# Deep Learning Homework 2

Meet Oswal (mo2532)

March 2024

## 1 Question 1

final input = $X_T$
final hidden state = $H_T = X_T - H_{T-1}$
output = $Y_T = sigmoid(1000 * H_T)$
after expanding $H_T$:
$Y_T = sigmoid(1000(X_T - H_{T-1}))$ after expanding $H_{T-1}$ and forward:
$H_{T-1} = X_{T-1} - H_{T-2}$
$H_{T-2} = X_{T-2} - H_{T-3}$
...
$H_1 = X_1 - H_0$
Hence, if we merge all equations we get:
$Y_T = sigmoid(1000(X_T - (X_{T-1} - (X_{T-2} ... - (X_1 - H_0)))$
$Y_T = sigmoid(1000(X_T - X_{T-1} + X_{T-2} - X_{T-3} ... - X_1 + H_0))$
as per given statement: T is a even number, therefore:
$Y_T = sigmoid(1000(\Sigma_{i=1}^{\frac{T}{2}} X_{2i} - \Sigma_{i=1}^{\frac{T}{2}} X_{2i-1} + H_0))$

$Y_T = sigmoid\left(\left(\Sigma_{i=1}^{\frac{T}{2}}(X_{2i} - X_{2i-1}) + H_0\right)\right)$
Hence, Answer computed by the output unit at the final time step is:

$$Y_T = sigmoid\left(1000 \,.\, \left(\Sigma_{i=1}^{\frac{T}{2}}(X_{2i} - X_{2i-1}) + H_0\right)\right)$$

## 2 Question 2

a)
$x_1 = d + b$ as we know norms of d and b, norm of $x_1$:
$||x_1|| = ||d|| + ||b||$
$||x_1|| = \sqrt{\beta^2 + \beta^2}$
$||x_1|| = \beta\sqrt{2}$

$x_2 = a$ as we know norm of a, norm of $x_2$ is:
$||x_2|| = ||a||$

$||x_2|| = \beta$

$x_3 = c + d$ as we know norm of c and d, norm of $x_3$ is:
$||x_3|| = ||c|| + ||d||$
$||x_3|| = \sqrt{\beta^2 + \beta^2}$
$||x_3|| = \beta\sqrt{2}$

b)
self attention outputs are: $y_1, y_2, y_3$
1) Calculate $W_{ij}$:
given that $q_i = k_i = v_i = x_i$
$W_{ij} = \langle q_i, k_j \rangle = \langle x_i, x_j \rangle$ where $i = 0, 1, 2, 3...$
therefore dot product of 2 orthogonal vectors is:
dot product of $(x_i, x_j)$ is 0 if i is not equal to j or $(x_i \, . \, x_j)$ otherwise.

2) Apply soft max:
$W_{ij} = softmax(< x_i, x_j >)$
therefore:'
$W_{ij} = 1$ when i = j
$W_{ij} = 0$ otherwise
This is because the dot product of a token with itself is $\beta^2$
Normalizing this using soft max, we get a attention score of 1 for each token attending to itself.

3) Dot product of weight matrix and value which is equal to input.
$y_i = \Sigma_{j=1}^{3} W_{ij} \, . \, x_j$
therefore:
$y_1 = x_1 = d + b$
$y_2 = x_2 = a$
$y_3 = x_3 = c + d$

c)
In the above example, each token computes the attention score with respect to itself and all other tokens, indicating the importance of each token w.r.t others. In case the vectors are created using orthogonal vectors the attention score between token and itself is 1, effectively emphasizing the token's own value during then attention mechanism. This results in the network 'copying' the input values to the output by directly mapping each token to its corresponding output without transformation.

## 3 Question 3

The standard self-attention mechanism computes the attention weights by taking the dot product of the query and key vectors, applying a soft max operation, and then multiplying by the value vector. This process involves a matrix multiplication operation which has a time complexity of $O(T^2)$ for an input with **T** tokens.
In contrast, the linear self-attention mechanism described in question simplifies the computation by dropping the exponential in the soft-max operation. This means that instead of computing $e^{QK}$ we just use $QK$ directly. This avoids the need for a matrix multiplication operation, which is the primary source of the quadratic time complexity in the standard self-attention mechanism.
Instead, the linear self-attention mechanism involves a series of vector operations (addition, multiplication, and normalization), each of which can be performed in linear time. Therefore, the overall time complexity of the linear self-attention mechanism is $O(T)$ which is significantly more efficient than the standard self-attention mechanism for large inputs.
Mathematically:
Original we did:

$$y_i = \Sigma_{i=1}^{T}(softmax(< q_i, k_j >), v_i)$$

In above statement there was a nonlinear operation of soft-max. But, in case of linear-attention it becomes:

$$y_i = \Sigma_{j=1}^{T}(q_i, k_j, v_j)$$

here there is no non linear operation and all multiplications and addictions can be carried out in $O(T)$ time.