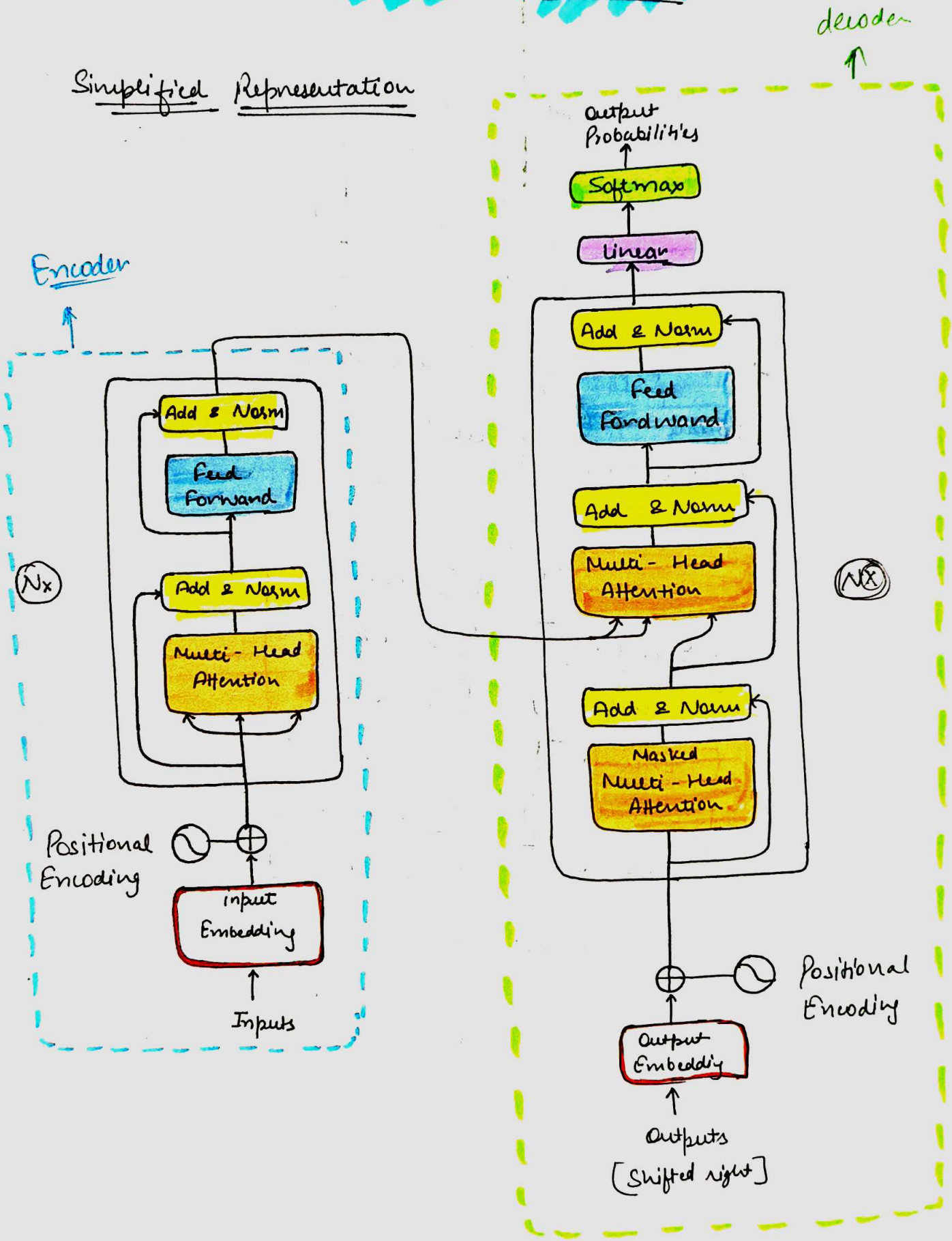
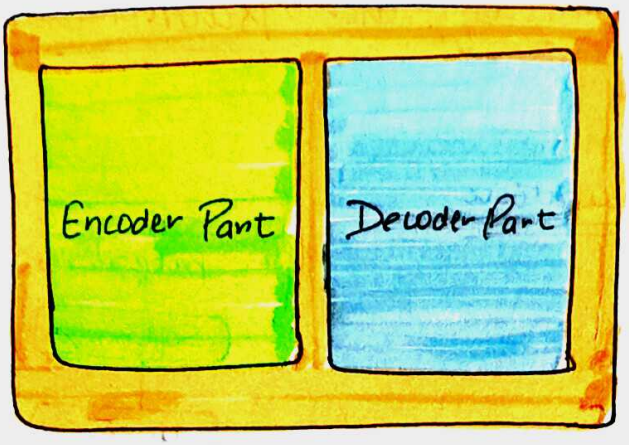
Transformer Architecture

Simplified Representation

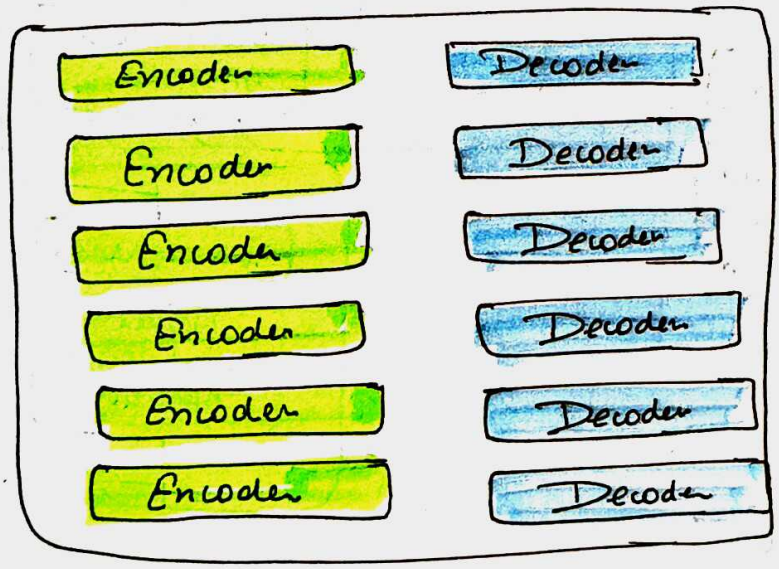




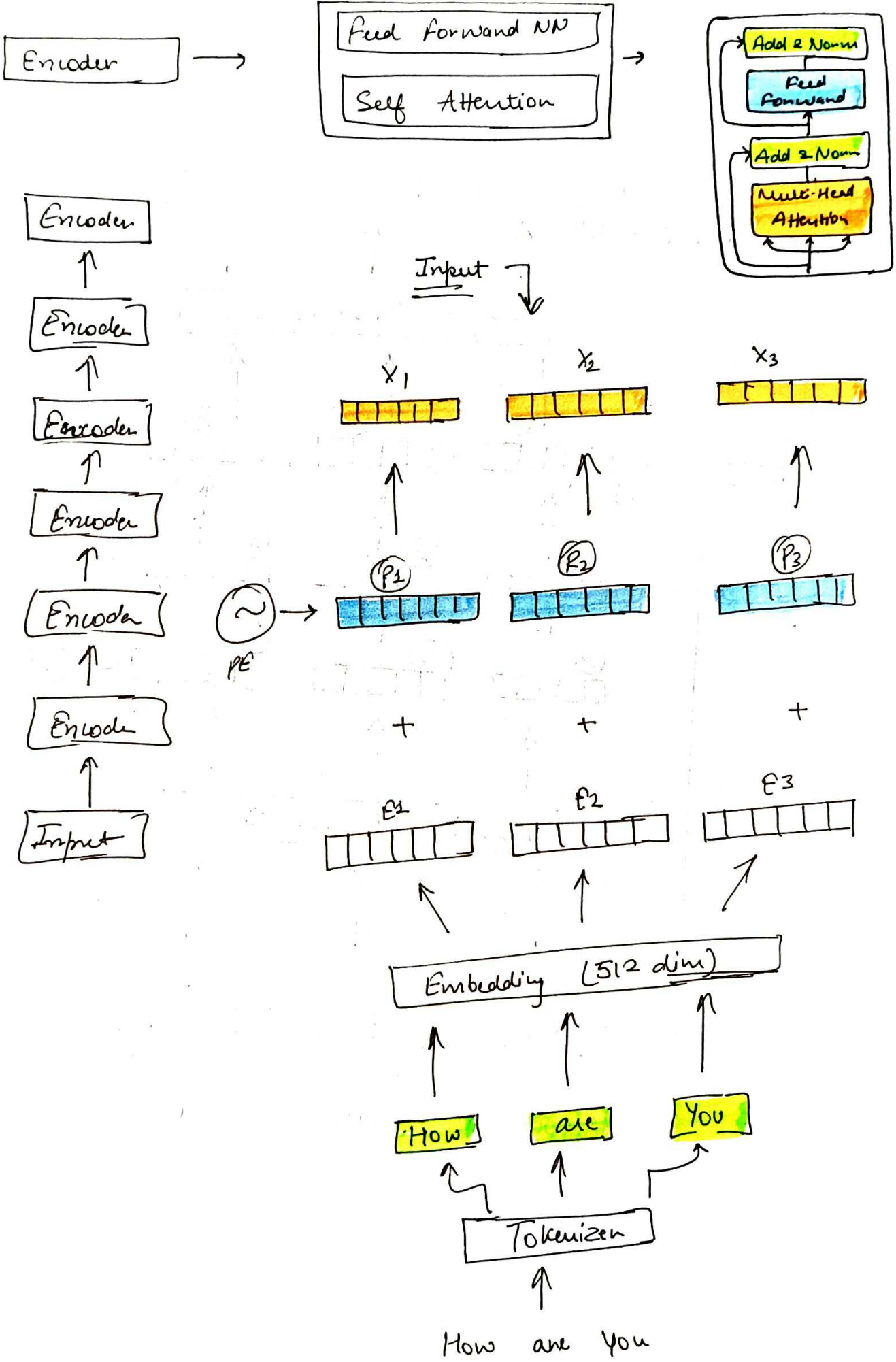
Transformer

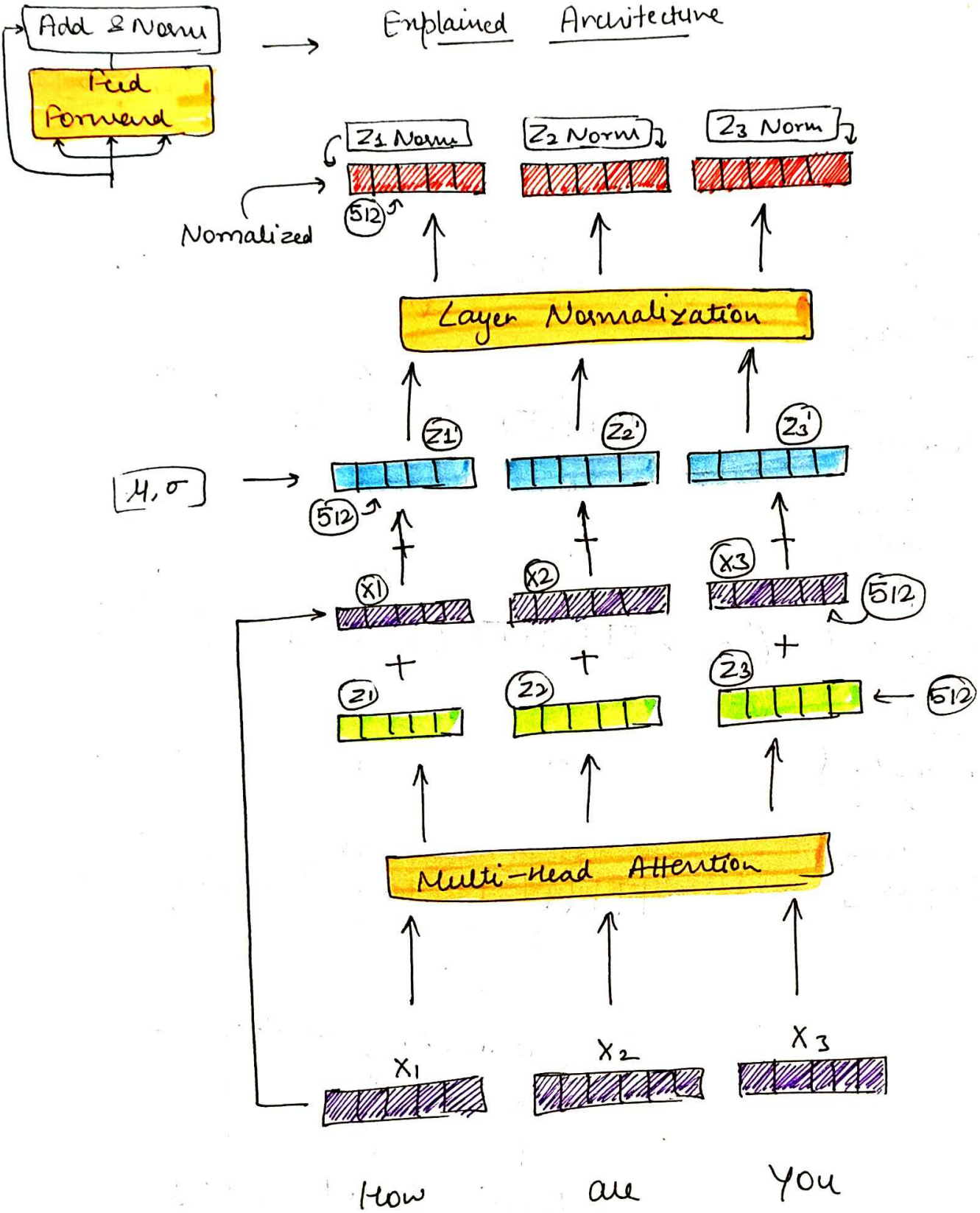
* In Research paper of Transformation

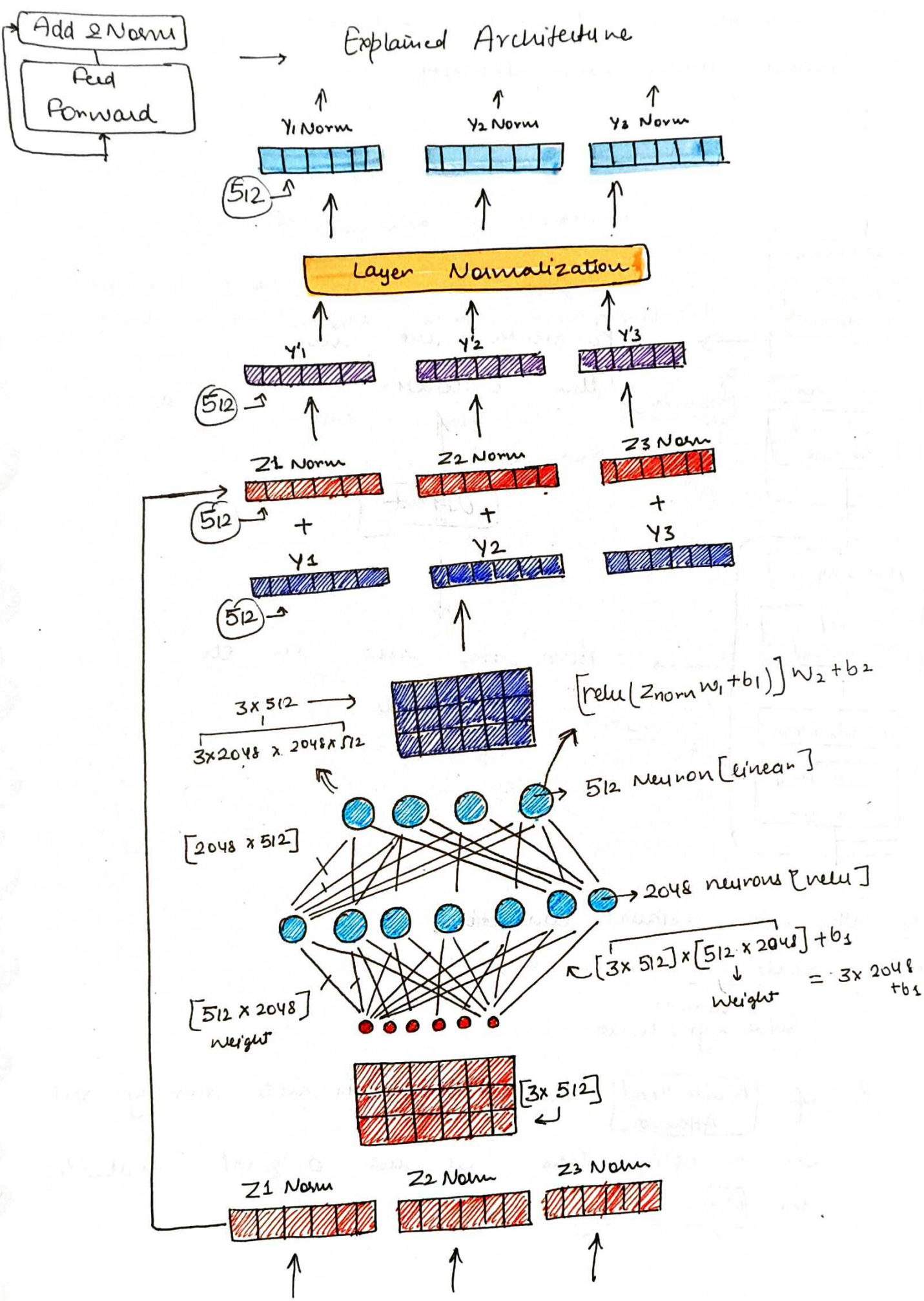
Encoder → 6
Decoder → 6
} used



Transformer

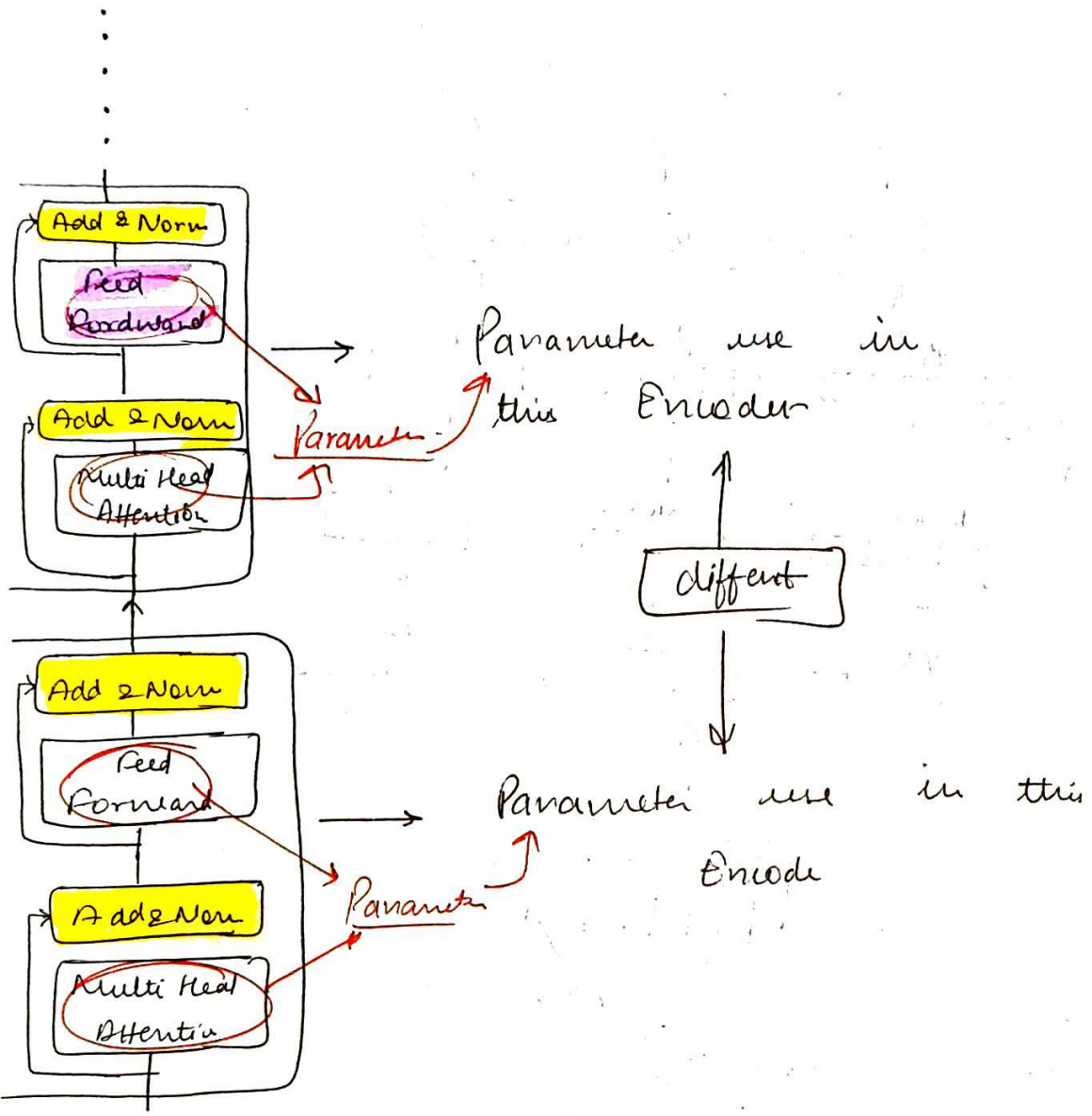

Encoder
Feed Forward NN
Self Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Encoder
Encoder
Encoder
Encoder
Encoder
Encoder
Input
Input
X1
X2
X3
P1
P2
P3
PE
+
+
+
E1
E2
E3
Embedding (512 dim)
How
are
You
Tokenizer
How are You


Add & Norm
Feed Forward
Explained Architecture
Z1 Norm
Z2 Norm
Z3 Norm
Normalized
512
Layer Normalization
Z1'
Z2'
Z3'
μ, σ
512
X1
X2
X3
512
Z1
Z2
Z3
512
Multi-Head Attention
X1
X2
X3
How
are
You


Add & Norm
Feed Forward
Explained Architecture
Y1 Norm
Y2 Norm
Y3 Norm
512
Layer Normalization
Y'1
Y'2
Y'3
Z1 Norm
Z2 Norm
Z3 Norm
Y1
Y2
Y3
3x512
3x2048 x 2048x512
[relu(Znorm W1+b1)] W2+b2
512 Neuron [Linear]
[2048 x 512]
2048 neurons [relu]
[3x 512] x [512 x 2048] +b1
weight
= 3x 2048 +b1
[512 x 2048]
weight
[3x 512]
Z1 Norm
Z2 Norm
Z3 Norm

Why increase size from 512 to 2048?

→ Because add non-linearity



Q. Why use residual connection?

Ans 1. Stable Maintain — deep NN

Solve vanishing gradient

2. If Multi-Head Attention and Feed Forward is not working well on embedding then we use original embedding in Norm and Add block

2. Why use a Feed Forward Neural Network?

Sol Capture Non-linearity in the process

3. Why use 6 encoder blocks?

Sol Language → complex to understand

Researchers use different no. of encoder blocks and 6 encoder give good accuracy.

Multi Encoder → Makes easy to understand Language.