令和元年度　特別研究報告書

機械学習を用いた野鳥の鳴き声分類手法に関する研究

龍谷大学　大学院　理工学研究科　情報メディア学専攻

T18M062　藤岡 優也

指導教員　三好 力　教授

目次

内容梗概

自然環境保護等の目的で野生動物の生息地分布の調査を行うことがある。その一環として、調査区域に設置された録音機から得た野生動物の鳴き声の音声データを音響情報として分析し、その種類を判別する方法がある。これは、捕獲等を行うことによって野生動物にストレスを与えてしまうことや、実際に現地に赴いての調査にはコストがかかる等の理由で行われる。現状この方法には非常に長時間に渡る録音データを専門家が直接聴いて判別しているという現状がある。この現状を打開するために機械学習が注目されている。本研究では特にニュージーランドの野鳥の鳴き声や環境雑音の録音データを例にして、①低コストでの機械学習②長時間録音データにおける音声認識システムに関しての検討を行う。

Abstract

In some cases, the distribution of habitats of wild animals is investigated for the purpose of protecting the natural environment. There is a method to analyze the sound data of the cry of wild animals obtained from recorder installed in the survey area as acoustic information and determine the type. This is performed because stress is given to wild animals by performing capture or the like, and it is costly to carry out a survey when actually going to the site. At present, there is a current situation in which this method is such that specialists directly listen to recorded data for a very long time and make a judgment. We are using the recording data of New Zealand's wild birds and environment noise , We consider about ①Machine learning at low cost②Speech recognition system for long recording data.

第1章 はじめに

1.1 研究背景

昨今、AI 技術の発展により、様々なものに関して機械学習等による自動化が図られてきている。しかし、未だに計算機の計算コストや、人手によるラベル付けの問題、機械学習による自動化が難しいものへの対応等、解決すべき問題は山積みとなっている。

例えば、ニュージーランド自然保護局(DOC)では、自然環境保護の一環として、野生動物の音響データを分析することによって、その種類を判別する手法を用いている。この手法は主に野生動物にストレスを与えないことや、人手での調査にかかるコストを削減するために行われ、調査区域に設置された録音データによって得られた音響データを、分析し、判別を行う。(図 1.1)

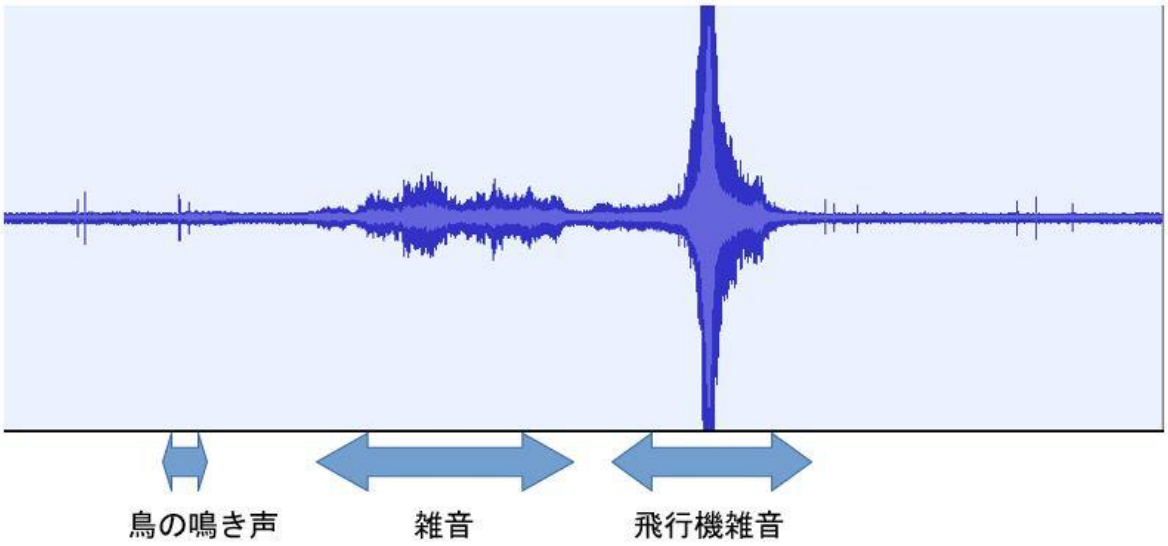

図 1.1 録音機によって得られる録音データの例

この手法には、かなりの長時間に渡る録音データを専門家が直接聴き分けて判別を行っているという現状があり、機械学習への自動化が期待されている。しかし、現状では長時間録音データに含まれる雑音と野生動物の鳴き声との混同等の問題で完全な自動化には至っていない。また、機械学習を行うデータの前処理の 1 つである特徴抽出による演算コストや、判別対象となる音響データの人手による切り出しによるコストを削減することも、解決するべき課題である。

また、生息地分布を調査する関係上、既知の野生動物だけでなく、未知の野生動物を検出する機構も必要とされている。しかしこれは現在の教師あり機械学習の枠組みでは解決が困難な課題である。

1.2 研究目的

野生動物の音響情報を分析する手法には、問題点があった。1 つ目の問題は機械学習を行うための特徴抽出の問題、2 つ目の問題は、学習用データに人手によるラベル付けが必要であるという点、3 つめの問題は対象部分の人手による切り出しが必要であるという点、4 つ目として、未知の生物の検出が挙げられる。本研究はこの 4 つの課題を解決するため、①特徴抽出に畳み込みニューラルネットワークを利用した機械学習②対象音声を切り出す必要がない音声認識システム③未知の野生生物の検出に関しての検討を行うことを目的とする。なお、人手によるラベル付けに関しては平成 30 年度卒業研究として既に発表を行っている。

①に関しては、特徴抽出に演算コストがかかるという点を改善するため、従来の音響特徴抽出法を用いずに 1 次元畳み込みニューラルネットワークによる特徴抽出によって学習を行う。②に関しては、人手によって鳥の鳴き声を鳥が一回鳴くごとに区切るのではなく、細かい間隔で分割して学習を行う手法を検討する。この手法を検討することによって、音声データの時系列情報は失われるが、人手による鳴き声の定義と、特徴抽出は行わずに機械学習を行うことができる。

③に関しては、2 値分類を行う分類器を並列に適応することによって、各 2 値分類器が既知かそうでないかを分類し、未知のデータを含めてデータを判別する手法を検討する。その未知データに人手でのラベル付けを行い新たな 2 値分類機を生成し加えていくことを繰り返すことにより、実用的な機械学習のシステムが提案できるのではないかと考えた。

第2章 基本事項

2.1 畳み込みニューラルネットワーク

2.1.1 畳み込みニューラルネットワークとは

畳み込みニューラルネットワーク(Convolutional Neural Network:CNN)[1]~[4]は、一般的な全結合層のみで構成されるネットワークとは異なり、主に畳み込み層とプーリング層から構成されるニューラルネットワークである。畳み込みニューラルネットワークは、畳み込み層、プーリング層、全結合層の 3 つの結合によって構成される。畳み込みニューラルネットワークのネットワーク構成例を図 2.1 に示す。ここで、Conv は畳み込み層、Pool はプーリング層を表す。

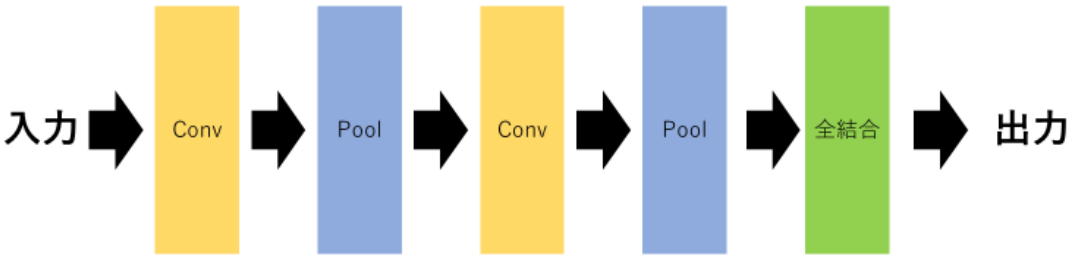

図 2.1:畳み込みニューラルネットワークの構成例

図に示したように、畳み込みニューラルネットワークは畳み込み層とプーリング層を繰り返し、最後に全結合層を通して出力するような構成になっている。特に畳み込み層は入力の局所的な特徴を抽出する役割を担い、プーリング層は局所ごとの特徴をまとめ上げる役割を担っている。畳み込み層とプーリング層によって入力データの特徴量を得ることができるため、特徴抽出法による特徴抽出を行う必要がない。この 2 つの層によって得られた特徴量を全結合層及び出力層へと渡すことにより、入力データの識別が可能となる。畳み込み層とプーリング層の処理の繰り返しは、それぞれの層が異なる特徴を識別するように学習が行われる。全結合層では取り出された特徴量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価を出力する。出力層では全結合層からの出力を元に、ソフトマックス関数等の活性化関数によって識別可能な形に変換し、分類を行う。畳み込みニューラルネットワークの多くは画像等の 2 次元データに使用されることが多いが、音や波形等の 1 次元データにおいても適用することができる。

2.1.2 畳み込み層

畳み込み層では入力に対して畳み込み演算が行われる。畳み込み層は重みとバイアスの 2 つの学習可能なパラメータを持ち、特に重みはフィルタと呼ばれる。フィルタは入力と同じ次元数の行列である。それぞれのフィルタは入力の全体に沿って畳み込み演算が行われる。特に入力データが 2 次元の場合の畳み込み演算の例を図 2.2 に示す。

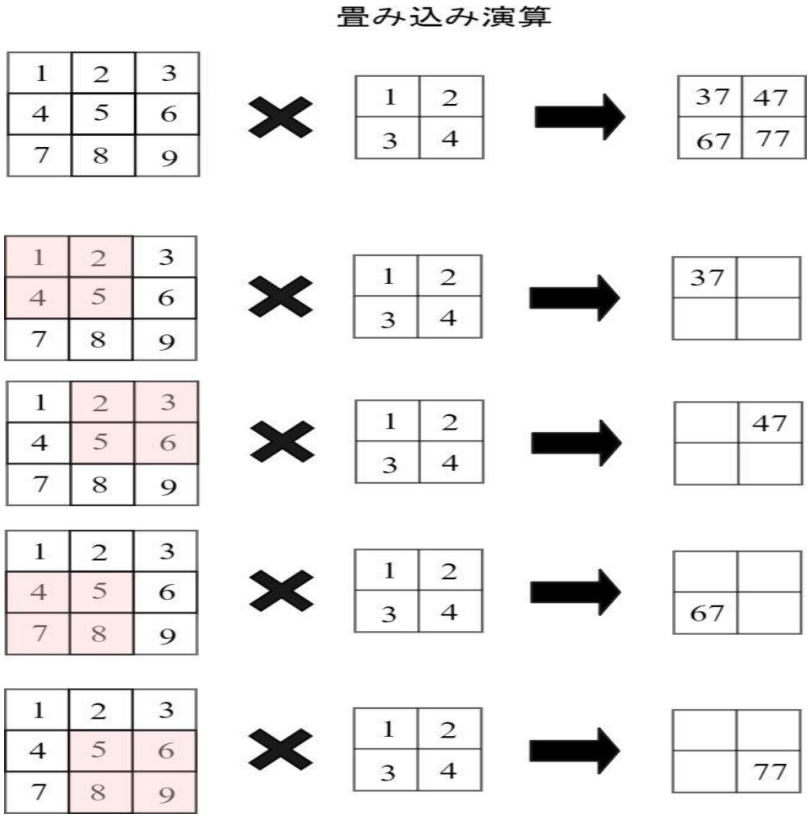

図 2.2 畳み込み演算の例(図は[2]より引用)

図のようにフィルタの対応する部分を掛け合わせ、それぞれの和をとることによって畳み込み演算を行う。この際畳み込みフィルタはストライドに沿って移動する。ストライドの価が大きくなるほど演算結果は小さくなる。畳み込み演算の結果生成される行列の大きさは次式によって求めることができる。

$$O_h = \frac{H + 2 - F_h}{S} + 1 \tag{2.1}$$

$$O_w = \frac{W + 2 - F_w}{S} + 1 \tag{2.2}$$

ここで、出力の高さをO_h、幅をO_w、フィルタサイズの高さをF_h、幅をF_w、ストライドをSとする。

2.1.3 プーリング層

プーリング層はマックスプーリング層とも呼ばれ、畳み込み層によって適用されたフィルタ内の最大を取る層である。最大を選択することによって畳み込み層によって得た特徴マップをより扱いやすい形に変形し、情報をダウンサンプリングする演算となる。画像等の 2 次元デー

タを入力とした場合のプーリング層の演算の例を図 2.3 に示す。

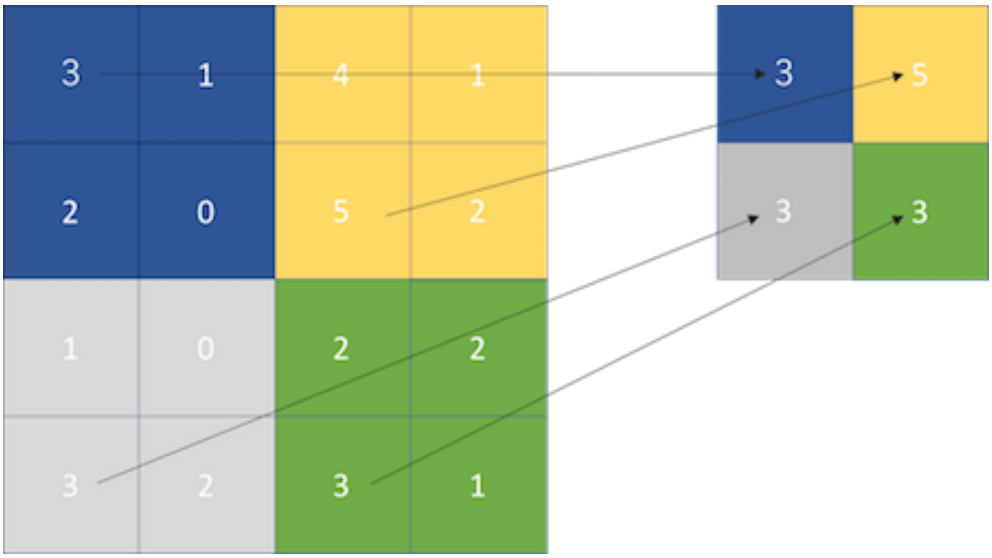

図 2.3:プーリング層の演算例(図は[3]より引用)

図に示されるように、プーリング層の演算は特徴マップのサイズが縮小される演算である。特徴マップのサイズを縮小することによって、位置変化に頑健となることや、過学習の抑制、演算コストの削減などのメリットがある。入力データが 2 次元の場合、出力の高さをO_h、幅をO_wとし、フィルタサイズの高さをF_h、幅をF_w、ストライドをSとすると、

$$O_h = \frac{H - F_h}{S} + 1 \tag{2.3}$$

$$O_w = \frac{W - F_w}{S} + 1 \tag{2.4}$$

となる。

2.1.4　全結合層

全結合層は畳み込みニューラルネットワークにおいては出力層の前に使われることが多い。畳み込み層およびプーリング層によって抽出した特徴から、予測結果に分類するための識別部としての役割を果たす。また、出力層のユニット数は分類結果のクラス数の数と一致していなければならない。出力層はそのクラスであると予測される確率を出力する。入力ユニットと出力ユニットの例を図 2.5 に示す。

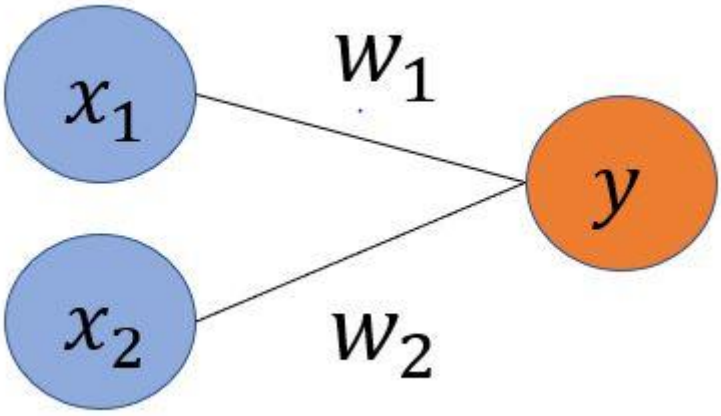

図 2.5:入力ユニットと出力ユニット

出力の価の計算は入力ユニットの価と接続の重みの内積をとり、バイアスを加算する。入力ユニットの価をx_1、x_2、重みをw_1、w_2、バイアスをbとすると、

$$y = \mathrm{f}(w_1 x_1 + w_2 x_2 + b) \tag{2.5}$$

となる。

2.1.5 活性化関数

畳み込みニューラルネットワークでは、活性化関数として、主にソフトマックス関数やReLUなどが用いられる。本項ではこれらの活性化関数に関して述べる。

2.1.5.1 ReLU

ReLU(Rectified Linear Unit)は、活性化関数の一つで、次式によって定義される。

$$\mathrm{f}(x)=\max(0,x) \tag{2.6}$$

すなわち、ReLU は入力された値が 0 以下の場合 0 を出力し、1 より大きい場合は入力をそのまま出力する活性化関数である。ReLU 関数のグラフは、図 2.6 のようになる。

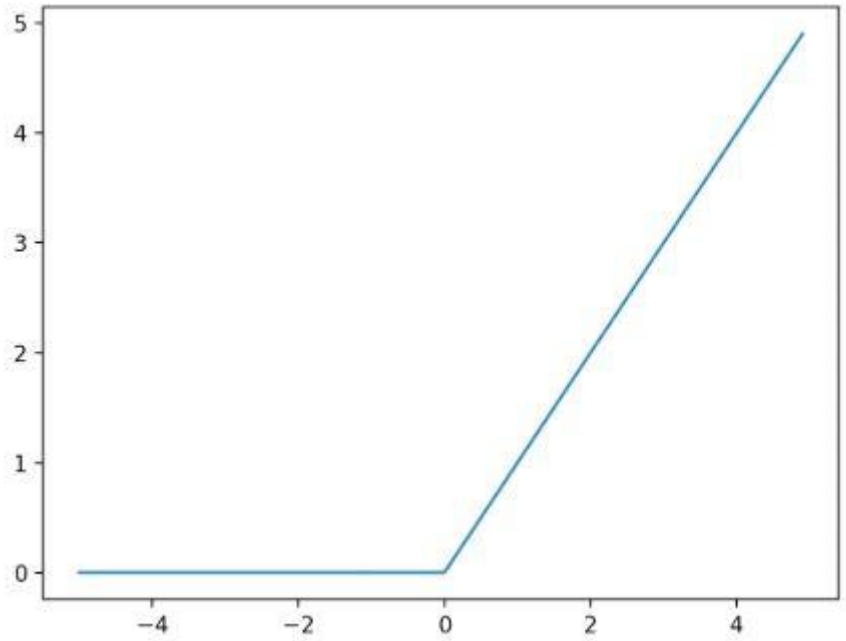

図 2.6:ReLU 関数

2.1.5.2 シグモイド関数

シグモイド関数は次式によって定義される活性化関数である。

$$f(x)=\frac{1}{1+e^{-x}} \tag{2.7}$$

ReLU 関数は主に中間層で適用されるのに対して、シグモイド関数は主に出力層で用いられる。シグモイド関数は入力が大きいほど 1 に収束し、入力が小さいほど 0 に収束する。シグモイド関数のグラフは図 2.7 のようになる。

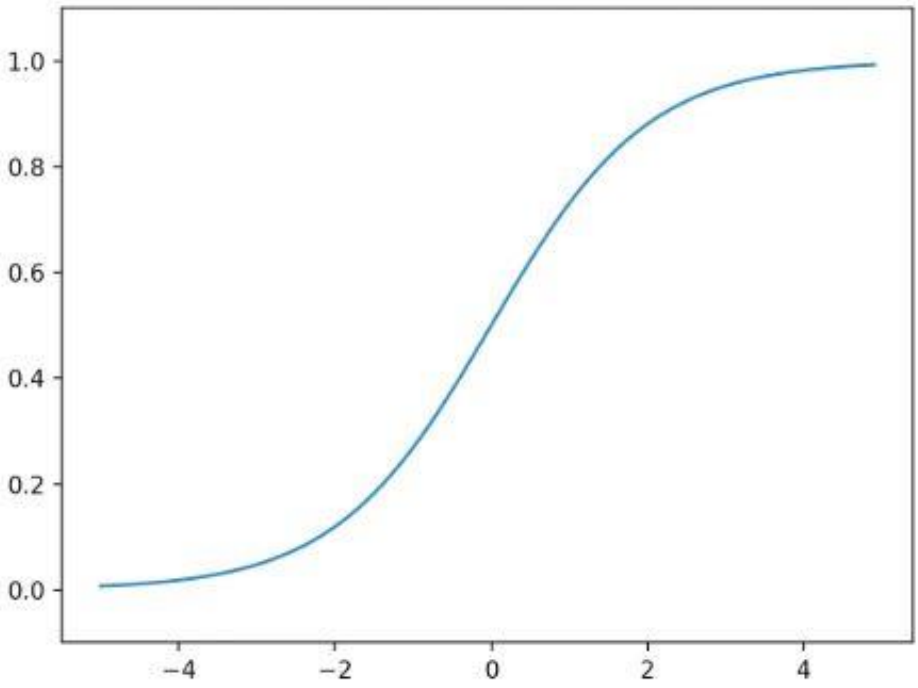

図 2.7:シグモイド関数

2.1.5.3 ソフトマックス関数

ソフトマックス関数は、シグモイド関数と同じく主に出力層にて適用される活性化関数で、次式によって定義される。

$$f_i(a)=\frac{e^{a_i}}{\sum_j^n e^{a_j}} \tag{2.8}$$

ソフトマックス関数の出力は必ず 0 から 1 であり、出力の総和は 1 となる。この性質から、出力を確率とみなすことができ、分類問題をより分かりやすくすることが可能となる。例えば、ある入力がそれぞれのクラスに分類される確率を算出する場合には、ソフトマックス関数が用いられる。ソフトマックス関数のグラフは図 2.8 のようになる。

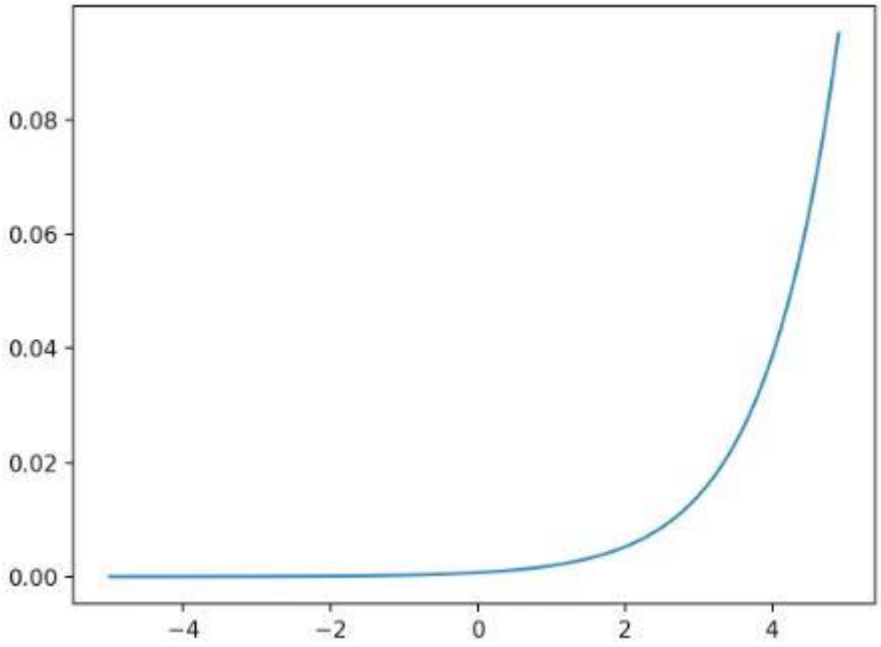

図 2.8:ソフトマックス関数

2.1.6 最適化アルゴリズム

畳み込みニューラルネットワークでは、最適化アルゴリズムとして Adam や AdaGrad などが用いられる。最適化アルゴリズムはこれまでに様々な手法が提案されており、入力変数と出力変数の予測精度が最大となるような重みを探し出す。本項ではこれらの最適化アルゴリズムについて述べる。

2.1.6.1 確率的勾配降下法(SGD)

確率的勾配降下法は(SGD)は、最急降下法をオンライン学習に改良した手法で、次式のように表すことができる。

$$g^{(t)} = \nabla E(w^{(t)}) \tag{2.9}$$

$$\Delta w^{(t)} = -\eta g^{(t)} \tag{2.10}$$

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)} \tag{2.11}$$

ここで、評価関数を E(w)、最適化するパラメータを w、η を学習率、g を評価関数の勾配とする。収束に長い時間を要することが欠点として挙げられるが、学習率が一定かつ収束結果が安定することから、最適化アルゴリズムとして用いられることがある。

2.1.6.2 Momentum SGD

SGD は収束に長い時間を要することが欠点であり、この問題を改善するために提案された手法が Momentum SGD である。一期前の勾配情報を用いることにより、収束するまでの時間を短くすることができ、安定した収束結果を得ることができる。Momentum SGD は次式によって表すことができる。

$$g^{(t)} = \nabla E(w^{(t)}) \tag{2.9}$$

$$\Delta w^{(t)} = \mu \Delta w^{(t-1)} - (1-\mu)\eta g^{(t)} \tag{2.12}$$

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)} \tag{2.11}$$

2.1.6.3 AdaGrad

AdaGrad は、確率的勾配降下法における学習率を学習の進行度にあわせて学習率を調整しながらパラメータの最適化を行うアルゴリズムであり、次式によって表すことができる。

$$g^{(t)} = \nabla E(w^{(t)}) \tag{2.9}$$

$$\Delta w^{(t)} = -\frac{\eta}{\sqrt{\sum_{s=1}^{t}(g^{(s)})^2}} g^{(t)} \tag{2.14}$$

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)} \tag{2.11}$$

2.1.6.4 Adam

Adam は畳み込みニューラルネットワークにおいて最もよく使われる最適化アルゴリズムの 1 つである。前項で述べた Momentum SGD と AdaGrad を組み合わせたアルゴリズムであり、次式のように表すことができる。

$$g^{(t)} = \nabla E(w^{(t)}) \tag{2.9}$$

$$m_t = \rho_1 m_{t-1} + (1-\rho_1)\, g^{(t)} \tag{2.15}$$

$$v_t = \rho_2 v_{t-1}(1-\rho_2)\, (g^{(s)})^2 \tag{2.16}$$

$$\widehat{m_t} = \frac{m_t}{1-\rho_1^t} \tag{2.17}$$

$$\widehat{v_t} = \frac{v_t}{1-\rho_2^t} \tag{2.18}$$

$$\Delta w = -\frac{\eta}{\sqrt{\widehat{v_t} + \varepsilon}} \widehat{m_t} \tag{2.19}$$

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)} \tag{2.11}$$

ここで、ε 、ρ_1、ρ_2、η はハイパーパラメータである。

2.1.7 損失関数

ニューラルネットワークに用いられる損失関数は、ラベルと出力の誤差を計算する関数である。ニューラルネットワークは訓練データと出力の差を小さくするように学習が行われ、損失関数はその差をどのようにして測るかを決める役割を持つ。これに関しても様々な手法が提案されており、それぞれの問題に対して適切な損失関数を選択することが必要である。

2.1.7.1 平均二乗誤差

平均二乗誤差は主に決定木やニューラルネットワークなどの回帰問題に用いられる。平均二乗誤差は外れ値に敏感であるため、外れ値が含まれるデータセットに平均二乗誤差を適用した場合、予測結果が不安定になってしまうことがある。平均二乗誤差は以下の式によって表すことができる。

$$MSE(y_i,\hat{y_i})=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y_i})^2 \tag{2.20}$$

2.1.7.2 平均絶対誤差

外れ値に敏感である平均二乗誤差に対して、平均絶対誤差は外れ値に強いため、外れ値が含まれるデータセットに対しても安定した予測結果を出力することができる。平均絶対誤差は以下の式によって表すことができる。

$$MAE(y_i,\hat{y_i})=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y_i}| \tag{2.21}$$

2.1.7.3 平均二乗対数誤差

平均二乗対数誤差は以下の式によって表すことができる。

$$MSLE(y_i,\hat{y_i})=\frac{1}{n}\sum_{i=1}^{n}\{\log(1+y_i)-\log(1+\hat{y_i})\}^2 \tag{2.21}$$

平均二乗対数誤差を用いる場合、予測結果が実数値を上回ってしまうことが多い。この傾向を利用し、例えば来客人数を予測する場合などに平均対数誤差が用いられる。

2.1.7.4 交差エントロピー誤差

交差エントロピー誤差は、主にニューラルネットワークなどにおける分類問題に多く用いられる損失関数である。交差エントロピー誤差は以下の式によって表すことができる。

$$E=-\sum_k t_k \log y_k \tag{2.21}$$

ここで、t_kは実際のクラスを 0 と 1 を用いて表し、y_kは予測確率を表す。すなわち、交差エントロピー誤差の計算は正解データt_k=1 の場合にのみ計算が行われる。図 2.9 に示すように、y=logx のグラフは、x の値が 0 に近い場合は大きな値を出力し、1 に近い場合は 0 に近い値を出力する。正解データt_kが 1 の場合、それに対応する出力y_kが 1 に近い値を出力している場合乗算結果を小さくなり、y_kが 0 に近い値を出力している場合、乗算結果は大きくなる。

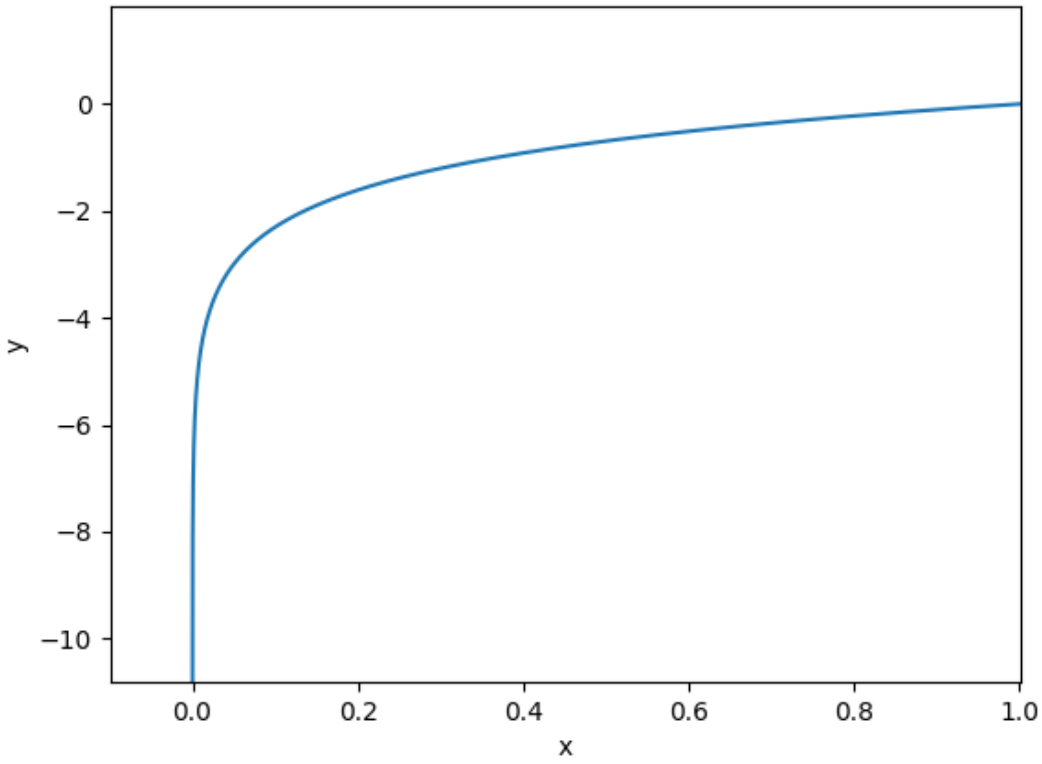

図 2.9:y=logx

2.2 音響特徴抽出法

2.2.1 メル周波数ケプストラム係数

メル周波数ケプストラム係数(MFCC)とは[5]、音声認識の分野で最も広く利用されている音響特徴量である。MFCC の特徴量抽出の手順を図 2.10 に示す。

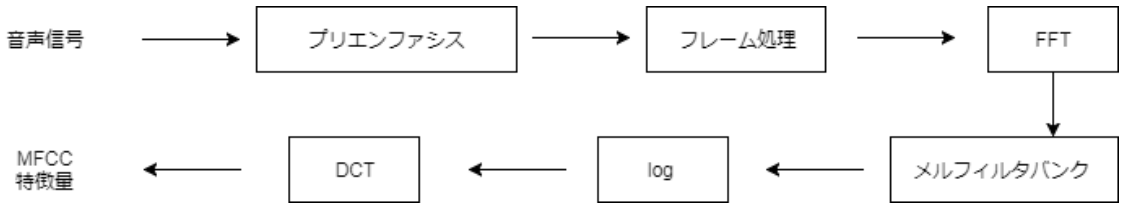

図 2.10:MFCC の特徴量抽出の手順

プリエンファシス処理は、音声信号における周波数の偏りを修正するために高周波成分を強調させる目的で行われる。高周波成分が強調された信号$y(t)$は次式によって表される。

$$y(t) = s(t) - p \cdot s(t-1) \tag{2.9}$$

ここで s(t)は時刻 t における音声波形データ、pはプリエンファシス係数で、0.97 を使うことが多い。プリエンファシス処理をしたのち、音声波形に FFT(高速フーリエ変換)を行う。その後周波数軸上に L 個の三角窓を配置し、窓幅に対応する周波数帯域の信号のパワーを m(l)とする。ここで窓幅は人間の聴覚特性にあわせて低周波ほど間隔が狭く、高周波ほど間隔を広くとる。人間の聴覚は低周波ほど細かい音の違いを聞き分けることができ、高周波ほど音の違いが分からなくなるという特性があり、この人間の聴覚特性を加味したフィルタバンクを、メルフィルタバンクという。メルフィルタバンクを可視化させた図を図 2.11 に示す。

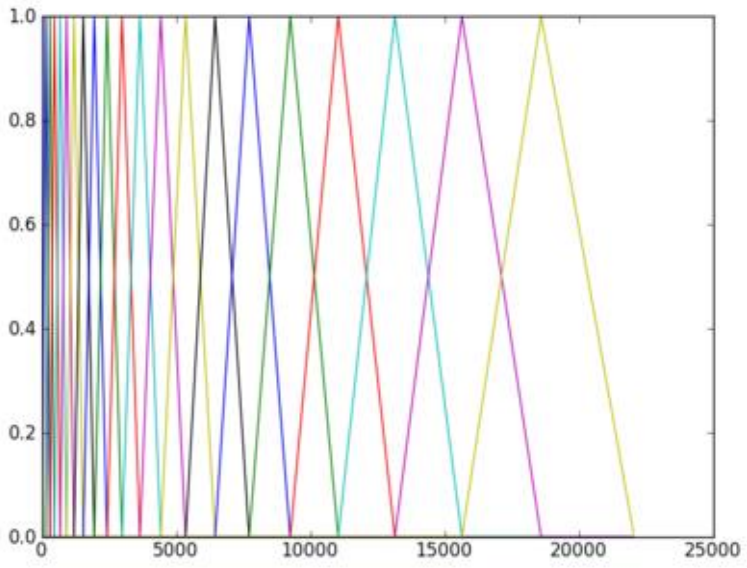

図 2.11:メルフィルタバンク

このメルフィルタバンクによって得られたＬ個の帯域におけるパワーを次式によって DCT(離

散コサイン変換)する。

$$c_k = \sqrt{\frac{2}{L}} \sum_{l=1}^{L} logm(l)\, cos\left\{\left(l - \frac{1}{2}\right)\frac{k\pi}{L}\right\} \tag{2.10}$$

ここでkはケプストラム係数の次元を表しており、低次の成分を取り出したものが MFCC 特徴量である。本研究では 13 次元までを抽出している。

2.2.2 短時間フーリエ変換(STFT)

短時間フーリエ変換(STFT)とは、音響信号の周波数分析に用いられる基本的な信号処理方法の一つである。雑音除去や音源分離など様々な用途で用いられ、機械学習における特徴抽出を行う際にも用いることができる。通常のフーリエ変換では信号の全てを変換するため、時間的な情報は全て失われてしまう。短時間フーリエ変換は時間情報を残したまま周波数領域に変換することができる。入力信号を窓関数をずらしながらかけ、時間軸方向に短時間ごとにフーリエ変換を行うことで、時間情報と周波数情報の 2 次の関数として信号を表すことができ、それを音響特徴量として用いることができる。短時間フーリエ変換は次式によって表される。ここで w は窓関数を表している。

$$STFT_{x,w}(t,\omega) = \int_{-\infty}^{\infty} x(\tau - t) e^{-j\omega\tau} \mathrm{d}\tau \tag{2.11}$$

短時間フーリエ変換には不確定性原理と呼ばれる問題点がある。不確定性原理とは、

$$\Delta x \Delta \omega \geq \frac{1}{2} \tag{2.12}$$

の関係が時間と周波数との間に成り立つことである。窓関数の窓の大きさによって周波数分解能か時間分解能を良くするかのトレードオフが存在する。窓関数の窓が広い場合、周波数分解能が良くなり、時間分解能は悪くなる。これに対して窓関数の窓が狭い場合は時間分解能はいいが周波数分解能が悪くなる。

2.3 音声認識システム

音声認識をパターン認識によって行うシステムは、一般に図 2.12 に示すようなモジュール構成で実現される。[6]

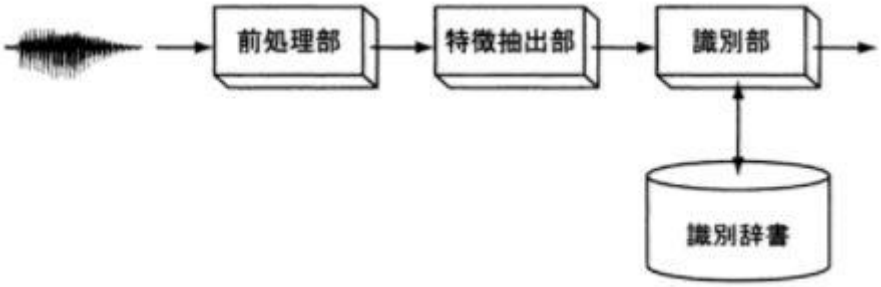

図 2.12:音声認識システムの構成

マイクなどの入力装置から入力されたアナログ信号を前処理によってコンピュータ内部で処理可能なディジタル信号に変換する。このディジタル化したデータを入力データとし、特徴抽出を行う。特徴抽出は一般にベクトルの形式で抽出される。この特徴ベクトルを識別辞書中に存在する各クラスの訓練データと比較し識別結果が決定される。

2.4 2 値分類器

機械学習において特に正例と負例を用意し、テストデータが正例か負例かを判別するものを 2 値分類器という。多値分類器においてどのラベルにも属さない未知データに対して多値分類器は設定されたラベルのどれかであるという識別結果を出力するが、これに対して 2 値分類器の場合は、そのデータが正例であるか違うかを知ることができる。(図 2.13,2.14)

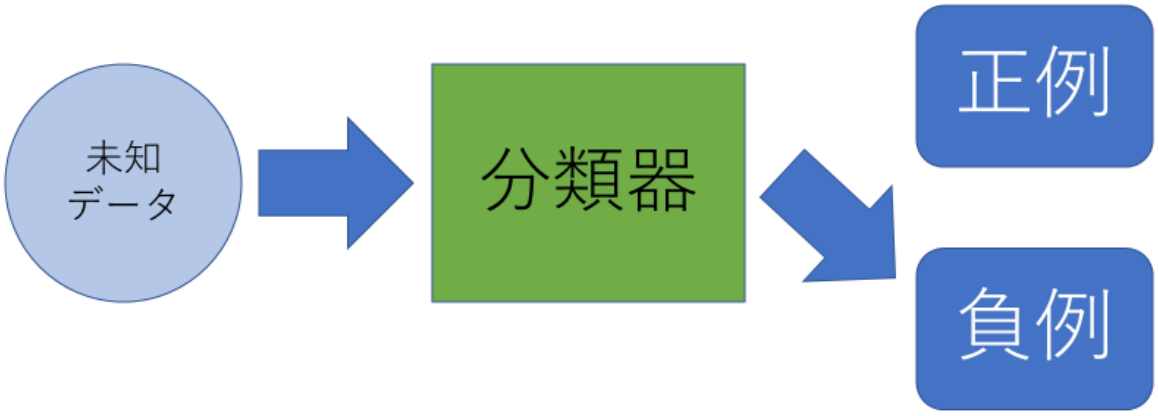

図 2.13:2 値分類器

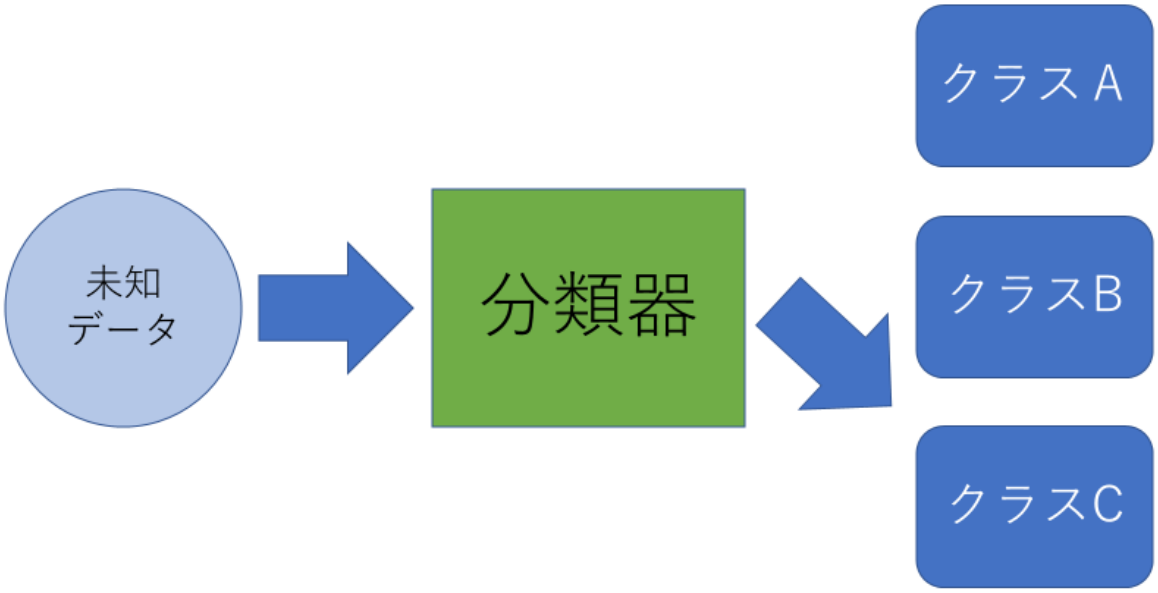

図 2.14:多値分類器

第3章 多値分類器による野鳥の鳴き声識別

3.1 概要

一般に教師あり機械学習を行う際には、特徴抽出法を用いた特徴抽出と、人手によるラベル付けが必須である。しかし、それらを行う際にはコストがかかり、そのコストを削減することは重要な課題である。従来の機械学習の手法は MFCC 等の特徴抽出法による特徴抽出によって得た特徴量を用い、人手による鳴き声のラベル付けを行うことによって機械学習を行う。

この従来手法の改善手法として、人手による鳴き声の一泣き分の定義を行わずに鳴き声の音声データを細かい感覚で分割し、従来の特徴抽出法ではなく 1 次元畳み込みニューラルネットワークを用いて音響データから直接学習を行う手法を検討した。この手法を用いることにより、人手による鳴き声の定義と、特徴抽出を行わずに機械学習を行うことができる。

一般に、畳み込みニューラルネットワークは画像等の 2 次元データに使用されることが多く、画像の局所的な特徴を学習することができることで知られている。特に 1 次元信号に畳み込みニューラルネットワークを使用する先行研究では、1 次元信号をスペクトログラム等の画像(2 次元データ)に変換し、1 次元信号を画像のように扱うことで機械学習を行う手法が提案されている[7]。しかし、ニューラルネットワークへと入力するために画像へと変換するためには演算コストがかかり、1 次元信号をスペクトログラムに変換する過程で位相情報が損なわれるため、認識精度に影響が出る恐れがある[8]。そこで、1 次元信号をそのままニューラルネットワークの入力として使用できる 1 次元畳み込みニューラルネットワークを使用した。1 次元畳み込みニューラルネットワークは 1 次元フィルタを用いて、時間軸方向へとスライドさせるような演算となり、画像等の多次元の場合と同じようにネットワークを構築することができる。また、通常の畳み込みニューラルネットワークと同じように、次元削減のためプーリング層を用いる。1 次元畳み込みニューラルネットワークは一次元信号の特徴的な周波数帯を強調するような特徴量を学習により獲得できるため、従来のような演算コストのかかる特徴抽出法を使用する必要がなくなり、演算コストが削減されることが期待される。

従来の特徴抽出法を用いず、且つ野鳥の鳴き声データを分割することによって切り出しのコストを削減する方法によって高い認識精度を出すことができるかどうかを確かめるため、①従来の特徴抽出法を用いニューラルネットワークで学習する手法と②人手による鳴き声を定義したデータと細切れ録音データによる学習方法での機械学習の成績をそれぞれ比較する実験を行った。具体的には①切り出しデータを MFCC で特徴抽出を行い学習、②切り出しデータを直接 CNN で学習③細切れデータを MFCC で特徴抽出を行い学習、④細切れデータを直接 CNN で学習の 4 通り行った。実験の処理の流れの一例を図 3.1 に示す。

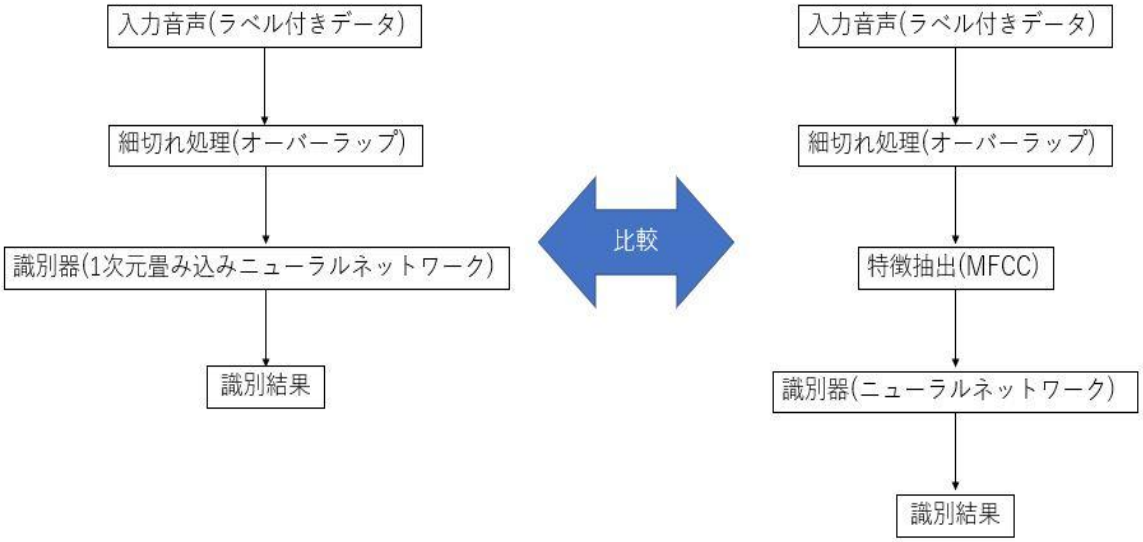

図 3.1:実験処理の流れの一例

3.2 実験

3.2.1 データセット

実験にはニュージーランドに生息する 3 種類の野鳥の鳴き声と、録音データに混在する環境雑音及び飛行機による雑音を加えた 5 種類のデータを使用した。学習成績の比較は雑音情報を除いた 3 値分類による比較と、雑音情報を含めた 5 値分類による比較の両方について行った。データは全て wave ファイルで保存されており、抽出した特徴量は全て csv ファイルで保存している。MFCC は入力音声のサンプリングレート 22050 で 512 フレームごとに抽出し、平均を取って 13 次元の特徴量を得た。畳み込みニューラルネットワークに入力するデータは、WAV データから RAW データに変換し、1 次元のデータ配列を csv ファイルに保存した。音声データを分割する場合は、分割間隔を 0.1 秒、オーバーラップ率 0.5 で分割したデータを学習に用い、分割しない場合は 0.7 秒分のデータを学習データに用い、0.7 秒を超過する鳴き声のデータに関しては切り捨てを行っている。学習の際にはデータセットの 10%を検証データに、20%をテストデータに用いた。

3.2.2 学習条件

データセットの音声データを分割する場合と、そうでない場合では入力データの次元数に非常に大きな差が出るため、畳み込み層のフィルタサイズはそれぞれ個別に調整を行った。音声データの分割をしない場合に用いる 1 次元畳み込みニューラルネットワークのネットワーク構成を表 3.1 に示す。

表 3.1:分割なしの場合に用いる 1 次元畳み込みニューラルネットワークの詳細

ネットワーク構成	
入力層	サイズ: (22400,1)
畳み込み層 1	フィルタサイズ:(128,64) ストライド:256 ドロップアウト率 0.2 活性化関数:relu
プーリング層 1	プールサイズ:2
畳み込み層 2	フィルタサイズ:(16,8) ドロップアウト率:0.2 活性化関数:relu
プーリング層 2	プールサイズ:2
畳み込み層 3	フィルタサイズ:(16,8) ドロップアウト率:0.2 活性化関数:relu
プーリング層 3	プールサイズ:2
全結合層	活性化関数:ソフトマックス
出力層	サイズ:(None,3 もしくは 5)

データセットの音声データを 0.1 秒刻みに分割する場合に用いる 1 次元ニューラルネットワークのネットワーク構成を表 3.2 に示す。

表 3.2:分割なしの場合に用いる 1 次元畳み込みニューラルネットワークの詳細

ネットワーク構成	
入力層	サイズ: (3200,1)
畳み込み層 1	フィルタサイズ:(128,64) ストライド:256 ドロップアウト率 0.2 活性化関数:relu
プーリング層 1	プールサイズ:2
畳み込み層 2	フィルタサイズ:(16,8) ドロップアウト率:0.2 活性化関数:relu
プーリング層 2	プールサイズ:2
畳み込み層 3	フィルタサイズ:(16,8) ドロップアウト率:0.2 活性化関数:relu
プーリング層 3	プールサイズ:2
全結合層	活性化関数:ソフトマックス
出力層	サイズ:(3 もしくは 5)

比較対象として特徴抽出法を用いて学習を行う場合のニューラルネットワークのネットワーク構成を表 3.3 に示す。

表 3.3:特徴抽出法を用いて学習するニューラルネットワークの詳細

ネットワーク構成	
入力層	サイズ: (13,1)
全結合層	活性化関数:ソフトマックス
出力層	サイズ:(3 もしくは 5)

以上のネットワーク構成でバッチ数 600、損失関数多クラス交差エントロピー、最適化アルゴリズム adam として学習を行った。また、入力の次元数が多い場合は少ないエポック数では学習が収束しない恐れがあるため、音声データを分割する場合は 50 エポック、分割しない場合は 200 エポックとして学習を行った。

3.2.3 実験環境

本実験にて使用したライブラリ等の実験環境を以下に示す。実装は全て Python によって行い、機械学習の実装に Keras、MFCC 特徴量の抽出に Librosa、グラフ描画に Matplotlib、配列処理に Numpy、WAV ファイルから RAW データへの変換に Pydub を用いた。

Ubuntu 18.04 LTS
Python 3.6.5
Keras 2.2.2
Librosa 0.6.2
Matplotlib 2.2.2
Numpy 1.14.3
Pydub 0.22.1

3.3 実験結果

MFCC によって特徴抽出を行い、分割間隔 0.1 秒で 3 値分類を行った学習成績を図 3.1 に、ニューラルネットワークによって特徴抽出を行い、分割間隔 0.1 秒で 3 値分類を行った学習成績を図 3.2 に示す。これを見ると、早期に 1.00 に近づいており、いずれの場合も非常に高い識別率となっていることがわかる。

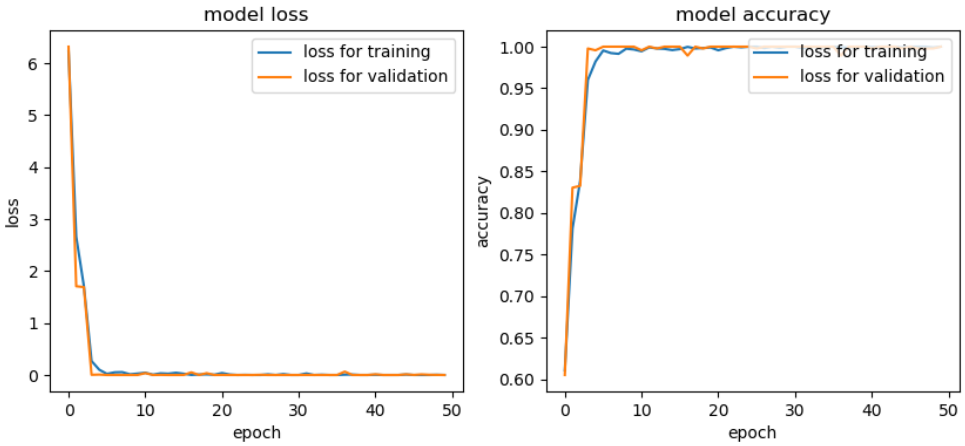

図 3.1:MFCC、3 値分類、分割間隔 0.1 秒

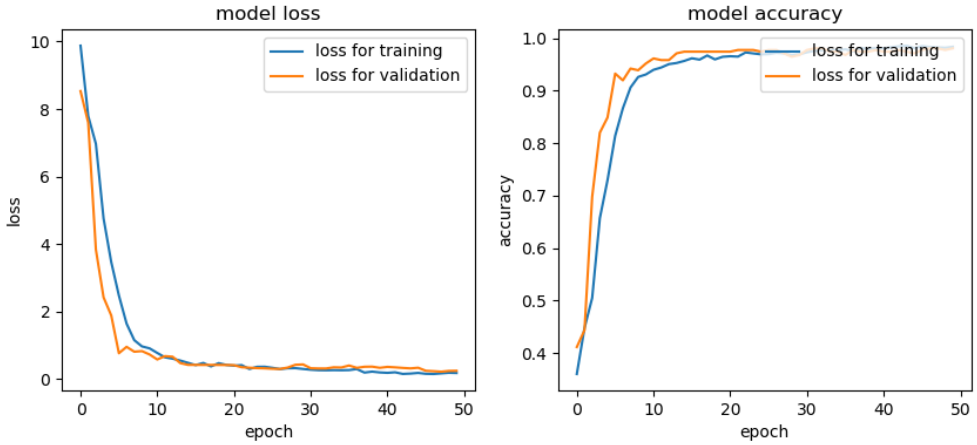

図 3.2:ニューラルネットワーク、3 値分類、分割間隔 0.1 秒

MFCC によって特徴抽出を行い、分割なしで 3 値分類を行った学習成績を図 3.3 に、ニューラルネットワークによって特徴抽出を行い、分割なしで 3 値分類を行った学習成績を図 3.4 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

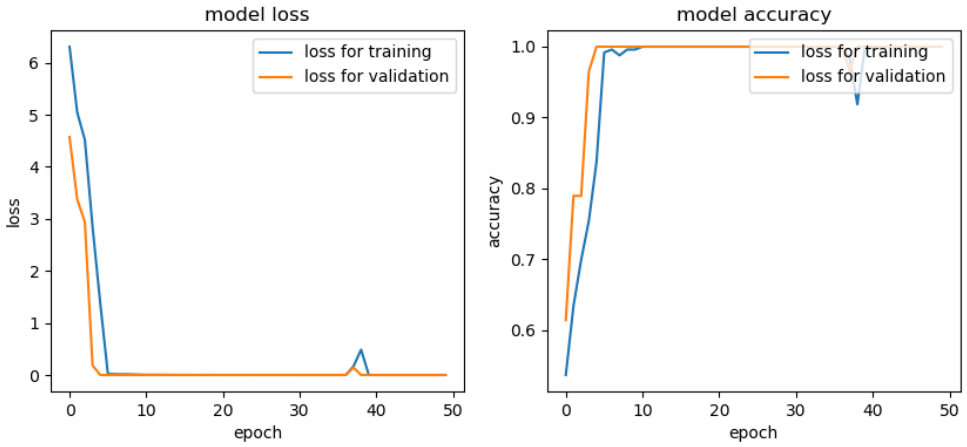

図 3.3:MFCC、3 値分類、分割なし

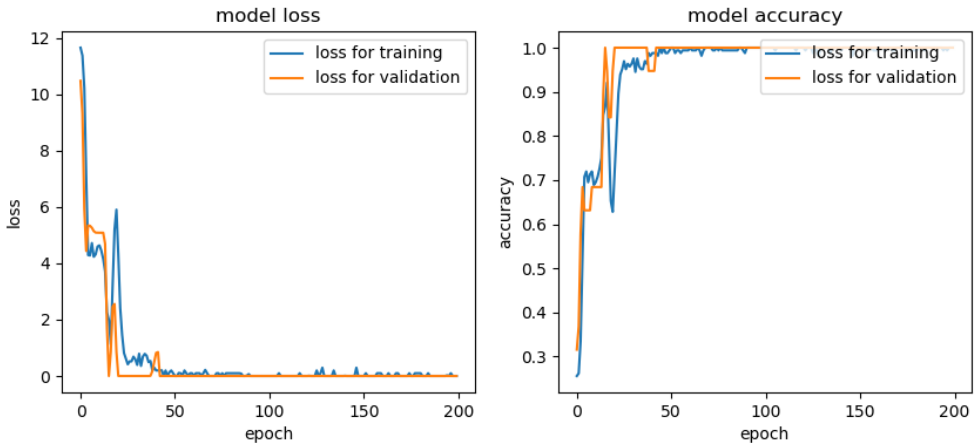

図 3.4:ニューラルネットワーク、3 値分類、分割なし

MFCC によって特徴抽出を行い、分割間隔 0.1 秒で 5 値分類を行った学習成績を図 3.5 に、を図 3.6 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

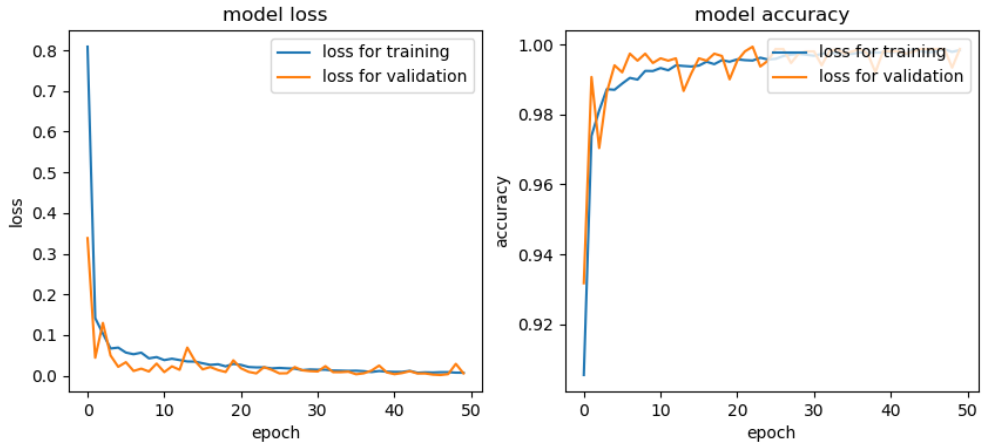

図 3.5:MFCC、5 値分類、分割間隔 0.1 秒

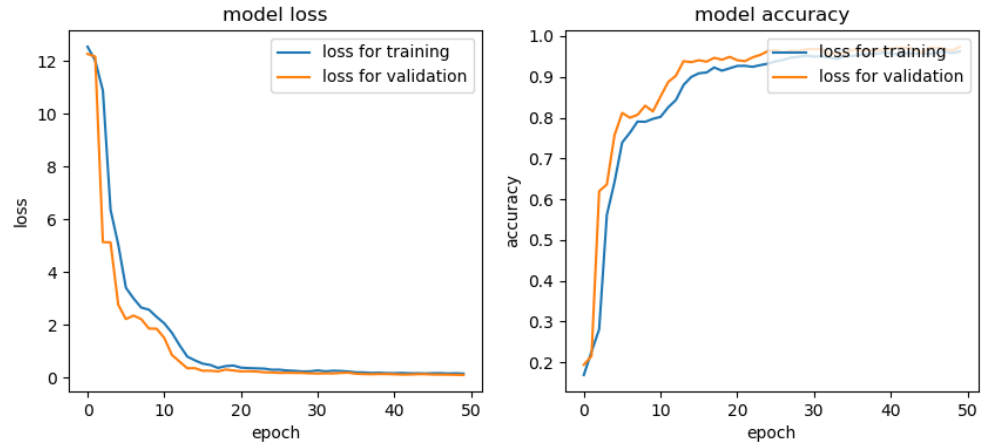

図 3.6:ニューラルネットワーク、5 値分類、分割間隔 0.1 秒

MFCC によって特徴抽出を行い、分割なしで 5 値分類を行った学習成績を図 3.7 に、ニューラルネットワークによって特徴抽出を行い、分割なしで 5 値分類を行った学習成績を図 3.8 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

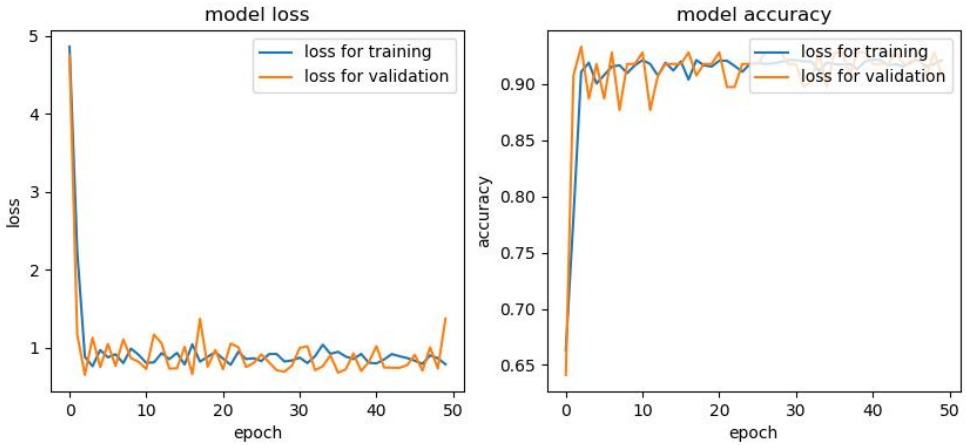

図 3.7:MFCC、5 値分類、分割なし

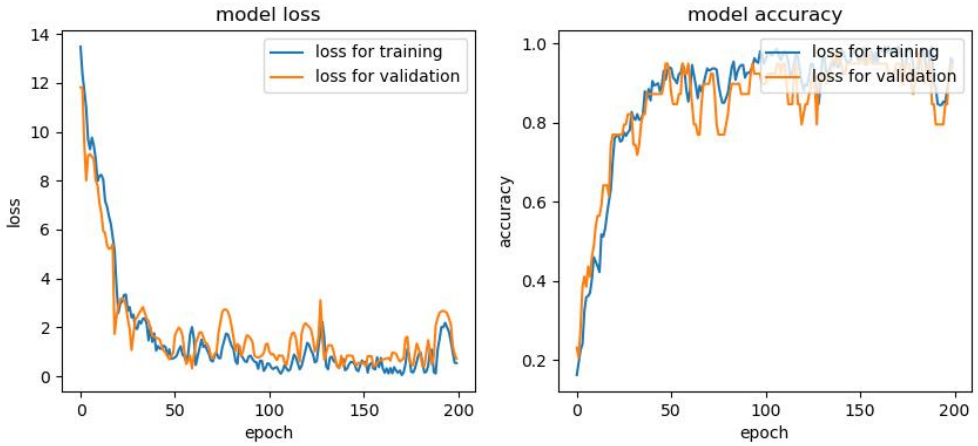

図 3.8:ニューラルネットワーク、5 値分類、分割なし

3.4 考察

図より、全てのパターンにおいて非常に高い精度で識別が行えていることがわかる。これによって、ニューラルネットワークによる特徴抽出し、音声データを細切れにする手法が従来手法である鳴き声を定義して音響特徴抽出法を用いて機械学習を行う手法と同程度の性能を示すということが分かった。野鳥の鳴き声のみで構成される 3 値分類だけでなく雑音も含まれる 5 値分類においても高い精度で認識することができていることから、音声データを細切れにする手法によってニューラルネットワークがあらゆる音の局所的な特徴を学習することができていると考えることができる。

今後の課題としては、さらに高い性能を示すことができるネットワーク構成を模索することや、ネットワーク構成と最適化アルゴリズム、エポック数やバッチ数の組み合わせなどを模索していくことが挙げられる。

第4章　複数の 2 値分類器による野鳥の鳴き声識別

4.1　概要

本研究において 2 つ目に着目する問題は、現在の教師あり機械学習の枠組みでは未知のデータを未知であると判別することが難しいという点があげられるである。本研究ではこの問題に対して、複数の 2 値分類器を並列に稼働させることによって長時間録音データに対応できるような手法を検討した。

音データに既知データかそうでないかを識別させる 2 値分類器を適用する。全ての既知のデータに対して 2 値分類器を並列に適用していくことで、全く未知のデータを検出することができる。検出することのできた未知データには専門家によるラベル付けを行い、新たな 2 値分類器を生成する。このプロセスを繰り返すことによって全ての既知データに対して正しく識別が可能で且つ、未知データが検出された場合には未知データが含まれると判定することができる音声認識システムが完成することが期待される。音声認識システムの構想を図 4.1 に示す。

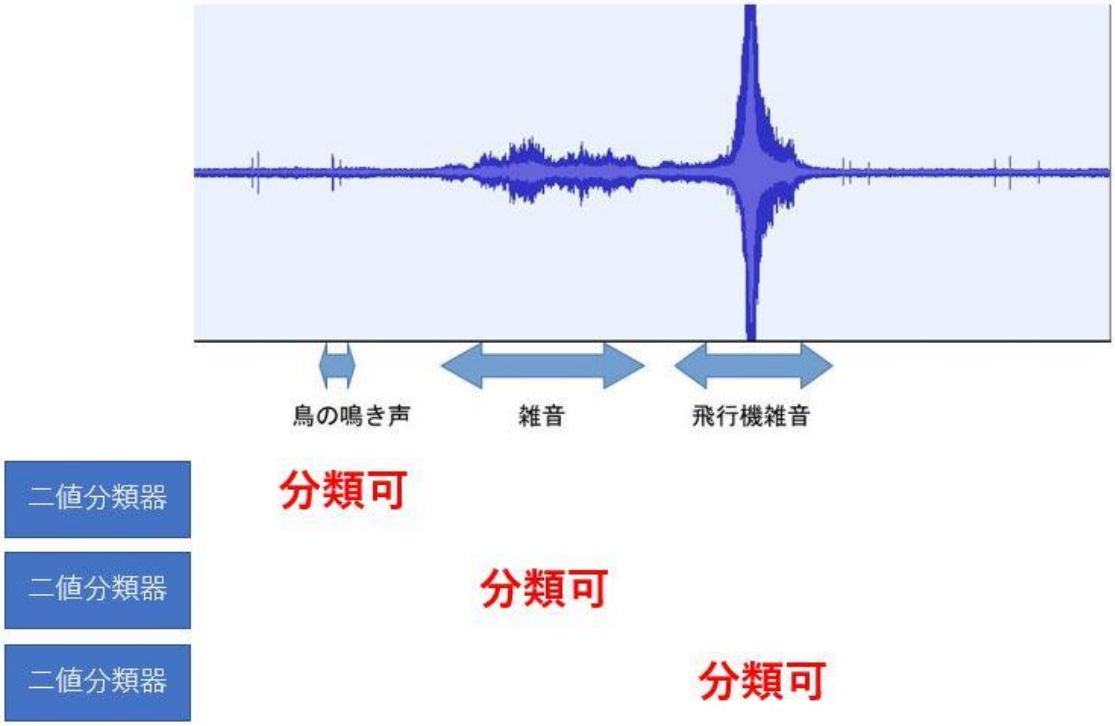

図 4.1:音声認識システムの構想

録音データに対して高い精度で 2 値分類を行うことが可能かどうかを確かめるため、録音データに混在する 3 種類の野鳥の鳴き声、及び飛行機雑音、及び環境雑音の 5 種類が含まれるデータセットを用意し、それぞれを正例にした 2 値分類器を生成し、その学習成績を確認する実験を行った。実験は 3 章にて確認したニューラルネットワークによってデータから直接学習し、音声を細切れにする機械学習手法によって行った。

4.2 実験

4.2.1 データセット

長時間録音データに混在するデータによる機械学習の成績を確かめるため、長時間録音データに混在する 3 種類の野鳥の鳴き声、飛行機雑音、環境雑音の 5 種類が含まれるデータセットを用意し､それぞれを正例にした 2 値分類器を生成し､その学習成績を確認する実験を行った。3 章にて、音声データを細切れにし、ニューラルネットワークによってデータから直接学習する手法は有効であることが確認されたため、その手法を用いて実験を行った。音声を分割する間隔は、0.1 秒と 0.3 秒それぞれについて実験を行った。

4.2.2 学習条件

本実験に使用した 1 次元畳み込みニューラルネットワークの詳細を表 4.1 に示す。畳み込みニューラルネットワークに入力するデータは、WAV データから RAW データに変換し、1 次元のデータ配列を csv ファイルに保存した。学習の際にはデータセットの 10%を検証データに、20%をテストデータに用いている。

表 4.1:1 次元畳み込みニューラルネットワークの詳細

ネットワーク構成	
入力層	サイズ: (3200,1)
畳み込み層 1	フィルタサイズ:(128,64) ストライド:256 ドロップアウト率 0.2 活性化関数:relu
プーリング層 1	プールサイズ:2
畳み込み層 2	フィルタサイズ:(16,8) ドロップアウト率:0.2 活性化関数:relu
プーリング層 2	プールサイズ:2
畳み込み層 3	フィルタサイズ:(16,8) ドロップアウト率:0.2 活性化関数:relu
プーリング層 3	プールサイズ:2
全結合層	活性化関数:ソフトマックス
出力層	サイズ:(2)

表に示したネットワーク構成でバッチ数 600、エポック数 50 エポック、損失関数多クラス交差エントロピー、最適化アルゴリズム adam によって学習を行った。

4.2.3 実験環境

本実験にて使用したライブラリ等の実験環境を以下に示す。実装は全て Python によって行い、機械学習の実装に Keras、MFCC 特徴量の抽出に Librosa、グラフ描画に Matplotlib、配列処理に Numpy、WAV ファイルから RAW データへの変換に Pydub を用いた。

Ubuntu 18.04 LTS
Python 3.6.5
Keras 2.2.2
Librosa 0.6.2
Matplotlib 2.2.2
Numpy 1.14.3
Pydub 0.22.1

4.3 実験結果

正例データをミヤマオウムにし、音声データの分割間隔 0.1 秒での学習成績を図 4.2 に、正例データをミヤマオウムにし、音声データの分割間隔 0.3 秒での学習成績を図 4.3 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

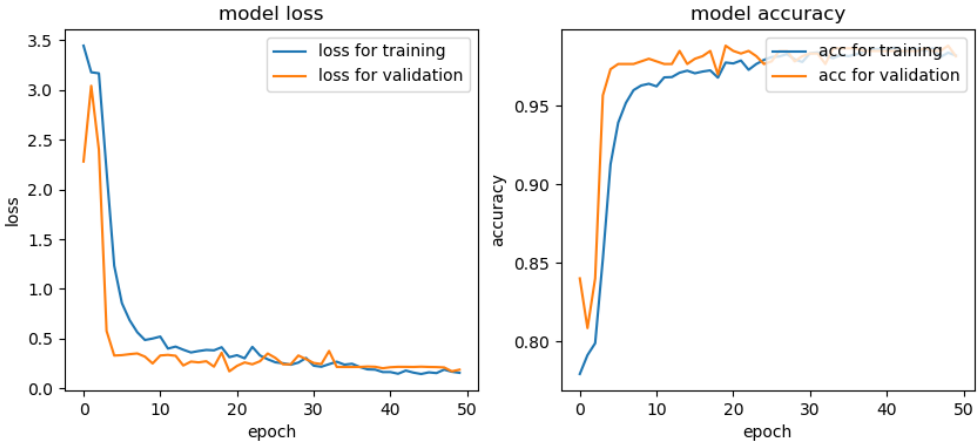

図 4.2:正例ミヤマオウム,分割間隔 0.1 秒

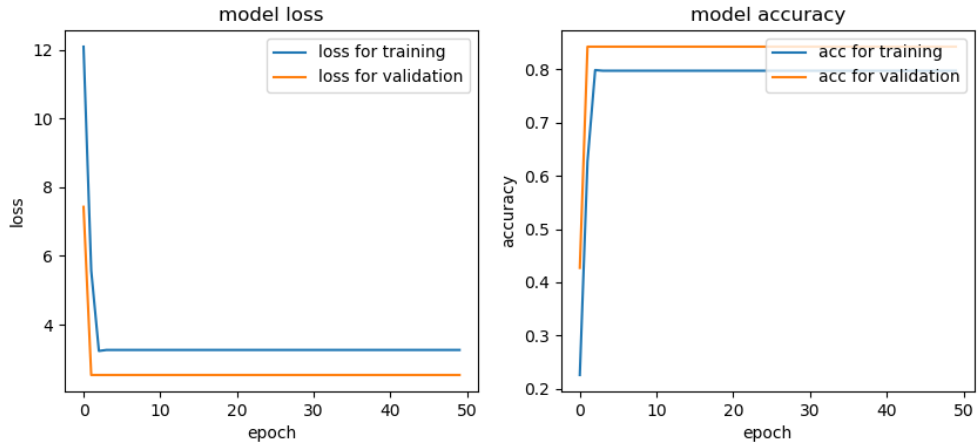

図 4.3: 正例ミヤマオウム,分割間隔 0.3 秒

正例データをキジカッコウにし、音声データの分割間隔 0.1 秒での学習成績を図 4.4 に、正例データをキジカッコウにし、音声データの分割間隔 0.3 秒での学習成績を図 4.5 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

図 4.4:正例キジカッコウ、分割間隔 0.1 秒

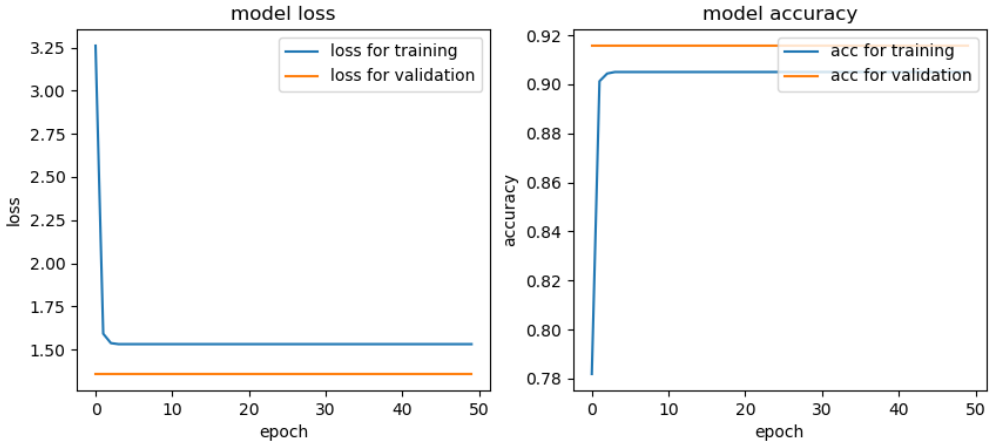

図 4.5:正例キジカッコウ、分割間隔 0.3 秒

正例データをニュージーランドアオバズクにし、音声データの分割間隔 0.1 秒での学習成績を図 4.6 に、正例データをニュージーランドアオバズクにし、音声データの分割間隔 0.3 秒での学習成績を図 4.7 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

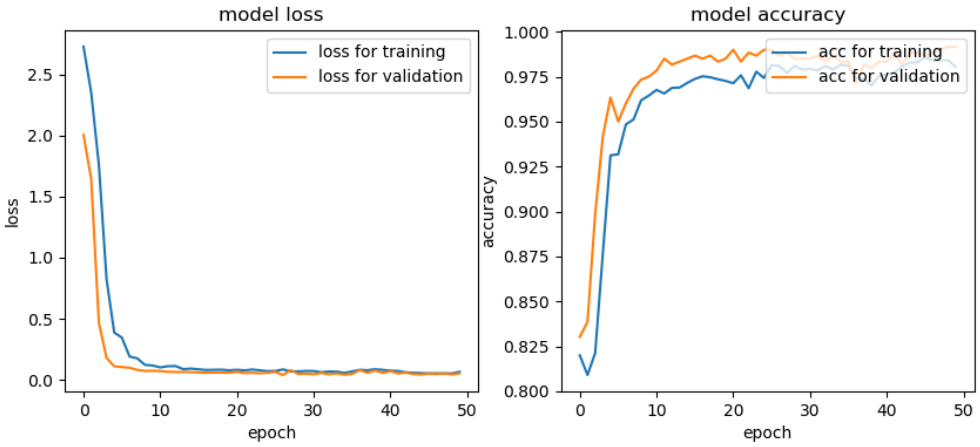

図 4.6:正例ニュージーランドアオバズク、分割間隔 0.1 秒

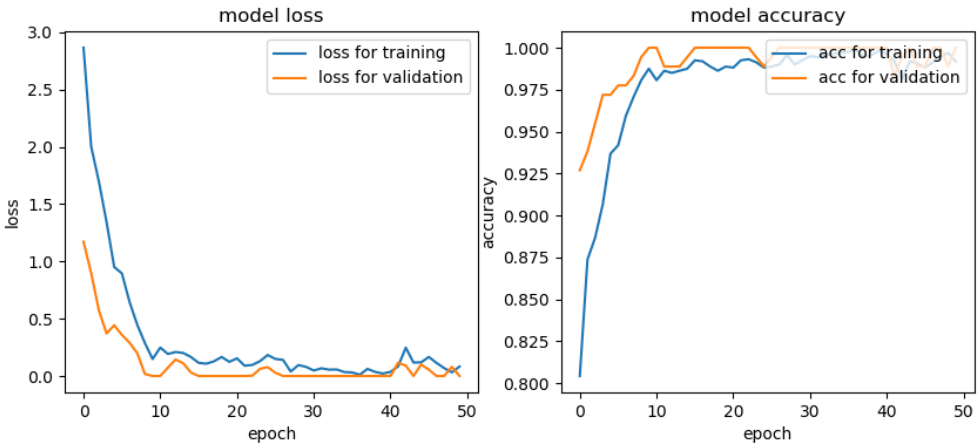

図 4.7:正例ニュージーランドアオバズク、分割間隔 0.3 秒

正例データを環境雑音にし、音声データの分割間隔 0.1 秒での学習成績を図 4.8 に、正例データを環境雑音にし、音声データの分割間隔 0.3 秒での学習成績を図 4.9 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

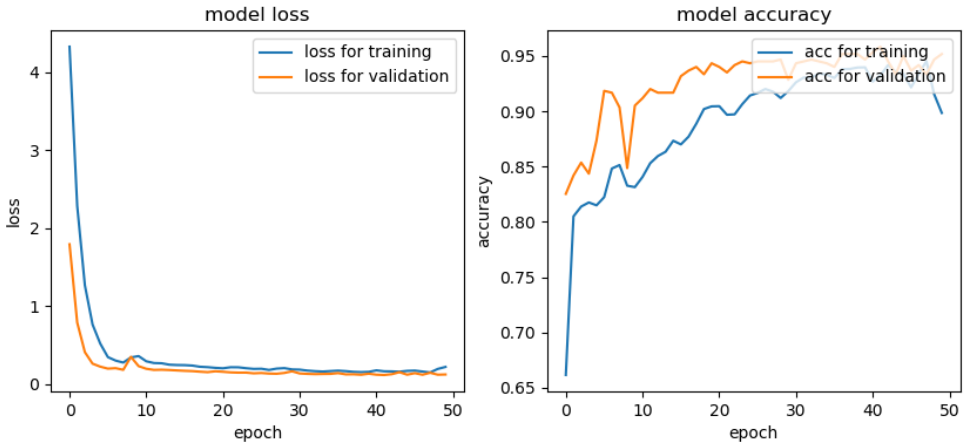

図 4.8:正例環境雑音、分割間隔 0.1 秒

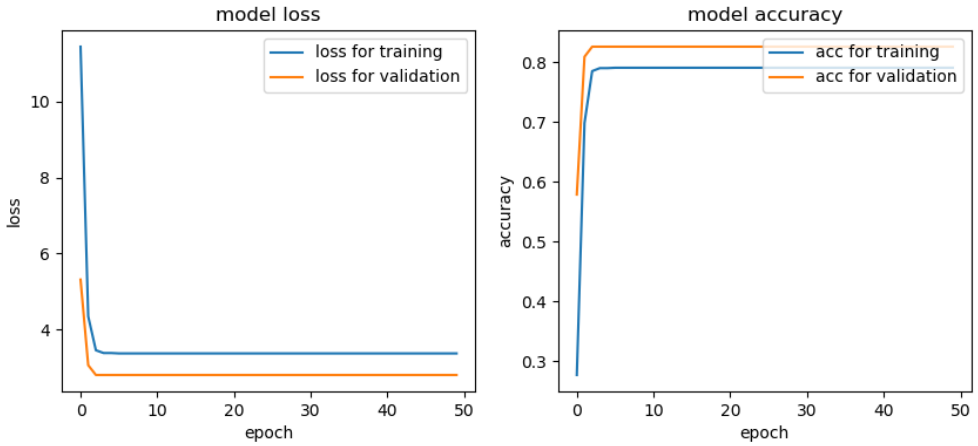

図 4.9:正例環境雑音、分割間隔 0.3 秒

正例データを飛行機雑音にし、音声データの分割間隔 0.1 秒での学習成績を図 4.10 に、正例データを飛行機雑音にし、音声データの分割間隔 0.3 秒での学習成績を図 4.11 に示す。これを見ると、いずれの場合も早期に 1.00 に近づいており、非常に高い識別率となっていることがわかる。

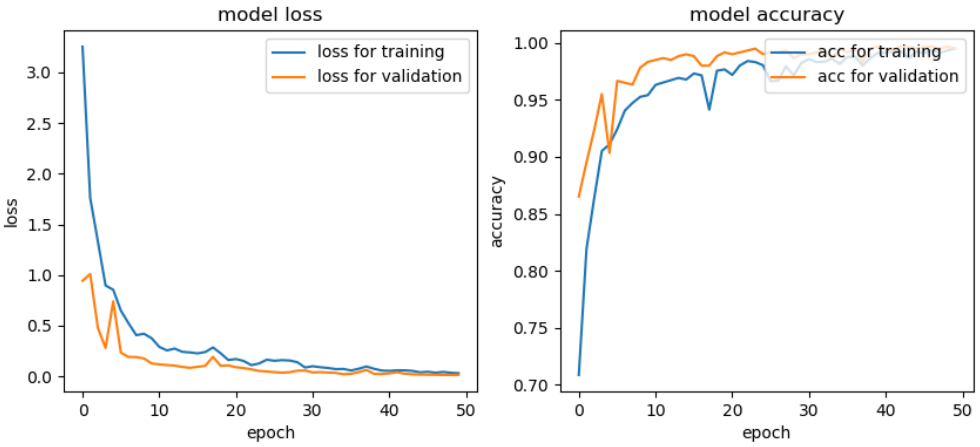

図 4.10:正例飛行機雑音、分割間隔 0.1 秒

図 4.11:正例環境雑音、分割間隔 0.3 秒

4.4 考察

長時間録音データに含まれる既知データを正例とした場合の 2 値分類の性能を確かめる実験により、ほぼ全てのパターンにおいて 95%以上の高い精度で識別が行えていることが確認できる。データの分割間隔については、0.1 秒と 0.3 秒について比較を行ったが図より 0.1 秒で分割した場合の方が学習成績が良いことがわかる。これは、分割間隔の短い場合その分多くのデータを学習データとして用いることになり、ニューラルネットワークは音声の局所的な特徴を学習し、総合的に学習成績が良くなるためであると考えられる。

この実験によって、現在の機械学習の枠組みでは困難である未知データの検出ができるという可能性が示された。

第5章 おわりに

本研究では、①特徴抽出に畳み込みニューラルネットワークを利用した機械学習②対象音声を切り出す必要がない音声認識システム③未知の野生生物の検出の 3 点に関しての検討を行った。①と②に関しては、従来手法の改善手法として、時系列情報が失われる代わりに人手による鳴き声の定義と、特徴抽出を行わずに学習を行う手法を検討し、これが従来手法と同程度の性能が示せるということが分かった。③に関しては、複数の 2 値分類器を並列に動作させることによって未知データを検出できるような新たな機械学習の枠組みの提案を行い、実験によって現在の機械学習では困難な未知データを含めての判別ができるという可能性が示された。

今後の課題としては、さらに高い性能を示すことができるネットワーク構成を模索することや、ネットワーク構成と最適化アルゴリズム、エポック数やバッチ数の組み合わせなどを模索していくことが挙げられる。

謝辞

本論文の作成にあたり、多くの助言、指導をしてくださった三好力教授に心からお礼申し上げます。また、議論に協力してくださった三好研究室の皆様や学友の皆様に心から感謝致します。

本論文には三好力教授が龍谷大学 2016 年度国外研究員の助成を受けて行った指導・研究の一部が含まれます。

参考文献

[1]”定番の Convolutional Neural Network をゼロから理解する”
https://deepage.net/deep_learning/2016/11/07/convolutional_neural_network.html#convolutional-neural-network%E3%81%AE%E7%89%B9%E5%BE%B4
[2]”畳み込みニューラルネットワークの基礎”https://screwandsilver.com/cnn_convolutional_net/
[3]“畳み込みニューラルネットワーク(CNN)”https://www.renom.jp/ja/notebooks/tutorial/basic_algorithm/convolutional_neural_network/notebook.html
[4]“畳み込みニューラルネットワーク_CNN(Vol.16)”https://products.sint.co.jp/aisia/blog/vol1-16
[5]”メル周波数ケプストラム係数(MFCC)”,
http://aidiary.hatenablog.com/entry/20120225/1330179868,2018/01/08
[6]荒木 雅弘, フリーソフトでつくる音声認識システム, 森北出版株式会社, 2007
[7]宮谷大輝,中山英樹,”スペクトログラム画像を用いた楽曲印象分類による時間及び周波数情報と印象の関係分析手法の提案”,情報処理学会研究報告,EC,エンタテイメントコンピューティング 2015-EC-35(24),pp1-5(2015)
[8]茂木貴弘,中澤友哉,田原哲也,”1dCNN-LSTM による調節弁内部の異常検知”,人工知能学会全国大会論文集 JSAI2018,3Pin144-3Pin144(2018)

学会発表履歴

[1]藤岡優也,三好力,"機械学習における特徴量類似性と認識精度に関する検討," FIT2018(第17回情報科学技術フォーラム) 論文集, 福岡工業大学, 福岡県, F-038 (2018-09)