15-780: Graduate AI
Lecture 9: transformers

Aditi Raghunathan

Logistics

- Homework 2
  - Grading complete
  - Solutions out this evening
- Homework 3
  - Due next Monday
- Midterm
  - Everything including todays and upcoming Wed lecture
  - Next Monday: **review session (optional attendance)**

Recap of deep networks

→ MLP $z_i = \sigma(W_i^T z_{i-1})$

→ expressivity

→ running gradient descent

autodiff framework chain rule

Batching in deep learning

Recap of stochastic gradient descent

Have a batch of examples B ⟹ if entire train set = gradient descent

In each step

$$w_i^{(t)} = w_i^{(t-1)} - \sum_{x,y \in B} \eta \, \nabla \, \mathrm{loss}(x, y; w)$$

Key idea: $\nabla \mathrm{loss}(x_i, y_i; w)$ is independent of other examples

$x_i \in \mathbb{R}^d$ B examples in a batch

$$\underline{X} = \begin{bmatrix} \leftarrow & x_1 & \rightarrow \\ \leftarrow & x_2 & \rightarrow \\ & \vdots & \\ \leftarrow & x_B & \rightarrow \end{bmatrix}$$

each row is a separate input

$w^T x \Rightarrow \underline{X} w$

$\underline{X} \in \mathbb{R}^{B \times d}$

$$\begin{bmatrix} \leftarrow & x_1 & \rightarrow \\ \leftarrow & x_2 & \rightarrow \\ & \vdots & \\ \leftarrow & x_B & \rightarrow \end{bmatrix} \begin{bmatrix} w \end{bmatrix} = \begin{bmatrix} \leftarrow & w^T x_1 & \rightarrow \\ \leftarrow & w^T x_2 & \rightarrow \\ & \vdots & \\ \leftarrow & w^T x_B & \rightarrow \end{bmatrix}$$

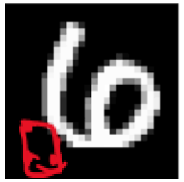
$$\underline{X} W \quad : \quad \underline{X} = \begin{bmatrix} \longleftarrow & x_1 & \longrightarrow \\ \longleftarrow & x_2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & x_B & \longrightarrow \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} = \begin{bmatrix} aa' + bc' & ab' + bd' \\ ca' + dc' & cb' + dd' \end{bmatrix}$$

Structured inputs

- Images

28

28 

each pixel & roll out

784

$[x_1 \; . \; . \; . \; - \; - \; - \;]$

pixels

⇓ missing out spatial information

- Text

The quick brown fox jumps over the lazy dog

Key idea

We need to reason about sets of inputs

→ collection of pixels
→ collection of words

Language modeling

$e_{word} \in \{0,1\}^V$

- Notation for one-hot vector e_{word} = zero everywhere except in the location of word in vocabulary
- Vocabulary V
- Input: **sequence** of T ~~tokens~~ words $e_{the}, e_{quick}, e_{brown}$ ------

$$X = \begin{bmatrix} \longleftarrow & e^T_{word_1} & \longrightarrow \\ \longleftarrow & e^T_{word_2} & \longrightarrow \\ & \vdots & \\ \longleftarrow & e^T_{word_T} & \longrightarrow \end{bmatrix} \qquad X \in \mathbb{R}^{T \times V}$$

we want $p(word_{T+1} \mid X) = \mathbb{R}^V$

probability distribution over V words

Batch operations?

$$XW$$

$$\begin{bmatrix} e_{\text{the}}^{T} \\ e_{\text{quick}}^{T} \\ \vdots \end{bmatrix} W = \begin{bmatrix} e_{\text{the}}^{T} W \\ e_{\text{quick}}^{T} W \\ \vdots \end{bmatrix}$$

Piazza poll

$$X = \left.\begin{bmatrix} \leftarrow e_{word1} \rightarrow \\ \leftarrow e_{word2} \rightarrow \\ \vdots \end{bmatrix}\right\} \text{same example}$$

- Which of the following operations allow for sharing information across words

A, W are some matrices of app dim

- ~~(A)~~ XW → batching
- ~~(B)~~ $\sigma(XW)$
- (C) A X →
- (D) $\sigma(AX)$

A(X) X

A matrix depends on X for self-attention

Mixing information

$$\begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a'a + b'c & a'b + b'd \\ c'a + d'c & c'b + d'd \end{bmatrix}$$

AX : combines info across rows

XW : treats each row separately

impractical

$[\leftarrow \text{word}_1 \rightarrow ; \leftarrow \text{word}_2 \rightarrow \quad \ldots\ldots \quad]$

A: a probability distribution

$$Y = A \; X$$

$$X \in \mathbb{R}^{T \times d} \qquad A \in \mathbb{R}^{T \times T} \qquad Y \in \mathbb{R}^{T \times d}$$

$$\begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{bmatrix} \begin{bmatrix} \leftarrow \text{word}_1 \rightarrow \\ \leftarrow \text{word}_2 \rightarrow \end{bmatrix}$$

$$\begin{bmatrix} \alpha & 1-\alpha \\ \beta & 1-\beta \end{bmatrix} \begin{bmatrix} \leftarrow \text{word1} \rightarrow \\ \leftarrow \text{word2} \rightarrow \end{bmatrix} = \begin{bmatrix} \alpha \, \text{word1} + (1-\alpha) \, \text{word}_2 \\ \beta \, \text{word}_1 + (1-\beta) \, \text{word2} \end{bmatrix}$$

P_i's are dist over T elements

$$\begin{bmatrix} \leftarrow P_1 \rightarrow \\ \leftarrow P_2 \rightarrow \\ \vdots \\ \leftarrow P_T \rightarrow \end{bmatrix} \begin{bmatrix} \leftarrow 1 \rightarrow \\ \vdots \\ \leftarrow T \rightarrow \end{bmatrix} = T \text{ diff combinations of } X$$

A: a probability distribution

$$Y = AX$$

- each row is a probability distribution
- each row of X corresponds to i^{th} word

Creating the matrix A

- construct scores $\longrightarrow$ softmax
- "similarity or dot product between words i & j"

$p^{(i)}$: i^{th} row $\begin{bmatrix} \longleftarrow & i & \longrightarrow \end{bmatrix} \begin{bmatrix} & & \end{bmatrix}$

$\sum p_j^{(i)}$ word$_j$

$p_j^{(i)}$: similarity b/w i & j

Creating the matrix A

$$A = \text{softmax}\left(\frac{(XW_K)\cdot(XW_Q)^T}{\sqrt{d}}\right)$$

$$P_j^{(i)} = \Big(\underbrace{X_i W_Q}_{\text{query}}\Big)^T \Big(\underbrace{X_j W_{\cancel{Q}K}}_{\text{key}}\Big)$$

Final form for self-attention

$$Y = \text{softmax}\left(\frac{X \underbrace{W_Q W_K^T}_{W_A} X^T}{\sqrt{d}}\right) X W_V$$

For now: W_Q, W_K & W_V are $\mathbb{R}^{d \times d}$ matrix

Properties of attention

- Full mixing

$p_j^{(i)}$: looks at "word$_i$" & "word$_j$")

$\forall\, i, j \in [1 \cdots T]$

Properties of attention

- Let us increase the size of the set T
  - What happens to W_q, W_k, W_v ? $\in \mathbb{R}^{d \times d}$

$\mathbb{R}^{T \times T}$ $O(T^2)$ in the construction of A

Properties of attention

- Does the ordering between words matter?

Fixed W_Q, W_K, W_V

does A: "The quick brown fox" : X

"the brown fox quick" : $\tilde{X}$

q, k, q, k

$A(X)$ & $A(\tilde{X})$ are same upto permutations