

# 知能プログラミング演習 I 第7回レポート

2025年7月27日 学籍番号 35714121 氏名 福富隆大

## 課題1

(a).

$$\mathbf{x} + \mathbf{y} : \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \end{bmatrix}, \quad \mathbf{x} - \mathbf{y} : \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \\ x_3 - y_3 \end{bmatrix}, \quad \mathbf{x} * \mathbf{y} : \begin{bmatrix} x_1 \cdot y_1 \\ x_2 \cdot y_2 \\ x_3 \cdot y_3 \end{bmatrix}, \quad \mathbf{x} / \mathbf{y} : \begin{bmatrix} x_1 / y_1 \\ x_2 / y_2 \\ x_3 / y_3 \end{bmatrix}$$

(b).

$$\mathbf{M} + \mathbf{x} : \begin{bmatrix} \mathbf{m}_1^\top + \mathbf{x}^\top \\ \mathbf{m}_2^\top + \mathbf{x}^\top \end{bmatrix}, \quad \mathbf{M} - \mathbf{x} : \begin{bmatrix} \mathbf{m}_1^\top - \mathbf{x}^\top \\ \mathbf{m}_2^\top - \mathbf{x}^\top \end{bmatrix}, \quad \mathbf{M} * \mathbf{x} : \begin{bmatrix} \mathbf{m}_1^\top \odot \mathbf{x}^\top \\ \mathbf{m}_2^\top \odot \mathbf{x}^\top \end{bmatrix}, \quad \mathbf{M} / \mathbf{x} : \begin{bmatrix} \mathbf{m}_1^\top \oslash \mathbf{x}^\top \\ \mathbf{m}_2^\top \oslash \mathbf{x}^\top \end{bmatrix}$$

(c).

$$\mathbf{N} + \mathbf{K} : [N_1 + K, N_2 + K] \in \mathbb{R}^{2 \times 4 \times 3}$$

(d).

$$\mathbf{N} @ \mathbf{M}.transpose(0,1) : [N_1 M^\top, N_2 M^\top] \in \mathbb{R}^{2 \times 4 \times 2}$$

$$\text{linear.weight} : \mathbf{W} \in \mathbb{R}^{2 \times 3}$$

$$\text{linear}(\mathbf{M}) : MW^\top \in \mathbb{R}^{2 \times 2}$$

$$\text{linear}(\mathbf{N}) : [N_1 W^\top, N_2 W^\top] \in \mathbb{R}^{2 \times 4 \times 2}$$

(e).

$$\mathbf{A} @ \mathbf{V} : [A_1 V_1, A_2 V_2] \in \mathbb{R}^{2 \times 4 \times 3}$$

(f).

$$\mathbf{V}.transpose(1,2)) : [V_1^\top, V_2^\top]$$

$$\mathbf{N} @ \mathbf{V}.transpose(1,2) : [N_1 V_1^\top, N_2 V_2^\top] \in \mathbb{R}^{2 \times 4 \times 5}$$

(g). F.softmax 関数の dim 引数について：

- dim=0 の場合：バッチ次元（最初の次元）に対して softmax が適用され、各位置で 2 つのバッチ間の確率分布が計算される。
- dim=1 の場合：長さ L 次元（2 番目の次元）に対して softmax が適用され、各系列の各位置間で確率分布が計算される。
- dim=2 の場合：特徴次元（最後の次元）に対して softmax が適用され、各位置の特徴間で確率分布が計算される。

## 課題 2

- (a). tensor 配列 input\_idx は 3 つの文字列 ("tiktok", "tictac", "tuktuk") をトークン ID の配列として表現している。配列のサイズは  $3 \times 6$  であり、1 次元目がバッチサイズ（文字列の数）、2 次元目が文字列の長さに対応している。
- (b). token\_embedding = nn.Embedding(char\_size, dim\_embed) は文字の ID を高次元ベクトルに変換する埋め込み層である。char\_size は文字の種類数 (26)、dim\_embed は埋め込み次元数 (2) を表す。 $x = \text{token\_embedding}(\text{input\_idx})$  で作成される  $x$  のサイズは  $3 \times 6 \times 2$  であり、1 次元目がバッチサイズ、2 次元目が文字列の長さ、3 次元目が埋め込み次元に対応している。
- (c). embed-1.pdf の図は、Token Embedding によって 3 つの文字列が 2 次元空間にプロットされた結果を示している。重複して同じ位置にある点があるのは、同じ文字は常に同じ埋め込みベクトルを持つためである。例えば、全ての 't' 文字は同じ座標にマップされ、'k' 文字も同様に重複している。
- (d). PositionalEncoding クラスの pe に以下のコードを実装した：

```
for pos in range(L):
    for i in range(0, dim_embed, 2):
        pe[pos, i] = torch.sin(torch.tensor(pos / (10000 ** (2 * i / dim_embed))))
        if i + 1 < dim_embed:
            pe[pos, i + 1] = torch.cos(torch.tensor(pos / (10000 ** (2 * i / dim_embed))))
```

結果として作成される embed-2.pdf の図では、Positional Encoding によって位置情報が加えられ、embed-1.pdf で重複していた点が分離されている。同じ文字でも位置によって異なる座標に配置されるようになった。

- (e). SelfAttention クラスの forward 関数に以下のコードを実装した：

```
K_t = K.transpose(-2, -1)
scaled_dot_product = Q @ K_t / torch.sqrt(torch.tensor(self.head_size, dtype=torch.float32))
atten_wei = F.softmax(scaled_dot_product, dim=-1)
out = atten_wei @ V
```

結果として作成される embed-3.pdf の図では、Self-Attention によって各位置の表現が他の位置との関係を考慮して更新され、embed-2.pdf からさらに点の配置が変化している。文脈を考慮した表現になったことで、重複の度合いがさらに変化した。

## 課題 3

- (a). loss\_fn(logits, y) は交差エントロピー損失を計算している。logits はモデルの出力（各文字の予測スコア）で、y は正解の文字 ID である。この損失関数により、モデルは次の文字を正確に予測するように学習される。

(b). Transformer クラスを以下のアーキテクチャで完成させた：

- Token 埋め込み → Positional encoding → Self-attention (single head) → 全結合層
- 全結合層：線形層 (output: 100 次元) → ReLU → 線形層 (output: char\_size 次元)

学習後の損失推移は loss.pdf に示されており、5000 ステップの学習で損失が約 4.5 から 0.2 まで減少した。

学習後に生成された文章の例：

- (a) 入力「ももからうまれた」→「ももからうまれたおとこのこを、おじいさんがはたけではたらいていますと、「やーい、よぼよぼじじい。よぼよぼじじい」と、」
- (b) 入力「ももからうまれた」→「ももからうまれたおとこのこを、おじいさんがはたけではたわへせかい」すずめをつか」おおばあさんはかにこのってかえってい」
- (c) 入力「ももからうまれた」→「ももからうまれたおとこのこを、おじいさんがはたけではたらいていますと、「やーい、よぼよぼじじい。よぼよぼじじい」と、」
- (d) (学習前の例) 「ももからうまれたおづぼでごはばがはそりっ! まびけ「つぼとえら3たらなも2『ねー「きわとゆぎご・て・え」
- (e) (学習後の特徴) 昔話らしい文体で、文法的に意味のある日本語が生成されるようになった