Learning Correlation between 'characters' by a nano-GPT

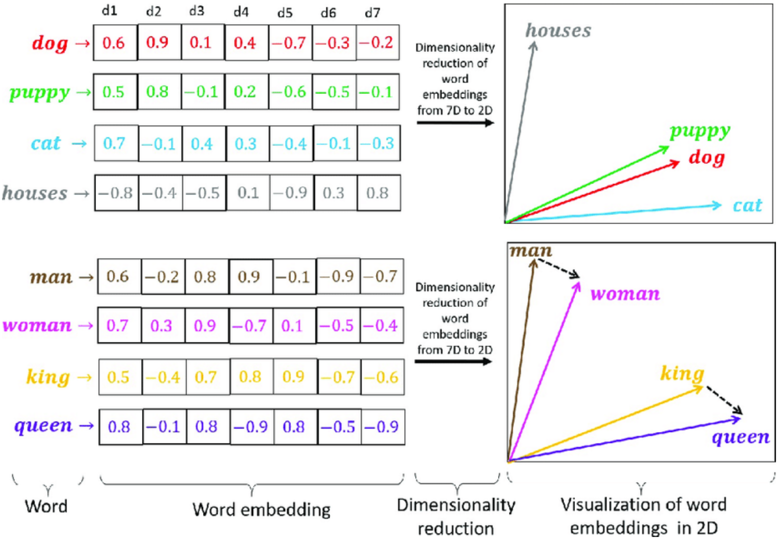
Ouraman Hajizadeh

May, 2024
ECT* Trento

Overview

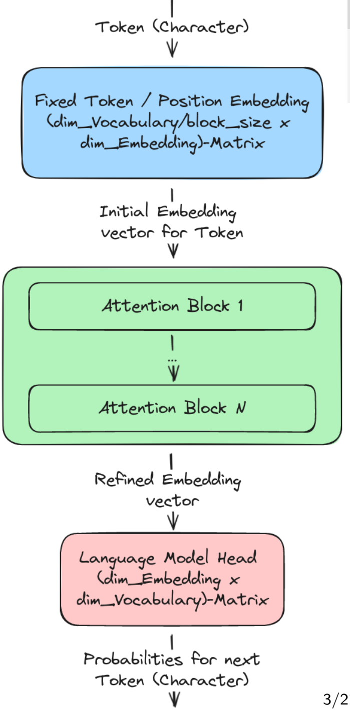
- A language model that takes sequence of tokens (words or characters) as input and returns another sequence:
  - Y, Yo, You, ,
  - Y, Ye, Yet, ,
  - Y, Yo, You, Your
- The input is a chunk of text and target is from the same text, one token shifted.
  - Input: You too Brutus
  - Target: ou too Brutus?
  - Prediction: Next Possible token probabilities: P(?), P(!),..

Vector Representation of Tokens



Overview of the Transformer

- Token and position embedding: Vectorization: each token is replaced by a vector
- Attention blocks: How the embedding of a token depends on the previous tokens
- Language model Head: Transform the embedding vector to logits for the next token



Token and Position Embedding

- Each character (token) is mapped to a vector of dimension C
  - Cicero $\rightarrow C_1, C_2, C_8, C_3, C_7, C_4$
  - Clock $\rightarrow C_1, C_5, C_4, C_8, C_6$
- The positions encoded in another vector with the same dimension C, independent of character
  - Cicero $\rightarrow P_1, P_2, P_3, P_4, P_5, P_6$
  - Clock $\rightarrow P_1, P_2, P_3, P_4, P_5$
- Output: sum of token and position embeddings for each character
  - Cicero $\rightarrow C_1 + P_1, C_2 + P_2, \ldots,$
  - Clock $\rightarrow C_1 + P_1, C_5 + P_2, \ldots,$

Attention block

- The output of Embedding layers: TxC matrix
  - T: number of character in one chunck of data (temporal dimension)
  - C: embedding dimension of each character
- Context dependent representation of characters is learnt through attention mechanism
- Each block conatins one or more **Attention Head**
- A token looks at the previous tokens from different angles: Attention Heads
- Each head reduce the embedding dimensionality $C \to h$
- Final output of a block: concatenated outputs of each attention head $Txh \to TxC$
- This will be input to other attention blocks with same properties

Attention head

- The output of Embedding layers: TxC matrix : $\mathbf{X} = [x_1, ..., x_T]$, $dim(x_i) = C$
- Attention is encoded in Key , Query and Value: $KQV = \{K_{h,C},\ Q_{h,C},\ V_{h,C}\}$
- $h = \frac{C}{n_{heads}}$: head size: number of features relevant to grasp a pattern
- $A(\mathbf{X}) = \mathbf{A}$, $\mathbf{A} = [a_1, ..., a_T]$, $dim(a_i) = h$, A in KQV
- $W = \mathbf{Q} \cdot \mathbf{K^T}$, $W_{i,j} = 0$ for $i < j$ Attends only to prev. tokens
- $W = softmax(W)$
- $W \cdot \mathbf{V} \sim [v_1,\ \sigma(q_2 k_1) v_1 + v_2,\ \sigma(q_3 k_1) v_1 + \sigma(q_3 k_2) v_2 + v_3, ...]$

Language Model Head

- After finding context dependent representation of a sequence, a linear transformation gives the logits for each token
  $logits = linear(dim(vocab), C), C = dim(embedding)$
- softmax (logits): probability of each possible token

Generated texts for different number of heads and layers

- Next token is generated by only knowing the previous token
- Given the probabilities of next token for each character a character is sampled form this distribution
- (0,0) : *llisous ching st, ouso whe gresindgome'd m irs mittherd inde ariz. KI s f m s'? The ce, prert ke*
- (1,1): *ringh const be delendeattes, Anty whow constse, pargee anot hissay hartre's then a cons, soe hear fi*
- (2,4): *Warwick, let's thy mista nopsted,Where should a worse found, a pale I may Of shall hounded me.*

Qualitative Comparison

- Some higher level properties are learned by all models
  - no special character seen in the generated text
  - average word length approaches English words'
  - number of vowels and consonants in a word approaches English words'
- Only models with enough complexity
  - generate mainly English words
  - are close to capturing Grammar

Quantifying Context Encoding

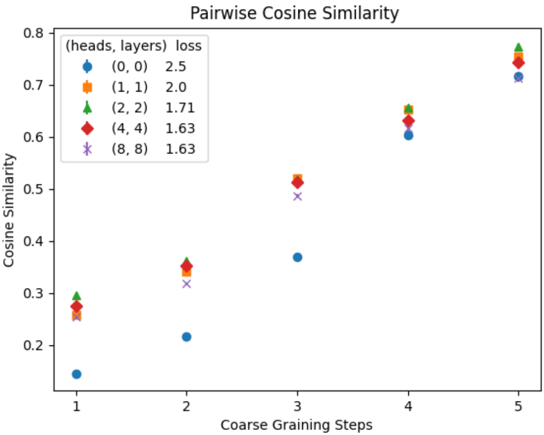
- Each token is represented by a vector
- Through Attention it became aware of tokens before
- Cosine between two vectors can be a measure of their similarity
- Correlation between two charactert embedding could reflect how relevant they are to onenother in a given context
- The output of the Transformer blocks is the subject of the following studies

Cosine Similarity

Cosine Similarity between characters can be a measure of learning patterns in the text

- Cosine similarity of neighboring characters ..., $Cos(T_i, T_{i+1})$ and $Cos(T_{i+2}, T_{i+3}), ..$ calculated
- Averaged over the sequence and all the batches
- Sequence is coarse grained: $T' = \frac{T}{n}$, n: coarse graining step
- The above is repeated for n = 1,..,5

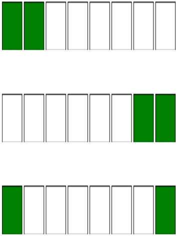
Cosine Similarity

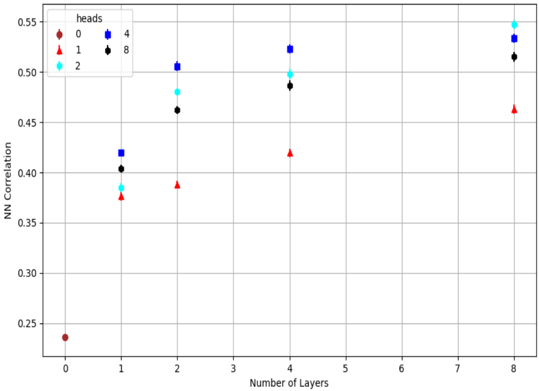


- All Models converge in higher steps
- Discrepancy between Attention based models in coarser sequence

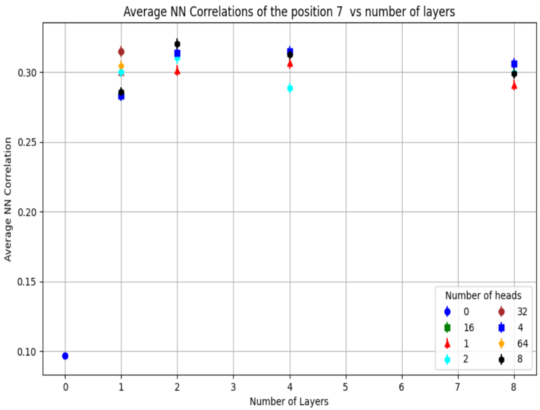
Correlation

- Each element in the sequence attends to those coming earlier
- We quantify this attention by measuring correlation (cosine similarity) between an embedding vector at certain position in the sequence with all the previous ones
- We investigate the following correlations:

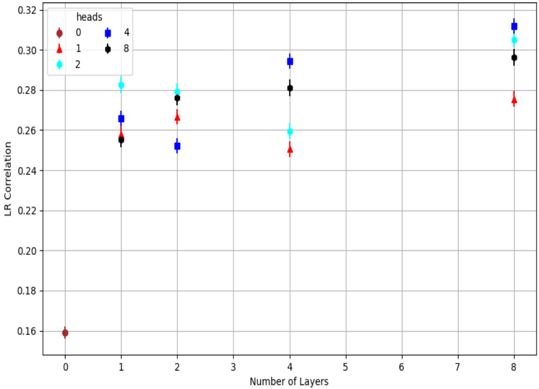




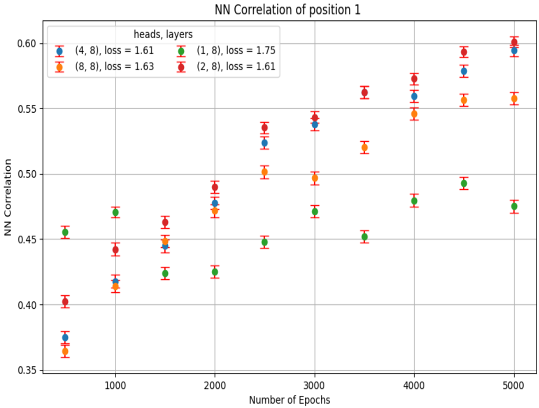
- The effect of number of heads is significant for more than one layer.
- Correlation is stronger for more than one head



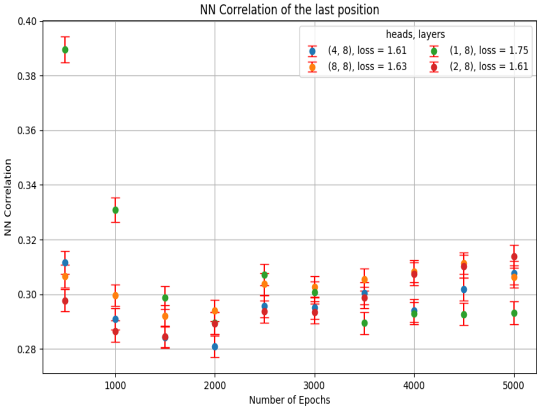
No strong dependence on the hyperparameters



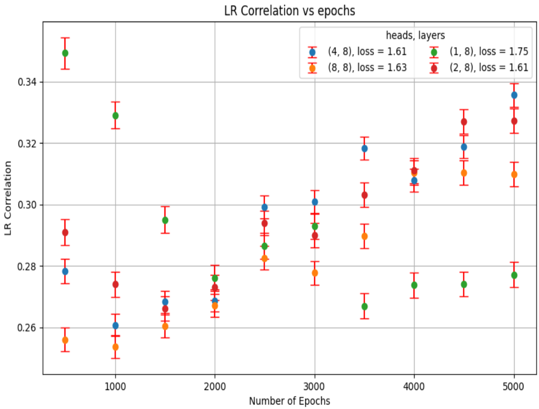
difference between number of heads visible for more than 4 layers



models with more than one head learns more about NN correlation of the position 1



NN correlation of last position is learnt in early epochs by all models quite similarly


LR Correlation vs epochs
heads, layers
(4, 8), loss = 1.61
(8, 8), loss = 1.63
(1, 8), loss = 1.75
(2, 8), loss = 1.61
LR Correlation
0.26
0.28
0.30
0.32
0.34
1000
2000
3000
4000
5000
Number of Epochs

Conclusion

- average correlation between token embeddings seems to be a qunatity that reflects power of the model
- correlations are position dependent. Highest value belongs to the first two tokens
- NN correlation of position 1 is the most sensitive to hyperparameters
- models with more than one head and layer have higher correlation
- correlations of last positions are not sensitive to the model hyperparameters
- learning dynamics of the correlation of the first position is sensitive to the number of heads
- long range and (X_{T-1}, X_{T-2}) correlations are learnt faster than (X_0, X_1)

Discussion and Outlook

- only one token to attend to: The high NN correlation of position 1
- monitoring key and query matrices to understand learning dynamics of correlations and the role of hyperparameters
- monitoring embedding of a specific character e.g. " "
- looking at covariance matrix of tokens