

『AI・データサイエンスのための数学入門』学習補助資料

attention を計算と図で理解する

内積で照合し、重みを作り、情報を混ぜる

1 query・key・value は何が違うのか

図書館で資料を探す場面を想像すると、query は「検索の条件」、key は「照合に使う見出し」、value は「受け取る内容」にたとえられます。これは役割を理解するための比喻です。モデルが人間の言葉で質問や見出しを作っているわけではありません。

ベクトル	計算上の役割	次元
query q	参照先の key との内積を計算する。	d_k
key k_i	query との照合に使い、重みを決める。	d_k
value v_i	得られた重みに従って混ぜる。	d_v

query と key は内積のために同じ次元をもちます。value の次元は、それらと異なっても構いません。本文の d は、ここでは d_k に当たります。

重みを決める段階と、内容を渡す段階を分ける

$$s_i = \frac{q^\top k_i}{\sqrt{d_k}}, \quad \alpha_i = \frac{e^{s_i}}{\sum_j e^{s_j}}, \quad c = \sum_i \alpha_i v_i.$$

この一回の計算で q, k_i を固定し、 v_i だけを変えれば、重みは同じままで出力が変わります。逆に value を固定しても、query が変われば重みが変わり、別の混ぜ方になります。

「内積が大きい」の意味

$$q^\top k_i = \|q\| \|k_i\| \cos \theta_i.$$

内積には向きだけでなく長さも影響します。したがって、内積をそのまま「意味の似ている度合い」やコサイン類似度と同一視はできません。学習された変換のもとで、情報を選ぶためのスコアとして働きます。

2 種類の softmax を区別する

attention の softmax は**参照位置**に重みを割り当てます。一方、出力層の softmax は**候補トークン**に確率を割り当てます。attention の出力 c は通常、確率分布ではなく特徴ベクトルです。

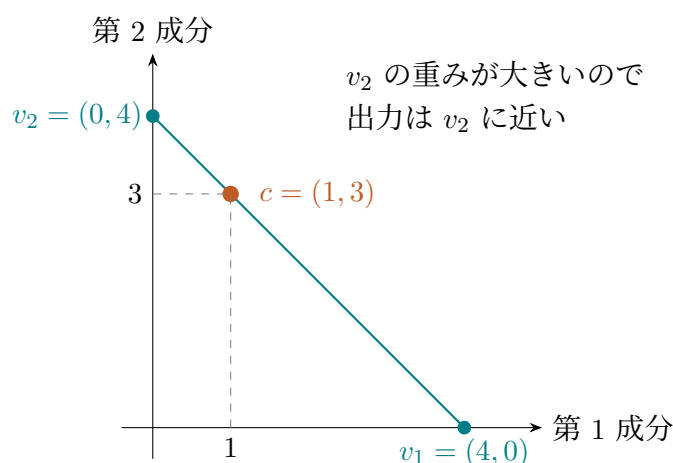
2 重み付き和は、value の間の点になる

$d_k = 1$ 、 $q = 1$ 、 $k_1 = 0$ 、 $k_2 = \log 3$ とします。 $\sqrt{d_k} = 1$ なので、関連度は $(0, \log 3)$ です。よって

$$(\alpha_1, \alpha_2) = \left(\frac{1}{1+3}, \frac{3}{1+3} \right) = \left(\frac{1}{4}, \frac{3}{4} \right).$$

value を $v_1 = (4, 0)^\top$ 、 $v_2 = (0, 4)^\top$ とすると、

$$c = \frac{1}{4} \begin{pmatrix} 4 \\ 0 \end{pmatrix} + \frac{3}{4} \begin{pmatrix} 0 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}.$$



図：出力 c は、 v_1 から v_2 へ線分を $3/4$ 進んだ点。

一般には「凸結合」になる

$\alpha_i \geq 0$ かつ $\sum_i \alpha_i = 1$ なので、 c は value の凸結合です。2 点なら線分上、3 点ならそれらが作る三角形の内部または辺上にあります。各成分 j についても

$$\min_i (v_i)_j \leq c_j \leq \max_i (v_i)_j$$

が成り立ちます。

この性質の適用範囲

これは、本文の基本式による一つの head の重み付き和の性質です。その後の出力行列による変換や残差加算まで含めた出力が、元の value の凸包内にあるとは限りません。また、重みが最大の value を丸ごと選ぶ操作でもありません。

3 行列表記を、形と成分から読む

入力ベクトルは列として $x_t \in \mathbb{R}^{d_{\text{model}}}$ と書き、行列 X には $x_t^\top$ を行に並べます。本文の変換 $q_t = W_Q x_t$ なら、まとめた行列は $Q = XW_Q^\top$ です。行と列の流儀を混ぜないことが大切です。

行列	形	各行の意味
X	$n \times d_{\text{model}}$	入力表現
Q, K	$n \times d_k$	query、key
V	$n \times d_v$	value
$S = QK^\top / \sqrt{d_k}$	$n \times n$	各参照先への関連度
$A = \text{softmax}_{\text{row}}(S)$	$n \times n$	各参照先への重み
$C = AV$	$n \times d_v$	合成された表現

行列積の各行が、一つの重み付き平均

$$A = \begin{pmatrix} 1/4 & 3/4 \\ 1/2 & 1/2 \end{pmatrix}, \quad V = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} \implies C = AV = \begin{pmatrix} 1 & 3 \\ 2 & 2 \end{pmatrix}.$$

第 1 行は位置 1 の出力、第 2 行は位置 2 の出力です。 A は**各行の和が 1**ですが、各列の和が 1 とは限りません。ある位置が多く query から強く参照されることもあります。

multi-head は、複数の混ぜ方を並列に作る

head ごとに異なる変換を使い、出力 $C^{(h)} \in \mathbb{R}^{n \times d_v}$ を作ります。 H 個の出力を列方向に並べると

$$C_{\text{concat}} \in \mathbb{R}^{n \times H d_v}, \quad C_{\text{out}} = C_{\text{concat}} W_O$$

となります。 $W_O \in \mathbb{R}^{H d_v \times d_{\text{model}}}$ なら、出力は $n \times d_{\text{model}}$ です。各 head が必ず「文法」「意味」などの明確な役割に分かれるとは限りません。

attention 全体は線形か？

Q, K を固定すれば $V \mapsto AV$ は線形です。しかし self-attention では Q, K, V がいずれも入力 X から作られ、重み A 自体も X で変わります。そのため、 X から出力への写像は一般に線形ではありません。

4 なぜ $\sqrt{d_k}$ で割るのか

スケールを考えるため、 $q_1, \dots, q_{d_k}, k_1, \dots, k_{d_k}$ が互いに独立で、平均 0、分散 1 であると仮定します。これは計算を理解するための理想化です。

$$E[q_j k_j] = 0, \quad V[q_j k_j] = E[q_j^2] E[k_j^2] = 1.$$

各積も独立なので、

$$V[q^\top k] = V\left[\sum_{j=1}^{d_k} q_j k_j\right] = d_k.$$

したがって、内積の標準偏差は $\sqrt{d_k}$ です。これで割れば

$$V\left[\frac{q^\top k}{\sqrt{d_k}}\right] = 1$$

となり、次元の増加による典型的なスケールの増大を補正できます。

数値例：重みの偏りを比較する

$d_k = 4$ で、内積が $(4, 0)$ だったとします。

	softmax への入力	重み
補正なし	$(4, 0)$	$(0.982, 0.018)$
$\sqrt{4} = 2$ で割る	$(2, 0)$	$(0.881, 0.119)$

順位は変わりませんが、差のスケールが小さくなります。補正は重みを一様にする操作でも、極端な偏りを必ず防ぐ保証でもありません。実際の学習済みベクトルが上の独立性や分散の仮定を満たすとも限りません。

softmax が極端に偏ると何が起こるか

$$\frac{\partial \alpha_i}{\partial s_j} = \alpha_i (\mathbf{1}_{\{i=j\}} - \alpha_j).$$

一つの重みが 1 に近く、他が 0 に近いと、これらの微分は小さくなります。スコアを少し変えても重みが動きにくい領域です。スケール補正には、このような状態になりすぎることを抑える狙いがあります。

数値計算上の安定化とは別

$\sqrt{d_k}$ で割る操作は確率重みを変えます。一方、各行の最大値を引いてから softmax を計算する操作は、理論上の重みを変えずに指数計算のオーバーフローを避けます。

5 因果マスクと「次トークン」の位置

位置 r に w_r を入力し、その位置の出力から w_{r+1} を予測する流儀で考えます。このとき、位置 r は $w_1, \dots, w_r$ を参照して構いません。

出力の位置	参照可能な入力	予測対象
1	w_1	w_2
2	w_1, w_2	w_3
3	w_1, w_2, w_3	w_4

したがって、 w_t を予測するときは w_t 自身もまだ入力しないことに注意します。「自己位置を参照できる」と「予測対象の正解を参照できる」は別です。先頭トークンの予測には開始記号などを使います。

関連度がすべて 0 の場合の計算

$$S_{\text{masked}} = \begin{pmatrix} 0 & -\infty & -\infty \\ 0 & 0 & -\infty \\ 0 & 0 & 0 \end{pmatrix} \implies A = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}.$$

値を $v_1 = 2, v_2 = 8, v_3 = 5$ とすれば、出力は $(2, 5, 5)^\top$ です。未来位置を除いてから、残った位置の間で正規化しています。

0 で置き換えても、参照は禁止できない

softmax では $e^0 = 1$ なので、関連度を 0 にするだけでは重みは 0 になりません。 $-\infty$ は $\exp(-\infty) = 0$ という極限を表します。各行には少なくとも一つ、マスクされない位置が必要です。

本文の練習問題：解答

問 1 関連度は 1, 2。指示どおりスケール補正を省くと、

$$\alpha_1 = \frac{1}{1+e}, \quad \alpha_2 = \frac{e}{1+e}, \quad c = \left(\frac{10}{1+e}, \frac{6e}{1+e} \right) \approx (2.689, 4.386).$$

問 2 マスク後は $\begin{pmatrix} 1 & -\infty \\ 3 & 4 \end{pmatrix}$ 。第 1 行の重みは $(1, 0)$ 、第 2 行は $(1/(1+e), e/(1+e))$ です。第 1 位置の出力には、第 2 位置の value が入りません。

6 追加問題と解答

問 A：重みと内容を分けて考える

重みが $(1/4, 3/4)$ のまま、value を $v_1 = (8, 0)^\top$ 、 $v_2 = (0, 4)^\top$ に変えます。出力を求め、2 ページの例と比べてください。

問 B：行列の形

系列長 $n = 5$ 、 $d_k = 3$ 、 $d_v = 2$ とします。 $QK^\top$ 、attention 重み行列 A 、出力 AV の形を答えてください。

問 C：全スコアが同じ場合

参照可能な 3 位置の関連度が $(7, 7, 7)$ 、value が $(1, 0)^\top$ 、 $(0, 2)^\top$ 、 $(2, 1)^\top$ のとき、重みと出力を求めてください。

問 D：マスク後の正規化

関連度 $(\log 2, \log 3, \log 5)$ に対し、第 3 位置をマスクします。残りの重みを求めてください。マスク前の重みの第 3 成分を 0 にするだけでよいでしょうか。

問 E：線形性と解釈

「attention の出力は value の線形結合だから、self-attention 全体も入力 of 線形写像である」「重みが最大の位置だけが最終回答の理由である」。それぞれ、何に注意すべきでしょうか。

解答・考え方

A $c = (2, 3)^\top$ 。重みは同じでも、渡す内容が変われば出力も変わります。

B 順に 5×5 、 5×5 、 5×2 。

C 重みは $(1/3, 1/3, 1/3)$ 、出力は $(1, 1)^\top$ 。

D $(2/5, 3/5, 0)$ 。マスク前は $(2/10, 3/10, 5/10)$ なので、第 3 成分を 0 にするだけでは総和が 1 になりません。

E 重み自体が入力に依存するため、全体は一般に非線形です。また、出力は他の value や、その後の head の合成、各層の処理にも依存します。一つの重みだけから最終回答の理由は断定できません。

参考：Vaswani et al., *Attention Is All You Need* (2017), §§3.1–3.2. スケール補正・multi-head・因果マスクの基本構成を参照。

<https://arxiv.org/abs/1706.03762>