

概要

A

Abstract

A

謝辞

皆様，本当にありがとうございました．

目次

1	はじめに	1
1.1	研究背景	1
1.2	研究目的	1
2	物体認識実験	2
2.1	物体認識	2
2.1.1	畳み込み層	2
2.1.2	プーリング層	3
2.1.3	全結合層	3
2.1.4	CNN のモデル	3
2.1.5	CNN の学習方法	5
2.2	RNN (Recurrent Neural Network) の概要	5
2.3	LSTM (Long Short - Term Memory) の概要	5
3	ニューラルネットワークモデル	7
4	データ間の特性	8
4.1	電力使用量	8
4.2	天気	8
4.3	気温	8
4.4	降水量	8
4.5	各データ間の相関係数	8
5	考察	10
6	まとめと今後の課題	11
6.1	まとめ	11
6.2	今後の課題	11

1 はじめに

1.2 研究目的

1.1 研究背景

2011 年 3 月 11 日に発生した東日本大震災以降, 低コストかつ効率的な発電方法であり全体の 2 割から 4 割を占めていた原子力発電は安全性に問題があるとされ 2021 年 5 月現在では停止中の 2 基も合わせて合計 9 基の原子力発電所が稼働するのみとなっている [1]. それによって, コストの高い火力発電の量が増え発電コストの増加による電気料金の値上げにつながり, 我々消費者の負担増加に繋がっている [2]. 電力発電において電力使用量の予測を行うことにより, 長期的には季節ごとの電力使用量に基づいた最適な時期での化石燃料の調達, 短期的には電力使用量が増える時間帯において各発電の稼働状態を適切に決めることが求められている. 大まかな電力使用量の推移は経験から判別が可能であると思われるが, 深層学習を用いて予測を行うことによってより細かい予測により今後の計画を立てることができるようになる. 適切な予測を行い適切な量の電力供給や燃料調達を行うことができれば電力会社は過剰な発電や, 供給不足による突発的に高コストな発電方法, 効率の悪い発電方法を行う必要がなくなると考えられる.

2 物体認識実験

本章では、物体認識の現状、手法及びオープンソースのコンピュータビジョンライブラリである Open Source Computer Vision Library 通称 OpenCV を用いた物体認識実験の結果について述べる。

2.1 物体認識

ニューラルネットワークとは、人間の脳の仕組みから着想を得たものであり、神経回路網をコンピュータ上で表現しようと作られた数理的モデルである。人間の脳はニューロンと呼ばれる神経細胞の結びつきで情報伝達や記憶の定着が行われており、その人間の脳神経系の持つ強力な学習能力を数学的に応用することにより、画像認識や音声認識などへの利用が期待され研究が行われている。

通常ニューラルネットワークは、入力層、出力層、隠れ層から構成されており、これらの層と層の間にニューロン同士のつながりを示す W があり、これを重みと呼ぶ。ニューラルネットワークの隠れ層は多層にすることができ、隠れ層が多数存在する多層構造のニューラルネットワークのことをディープラーニングと呼ぶ。そのディープラーニングの主力となっているものの一つに畳み込みニューラルネットワークがあり、畳み込み層とプーリング層を1つのペアとして、それらが複数回重ね合わせて構成される順方向性ニューラルネットワークである。

2.1.1 畳み込み層

畳み込みニューラルネットワークに入力としての画像が与えられたとき、その画像の特徴量を抽出する操作のことを畳み込みと呼ぶ。

畳み込み層は単純型細胞をモデルに考えられたものであり、元の画像にフィルタをかけて元の画像よりも少し小さい特徴マップを出力する、またこのフィルタは画像全体をスライドするので特徴がどこにあって抽出することができ、その際にフィルタの数だけ特徴マップが出力される。

ここで例として、対象とする画像を $X \times Y$ Pixels の RGB の階調値とし、 k 番目の階調の素子 (i, j) の画素値を I_{ij}^k とする。ただし、 $k = 1$ が R、 $k = 2$ が

G、 $k = 3$ が B とする。最初の層の畳み込み層の a 番目のフィルタの (i, j) 番目の素子の内部状態 $y_{ij}^{(1)(a)}$ は、

$$y_{ij}^{(1)(a)} = \sum_{k=1}^3 \left(\sum_{x \in W} \sum_{y \in W} w_{ij}^{(1)(a)(k)} I_{i+x, j+y}^{(k)} + b_{ij}^{(1)(a)(k)} \right) \quad (1)$$

その出力 $\tilde{y}_{ij}^{(1)(a)}$ は、

$$\tilde{y}_{ij}^{(1)(a)} = \max \left(y_{ij}^{(1)(a)}, 0 \right) \quad (2)$$

ℓ 番目の層の畳み込み層の内部状態 $y_{ij}^{(\ell)(a)}$ は、

$$y_{ij}^{(\ell)(a)} = \sum_{\alpha=1}^{N(\ell-1)} \sum_{x \in W} \sum_{y \in W} w_{ij}^{(\ell)(a), \alpha} z_{i+x, j+y}^{(\ell-1)(\alpha)} + b_{ij}^{(\ell)(a)} \quad (3)$$

で表される。ここで、 $w_{ij}^{(1)(a)(k)}$ は入力層と畳み込み層間のシナプス結合加重、 W は各素子が入力を受ける範囲を与える配置集合（受容野）、 $b_{ij}^{(1)(a)(k)}$ は閾値である。最終層 (L) の内部状態 $y_k^{(L)}$ は、前層のプーリング層の出力 $z_{ij}^{(L-1)(a)}$ との全結合として、以下のように与えられる。

$$y_k^{(L)} = \sum_{\alpha=1}^{N(L-1)} \sum_i \sum_j w_{kij}^{(L)(\alpha)} z_{ij}^{(L-1)(\alpha)} + b^{(L)k} \quad (4)$$

これらの内部状態は活性化関数と呼ばれる非線形関数を経て、畳み込み層の出力としてプーリング層へ伝播する。活性化関数の例として、ステップ関数、シグモイド関数、ソフトマックス関数、恒等関数、ReLU などが挙げられる。ステップ関数は、階段関数とも呼ばれる閾値を境にして出力が切り替わるようにする関数である。シグモイド関数は、式で表すと以下のように表される。

$$h(x) = \frac{1}{(1 + e^{-x})} \quad (5)$$

入力した値が大きいほど 1 に近づき、入力した値が小さいほど 0 に近づく関数である。ソフトマックス関数は、式で表すと以下のように表される。

$$y_i = \frac{e^{x_i}}{\sum_{i=1}^n e^{x_i}} \quad (6)$$

恒等関数は、受け取った値をそのまま出力したい場合に用いる関数である。ReLU は 0 以下の値は、0 にして 0 以上の値の場合にそのままの値を出力するという関数である。ReLU を式で表すと以下になる。

$$f = \max(0, x) \quad (7)$$

入力層が 6×6 , 畳み込みフィルタが 3×3 のときの畳み込み操作の例を以下に示す. ただし入力を x , 出力を y とし, 畳み込みフィルタの重みを w としたときの畳み込み層での点 (a,b) についての情報量を求めたい場合を図に示す. その後左上から右下への畳み込みを考えストライドを 1 としたときの $(a+1,b)$ での畳み込みを図に示す.

2.1.2 プーリング層

畳み込み層は単純型細胞をモデルに考えられたものであるが, プーリング層は複雑型細胞をモデルにして作られたモデルである.

画像から重要な情報を残して縮小する方法であり, 特徴マップからとある一部分の領域を取り出し, そのうちの最大値を取り出す最大プーリング (max pooling) や, 特徴マップからとある一部分の領域を取り出し, その領域の平均値を取る平均プーリング (average pooling) などがあるが, 画像認識の分野では最大プーリングが使われていることが多い. また, プーリング層の出力を $z_{ij}^{(1)(a)}$ とすると以下のように表される.

$$z_{ij}^{(1)(a)} = \max_{x \in W, y \in W} \tilde{y}_{i+x, j+y}^{(1)(a)} \quad (8)$$

最大プーリングと平均プーリングの例をそれぞれ図 2 と図 3 に示す. 畳み込み層やプーリング層で行われる手法にパディングというものがある. パディングとは入力として与えられた画像のピクセルの周りにピクセルを配置し囲むという方法である. パディングを行うことによるメリットは, 畳み込み層やプーリング層を経ることによって入力画像のサイズが小さくなっていくので, パディングを行うことにより画像サイズを小さくしなくて済むという点と, 畳み込み層やプーリング層を経ることにより外側から縮小されていく画像にパディングを行うことで, 画像の端のピクセルも畳み込みの回数を増やすことができるので精度を高めることができるという点がある. パディングの手法の一つにゼロパディングと呼ばれるピクセルの周りをゼロのピクセルで囲うというものがある, このゼロパディングの例を図 4 に示す.

2.1.3 全結合層

畳み込み層, プーリング層を経ても入力として与えられた画像の配列のままであるが, これらの層を経て特徴データが取り出された入力として与えられた画像データを一つのノードに結合し, 活性化関数によって変換された値を出力する層である.

2.1.4 CNN のモデル

CNN の最初のモデルは LeCun たちにより提案された LeNet と呼ばれるものであり, その構造は畳み込み層とプーリング層が交互に並んでいる構造である. 第一層の畳み込み層で 32×32 の入力データを 5×5 のフィルタで畳み込みを行い 28×28 の特徴マップを出力する. この出力を 2×2 のフィルタでプーリングを行い, 次に 14×14 の出力にする. この特徴マップを 5×5 のフィルタで畳み込みを行い, 10×10 の特徴マップとしてプーリング層に出力する. 2×2 フィルタを持つプーリング層は 5×5 の特徴マップを出力して全結合層に入力信号として渡され, 出力は活性化関数を通して 10 クラスの分類が行われる.

14 年後の 2012 年にトロント大学の研究者である Alex Krizhevsky たちが提案したモデルが AlexNet と呼ばれる CNN で畳み込み層とプーリング層が交互に並ぶ構造である. ネットワーク構造を図に示す. この AlexNet には五層の畳み込み層と三層のプーリング層が存在する. ネットワークの最終部分には分類識別のために三種類の全結合層 (Fully Connected Layer : FC) が配置され, 最後の層からは softmax 関数を通じて, 1000 クラスに対する確率分布が出力される. これは ImageNet と呼ばれる 1400 万枚以上の画像のデータセットに採用されている 1000 クラス分類に対応させるためである. また畳み込み層が二つに分割されている理由としては GPU の性能によるものである.

オックスフォード大学の Visual Geometry Group という研究グループが開発したモデルに VGG がある. 畳み込み層とプーリング層から構成されており, AlexNet の進化系で層の深さを 16 層, 19 層にしたモデルである. 非常に小さな 3×3 の畳み込みフィルタを用いて, ネットワークの深さを 16 層から 19 層に増加させることにより識別の精度を改善している. この

表 1: AlexNet の層構造

順番	層	カーネルサイズ
1	conv/4	$11 \times 11 \times 3 \times 96$
2	max-pooling	-
3	conv	$5 \times 5 \times 4 \times 256$
4	max-pooling	-
5	conv	$3 \times 3 \times 256 \times 384$
6	conv	$3 \times 3 \times 192 \times 384$
7	conv	$3 \times 3 \times 192 \times 256$
8	max-pooling	-
9	FC-4096	$1 \times 1 \times 4096$
10	FC-4096	$1 \times 1 \times 4096$
11	FC-1000	$1 \times 1 \times 1000$
-	softmax	-

小さなフィルタを持つ畳み込み層を 2 から 4 回連続して重ね、それをプーリング層でサイズを半分にするを繰り返して行う構造が特徴である。

2014 年の ILSVRC (ImageNet Large Scale Visual Recognition Challenge) と呼ばれる 2010 年から始まった大規模画像認識の競技会で 2 位という成績を残したことで有用性が実証され、更にシンプルなアーキテクチャであり実用に向いている。

入力画像は 224×224 の RGB 画像であり、最初の 3×3 の畳み込み層でのフィルタのチャンネル数は 64 で、マックスプーリングごとに 2 倍になり、128, 256, 512 と増加していき、出力は 1×1000 次元の配列である。また VGG16 のネットワークの構造を図に示す。また FC-4096 は並列次数 4096 の全結合層を表す。次に先述した 2014 年の ILSVRC の優勝モデルである GoogleNet は基本的なネットワークの構造は CNN と同様であるが、特徴としてネットワークの構造が縦方向だけでなく横方向にも深さを持つことが挙げられる。それは複数の畳み込み層やプーリング層から構成される Inception モジュールと呼ばれる小さなネットワークを作成し、これらを通常の畳み込み層のように重ねていき CNN を構成していると言える。Inception モジュールを組み込んだ GoogleNet のネットワークの構造を図に示す。

ここで conv3d は 3×3 のフィルタを持つ畳み込み層であり、 $112 \times 112 \times 64$ は出力のサイズを意味

表 2: VGG16 の層構造

順番	層	カーネルサイズ
1	conv3-64	$224 \times 224 \times 64$
2	conv3-64	$224 \times 224 \times 64$
-	max-pooling	$112 \times 112 \times 128$
3	conv3-128	$112 \times 124 \times 128$
4	conv3-128	$112 \times 124 \times 128$
5	conv3-128	$112 \times 124 \times 128$
-	max-pooling	$56 \times 56 \times 256$
6	conv3-256	$56 \times 156 \times 256$
7	conv3-256	$56 \times 156 \times 256$
8	conv3-256	$56 \times 156 \times 256$
-	max-pooling	$28 \times 28 \times 516$
9	conv3-512	$28 \times 28 \times 516$
10	conv3-512	$28 \times 28 \times 516$
-	max-pooling	$28 \times 28 \times 516$
11	conv3-512	$14 \times 14 \times 516$
12	conv3-512	$14 \times 14 \times 516$
13	conv3-512	$14 \times 14 \times 516$
-	max-pooling	$7 \times 7 \times 516$
14	FC-4096	$1 \times 1 \times 4096$
15	FC-4096	$1 \times 1 \times 4096$
16	FC-1000	$1 \times 1 \times 1000$
-	softmax	$1 \times 1 \times 1000$

している。また Inception モジュールで 5×5 の畳み込み層を 3×3 の畳み込み層を 2 つ重ねたものに変更し、改良を施したモデルのことを Inception-V2 と呼ぶ。さらに総構造を深くしたものの代表例として、ResNet と呼ばれる Microsoft 社の Kaiming He の研究グループによって提案され、2015 年の ILSVRC で優勝した CNN モデルが挙げられる。それまでのネットワーク構造ではあまり層を深くしすぎると性能が逆に落ちてしまうという問題があったがそれを残差学習構造と呼ばれるある層への入力をバイパスして層をまたぎ奥の層へと入力し、勾配の消失や発散を防ぎ超多層のネットワークの実現を可能にした。

このように CNN の層の数を増やしよりディープな構造にしたものを DNN (Deep Neural Network) とも呼ぶ。

表 3: GoogleNet の層構造

順番	層	カーネルサイズ
1	conv7	$112 \times 112 \times 64$
2	max-pooling3	$56 \times 56 \times 64$
3	conv3	$56 \times 56 \times 192$
4	max-pooling3	$28 \times 28 \times 192$
5	inception	$28 \times 28 \times 256$
6	inception	$28 \times 28 \times 480$
7	max-pooling3	$14 \times 14 \times 480$
8	inception	$14 \times 14 \times 512$
9	inception	$14 \times 14 \times 512$
10	inception	$14 \times 14 \times 512$
11	inception	$14 \times 14 \times 528$
12	inception	$14 \times 14 \times 832$
13	max-pooling3	$7 \times 7 \times 832$
14	inception	$7 \times 7 \times 832$
15	inception	$7 \times 7 \times 1024$
16	avg-pooling7	$1 \times 1 \times 1024$
17	dropout(40%)	$1 \times 1 \times 1024$
18	linear	$1 \times 1 \times 1000$
19	softmax	$1 \times 1 \times 1000$

2.1.5 CNN の学習方法

機械学習には教師あり学習、教師なし学習、強化学習の三つの学習法がある。教師あり学習とは、学習データに正解ラベルをつけて学習する方法である。分類や予測などは教師あり学習のクラスである。それに対して教師なし学習は、学習するデータに正解ラベルをつけずに学習する方法である。こちらの教師なし学習は、様々な入力を読み取っていくうちに自律的に対象を認識できるようになっていく学習方法であり、教師あり学習に比べ実装難度が高い。クラスタリングや次元削減などは教師なし学習のクラスである。強化学習は価値を最大化するような行動を学習する方法である。囲碁や将棋で目先の取れるコマがあったとしてもそれが敗北につながるような手だとしたら、取らずに勝利に近づく手を打つことが強化学習の目的である。強化学習のアルゴリズムには、Q 学習、Sarsa、モンテカルロ法などがある。

Google DeepMind の開発した AlphaGo はコンピ

ュータ囲碁プログラムであり、2015 年に人間のプロ囲碁棋士をハンディキャップ無しで破った初のコンピュータ囲碁プログラムである。膨大な数の対局データを教師あり学習として学習し、さらに強化学習を行い勝つためのパターンを特徴量として学習させており、その後コンピュータ同士を対局させ競わせていくことにより強力なコンピュータ囲碁プログラムが誕生した。

2.2 RNN (Recurrent Neural Network) の概要

ニューラルネットワークの出力を別のニューラルネットワークの入力として利用するような再帰的構造を持ったニューラルネットワークのことを RNN (Recurrent Neural Network) と呼ぶ。

時間を追って得られたデータを時系列データと呼び、画像などのデータが一つのベクトル x_n で表されるのに対し、 $(x(1), \dots, x(t), \dots, x(T))$ という T 個のデータが 1 つの入力データ群このデータを用いて更に次に得られるであろうデータを予測する方法の一つである。時系列データには、気温の推移や降水量の変化などの気象データ、株価や売上の推移などが挙げられ、これらのデータは一般的なデータの集まりではなく、データの並び自体に意味を持つ。

時刻 t での入力を $x(t)$ 、入力層と隠れ層の間の重みを U 、隠れ層と出力層の間の重みを W とすると隠れ層の値 $s(t)$ とネットワークの出力 $y(t)$ は、以下のように表すことができる。

$$s(t) = f(Ux(t) + Ws(t-1)) \quad (9)$$

$$y(t) = g(Vh(t)) \quad (10)$$

また RNN の基本構造の例を図 5 に示す。

2.3 LSTM (Long Short - Term Memory) の概要

LSTM (Long Short - Term Memory) は 1995 年に RNN の拡張として 1995 年に登場した時系列データを扱うモデルである。Long Term Memory (長期記憶) と Short Term Memory (短期記憶) という神経科学における用語から取られており、従来の RNN で学習できなかった長期依存を学習できるという特徴

を持つ. LSTM は RNN の中間層のユニットを LSTM block と呼ばれるメモリと 3 つのゲートを持つブロックに置き換えることで実現されている. 畳み込み層の内部状態を以下のように表す.

$$y_{ij}^{(\ell)(a)}(t) = y_{ij}^{(\ell)(a)} + \sum_{\tau=1}^T \sum_{\alpha=1}^{N(\ell)} \sum_{x \in W} \sum_{y \in W} v_{ij}^{(\ell)(a, \alpha)} y_{i+x, j+y}^{(\ell)(\alpha)}(t - \tau) \quad (11)$$

また, LSTM の基本構造の例を図 6 に示す.

3 ニューラルネットワークモデル

4 データ間の特性

本研究で電力使用量の予測に用いるデータは、電力使用量、天気、気温、降水量を用いる。また、LSTM で入力としてこれらのデータを扱う場合に、データ間に関連性がない時に入力として適していないと考えられるので、データ間の相関関係として相関係数の計算を行う。

4.1 電力使用量

東京電力パワーグリッド株式会社が提供している 2016 年 4 月から 2020 年 12 月までの東京電力が供給した電力使用量のデータであり、年度、日にち、1 時間ごとの電力使用量（万 Kw）の 3 要素が csv ファイルで提供されている。2016 年は 4 月からのデータの公開のため、6600rows * 3columns, 2017 年から 2019 年は 8760rows * 3columns, 2020 年は閏年であり通常よりも 1 日分多いため 24 列多い 8784rows * 3columns のデータである。また、用いるデータの一例を図 0 に示す。これらのデータの日にちと時間を結合させて日時のデータとする。また、その際に日付と時間が並んでいるだけの文字列であるので、Python のデータ解析用ライブラリである pandas を用いて文字列を日付データに変換する。また、データを可視化してグラフにしたものの例として 2019 年、2020 年のグラフ及び、2020 年 4 月の 5 日から 11 日、12 日から 18 日、19

日から 25 日の 3 週分のグラフをそれぞれ図に示す。

4.2 天気

4.3 気温

4.4 降水量

4.5 各データ間の相関係数

数値計算を効率的に行うための拡張モジュールである、numpy を用いて相関係数を計算する。相関係数は、2 組のデータ間の関係性を示す指標の一つとして用いられているものである [Numpy データ処理入門 p346 ~]。相関係数を r とすると r は、 $-1 \leq r \leq 1$ の範囲に値を取り、 r の絶対値が大きいほど、大きい負の相関や正の相関を持つ。例えば、2 組のデータ、 X と Y が存在しているとし、 X の分散を V_X 、 Y の分散を V_Y とし、 X の標準偏差を S_X 、 Y の標準偏差を S_Y とし、共分散を S_{XY} とおくと、相関関係 r は以下の式 () で表される。

$$r = S_{XY} / \sqrt{V_X V_Y} = S_{XY} / S_X S_Y \quad (12)$$

2018/1/1 18:10 UPDATE

```
DATE, TIME, 実績(万kw)
2016/4/1, 0:00, 2555
2016/4/1, 1:00, 2433
2016/4/1, 2:00, 2393
2016/4/1, 3:00, 2375
2016/4/1, 4:00, 2390
2016/4/1, 5:00, 2467
2016/4/1, 6:00, 2631
2016/4/1, 7:00, 2775
2016/4/1, 8:00, 2965
2016/4/1, 9:00, 3159
2016/4/1, 10:00, 3211
2016/4/1, 11:00, 3265
2016/4/1, 12:00, 3160
2016/4/1, 13:00, 3288
2016/4/1, 14:00, 3361
2016/4/1, 15:00, 3362
2016/4/1, 16:00, 3406
2016/4/1, 17:00, 3438
2016/4/1, 18:00, 3501
2016/4/1, 19:00, 3381
2016/4/1, 20:00, 3242
2016/4/1, 21:00, 3084
2016/4/1, 22:00, 2973
2016/4/1, 23:00, 2811
2016/4/2, 0:00, 2613
2016/4/2, 1:00, 2492
```

図 1: 用いる電力使用量データ例

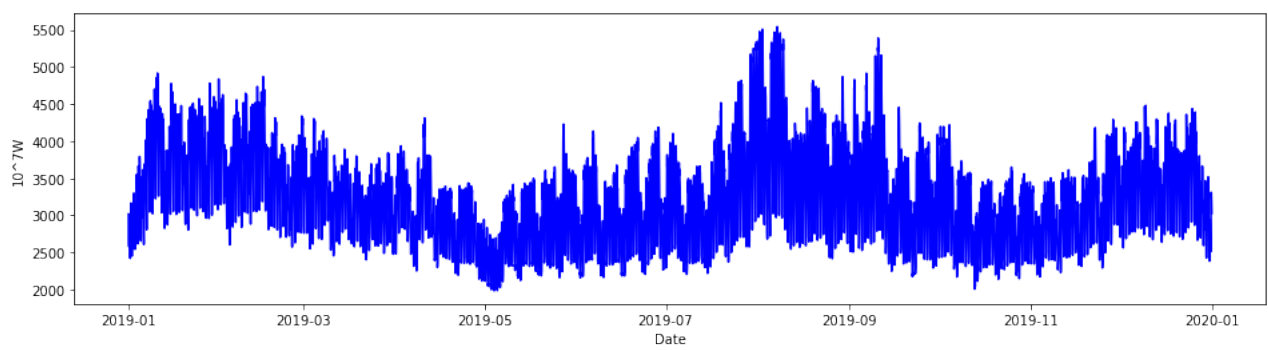


図 2: 2019 年の電力使用量のグラフ

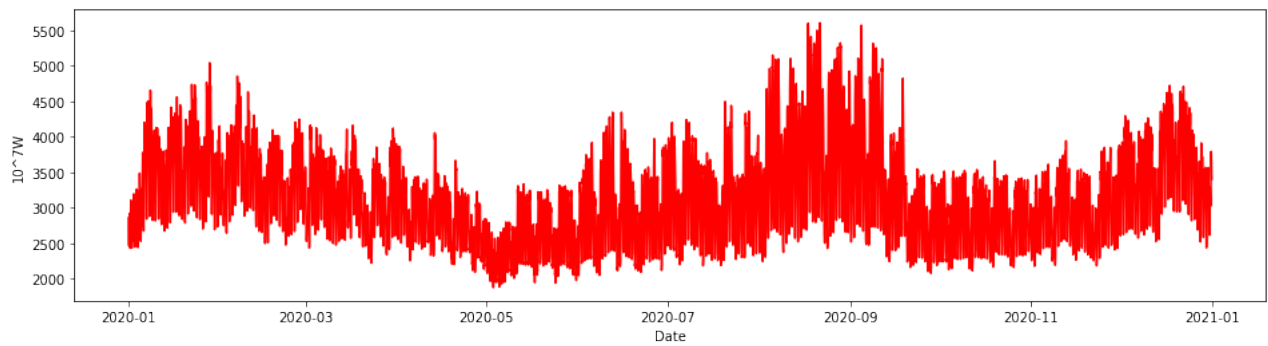


図 3: 2020 年の電力使用量のグラフ

5 考察

本実験では、フィルタサイズとフィルタの枚数を別とした、3種類の層構造のCNNによって4種類の表情の分類を行った。1層では、どのカーネルサイズにおいてもフィルタ数を増やすことにより同等以上の識別率となった。カーネルサイズによる違いはあまり大きい数値の変化はなかった。この結果から、1層の場合ではカーネルサイズによる変化はあまりないが、フィルタ数の増加による識別率向上を望むことができると考えられる。一般的にカーネルサイズを小さくすることにより、得られる特徴量が増えるので識別率の向上が考えられるが、1層の場合は畳み込み層が1層なのであまりその恩恵を受けられなかったのではないかと考えられる。フィルタの数を増やすことによって、特徴マップの数を増やせるので識別率が向上することが考えられるが結果のようにフィルタの数の増加による識別率の向上が見られた。フィルタのサイズで識別率の2層の場合にはフィルタの数が48と48、48と96のときに比べて、96と96のときの識別率はあまり低下しなかった。カーネルサイズが 3×3 のときに比べ

て 5×5 の場合には高い識別率を示した。これはカーネルサイズが大きいほど識別率が下がるが畳み込み層が2層に増えたことによる特徴量の減少が理由だと考えられる。3層の場合には、1層、2層のときに比べて識別率が落ちてしまっていた。以上よりフィルタの枚数については、層数が1の場合には増やすことによる認識率の向上を望むことができたが、総数が2層、3層と増えていくとフィルタの枚数が増えるにつれて識別率の低下も見られるようになったため、単純に増やすことによる識別率の向上は認められないのではないかと考えられる。識別率が上がらなかった理由として考えられる点は、かさ増しをしたが十分とは言えない訓練データの枚数と、表情画像を選ぶ際に動画をフレームごとに切り、それをエクセルシートで確認しながら選ぶ際に目視で選んだことによる主観の違い、などが原因だと考えられる。訓練データの数が少ないと得られる特徴が少なくなり、それに伴って識別率も低下してしまったと考えられる。以上の結果からわかることは、層数、フィルター数、カーネルサイズの違いによって識別率の変化があるため、表情認識を行うにあたってネットワークの構造の見直しを行うことにより認識率を挙げることは可能であると考えられる。

6 まとめと今後の課題

6.1 まとめ

本研究では, CNN の構造の違いによる表情認識性能を明らかにすることを目的とした. その結果, 層の深さ, カーネルサイズやフィルタの枚数を適したものにより認識率を高めることが可能であることがわかった. 識別率自体は半分を上回るものがなかったが, その理由として Bothered と Sadness のような悩みと悲しみという似たような表情で人間でも判断の難しい表情を選択したことや, 髪の毛や肌など表情認識に必要な部分の特徴を得ていたり, 適切なデータの数を用意できなかったことが考えられる. よって今後の課題としては, 表情認識に用いる適切な表情選択や, 表情変化がわかる目や口などを重点的に判別することや, 表情認識に用いるデータを収集し, 適切な数で表情認識をすることなどが考えられる.

6.2 今後の課題

課題

参考文献

- [1] 小松大起, 岡田直人, 濱田利行, 黒岩丈介, 小高知宏, 諏訪いずみ, 白井治彦, 令和 1 年度電気・情報関係学会
北陸支部連合大会 発表, F2-13 ネットワークの構造の違いによる表情認識性能の差異 (2019.8)