes: transposes appear whenever gradients flow backward through linear transformations.

5. Worked Example

Let

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Forward:

$$y = Ax = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

Suppose gradient from above: $\nabla_y L = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$. backpropagation:

$$\nabla_x L = A^\top \nabla_y L = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

6. Why This Matters in AI

- **Neural Networks:** Every dense layer is $y = Wx + b$; backpropagation requires $W^\top$.
- **Embeddings:** Dot products use transposes implicitly ($q^\top k$).
- **Linear Algebra Foundations:** Understanding A vs. $A^\top$ gives intuition for forward/backward passes.

7. Mini-Exercises

1. Show that $(AB)^\top = B^\top A^\top$.
2. If A is orthogonal ($A^\top A = I$), prove that $\nabla_x L = A^\top \nabla_y L$ just rotates/reflects gradients without scaling.
3. In backpropagation, what would happen if we used A instead of $A^\top$?

Next

Day 5: Determinants, invertibility, and why singular matrices break models.