
Karpathy Inference for Llama-2

Ramesh Subramonian

December 31, 2025

Abstract

Step-by-step re-implementation of Karpathy code for llama2.

1 XXX

1.1 Conventions

If T is a table of dimensions $n_1 \times n_2$, then T_i is a vector of length n_2

1.2 Glossary

Abbreviation	C code	Comments
n_D	dim	
n_{HD}	hidden_dim	
n_{KVD}	n_kv_dim	$n_D \times n_{KVH} / n_H$
n_L	n_layers	
n_H	n_heads	
n_{KVH}	n_kv_heads	
n_V	vocab_size	
n_S	seq_len	
s_H	head_size	
n'_D	ispc_dim	

Table 1: Scalars: Mapping math notation to C code

Purpose	Abbreviation	C code	Dim1	Dim2	Dim3
	x	x	n_D		
	xb	xb	n_D		
	$xb2$	xb2	n_D		
	xhb	xhb	n_{HD}		
	$xhb2$	xhb2	n_{HD}		
	q	q	n_H	s_H	
	kc	key_cache	n_L	n_S	n_{KVD}
	vc	val_cache	n_L	n_S	n_{KVD}
	att	att	n_H	n_S	
	$logits$	logits	n_V		

Table 2: State: Mapping math notation to C code

Purpose	Abbreviation	C code	Dim1	Dim2	Dim3
Token Embedding	W_t	token embedding table	n_V	n_D	
Normalization	W_{att}	rms_att_weight	n_L	n_D	
Normalization	W_{ffn}	rms_ffn_weight	n_L	n_D	
XXX	W_{fin}	rms_final_weight	n_D		
Attention	W_q	w_q	n_L	n_D	$n_H \times s_H$
Attention	W_k	w_k	n_L	n_D	$n_{KVH} \times s_H$
Attention	W_v	w_v	n_L	n_D	$n_{KVH} \times s_H$
Attention	W_o	w_o	n_L	$n_H \times s_H$	n_D

Table 3: Weights: Mapping math notation to C code

2 Utilities

```

forward( $T, t, p$ )
   $T$  Transformer
   $t$  token
   $p$  pos
   $x \leftarrow W_t[t]$ 

  for each layer  $l$  do
    attention rmsnorm
     $xb \leftarrow rmsnorm(x, W_{ra}[l])$ 
    locate key and value in cache
     $k \leftarrow (S_{kc}[l][p])$ 
     $v \leftarrow (S_{vc}[l][p])$ 
     $q \leftarrow matmul(xb, W_{wq}[l], \text{dim}, \text{dim})$ 
    SOME JUNK
  endfor

```

Figure 1: Pseudo-code for forward

3 Forward

Arguments in Table 4

Argument	Type	Comments
T	Transformer	pointer
t	integer	token $0 \leq t < n_V$
p	integer	pos

Table 4: Arguments for forward

3.1 rmsnorm

Input is in Table 5. Output is o , float vector of length n

Argument	Type
x	float vector of length n
w	float vector of length n

Table 5: Arguments for rmsnorm

1. $\alpha = ((\sum_i x_i^2)/n) + \epsilon$
2. $o_i \leftarrow \frac{w_i \times x_i}{\alpha}$

3.2 softmax

Arguments in Table 6

Argument	Type
x	float vector of length n
n	integer

Table 6: Arguments for softmax

1. $m = \max_{i=0}^{i=n-1} x_i$
2. $\forall_{i=0}^{i=n-1} x_i \leftarrow e^{x_i - m}$
3. $s = \sum_{i=0}^{i=n-1} x_i$
4. $\forall_{i=0}^{i=n-1} x_i \leftarrow \frac{x_i}{s}$