

Transformer Network - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs	1
1.1 Introduction	1
1.2 Data and Notation	1
1.3 Model Formulation	2
1.4 Training Objective (Autoregressive Likelihood)	3
1.5 Mathematical Properties and Proofs	3
1.6 Backpropagation Through Scaled Dot-Product Attention	3
1.7 Positional Encodings (Sinusoidal)	4
1.8 Computational Complexity	4
1.9 Algorithm (Encoder–Decoder Transformer)	5
1.10 Summary of Variables and Their Dimensions	5
1.11 Summary	5

1 Mathematical Derivations & Proofs

1.1 Introduction

The **Transformer** is a sequence model built from *self-attention* and *position-wise* feed-forward layers, organized in residual blocks with layer normalization. Unlike RNNs, it dispenses with recurrence and relies on *content-based addressing*: each position attends to a convex combination of all positions via learned *queries*, *keys*, and *values*. We derive scaled dot-product attention, multi-head attention, encoder/decoder blocks, the autoregressive training objective, and full backpropagation through attention with explicit gradients and dimensional checks.

1.2 Data and Notation

Let a tokenized input (source) sequence be $\{\mathbf{x}_t\}_{t=1}^{T_x}$ with $\mathbf{x}_t \in \{1, \dots, V\}$ (vocabulary size V), and a target sequence $\{\mathbf{y}_t\}_{t=1}^{T_y}$. Embeddings $\mathbf{E} \in \mathbb{R}^{V \times d_{\text{model}}}$ map indices to rows:

$$\mathbf{X} \in \mathbb{R}^{T_x \times d_{\text{model}}}, \quad \mathbf{X}_{t:} = \mathbf{E}_{\mathbf{x}_{t:}}, \quad \mathbf{Y}^{\text{in}} \in \mathbb{R}^{T_y \times d_{\text{model}}}, \quad \mathbf{Y}_{t:}^{\text{in}} = \mathbf{E}_{\mathbf{y}_{t-1:}} \text{ (teacher forcing).}$$

Add positional encodings $\mathbf{P} \in \mathbb{R}^{T_{\text{max}} \times d_{\text{model}}}$ (fixed sinusoidal or learned). We denote by bold capitals matrices and by bold lower-case row vectors (row-major convention).

Dimensions used throughout: d_{model} (channel width), d_k (per-head key/query dim), d_v (per-head value dim), H (# heads), d_{ff} (FFN hidden width).

1.3 Model Formulation

Scaled Dot-Product Attention (Single Head)

Given a sequence representation $\mathbf{S} \in \mathbb{R}^{T \times d_{\text{model}}}$, define linear projections

$$\mathbf{Q} = \mathbf{S}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{S}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{S}\mathbf{W}_V, \quad \mathbf{W}_Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, \mathbf{W}_K \in \mathbb{R}^{d_{\text{model}} \times d_k}, \mathbf{W}_V \in \mathbb{R}^{d_{\text{model}} \times d_v}.$$

Compute scores, apply (optional) mask $\mathbf{M} \in \mathbb{R}^{T \times T}$ with entries 0 (keep) or $-\infty$ (block), then row-wise softmax:

$$\mathbf{S}^{(\text{att})} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M} \in \mathbb{R}^{T \times T}, \quad (1)$$

$$\mathbf{A} = \text{softmax}_{\text{row}}(\mathbf{S}^{(\text{att})}) \in \mathbb{R}^{T \times T}, \quad \sum_{j=1}^T \mathbf{A}_{ij} = 1, \quad \mathbf{A}_{ij} \geq 0, \quad (2)$$

$$\mathbf{O} = \mathbf{AV} \in \mathbb{R}^{T \times d_v}. \quad (3)$$

Convex-combination property. For each row i , $\mathbf{O}_{i:} = \sum_j \mathbf{A}_{ij} \mathbf{V}_{j:}$ is a convex combination of value rows since $\mathbf{A}_{ij} \geq 0$ and sums to 1.

Multi-Head Attention

For heads $h = 1, \dots, H$, with per-head parameters $\mathbf{W}_Q^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_K^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_V^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_v}$, compute $\mathbf{O}^{(h)} = \text{Attn}(\mathbf{S}; \mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)}) \in \mathbb{R}^{T \times d_v}$ (with the same mask). Concatenate and project:

$$\text{MHA}(\mathbf{S}) = \text{Concat}(\mathbf{O}^{(1)}, \dots, \mathbf{O}^{(H)}) \mathbf{W}_O \in \mathbb{R}^{T \times d_{\text{model}}}, \quad \mathbf{W}_O \in \mathbb{R}^{(Hd_v) \times d_{\text{model}}}.$$

Position-wise Feed-Forward Network (FFN)

Apply the same two-layer MLP to each position independently:

$$\text{FFN}(\mathbf{U}) = \phi(\mathbf{U}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \quad \mathbf{W}_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}, \quad \mathbf{W}_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}},$$

with nonlinearity ϕ (e.g., ReLU or GELU).

Residual Blocks and Layer Normalization

We present the *Pre-LN* variant (widely used for stability); Post-LN equations are analogous. For an input \mathbf{U} to a sublayer $g(\cdot)$:

$$\text{SublayerOut} = \mathbf{U} + g(\text{LN}(\mathbf{U})), \quad \text{LN}(\mathbf{u}) = \frac{\mathbf{u} - \mu}{\sigma} \odot \gamma + \beta \quad (\text{per position}).$$

Encoder and Decoder Stacks

Encoder layer takes $\mathbf{E} \in \mathbb{R}^{T_x \times d_{\text{model}}}$:

$$\mathbf{U}_0 = \mathbf{X} + \mathbf{P}_{1:T_x}, \quad (4)$$

$$\mathbf{H}_\ell^{(1)} = \mathbf{H}_{\ell-1} + \text{MHA}(\text{LN}(\mathbf{H}_{\ell-1})), \quad (5)$$

$$\mathbf{H}_\ell = \mathbf{H}_\ell^{(1)} + \text{FFN}(\text{LN}(\mathbf{H}_\ell^{(1)})), \quad \ell = 1, \dots, L_e, \quad (6)$$

with $\mathbf{H}_0 = \mathbf{U}_0$ and final encoder output $\mathbf{H}^{\text{enc}} = \mathbf{H}_{L_e}$.

Decoder layer uses masked self-attention and encoder–decoder (cross) attention. Let $\mathbf{U}_0^{\text{dec}} = \mathbf{Y}^{\text{in}} + \mathbf{P}_{1:T_y}$. For layer $m = 1, \dots, L_d$, Masked self-attn:

$$\mathbf{Z}_m^{(1)} = \mathbf{Z}_{m-1} + \text{MHA}(\text{LN}(\mathbf{Z}_{m-1}); \mathbf{M}_{\text{causal}}), \quad (7)$$

Cross-attn (Q from decoder, K/V from encoder):

$$\mathbf{Z}_m^{(2)} = \mathbf{Z}_m^{(1)} + \text{MHA}\left(\text{LN}(\mathbf{Z}_m^{(1)}), \text{keys/values} = \text{LN}(\mathbf{H}^{\text{enc}})\right), \quad (8)$$

FFN:

$$\mathbf{Z}_m = \mathbf{Z}_m^{(2)} + \text{FFN}(\text{LN}(\mathbf{Z}_m^{(2)})), \quad (9)$$

with $\mathbf{Z}_0 = \mathbf{U}_0^{\text{dec}}$ and final decoder output $\mathbf{Z}^{\text{dec}} = \mathbf{Z}_{L_d}$. The causal mask $\mathbf{M}_{\text{causal}}$ sets (i, j) entries to $-\infty$ for $j > i$.

Readout. For language modeling, apply a linear head and softmax at each target step:

$$\mathbf{Z}_t \mapsto \mathbf{r}_t = \mathbf{Z}_t \mathbf{W}_{\text{lm}} + \mathbf{b}_{\text{lm}} \in \mathbb{R}^V, \quad \hat{\mathbf{p}}_t = \text{softmax}(\mathbf{r}_t),$$

(optionally *tying* $\mathbf{W}_{\text{lm}} = \mathbf{E}^{\top}$).

1.4 Training Objective (Autoregressive Likelihood)

Given targets $\mathbf{y}_{1:T_y}$, maximize the (teacher-forced) conditional log-likelihood

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T_y} \log \mathbb{P}_{\Theta}(Y_t = \mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{x}_{1:T_x}) = \sum_{t=1}^{T_y} -\text{CE}(\mathbf{y}_t, \hat{\mathbf{p}}_t),$$

i.e., minimize cross-entropy. Class or token weights can be applied if needed.

1.5 Mathematical Properties and Proofs

(P1) Causality under masking. Let $\mathbf{A} = \text{softmax}_{\text{row}}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}} + \mathbf{M}_{\text{causal}}\right)$. For row i , entries with $j > i$ receive score $-\infty$ and thus attention weight $\exp(-\infty)/\sum_{t \leq i} \exp(\cdot) = 0$. Hence $\mathbf{O}_{i:}$ depends only on $\{\mathbf{V}_{j:}\}_{j \leq i}$: the decoder is strictly autoregressive. ■

(P2) Scale factor $\frac{1}{\sqrt{d_k}}$. If pre-projection rows of \mathbf{S} are approximately zero-mean with unit variance and entries of $\mathbf{W}_Q, \mathbf{W}_K$ are independent with variance $1/d_{\text{model}}$, then for a fixed pair (i, j) , $\text{Var}((\mathbf{Q}\mathbf{K}^{\top})_{ij}) \approx d_k$. Dividing by $\sqrt{d_k}$ normalizes the score variance near 1, keeping softmax in a sensitive (non-saturated) regime and stabilizing gradients. ■

(P3) Attention as kernel smoother. Row i output $\mathbf{O}_{i:} = \sum_j \alpha_{ij} \mathbf{V}_{j:}$ with $\alpha_{ij} \propto \exp(\langle \mathbf{Q}_{i:}, \mathbf{K}_{j:} \rangle / \sqrt{d_k})$ is a Nadaraya–Watson estimator with exponential (dot-product) kernel, i.e., a learned, content-dependent smoother over $\{\mathbf{V}_{j:}\}$.

1.6 Backpropagation Through Scaled Dot-Product Attention

Consider one (unmasked) head with

$$\mathbf{S} = \frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}, \quad \mathbf{A} = \text{softmax}_{\text{row}}(\mathbf{S}), \quad \mathbf{O} = \mathbf{AV}.$$

Let $\mathbf{G}_O = \frac{\partial \mathcal{L}}{\partial \mathbf{O}} \in \mathbb{R}^{T \times d_v}$ be the upstream gradient.

Step 1: Through the value projection.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = \mathbf{A}^\top \mathbf{G}_O \in \mathbb{R}^{T \times d_v}, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{A}} = \mathbf{G}_O \mathbf{V}^\top \in \mathbb{R}^{T \times T}.$$

Step 2: Through the row-wise softmax. For each row i , with $\mathbf{a} = \mathbf{A}_{i:}$ and $\mathbf{g} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{A}}\right)_{i:}$, the Jacobian of softmax is $J(\mathbf{a}) = \text{Diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^\top$. Hence

$$\left(\frac{\partial \mathcal{L}}{\partial \mathbf{S}}\right)_{i:} = J(\mathbf{A}_{i:})^\top \left(\frac{\partial \mathcal{L}}{\partial \mathbf{A}}\right)_{i:} = (\mathbf{g} - (\mathbf{g}\mathbf{a}^\top)\mathbf{1}\mathbf{1}^\top) \odot \mathbf{a}.$$

In matrix form:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{A}} - ((\frac{\partial \mathcal{L}}{\partial \mathbf{A}}) \odot \mathbf{A}) \mathbf{1}\mathbf{1}^\top \right) \odot \mathbf{A},$$

where the subtraction applies row-wise using the row-sum scalar $\langle (\partial \mathcal{L}/\partial \mathbf{A})_{i:}, \mathbf{A}_{i:} \rangle$.

Step 3: Through the score matrix to \mathbf{Q}, \mathbf{K} . Since $\mathbf{S} = \frac{1}{\sqrt{d_k}} \mathbf{Q} \mathbf{K}^\top$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = \frac{1}{\sqrt{d_k}} \left(\frac{\partial \mathcal{L}}{\partial \mathbf{S}}\right) \mathbf{K} \in \mathbb{R}^{T \times d_k}, \quad (10)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{K}} = \frac{1}{\sqrt{d_k}} \left(\frac{\partial \mathcal{L}}{\partial \mathbf{S}}\right)^\top \mathbf{Q} \in \mathbb{R}^{T \times d_k}. \quad (11)$$

Step 4: Through linear projections to inputs and parameters. Given $\mathbf{Q} = \mathbf{S}_{\text{in}} \mathbf{W}_Q$, $\mathbf{K} = \mathbf{S}_{\text{in}} \mathbf{W}_K$, $\mathbf{V} = \mathbf{S}_{\text{in}} \mathbf{W}_V$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}_{\text{in}}} = \frac{\partial \mathcal{L}}{\partial \mathbf{Q}} \mathbf{W}_Q^\top + \frac{\partial \mathcal{L}}{\partial \mathbf{K}} \mathbf{W}_K^\top + \frac{\partial \mathcal{L}}{\partial \mathbf{V}} \mathbf{W}_V^\top, \quad (12)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_Q} = \mathbf{S}_{\text{in}}^\top \frac{\partial \mathcal{L}}{\partial \mathbf{Q}}, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{W}_K} = \mathbf{S}_{\text{in}}^\top \frac{\partial \mathcal{L}}{\partial \mathbf{K}}, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{W}_V} = \mathbf{S}_{\text{in}}^\top \frac{\partial \mathcal{L}}{\partial \mathbf{V}}. \quad (13)$$

For multi-head attention, apply these formulas per head and accumulate through concatenation and the output projection \mathbf{W}_O .

Correctness (shape and chain-rule check). Each multiplication respects the listed dimensions; the softmax Jacobian identity $J = \text{Diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^\top$ follows from $\partial a_k / \partial s_j = a_k(\mathbf{1}\{k=j\} - a_j)$; the remaining steps are standard reverse-mode rules for bilinear maps. ■

1.7 Positional Encodings (Sinusoidal)

A fixed, non-learned scheme defines

$$\mathbf{P}_{t,2i} = \sin\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right), \quad \mathbf{P}_{t,2i+1} = \cos\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right),$$

which allows the model to express relative offsets via linear projections since $\sin(a+b)$ and $\cos(a+b)$ are linear in $\sin a, \cos a$ for fixed b .

1.8 Computational Complexity

For length T and width d_{model} , attention per layer costs $\mathcal{O}(HT d_{\text{model}}(d_k+d_v))$ for projections and $\mathcal{O}(HT^2 d_k)$ for score/softmax/products; the T^2 term typically dominates. FFN cost is $\mathcal{O}(Td_{\text{model}} d_{\text{ff}})$.

1.9 Algorithm (Encoder–Decoder Transformer)

1. **Input:** source tokens $\mathbf{x}_{1:T_x}$, target tokens $\mathbf{y}_{1:T_y}$ (shifted for input), parameters Θ .
2. **Embed/position:** $\mathbf{X} = \text{Embed}(\mathbf{x}) + \mathbf{P}$, $\mathbf{Y}^{\text{in}} = \text{Embed}(\mathbf{y}_{\text{shift}}) + \mathbf{P}$.
3. **Encoder stack:** apply L_e Pre-LN layers: self-attention (all-to-all, unmasked) then FFN with residuals.
4. **Decoder stack:** apply L_d Pre-LN layers: (i) masked self-attention with $\mathbf{M}_{\text{causal}}$, (ii) cross-attention with keys/values from encoder, (iii) FFN; residuals around each sublayer.
5. **Readout & loss:** logits \mathbf{r}_t , probabilities $\hat{\mathbf{p}}_t = \text{softmax}(\mathbf{r}_t)$, loss $\mathcal{L} = \sum_t -\log \hat{\mathbf{p}}_t[\mathbf{y}_t]$.
6. **Backward:** backprop through readout, decoder, cross-attn, encoder, and embeddings; use the attention gradients above.
7. **Update:** apply optimizer (SGD/Adam/AdamW); optionally clip gradients.

1.10 Summary of Variables and Their Dimensions

- Tokens: $\mathbf{x}_t, \mathbf{y}_t \in \{1, \dots, V\}$; embeddings $\mathbf{E} \in \mathbb{R}^{V \times d_{\text{model}}}$.
- Positional encodings: $\mathbf{P} \in \mathbb{R}^{T_{\text{max}} \times d_{\text{model}}}$.
- Encoder inputs/outputs: $\mathbf{X} \in \mathbb{R}^{T_x \times d_{\text{model}}}$, $\mathbf{H}^{\text{enc}} \in \mathbb{R}^{T_x \times d_{\text{model}}}$.
- Decoder inputs/outputs: $\mathbf{Y}^{\text{in}} \in \mathbb{R}^{T_y \times d_{\text{model}}}$, $\mathbf{Z}^{\text{dec}} \in \mathbb{R}^{T_y \times d_{\text{model}}}$.
- Attention projections (per head): $\mathbf{W}_Q^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_K^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_V^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_v}$; output $\mathbf{W}_O \in \mathbb{R}^{(Hd_v) \times d_{\text{model}}}$.
- FFN: $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$, biases $\mathbf{b}_1, \mathbf{b}_2$.
- LayerNorm: per-channel gain/bias $\gamma, \beta \in \mathbb{R}^{d_{\text{model}}}$.
- Masks: $\mathbf{M}_{\text{causal}} \in \mathbb{R}^{T_y \times T_y}$ (0 below/diagonal, $-\infty$ above).
- Readout: $\mathbf{W}_{\text{lm}} \in \mathbb{R}^{d_{\text{model}} \times V}$, $\mathbf{b}_{\text{lm}} \in \mathbb{R}^V$.
- Gradients (single head): $\frac{\partial \mathcal{L}}{\partial \mathbf{O}} \in \mathbb{R}^{T \times d_v}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{A}} \in \mathbb{R}^{T \times T}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{S}} \in \mathbb{R}^{T \times T}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{Q}}, \frac{\partial \mathcal{L}}{\partial \mathbf{K}} \in \mathbb{R}^{T \times d_k}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{V}} \in \mathbb{R}^{T \times d_v}$.

1.11 Summary

From first principles, the Transformer applies (masked) *scaled dot-product attention* to form convex combinations of value vectors using content-similarity scores between queries and keys, with multi-head diversification, residual connections, and position-wise feed-forwards. An encoder builds contextual source representations; a decoder uses causally masked self-attention and cross-attention to condition next-token predictions on past targets and the source. We proved causality via masking, motivated score scaling, and gave exact backpropagation formulas through attention, completing a self-contained derivation consistent in notation and dimensions.