Full Transformer Forward + Backprop

Sunday, 26 October 2025 08:36

FORWARD

Transformer.forward():	SHAPES
$X = \text{Tokeniser}(x)$	(B, L)
$X_{one\_hot} = \text{one\_hot}(x)$	(B, L, d_{vocab})
$x_{emb} = x W_{emb}$	$(B, L, d_{vocab})(d_{vocab}, d_{model}) \rightarrow (B, L, d_{model})$
$X_{pos} = X_{emb} + \text{pos\_enc}(x)$	(B, L, d_{model})
$X = X_{pos}$	
FOR EACH LAYER:	
$x = \text{Layer.forward}(x)$	
$X_{norm\_out} = \text{RMSNorm.forward}(x)$	(B, L, d_{model})
$out = X_{norm\_out} W_{unemb}$	$(B, L, d_{model})(d_{model}, d_{vocab}) \rightarrow (B, L, d_{vocab})$
$pred = \text{softmax}(out)$	(B, L, d_{vocab})
	↑ Take softmax(pred[-1]) to get Probability distribution over next token

(B, L, d_model)

Layer.forward(x)	SHAPES
$X_{norm} = \text{RMSNorm.forward}(x)$	(B, L, d_{model})
$X_{multi\_att} = X + \text{MultiHead.forward}(X_{norm})$	$(B, L, d_{model}) + (B, L, d_{model})$
$X_{multi\_att\_norm} = \text{RMSNorm.forward}(X_{multi\_att})$	(B, L, d_{model})
$layer\_out = X_{multi\_att} + \text{FeedForward.forward}(X_{multi\_att\_norm})$	$(B, L, d_{model}) + (B, L, d_{model})$
return layer_out	

(B, L, d_model)

MultiHead.forward(x)	SHAPES
FOR EACH HEAD:	
$X_{head} = \text{Head.forward}(x)$	(B, L, d_v)
$X_{head\_out} = \text{Concat}(X_{head})$	$(B, L, h \cdot d_v)$
$out = X_{head\_out} \cdot W_o$	$(B, L, h \cdot d_v)(h \cdot d_v, d_{model}) \rightarrow (B, L, d_{model})$
return out	

(B, L, d_model)

FeedForward.forward(X_{FF})	SHAPES
$FF_1 = X_{FF} W_{FF_1} + b_{FF_1}$	$(B, L, d_{model})(d_{model}, d_{FF}) + (d_{FF}) \rightarrow (B, L, d_{FF})$

FeedForward forward (X_{FF})	SHAPES
$FF_1 = X_{FF} W_{FF_1} + b_{FF_1}$	$(B, L, d_{model})(d_{model}, d_{FF}) + (d_{FF}) \rightarrow (B, L, d_{FF})$
$FF_1' = ReLU(FF_1)$	(B, L, d_{FF})
$FF_2 = FF_1' W_{FF_2} + b_{FF_2}$	$(B, L, d_{FF})(d_{FF}, d_{model}) + (d_{model}) \rightarrow (B, L, d_{model})$
return FF_2	

(B, L, d_{model})

Head. forward (x)	SHAPES
$K = xW_K$	$(B, L, d_{model})(d_{model}, d_k) \rightarrow (B, L, d_k)$
$Q = xW_Q$	$(B, L, d_{model})(d_{model}, d_k) \rightarrow (B, L, d_k)$
$V = xW_V$	$(B, L, d_{model})(d_{model}, d_v) \rightarrow (B, L, d_v)$
$X_{att\_lin} = QK^T / \sqrt{d_k}$	$(B, L, d_k)(B, d_k, L) \rightarrow (B, L, L)$
$X_{att\_mask} = X_{att\_lin} + mask$	
$X_{att} = Softmax.forward(X_{att\_mask})$	(B, L, L)
$X_{attv} = X_{att} V$	$(B, L, L)(B, L, d_v) \rightarrow (B, L, d_v)$
return X_{attv}	

Softmax. forward (x):	
$y = e^x / \sum_{i=1}^{n} e^{x_i}$	(B, L, L)
return y	

RMSNorm. forward (x):	Whatever RMSNorm was init. to. Usually d_{model}
$rms\_norm = x / \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}$	(B, L, d_{rms})
$rms = rms\_norm * g$	$(B, L, d_{rms}) * (d_{rms})$
return rms	

BACKWARD

$L = CrossEntropyLoss(pred)$

Transformer.backward(Pred, labels):	SHAPES
$\frac{dL}{dout} = outputs - labels$	$(B,L,d_{vocab}) - (B,L,d_{vocab}) \rightarrow (B,L,d_{vocab})$
$\frac{dL}{dX_{norm\_out}} = \frac{dL}{dout} \cdot W_{unemb}^T$	$(B,L,d_{vocab})(d_{vocab},d_{model}) \rightarrow (B,L,d_{model})$
$\frac{dL}{dW_{unemb}} = X_{norm\_out}^T \cdot \frac{dL}{dout}$	$(B,d_{model},L)(B,L,d_{vocab}) \rightarrow (B,d_{model},d_{vocab})$
$\frac{dL}{layer\_out} = RMSNorm.backward(\frac{dL}{dX_{norm\_out}})$	(B,L,d_{model})
$Prev\_grad = \frac{dL}{layer\_out}$	
FOR EACH reversed(LAYER):	
$Prev\_grad = Layer.backward(Prev\_grad)$	(B,L,d_{model})
$\frac{dL}{dW_{emb}} = X_{one\_hot}^T \cdot Prev\_grad$	$(B,d_{vocab},L)(B,L,d_{model}) \rightarrow (B,d_{vocab},d_{model})$

(B,L,d_{model})

Layer.backward(Prev_grad):	SHAPES
$\frac{dL}{dX_{multi\_att\_norm}} = FeedForward.backward(Prev\_grad)$	(B,L,d_{model})
$\frac{dL}{dX_{multi\_att}} = RMSNorm.backward(\frac{dL}{dX_{multi\_att\_norm}}) + Prev\_grad$	$(B,L,d_{model}) + (B,L,d_{model})$
$\frac{dL}{dX_{norm}} = MultiHead.backward(\frac{dL}{dX_{multi\_att}})$	(B,L,d_{model})
$\frac{dL}{dx} = RMSNorm.backward(\frac{dL}{dX_{norm}}) + \frac{dL}{dX_{multi\_att}}$	$(B,L,d_{model}) + (B,L,d_{model})$
return $\frac{dL}{dx}$	

(B,L,d_{model})

FeedForward.backward(Prev_grad):	SHAPES
$\frac{dL}{dW_{H_2}} = FF_1^T \cdot Prev\_grad$	$(B,d_H,L)\ (B,L,d_{model}) \rightarrow (B,d_H,d_{model})$
$\frac{dL}{db_{H_2}} = Prev\_grad$	(B,L,d_{model})
$\frac{dL}{dFF_1} = Prev\_grad \cdot W_{H_2}^T$	$(B,L,d_{model})(d_{model},d_H) \rightarrow (B,L,d_H)$
$\frac{dL}{dFF_1} = ReLU.backward(\frac{dL}{dFF_1})$	(B,L,d_H)
$\frac{dL}{dW_{H_1}} = X_H^T \cdot \frac{dL}{dFF_1}$	$(B,d_{model},L)(B,L,d_H) \rightarrow (B,d_{model},d_H)$
$\frac{dL}{db_{H_1}} = \frac{dL}{dFF_1}$	(B,L,d_H)
$\frac{dL}{dX_H} = \frac{dL}{dFF_1} \cdot W_{H_1}^T$	$(B,L,d_H)(d_H,d_{model}) \rightarrow (B,L,d_{model})$

$\frac{dL}{dX_H} = \frac{dL}{dFF_1} \cdot W_{H_1}^T$ | $(B, L, d_H)(d_H, d_{model}) \rightarrow (B, L, d_{model})$

return $\frac{dL}{dX_H}$

Mult-Head. backward (prev_grad):	SHAPES
$\frac{dL}{dW_o} = X_{head\_out}^T \cdot prev\_grad$	$(B, h \cdot d_v, L)(B, L, d_{model}) \rightarrow (B, h \cdot d_v, d_{model})$
$\frac{dL}{dX_{head\_out}} = prev\_grad \cdot W_o^T$	$(B, L, d_{model})(d_{model}, h \cdot d_v) \rightarrow (B, L, h \cdot d_v)$
FOR EACH HEAD: (B, L, d_v)	
$\frac{dL}{dX} \mathrel{+}= Head.backward\left(\frac{dL}{dX_{head\_out}}[head]\right)$	(B, L, d_{model})
$\frac{dL}{dX} /= num\_heads$	
return $\frac{dL}{dX}$	

(B, L, d_v)

Head. backward (prev_grad):	SHAPES
$\frac{dL}{dV} = X_{att}^T \cdot prev\_grad$	$(B, L, L)(B, L, d_v) \rightarrow (B, L, d_v)$
$\frac{dL}{dX_{att}} = prev\_grad \cdot V^T$	$(B, L, d_v)(B, d_v, L) \rightarrow (B, L, L)$
$\frac{dL}{dX_{att\_mask}} = softmax.backward\left(\frac{dL}{dX_{att}}\right)$	(B, L, L)
$\frac{dL}{dX_{att\_1:n}} = mask\left(\frac{dL}{dX_{att\_mask}}\right)$	(B, L, L)
$\frac{dL}{dQ} = \frac{1}{\sqrt{d_k}} \frac{dL}{dX_{att\_1:n}} \cdot K$	$(B, L, L)(B, L, d_k) \rightarrow (B, L, d_k)$
$\frac{dL}{dK} = \frac{1}{\sqrt{d_k}} \frac{dL}{dX_{att\_1:n}}^T \cdot Q$	$(B, L, L)(B, L, d_k) \rightarrow (B, L, d_k)$
$\frac{dL}{dW_Q} = X^T \cdot \frac{dL}{dQ}$	$(B, d_{model}, L)(B, L, d_k) \rightarrow (B, d_{model}, d_k)$
$\frac{dL}{dW_K} = X^T \cdot \frac{dL}{dK}$	$(B, d_{model}, L)(B, L, d_k) \rightarrow (B, d_{model}, d_k)$
$\frac{dL}{dW_V} = X^T \cdot \frac{dL}{dV}$	$(B, d_{model}, L)(B, L, d_v) \rightarrow (B, d_{model}, d_v)$
$\frac{dL}{dX} = \left(\frac{dL}{dK} \cdot W_K^T\right) + \left(\frac{dL}{dQ} \cdot W_Q^T\right) + \left(\frac{dL}{dV} \cdot W_V^T\right) \cdot \frac{1}{3}$	(B, L, d_{model})
$(B, L, d_k)(d_k, d_{model})$ $(B, L, d_v)(d_v, d_{model})$	
$\rightarrow (B, L, d_{model})$ $\rightarrow (B, L, d_{model})$	

return $\frac{df}{dx}$

SoftMax. backward (Prev_grad):

Let $C = \sum_{i=1}^{n} e^{x_i}$, so $y_i = \frac{e^{x_i}}{C}$

Use quotient rule: $\frac{V\frac{du}{dx} - U\frac{dv}{dx}}{V^2}$

Each y_i depends on all x (due to $C = \sum_{i=1}^{n} e^{x_i}$)

∴ 2 cases:

Case 1: $i = k$

$U = e^{x_i}$, $\frac{du}{dx_i} = e^{x_i}$, $V = C$, $\frac{dv}{dx_i} = e^{x_i}$

$$\frac{dy_i}{dx_i} = \frac{Ce^{x_i} - e^{2x_i}}{C^2}$$
$$= \frac{e^{x_i}}{C} - \frac{e^{2x_i}}{C^2}$$
$$= y_i - y_i^2$$
$$= y_i(1 - y_i)$$

Case 2: $i \neq k$

$U = e^{x_i}$, $\frac{du}{dx_k} = 0$, $V = C$, $\frac{dv}{dx_k} = e^{x_k}$

$$\frac{dy_i}{dx_k} = \frac{(0 \cdot C) - e^{x_i}e^{x_k}}{C^2}$$
$$= -\frac{e^{x_i}e^{x_k}}{C^2}$$
$$= -\frac{e^{x_i}}{C}\frac{e^{x_k}}{C}$$
$$= -y_i y_k$$

$$\frac{dy_i}{dx_k} = \begin{cases} y_i(1-y_i) & \text{if } i = k \\ -y_i y_k & \text{if } i \neq k \end{cases}$$

RMS Norm. backward (Prev_grad):

$\frac{df}{dg}$ = rms_norm * Prev_grad

$(B, L, d_{rms}) * (B, L, d_{rms}) \rightarrow (B, L, d_{rms})$

$\overline{dg}$ = rms_norm * Prev_grad

$(B, L, d_{rms}) * (B, L, d_{rms}) \rightarrow (B, L, d_{rms})$

$\frac{dL}{dx}$:

Let $r = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$, so $rms_i = \frac{x_i}{r} * g_i$

$\frac{drms_i}{dx} = \frac{d}{dx}\left(\frac{x_i}{r}\right) * g_i$, but rms_i depends on all x due to r

$\therefore$ 2 cases:

Case 1: $i = k$:

$u = x_i$, $\frac{du}{dx_i} = 1$, $v = r$, $\frac{dv}{dx_i} = \frac{d}{dx_i}(r)$

$\rightarrow \frac{dr}{dx_i}$: $\frac{dr}{du}\frac{du}{dx_i}$, $u = \frac{1}{n}\sum_{i=1}^{n} x_i^2$, $r = u^{\frac{1}{2}}$

$= \frac{1}{2}u^{-\frac{1}{2}} \cdot \frac{2x_i}{n}$

$= \frac{1}{2r} \cdot \frac{2x_i}{n}$

$= \frac{x_i}{rn}$

$\frac{drms_i}{dx_i} = \frac{r - \frac{x_i^2}{rn}}{r^2}$

$= \left(\frac{1}{r} - \frac{x_i^2}{r^3 n}\right) * g_i$

Case 2: $i \neq k$

$u = x_i$, $\frac{du}{dx_k} = 0$, $v = r$, $\frac{dv}{dx_k} = \frac{x_k}{rn}$

$\frac{drms_i}{dx_k} = \frac{-x_i \frac{x_k}{rn}}{r^2}$

$= -\frac{x_i x_k}{r^3 n} * g_i$

$\frac{drms_i}{dx_k} = \begin{cases} \left(\frac{1}{r} - \frac{x_i^2}{r^3 n}\right) * g_i & \text{if } i = k \\ -\frac{x_i x_k}{r^3 n} * g_i & \text{if } i \neq k \end{cases}$

$\frac{dL}{dx_k} = \text{Prev-grad} \cdot \frac{drms_i}{dx_k}$

$(B, L, d_{rms})\ (B, L, d_{rms}, d_{rms}) \rightarrow (B, L, d_{rms})$