

先端機械学習 後半課題

提出日：2021 年 8 月 9 日

情報工学系

学籍番号：18B14822

宮崎 直哉

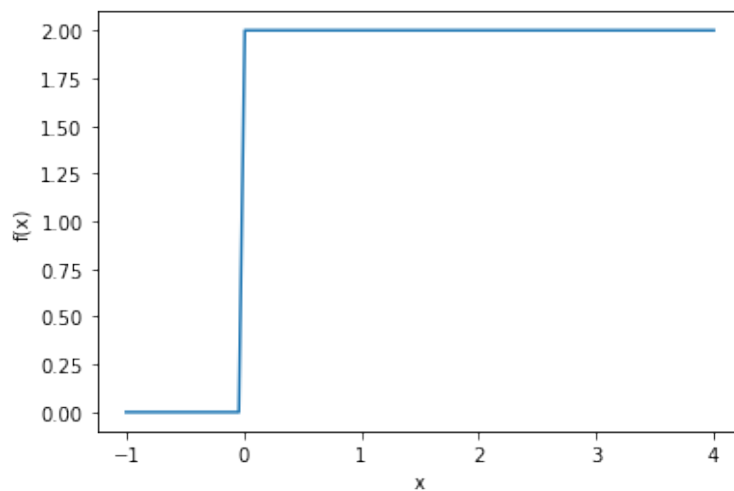
1 問題 1

パラメータ b の値 :

$$b = (0, -1000, -1000, -2000, -2000, -3000)$$

パラメータ b を求める方法 :

本問題では $w_i = 1000 (i = 1, 2, 3, 4, 5, 6)$ である。この時、シグモイド関数は下図のようになる。この時、このシグモイド関数はステップ関数と考えることができる。



この時、 $g_i = v_i \sigma(1000x + b_i)$ という式の意味を考えてみる。すると、シグモイド関数は x から $[0,1]$ 区間への写像であるので、 v_i はステップ関数の高さを表すことがわかる。また、

$$\begin{aligned} g_i &= v_i \sigma(1000x + b_i) \\ &= v_i \sigma\left(1000\left(x + \frac{b_i}{1000}\right)\right) \end{aligned}$$

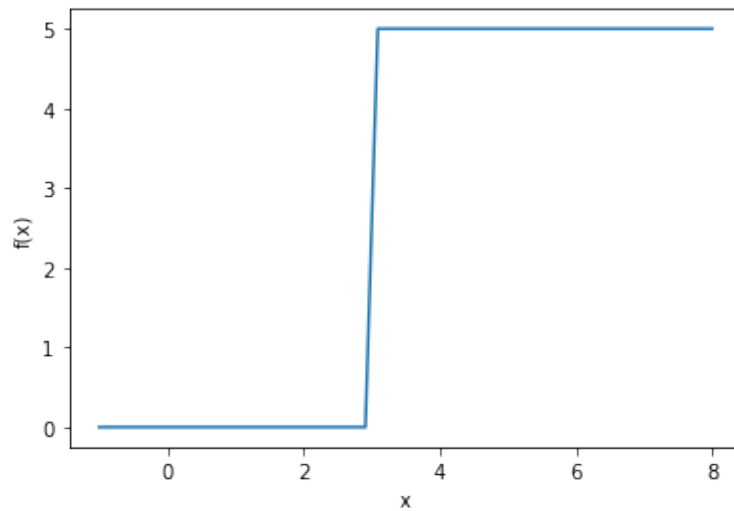
と変形すると、 $x = -\frac{b_i}{1000}$ がステップ関数の変換点となっていることがわかる。例えば、

$$f(x) = \begin{cases} 0 & (x < 3) \\ 5 & (3 \leq x) \end{cases}$$

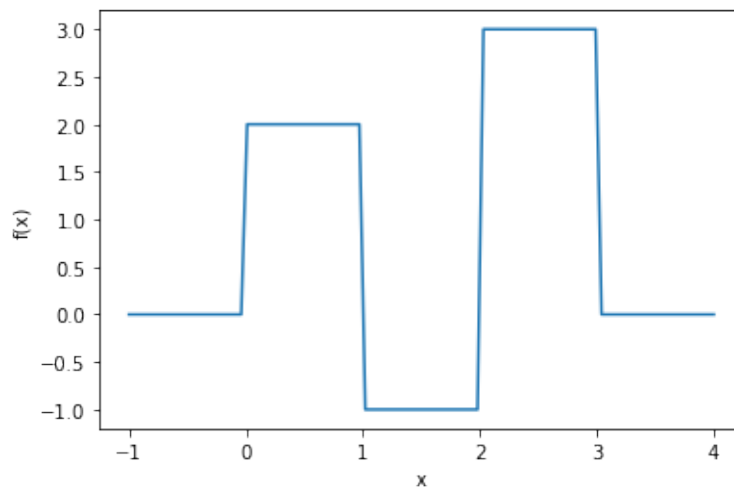
となるようなステップ関数をシグモイド関数で近似することを考えると、変換点が $x = -\frac{b_i}{1000} = 3$ で高さ $v_i = 5$ であるので、

$$g = 5\sigma(1000x - 3000)$$

とすれば、下図のようにステップ関数をシグモイド関数で近似したグラフを意図して得られる。



この性質を利用して本問題を考えてみる。 $f(x)$ は以下のグラフのようである。



式 $g(x)$ はシグモイド関数の和で表現されているので、上記の性質を合成した関数 g を作成することができる。ここで考えることは、ステップ関数の変換点と高さである。変換点として考えるべきなのは $x = 0, 1, 2, 3$ である。高さの変化については、 $x = 0 : +2$ 、 $x = 1 : -3$ 、 $x = 2 : +4$ 、 $x = 3 : -3$ となっている。高さについては、パラメータ v の値を見ていくと、 $x = 0$ において高さ $v_0 = 2$ 、 $x = 1$ において高さ $v_1 = -2$ 、 $v_2 = -1$ 、 $x = 2$ において高さ $v_2 = 1$ 、 $v_3 = 3$ 、 $x = 3$ において高さ $v_6 = -3$ のステップ関数をそれぞれ適用することで、 $\|f(x) - g(x)\| < \epsilon$ となるようなパラメータ b を決定することができる。

(本問題で作成したソースコードは<https://github.com/naoyaaan/AdvancedMachineLearning/blob/main/Okazaki/final.ipynb> にアップロードしています。)

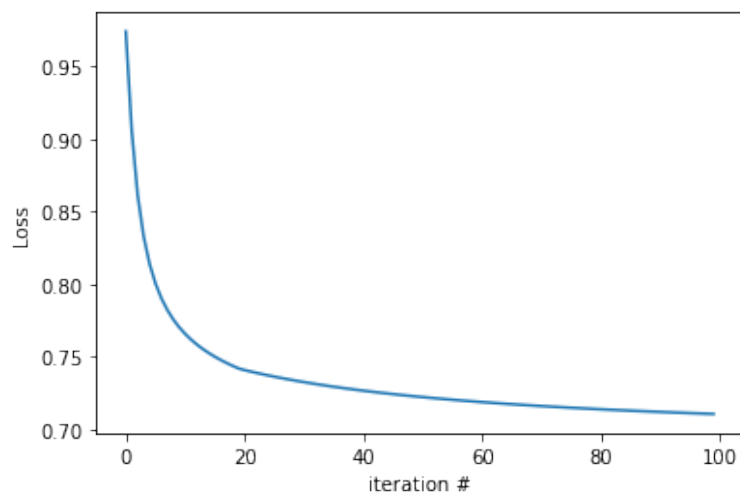
2 問題 2

(1)

$$h = \begin{pmatrix} 0 \\ 1.5 \end{pmatrix}$$

$$\hat{y} = 0.5$$

(2)



```
w:
Parameter containing:
tensor([[ 0.4912,  0.4912],
        [-1.0000, -1.0000]], requires_grad=True)
q:
Parameter containing:
tensor([[0.4744],
        [1.0000]], requires_grad=True)
b:
Parameter containing:
tensor([[ -1.0088],
        [ 1.5000]], requires_grad=True)
c:
Parameter containing:
tensor([[ -3.3401]], requires_grad=True)
Text(0, 0.5, 'Loss')
```

(本問題で作成したソースコードは<https://github.com/naoyaaan/AdvancedMachineLearning/blob/main/Okazaki/final.ipynb>にアップロードしています。)

3 問題3

3.1 WordSim-353

評価に用いられるタスクの概要

WordSim-353(WS353) は単語の類似性または関連性を測るためのデータセットである。Word Similarity Task(単語の類似性を測るタスク) では、モデルによる単語間の類似性の評価と人間による評価の相関を計測する。

データセットの統計情報

データセットの統計情報は以下の表 [2] のようである。

	pairs	type	raters	scale	Complex words	
					token	type
WS353	353	437	13-16	0-10	24	17
MC	30	39	38	0-4	0	0
RG	65	48	51	0-4	0	0
SCWS*	1762	1703	10	0-10	190	113
RW (new)	2034	2951	10	0-10	987	686

pairs は類似性を測る二つの単語を、raters は評価した人間の人数, scale は類似性のスコアを表している。

評価の尺度

上記のタスクによる評価には Spearman ' s rank correlation coefficient の値 ($\rho \times 100$) が用いられる。 ρ は $[-1,1]$ の値をとり、この値が大きいほど、人間の評価とモデルの評価に相関があることを表している。以下の表 [1] は実際にこのデータセットを用いてモデルを評価したときの結果である。

		sg	cbow	sisg-	sisg
AR	WS353	51	52	54	55
	GUR350	61	62	64	70
DE	GUR65	78	78	81	81
	ZG222	35	38	41	44
EN	RW	43	43	46	47
	WS353	72	73	71	71
ES	WS353	57	58	58	59
FR	RG65	70	69	75	75
Ro	WS353	48	52	51	54
RU	HJ	59	60	60	66

3.2 IMDB

評価に用いられるタスクの概要

IMDb はテキスト分類のタスクを評価するために用いられる。具体的には、Sentiment Analysis, Topic Detection, Language Detection などのタスクがある。Sentiment Analysis は、テキストがポジティブかネガティブがどちらでもないかを測るタスクである。Topic Detection は、自動的にそのテキスト、文章のテーマ、トピックを特定するタスクである。Language Detection は、与えられたテキストの言語を決定するタスクである。

データセットの統計情報

IMDb は the Internet Movie Database というサイトにおける、ポジティブ、ネガティブをラベル付けされた 50000 件のレビューからなるデータセットである。ポジティブとネガティブのレビューが同じ数だけ収録されている。10 段階の評価で 4 以下がネガティブ、7 以上がポジティブと判断している。

評価の尺度

IMDb を用いた Sentiment Analysis では、モデルのポジティブ or ネガティブの正答率 (%) を用いて評価している。以下の表 [3] は、現在の IMDb におけるモデルの予測精度のランキングである。

Model	Accuracy	Paper / Source
XLNet (Yang et al., 2019)	96.21	XLNet: Generalized Autoregressive Pretraining for Language Understanding
BERT_large+ITPT (Sun et al., 2019)	95.79	How to Fine-Tune BERT for Text Classification?
BERT_base+ITPT (Sun et al., 2019)	95.63	How to Fine-Tune BERT for Text Classification?
ULMFIT (Howard and Ruder, 2018)	95.4	Universal Language Model Fine-tuning for Text Classification
Block-sparse LSTM (Gray et al., 2017)	94.99	GPU Kernels for Block-Sparse Weights
oh-LSTM (Johnson and Zhang, 2016)	94.1	Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings
Virtual adversarial training (Miyato et al., 2016)	94.1	Adversarial Training Methods for Semi-Supervised Text Classification
BCN+Char+CoVe (McCann et al., 2017)	91.8	Learned in Translation: Contextualized Word Vectors

4 問題 4

系列変換タスクにおいて、Transformer が再帰型ニューラルネットワーク (RNN) よりも優れている点として以下の 2 つが挙げられる。

1. 計算時間が早い
2. 並列化が可能

まず Transformer では下図 [4] のような構造を用いている。

この中でも Transformer においては Self-Attention をベースとした構造となっている。RNN をベースとした構造と比較したのが下表 [4] である。

これによると、RNN のレイヤーでは、 $O(n^2 \cdot d)$ の計算量となっているが、Self-Attention では $O(n \cdot d^2)$ となっている。これだけでは RNN が Transformer より計算量が早いことはわからない。しかし、 n (sequence length) や d (representation dimension) のサイズに注目してみると、たいいていの場合 $n < d$ と n よりも d のほうがはるかに大きくなっている。そのため、Self-Attention のレイヤーのほうが計算量が少ないことがわかる。

また、RNN では入力に対してステップを経て処理をするので、Maximum path length の計算量は $O(n)$ である。それにたいして、Self-Attention は並列計算を行うので、Maximum path length の計算量は $O(1)$ となっている。GPU 演算を行うことを考えると Self-Attention は並列化できることというのが計算量の削減につながっており、RNN の構造よりも優れた点であるといえる。

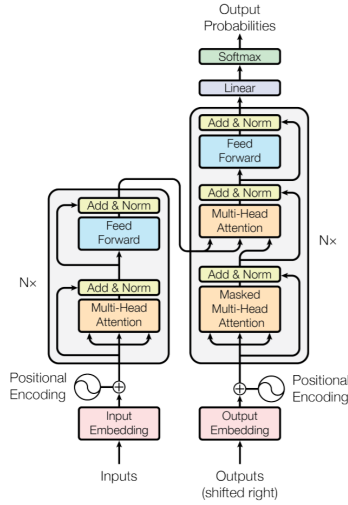


Figure 1: The Transformer - model architecture.

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

5 問題 5

この論文で検証したい仮説

近年の Transformer のような自然言語処理システムでは、事前に学習された言語表現を用いる傾向がある。これらのモデルは文章読解や質問応答など多くの自然言語処理のタスクにおいて、充実した成果を挙げている。しかし、これらのモデルはタスクにとらわれない構造をしているものの、タスクに特化したデータセットとファインチューニングが必要である。これには、言語モデルの応用を制限してしまったり、ファインチューンによってきわめて限定的なタスク分布となってしまう、人間であればほとんどの言語タスクを学習するためにたくさんの教師付きデータセットを必要としない、などの問題点がある。この論文では、データセットに特化した事前学習とファインチューニングを必要としない GPT-3 のモデルで最新の研究を超えるような性能を発揮することを検証する。

その仮説を検証するための実験方法

1750 億のパラメーターを持つ自己回帰モデルの GPT-3 を学習させ、文章学習能力を測ることで仮説を検証している。具体的には、20 を超える自然言語処理のデータセットと、訓練データに直接含まれていないようなタスクへの高速な適応度をテストするために設計されたいくつかの重要なタスクにおいて GPT-3 を評価している。各データセットにおいて、“few-shot learning”、“one-shot

learning”、”zero-shot learning”の3つの状況において評価する。few,one,zero はテストする際のデモンストレーションの数を表す。few-shot learning では10-100のデモンストレーションを用いる。

GPT-3 モデルのアーキテクチャ

GPT-3 では GPT-2[6] と同様のアーキテクチャを用いている。しかし GPT-3 では、transformer において、高密度でまとめられたスパースなアテンションが用いられている。

GPT-3 モデルの学習に用いられたデータ

下表 [5] に記載されているデータセットの集合を用いている。

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets used to train GPT-3. “Weight in training mix” refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

実験に用いられたタスクやデータセットの概要

以下に挙げるようなタスクやデータセットにおいて実験している。traditional language modeling tasks, ”closed book” question answering tasks, model’s ability to translate between language, Winogra Schema-like tasks, commonsense reasoning or question answering, reading comprehension tasks, SuperGLUE, NLI, additional tasks

実験結果の概要および GPT-3 で実現できたこと

GPT-3 では勾配の更新やアフィンチューニングなしに多くの自然言語処理のデータセットにおいて強力なパフォーマンスを達成した。zero-shot, one-shot learning では有望な結果が得られ、few-shot learning では大体のタスクにおいて最新研究と同等か、それ以上の結果を出している。

GPT-3 モデルでも実現できなかったこと

ANLI データセットのような自然言語推論タスクや、RACE や QuAC のような読解のデータセットにおいては few-shot learning は苦戦している。テキスト合成では意味的に同じことを繰り返すことがあるなどの問題があることがある。GPT-3 には双方向性がないので双方向性が必要なタスクが少々困難、学習において重要度を考慮しておらず、効率が悪いこと、必要な容量が過多で不便であること、GPT-3 の出力については解釈不能であることなどの問題点が残っている。

参考文献

- [1] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. Enriching Word Vectors with Subword Information. arXiv:1607.04606. 2017
- [2] Minh-Thang, Luong Richard Socher, Christopher D. Manning. Better Word Representations with Recursive Neural Networks for Morphology.
- [3] NLP-progress. http://nlpprogress.com/english/sentiment_analysis.html. 閱覽日 : 2021 年 8 月 3 日
- [4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762. 2017
- [5] T. B. Brown, et al. Language Models are Few-Shot Learners. arXiv:2005.14165. 2020.
- [6] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.