CS 1678/2078 Homework Transformer

April 18, 2025

Abstract

In this assignment you will be implementing a transformer model and using it to predict the next character. To submit this assignment, upload a `.pdf` to Gradescope containing your responses to the questions below. You are required to use LaTeX for your write up. Upload a zip of your code to the Code portion of the assignment.

1 Building a Transformer

One of the goals of this assignment is for you to build your own transformer model in PyTorch. Specifically, this means creating a neural network with the following structure, where x is the input to the network of t tokens:

$$\begin{aligned} y &= \text{embedding}(x) + \text{pos\_enc}(1:t) \\ h^1 &= \text{transformer\_layer}(y) \\ h^2 &= \text{transformer\_layer}(h^1) \\ &\vdots \\ h^k &= \text{transformer\_layer}(h^{k-1}) \\ \hat{x} &= \text{linear\_layer}(\text{layer\_norm}(h^k)), \end{aligned}$$

where $\hat{x}$ represents a prediction of the next token for each of the input token, i.e., $\hat{x}_i$ is a prediction of x_{i+1}, embedding is an embedding layer which maps a token to a vector, pos_enc is provides the positional encoding of each token (this must have the same dimensions as the embedding layer output), transformer_layer is group of operations defined below that contain self-attention and a MLP and will be repeated k times. The last linear layer maps the embedding space back to the space of tokens to compute probabilities over the next token.

The transformer will be trained using the negative log-likelihood of the next token prediction. You should use the PyTorch class CrossEntropy to compute the loss. The operations of the transformer are described further below.

1.1 Embedding and Positional Encoding

The embedding layer is a mapping from a one-hot encoding of each token to a vector embedding. This layer is just a linear operation of xW, where $x \in [0,1]^d$ is a one hot vector representing one of d possible tokens and $W \in \mathbb{R}^{d \times n}$ maps that token to an n dimensional embedding space. For efficiency x never has to be explicitly represented by a one-hot vector and instead is just an integer. The layer returns the i^{th} row of the matrix if x represents the i^{th} possible token.

The positional encoding layer returns the positional encoding for each position of the sequence, i.e., if the sequence has length L, then the positional encoder returns a matrix of dimension $L \times n$, where n is the same dimension as the embedding layer. The positional encoding for the t position is

$$p(t) = \begin{bmatrix} \sin(w_1 t) \\ \cos(w_1 t) \\ \sin(w_2 t) \\ \cos(w_2 t) \\ \vdots \\ \sin(w_{n/2} t) \\ \cos(w_{n/2} t) \end{bmatrix},$$

where $w_i = \frac{1}{N^{2i/n}}$ is the frequency of the sine and cosine waves and N is a hyperparameter the controls the how low the frequencies go. Smaller frequencies means representing long periods of time. For numerical stability reasons the computation of w_i is broken up as follows

$$w_i = e^{2i \frac{-\ln N}{n}}.$$

Note that these position encoding do not have to be computed every time the network is run. Instead they can compute precomputed for some maximum sequence length, then only the L encodings that are needed for the input sequence are used.

Also notice that the positional encoding is added to the embedding layer representation for the character. Since this embedding layer will be optimized, it will be able to learn a feature representation that works with the positional encoding.

1.2 Transformer Layer

Using PyTorch implement your own transformer layer. This layer should will be similar to what we discussed in class and contain the following sequence of operations for the input x

$$y = x + \text{self\_attention}(\text{layer\_norm}(x))$$
$$z = y + \text{dropout}(\text{MLP}(\text{layer\_norm}(y))),$$

where self_attention is multi-head self-attention, layer_norm normalizes the activations of the hidden units, MLP is a MLP perception with one hidden layer and ReLU activation function, and dropout is a dropout layer that randomly sets layer outputs to zero with probability $p = 0.2$. We have not discussed dropout so far in the course, but it is a regularization method that helps prevent overfitting in neural networks. Note that all of these observations preserve the width of the network, i.e., the dimensions of x, y, and z are the same. For each of the components above you may use the built in pytorch classes and functions, but you must write your own TransformerLayer class, which brings these all together. It is important to read the pytorch documentation for each operation to make sure you have the correct dimensions and inputs for each operation.

2 Language Modeling

You will be training the transformer model on a next token prediction task. Specifically we will be using the tiny Shakespeare dataset which is a collection of scripts from Shakespeare's plays. For this part of the assignment you will need to parse the entire text file and find all the unique characters, build a dictionary that maps a character to an integer representing one of the characters, and build a dictionary to reverse that mapping, i.e., map from an integer to a character. The last dictionary is used to map model outputs to generate text.

Since it is not possible to train on the whole dataset at once, you will also need to create a data sampler that will sample a mini-batch of subsequences of length L from the dataset. The data sampler will also return the sequence of targets which is the same first sampled sequence, but shifted to the right by one, i.e., the input data is $x_{t:t+L}$ and the targets are $x_{t+1:t+L+1}$. For simplicity of the model we will assume all subsets are of length L, which means if the data set has T characters in it, the last possible subsequence that can be chosen starts at position $T - L - 1$.

3 Questions

Find the best hyperparameters that you can to minimize the validation loss. For non-small models it will be computationally expensive to train a these models on a laptop. If you have access to a GPU I highly recommend using it for this assignment. One way to get access to a GPU is to use Google's colab. You will be given a small amount of GPU credits to test your code on if you haven't used it before. I recommend testing your code locally on your own computer then trying it out on the cloud or other resources. If you do not have GPU access then you can keep your model size small. The hyperparameters below should be runable on a modern laptops 60 minutes. These hyperparameters produced an ok model that mostly produced correct looking words, but it was not very coherent.

Context Size	Embedding Size	Number of Transformer Layers	Number of heads
32	64	4	4

Try playing around with these parameters to see which ones impact the model's ability to produce good look samples the most. After finding the best set of hyperparameters train the model again without the positional encoding. Compare the two models in terms of their performance, ability to produce good samples, and examine how the attention mechanism places higher or lower weight on different aspects. To do this you will need to find "prompts" or initial context vectors that the model will then use to generate a sequence of tokens. Find prompts for which both models work well and do not work well.

1. List the hyperparameter and the best validation losses for both models : Answer : Although I played for hours with hyperparameter and even made some edit like changing the dropout ration or regularization of Adam optimizer, the model still show over fitting that the validation loss is larger than the train loss.

Context Size	256
Embedding Size	256
Number of Transformer Layers	8
Number of heads	8
width of MLP	1024
batch size	256
N	512
validation loss with positional encoding (min(vlosses))	3.495896
train loss with positional encoding (last epoch)	0.95375
validation loss without positional encoding (min(vlosses))	3.59661
train loss without positional encoding (last epoch)	1.11338

2. Provide prompts and model outputs that make each model perform good and bad. Limit the model generated response to 500 characters.

   Answer: I requested my advisor, and I was able to get access for a few hours to my resrearch lab gpus and train the models. generally , to my understanding, my trained model specifically the p.e model showed (ok) performance, considering all the limitation and simplicity of the mdoel . it is generally create some dialogues that may be meaningful in context, or may not in some cases. it also shows some understanding of part of speech.

   Positional encoding doing well:

   Answer: (as it is seen, with this prompt, it clearly make a dialogue with Clifford and other person it shows that the continuation of the word are good and is a evidence of positional encoding works. although some nonsense exist. Words like “bailian” or “prescribe” (even if slightly off) and constructions like “Wi wish here!” or “We will barr’d” show it learned the flavor of early english language. the model without positional encoding still generates dialogue-like text, but it produced some non-sense words like (possighnible) or (rap-which) and it it struggles to maintain longer-range consistency.

   Prompt: ’CLIFFORD:’

   model with position encoding doing well here (generated text): (”CLIFFORD: I would be made account.
   DUKE VINCENTIO: Wi wish here! my wife, but mine unfold To call myself; you are a bailian: the gods Will not make me all the men to hear their sighs; Not strike to be begin them: I’ll ry, in progettive Against the senated.
   LUCIO: Well said, girl.
   DUKE VINCENTIO: So: It is a prescribe, Your painting, sir.
   LUCIO: That is the new begins awake: the time in heaven, We will barr’d, but in the goose, no hand think would hold bestowom untimely the land What’s to deny how”)

   model without position encoding here (generated text) :

   (”CLIFFORD: What is this possighnible! York and the way And rap-which his to-day; which he his proper Lives and again: what he shower you his?
   ROMEO: Man, every hath angry like I throw, That shame the new for our away?
   FRIAR LAURENCE: I will not deper, my grace lord; If I should as said her was the times o’ it Than the body is a from the Benek’t. And therefore York’s love’s to ’anjoin’ there, And bear Saint King Richard Deppuliy, O his, poor leave had so noble forfeiths, Which his manConted that holl’s d ”)

   Positional Encoding not doing well: (I will use prompt containing words that are modern and out of distribution of the training data) : as it is seen, the model is not working well and the sentence does not have context about tthe prompt at all.

Prompt: ¡computer television¿

model with position encoding doing not well here (generated text :)

("computer television?

BRUTUS: Seeing Romeo, sir, you well changed you guest: You have not done of more divided to be Even the downributation: see you shall not last, Shame her brings to the duke tongue to court, But ministers sweath Prince Oxford, None of a dish's eyes have disturbed of deserved, As for all that gazes the held off their eyes; Which our said you misback'd prateglance!

GLOUCESTER: His body much believe stay the hour lives; More rivers in our puls on the sun: Come, we queen, all hail the gentleman. I cannot seem no other King County. 'I'll not yielded in any merit, that parchmets appoint, Let's shame for the war's off here for meats it. Away! a with power to one!

ISABELLA: It is the readiest way it wills speed.

LUCIO: Will miss it edle. The lives true; where was notice makes the corrects? what was this?

ISABELLA: It but many do we fair lord. ")

Same prompt for model without position encoding here (generated text) : (nonsense first sentences)
(computer television my scool's name, And did to my heart more than my brother. And by this, I answer yes, Escalus, by Catesby: FareXloxand, but answer to his cause and the shepherd wounds, to rich seveme much for thy father's, Cannot from youth, by owe depute me not. But I safelly, you were gone: for you hear Yet aupide of full of your daughters, as the does, blest you to your face, if, it will please it distanced; for three comes if my choose, outward our drown's death, never my stand daughter.)

Model without Positional Encoding doing well:

Prompt: ¡you cannot¿

Place prompt for model without position encoding here (generated text) :
(you cannot.

BENVOLIZELIO: It is all that know you are what you.

SICINIUS: I will you all painted of it what your good Where you hose chances?

BRUTUS: Not other has been made! The people white the voice insture The causes of his own: I am hasten'd tongue Snothes campare unto your highness' his contract, And thus ere but even Elband's rosember'd, Being with his walk words and the king: His noble to be purse out of a glass. My words, hath make his encounter's vile That have slack'd by the chafter of my cu)

model with position encoding here (generated text)
(" you cannot is bear. I'll to try them book of with with a wife: I'll task bands one sea that through mourning now fortunes shall echievily some other bird. My dire enmity, nurse, one of nature, A very power in a woman's cheirs and land To make my coffed hold me the hath seen: Thy need had in hath broad boar, and therein: I can kill'd my lady-hardening slain; And do I fight to hold zeat here my ladies King Richard and lost their hearts to love. 3 KING HENRY VI

GLOUCESTER: Now, my Lord Hastings all that Edw ")

Model without Positional Encoding doing not well: (nonsense sentence)

Prompt: ¡hydrogen elephant ¿

Place prompt for model without position encoding here (generated text) :
(hydrogen elephant itself to rash.

HENRY BOLINGBROKE: The hath got of him of the state do't?

DUKE OF YORK: By possession, the thought irumner the male; For I did have done frosame their shields.

DUKE OF AUMERLE: She was and hang to London that all influed blaste! Are 'Verily now have was but that I insue, And this ere gross proud his worth unto there.

KING RICHARD II: Madam, for you as I were reprieve Warwick, And very every now their good commanded freely A given you against your hands, even I, Meneaing the t)

Same prompt for model with position encoding here (generated text) :

(" hydrogen elephant in crossing the insisterhood, will to cheek him place in him. If he not the privilege world's helms? why would they none us, mother? from returning from thence? see, and hath single comes hose found that follow, And to aunntiment it, or woman true.

PRINCE: I shall seek that have spent mad soft error In his necessity of his friends With noble hatch'd than that shame when he made Of Rushrald he marked? how false Montagus?

CAPULET: Sir, if it do it, it is not so. He was a big tragedate and trift:")

3. Provide the plots of the average attention weights for both models with and without positional encoding. You can use the provided models to answer the following questions.

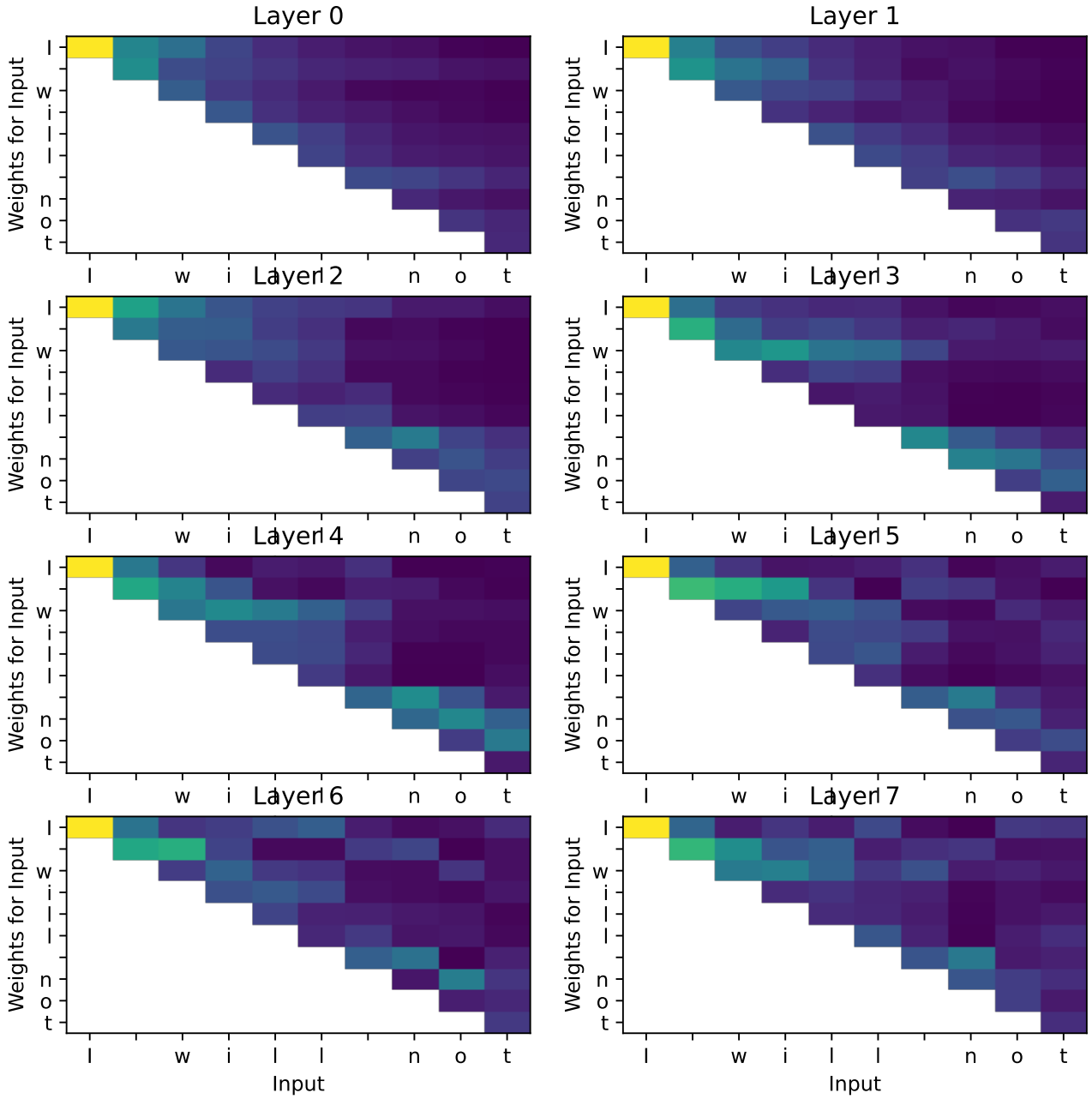
   (a) What do trends do you notice in the attention weights? Do the lower level layers capture different properties than the upper level? Are there any intuitive connections you can see in the pattern of attention weights? Give specific examples

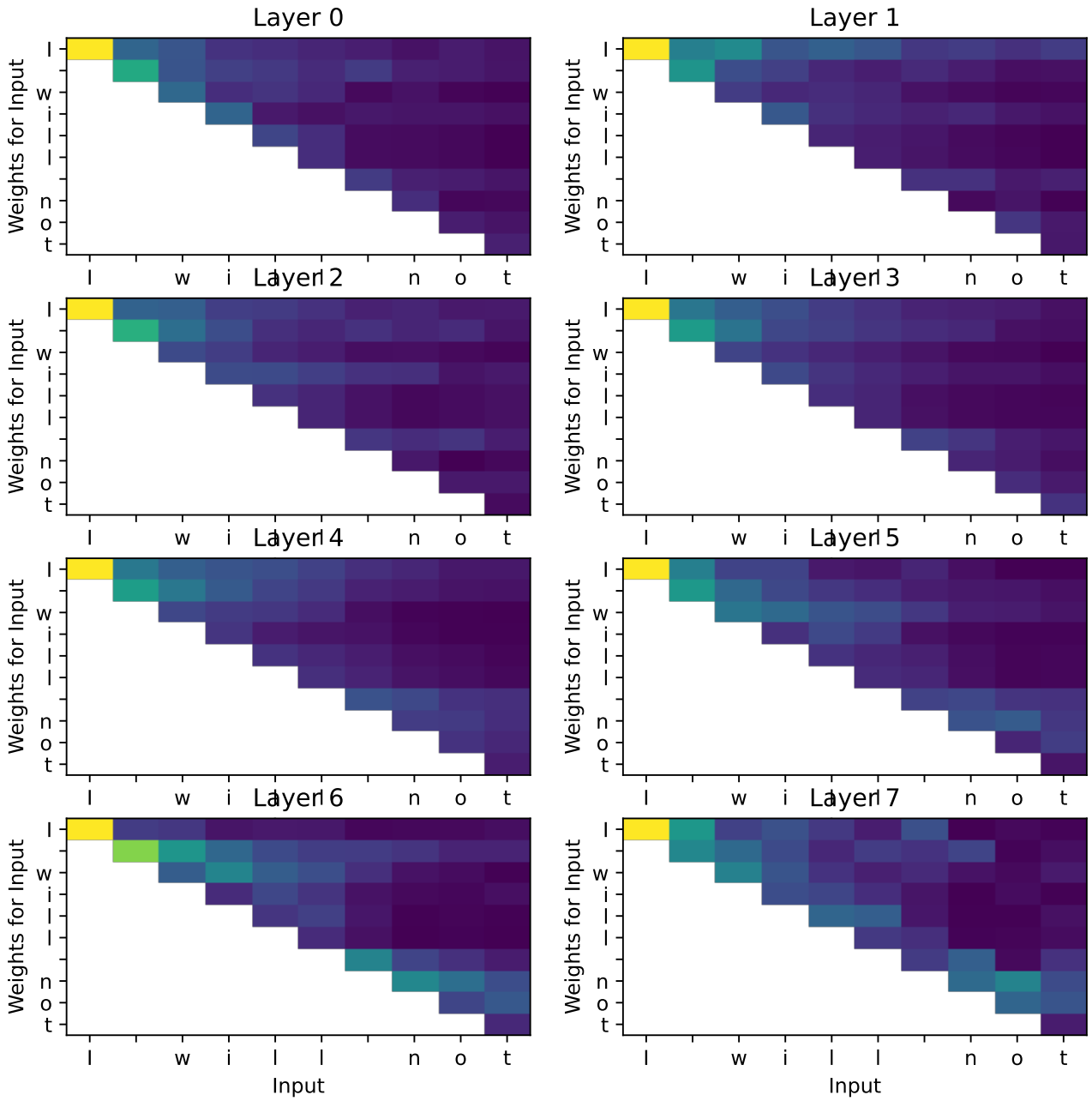
   Answer:based on my trained models: low level layers (layer 0) a very bright diagonal: each token gives attention almost entirely to itself (the (i,i) entries are brightest). also some of the off-diagonal tokens give some attention to the neighbors (layer 1), like space to the other other space, and I attend w and i in (will). i can summarize as sharp self attention and a small shift to neighbors.
   the big model: quantitatively shows the same pattern, but is show a more clear gradual fading in later layers , and also shows more distance atention, such as w in will to second l, and I to the (ll) in layer 2, and also space tot eh second space in layer 5 of the big model. my general understanding based on my text book and lecture reading is that the innital layers typically learns local or immediate patterns, and later layers displays long range dependecies or informations.

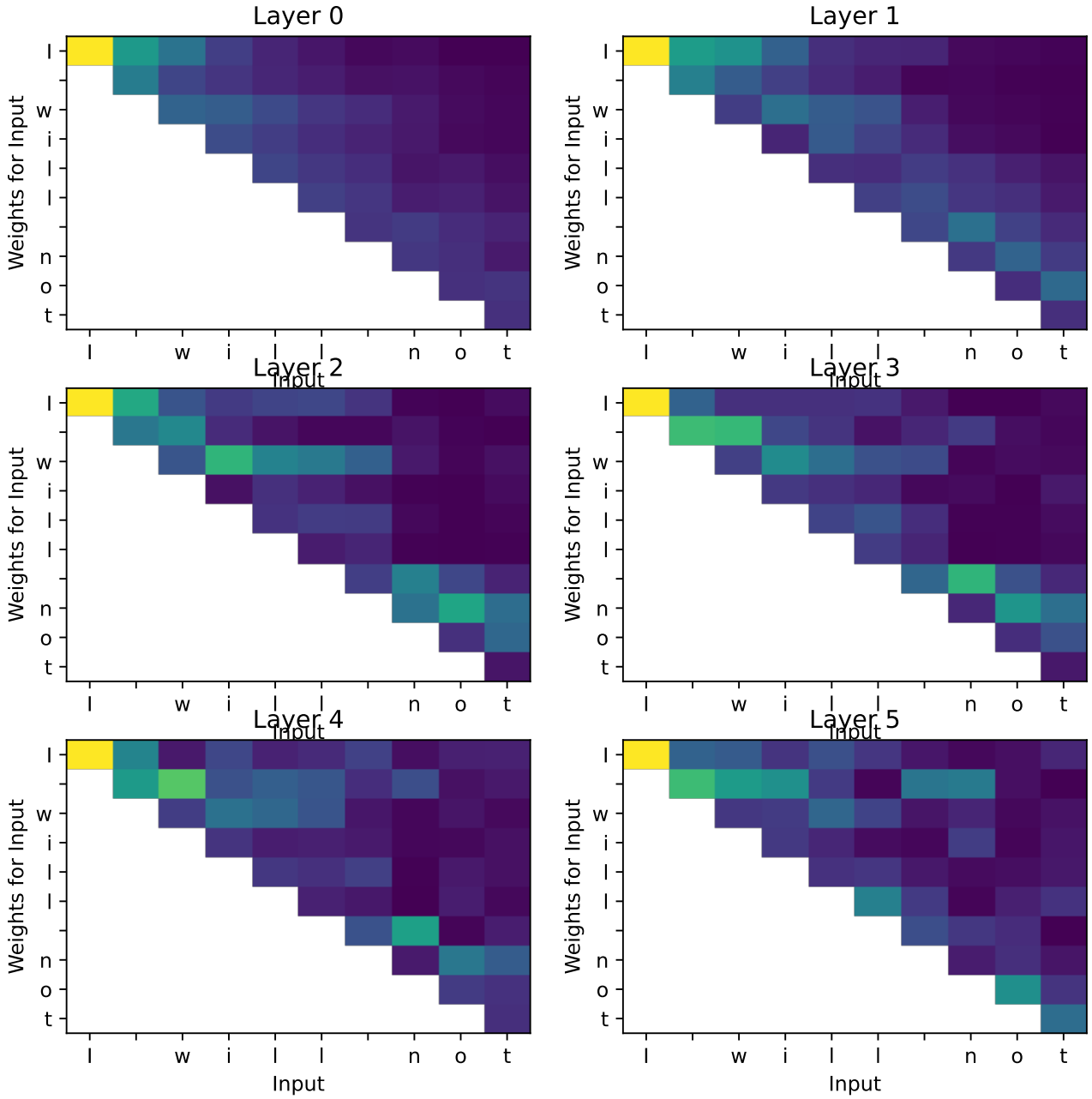
   (b) Plot the averaged attention weights of the model without the positional encoding. Is it any different than the model with the positional encoding? Reference specifics in the figure.

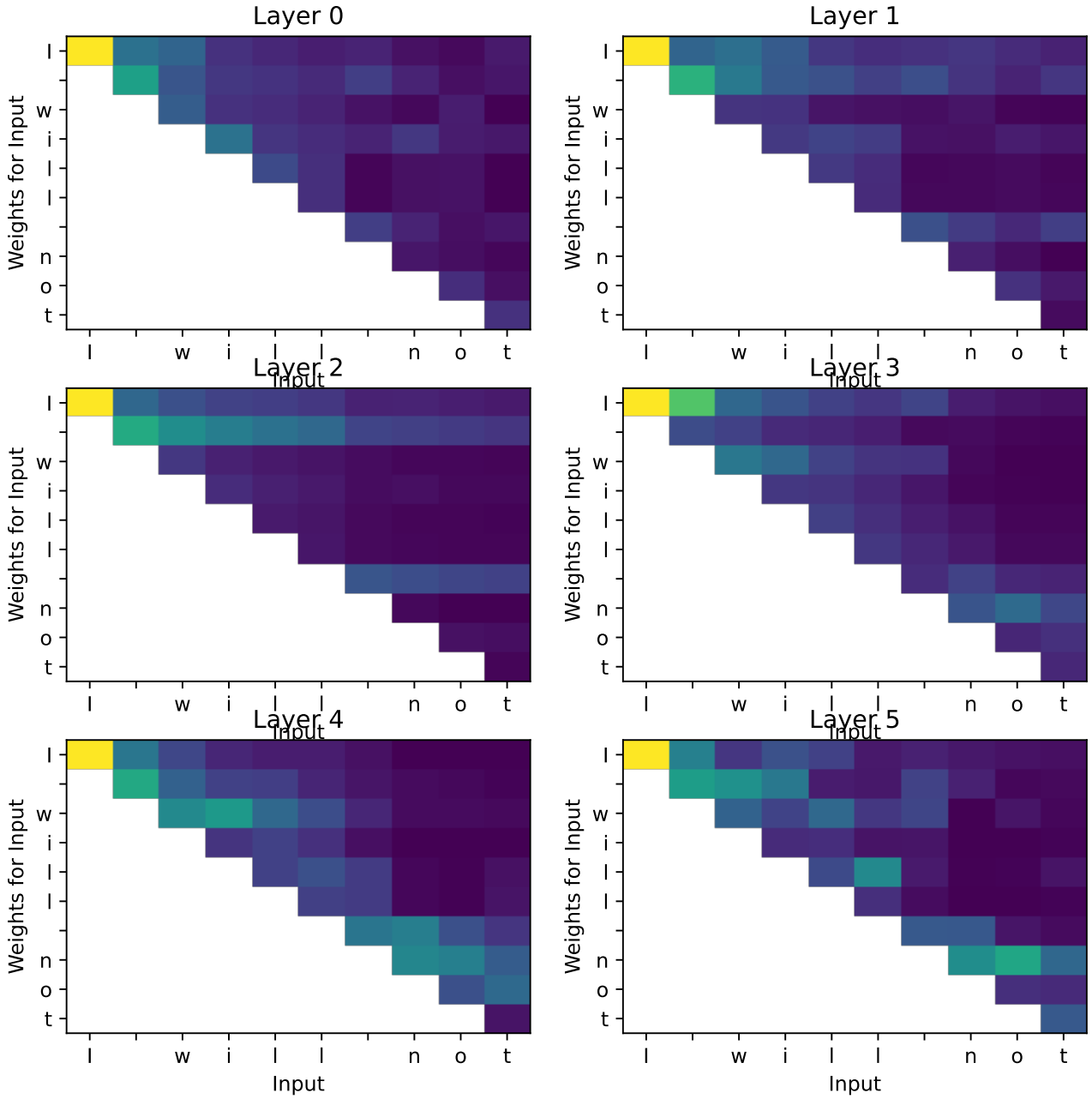
   Answer: No Positional Encoding (Layer0) the diagonal is still the brightest, but its already much blurrier—offdiagonals (positions 2–4 away) are almost as strong as the immediate neighbor. the no P.E model aslo shows less layer wise evolution: all eight layers look very similar to each other. Theres no systematic widening of the diagonal in higher layers—each layer just re-learns the same local bias. by looking a the heat map of the big model (non p.e) , similar patterns are better understood : layer0 No PE (big): A faint diagonal, but the peak at "self" tokens is barely brighter than off-diagonals— attention is almost uniformly disperesed. across all 10 tokens. (for exampel, (i) give relatively high attention to (n) in layer 0. With PE (big or small): A sharp, bright diagonal and clear first off-diagonal, showing the model strongly prefers "look at myself" or "look one step of back to the neighbors." also in no p.e the heatmaps remain noisy and flat small green patches such as tokens "n" or "o" in Layers 4 - 5) are random heads picking up random co-occurrences, not a learned positional pattern.

Replace the image files with the ones for your own project. Then delete this line.


Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Weights for Input
Input
l
w
i
l
l
n
o
t


Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Weights for Input
Input
l
w
i
l
l
n
o
t


Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Weights for Input
Input
I
w
i
l
l
n
o
t


Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Weights for Input
Input
I
w
i
l
l
n
o
t