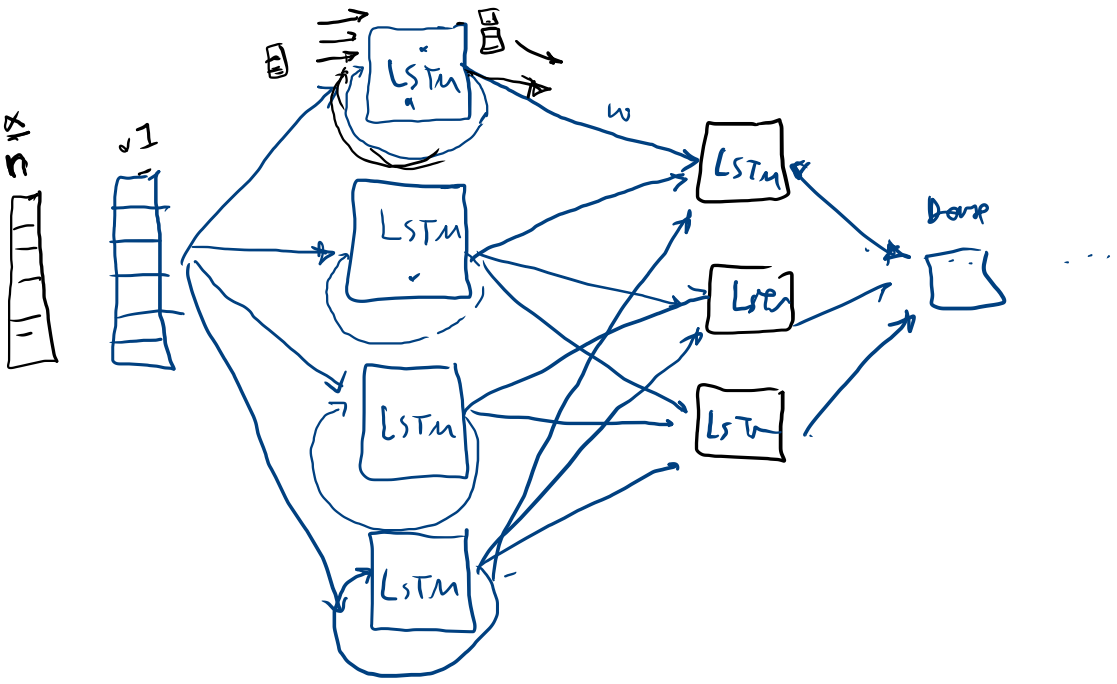
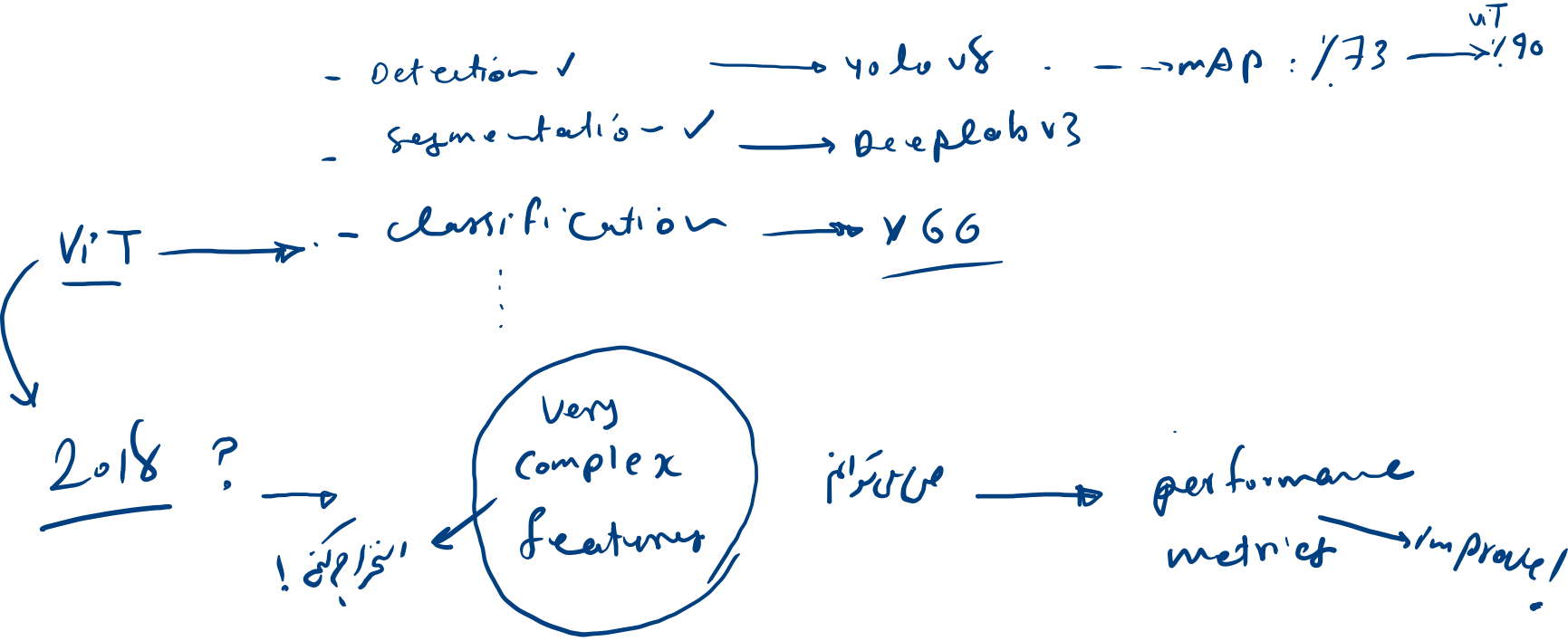
ViT (Vision Transformer)

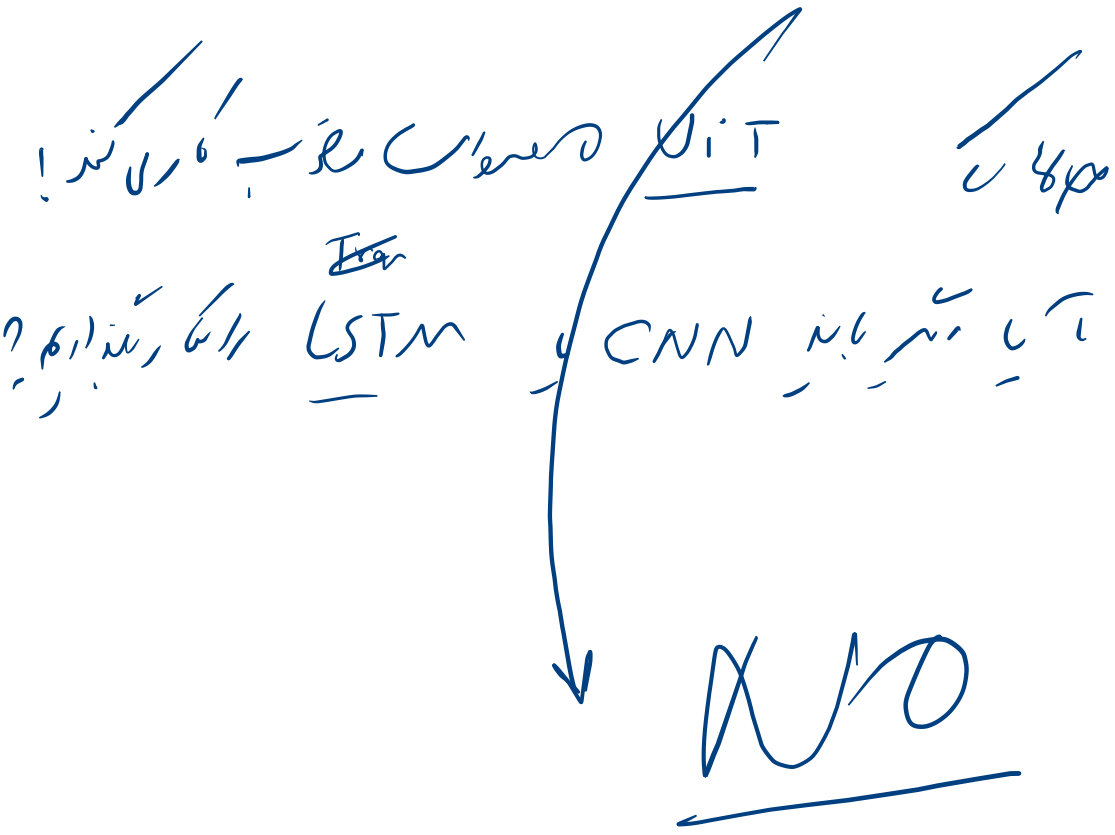
Transformer: Encoder / decoder

Vision Transformer:

layer1: LSTM (units = 5) / layer2: LSTM (units: 3)




- Detection ✓ → yolo v8 . → mAP : 73 → 90 (ViT)
- segmentation ✓ → Deeplab v3
- classification → VGG
ViT →
2018 ?
Very Complex features
performance metrics → Improve!


ViT
NO

Data Hungry

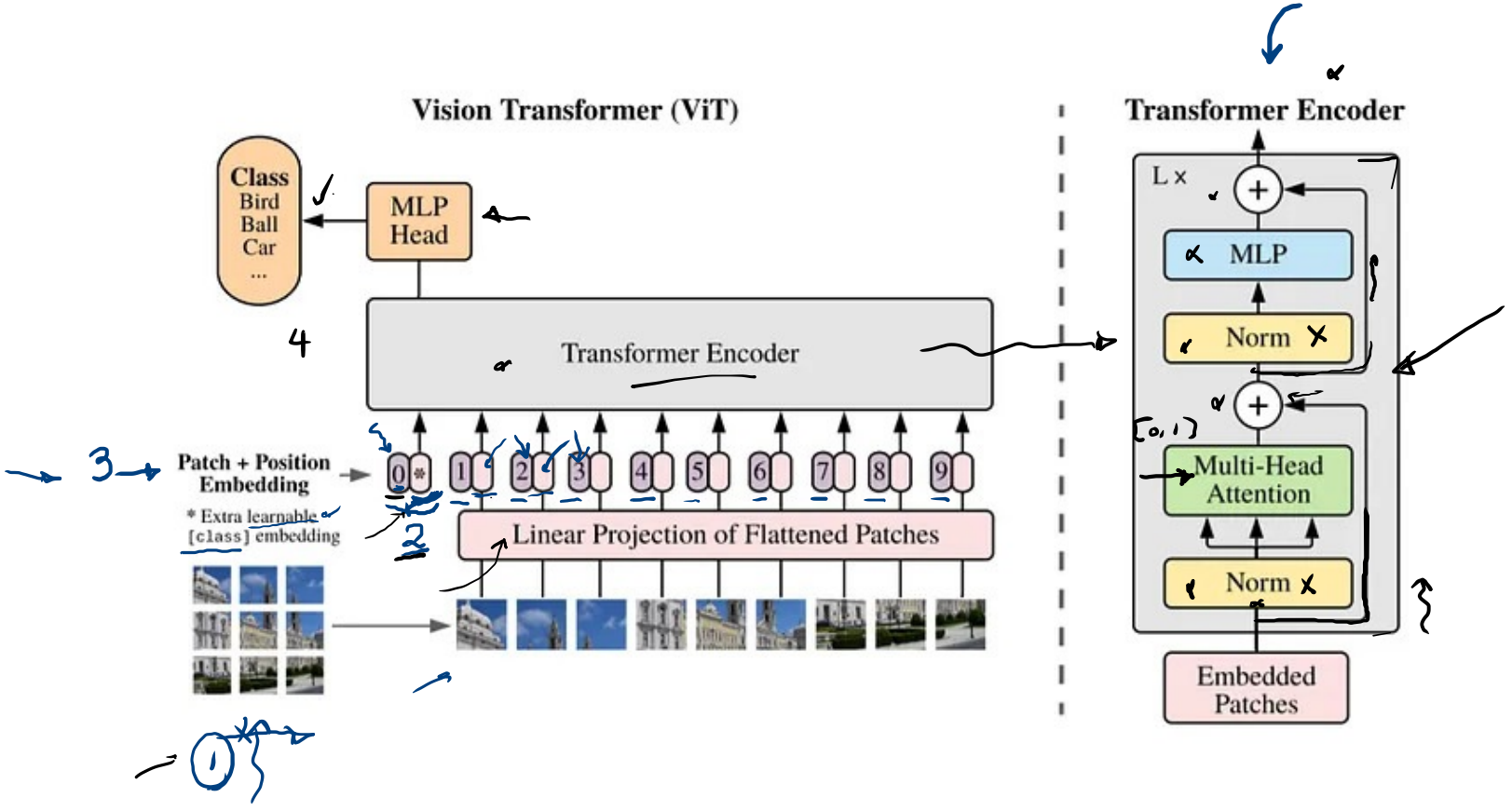
۱. ViT تشنه‌ی داده‌ها! → نیازمند دیتاست بزرگ هستیم!

ImageNet → عدم امکان کار

۲. Computation Cost.

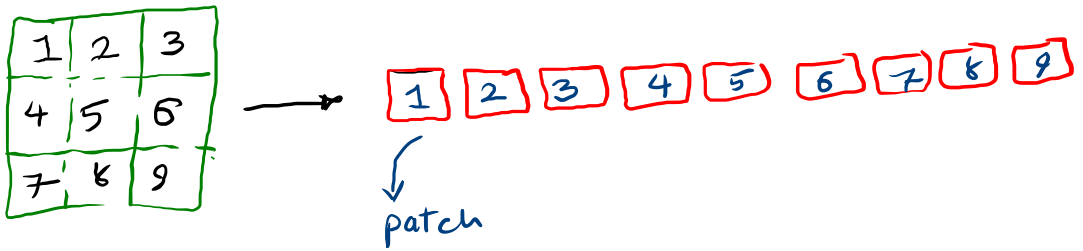
ViT >>> CNN

۳. هنوز بسیاری از معماری‌ها نتوانسته‌اند کارایی الگوریتم‌های CNN را داشته باشند!


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
Linear Projection of Flattened Patches
0
1
2
3
4
5
6
7
8
9
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

1. preprocessing!

Image to patches!

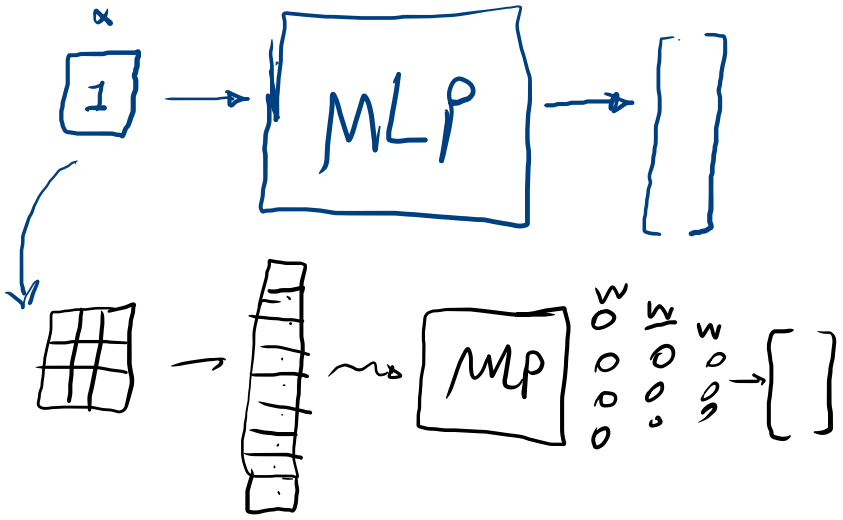


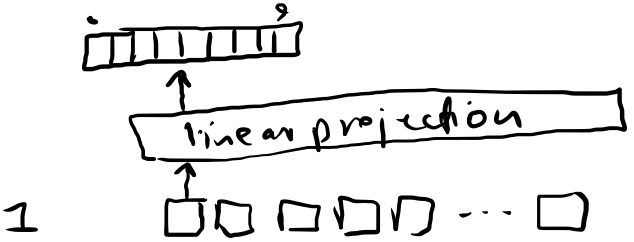
Conv2D (kernel size = stride)

2. Linear projection!

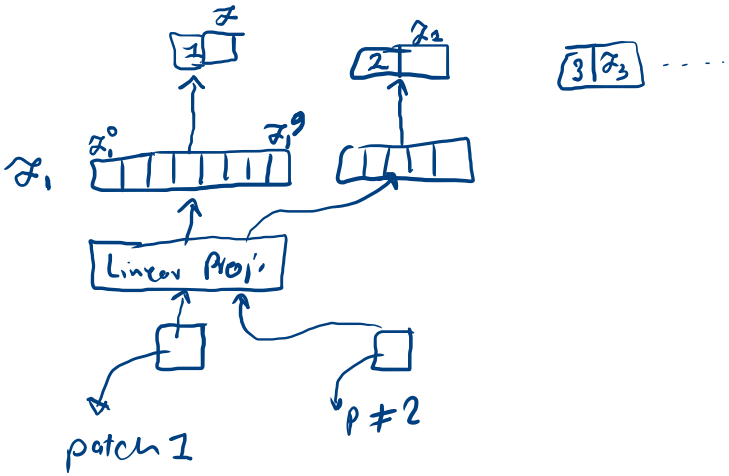
↳ تبدیل patch به

یک بردار ① بعدی!

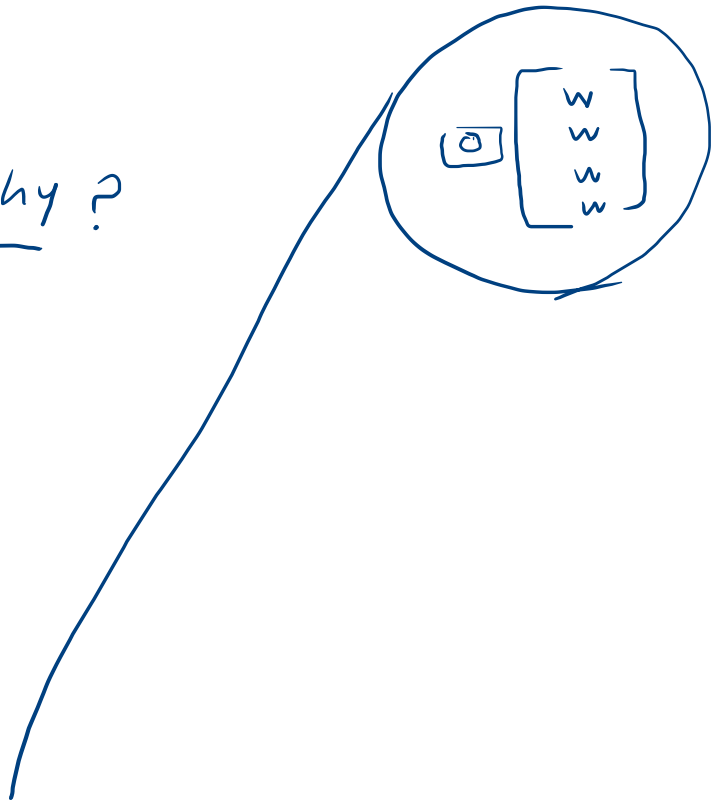


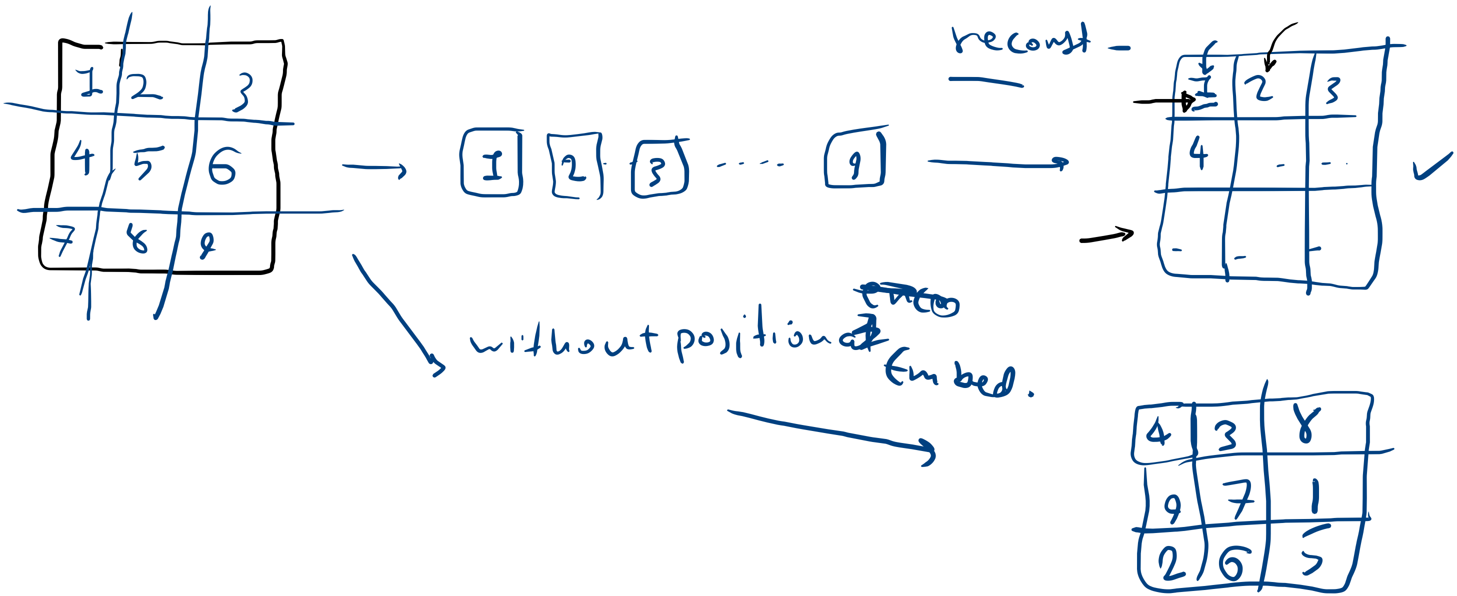


3. Position Embedding !



why ?





Conv ⇝ { scaling
rotation → not invariant.

Conv, pooling!

pool → scaling, invariant.

4. Transformer encoder!

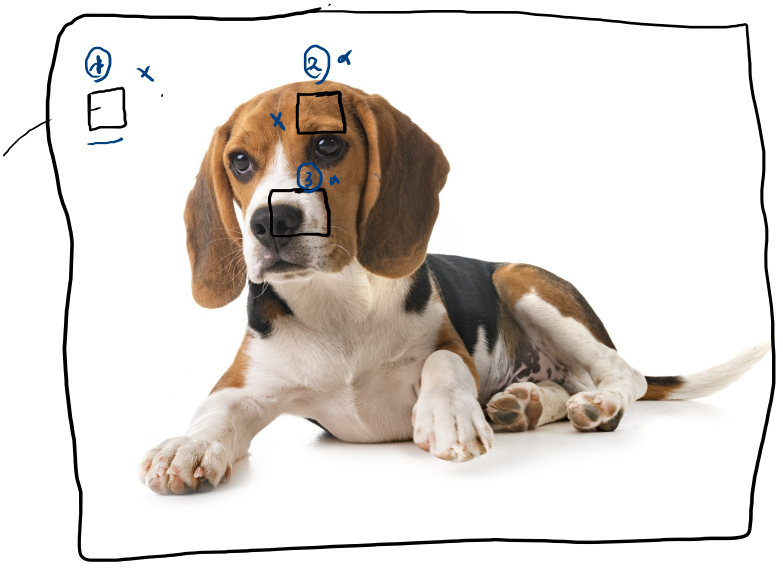
↓ Attention!

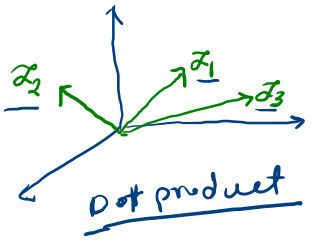
(.9, .8, .7)

x_1 — vector 3

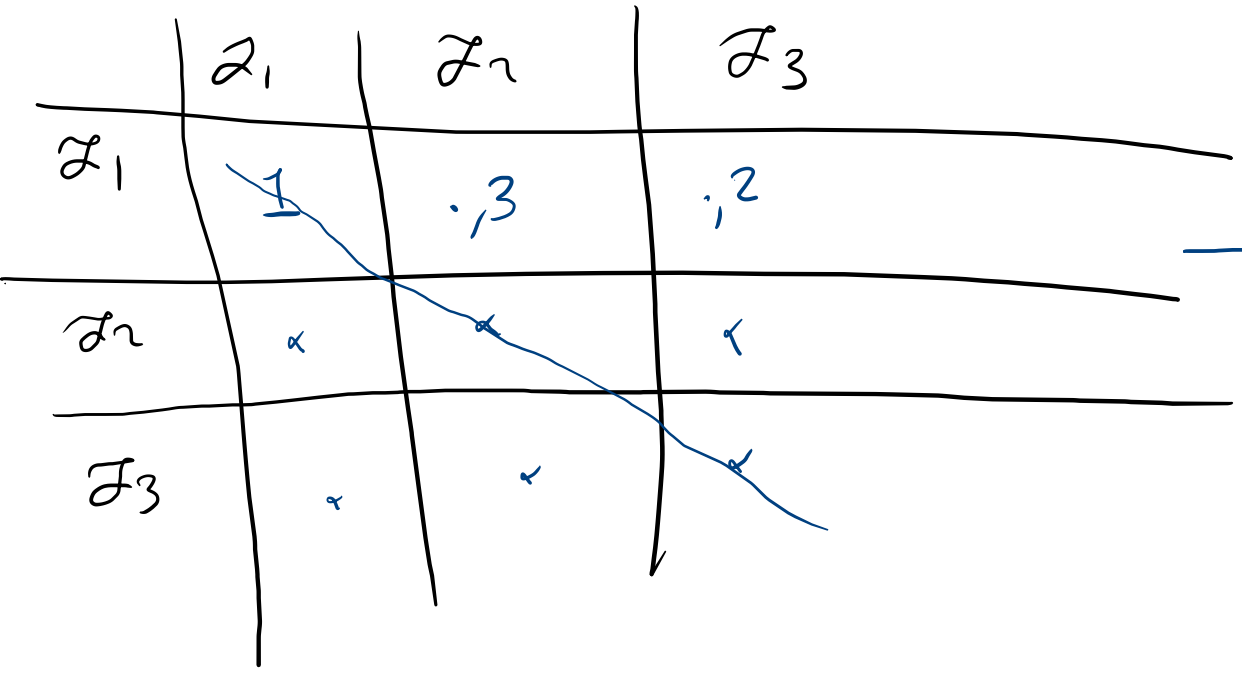
x_2 — vector

x_3 — vector

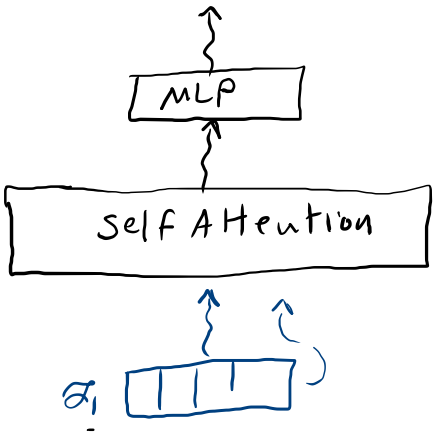




$x_2 x_1 \ll x_3 x_2$


x_1
x_2
x_3
z_1
z_2
z_3
1
.3
.2
✓
Attention!

(A)



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

Query: Feature of interest!

Key

Normalization factor

Value: original features to be scaled by probability!

$$Q = [x_1, x_2, \cdots, x_9]\, W^Q$$

$$K: \text{Key} = [x_1, x_2 \cdots, x_9]\, W^K$$

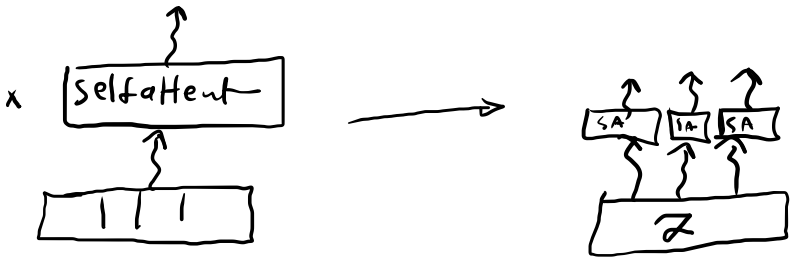
$$V: \quad [x_1, \cdots, x_9]\, W^V$$

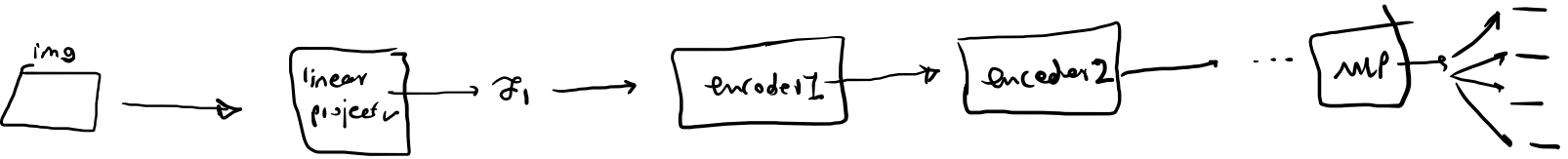
5. Improvement in Encoder.

↓

a. Normalization layer and skip connection!

b. multi head self attention! (MHSA)


x
SelfaHeut
SA
IA
SA
z


img
linear projection
$\mathcal{F}_1$
encoder1
encoder2
...
MLP

End