ChatGPTの仕組みを理解したい！３ Transformer編

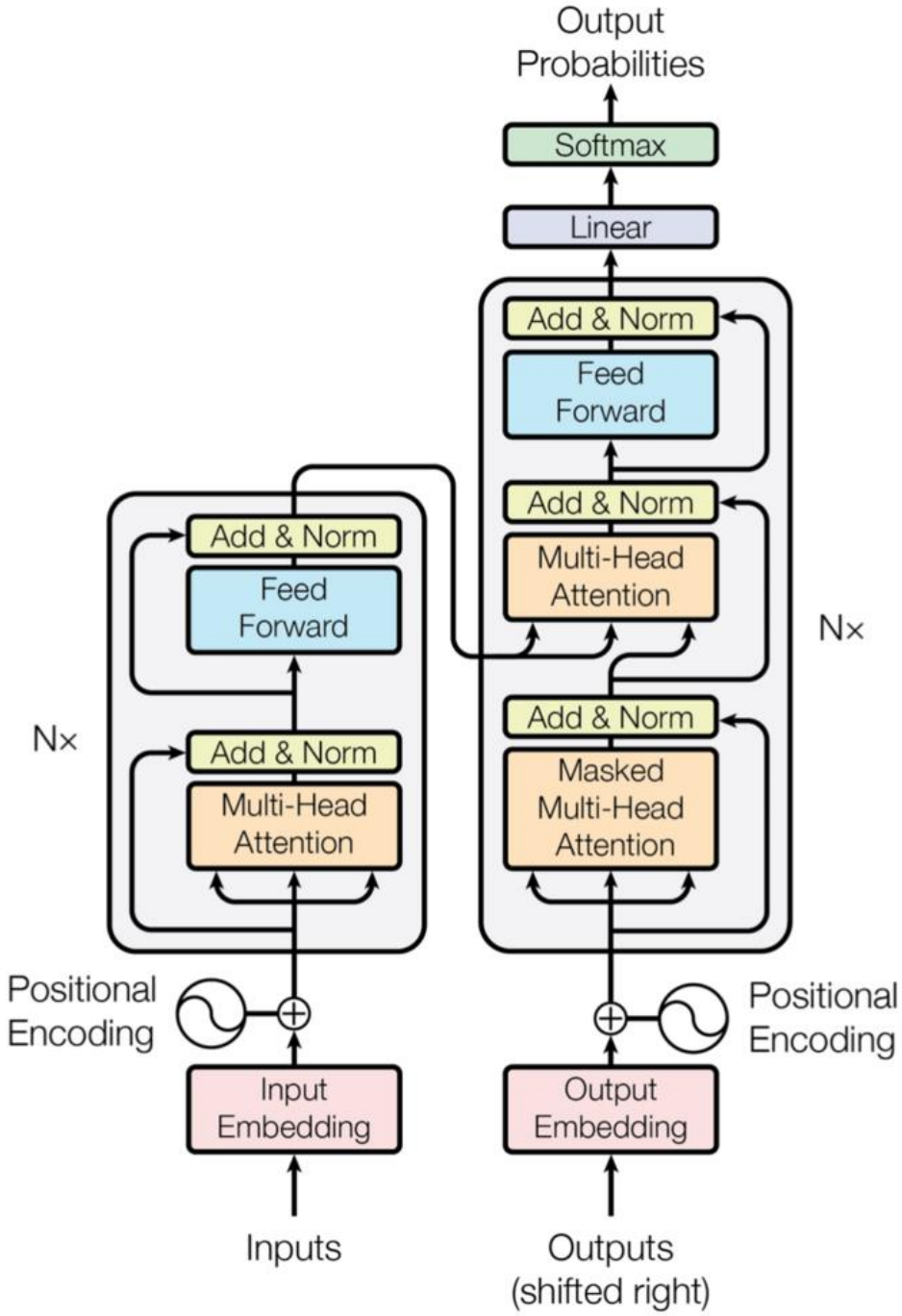
坂井吉弘 (@sakarush)

Transformer の全体像

- こんな感じのモデル

- 重要な要素
  - Self-Attention
  - Multi-Head
  - 残差接続
  - 位置埋め込み

- 入出力
  - 入力：単語の集合（文章）
  - 出力：辞書単語の可能性の予測値



Ashish Vaswani, Et Al., Attention Is All You Need. NIPS 2017

Self-Attention

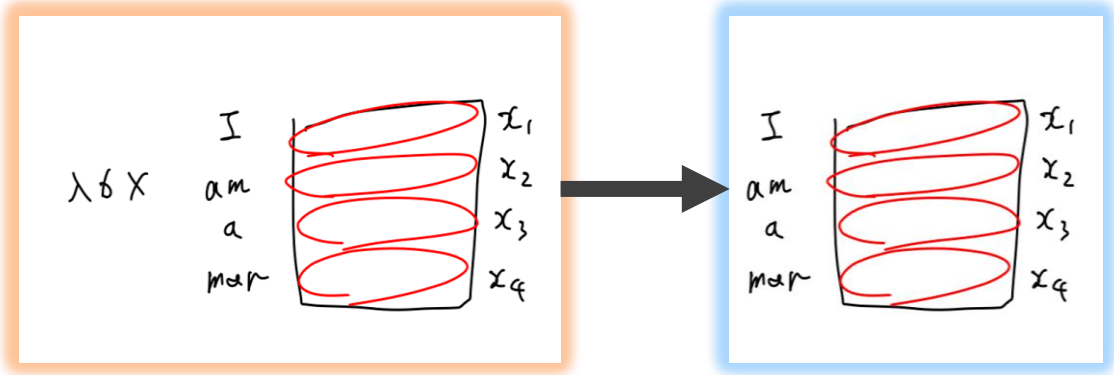
Attention is All You Need

- 2017年の論文
- Transformer の最重要要素「Self-Attention」はAttentionの発展版
  ここでは Self-Attention だけを紹介します
- 今回の目標は以下の数式を理解すること

$$Z = \mathrm{softmax}\left(QK^T\right)V$$

Self-Attention がやること

- 入出力
  - 入力：埋め込みベクトルの集まり（行列）
  - 出力：埋め込みベクトルの集まり（行列）
    → Self-Attentionはベクトルの加工機



- 入力から3つの別々のベクトルを生成して、出力ベクトルを再構築する
  - Query
  - Key
  - Value

Self-Attention の流れ

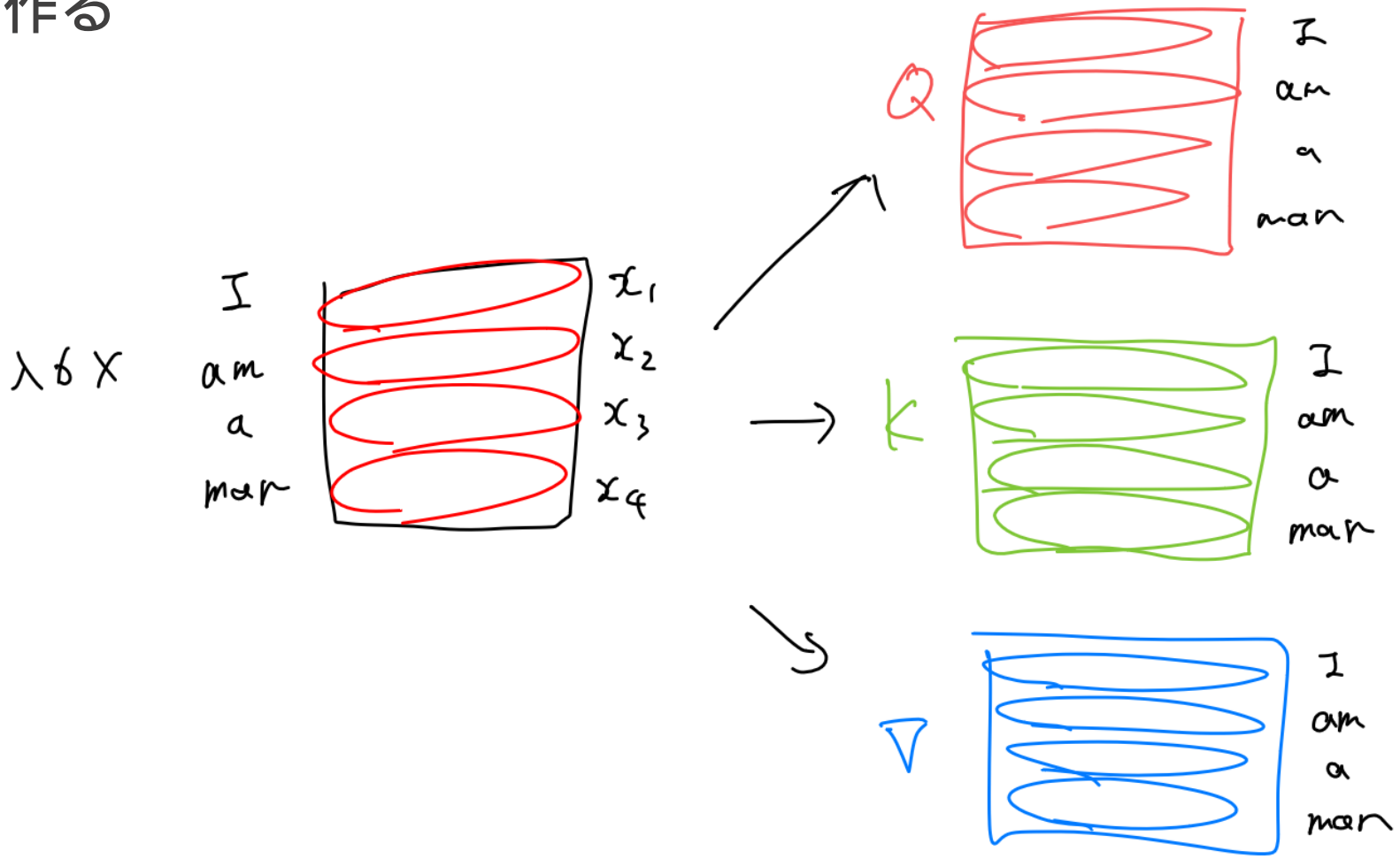
1. 入力 X から Q,K,V を生成
   - $Q = W^{(Q)}X$
   - $K = W^{(K)}X$
   - $V = W^{(V)}X$

2. QueryとKey の
   関連性を調べる
   - $A = QK^T$

2. Q-K 間の関連性の値をもとに
   Value の重み付け和を取る
   - $Z = \mathrm{softmax}(A)V$

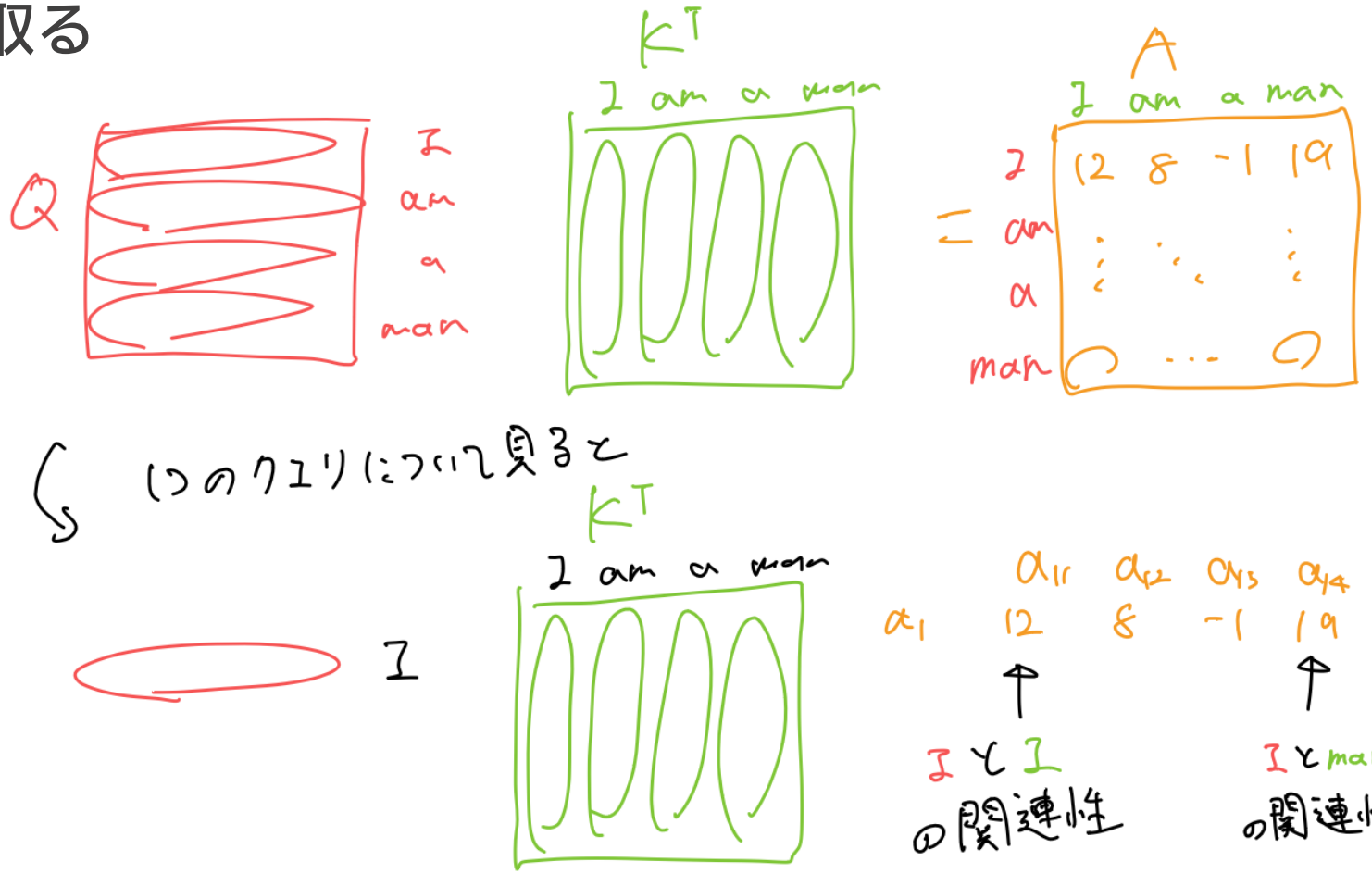
Step 1　入力から QKV を生成する

- 重み行列Wをかけて作る
  - $Q = W^{(Q)}X$
  - $K = W^{(K)}X$
  - $V = W^{(V)}X$



Step 2　Query と Key の関連性を調べる

- QueryとKeyの内積を取る
  - $A = QK^T$

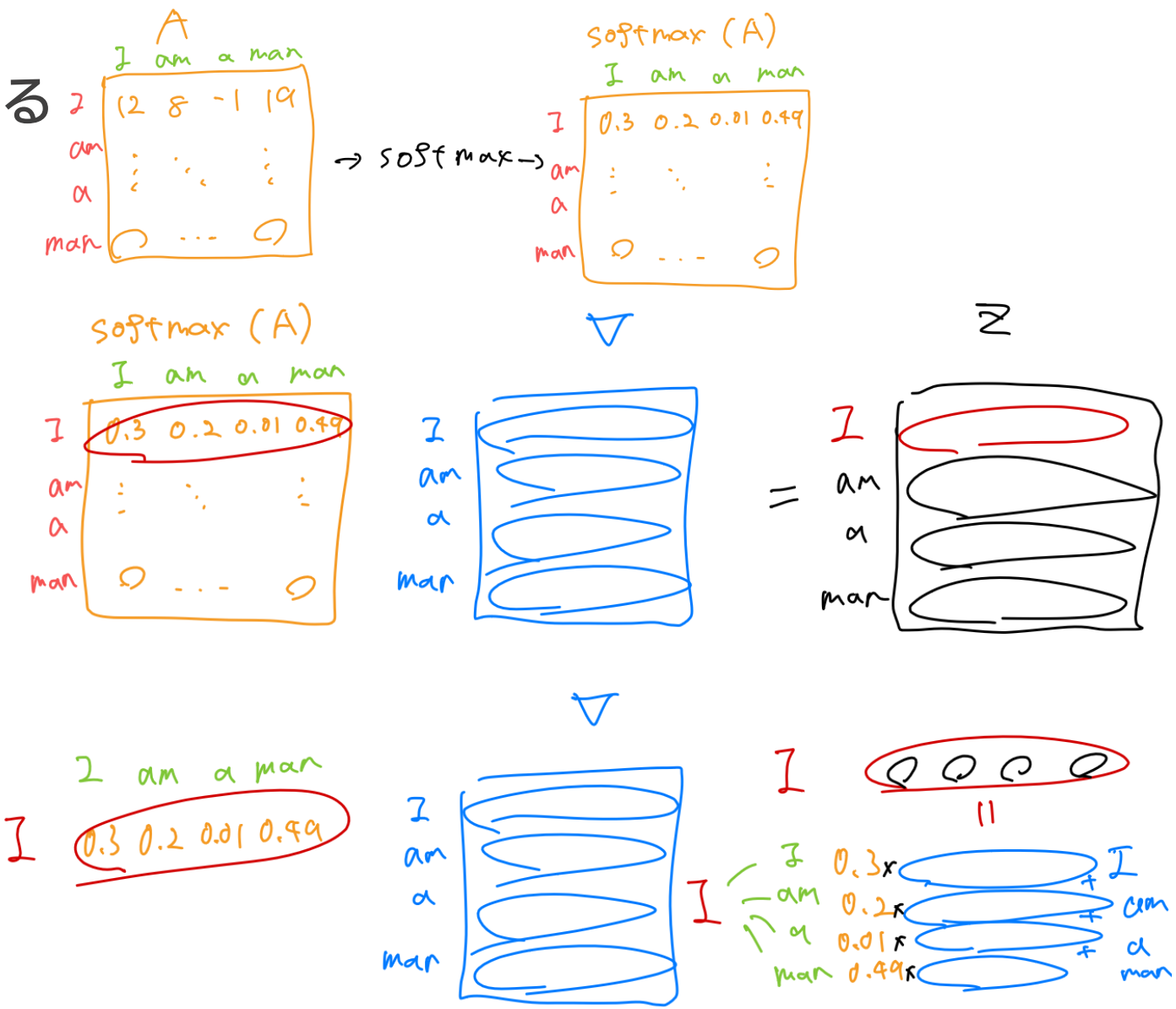


Step 2　Query と Key の関連性を調べる

- Softmaxでならしてをかける
  - $Z = \mathrm{softmax}(A)V$

- Softmaxは和を1にする
  - (12, 8, -1, 19)
    ↓
    (0.3,0.2,0.01,0.49)

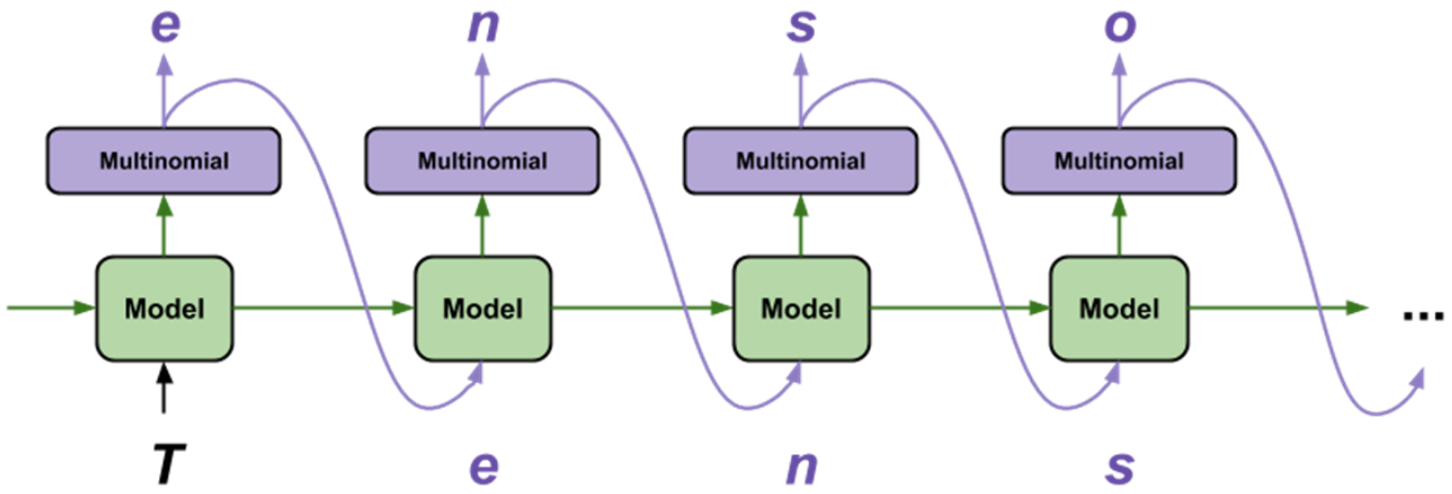
  - 確率っぽく出来る



Self-Attention の利点

- 単語間の関係を考慮出来る
  - 前回のWord2Vecでは考慮できなかった

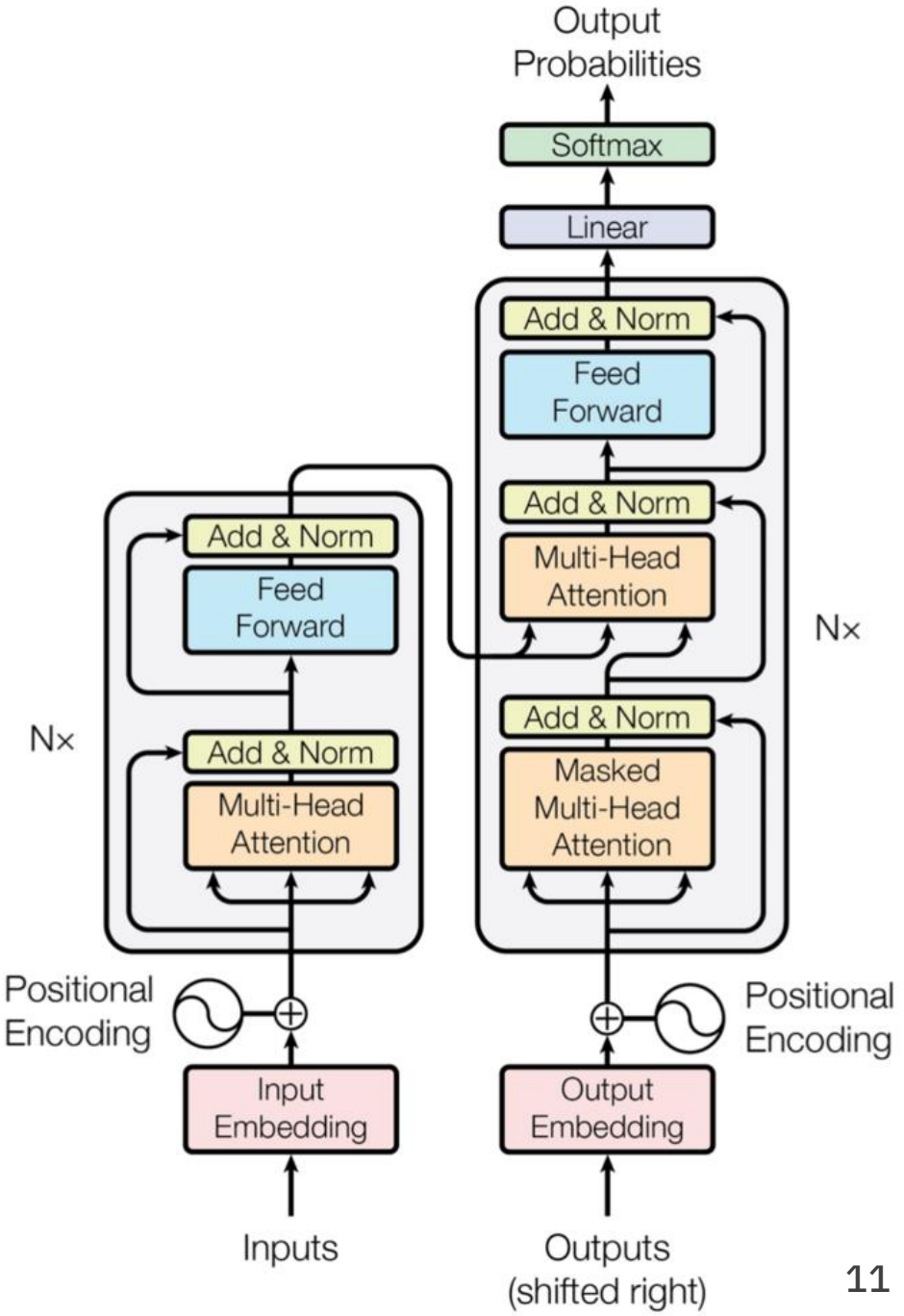
- ただの行列計算なので、文章全てを一度に並列計算可能
  - RNNでは順次計算するので、並列にできなかった



https://blog.tensorflow.org/2019/02/mit-deep-learning-basics-introduction-tensorflow.html

Multi-Head 化

- Self-Attentionでやっていることを分割する
- 並列計算がしづらくなるが、表現力が上がる
- これを施したAttention が Multi Head Attention



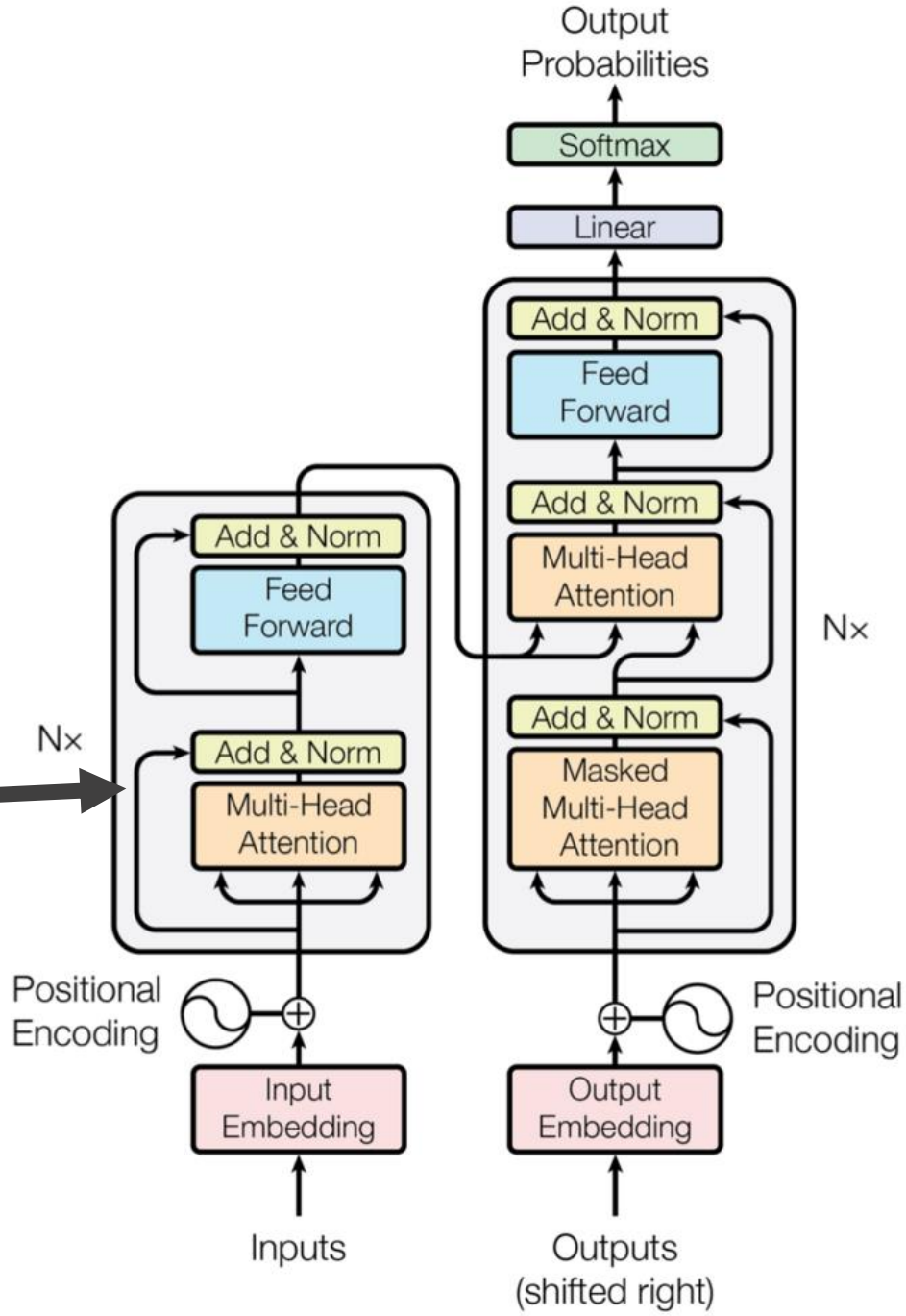
Ashish Vaswani, Et Al., Attention Is All You Need. NIPS 2017

その他の重要要素

Transformer の全体像

- 重要な要素
  - Self-Attention
  - Multi-Head
  - 残差接続
  - 位置埋め込み

- 残差接続
  - これ
  - 学習をうまくやるために用意する

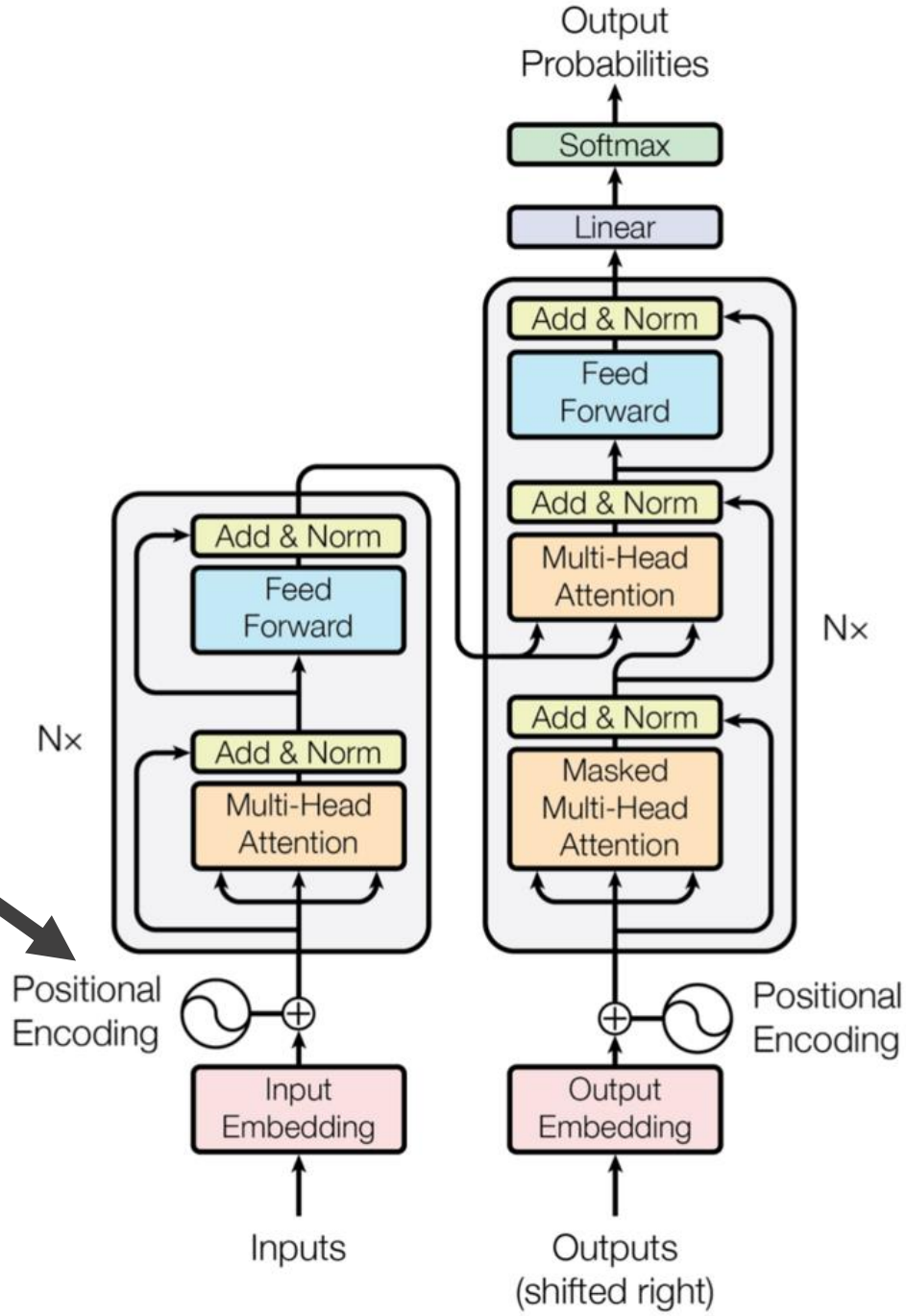


Ashish Vaswani, Et Al., Attention Is All You Need. NIPS 2017

Transformer の全体像

- 重要な要素
  - Self-Attention
  - Multi-Head
  - 残差接続
  - 位置埋め込み

- 位置埋め込み
  - Word2Vecでは位置情報がなかった
  - “I” が 文頭にあるのか、2単語目にあるのか…
  - 位置の情報をベクトルに変換して加えること
  - 結構面白い技術だが、説明するには余白が



Ashish Vaswani, Et Al., Attention Is All You Need. NIPS 2017

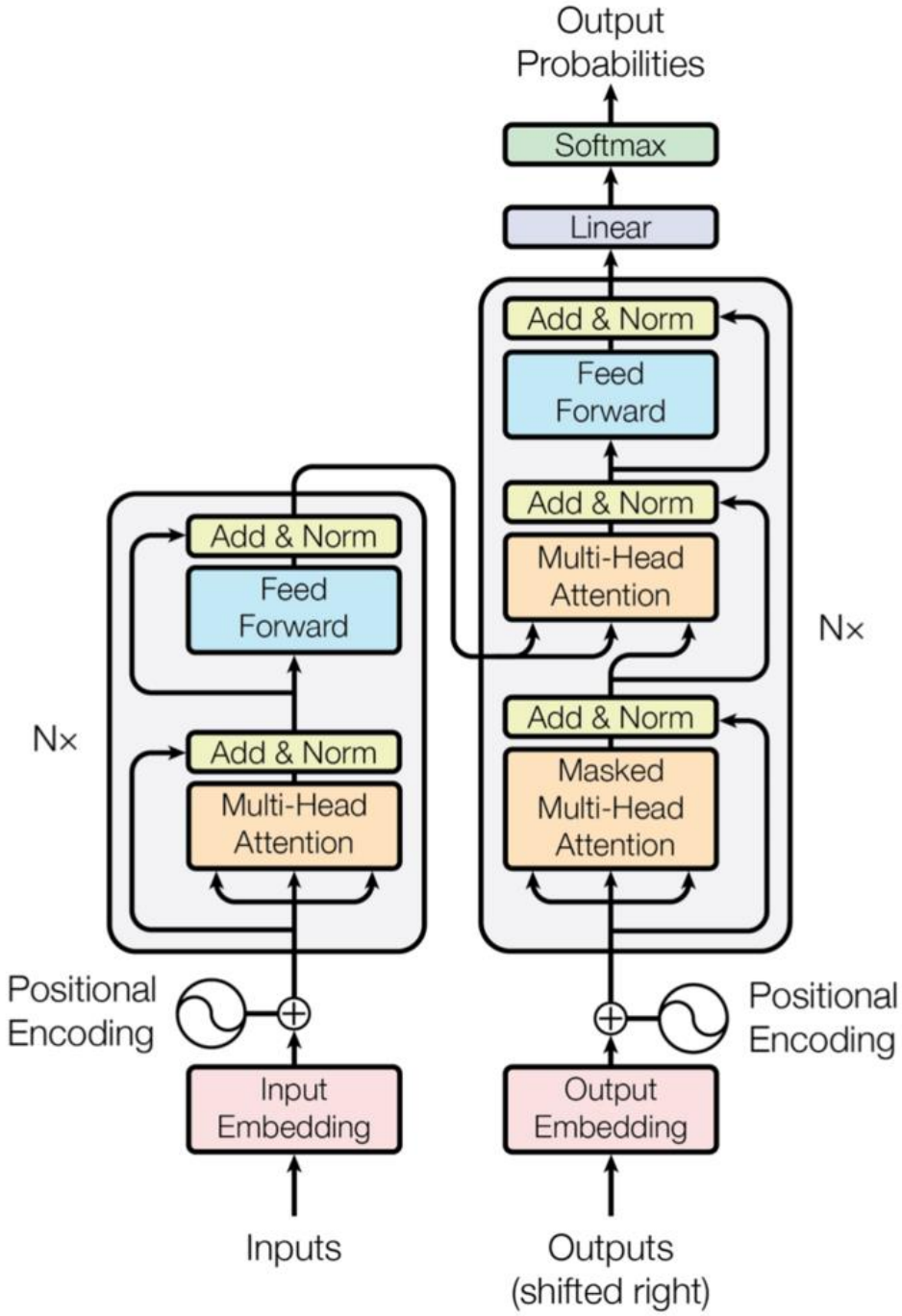
Transformerをながめる

Transformer の流れ

1. 入力を埋め込みベクトルに変換
   - 埋め込みベクトルを集めた行列として表現

2. 単語の位置の情報を埋め込む

3. N回繰り返す
   - Self-Attentionして正規化
   - Feed Forward して正規化



Ashish Vaswani, Et Al., Attention Is All You Need. NIPS 2017

翻訳の仕方

- 入力した文（の言語）でKとVを作る
  （こいつがエンコーダー）

- デコーダーでは予め作ったKとVと
  自分がこれまで出力してきた文章を使う