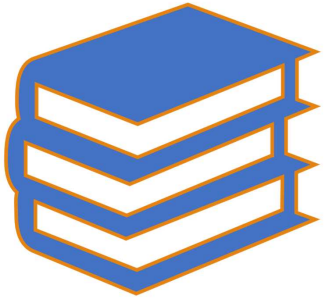
需要予測におけるニューラルネットワークの活性化関数の選定に関する研究

東京海洋大学大学院
海洋科学技術研究科　海運ロジスティクス専攻
1955018　李子揚
指導教員　黒川久幸　教授

国立大学法人 東京海洋大学 Tokyo University of Marine Science and Technology

目次



1・研究背景

研究背景

コロナショック

世界貿易が急速に縮小している

		2019	楽観的シナリオ		悲観的シナリオ	
			2020	2021	2020	2021
世界		-0.1	-12.9	21.3	-31.9	24.0
輸出	北米	1.0	-17.1	23.7	-40.9	19.3
	中南米	-2.2	-12.9	18.6	-31.3	14.3
	欧州	0.1	-12.2	20.5	-32.8	22.7
	アジア	0.9	-13.5	24.9	-36.2	36.1
	その他の地域	-2.9	-8.0	8.6	-8.0	9.3
輸入	北米	-0.4	-14.5	27.3	-33.8	29.5
	中南米	-2.1	-22.2	23.2	-43.8	19.5
	欧州	0.5	-10.3	19.9	-28.9	24.5
	アジア	-0.6	-11.8	23.1	-31.5	25.1
	その他の地域	1.5	-10.0	13.6	-22.6	18.0

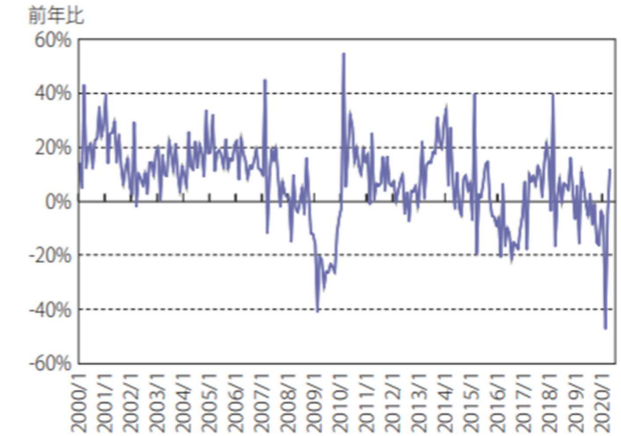
資料：世界貿易機関「TRADE STATISTICS AND OUTLOOK Trade set to plunge as COVID-19 pandemic upends global economy」。

図1　世界貿易の見通し

研究背景

供給ショック

感染抑制のためにフェイス･トゥ･フェイスのコミュニケーションに制限が発生し、人や物の移動に制限が生じ、その結果、供給制約が発生している。



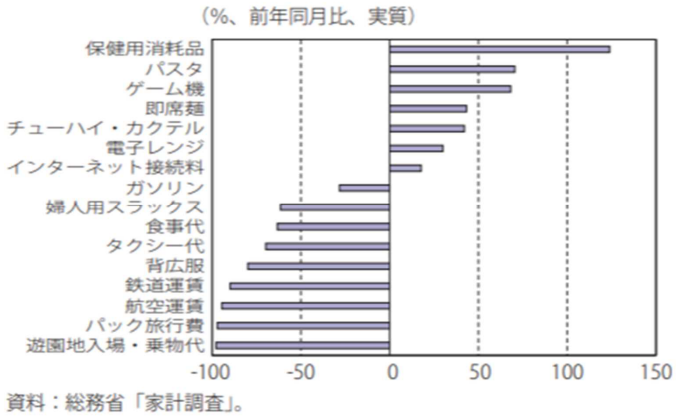
資料：財務省「貿易統計」。

図2　日本の中国からの輸入（前年同月比）

研究背景

需要のショック

商品供給側もコロナの影響を受けている。



資料：総務省「家計調査」。

図3　需要変化

研究背景

需要 ＝ 供給

過剰在庫

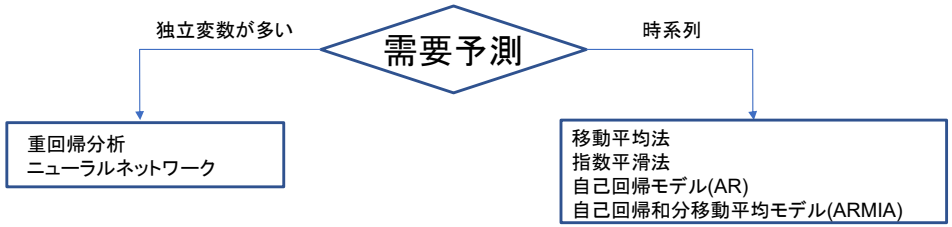
供給側（企業）

欠品による販売機会の損失

需要予測の精度向上

2．研究目的

研究対象と研究目的



Kourentzes N[(1)]（2013）は、不連続需要を調査するときに、時系列モデル、指数平滑法、移動平均法などの従来の予測方法の不十分さを分析し、ニューラルネットワーク法を導入して、不連続な需要予測の分野で従来の予測方法より精度が高いことを証明しました。

(1) Intermittent Demand Forecasts with Neural Networks , Nikolaos Kourentzes ,2013

研究対象と研究目的

既存研究の不足

しかし、その多くは特定の商品を対象とした需要予測に焦点を当てており、そもそものニューラルネットワークを用いた予測モデルを構築する上で重要な活性化関数の選択については十分な検討がなされていない。

そこで本研究では、活性化関数として、sigmoid、tanh、ReLUの3つを対象とし、各活性化関数を用いた場合の予測精度と学習時間の比較を行う。そして、この比較結果から活性化関数の選択が、需要予測の精度と学習時間に与える影響について考察することを目的とする。

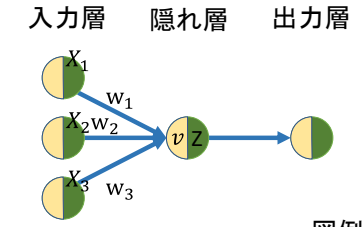
研究の目的

3．研究方法

研究方法

活性化関数

活性化関数は、ニューラルネットワークにおけるニューロン間の伝達される値を決定するための関数である。ニューロンは、複数のニューロンからの入力値の総和（線形変換）から活性化関数（非線形変換）を通して、次のニューロンへの出力値を伝達する。



$$v = \sum_{i=1}^{3} W_i \times X_i + b$$

$$Z = \varphi(v)$$

図5　簡単なニューラルネットワーク

図例

○印:ニューロン
X_i:独立変数の値
W_i:重み係数
b:バイアス
$\varphi()$:活性化関数

活性化関数は二回を使います

研究方法

活性化関数の意義

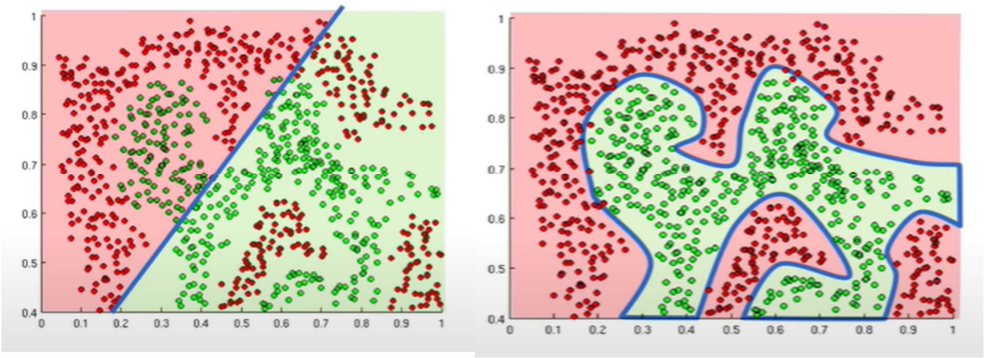


図4　Importance of Activation Functions,
MIT Introduction to Deep Learning, 2020

研究方法

$sigmoid(x) = \frac{1}{1+e^{-x}}$　　$tanh(x) = \frac{e^{x}-e^{x}}{e^{x}+e^{-x}}$　　$ReLU(x) = maximum(0, x)$

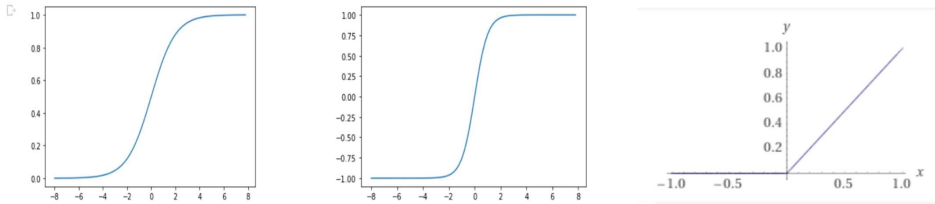


図7　よく使われている活性化関数

研究方法

4.　モデル構築

モデル構築

ニューラルネットワークの設定

層数：３層

入力層ユニット数：４;

隠れ層ユニット数：４;

出力層ユニット数：１;

隠れ層の決め方：$S = a + \sqrt{m+n}$

公式例：

S：隠れ層のユニット数;

m：入力層のユニット数;

n ：出力層のユニット数;

a ：１から１０までの整数

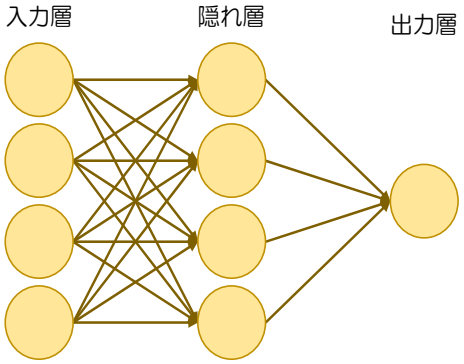


図5　今回のニューラルネットワークの構造

モデル構築

ほかのパラメータの設定

損失関数：平均二乗誤差$MSE = \frac{\sum(\hat{y}-y)^2}{m}$

- $\hat{y}$：需要の予測値,
- Y：実際の需要,
- m：データ数

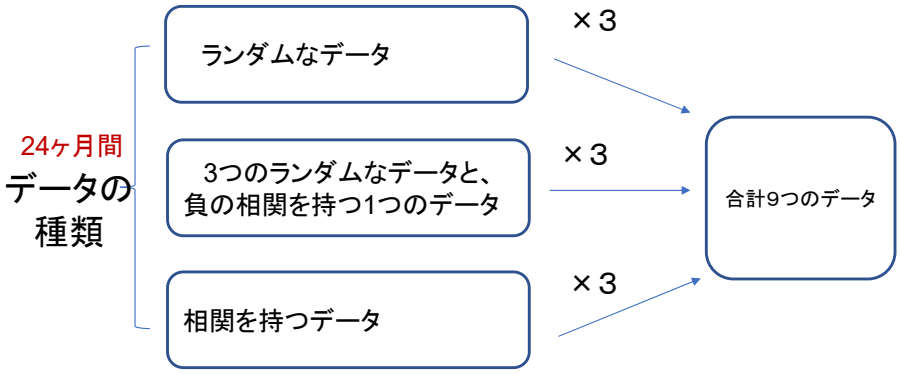
活性化関数：sigmoid関数、tanh関数、ReLU関数。

学習率（ｒ）：0.1、0.01、0.001から
適切にフィッティングされた学習率を選択することとした。

サイクル回数は１００００回を設定した。

モデル構築

データ作り



モデル構築

- グループ化

データ
（24）

ランダムに

→テストデータ:６

→トレニンーグデータ:１８

a1	a2	a3	a4	販売量
23495	577	16169	0.583	21228
25507	579	19246	0.283	24024
25722	539	18051	0.890	22062
21412	562	15066	0.368	28941
20982	594	11167	0.581	20990
23053	583	12699	0.073	27970

図6　テストデータ　1

a1	a2	a3	a4	販売量
20063	533	14163	0.749	28165
28816	516	18226	0.456	25716
23765	563	19520	0.904	21860
20882	552	11593	0.668	23829
22529	515	18647	0.281	23075
20994	578	15012	0.014	21360
25259	515	10330	0.295	27848
26404	522	18718	0.984	27417
29286	508	13925	0.761	23972
27111	575	15331	0.606	20109
24118	522	14637	0.268	24903
26068	565	15109	0.508	28373
25496	537	14574	0.794	29648
24635	531	11817	0.857	22489
22869	570	11253	0.093	26901
27326	586	12883	0.542	24468
29176	597	10381	0.415	28544
25024	570	11564	0.451	21389

図7　トレニンーグデータ　1

モデル構築

- データの前処理

- $z = \frac{x-\mu}{\sigma}$

- z：標準得点，
- X：変数の原始値，
- μ：変数の平均値，
- σ：変数の標準偏差

a1	a2	a3	a4	販売量
-1.80487	-0.51414	-0.05343	0.787921	1.108199
1.401656	-1.1513	1.368127	-0.29537	0.249726
-0.44886	0.564754	1.820765	1.360991	-1.10219
-1.50499	0.152006	-0.95261	0.488445	-0.41195
-0.90179	-1.1632	1.515601	-0.94238	-0.67632
-1.46385	1.095304	0.243564	-1.92954	-1.27741
0.09867	-1.18017	-1.39473	-0.89062	0.9973
0.517905	-0.93351	1.540214	1.65677	0.84621
1.573796	-1.41947	-0.13658	0.832287	-0.36157
0.777002	0.991823	0.355206	0.259217	-1.71597
-0.31946	-0.91723	0.112511	-0.99045	-0.03526
0.394803	0.642651	0.277676	-0.10311	1.181402
0.185268	-0.36912	0.090367	0.954296	1.628205
-0.12992	-0.61287	-0.87427	1.187221	-0.88154
-0.77701	0.803009	-1.07163	-1.63746	0.665178
0.855602	1.408477	-0.50146	0.022594	-0.18772
1.533505	1.786265	-1.37662	-0.44695	1.241184
0.012545	0.816708	-0.9627	-0.31385	-1.26748

図8　正規化

5.　研究結果

研究結果

トレニンーグデータの損失

平均		隠れ層-出力層		
		sigmoid	tanh	ReLU
入力層-隠れ層	sigmoid	0.480	0.095	0.368
	tanh	0.313	0.095	0.327
	ReLU	0.492	0.247	0.420
標準偏差		隠れ層-出力層		
		sigmoid	tanh	ReLU
入力層-隠れ層	sigmoid	0.017	0.046	0.046
	tanh	0.081	0.044	0.042
	ReLU	0.010	0.136	0.074

図9　損失の平均と標準偏差

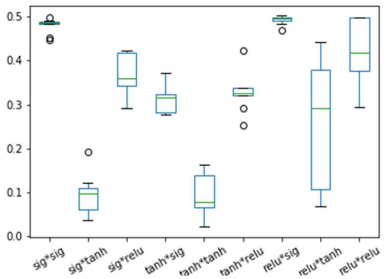


図10　トレニンーグ損失　箱ひげ図

1. sigmoid関数とReLU関数を用いた場合に損失が大きい傾向にあることが分かった。また、損失のバラツキは、隠れ層-出力層に、sigmoid関数を用いた場合に小さく、入力層-隠れ層に、ReLU関数を用いた場合に大きい傾向にあることが分かった。
2. 個別の組み合わせでは、sigmoid-tanh関数とtanh-tanh関数の組み合わせが、損失が小さく、精度が良いことが分かった。次点として、ReLU-tanh関数の組み合わせがあるが、損失のバラツキが大きく、安定しない。学習率の調整を何度も繰り返す必要がある。
3. なお、sigmoid- sigmoid関数と、ReLU- sigmoid関数の組み合わせは、損失が大きく、精度は悪いが、バラツキは小さく、結果は安定している。

研究結果

トレニンーグデータの時間

平均		隠れ層-出力層		
		sigmoid	tanh	ReLU
入力層-隠れ層	sigmoid	27,676	30,097	24,139
	tanh	25,065	23,820	24,608
	ReLU	23,870	24,357	25,450
標準偏差		隠れ層-出力層		
		sigmoid	tanh	ReLU
入力層-隠れ層	sigmoid	402	402	295
	tanh	177	435	135
	ReLU	295	155	702

図11　時間の平均と標準偏差

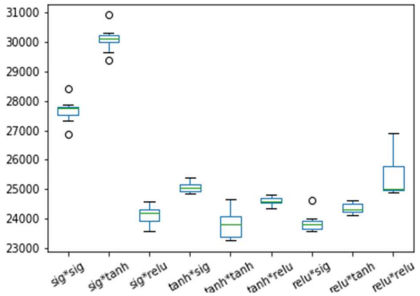


図12　時間の箱ひげ図

1. 表及び図から、全体傾向として、入力層-隠れ層に、ReLU関数を用いた場合に学習時間が短く、逆に、sigmoid関数を用いた場合に長くなる傾向にあることが分かった。
2. 個別の組み合わせでは、ReLU-sigmoid関数とtanh-tanh関数の組み合わせの学習時間が短い。しかし、tanh-tanh関数の組み合わせは学習時間のバラツキが大きく、結果は安定していない。
3. そして、sigmoid-tanh関数とsigmoid-t¥sigmoid関数の組み合わせは、学習時間が長く、バラツキも大きいことから、学習時間の観点からは望ましくない組み合わせといえる。

研究結果

テストデータ損失

平均		隠れ層-出力層		
		sigmoid	tanh	ReLU
入力層-隠れ層	sigmoid	0.603	0.091	0.554
	tanh	0.458	0.104	0.461
	ReLU	0.625	0.237	0.570

標準偏差		隠れ層-出力層		
		sigmoid	tanh	ReLU
入力層-隠れ層	sigmoid	0.149	0.066	0.165
	tanh	0.210	0.061	0.206
	ReLU	0.136	0.146	0.133

図13 損失の平均と標準偏差

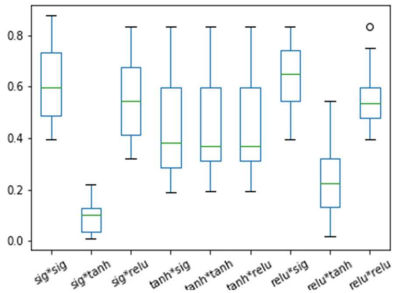


図14 損失の箱ひげ図

1. 表及び図から、全体傾向として、隠れ層-出力層に、tanh関数を用いた場合に損失が小さく、逆に、sigmoid関数とReLU関数を用いた場合に損失が大きい傾向にあることが分かった。また、損失のバラツキは、隠れ層-出力層に、tanh関数を用いた場合に小さく、sigmoid関数とReLU関数を用いた場合に大きい傾向にあることが分かった。
2. 個別の組み合わせでは、sigmoid-tanh関数とtanh-tanh関数の組み合わせが、損失が小さく、精度が良いことが分かった。また、損失のバラツキも小さく、結果も安定している。

6. まとめ

まとめ

まとめ

本研究では、活性化関数として、sigmoid、tanh、ReLUの3つを対象とし、需要予測の精度と学習時間に与える影響から望ましい活性化関数の組み合わせについて検討した。

その結果、トレーニングデータ及びテストデータともsigmoid-tanh関数とtanh-tanh関数の組み合わせが、損失が小さく、精度が良いことが分かった。特に、tanh-tanh関数は学習時間も短く、活性化関数の組み合わせとして、最も適切な組み合わせといえる。

なお、本研究で用いた入出力データは実際のデータではないため、本研究で得られた結果と同様の結果となるか実際のデータを用いた検証が必要である。

参考文献

- 小野田 崇・大場 英二: 翌日最大電力需要予測における最適なニューラルネットワーク構成の決定法, 電気学会論文誌, Vol.118, No.5, p.497-504, 1998.5.
- 麻生 英樹：ニューラルネットワーク情報処理, 産業図書, 1998.
- 千葉 周一:食品小売デリカ部門の需要予測と発注管理による食品ロス対策, 電気設備学会誌, Vol.40, No.10, p.638-641, 2020.
- 田辺 正志・吉田 律: ニューラルネットワークによる在庫予測の精度, 経営学紀要, Vol.1, No.2, p.127-133, 1994.1.
- 麻生英樹：ニューラルネットワーク情報処理, 産業図書, 1998.
- 赵馨宇，黄福珍，周晨旭:基于ReLU稀疏的积神经网络的MAXOUT卷积神经网络的数据分类算法,上海電力大学学报, ·Vol.36, No.3, 2020.6.
- 刘楚辉:基于BP网络的连锁超市供应链库存预测和控制研究, 甘肃政法学院, 2019.5.
- 腾杨刚・陈劲杰・葛桂林:基于PCA主成分分析和BP神经网络企业库存预测的研究, 软件工程 SOFTWARE ENGINEERING, Vol.21, No.7, 2018.7.
- David Simchi-Levi : Designing & Managing the Supply Chain, p305-307, 2010.
- Baiquan Lu・Junichi Murata・Kotaro Hirasawa: A New Method Based on Determining Error Surface for Designing Three Layer Neural Networks, 計測自動制御学会論文集, Vol.36, No.7, 589-598, 2000.
- Cousera, Neural Networks and Deep Learning, https://www.coursera.org/learn/neural-networks-deep-learning/home, 2020.
- Edx, Supply Chain Mnagament, https://www.edx.org/micromasters/mitx-supply-chain-management

ご清聴ありがとうございました