TRANSFORMER(＆ATTENTION)がとってもよく分かる資料

岡田　隆之

目次

- ◆TRANSFORMER概要
- ◆TRANSFORMERアルゴリズム解説
- ◆可視化時の注意点

◆TRANSFORMER概要

TRANSFORMERとは

- **Transformer**とは、2017年に発表された“Attention Is All You Need”という自然言語処理の論文で登場したDeap Learningモデル。Attentionという計算アルゴリズムを使用（後述）。
- それまでは単語一つ一つを順番に取り扱うRNNモデルが主流だったが、文章全体をいっぺんに取り扱うTransformerモデルに主流が移り変わり、主に自然言語処理において中心的な役割を持つようになった。並列処理もできるため、処理も早い。
- 自然言語処理の文章のように、データが順番に流れていくような処理に強く、音声解析や**時系列解析**、最近では画像処理においても広く使用が広まっている。大変人気。

TRANSFORMERの成果

- 自然言語だと100%の正解がないため、正解率が示しにくい。例えば画像処理においては次のような成果が出ている。AI画像分類でも有名なデータセットTransformerを使ったものがTop！！
  ＊ViT=Vision Transformer
- 以下、論文での報告：https://openreview.net/pdf?id=YicbFdNTTy

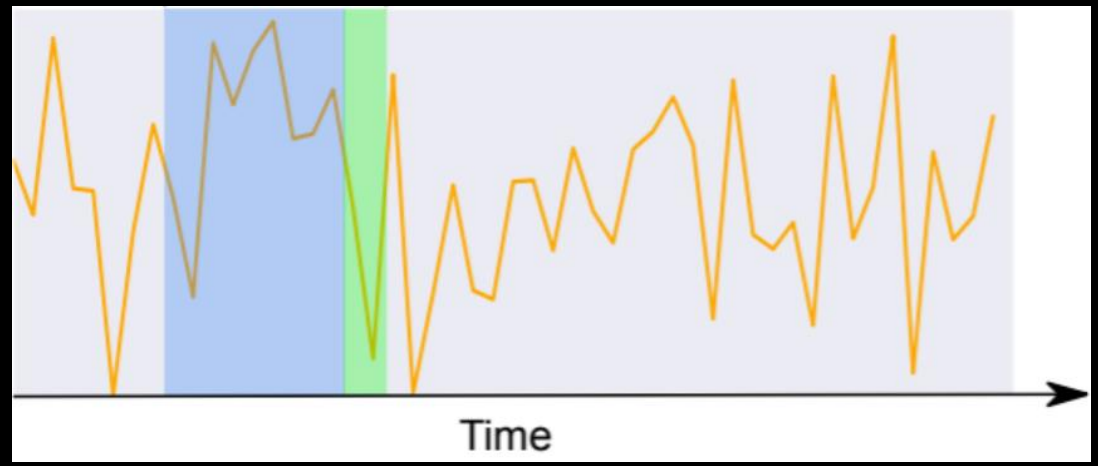
	Ours (ViT-H/14)	Ours (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.36	87.61 ± 0.03	87.54 ± 0.02	$88.4/\mathbf{88.5}^*$
ImageNet ReaL	**90.77**	90.24 ± 0.03	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.63 ± 0.03	–
VTAB (19 tasks)	**77.16** ± 0.29	75.91 ± 0.18	76.29 ± 1.70	–
TPUv3-days	2.5k	0.68k	9.9k	12.3k

TRANSFORMERの時系列データへの適用

- 時系列データに対してもさまざまなところからTransformerによる制度改善が報告されている。例）上図

- また、右下図のように、分析にどこが重要ににになったのか教えてくれるようにすることもできる。

Model	Pearson Correlation	RMSE
ARIMA	0.769 (+0 %)	1.020 (-0 %)
LSTM	0.924 (+19.9 %)	0.807 (-20.9 %)
Seq2Seq+attn	0.920 (+19.5 %)	0.642 (-37.1 %)
Transformer	0.928 (+20.7 %)	0.588 (-42.4 %)



◆TRANSFORMER
アルゴリズム解説

TRANSFORMERの構成図

- 右がDeap LearningモデルのTransformerの構成図。
- Transformerの構成はおおよそ①、②、③の3つの部分に分類されるが、3つとも全部Attentionを使っているところが重要。RNNなどの活用はない。それぞれの処理は以下：

①⇒入力分の単語間の関係性を理解

②⇒出力したところまでの単語間の関係性の理解

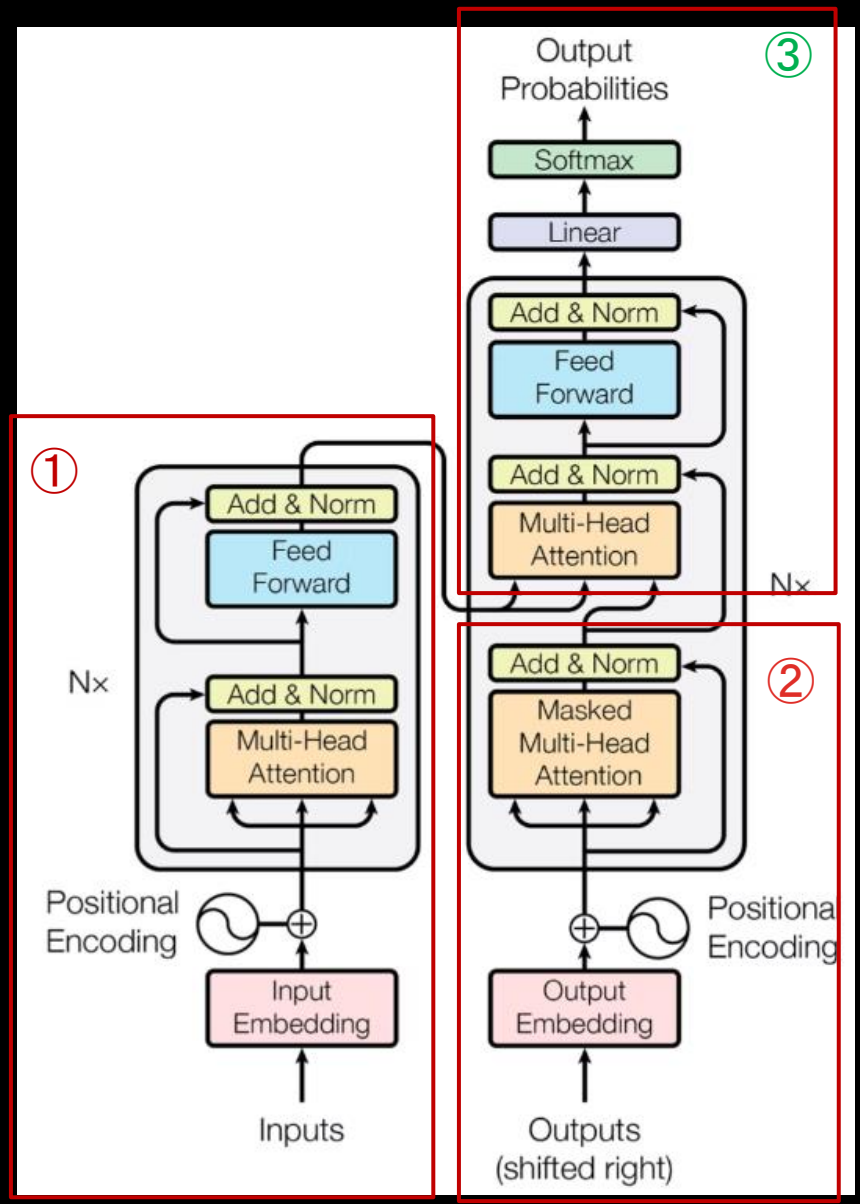
③入力と出力の関係性を理解

- そして、利用されているAttentionの詳しい技術は以下：

・Self-Attention

・Multi-Head Attention

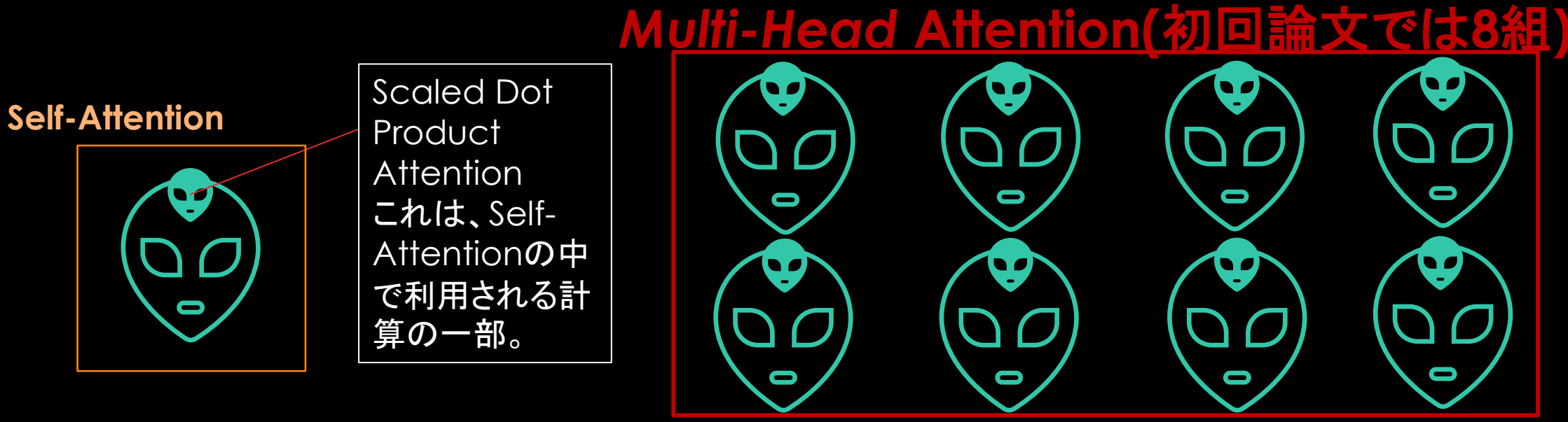
・Scaled Dot-Product Attention



3つのATTENTIONの関係性

- 前ページでAttentionが3種類あるように書いたが、実は、

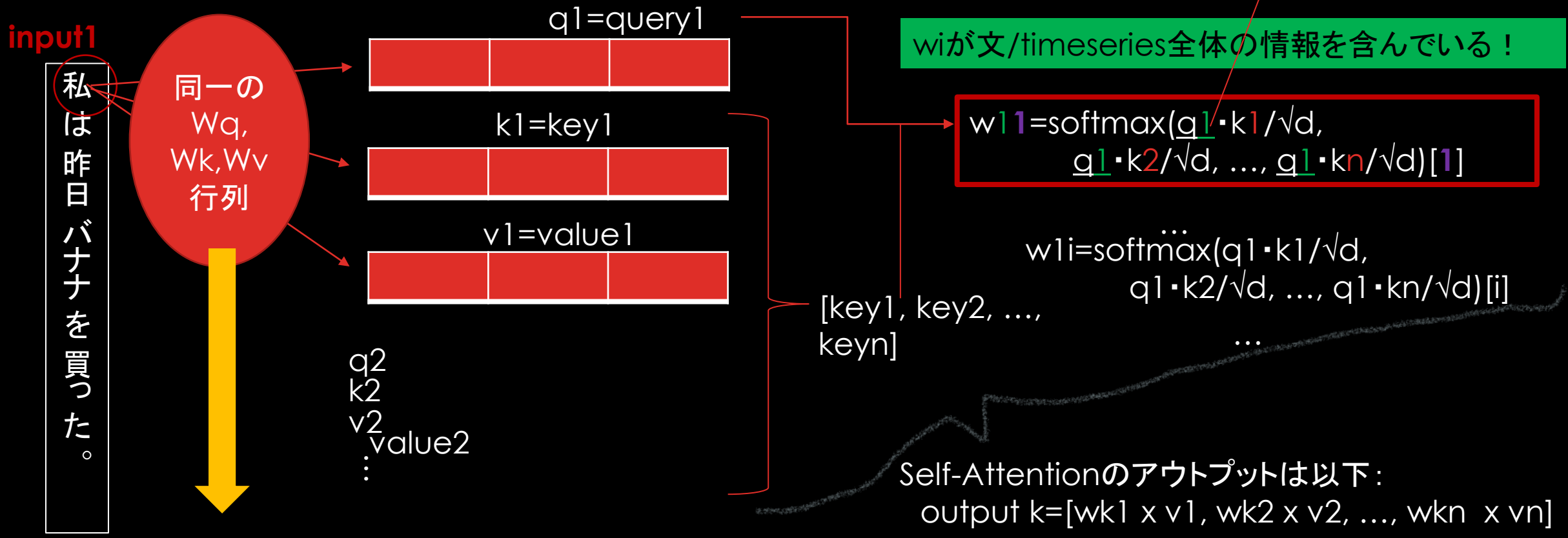
Scaled Dot-Product AttentionはSelf-Attentionの中で使用するAttention活用に向けた内積化技術。
Multi-Head AttentionはSelf-Attentionの入り口(入力の部分)を増やしたもの。
なので、内容的にはSelf-Attentionの部分しかなく、実装的にはより強化されたversionのMulti-Head Attentionを使えばよい。



SELF-ATTENTIONの計算解説

内積をとっている。この部分がScaled-Dot Product Attention

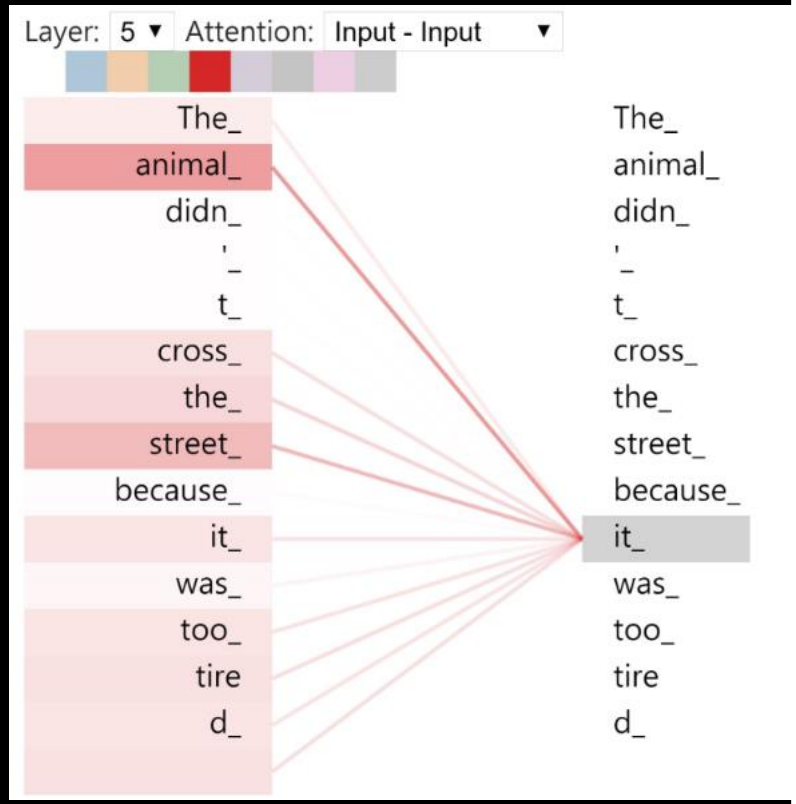
- 前ページでみたように、Multi-Head Attentionは入力部分を増やしただけなので、Self-Attentionさえ理解できれば良いだろう。次のようになる。

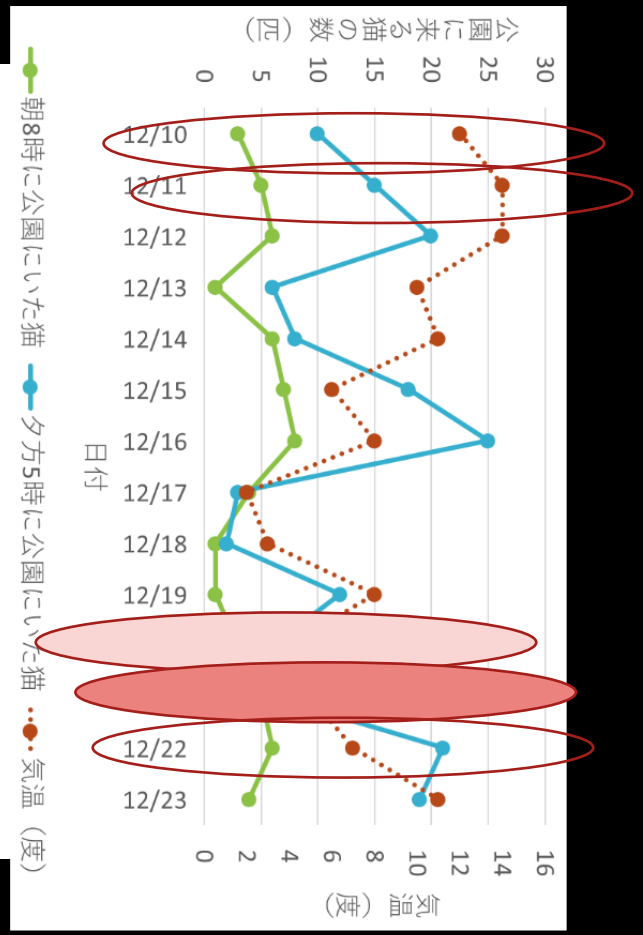


◆可視化時の注意点

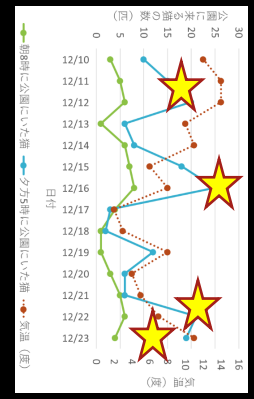
（注意点）時系列の1点1点に印付けすることはできない

[high, low, close](t番目)などの組が単語ひとつ(t番目)に相当。
⇒単語ひとつ≒組の重要度はわかるが、値一つ一つの重要度は出せるか？？
⇒計算方法的に難しい





点ごとには無理



重要度の割り出しは時間ごと＆全体としての列になる。
(例：2021年8月15日の変動が重要！など、その日/その時点の値の組が重要であることを教えてくれる。)