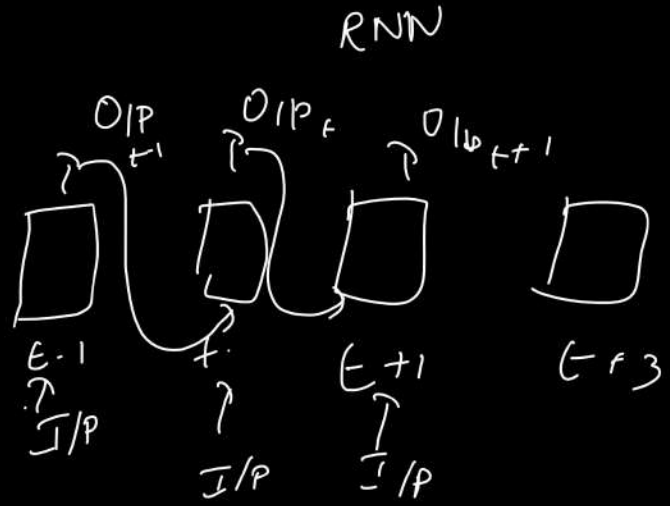
# ATTENTION IS ALL YOU NEED PAPER EXPLANATION AND IMPLEMENTATION FROM SCRATCH

TABLE OF CONTENTS






I like eating apple but I like using mobile phones from the apple company more.

128

{ apple : [ . . . . ]

orange : [ ] }

Disadvantage

1. No context awareness
2. Sequential processing × 3. Long context failure

↳ Parallel processing

Advantages

1. Generate context rich vectors by using Attention
2. Parallel processing
3. It can handle long contexts.

# TRANSFORMERS

-> Attention is All you Need - Google team - 2017.

=> Encoder
Decoder
Enc & Dec

-> Architecture Deep Dive.

Introduction -> Architecture Deep Dive -> Positional encoding

↓

Attention

↓

Decoder <- Encoder <- FFN <- Layer Normalization

↓

Transformer -> Dataset building -> Tokenizer

↓

Decoding strategies <- Attention visualization <- Training & Inference

How tokens are processed

# 2. TRANSFORMERS ARCHITECTURE IN DEPTH DEEP DIVE

Encoder → BERT
↳ Learn representations

Decoder → GPT
Decoder → Generation

Q/A, TC, TokL

Enc & Dec → BART, T5

Enc try to understand context
Will provide to decoder
↳ Decoder will try to complete the task accordingly

Token Embedding ⊕ → I/P

Positional Encoding

Attention
↳ Self → Multihead
Masked Multihead (Decoder)
↳ Cross

Layer Normalization (Residual connection)

FFN

Projection

Parallelization

# POSITIONAL ENCODING

Embeddings

→ Input embedding

→ Token embedding

→ Positional encoding

Token embedding :- Tokenizers.

dmodel = 512

The apple is tasty → bs → 1, seq_len → 4,

[The, apple, is, tasty]

1, 10, 8, 7

{ 1: [ ] 512 → vectors / embeddings

}

512

[ [ ],
[ ],
[ ],
[ ] ]

→ Token embedding

0 1 2 3 4 5 6 7

I had two apples 1 orange apple and another 1 red apple where the second one was slightly tastier it comes from Apple Inc.

apple → 6

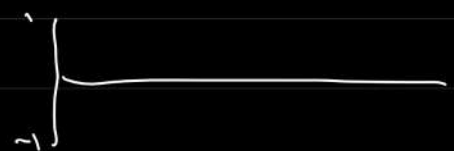
apple → 11

Sinusoidal

0, 1, 2, 3 $\boxed{512}$

→ Scaled in nature.

cosine, sin



Encodes relative position word.

sin( ) even
cos( ).

$$PE(pos, 2i) \rightarrow \sin\left(pos / \underbrace{10000^{2i/d_{model}}}_{6}\right)$$

0

i=0 frequency n

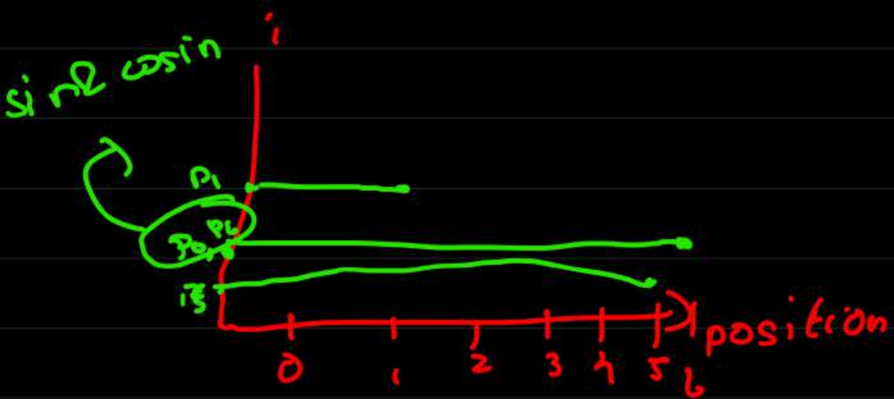
$$PE(pos, 2i+1) \rightarrow \cos\left(pos / 10000^{2i+1/d_{model}}\right)$$

1

## Why Sinusoidal

* Unique encoding of position of words
* Encode relative to other position
* Range → Normalized [-1, 1]

## Why sin & cosin

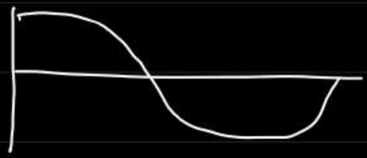



512 → embed
i=0
[ ]

sin

$\sin(0) = 0$

cos

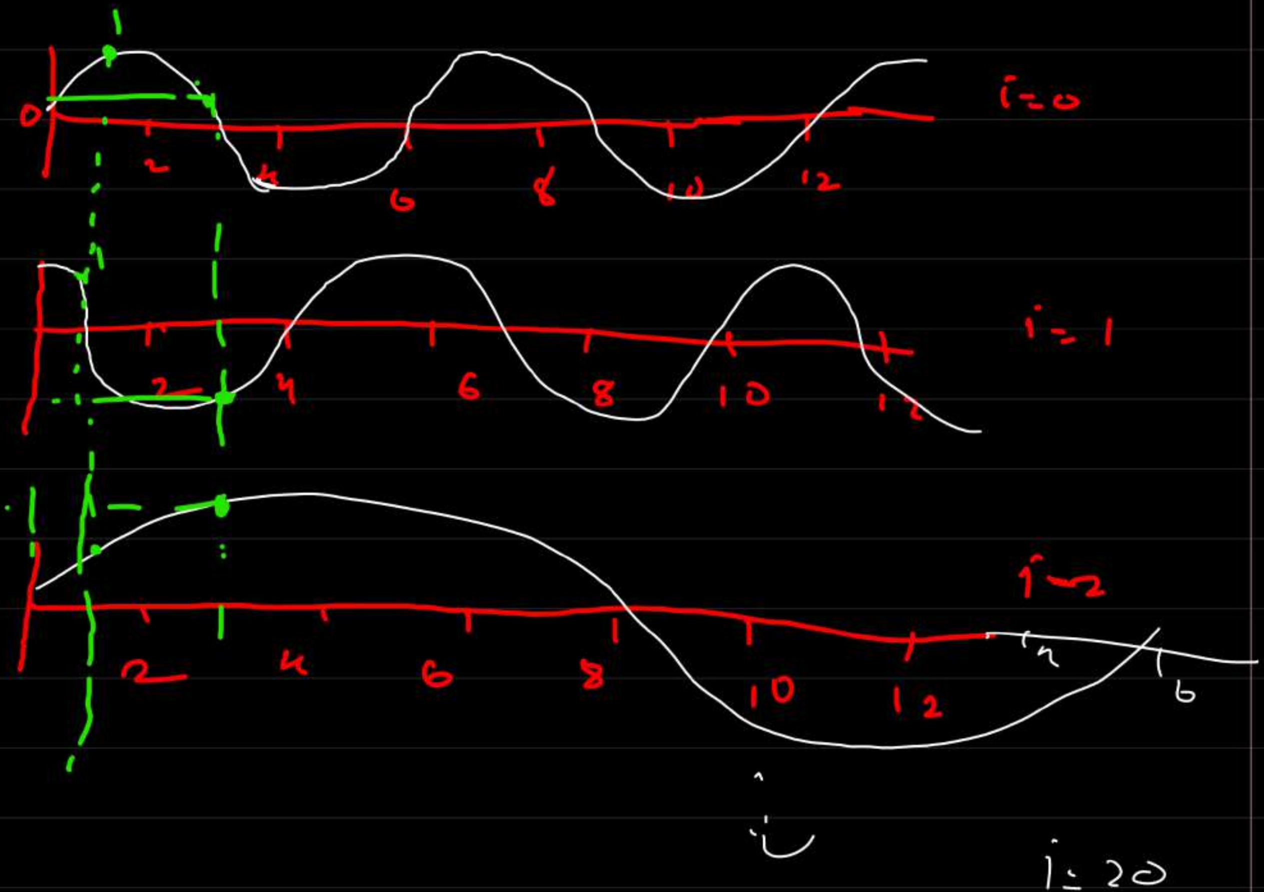


$\cos(0) = 1$

seq → 4

1

# DIMENSIONAL DEPENDENCY:-



Formula: -

$PE(pos, 2i) \rightarrow \sin(pos/10000^{2i/d_{model}})$

$PE(pos, 2i+1) \rightarrow \cos(\quad\quad 2i+1 \quad )$

The water is tasty → pos [0, 3]

$d_{model} = 4.$

i $n \rightarrow 100$

| | | 0 | 0 | 1 | 1 |
|---|---|---|---|---|---|
| pos 0 | The | sin(0) = 0 | cos(0) = 1 | sin(0) = 0 | cos(0) = 1 |
| 1 | water | sin(1/1) = 0.84 | cos(1/1) = 0.54 | sin(1/10) 0.10 | cos(1/10) ≃ 1.0 |
| 2 | is | | | | |
| 3 | tasty | | | | |

POS=0 i= 1

$(0/ - - -) \rightarrow 0$

POS = 1 i = 0 → i = 1

i = 0, pos = 1

sin(1/1)

cos(1/1)

$\sin\left(1/100^{2 \times 1/4}\right) \Rightarrow \sin\left(1/100^{1/2}\right)$

$= \sin(1/10)$

$\cos\left(1/100^{\cancel{2 \times 1 + 1}/4}\right) \rightarrow \cos\left(\frac{1}{10}\right) = 1.0$

512 → [seqlen , embed dim]

[ ]

I/P → Token Embed + Pos Encoding .

# ATTENTION

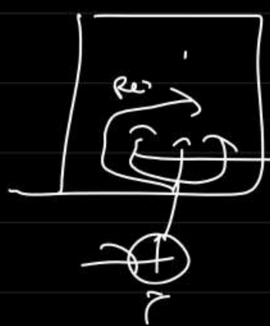
* Generation of context rich vectors
* Parallelization

My name is Vasanth → Scaled Dot Product Attention

| | My | name | is | Vasanth |
|---|---|---|---|---|
| My | * | * | - | + |
| name | * | * | - | * |
| is | - | + | * | * |
| Vasanth | * | * | - | * |

Increase seq length ↑ Use ↑

→ Generating context rich vectors
↳ Encoder → Decoder
(Generate)



Query, key & Value.

**Query :-**
Represents what you want

**Key :-**
The location of the answer you want

Value: -

The answer.

Eg:

Context: Vasanth is a youtuber who has bad handwriting → Encoder

Question: Who has a bad handwriting?

→ Encoder

→ Query → Q

Decoder → Question: Who has a bad handwriting?

Q → $Q^T K(\text{Vasanth}) \rightarrow 0.8$

$Q^T K(\text{is}) \rightarrow 0.01$

$\vdots \qquad 0.05$

$Q^T \text{handwriting}) \rightarrow \cdot$

1512

1

$$\text{Attention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_{model}}}\right) * V$$

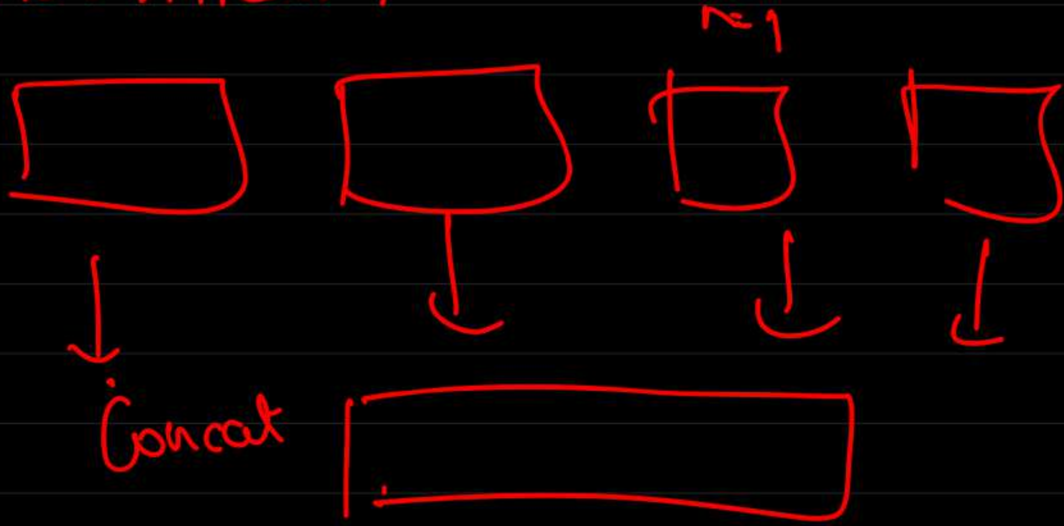
Vasanth.

| 0.9 | 0.01 | 0.05 | 0.02 | 0.02 |
|---|---|---|---|---|
| $\text{softmax}((Q \cdot K^T) / \sqrt{d_{model}}) \times V$ | $\frac{V}{Q \cdot K^T}$ | $\frac{V}{Q \cdot K}$ | $\frac{V}{Q \cdot K}$ | $\frac{V}{Q \cdot K}$ → |

$\Theta$

→ 1 head.

Multihead Attention




→ VASANTH

Why we divide $\sqrt{d_{model}}$ → Variance ??

↳ To reduce variance

MASKED MULTIHEAD ATTENTION

↳ Autoregressive

Vasanth has bad handwriting

(inf)

[[1, 0, 0, 0], → Teacher forcing

[1, 1, 0, 0]

[1, 1, 1, 0]

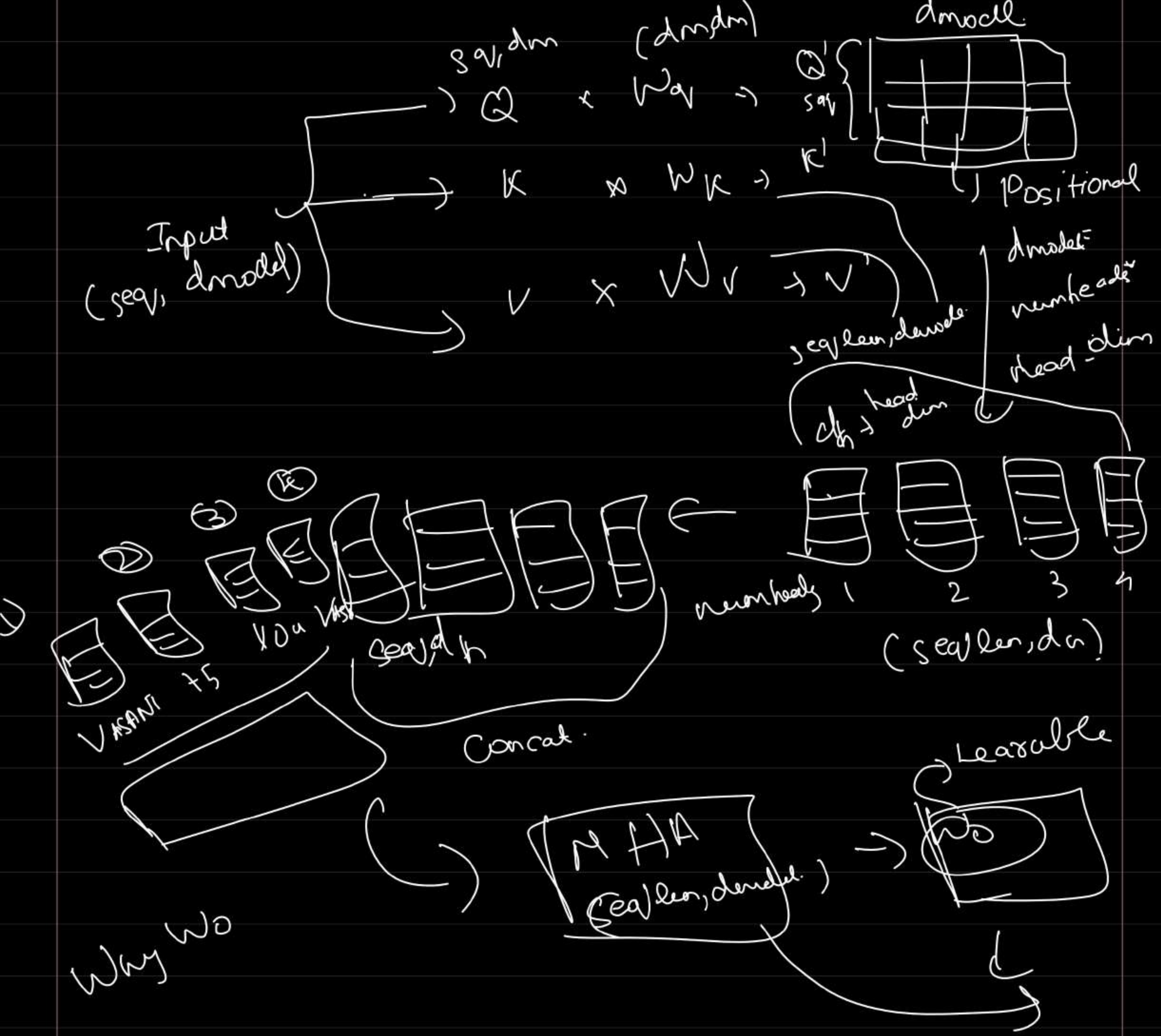
[1, 1, 1, 1]]

LOOK AHEAD MASK

PADDING MASK    10 -> 1, 8, 9, 17, 170

170, 170

Self Attention -> Cross Attention -> Decoder Architecture



# Flow

Input → Query × $W_Q$
└→ Key × $W_K$
└→ Value × $W_V$

↓

Split among heads

↓

Scaled dotproduct attention

At each head. $\text{softmax}\left(\frac{QK^T}{\sqrt{d_{model}}}\right) * V$

↓

Concat o/p of each head

↓

Project / Learn the concatenation

↓

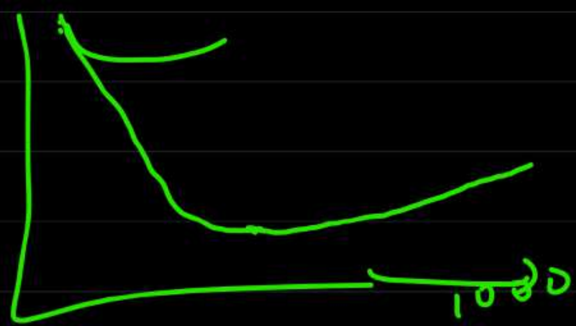
ATTENTION OUTPUT
(Context rich vectors)

# RESIDUAL CONNECTION AND LAYER NORMALIZATION

Add & Layer Norm → Paper → 1

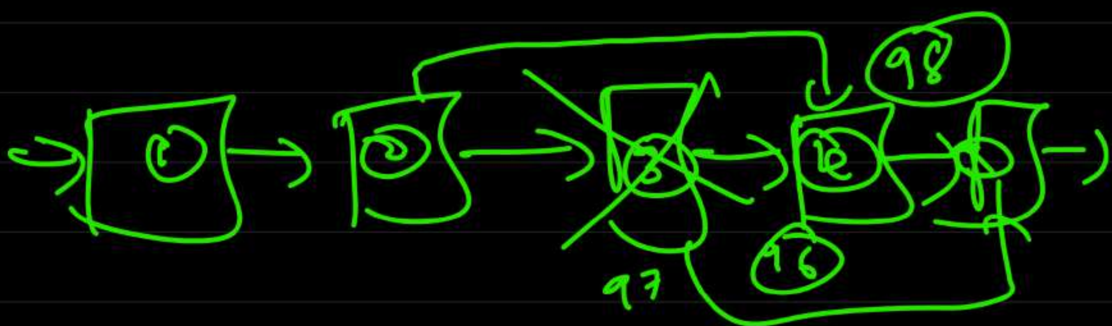
ResNet → 2

VGG 19, 16 → ImageNet → CNN.




Problem: On an extended training, loss increased

Soln: ResNet → Residual Network → Skip connection




if accuracy >= layer is considered

else
skip layer

# Add & Layer Norm


Add :-

## Layer Normalization

Formula: X → LayerNorm [Sublayer(x) + Output]

What?

Normalization is a process of scaling the values close to zero & around

[-3, 3] → [0, 1]

Why?

* Stable training
* Reach the global minima (optimum value) faster.
  ↳ Training becomes faster.

Layer Norm: .

$$x_i' = f[W^T x_i + b_i] \rightarrow \text{Forward O/P}$$

Model will calc.

$$y = \gamma_i \boxed{\left[\frac{x_i' - \mu_i}{\sigma_i + \epsilon}\right]} + \beta_i$$

→ Standardication

→ ( $W^T x + b$ | → Act() ) → $X_i'$

$\gamma_i$, $\beta_i$ → learnable

Result {
All the values in this layer
will become [-3 to 3]
mean = 0, std → 1

token → 2 xemb → 3

$$\begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.2 & 0.3 \end{bmatrix}$$

$$\mu_{11} = \frac{1}{3}[0.1 + 0.2 + 0.3] = 0.2$$

$$\mu_{21} = \frac{1}{3}[0.9] = 0.3$$

$$\sigma_{11} = \sqrt{\frac{1}{3}\left[\left[0.1-0.2\right]^2 + (0.2-0.2)^2 + \left[0.3-0.2\right]^2\right]}$$

$$= \sqrt{\frac{1}{3}\left[0.02\right]} = \sqrt{0.066}$$

$$= 0.08$$

$$\sigma_{21} = 0.08$$

$$\text{mean } \mu = \begin{bmatrix} \mu_{11} \\ \mu_{21} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$$

$$\text{std } \sigma = \begin{bmatrix} \sigma_{11} \\ \sigma_{21} \end{bmatrix} = \begin{bmatrix} 0.08 \\ 0.08 \end{bmatrix}$$

$$Y = \begin{bmatrix} \frac{0.1-0.2}{0.08} & \frac{0.2-0.2}{0.08} & \frac{0.3-0.2}{0.08} \\ \frac{0.4-0.3}{0.08} & \frac{0.2-0.3}{0.08} & \frac{0.3-0.3}{0.08} \end{bmatrix}$$

$$\textcircled{\scriptsize{}} \ \sigma_1 = \gamma Y + \beta$$

$$Y = \begin{bmatrix} -1.2248 & 0 & 1.2248 \\ 1.2248 & -1.2248 & 0 \end{bmatrix}$$