Date ___/___/___

Assignment 2

1. Perceptron cannot solve XOR as; it is a linear classifier i.e it computes '$y = step(w_1x_1 + w_2x_2 + b)$'
It draws a single straight line (in 2D) or a hyperplane. Everything on one side of the class is 1 and other side is 0.
Therefore, it solve linearly separable problems and XOR is not linearly seperable, so a single perceptron fails.

A multi layer perceptron (MLPs) was introduced and it brought following changes:- hidden layers were introduced which allows network to learn intermediate representations and combine non linear boundaries into non linear decision regions. Hidden layers uses non linear activations. Thus, XOR becomes solvable via feature transformation.

2. A linear layer is defined as, $f(x) = Wx + b$
where W = weight matrix & b = bias vector
stacking two layers gives

Layer 1; $h = W_1x + b_1$

Layer 2; $y = W_2x + b_2$

Subsituting $y = W_2(W_1x + b_1) + b_2$

$= W_2W_1x + (W_2b_1 + b_2)$

$y = W'x + b'$

$\hookrightarrow$ again a linear format

Therefor linear layers stacked on linear layers remain linear.

Page No. ___

Date __/__/__

→ As, back propagation uses chain rule and gradients are product of many derivatives. Most activation functions have derivatives less than 1, which no ~~pro~~ multiplying, disappears. Therefore, gradients shrink in deep networks.

→ Sigmoid: function is $\sigma(x) = \frac{1}{1+e^{-x}}$

which saturates between 0 and 1 and gradient is 0 when large ~~to~~ positive or negative input

while $\text{ReLU}(x) = \max(0, x)$

$$\text{ReLU}'(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

So, we can conclude ReLU's derivative has no shrinking gradient and no saturation for positive outcomes.

Therefore, ReLU solves vanishing gradients better than Sigmoid.

3 Positive encoding is necessary to provide the model with information about token order, enabling it to understand syntax, sequence structure and temporal relationship in data such as natural language

In absolute positional Encoding, each position in the sequence is assigned a unique positional vector which is added to token embedding which may be learned or fixed ~~while~~

While;

Date __/__/__

In Sinusoidal positioning encoding, it uses fixed sine and cosine functions of different frequencies to represent positions. It generalizes better than absolute encoding and does not require additional parameters.

→ RoPE introduces positional information by rotating query and key vectors in self attention by an angle proportional to token position. As a result, RoPE integrates positional information directly into additional computation, enabling stable extrapolation to long contexts & improving performance on long sequence tasks.

4 In self attention mechanism of transformer, each input token is projected into three vectors

1) Queries (Q); Represents what information the current token is looking for
2) Key (K); Represents what information ~~current~~ token contains
3) Value (V); Represents actual information or content to be aggregated

→ Attention scores is computed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

as dimension d_K increases, dot product $Q \cdot K$ tends to grow in magnitude.
Large values passed to softmax function

Page No.

Date ___/___/_____

cause it to become very sharp, resulting in small gradients & unstable training.

→ Query (Q) and Key(K) vectors for the same token are derived from same embedding. Their dot product is usually larger than other tokens and leads to higher attention score along the diagonal.

Thus, diagonal values are highest, indicating strong self-attention.

5 Multi-head attention allows the transformer to attend to information from multiple representation subspaces at the same time. Instead of computing single attention distribution, the model splits attention into several heads, each learnig different relationships such as syntax, sementics or long range dependences.

This improves model's ability to capture complex patterns.

→ d-model represents the dimensionality of token embeddings & hidden representations in the transformer.

h; Represents no of attention heads to used in multi head attention

d-head ; Represents dimensionality of each attention head. given as dhead = dmodel

Page No.

Date ___/___/_____

6 Greedy Decoding is suboptimal because to it makes locally optimal decisions, not globally optimal ones.
It may choose a high-probability word early that leads to a low probability overall sentence

→ Beam Search keeps track of the top K candidate sequences at each decoding step instead of only one. It explores multiple possible continuations simultaneously and balances local and global optimization
Therefore, Beam Search produces better sequences

→ Example; "Do you like coffee ____"?

step 1; • ? → probability 0.55
• or → probability 0.45

Greedy chooses '?'

Sentence ends • Do you like coffee ? → probability 0

Alternative; or → tea → ?
Total = $0.45 \times 0.9 \times 0.9 = 0.3645$

Beam search may keep both and choose "Do you like coffee or tea ?"

Greedy prefers early termination.

Page No. ____

Date __/__/__

Solving

1. d_model = 768
h = 12

(a) d_head

$d\_head = \frac{d_{model}}{h} = \frac{768}{12} = 64$ Ans

(b) Parameters for Q, K, V projection matrices of one layer

Each projection matrix 768×768 = 589,824

Total parameter 3×589824 = 1769472
Ans

2) Softmax values
Attention score from one row [2.0, 1.0, 0.0]

Exponentials
$e^2 = 7.389 \quad e^1 = 2.718 \quad e^0 = 1$

$7.389 + 2.718 + 1 = 11.107$

$\left[\frac{7.389}{11.107}, \frac{2.718}{11.107}, \frac{1}{11.107}\right] \approx [0.665, 0.245, 0.09]$

Ans