

ヘイトスピーチ検出

問題設定と問題分析

課題D-2:ヘイトスピーチ検出

- ・twitterの投稿文章(英文)を0:ヘイトスピーチ(hate_speech), 1:暴言(offensive_language), 2:どちらでもない(neither)に分類するモデルの作成。
- ・「hatespeech-train.csv」, 「hatespeech-test.csv」のデータを使用外部データの追加使用も可能

ヘイトスピーチ (英: hate speech、憎悪表現) とは、

ツイッターポリシーの定義：

人種、民族、出身国、性的指向、性別、性同一性、宗教、年齢、障害、深刻な病気などに基づいて、他の人々に対する直接の暴言、あるいは攻撃のこと

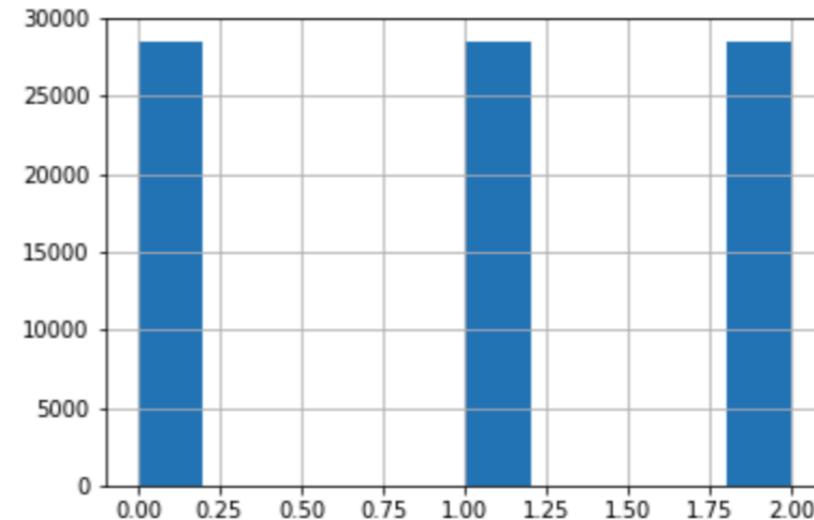
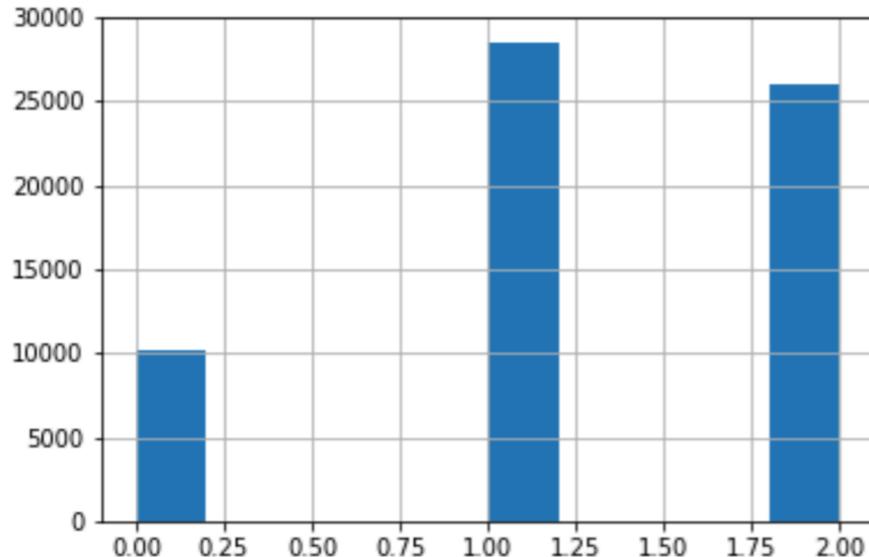
問題分析：

- ・計算のために、単語を数値化
- ・ヘイトスピーチと暴言の分類が一番難しい (この検出効果一番知りたい)

前処理

- ・以下のサイトから外部データ取得
Automated Hate Speech Detection - Data¹
 - ・上記のデータと訓練データと合併する
 - ・正規表現でテキストにいらない文字を削除 ("http","RT"など)
 - ・stop words削除 (you,me,myself..)
- ・トレーニングデータからテストデータを分離 (これをしないと、ヘイトスピーチと暴言の分類は100%に近い)
 - ・カテゴリー1に等しくなるようにクラスのサンプル数を複製する。²

参考：The Impact of Imbalanced Training Data for Convolutional Neural Networks



1

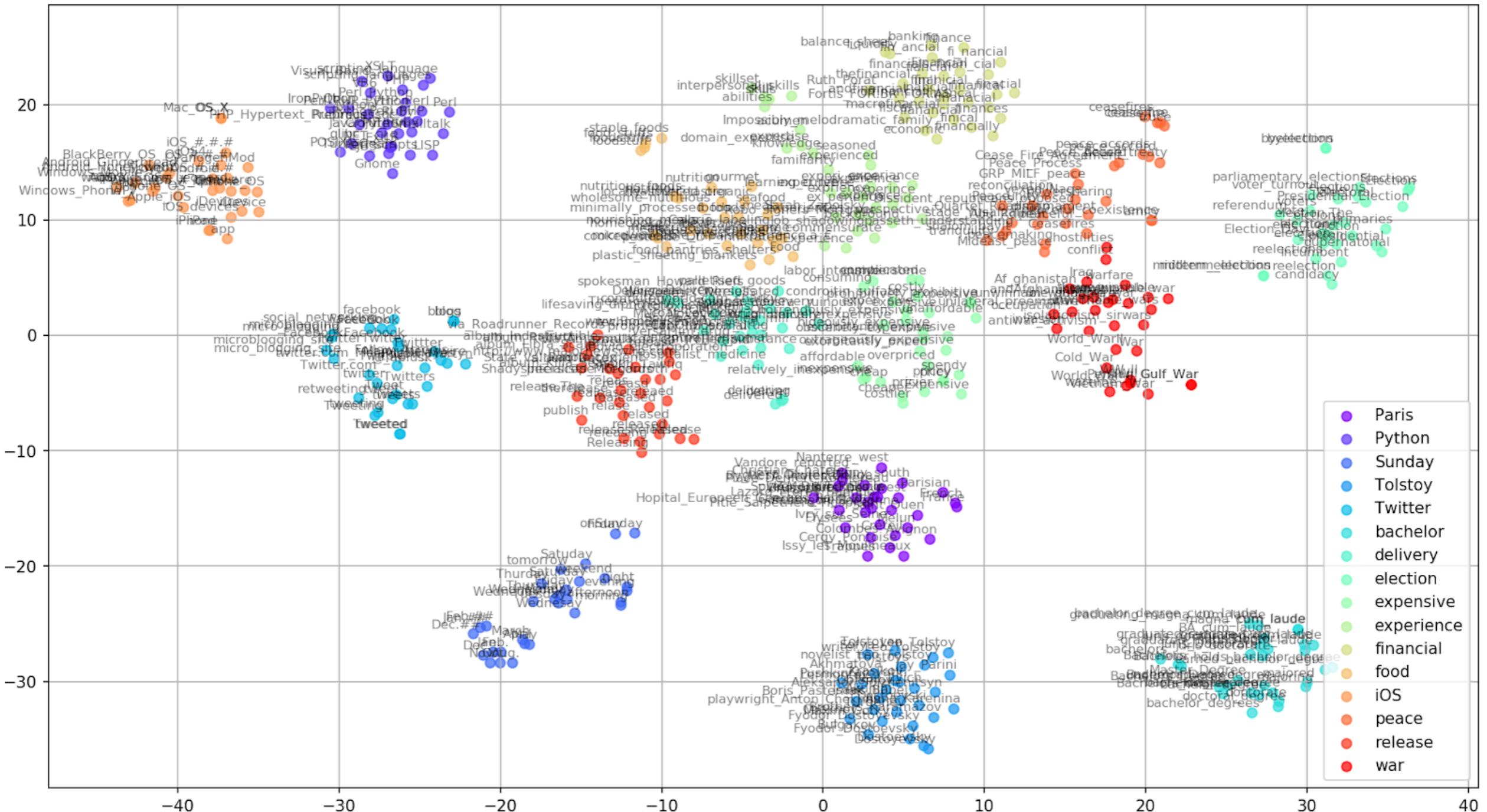
<https://data.world/ml-research/automated-hate-speech-detection-data>

2

Paulina Hensman, David Masko "The Impact of Imbalanced Training Data for Convolutional Neural Networks" (2015)

CNN&LSTMのための前処理

GoogleNews vectors Embedding



¹ Ziqi Zhang、Lei Luo "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter" arXiv:1803.03662(2018)

評価

Confusion matrix

		真実	
		Positive	Negative
予測	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

- 精度 (Precision)
- 再現率 (Recall)
- F値 (f-measure)
- 正解率 (Accuracy)

$$\begin{aligned} &= \frac{TP}{TP+FP} \\ &= \frac{TP}{TP+FN} \\ &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\ &= \frac{TP+TN}{TP+FP+FN+TN} \end{aligned}$$

深層学習的なアプローチ

畳み込みニューラルネットワーク(CNN)

- ・全結合層だけでなく畳み込み層(Convolution Layer)とプーリング層(Pooling Layer)から構成されるニューラルネットワーク。
- ・最初は画像認識の分野で優れた性能を発揮しているネットワークだが、最近ではテキスト分類でもよく使われる

LSTM + Attention layer

- ・時系列で可変長のデータを処理するために再帰構造を持ったモデルだ。
- ・時系列解析のほか、Encoding-decodingモデル (ex.グーグル翻訳) 、テキスト分類なども使われている。
- ・長期学習による勾配がゼロになるRNNの問題をなくす。

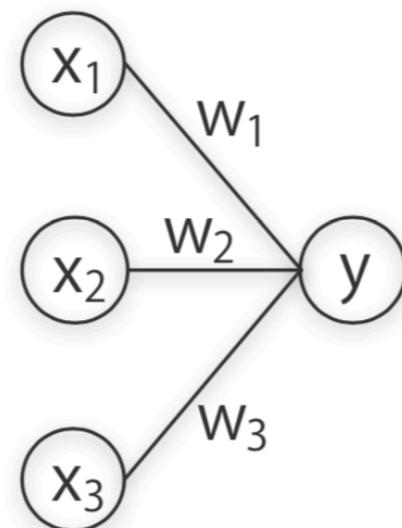
深層学習的なアプローチの流れ：

CNNに至るまでのニューラルネットワークの簡単紹介

→CNN→LSTM

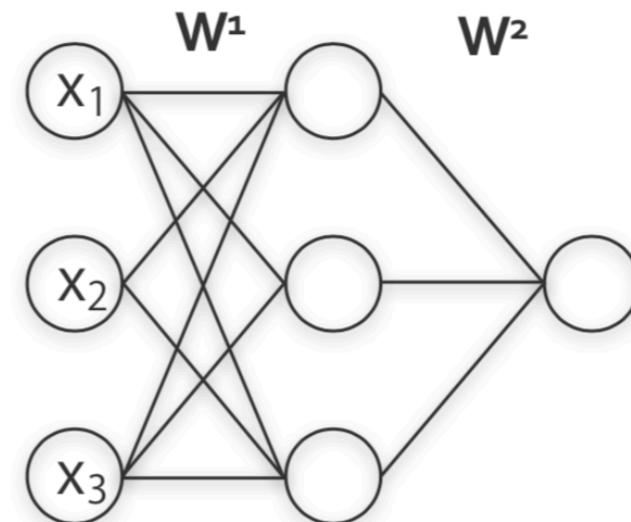
MLPの特徴 1：隠れ層

ロジスティック回帰



$$y = \mathbf{W}\mathbf{x} + b$$

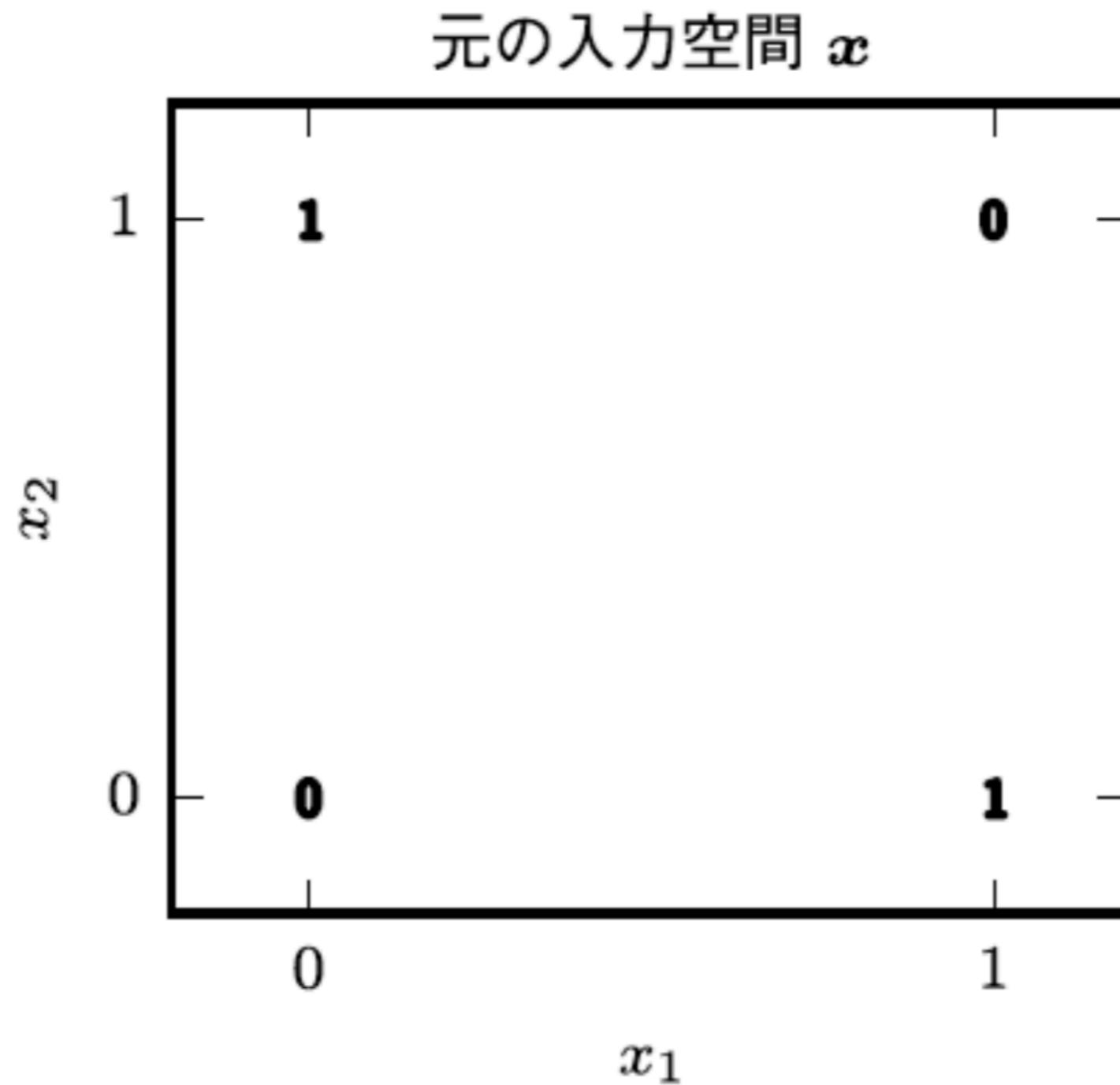
ニューラルネットワーク



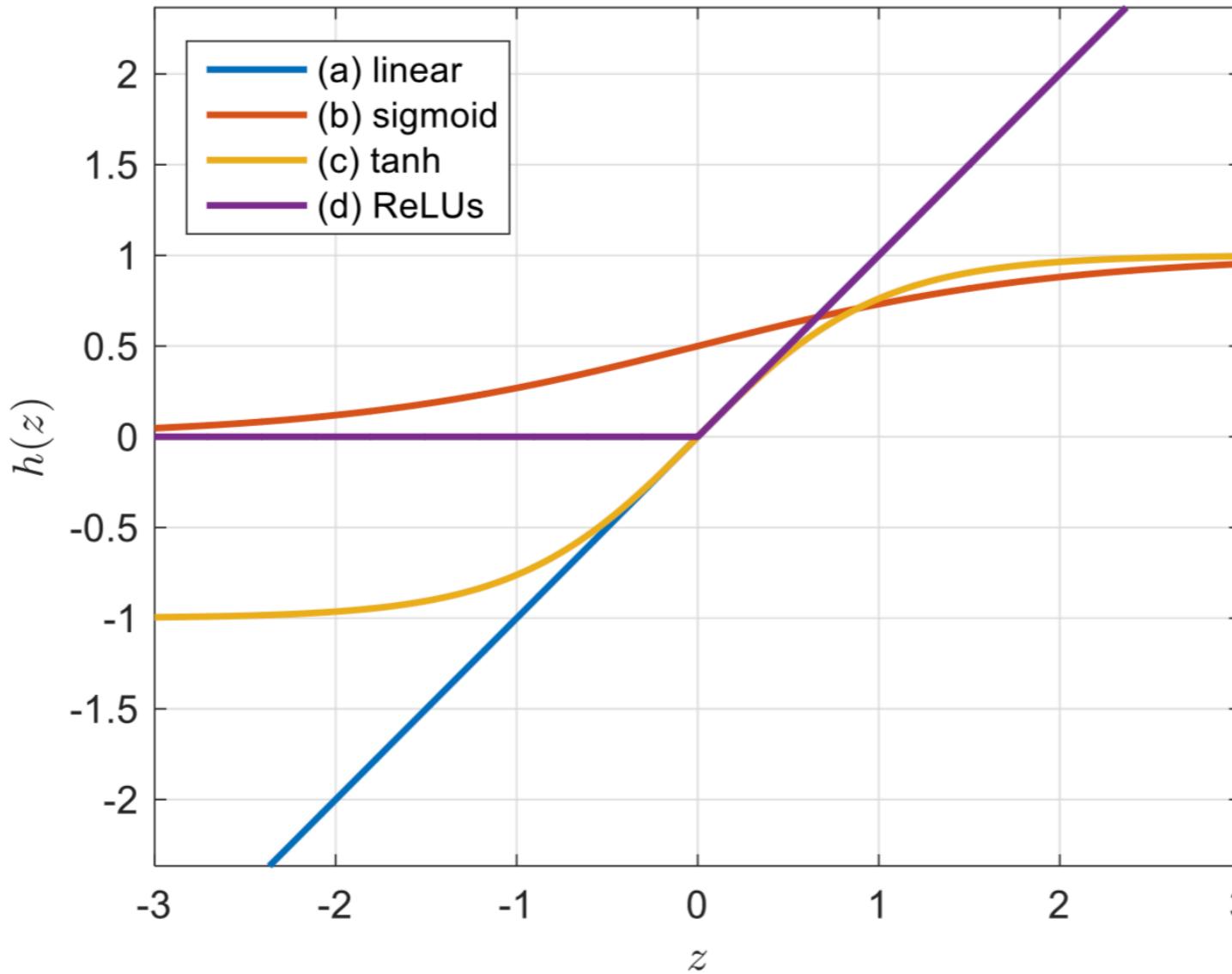
$$y = \mathbf{W}^2 \sigma(\mathbf{W}^1 \mathbf{x} + \mathbf{b}^1) + \mathbf{b}^2$$

- ・線形回帰：1回の変換
- ・MLP：複数回変換
 - 図の例だと、1度変換をかけてから線形回帰しているのと同じ
- ・間の層を、**隠れ層（あるいは中間層）**と呼ぶ

線形でない簡単な例:XORデータ

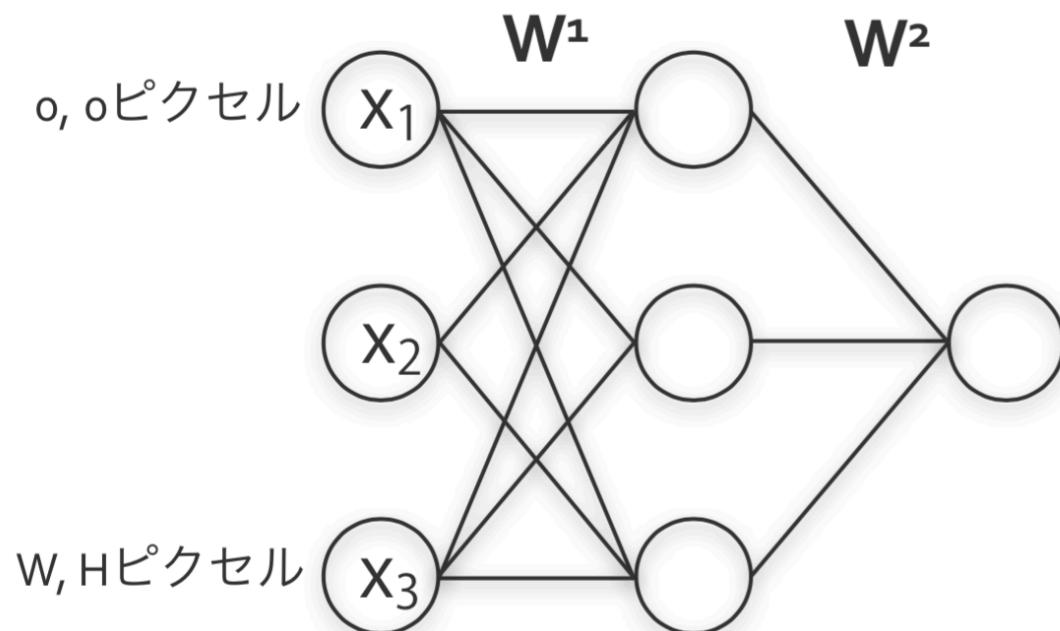


MLPの特徴 2 : 活性化関数



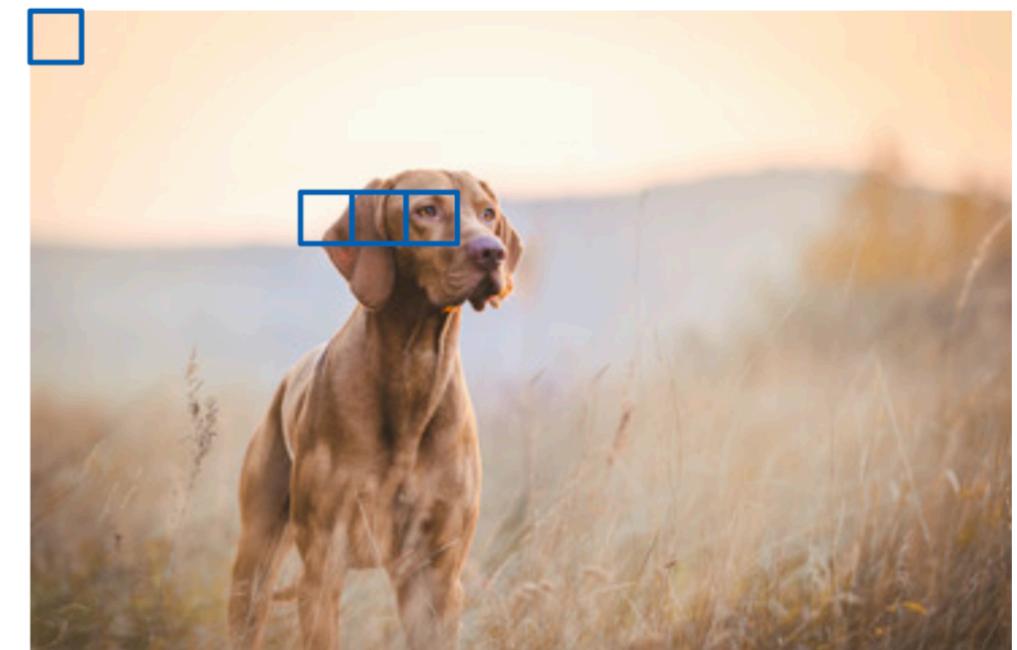
- Sigmoid
 - $\text{sigmoid}(h) : \frac{1}{1+\exp(-h)}$
- Tanh
 - $\tanh(h) = \frac{e^h - e^{-h}}{e^h + e^{-h}}$
- ReLUs
 - $\text{relu}(h) = \max(h, 0)$

NLPモデルの問題



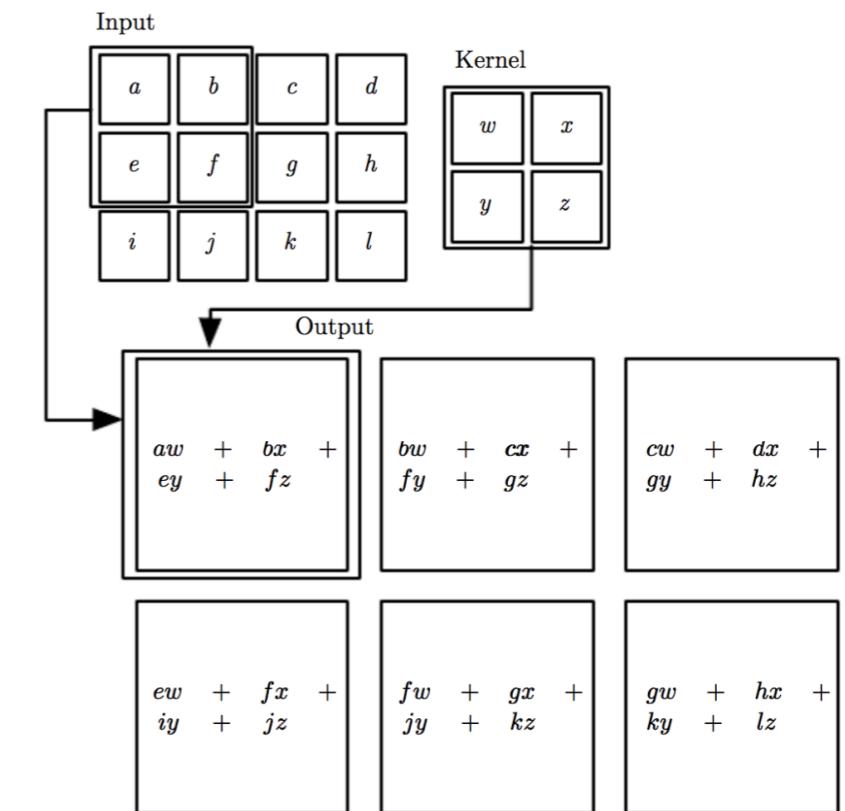
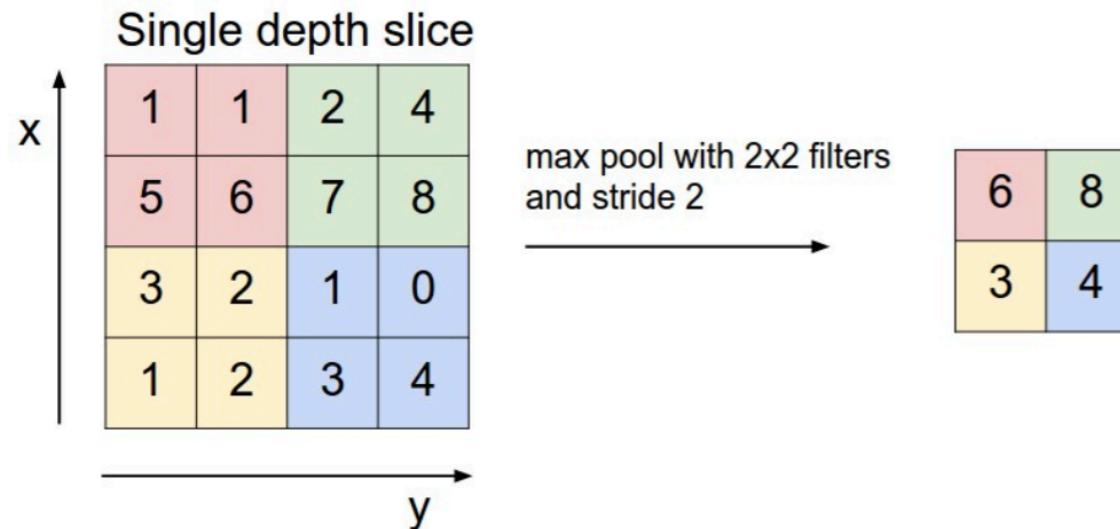
$$y = W^2 \sigma(W^1 x + b^1) + b^2$$

- すべての画素値が対等な関係
- 通常は、画像としてつながっているものは何らかの塊として捉えたい

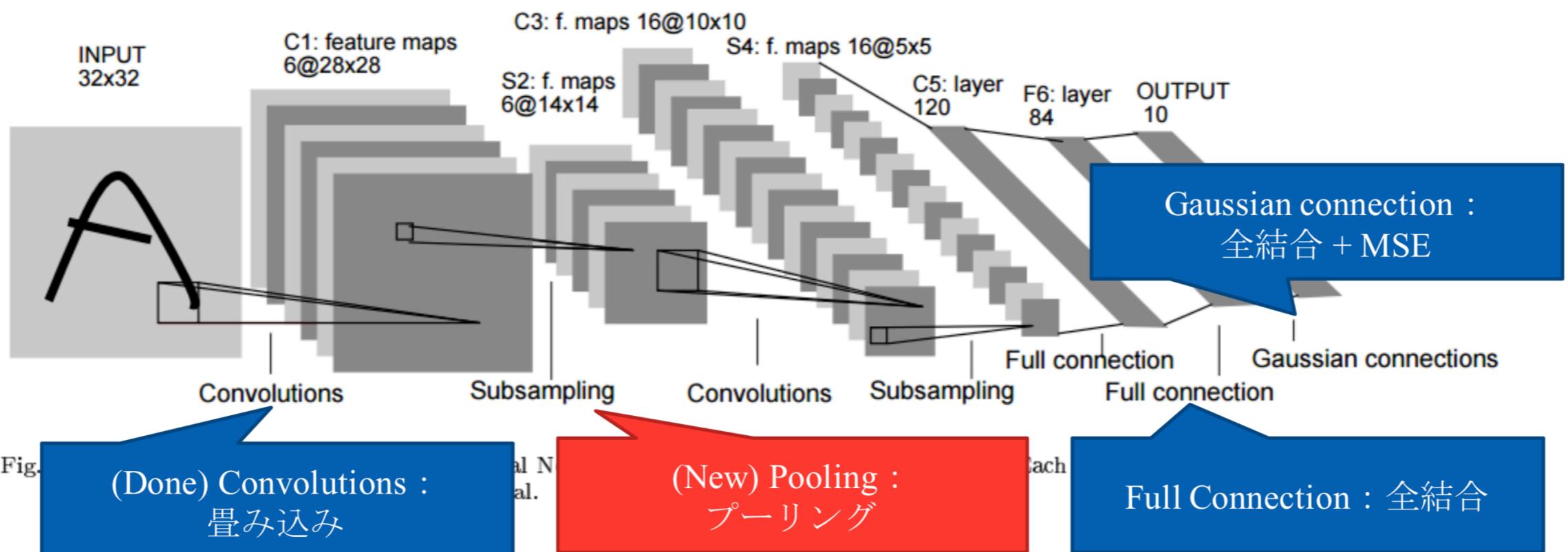


CNNモデル プーリング層

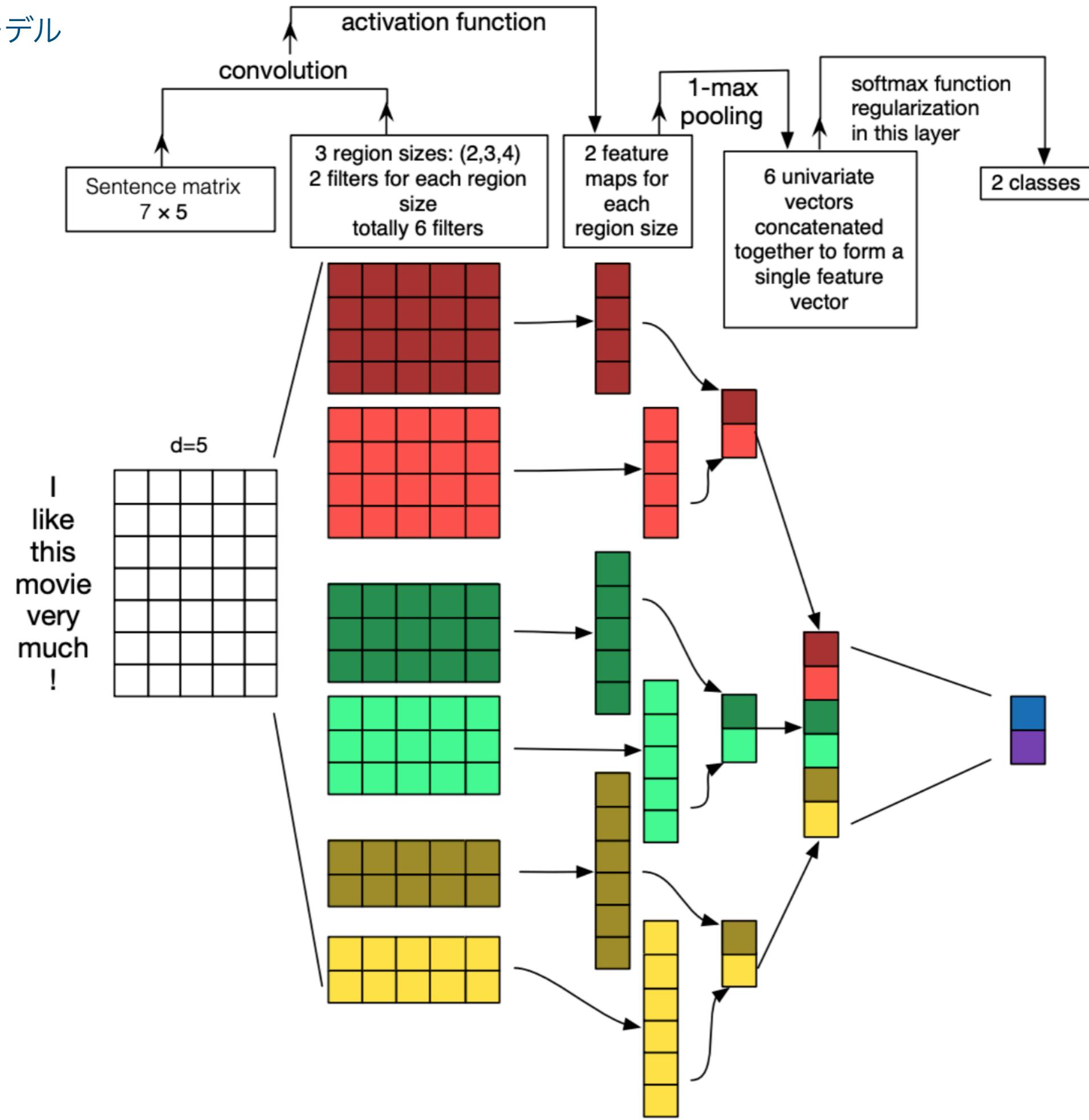
畳み込み層



CNNの全体像



CNNモデル



CNNモデル

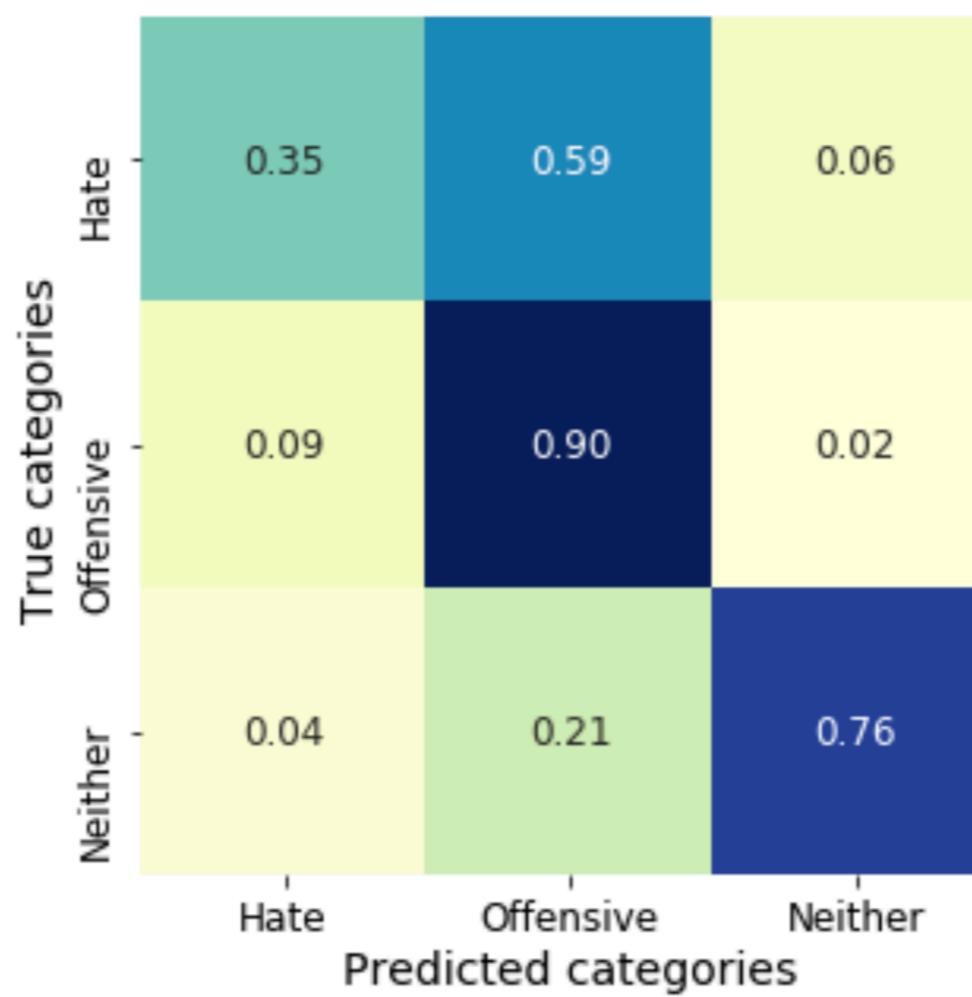
パラメーターの設定

元データ	トレーニングデータ (90%)	+ 検証データ (10%)	+ テストデータ
Embedding		GoogleNews-vectors-	21206
畠み込み層		128、128、128	
畠み込み層のフィルター		3、4、5	
プーリング層		3、3、3	
出力層		3	
出力活性化関数		ソフトマックス関数	
誤差関数		交差エントロピー誤差関数	
学習回数		10	

CNNモデル

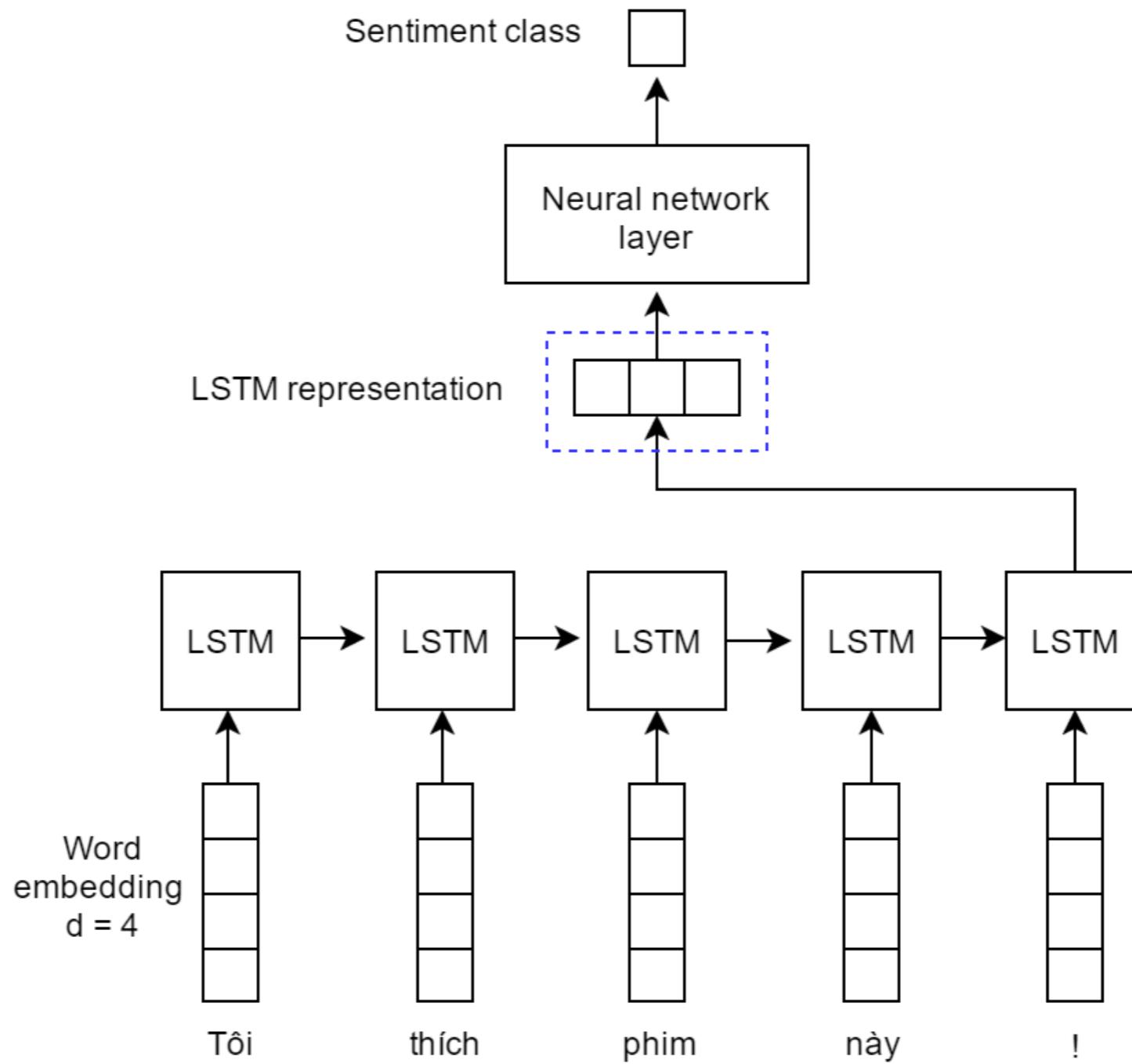
CNN

	precision	recall	f1-score	support
0	0.46	0.35	0.40	140
1	0.73	0.90	0.81	500
2	0.95	0.76	0.84	400
micro avg	0.77	0.77	0.77	1040
macro avg	0.71	0.67	0.68	1040
weighted avg	0.78	0.77	0.76	1040



LSTMモデル+attention layer

LSTM



Attention LSTM

$$p(\mathbf{y}(1), \dots, \mathbf{y}(T') \mid \mathbf{x}(1), \dots, \mathbf{x}(T)) = \prod_{t=1}^{T'} p(\mathbf{y}(t) \mid \mathbf{y}(1), \dots, \mathbf{y}(t-1), \mathbf{c})$$

$$\mathbf{h}_{dec}(t) = f(\mathbf{h}_{dec}(t-1), \mathbf{y}(t-1), \mathbf{c})$$



$$\mathbf{h}_{dec}(t) = f(\mathbf{h}_{dec}(t-1), \mathbf{y}(t-1), \underline{\mathbf{c}(t)})$$

時間的な重みという特徴量を追加

時間を考慮

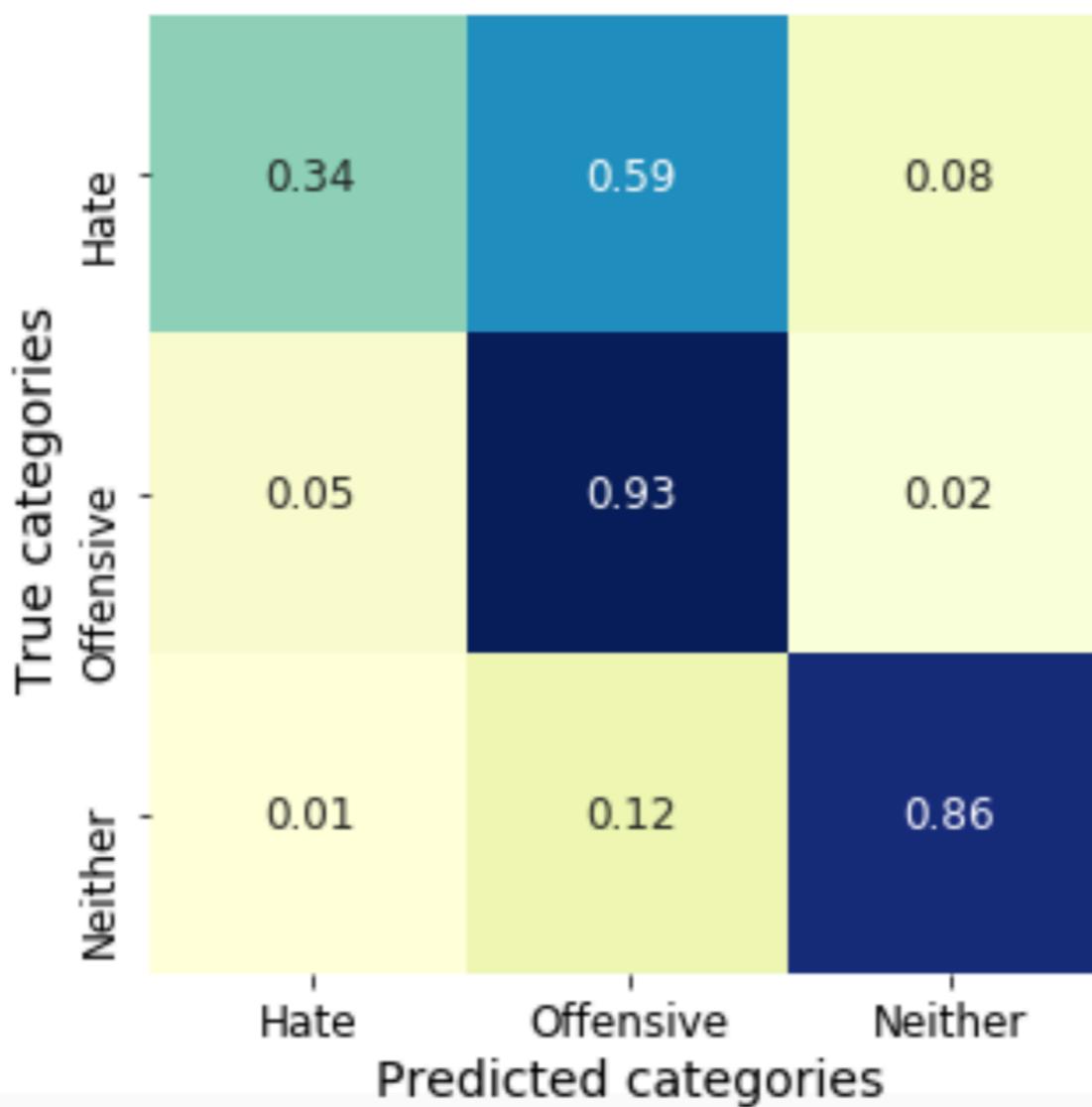
LSTMモデル+attention layer

パラメーターの設定

元データ	トレーニングデータ (90%)	+ 検証データ (10%)	+ テストデータ
Embedding		GoogleNews-vectors- 21206	
入力層		83	
LSTMブロック		300	
attention		Attention layer	
出力層		3	
出力活性化関数		ソフトマックス関数	
誤差関数		交差エントロピー誤 差関数	
学習回数		10	

LSTMモデル+attention layer

	precision	recall	f1-score	support	
0	0.61	0.34	0.43	140	
1	0.78	0.93	0.85	500	
2	0.94	0.86	0.90	400	
micro avg	0.82	0.82	0.82	1040	
macro avg	0.78	0.71	0.73	1040	
weighted avg	0.82	0.82	0.81	1040	



考えられる原因

モデルによる要因

- ・CNNでは、キーワードだけ抽出して、語順で捉えられない。
- ・LSTMでは、語順捉えられても、単語を塊として捉えられない。

特徴量の要因

- ・侮辱など汚い言葉は使われていないヘイトスピーチ
ex.“Assimilate? No they all need to go back to their own countries.
#BanMuslims Sorry if someone disagrees too bad.’
- ・ヘイトスピーチには暴言の言語は混ざっていることが多い、識別のが難しい
ex.'All you perverts (other than me) who posted today, needs to leave the O
Board. Dfasdfdasfadfs'
- ・単語の**希少性**が特徴として抽出できない。
- ・特定の個人あるいは団体に対する攻撃のため、名詞と、形容詞が必要になる。その特徴量は捉えていない。（**品詞のタグ**がない）
- ・単語に対する感情を捉える特徴量がない。ex,Positive,Neutral,Negative

外的要因：

- ・人によるヘイトと暴言の線引きの**主観性**
- ・深層学習の限界（補足として一番最後に説明）

機械学習的なアプローチ

これから使う機械学習のモデル：

- ロジスティック回帰

- LightGBM

深層学習分析との違い

- 例えば、ロジスティック回帰では、線形分類しかできないためGoogleNews vectorsを使って複雑なembedding必要がない。
- その代わりに全単語を、希少性、品詞のタグ、文章の単語数、リーダビリティ(読みやすさ)、音節の数、単語ごとの平均音節数、単語ごとのPositive、Negative、Neutral、CompoundやN-gramなど色々な特徴量が作れる。

参考

リーダビリティの指標

$$206.835 - (1.015 \times \alpha) - (84.6 \times \beta)$$

where,

α = 文章ごとの平均単語数

β = 単語ごとの音節数

$$(0.39 \times \alpha) + (11.8 \times \beta) - 15.59$$

Flesch-Kincaid Grade Level

where,

α = 文章ごとの単語数

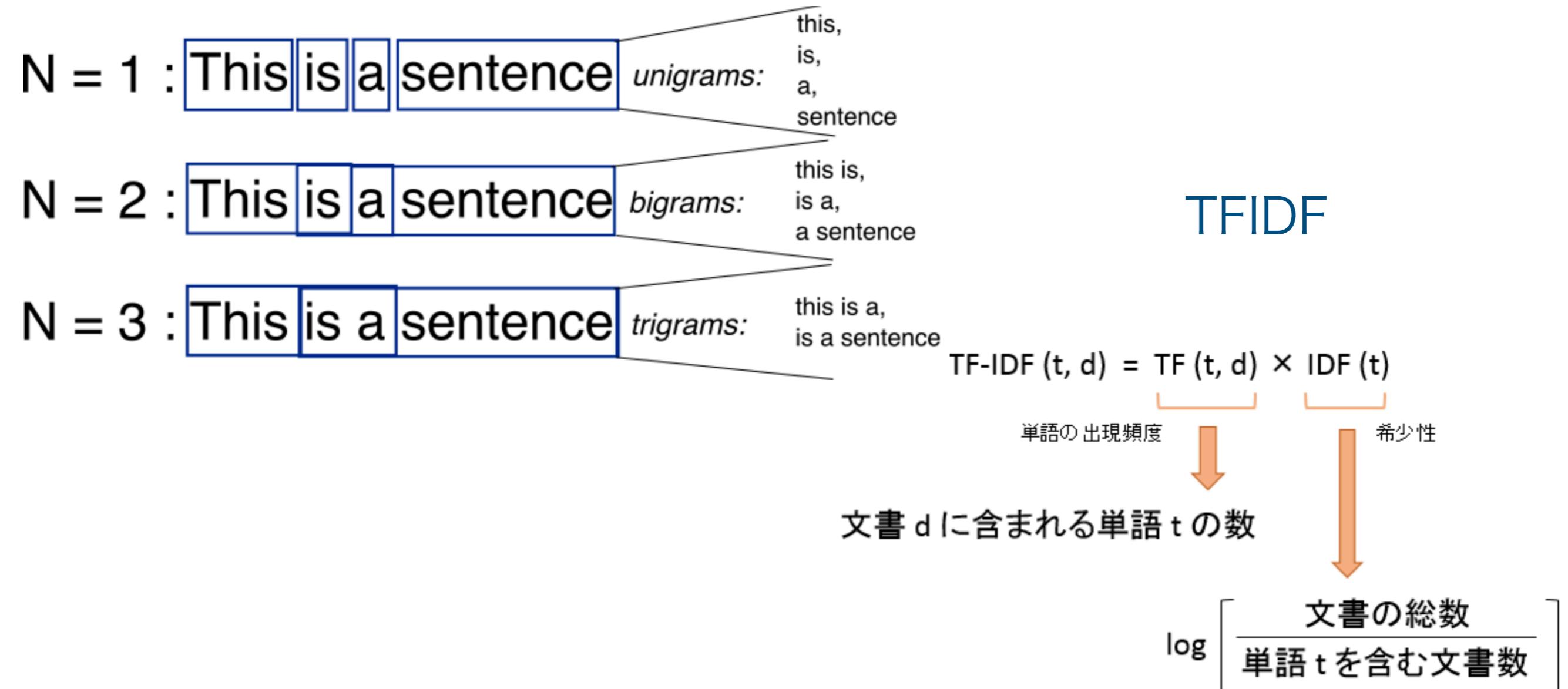
β = 単語ごとの音節数

ロジスティック回帰のための前処理 1

ロジスティック回帰

N-gram and TFIDF

- n-gram range 1-3
- 単語の順番と固まりを分割



t が、すべての文書に含まれる場合は 0 だが、少数の文書にしか含まれない場合は値が大きくなる。

ロジスティック回帰

POS	意味	単語例
ADJ	形容詞	<i>big, old, green, incomprehensible, first, second, third</i>
ADP	設置詞	<i>in, to, during</i>
ADV	副詞	<i>very, well, exactly</i>
AUX	助動詞	<i>has(done), is(doing), will(do), was(done), got(done), should(do), must(do)</i>
CCON_J	接続詞	<i>and, or, but</i>
DET	限定詞	<i>the, a, an, this, that, my, your, a few, a little, one, ten, all, both, another, such, what</i>
INTJ	間投詞	<i>psst, ouch, bravo, hello, well, you know, excuse me</i>
NOUN	名詞	<i>girl, cat, tree, air, beauty</i>
NUM	数詞	<i>0, 1000, 3.14, one, two, seventy-seven, I, II, III, IV, V, MMXIV</i>
PART	助詞	<i>'s, not</i>
PRON	代名詞	<i>I, you, he, it, they, myself, yourself, who, what, somebody, anything, everybody, nothing</i>
PROP_N	固有名詞	<i>Mary, John, London, NATO, HBO</i>
PUNC_T	句読点	<i>.(ピリオド), ,(カンマ), ()(括弧)</i>
SCON_J	連結詞	<i>that, if, while</i>
SYM	シンボル	<i>\$, %, §, ©, +, -, ×, ÷, =, <, >, :-)(顔文字, 😊 (絵文字), kei.0324@example.com, http://example.com/</i>
VERB	動詞	<i>run, eat, runs, ate, runnning, eating</i>
X	その他	<i>上記品詞に当てはまらない単語</i>

Universal POS tags

vader sentiment analysis

Article	Noun and frequency	Negative Score	Neutral Score	Positive Score	Compound Score	# Negative Sentiment Words	# Neutral Sentiment Words	# Positive Sentiment Words
1	('Everton', 9) ('United', 8) ('Martyn', 8)	0.102	0.676	0.221	0.995	19	292	32
2	('Van', 5) ('Nistelrooy', 5) ('United', 4)	0.083	0.705	0.212	0.9726	6	132	14
3	('Moyes', 5) ('Beattie', 5) ('Gallas', 5)	0.084	0.805	0.111	0.4504	8	204	13

パラメーターの設定

以下の論文に従い、以下の設定を行った¹

元データ

トレーニングデータ
(90%)+ 検証データ
(10%)

+ テストデータ

Grid search

最適パラメート

class weight

balanced

正則化

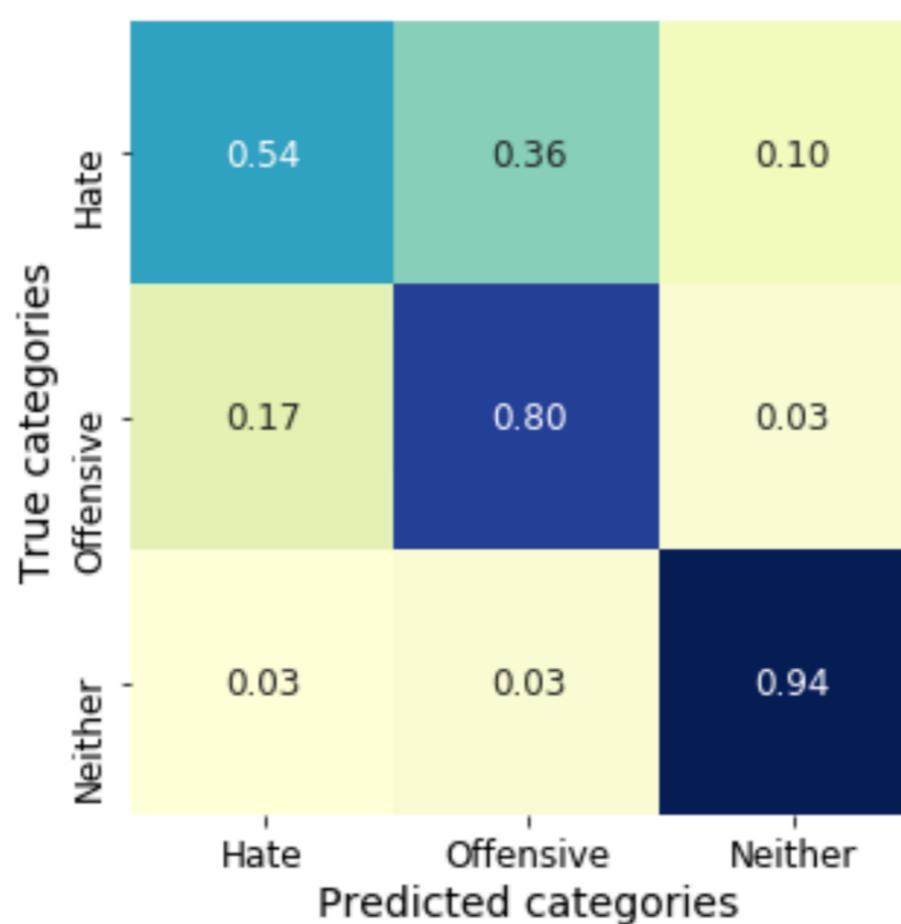
L2

¹

Thomas Davidson, Dana Warmsley , Michael Macy, Ingmar Weber "Automated Hate Speech Detection and the Problem of Offensive Language"(2017)

ロジスティック回帰

	precision	recall	f1-score	support
0	0.44	0.54	0.49	140
1	0.86	0.80	0.83	500
2	0.92	0.94	0.93	400
micro avg	0.82	0.82	0.82	1040
macro avg	0.74	0.76	0.75	1040
weighted avg	0.83	0.82	0.82	1040



LightGBM

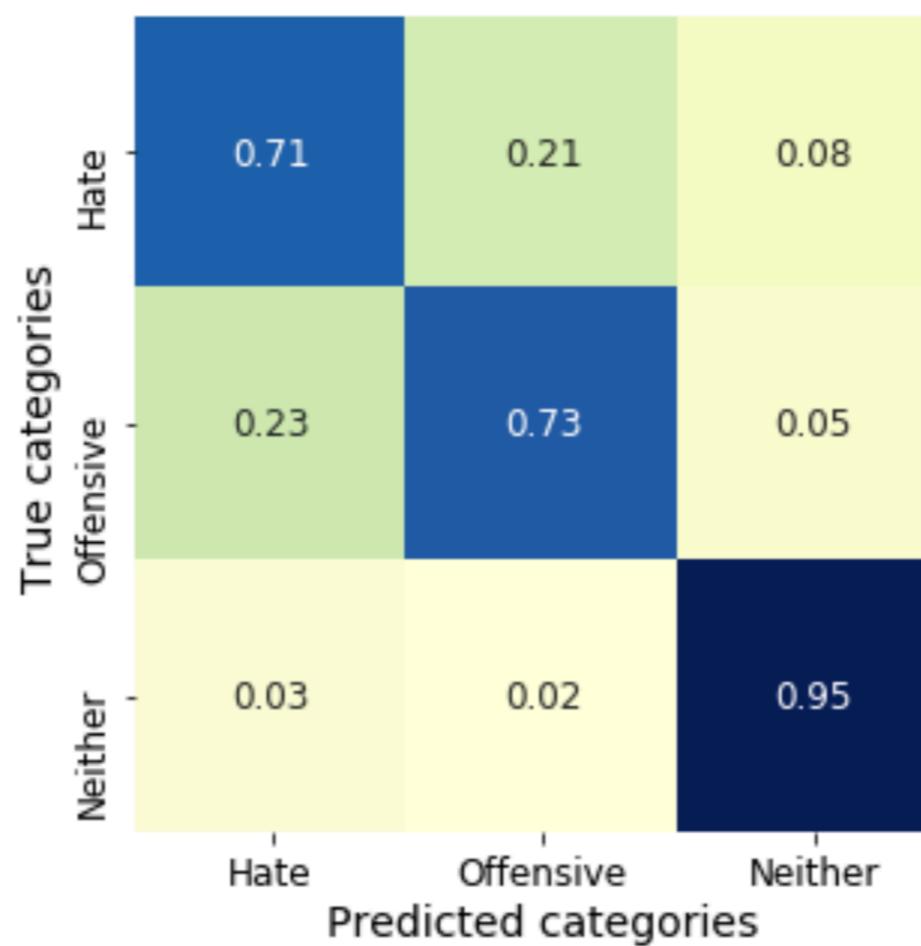
LightGBMの再訪

- ・先ほどサンプル数が少ないデータでは、LightGBMの方が良いという結論でHate分類に使う
パラメーターの設定

元データ	トレーニングデータ (90%)	+ 検証データ (10%)	+ テストデータ
各木の特徴量利用の割合		1.0(default)	
バギング		1.0(default)	
学習率		0.016	
葉の数		33	
学習回数		200	

LightGBM

	precision	recall	f1-score	support
0	0.44	0.71	0.54	140
1	0.91	0.73	0.81	500
2	0.92	0.95	0.93	400
micro avg	0.81	0.81	0.81	1040
macro avg	0.76	0.80	0.76	1040
weighted avg	0.85	0.81	0.82	1040



まとめと反省点

まとめ

- ・ CNN、LSTM、ロジスティック回帰とLightGBMを用いて予測してみた。
- ・ ヘイトスピーチを暴言から分類の場合は、LightGBMの方が良い。
- ・ 前処理の手間を省けたい、（ヘイトスピーチ、暴言）どちらでもない(neither)のみ二値分類したい場合は、LSTMかCNNの方が良い。

反省

- ・ LSTM-CNN、CNN-LSTMあるいは、Bidirectional LSTMは精度が上がるが、Hateの分類精度がそんなに変わらないので、試していない。
- ・ LightGBMは、Grid searchを使えばもっといい結果が出るかもしれない。時間的な原因で使わなかった。

補足：深層学習の限界

- ・人間の知能とは、世界の認識能力であり、人間は、**言語蓄積**と**社会経験**を通して、世界の構造を認識しているので**モデル学習では得られない**。
- ・暴言かヘイトという**境界線**は、世界の認識能力に基づいて、人間の纖細な感情機能が働いているため、世界の認識能力（つまり、社会的経験によって外界の刺激を元に反応する能力）がまだ欠けている今の人工知能では、機械学習にしても、深層学習にしても、人間並に、言葉の意味や、感情に捉えることは難しいと思う。
- ・今回の課題を通して今の人工知能の**限界**が見えてくると思う。