CS 6180 10/15

Review from last time

Transformer architecture (GPT, a lot of generative models)

self Attention vs Cross attention

↓ propagate info from previous relevant words

↑ translation

put attention on words in source sentence (between Encoder and Decoder)

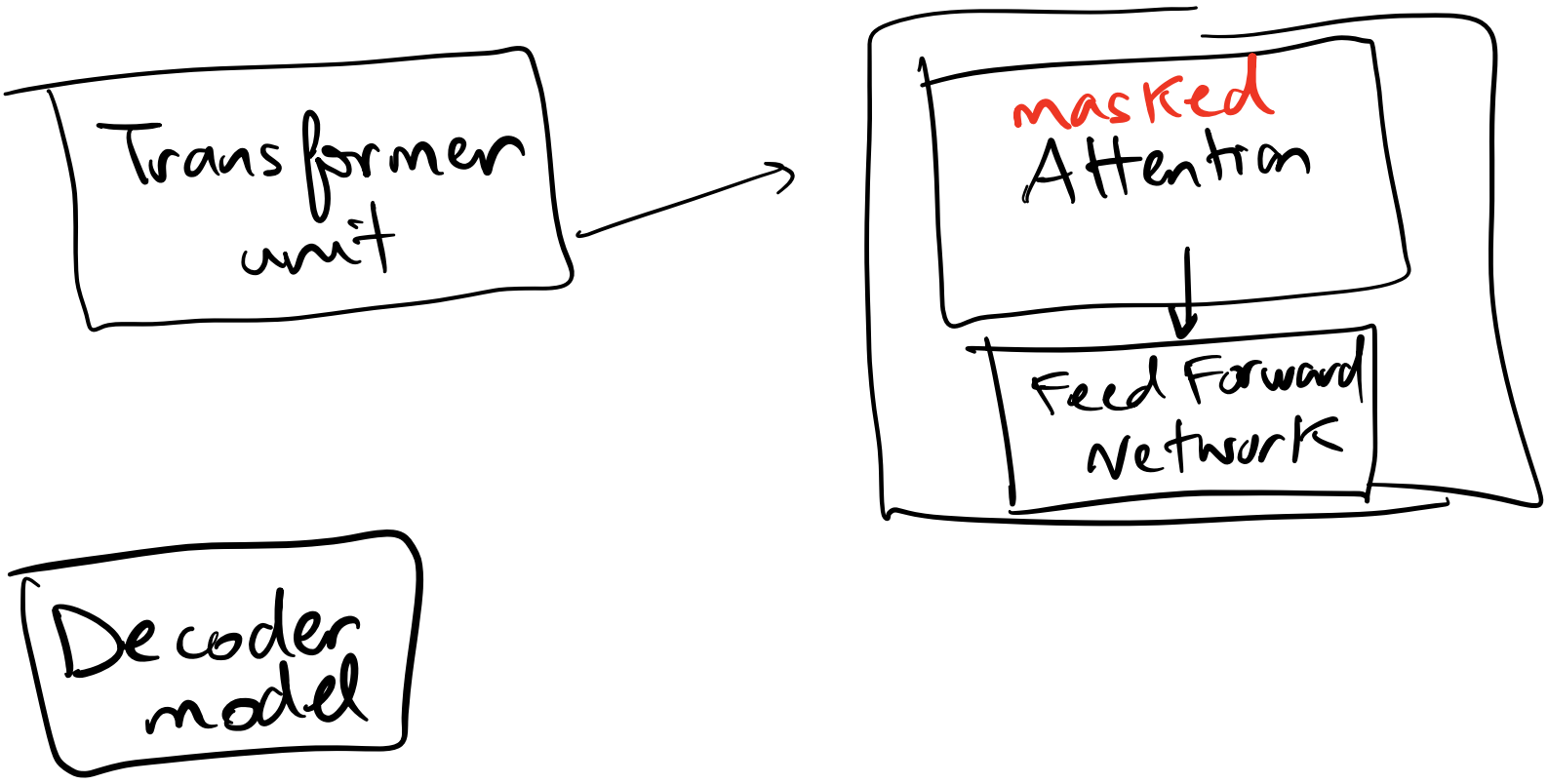
Let's build some models that are purely based on Attention.

$\vec{x} \xrightarrow{E} \vec{e} \xrightarrow{\text{add position embedding}} \vec{e}+\vec{p} \rightarrow$ [one transformer unit]

↓

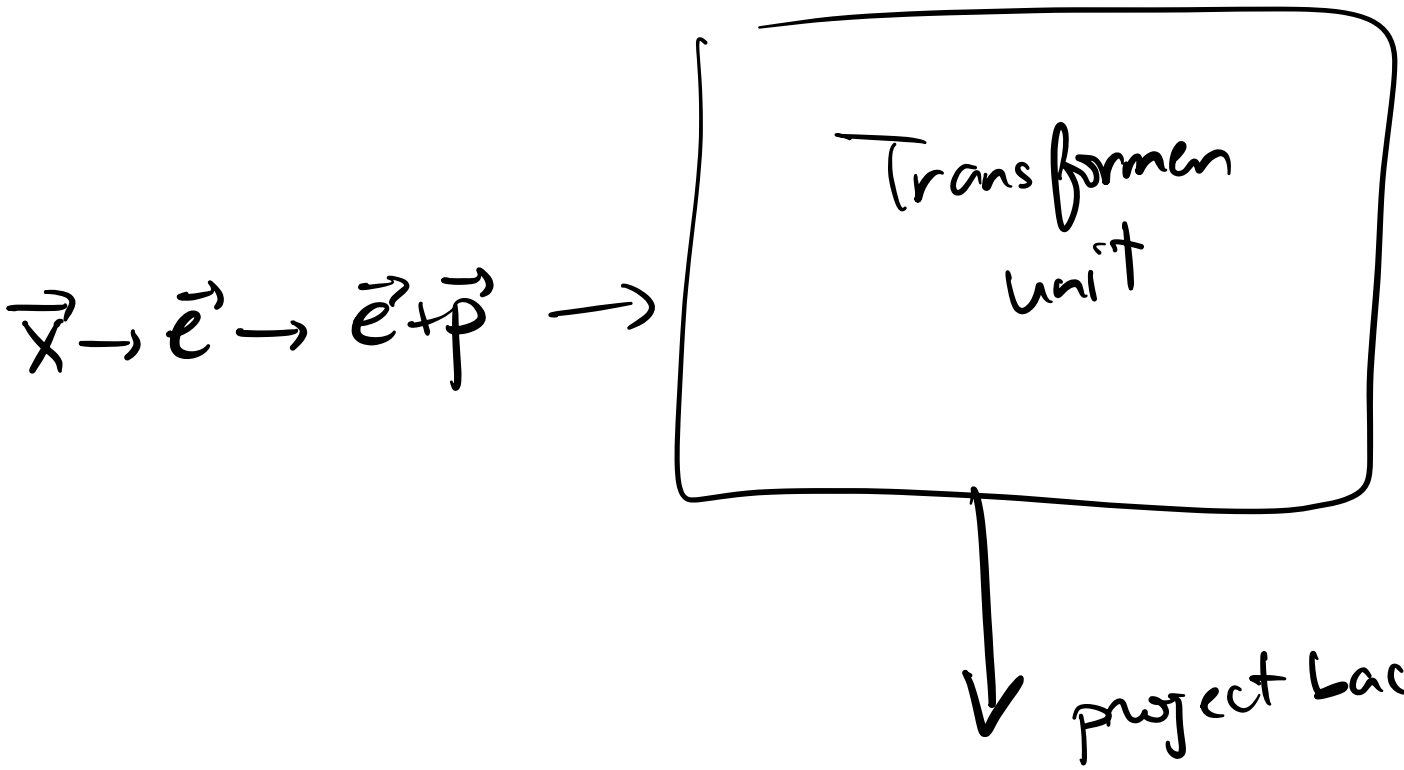
Zuko made his uncle tea at Zuko's place.

His uncle made Zuko tea at Iroh's place.



Decoder model

masking : set attention scores of futures words to $-\infty$ so that their contribution will be 0 in the softmax.



Add & Norm (component will be added to the transformer unit)

* Layer Normalization

example:

$$z = Wx + b \qquad x \sim N(0, 1)$$

$h = \sigma(z)$

ex1

$W = 1$

$b = 0$

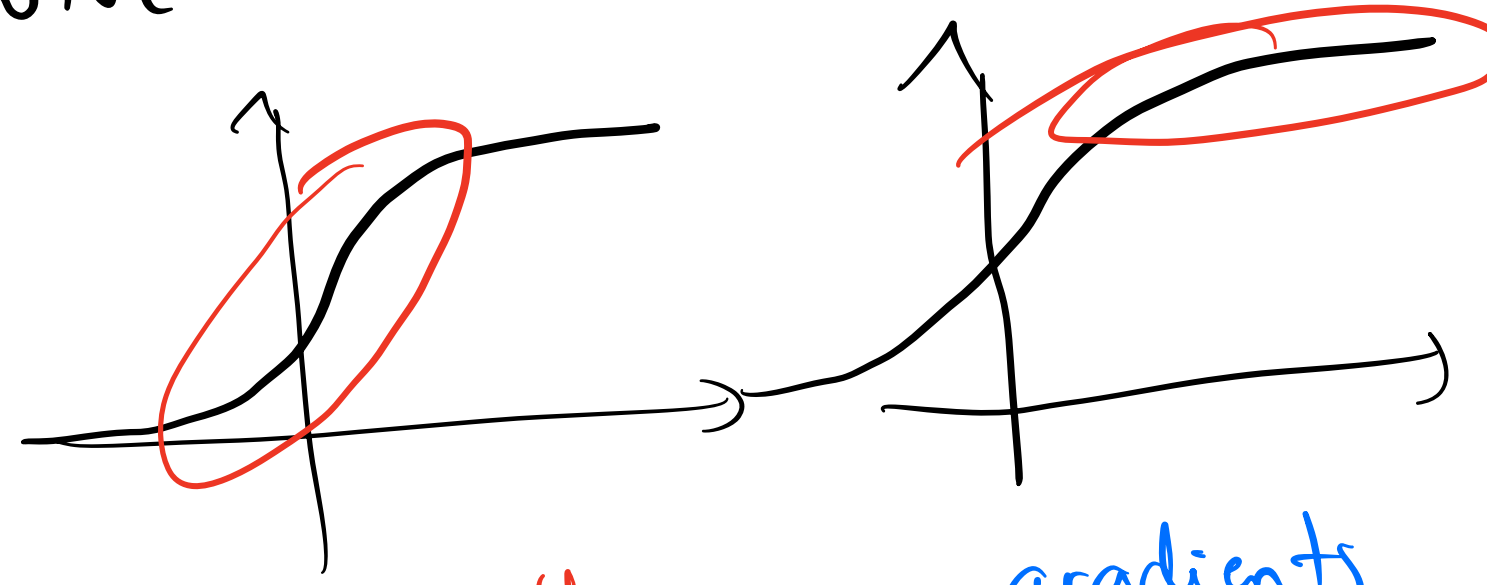
$z \sim N(0,1)$

ex2

$W = 3$

$b = 2$

$z \sim N(2,9)$



regime where the gradients are not zero

$\Rightarrow$ learning will be happening $\Rightarrow$ yay

gradients are closer to 0

learning will be slow

might even get interrupted.

⇒ vanishing gradients

⇒ Let's normalize

$$\hat{z} = \frac{z - \mu}{\sqrt{\sigma^2}}$$

$$E(\hat{z}) = 0$$

$$Var(\hat{z}) = 1$$

μ = average over the features

σ^2 = variance over the features

⇒ will go back to the "nice" area of the sigmoid so that the gradients won't vanish.

Limitations

* if σ^2 is small
  ⇒ dividing by a very small number
  ⇒ exploding values occuring with $\hat{z}$

fix: $\hat{z} = \frac{z - \mu}{\sqrt{\sigma^2 + \varepsilon}}$

$\varepsilon \sim 10^{-5} - 10^{-6}$

helps with stability of the normalization so that we don't have exploding $\hat{z}$.

* We might need to end up at the regimes where the gradients are close to 0 (especially when we have almost converged)

Fix: $y = \gamma \hat{z} + \beta$

learnable parameters

$\gamma = \sigma$

$\beta = \mu$

$\hat{z} = \frac{z - \mu}{\sigma}$

$\hat{z}\sigma + \mu = z$

* at the beginning of the model
  => the model will try to keep the features standardized ($\gamma = 1, \beta = 0$)

* As the model is learning and has learned quite a bit (almost converged)
  model now has flexibility (through γ and β) to go back to

the unstandardized features.

(yay)

mainly helps with training

we have flowing gradients
and could also end up with the
0 gradients when converged.

Residual addition

Add & Norm

(Attention is all you need, original transformer)

another version of Add & Norm

(used in GPT models for example)

Approach!

AH : attention heads

LN : layer normalization

FFN : feed forward Network

$X \longrightarrow$ transformer unit

$$y_1 = LN(x + AH(x))$$

$$y_2 = LN(y_1 + FFN(y_1))$$

residual (adding x to $AH(x)$
" y_1 to $FFN(y_1)$)

making sure we have not lost any previous info and would keep on adding new info.

Approach 2

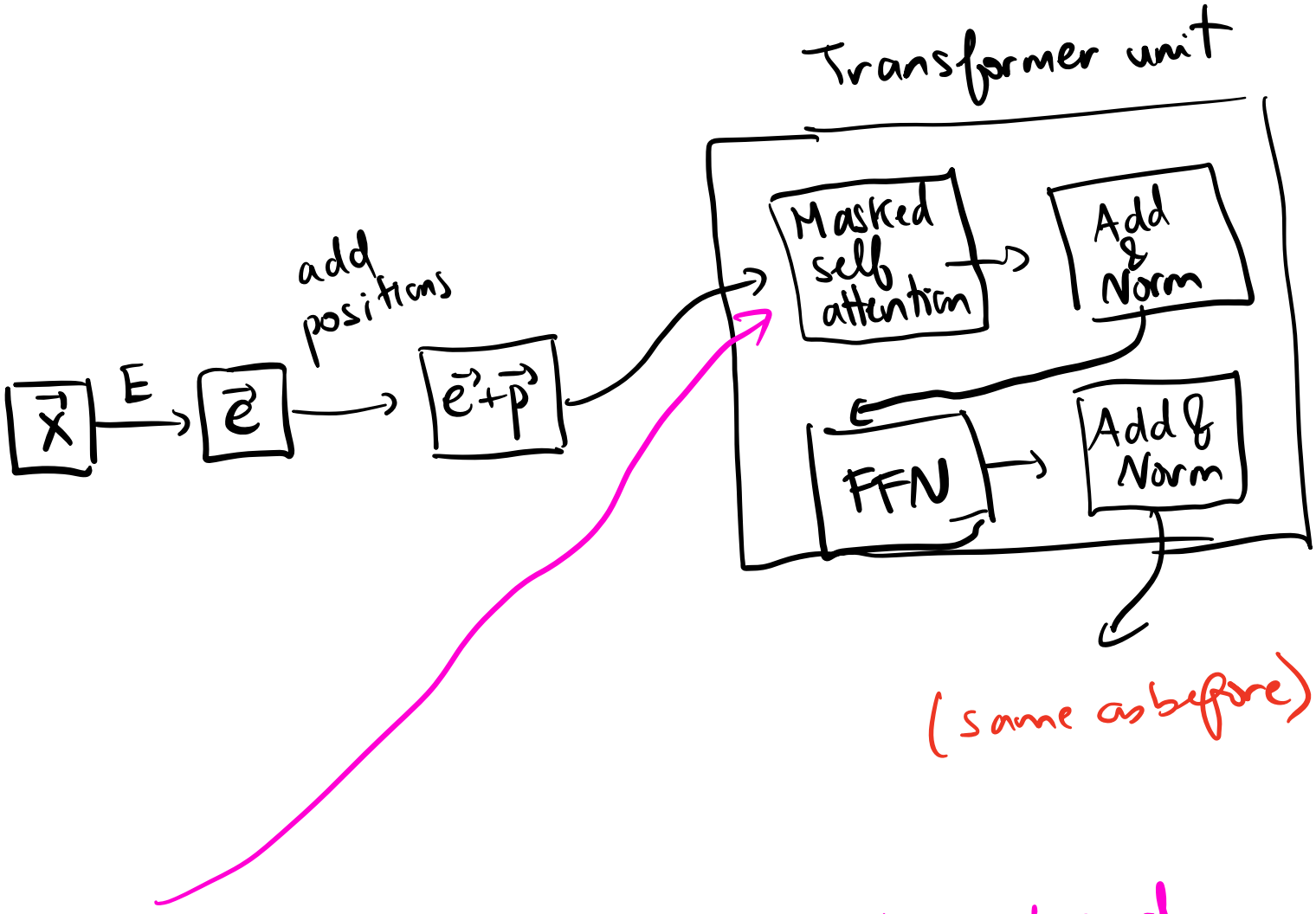
$$y_1 = x + \text{Alt}(\text{LN}(x))$$

$$y_2 = y_1 + \text{FFN}(\text{LN}(y_1))$$

gradients flow more directly

this is more stable for deep networks

(more used now)



Instead of only one attention head
why not use many so that each
attention head could attention to
different features of the sentence?
(meaning, grammar, pronoun, ...)

Attention head

$$\vec{K}_i = K\vec{e}_i \;;\quad \vec{q}_i = Q\vec{e}_i \;;\quad \vec{V}_i = V\vec{e}_i$$

$$S_{ij} = \vec{q}_i^{\,T}\vec{K}_j \qquad j = 1, 2, \ldots, m$$

$$\text{softmax}(\vec{S}_i) = \left[\frac{\exp(S_{ij})}{\sum_t \exp(S_{it})} \;\; \cdots\right]$$

$$\| \\ \alpha_{ij}$$

$$\vec{O}_i = \sum_j \alpha_{ij}\vec{V}_j$$

$\vec{O}_i^{(1)}$ $\vec{O}_i^{(2)}$ $\vec{O}_i^{(3)}$

parameters

$$\underbrace{\left[\sigma_i^{(1)} \quad \sigma_i^{(2)} \quad \sigma_i^{(3)}\right]}_{d \times 3} \underbrace{U}_{3 \times 1}$$

Attention head 1 | Attention head 2 | Attention head 3

$Q^{(1)}$ $K^{(1)}$ $V^{(1)}$

q_i

different $\rightarrow$ $\vec{K}_i^{(1)} =$

$Q^{(2)}$ $K^{(2)}$ $V^{(2)}$

$Q^{(3)}$ $K^{(1)}$ $V^{(1)}$

each attention head will focus on a different task

↳ (intuition hope)

how do we know what they're focusing on?

open research question

Mechanistic interpretability

$$O^{(\ell)} = \text{softmax}\left(\frac{(XQ)\,(XK)^T}{\sqrt{d}}\right)XV$$

if dot products become very large

→ let's scale by dimension

Transformers, LSTMs, RNNs

Autoregressive models

what are autoregressive models?

$$p(x_1, x_2, x_3, \ldots, x_m)$$

$$= p(x_1) \cdot p(x_2 | x_1)$$

$$p(x_3 | x_1, x_2)$$

$$\vdots$$

$$p(x_m | x_1, x_2, \ldots, x_{m-1})$$

autoregressive property

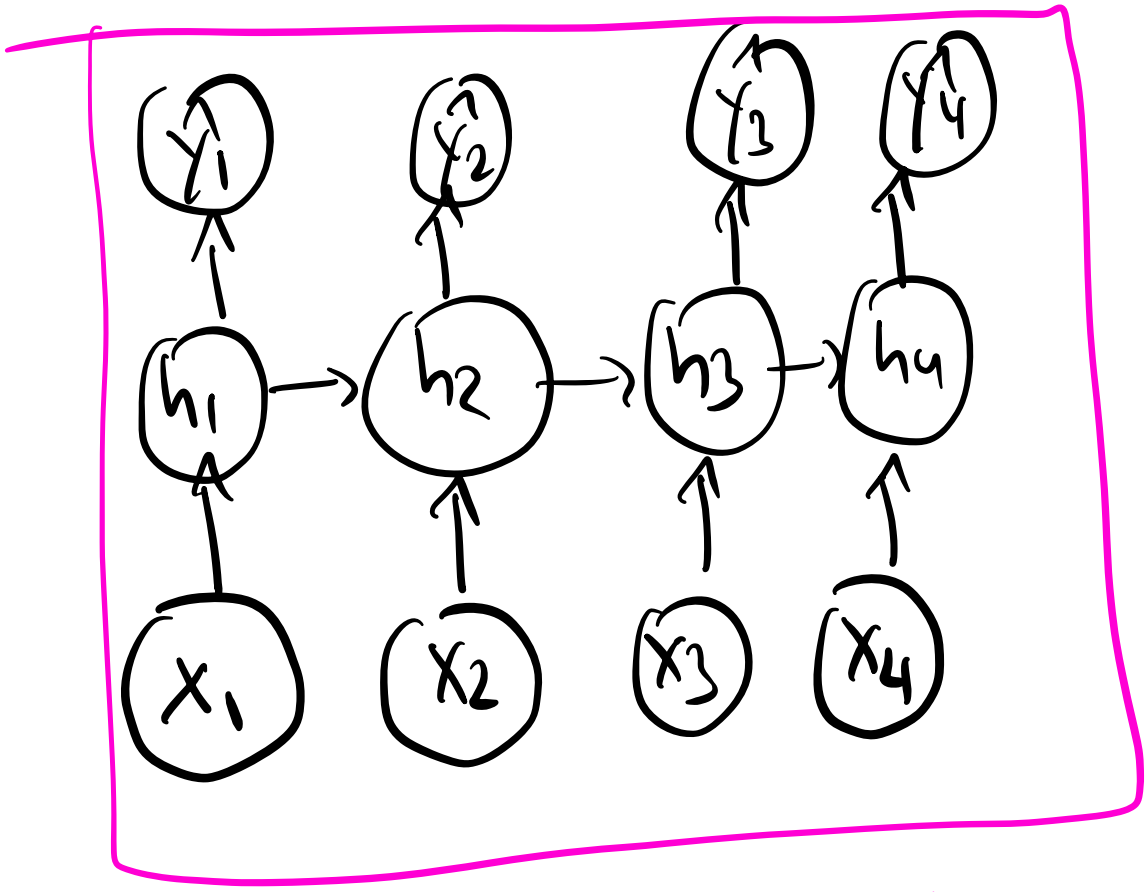
where x_m

depends on the past words that occured.

literature on time series (observations from previous times to predict an output at the next time step.)

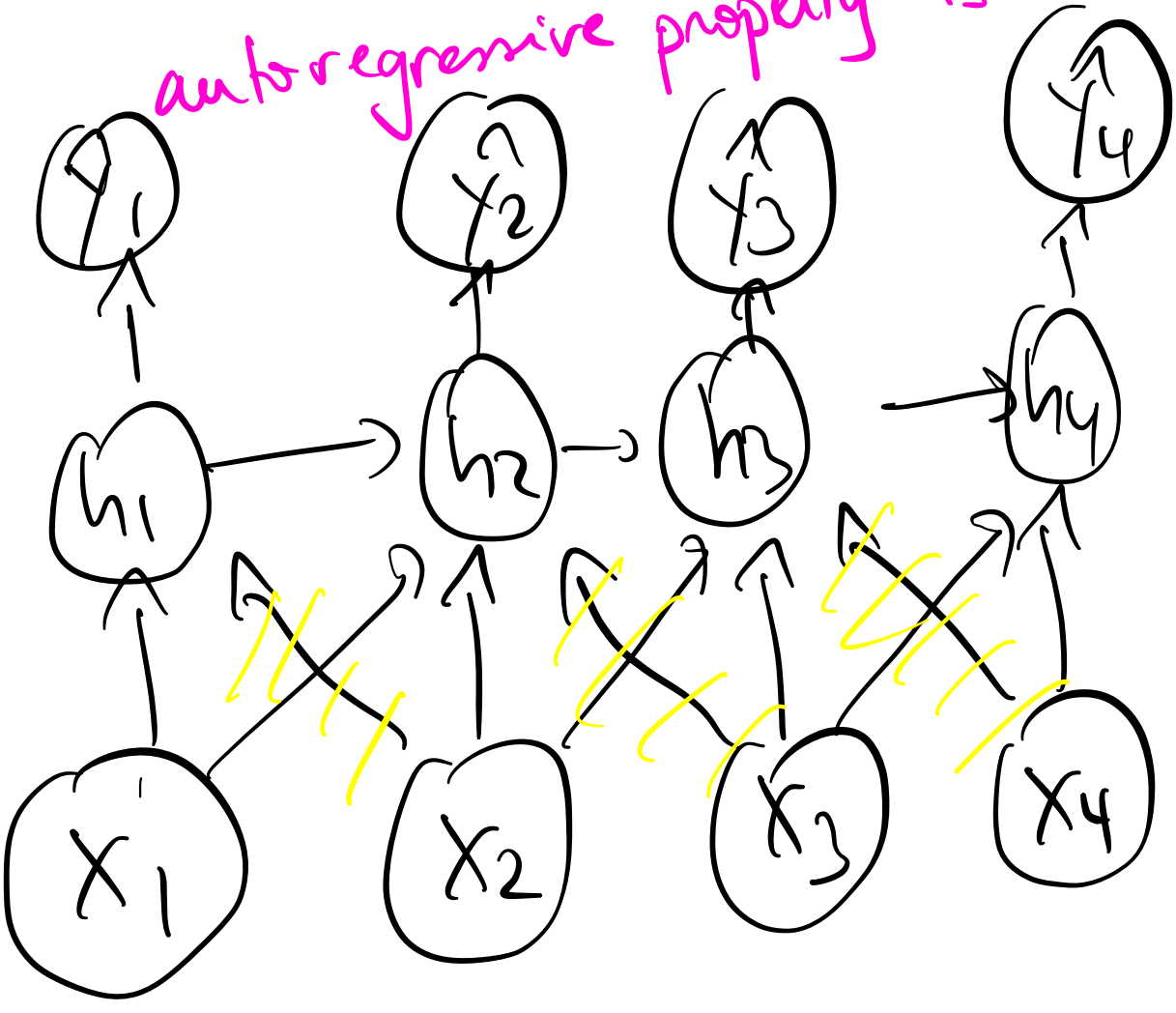
$$p(X_i \mid X_1, \cdots, X_{i-1})$$

can use a neural network to model this function with a certain # of parameters



RNN

autoregressive property is satisfied



CNN

not auto regressive model

removing these 3 edges makes it autoregressive.

1-4

1-2

2-3

3-4

2

1

6

priority for you all

21

attention head 1 : Grammar

$\vec{K}_i$ $\vec{q}_i$, $\vec{V}_i$

attention head 2: Meaning

teach

taught