Vol.2018-MUS-119 No.2
Vol.2018-SLP-122 No.2
2018/6/16

音楽情報処理のための深層学習

阪上 大地[1,a)]

概要：本稿では音楽情報処理に深層学習を応用する方法を解説する．本稿は二部構成で，ニューラルネットの一般的な設計と学習方法を解説した後，音楽情報処理を含めた各種メディア処理への応用例を紹介する．前半ではパラメータの初期化方法など常識となってしまっている基本的な事柄，Dropout などのよく使われるテクニック，最先端でまだ整理が進んでいない研究成果を順番に解説する．後半では WaveNet や Encoder-Decoder モデルなど各分野の代表的な応用例を紹介し，和音認識・ビートトラッキングなどに深層学習を応用した研究を紹介する．

1. はじめに

音楽情報処理 [1], [2], [3] は計算機によって音楽の内容を理解し，作曲・演奏・鑑賞などの音楽体験を豊かにするための研究分野である[*1]．人々の音楽とのさまざまなインタラクションを反映し，本分野にもさまざまなアプリケーションがある．代表例は，旋律への和声付け [4]，和音やビートの認識 [5], [6]，音響信号の加工 [7]，歌声情報処理 [8], [9]，能動的音楽鑑賞 [10] などである．このようなタスクの幅広さに加え，音楽がオーディオ・楽譜・歌詞などの様々なフォーマットで表現されることも本分野の挑戦的性格につながっている [1]．

こうした複合的なデータやタスクに立ち向かうためのアルゴリズムとして，近年深層学習が急速に注目を集めている．深層学習は圧倒的な汎化性能を持つことが特徴で，画像認識 [11] や機械翻訳 [12] などの分野で次々と最高性能を達成した．この波はまたたく間に音声・音楽・言語・画像処理の分野に広がり，深層学習を用いた論文の割合は，言語処理分野の四大会議 (ACL, EMNLP, EACL, NAACL) の全てで 50% を超えている [13]．音楽情報処理の国際会議 ISMIR でも深層学習は急速に存在感を増している [14]．

深層学習のアルゴリズムは成熟に長い時間を必要としたため，学習が難しく，チューニングに職人的技能が必要だというイメージが強い．しかし，ILSVRC 2012[*2] での大勝 [11] を境に深層ニューラルネットの学習アルゴリズムは急速に整備されつつある．また，特定の大規模タスクに特化したニューラルネットを鍛え直す転移学習という枠組みを使えば，比較的少数の教師データからネットワークを学習することも可能になってきている [15]．本稿ではこのような深層学習の最先端の知見に焦点を当て，音楽情報処理に深層学習を応用する方法について検討したい．

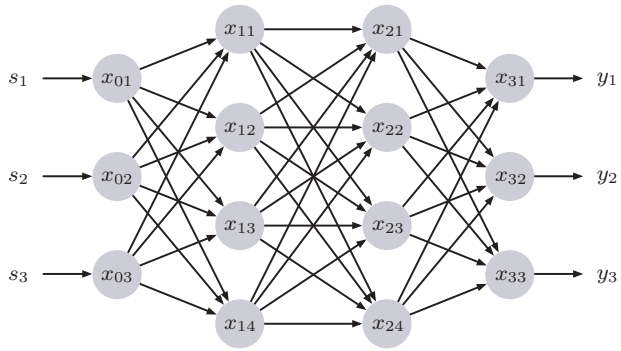

図 1 4 層の全結合ニューラルネットの模式図．

深層学習アルゴリズムは特定のデータ表現やモダリティに依存しないことが多く，分野を横断して使える技術が多いのが魅力である [16]．深層学習では実験の再現性が重視されるため，著者や第三者がソースコードを公開することも多い．研究用の深層学習フレームワークも整備が進んでおり，参入の敷居は下がりつつある．

本稿は二部構成で，ニューラルネットの設計と学習方法について説明した後，音楽情報処理を含めた各種メディア処理への応用例を紹介する．各分野から理論上重要なもの，音楽情報処理と共通点のあるトピックを取り上げつつ，音楽情報処理におけるアプリケーションを重点的に取り上げる．

[1] 株式会社コルグ
4015-2 Yanokuchi, Inagi-City, Tokyo 206-0812, Japan
[a)] sakaue@korg.co.jp

[*1] これだけにとどまらないと考えられるが，本稿の主旨に合わせて便宜上このように定義する．
[*2] ImageNet Large Scale Visual Recognition Competition

Vol.2018-MUS-119 No.2
Vol.2018-SLP-122 No.2
2018/6/16

表 1　各変数の次元数と役割.

変数	形	役割
S	$\mathbb{R}^{N\times I_S}$	入力データ
T	$\mathbb{R}^{N\times I_T}$	正解ラベル
$\boldsymbol{s}_n$	$\mathbb{R}^{I_S}$	n 番目の入力データ
$\boldsymbol{y}_n$	$\mathbb{R}^{I_T}$	n 番目の出力
$\boldsymbol{t}_n$	$\mathbb{R}^{I_T}$	n 番目の正解ラベル
$\boldsymbol{x}_d$	$\mathbb{R}^{I_d}$	中間表現
W_d	$\mathbb{R}^{I_d\times I_{d-1}}$	重み（係数）行列
$\boldsymbol{b}_d$	$\mathbb{R}^{I_d}$	バイアス
f_d	$\mathbb{R}^{I_d}\to\mathbb{R}^{I_d}$	活性化関数

2. 深層学習の基本

2.1 基本構造

ニューラルネットには多種多様な構造があり，入力変数（観測データ）から出力変数（推論結果）への接続方法を工夫することで学習する内容をおおまかに制御することができる．このネットワークは入力と出力以外に多数の中継地点を含んでおり，これらをまとめて中間ノードという．ネットワークの構造は入力から出力へ向かう有向非循環グラフ (Directed Acyclic Graph, DAG) で表現できる．

各種のネットワーク構造のうち，最も汎用性が高いのが全結合ネットワークと呼ばれるモデルである（図 1）．このモデルでは中間ノードをいくつかのグループ（レイヤ）に分け，隣り合うレイヤに含まれるノードの間を完全に接続する．各レイヤに均等な数のノードを配置したとき，このネットワークはレイヤ数（深さ）に対して指数的な表現能力を持つことが知られている [17].

全結合ネットワークを使い，I_S 次元の入力データから I_T 次元の出力データを予測するタスクを考える．観測データの数を N, データセットを $S=[\boldsymbol{s}_1,\cdots,\boldsymbol{s}_N]$ とする．また，ニューラルネットに $\boldsymbol{s}_n$ をあたえたときの出力を $\boldsymbol{y}_n$ とする．学習データには同じ数の正解ラベルが与えられるものとし，これを $T=[\boldsymbol{t}_1,\cdots,\boldsymbol{t}_N]$ とする．したがって，我々の目標は $\boldsymbol{y}_n$ が $\boldsymbol{t}_n$ になるべく近くなるようなネットワークのパラメータを探し出すことである．

$\boldsymbol{s}_n$ と $\boldsymbol{t}_n$ はそれぞれ I_S, I_T 次元のベクトルなので，$\boldsymbol{s}_n=[s_{n1},\cdots,s_{nI_S}]$, $\boldsymbol{t}_n=[t_{n1},\cdots,t_{nI_T}]$ と書くことができる．また，$\boldsymbol{s}_n$ に対する $D+1$ 層の全結合ネットワークの出力 $\boldsymbol{y}_n$ は次のように書ける．

$$\begin{aligned}\boldsymbol{x}_0&=\boldsymbol{s}_n\\ \boldsymbol{x}_1&=f_1(W_1\boldsymbol{x}_0+\boldsymbol{b}_1)\\ &\vdots\\ \boldsymbol{x}_D&=f_D(W_D\boldsymbol{x}_{D-1}+\boldsymbol{b}_D)\\ \boldsymbol{y}_n&=\boldsymbol{x}_D\end{aligned}\tag{1}$$

ネットワークの各層は入力をアフィン変換 $(W_d\boldsymbol{x}_{d-1}+\boldsymbol{b}_d)$ したのち，活性化関数とよばれる非線形関数 f_d を通して出力する．W_d はネットワークの重み，$\boldsymbol{b}_d$ はバイアスと呼ばれる．各変数の次元数は表 1 のとおりである．

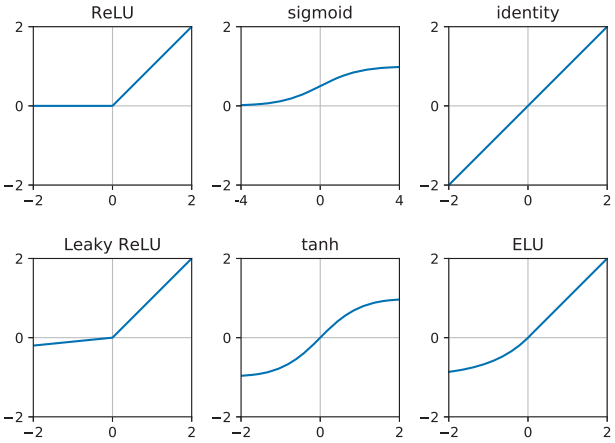

図 2　さまざまな活性化関数.

中間層 $(d<D)$ と出力層 $(d=D)$ では通常異なる活性化関数を使う．学習性能の良さから，中間層では Rectified Linear Unit (ReLU) [18] またはその派生形が使われることが多い（図 2）.

$$f_{\mathrm{ReLU}}(x)=\max(0,x)\tag{2}$$

$$f_{\mathrm{LReLU}}(x)=\max(ax,x)\tag{3}$$

$$f_{\mathrm{ELU}}(x)=\begin{cases}x & (x\geq 0)\\ \alpha(e^x-1) & (x<0)\end{cases}\tag{4}$$

また，時系列ネットワークでは sigmoid や tanh 関数もよく使われる.

$$f_{\mathrm{sigmoid}}(x)=\frac{1}{1+\exp(-x)}\tag{5}$$

$$f_{\tanh}(x)=\tanh(x)\tag{6}$$

線形変換のくり返しは単一の線形変換に書き直せるため $(A(B\boldsymbol{x}+\boldsymbol{c})+\boldsymbol{d}=AB\boldsymbol{x}+(A\boldsymbol{c}+\boldsymbol{d}))$，中間層の活性化関数は非線形とする必要がある.

出力層の活性化関数は正解ラベルのフォーマットに合わせて選ぶ必要がある．正解データが真偽値の場合は通常 sigmoid 関数を使う．また，クラス分類のための 1-of-K 表現のときは softmax 関数を使う.

$$f_{\mathrm{softmax}}(\boldsymbol{x}_n)=\frac{\exp(x_{ni})}{\sum_j\exp(x_{nj})}\tag{7}$$

回帰問題を解く場合は恒等関数

$$f_{\mathrm{identity}}(x)=x\tag{8}$$

を使う．選択が正しくない場合，予測ラベル $\boldsymbol{y}_n$ と正解ラベル $\boldsymbol{t}_n$ の間に定義されるコスト（損失）関数 $L(\boldsymbol{t}_n,\boldsymbol{y}_n)$ が

計算できなくなることがある．以上の活性化関数のうち，softmax 関数のみ各成分が相互に依存していることに注意すること．

活性化関数の選択は性能に直結する重要なテーマである．ReLU には Dead Neuron と呼ばれる弱点があり，すべての学習データに対して活性化関数の値がゼロになると勾配法を使った最適化ができなくなってしまう．Leaky ReLU [19] は ReLU の負側にもゆるやかなスロープを付けることでこの問題に対処している．ほかにも，Exponential Linear Unit (ELU) [20] などが提案されている．現在のところ，フィードフォワードネットワークには ReLU または Leaky ReLU，時系列を扱うリカレントニューラルネットワークには sigmoid と tanh を組み合わせて使うことが多い．レイヤ内のノード数を増やすと単純に性能が上がることも多いため [21], [22]，活性化関数の選択にあたっては GPU 上での計算速度も考慮する必要がある．

2.2 ニューラルネットの学習方法

深層学習の目標は，ネットワークの出力値 $\boldsymbol{y}_n$ が正解ラベル $\boldsymbol{t}_n$ になるべく近くなるようなパラメータ $W, \boldsymbol{b}$ を見つけることである．この問題は正解ラベル $\boldsymbol{t}_n$ とその予測値 $\boldsymbol{y}_n$ の間に定義される平均損失関数 $L(T,Y) = \sum_n L(\boldsymbol{t}_n, \boldsymbol{y}_n)/N$ の最小化問題として解くことができる．

活性化関数の場合と同様に，いくつかの重要な損失関数が集中的に使われている．中でも特に重要なのが，多クラスのソフトラベル間に定義されるクロスエントロピーである．

$$L_{\text{cross}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{I_T} t_{ni}\log y_{ni} \tag{9}$$

たとえば，3 クラス分類で n 番目の正解ラベルが $\boldsymbol{t}_n = [0,1,0]$ であったとする．ここへ $\boldsymbol{y}_n = [0.25, 0.5, 0.25]$ という予測をあたえる時，$\boldsymbol{t}_n$ と $\boldsymbol{y}_n$ の間のクロスエントロピーはおよそ 0.69 である．図 3 にクロスエントロピーのグラフを示した．

深層学習とクロスエントロピーの相性はとても良く，本来の用途を超えて使われることもある．連続値の回帰問題を解く場合，一般的な損失関数は L^2 ノルム

$$L^2 = \frac{1}{N}\sum_{n=1}^{N}(t_n - y_n)^2 \tag{10}$$

であるが，WaveNet [23] では時間領域の音響信号を μ-law 変換[*3]して 8bit の値に離散化したのち，クロスエントロピーを計算して損失関数としている．

深層学習では過学習がよく問題となるため，データセットを 3 つに分割することが多い．これを順番にトレーニング・バリデーション・テストという．トレーニングデータは，パラメータの更新に使うためのデータである．パラメータを更新したら定期的にバリデーションデータでの性能を確認し，トレーニングデータへの過学習が始まったらすぐに更新を終了する．バリデーションデータは汎化性能を予測するための見本である．テストデータは言うまでもなく性能評価用のデータセットで，決して学習に使ってはならない．

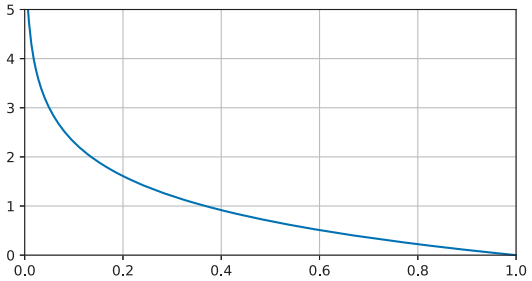

図 3　2 クラスの場合のクロスエントロピー．

深層学習はその性格から，正解ラベル付きの大規模な学習データが必要になりやすい．経験上，実世界のデータは計算機上の次元数より小さい多様体上に存在する[*4]と考えられており，これは多様体仮説と呼ばれている [24]．ニューラルネットの汎化性能を引き出すには，十分な量のデータを準備しこの多様体の形をネットワークに正しく認識させる必要がある．音楽情報処理の場合は数百から数千曲程度のデータセットがよく使われる [4], [5], [6], [25]．

2.3 重みとバイアスの初期化

ニューラルネットのパラメータは勾配法によって学習されるため，適切な初期値をあたえる必要がある．実際によく使われる初期化アルゴリズムは限られており，これらは考案者の名前で呼ばれている [26], [27], [28]．いずれも，深層ニューラルネットの各レイヤで勾配のスケールを一定に保つことを主な目的としている．

現行のアルゴリズムで最も古くから使われているのは LeCun のアルゴリズムである [26]．この方法では tanh の原点付近でのふるまいが恒等写像となることに着目し，この近似のもとで $\mathbb{V}[\boldsymbol{x}_d] = I_{d-1}\mathbb{V}[W_d]\mathbb{V}[\boldsymbol{x}_{d-1}]$ となることから[*5]

$$W_d \sim \mathcal{N}\left(0, \frac{1}{I_{d-1}}\right) \tag{11}$$

とすることでレイヤ間の分散を揃えている．I_{d-1} はノードに流れ込むリンクの数であり，fan-in と呼ばれている．Glorot らは tanh 関数を使ったネットワークの初期化に

[*3] おおまかに言うと，聴覚特性を考慮して信号の対数値を量子化している．

[*4] たとえば 32x32 の画像は 1024 次元のベクトルとして表現できるが，数字として有効な画像は 100 次元程度でパラメータ付けできる，という考え方．

[*5] $\mathbb{V}$ は分散をあらわす．

経験上一様分布が使われていることを確認し[*6]，一様分布 $\mathcal{U}(-1,1)$ の分散が 1/3 となることから

$$W_d \sim \mathcal{U}\left(-\frac{\sqrt{6}}{\sqrt{I_{d-1}+I_d}}, \frac{\sqrt{6}}{\sqrt{I_{d-1}+I_d}}\right) \tag{12}$$

という初期化方法を提案している．I_d はノードから流れ出すリンクの数であり，fan-out と呼ばれている．この方法は筆頭著者のファーストネームから Xavier のアルゴリズムとも呼ばれる．He らは ReLU の勾配が $x > 0$ の範囲にしかないことを指摘し，LeCun のアルゴリズムの 2 倍の分散を使うことを提案している [28].

$$W_d \sim \mathcal{N}\left(0, \frac{2}{I_{d-1}}\right) \tag{13}$$

この方法は 10 層を超える超深層の ReLU ネットワークで大幅な性能向上を達成することが報告されている [28]. Mishkin らは重み行列を直交行列で初期化した後，ネットワークに実際に信号を流しつつこの信号を正規化するようなスケーリングを提案している [29].

こうした数々の努力にもかかわらず，あらゆるケースに使える統一的なアルゴリズムは未だ存在していない．したがって，計算機リソースの範囲内で各種の初期化アルゴリズムを試してみるのが理想的である．時系列を取り扱うリカレントネットワークの初期化は特に難しく，先行研究を参照することが望ましい [30]．バイアス変数は通常ゼロに初期化される[*7].

2.4 更新式

ニューラルネットは入力 $\boldsymbol{x}_n$ とネットワークパラメータ θ から出力 $\boldsymbol{y}_n$ を計算する関数とみなすことができる．ここで，$\theta = (W_1, \cdots, W_D, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_D)$ である．

ニューラルネットによる非線形変換を $F(\boldsymbol{x}, \theta)$ と書くと，損失関数は

$$L(T,Y) = \frac{1}{N}\sum_{n=1}^{N} L(\boldsymbol{t}_n, F(\boldsymbol{x}_n, \theta)) \tag{14}$$

と書ける．損失関数を θ で微分すると，最急降下法によってパラメータを学習することができる．

$$\theta^{\text{new}} = \theta^{\text{old}} - \eta \frac{\partial L}{\partial \theta}(T,Y) \tag{15}$$

η は学習率をあらわす．損失関数の微分はバックプロパゲーションと呼ばれる動的計画法アルゴリズムによって効率的に解くことができる．このアルゴリズムは各種の深層学習ライブラリに標準で実装されている．

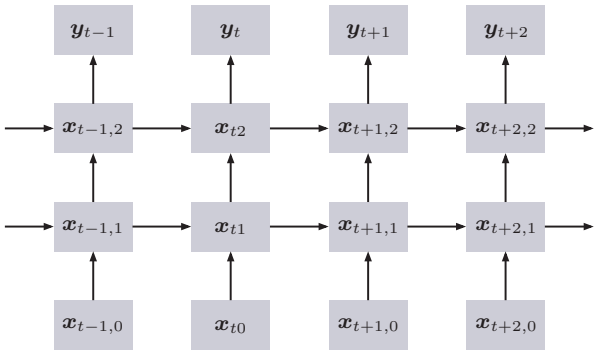

図 4 2 層リカレントニューラルネットワークの模式図.

実際にはすべての学習データを使って勾配を計算するバッチ学習ではなく，ランダムに選択した少数のデータを使う確率的最急降下法 (Stochastic Gradient Descent, SGD) を使う．この少数データをミニバッチといい，単一のミニバッチを使って学習を行うことをイテレーション，イテレーションを全データ数分実行することをエポックという [26]．たとえば，データセットのサイズが 60000，ミニバッチの大きさが 200 の場合，1 エポックは 300 イテレーションに相当する．

1 エポックを 300 イテレーションに分割すると，同じ学習率でも 300 倍のペースで学習を進めることができる．Bengio 氏はこのことについて，最急勾配は場所によって変わり続けるため，正確に計算することにはあまり価値がないと指摘している [21]．ミニバッチのサイズはデータセット全体よりはるかに小さくてもよい．[21] ではミニバッチのデフォルトの大きさとして 32 が推奨されている．

2.5 オプティマイザ

実際の応用で確率的最急降下法 (SGD) が定義通り使われることは少なく，学習速度や汎化性能の向上を狙った各種の改良アルゴリズムが使われている [33]．これらを総称してオプティマイザという．たとえば，モーメンタム法 [34] とよばれるアルゴリズムではパラメータの更新に慣性をつけ，数値的な安定性を増している．

$$\boldsymbol{v}^{\text{new}} = \alpha \boldsymbol{v}^{\text{old}} + \eta \frac{\partial L}{\partial \theta}(\hat{T}, \hat{Y}) \tag{16}$$

$$\theta^{\text{new}} = \theta^{\text{old}} - \boldsymbol{v}^{\text{new}} \tag{17}$$

$\hat{T}$ と $\hat{Y}$ はミニバッチをあらわす．

最適な学習率はネットワーク構成やパラメータによって異なるため，学習率を適応的に変化させることのできる各種のアルゴリズムが提案されている．たとえば，AdaGrad [35], RMSprop[*8]などがある．また，Adam [36] と呼ばれる応用アルゴリズムが実用上よく使われている．優れたオプティマイザの開発は重要なテーマであり，この他にも様々な手法が提案されている．

[*6] 初期化に一様分布を使う理由は詳しく述べられていないが，個人的に LeCun の近似が $x \gg 1$ で成り立たないためではないかと考えている．

[*7] 長距離の学習を優先するため，LSTM の忘却ゲートのバイアスをあらかじめ 1 に初期化する方法が Gers らによって提案されている [31]．この手法は十分に広まっていないとして Jozefowicz らの近年の論文で再度強調されている [32].

[*8] RMSprop は Hinton 氏の講義資料の中で発表されている [33].

2.6 時系列モデリング

可変な長さをとる時系列データのモデル化にはリカレントニューラルネットワーク (Recurrent Neural Network, RNN) が使われる．これは隠れマルコフモデルと同様に，各時刻で固定長の入力を受け取り，ニューラルネットの「状態」を更新し，次の時刻での入力の一部とするようなネットワークである．RNN も全結合ネットワークのように多層とすることができる．図 4 に概要を示した．

RNN の学習は難しく，勾配爆発・勾配消失と呼ばれる問題が起きやすい [30]．RNN の時間方向のリンクは超深層のニューラルネットと考えることができるが，RNN では時刻間の重み行列が共有されているため，この影響が指数的に大きくなってしまう．Pascanu らは，勾配爆発にはノルムの最大値を制限する Gradient Clipping, 勾配消失には正則化を使うことを提案している [30]．

もう一つのアプローチではネットワークの構成を工夫し，勾配の指数的なスケーリングを回避する．このため，RNN の各ノードを Long Short-Term Memory (LSTM) [37]，および Gated Recurrent Unit (GRU) [38] で置き換える方法が提案された（図 5, 6）．

LSTM では，信号の伝搬時に入力ゲート $\boldsymbol{i}_t$，忘却ゲート $\boldsymbol{f}_t$，出力ゲート $\boldsymbol{o}_t$ とよばれる 3 種類のゲート値を計算する．記号 $\odot$ は要素積をあらわす．t は時刻である．

$$\tilde{\boldsymbol{i}}_t = W_i \boldsymbol{x}_t + R_i \boldsymbol{y}_{t-1} + \boldsymbol{p}_i \odot \boldsymbol{c}_{t-1} + \boldsymbol{b}_i \tag{18}$$

$$\tilde{\boldsymbol{f}}_t = W_f \boldsymbol{x}_t + R_f \boldsymbol{y}_{t-1} + \boldsymbol{p}_f \odot \boldsymbol{c}_{t-1} + \boldsymbol{b}_f \tag{19}$$

$$\tilde{\boldsymbol{o}}_t = W_o \boldsymbol{x}_t + R_o \boldsymbol{y}_{t-1} + \boldsymbol{p}_o \odot \boldsymbol{c}_{t-1} + \boldsymbol{b}_o \tag{20}$$

各ゲートの値は現在の入力 $\boldsymbol{x}_t$，前回の出力 $\boldsymbol{y}_{t-1}$，LSTM の前回の状態 $\boldsymbol{c}_{t-1}$ を参考にして決められる．$W, R, \boldsymbol{p}$ はネットワークの重み，$\boldsymbol{b}$ はバイアスである[*9]．ゲートの最終的な値は sigmoid 関数で非線形変換した値とする．

$$\boldsymbol{i}_t = \sigma(\tilde{\boldsymbol{i}}_t) \tag{21}$$

$$\boldsymbol{f}_t = \sigma(\tilde{\boldsymbol{f}}_t) \tag{22}$$

$$\boldsymbol{o}_t = \sigma(\tilde{\boldsymbol{o}}_t) \tag{23}$$

これらのゲート値を使い，現在の時刻に対する状態 $\boldsymbol{c}_t$ と出力 $\boldsymbol{y}_t$ を計算する．

$$\tilde{\boldsymbol{z}}_t = W_z \boldsymbol{x}_t + R_z \boldsymbol{y}_{t-1} + \boldsymbol{b}_z \tag{24}$$

$$\boldsymbol{z}_t = \tanh(\tilde{\boldsymbol{z}}_t) \tag{25}$$

$$\boldsymbol{c}_t = \boldsymbol{z}_t \odot \boldsymbol{i}_t + \boldsymbol{c}_{t-1} \odot \boldsymbol{f}_t \tag{26}$$

$$\boldsymbol{y}_t = \tanh(\boldsymbol{c}_t) \odot \boldsymbol{o}_t \tag{27}$$

現在の時刻に対応した新しい状態の候補 $\boldsymbol{z}_t$ を計算したのち，入力ゲートと忘却ゲートの値を使って新しい状態 $\boldsymbol{c}_t$ を計算している．状態 $\boldsymbol{c}_t$ をさらに tanh 関数で変換し，出力ゲートの値を掛けたものが最終的な出力 $\boldsymbol{y}_t$ となる．忘却ゲート[*10] の値が大きいセルは既存の値を引き継ぎやすいため，時間差のある入出力の関係を学習できる．LSTM には Peephole Connection などを含めたいくつかのバリエーションがあるため，既存研究の追試をする場合は注意が必要である [39].

[*9] R は Recurrent, $\boldsymbol{p}$ は Peephole に対応している [39].

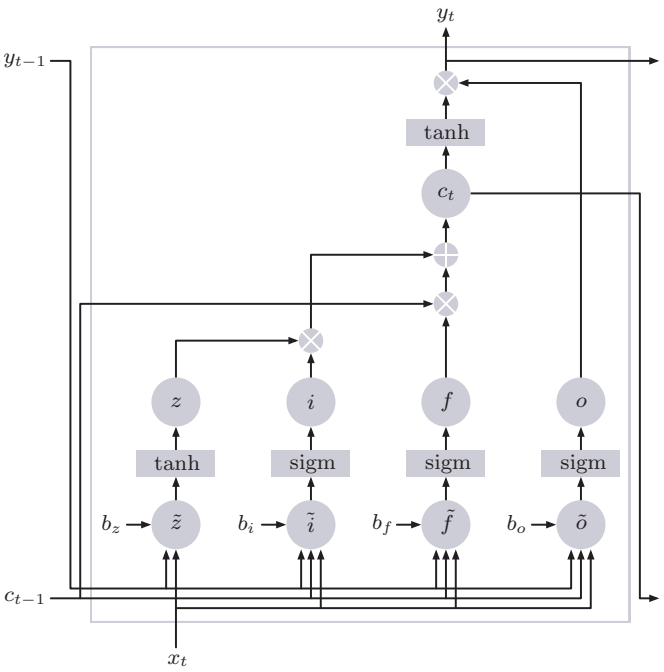

図 5 LSTM の模式図.

GRU は 2014 年に Cho らによって発表された新しいモデルである．GRU ではリセットゲート $\boldsymbol{r}_t$ と更新ゲート $\boldsymbol{z}_t$ を使い，新しい状態を計算する．

$$\boldsymbol{r}_t = \sigma(W_r \boldsymbol{x}_t + U_r \boldsymbol{h}_{t-1}) \tag{28}$$

$$\boldsymbol{z}_t = \sigma(W_z \boldsymbol{x}_t + U_z \boldsymbol{h}_{t-1}) \tag{29}$$

$$\tilde{\boldsymbol{h}}_t = \tanh(W_h \boldsymbol{x}_t + U_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})) \tag{30}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t)\tilde{\boldsymbol{h}}_t \tag{31}$$

LSTM と同様に，新しい状態の候補 $\tilde{\boldsymbol{h}}_t$ を計算する．これを過去の状態と統合し，最終的な出力 $\boldsymbol{h}_t$ とする．通常の RNN と同様に，LSTM や GRU も多層にすることができる．Sutskever らによる seq2seq モデルの実装では 4 層の Deep LSTM が使われている [40].

LSTM と GRU はどちらも複雑な形をしているため，よりシンプルな時系列モデルを作り出す試みが数多くなされてきた．Chung らは音楽と音声の 2 つのタスクで vanilla RNN[*11], LSTM, GRU の性能を比較し，LSTM と GRU が vanilla RNN に対して優位な結果であること，LSTM

[*10] この名称には語弊がある．ゲート値の本来の意味は維持ゲートである．

[*11] 時系列方向を全結合ネットワークで接続した RNN．LSTM-RNN, GRU-RNN と区別するための表現.

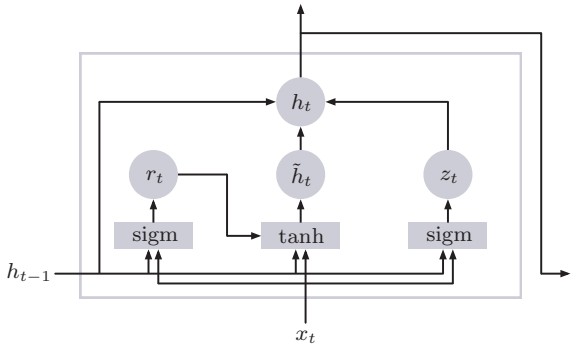

図 6 GRU の模式図.

と GRU が同程度の性能であることを確認した [41]．Greff らは LSTM から各ゲートを省略して総合的な性能評価を行い，忘却ゲートと出力の非線形関数が最重要の構成要素であると結論づけた [39]．Jozefowicz らは遺伝的アルゴリズムにより代替モデルを探索し，LSTM や GRU と同程度の性能を持つ 3 種類のネットワークを発見した [32]．Karpathy らは LSTM の動作を定性的に解析し，括弧の対応や行の長さなどの特徴が学習されることを確認している [42]．しかしこれまでのところ，LSTM と GRU の特性を完全に解き明かし，より性能の高いモデルを提案することはできていない．個別のタスクでは LSTM と GRU のパフォーマンスが異なることも多いため，両方を試した上で性能の高いモデルを選択することが望ましい．

2.7 畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) は画像の認識に特化したニューラルネットで，全結合ネットワーク，RNN と並び重要なネットワーク構造である．CNN の各レイヤはノードが二次元に整列された形で並べられており，その位置は画素の x, y 座標に対応している．ノードへの入力は隣接レイヤの「近い」座標のノードに限定されており，この構造をフィルタという (図 7)．フィルタサイズが 3x3 の場合，d 番目の層の値は以下の通り計算できる．

$$x_{ij}^{d} = f\left(\sum_{a=1,b=1}^{3,3} w_{ab}^{d} x_{i-1+a,j-1+b}^{d-1} + b^{d}\right) \qquad (32)$$

w_{ab}^{d} はフィルタの重み，b^{d} はバイアスである．ネットワーク中の各ノードについて，ノードに到達できる入力画像の範囲を受容野 (Receptive Field) という．上位のレイヤほどノードの受容野は広く，大きく複雑な物体を識別することができる [43]．CNN の利点は，全結合ネットワークに比べパラメータ数が少なく安定に学習できること，画像の持つ並進不変性をうまく表現できることである．実用上，CNN の各レイヤは複数の特徴マップを保持している．これをチャンネルという．CNN の入力は通常 3 チャンネルで，RGB の輝度をあらわす．音響信号のスペクトログラムを画像とみなすことで，音楽音響信号にも応用できる [44].

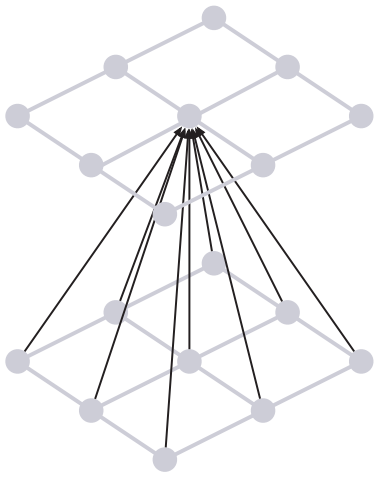

図 7 CNN の模式図．フィルタサイズは 3x3.

ニューラルネットでは非線形変換を繰り返すため，恒等写像を学習することは案外難しい．このためレイヤ数が必要よりも余分に存在する場合に性能が下がるという問題がある．He らが提案した ResNet [45] は CNN の拡張アルゴリズムで，各レイヤは恒等写像からの差分を学習するよう変形されている ($y = f(x) + x$). ResNet では 1000 層を超えるネットワークを学習することができる．[46] は CNN の詳細な解説論文である．

2.8 汎化のためのテクニック

ニューラルネットは適切な学習のもと高い汎化性能を発揮するが，設定によっては過学習が起こる [47], [48]．過学習の原因となる高い表現能力は，普遍性定理という名前で知られている [49], [50]．この定理は，十分な数のノードを持つネットワークが任意の非線形変換を学習できるというものである．このような過学習の実例として，Zhang らは ImageNet の各画像にランダムなラベルを付与した場合でも画像とラベルの対応を 100% 学習できてしまうことを示した [51]．この結果はニューラルネットの性能がデータセットの品質に大きく影響されることと，過学習を回避するための正則化が重要なことを示している [48]．各種の正則化の内でも特に重要なのが Dropout, Batch Normalization, Weight Decay, Early Stopping である．

Dropout [47] と呼ばれる手法では，ミニバッチの学習時にノードの半数[*12]をランダムに削除して勾配を計算する (図 8)．ネットワーク中の各ノードは他のノードの存在を仮定した学習ができなくなるため，スパースかつ単独で意味のある特徴を学習するよう誘導される．ネットワークの学習後には全てのノードを使うことで高い性能を発揮する．

Batch Normalization [52] は中間レイヤの出力をミニバッチごとに白色化し，平均を 0，分散を 1 に揃えるアルゴ

[*12] 削除するノードの割合は変えてもよい.

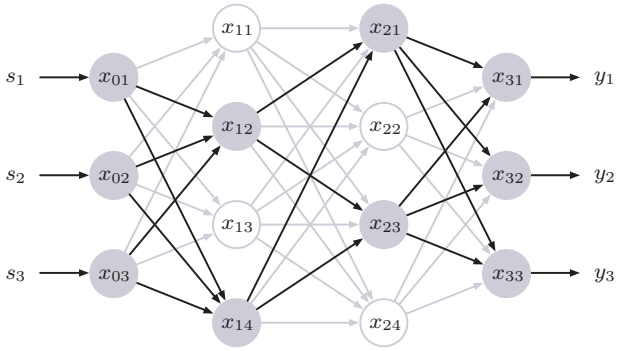

図 8 Dropout の模式図．白で表示されたノードはミニバッチの推論から除外される．

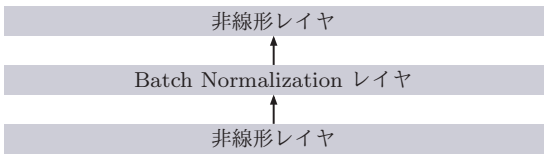

図 9 Batch Normalization レイヤの模式図．隣り合う非線形レイヤの間に追加する．

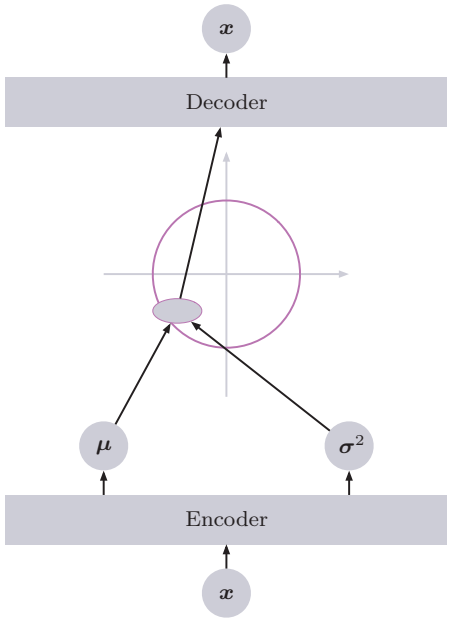

図 10 変分オートエンコーダの模式図．エンコーダは潜在ベクトルの事後分布をモデル化している．

リズムである（図 9）．Batch Normalization レイヤを挟まない場合，学習によって特定のレイヤの出力傾向が大きく変わり，後続レイヤの学習がやり直しになってしまうことがある．この現象を内部共変シフトと呼ぶ．本手法は元々学習を高速化するために提案されたが，一定の汎化性能を持つことが分かってきている [53]．

Weight Decay （荷重減衰） [53] は確率モデルの場合と同様に，大きすぎる係数に対してペナルティを加える方法である．ペナルティ項は損失関数に追加する形で与えることができる．L1 $\|\theta\|$ または L2 $\|\theta\|^2$ ノルムが通常よく用いられる．荷重減衰はニューラルネットの自由な学習を制限してしまうため，Dropout と Batch Normalization のみを使うケースも多い．

バリデーションデータでの性能が最大となるエポック付近で学習を止める Early Stopping もよく使われる．Arpit らは一部の正解ラベルにノイズを加えることで本手法の有効性を検証している [48]．この論文では，ニューラルネットの学習には 2 つのフェーズがあり汎化フェーズと過学習フェーズに分かれることを主張している．

深層学習の汎化性能は重要なトピックであり，さまざまな正則化手法が提案されている．Gal らは通常の Dropout が RNN の性能向上につながらない理由を検討し，理論的な立場から RNN に合わせた Dropout の適用方法を提案した [54]．DropConnect [55] は Dropout の応用アルゴリズムで，ノードではなくエッジ単位で重みを削除する．ほかにも Maxout [56], Layer Normalization [57], Zoneout [58] などが提案されている．Pre-training のように，かつて必須と考えられたが深層学習アルゴリズムの発展により使われなくなった手法もある [21], [59]．

2.9 転移学習

ニューラルネットは複数のレイヤを組み合わせてタスクの学習を行う．このとき，入力に近いレイヤはデータの基本的な特徴を学習し，出力に近いレイヤはこの特徴量を組み合わせてタスクを解くように学習が進む [15]．CNN の学習結果から，入力データ付近のレイヤで学習される特徴量はタスクによらず普遍的であることが知られている [43]．

転移学習の枠組みでは，十分なデータのあるタスクで普遍的な特徴量を獲得した後，この特徴量を各種のタスクに応用することを考える [60]．ニューラルネットが獲得する普遍的な特徴量は入力データ付近に集中するため，ことなる目的で一度学習されたネットワークの出力側のレイヤを再初期化し，真の目的タスクに合わせて学習し直すことが考えられる．また，Bojanowski [61] らはニューラルネットの特性に着目し，乱数をターゲットとした場合でもこの普遍的特徴量が獲得できることを示した．Choi らは転移学習を音楽情報処理に応用した事例を報告している [62]．

2.10 教師なし学習

ニューラルネットの入出力や学習アルゴリズムを工夫することで，観測データの分布を推定するための教師なし学習を行うことができる．代表的なアルゴリズムは変分オートエンコーダ (Variational Autoencoder, VAE) [63], [64] および敵対的生成ネットワーク (Generative Adversarial Network, GAN) [65] の二種類である．

VAE の目標は観測データを低次元の潜在ベクトルで表現し，この潜在ベクトルと観測データの対応関係を学習することである．このような手法を総称して表現学習という

[66]．VAE では潜在ベクトル $\boldsymbol{z}$ が標準正規分布 $\mathcal{N}(0, I)$[*13] に従うことを仮定する．VAE のデコーダは $\boldsymbol{z}$ があたえられた時の観測データ $\boldsymbol{x}$ の分布 $p(\boldsymbol{x}|\boldsymbol{z})$ を，エンコーダは $\boldsymbol{x}$ があたえられたときの $\boldsymbol{z}$ の事後分布 $p(\boldsymbol{z}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$ をニューラルネットによってそれぞれモデル化する．

VAE では潜在ベクトルをベイズ的にモデル化するため，エンコーダとデコーダの接続時には確率分布からのサンプリングが必要となる（図 10）．エンコーダは潜在ベクトルの位置を正規分布でモデル化し，この平均 $\boldsymbol{\mu}$ と分散 $\boldsymbol{\sigma}^2$ を出力する[*14]．この分布から実際の潜在ベクトルの値をサンプリングすることで，デコーダの出力を計算できる．ネットワーク中に確率的なノードを含むグラフを確率的計算グラフ (Stochastic Computation Graph, SCG) と呼ぶ [67]．

確率的計算グラフの損失関数は潜在ベクトルのすべての可能性についての期待値

$$L(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\sigma}^2)}[L(\boldsymbol{x}, F_{\mathrm{Decoder}}(\boldsymbol{z}))] \tag{33}$$

として計算できるが，VAE ではニューラルネットを含むため期待値を解析的に計算することも，サンプリングによって近似することもできない [63]．VAE ではこの問題を回避するため，Reparametrization Trick というテクニックを使う．この方法では SCG を乱数源と確定的な計算グラフの組み合わせに分解するため，誤差逆伝搬法によるパラメータの更新が可能になる．この乱数はミニバッチごとに新規にサンプリングされる．

観測データの生成分布をモデル化するもう一つの方法は敵対的生成ネットワーク (GAN) である．GAN ではその名の通り，入力データの真贋を判定する Discriminator ネットワーク $D(\boldsymbol{x})$ と偽の入力データを作る Generator ネットワーク $G(\boldsymbol{z})$ を敵対的に学習する．$\boldsymbol{z}$ はデータを生成するための乱数シードである．Discriminator の目標が真のデータに対して $D(\boldsymbol{x}) = 1$，偽のデータに対して $D(\boldsymbol{x}) = 0$ を返すことであるのに対し，Generator の目標は Discriminator を欺くこと，すなわち $D(G(\boldsymbol{z})) = 1$ を出力させることである．このとき最適化すべき目標関数は

$$\min_G \max_D \tilde{V}(\hat{X}, \hat{Z}) = \sum_{\boldsymbol{x}\in\hat{X}} D(\boldsymbol{x}) - \sum_{\boldsymbol{z}\in\hat{Z}} D(G(\boldsymbol{z})) \tag{34}$$

のように書ける．$\hat{X}, \hat{Z}$ は真のデータおよび Generator からサンプリングされたミニバッチをあらわす．Discriminator と Generator の目標が相反するため，最適化は min max の形で書ける．実際には，より収束特性のよい

$$V(\hat{X}, \hat{Z}) = \sum_{\boldsymbol{x}\in\hat{X}} \ln D(\boldsymbol{x}) + \sum_{\boldsymbol{z}\in\hat{Z}} \ln(1 - D(G(\boldsymbol{z}))) \tag{35}$$

という形の目標関数を最適化する．

[*13] I は単位行列をあらわす．
[*14] Kingma らは分散を対角行列でモデル化している [63]．

GAN の学習は非常に難しく，タスクに合わせた試行錯誤が必要になる [68]．よく問題になるのは Generator が同じ出力を生成し続ける Mode Collapse という現象である[*15]．また，GAN では複雑なオブジェクトの構造を捉えることが難しく，論理的に正しくない画像が生成されることも多い．これらの問題に対処すべく，さまざまな亜種が開発されている [68]．こうした手法の代表例として，DCGAN [69] や StackGAN [70] が挙げられる．

2017 年になって，収束性能の高い GAN のアルゴリズム (Wasserstein GAN, WGAN) が発表された [71], [72]．WGAN では著者らによって Mode Collapse が起きないことが主張されており，実際に多くのタスクで高い性能を示すことから注目を集めている．WGAN では通常の GAN のような敵対的学習ではなく，最適輸送距離（Wasserstein 距離）の近似と最適化によって学習を進める．WGAN の詳細な解説は本稿の範囲を超えるため，最適輸送理論 [73] および関連分野の専門書 [74], [75] を紹介する．Lucic らは [76] で WGAN を含む包括的な性能評価を行なっている．

2.11 学習・汎化性能向上へのさらなる取り組み

ニューラルネットの学習速度・汎化性能の向上は重要なテーマであり，最新のアルゴリズムが次々と発表されている．ImageNet の学習時にはエポック間で学習率を落としていくことが多いが，Smith らは学習率の代わりにバッチサイズを上げることで同等の性能を高速に実現している [77]．また，宮戸らはスペクトル正則化という方法により GAN を安定かつ高速に学習することに成功している [78]．深層学習は進歩が著しい分野であるため，最新の研究を調査することで困難なタスクが解けることもある．

3. 音声・言語・画像処理分野での応用

ここでは各分野の研究成果のうち，音楽情報処理に関係する内容を中心に取り上げる．深層学習のブレイクスルーとして有名になったのは 2012 年度の ImageNet Competition で，Hinton 氏率いるチームが次点に 10 ポイント近い大差をつけて優勝した [11]．このタスクでは約 100 万枚のラベル付き学習データが提供され，1000 クラスの分類問題に回答する．回答方式は Top-5 と呼ばれ，5 個の予測ラベルのうちどれかが正解と一致すればよい．これ以降しばらく ImageNet は新しい学習アルゴリズムを発表する場となり，重要な手法が次々に提案された [28], [45], [79], [80]．

画像の自動生成に関する研究も盛んである．VAE と GAN については先に説明した．Gatys らは画像をオブジェクトとスタイルに分割し，任意の絵画から抽出したスタイルと他の画像を組み合わせる Neural Artistic Style Transfer を開発した [81]．Li らは，この手法においてグラ

[*15] この出力が式 (35) の局所最適解となることに注意すること．

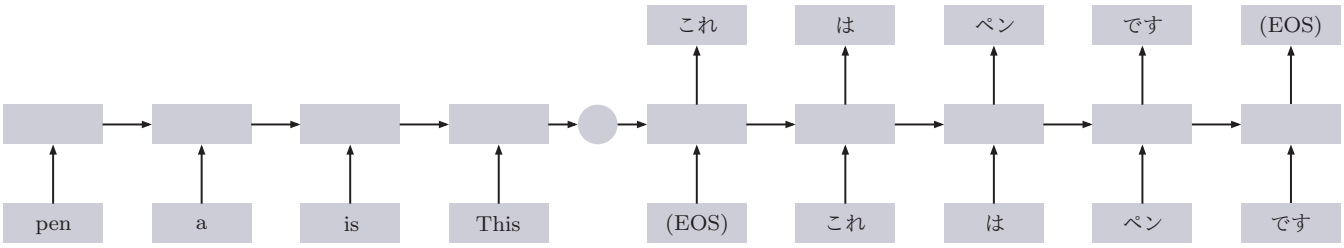

図 11 Sequence-to-Sequence (seq2seq) モデルの模式図．中央の丸印は埋め込みベクトル．

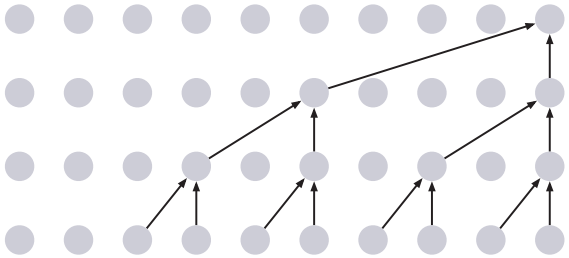

図 12 WaveNet の模式図．

ム行列がスタイルを表現する理由を解析している [82]．

機械翻訳の分野では，Cho らによって Encoder-Decoder ネットワークが開発された [38]．これは長さのことなる入出力の対応関係を学習できるニューラルネットで，自動翻訳などに活用されている（図 11）．Encoder と Decoder はそれぞれことなる RNN で，Encoder は可変長の入力を固定長のベクトルに変換し，Decoder は固定長のベクトルを可変長の出力に変換する．学習時には，ネットワーク全体の誤差逆伝搬法によってパラメータを最適化する．Sutskever らは Encoder-Decoder ネットワークと LSTM を組み合わせた Sequence-to-Sequence (seq2seq) モデルを使い，英仏翻訳の結果を報告している [40]．彼らは同時に，Encoder への単語の入力を逆順とすることで性能が向上することを報告している．その後さらに注意機構 (Attention) と呼ばれるモデル [83], [84] が開発され，Google などが提供するニューラル機械翻訳 (Neural Machine Translation, NMT) システムに応用されている [12]．

van den Oord らが開発した WaveNet [23] は時間領域の音響信号を直接モデル化することができる．時間的に離れたサンプル間の相関を扱うため，レイヤごとに異なる幅で畳み込み処理をしているのが特徴である（図 12）．この構造を Causal Dilated Convolution という．WaveNet は計算コストが非常に大きいため，Kalchbrenner らはより高速な WaveRNN [85] を開発している．音声合成の最新の研究成果としては，Shen らによる Tacotron 2 [86] などがある．また，小山田らは深層学習を使って位相復元を行うアルゴリズムを提案している [87]．

深層学習の特徴のひとつは複数のモダリティに関係したタスクに強いことである．Vinyals らは CNN で生成した固定長のベクトルを LSTM ネットワークに入力し，画像のキャプションを自動生成している [88]．また，梶原らは Google ストリートビューの表示にあわせて適切な環境音を選択する Imaginary Soundscape [16] を実装している．

4. 音楽情報処理分野での応用

深層学習を音楽情報処理に適用した初期の例としては [89] が挙げられる．本節では音楽情報処理に深層学習を活用したさまざまな研究を紹介する．

4.1 和音認識・多重基本周波数推定

深層学習は音楽情報処理のさまざまなタスクに応用されている．Korzeniowski らは音響信号のスペクトル情報から深層学習を使ってクロマベクトルを計算するアルゴリズムを開発し，Deep Chroma Extractor [5] と名付けた．このアルゴリズムでは音響信号のスペクトルを 100 ms ごとに計算し，対数周波数フィルタバンクを通して 178 次元に圧縮する．さらに，スペクトルの強度を対数値 $S = \log(1+|X|)$ に変換する[*16]．非調波音や打楽器音の強いフレームでもコードを正しく推定するため，認識対象の音響フレームを含む前後 15 フレームを同時に入力する．実験には 383 曲が使われている．ネットワークの構成を図 13 に示した．

McFee らは和音のテンションや転回形を判別できるニューラルネットを開発した [25]．音響信号の短時間フーリエ変換 (STFT) を CNN, 双方向 GRU (Bidirectional GRU) に順番に入力し，GRU の値から 4 種類の識別器（和音の基本形，根音，クロマベクトル，ベース音）の出力を計算する．これらの出力を統合し最終的な和音ラベルを出力している．

Bittner らは多重基本周波数解析を行うため，Harmonic CQT という畳み込みネットワークを開発した [44]．この手法では，倍音間の相互作用を扱うため Constant-Q スペクトログラムを周波数方向に $\log N$ ずつシフトした値を CNN の各チャンネルに入力している．

4.2 ビートトラッキング・ドラム譜推定

Böck らは 3 層の Bidirectional LSTM を使い，ビートトラッキングを実装している [6]．構成音の変化や打楽器音

[*16] 音響信号のもつ指数的な特性が ReLU ネットワークに与える負担を減らすためと考えられる．

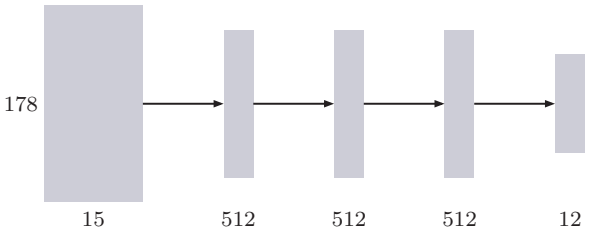

図 13 Deep Chroma Extractor [5] の構成図．STFT を 15 フレーム連続で切り出し，12 次元のクロマベクトルを推定する．

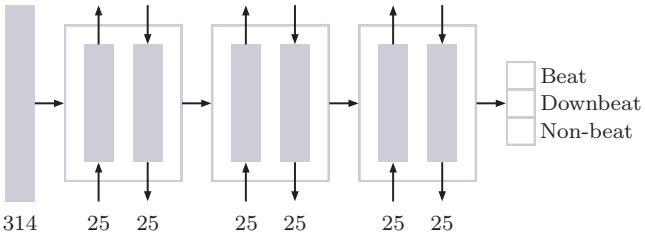

図 14 3 層の Bidirectional LSTM を使ったビートトラッキング [6] の構成図．

を同時にとらえるため，3 種類の窓幅の STFT から算出した対数周波数スペクトログラムを統合し，入力としている．この特徴量は 157 次元で，さらに微分値を加えた 314 次元をニューラルネットに入力する．図 14 にネットワークの構成を示す．

Vogl らは楽曲のドラム譜面を推定するため，拍位置とドラム音のオンセット位置を同時推定するニューラルネットを開発した [90]．このネットワークでは連続する 13 フレームを複数段の CNN に入力した後，この出力をさらに三段の Bidirectional GRU レイヤに入力している．GRU の出力を Bass, Snare, Hi-Hat, Downbeat, Beat の 5 種類の識別ニューロンに入力することでドラム譜面を推定している．著者らはこのネットワーク構成を Convolutional Recurrent Neural Network (CRNN) と呼んでいる．

4.3 歌声情報処理

Blaauw らはニューラルネットによって歌声を合成するシステムを開発し，Neural Parametric Singing Synthesizer [8] と名付けた．本手法では WaveNet の出力をボコーダーのパラメータとすることで高速な推論を実現している．Jansson らは U-Net と呼ばれるネットワーク構成を使い，楽曲中の歌声を分離している [91]．

4.4 転移学習

Choi らは転移学習を音楽情報処理に適用した結果を報告している [62]．音楽情報処理のタスクは多岐にわたり，その中には十分な教師データが用意できないケースも多くある．この論文では大規模な楽曲データベースを使ったタグ情報の学習を行い，得られたネットワークを 6 種類のタスクに転移学習した結果を報告している．

4.5 Google Magenta

Google の音楽情報処理チーム Magenta からも活発に研究成果が報告されている．Engel らは WaveNet を応用した NSynth というニューラルネットを使い楽器音を合成している [92]．Hawthorne らはピアノ音のオンセット位置と発音区間に個別の損失関数をあたえることで，高性能な自動採譜を実現している [93]．Roberts らは LSTM と VAE を組み合わせた MusicVAE というネットワークを開発し，複数の音楽フレーズを意味のある形で補間することに成功している [94]．

5. Further Reading

本稿を超えて深層学習を勉強される方のために，各種の参考文献を紹介する．Goodfellow らによる Deep Learning Book [59] は，現在の深層学習のトレンドをおさえた標準的な教科書である．また，古典から既に暗黙知となってしまっている基本的な知見が得られることも多い．このためには，LeCun 氏による Efficient BackProp [26] と Bengio 氏による Practical Recommendations for Gradient-based Training of Deep Architectures [21] を推奨したい．Schmidhuber 氏は深層学習の発展の歴史を整理した文献リスト [95] を作成している．

深層学習と音楽情報処理の関係を扱った他の論文としては，Choi らによる A Tutorial on Deep Learning for Music Information Retrieval [14] がある．言語処理分野では Recent Trends in Deep Learning Based Natural Language Processing [13] などがある．Goldberg 氏による入門記事 [96] も特によくまとまっていて読みやすい．

また，深層学習を個別のタスクに適用する際，深層学習アルゴリズムそのものの知識に加えて，対象とする分野固有の知識（ドメイン知識）を得ることが望ましい．このためには，音楽情報処理分野の権威ある論文やサーベイ論文が参考になる [97], [98], [99]．

6. おわりに

本稿では，深層学習の基本を概説し，音楽・音声・言語・画像情報処理への各種の応用例を紹介した．深層学習の理論的な検証はここ数年で急速に進み，深層学習の効果的な使い方や性能の諸元も少しずつ明らかになってきている．しかし反対に，深層学習が音楽情報処理にもたらす可能性についてはまだ未解明な部分が多くある．たとえば各種のアルゴリズムを詳細な条件で追試すること，提案されたアーキテクチャを細かく改良すること，音声・言語・画像処理分野の最新の研究成果を応用することなど，為すべきことは多岐にわたっている．深層学習による音楽情報処理を志す方々にとって，本稿が何らかの形で参考になれば幸いである．

参考文献

[1] Downie, J. S.: Music Information Retrieval, *Annual Review of Information Science and Technology*, Vol. 37, No. 1, pp. 295–340 (2003).
[2] Casey, M. A., Veltkamp, R. C., Goto, M., Leman, M., Rhodes, C. and Slaney, M.: Content-Based Music Information Retrieval: Current Directions and Future Challenges, *Proceedings of the IEEE*, Vol. 96, No. 4, pp. 668–696 (2008).
[3] Müller, M., Ellis, D. P. W., Klapuri, A. and Richard, G.: Signal Processing for Music Analysis, *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, No. 6, pp. 1088–1110 (2011).
[4] Hadjeres, G., Pachet, F. and Nielsen, F.: DeepBach: a Steerable Model for Bach Chorales Generation, *International Conference on Machine Learning (ICML)*, pp. 1362–1371 (2017).
[5] Korzeniowski, F. and Widmer, G.: Feature Learning for Chord Recognition: The Deep Chroma Extractor, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 37–43 (2016).
[6] Böck, S., Krebs, F. and Widmer, G.: Joint Beat and Downbeat Tracking with Recurrent Neural Networks, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 255–261 (2016).
[7] Yoshii, K., Goto, M., Komatani, K., Ogata, T. and Okuno, H. G.: Drumix: An Audio Player with Real-Time Drum-Part Rearrangement Functions for Active Music Listening, *Information and Media Technologies*, Vol. 2, No. 2, pp. 601–611 (2007).
[8] Blaauw, M. and Bonada, J.: A Neural Parametric Singing Synthesizer Modeling Timbre and Expression from Natural Songs, *Applied Sciences*, Vol. 7, No. 12 (2017).
[9] Nakano, T. and Goto, M.: VocaListener2: A Singing Synthesis System Able to Mimic a User's Singing in terms of Voice Timbre Changes as well as Pitch and Dynamics, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 453–456 (2011).
[10] Goto, M., Yoshii, K., Fujihara, H., Mauch, M. and Nakano, T.: Songle: A Web Service for Active Music Listening Improved by User Contributions, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 311–316 (2011).
[11] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems (NIPS)*, pp. 1106–1114 (2012).
[12] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. and Dean, J.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *arXiv preprint arXiv:1609.08144* (2016).
[13] Young, T., Hazarika, D., Poria, S. and Cambria, E.: Recent Trends in Deep Learning Based Natural Language Processing, *arXiv preprint arXiv:1708.02709* (2017).
[14] Choi, K., Fazekas, G., Cho, K. and Sandler, M. B.: A Tutorial on Deep Learning for Music Information Retrieval, *arXiv preprint arXiv:1709.04396* (2017).
[15] Yosinski, J., Clune, J., Bengio, Y. and Lipson, H.: How Transferable are Features in Deep Neural Networks?, *Advances in Neural Information Processing Systems (NIPS)*, pp. 3320–3328 (2014).
[16] Kajihara, Y., Dozono, S. and Tokui, N.: Imaginary Soundscape: Cross-Modal Approach to Generate Pseudo Sound Environments, *Workshop on Machine Learning for Creativity and Design at Neural Information Processing Systems*, pp. 1–3 (2017).
[17] Pascanu, R., Montúfar, G. and Bengio, Y.: On the Number of Response Regions of Deep Feedforward Networks with Piecewise Linear Activations, *arXiv preprint arXiv:1312.6098* (2013).
[18] Glorot, X., Bordes, A. and Bengio, Y.: Deep Sparse Rectifier Neural Networks, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 315–323 (2011).
[19] Maas, A. L., Hannun, A. Y. and Ng, A. Y.: Rectifier Nonlinearities Improve Neural Network Acoustic Models, *ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (2013).
[20] Clevert, D.-A., Unterthiner, T. and Hochreiter, S.: Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), *International Conference on Learning Representations (ICLR)*, pp. 1–14 (2016).
[21] Bengio, Y.: Practical Recommendations for Gradient-based Training of Deep Architectures, *Neural Networks: Tricks of the Trade: Second Edition*, pp. 437–478 (2012).
[22] Zagoruyko, S. and Komodakis, N.: Wide Residual Networks, *British Machine Vision Conference (BMVC)*, pp. 1–15 (2016).
[23] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W. and Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio, *arXiv preprint arXiv:1609.03499* (2016).
[24] Rifai, S., Dauphin, Y. N., Vincent, P., Bengio, Y. and Muller, X.: The Manifold Tangent Classifier, *Advances in Neural Information Processing Systems (NIPS)*, pp. 2294–2302 (2011).
[25] McFee, B. and Bello, J. P.: Structured Training for Large-Vocabulary Chord Recognition, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 188–194 (2017).
[26] LeCun, Y., Bottou, L., Orr, G. B. and Müller, K.: Efficient BackProp, *Neural Networks: Tricks of the Trade*, pp. 9–50 (1998).
[27] Glorot, X. and Bengio, Y.: Understanding the Difficulty of Training Deep Feedforward Neural Networks, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 249–256 (2010).
[28] He, K., Zhang, X., Ren, S. and Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, *International Conference on Computer Vision (ICCV)*, pp. 1026–1034 (2015).
[29] Mishkin, D. and Matas, J.: All You Need is a Good Init, *International Conference on Learning Representations (ICLR)*, pp. 1–13 (2016).
[30] Pascanu, R., Mikolov, T. and Bengio, Y.: On the Difficulty of Training Recurrent Neural Networks, *International Conference on Machine Learning (ICML)*, Vol. 28, No. 3, pp. 1310–1318 (2013).
[31] Gers, F. A., Schmidhuber, J. and Cummins, F. A.: Learning to Forget: Continual Prediction with LSTM, *Neural Computation*, Vol. 12, No. 10, pp. 2451–2471 (2000).

[32] Jozefowicz, R., Zaremba, W. and Sutskever, I.: An Empirical Exploration of Recurrent Network Architectures, *International Conference on Machine Learning (ICML)*, pp. 2342–2350 (2015).
[33] Ruder, S.: An Overview of Gradient Descent Optimization Algorithms, *arXiv preprint arXiv:1609.04747* (2016).
[34] Qian, N.: On the Momentum Term in Gradient Descent Learning Algorithms, *Neural Networks*, Vol. 12, No. 1, pp. 145–151 (1999).
[35] Duchi, J. C., Hazan, E. and Singer, Y.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, *Journal of Machine Learning Research*, Vol. 12, pp. 2121–2159 (2011).
[36] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *arXiv preprint arXiv:1412.6980* (2014).
[37] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).
[38] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734 (2014).
[39] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. and Schmidhuber, J.: LSTM: A Search Space Odyssey, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 10, pp. 2222–2232 (2017).
[40] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *Advances in Neural Information Processing Systems (NIPS)*, pp. 3104–3112 (2014).
[41] Chung, J., Gülçehre, Ç., Cho, K. and Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, *arXiv preprint arXiv:1412.3555* (2014).
[42] Karpathy, A., Johnson, J. and Li, F.: Visualizing and Understanding Recurrent Networks, *arXiv preprint arXiv:1506.02078* (2015).
[43] Zeiler, M. D. and Fergus, R.: Visualizing and Understanding Convolutional Networks, *European Conference on Computer Vision (ECCV)*, pp. 818–833 (2014).
[44] Bittner, R. M., McFee, B., Salamon, J., Li, P. and Bello, J. P.: Deep Salience Representations for F0 Estimation in Polyphonic Music, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 63–70 (2017).
[45] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016).
[46] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T.: Recent Advances in Convolutional Neural Networks, *Pattern Recognition*, Vol. 77, pp. 354–377 (2018).
[47] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*, Vol. 15, pp. 1929–1958 (2014).
[48] Arpit, D., Jastrzebski, S. K., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A. C., Bengio, Y. and Lacoste-Julien, S.: A Closer Look at Memorization in Deep Networks, *International Conference on Machine Learning (ICML)*, pp. 233–242 (2017).
[49] Cybenko, G.: Approximation by Superpositions of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*, Vol. 2, No. 4, pp. 303–314 (1989).
[50] Hornik, K.: Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks*, Vol. 4, No. 2, pp. 251–257 (1991).
[51] Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O.: Understanding Deep Learning Requires Rethinking Generalization, *International Conference on Learning Representations (ICLR)*, pp. 1–15 (2017).
[52] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *International Conference on Machine Learning (ICML)*, Vol. 37, pp. 448–456 (2015).
[53] Hoffer, E., Banner, R., Golan, I. and Soudry, D.: Norm Matters: Efficient and Accurate Normalization Schemes in Deep Networks, *arXiv preprint arXiv:1803.01814* (2018).
[54] Gal, Y. and Ghahramani, Z.: A Theoretically Grounded Application of Dropout in Recurrent Neural Networks, *Advances in Neural Information Processing Systems (NIPS)*, pp. 1019–1027 (2016).
[55] Wan, L., Zeiler, M., Zhang, S., Cun, Y. L. and Fergus, R.: Regularization of Neural Networks using DropConnect, *International Conference on Machine Learning (ICML)*, Vol. 28, No. 3, pp. 1058–1066 (2013).
[56] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A. C. and Bengio, Y.: Maxout Networks, *International Conference on Machine Learning (ICML)*, pp. 1319–1327 (2013).
[57] Ba, L. J., Kiros, R. and Hinton, G. E.: Layer Normalization, *arXiv preprint arXiv:1607.06450* (2016).
[58] Krueger, D., Maharaj, T., Kramar, J., Pezeshki, M., Ballas, N., Ke, N. R., Goyal, A., Bengio, Y., Courville, A. and Pal, C.: Zoneout: Regularizing RNNs by Randomly Preserving Hidden Activations, *International Conference on Learning Representations (ICLR)*, pp. 1–11 (2017).
[59] Goodfellow, I., Bengio, Y. and Courville, A.: *Deep Learning*, MIT Press (2016).
[60] Pan, S. J. and Yang, Q.: A Survey on Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345–1359 (2010).
[61] Bojanowski, P. and Joulin, A.: Unsupervised Learning by Predicting Noise, *International Conference on Machine Learning (ICML)*, pp. 517–526 (2017).
[62] Choi, K., Fazekas, G., Sandler, M. B. and Cho, K.: Transfer Learning for Music Classification and Regression Tasks, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 141–149 (2017).
[63] Kingma, D. P. and Welling, M.: Auto-Encoding Variational Bayes, *arXiv preprint arXiv:1312.6114* (2013).
[64] Doersch, C.: Tutorial on Variational Autoencoders, *arXiv preprint arXiv:1606.05908* (2016).
[65] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative Adversarial Nets, *Advances in Neural Information Processing Systems (NIPS)*, pp. 2672–2680 (2014).
[66] Bengio, Y., Courville, A. C. and Vincent, P.: Representation Learning: A Review and New Perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1798–1828 (2013).
[67] Schulman, J., Heess, N., Weber, T. and Abbeel, P.: Gra-

dient Estimation using Stochastic Computation Graphs, *Advances in Neural Information Processing Systems (NIPS)*, pp. 3528–3536 (2015).

[68] Goodfellow, I. J.: NIPS 2016 Tutorial: Generative Adversarial Networks, *arXiv preprint arXiv:1701.00160* (2016).

[69] Radford, A., Metz, L. and Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, *International Conference on Learning Representations (ICLR)*, pp. 1–16 (2016).

[70] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. and Metaxas, D. N.: StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks, *arXiv preprint arXiv:1710.10916* (2017).

[71] Arjovsky, M., Chintala, S. and Bottou, L.: Wasserstein Generative Adversarial Networks, *International Conference on Machine Learning (ICML)*, Vol. 70, pp. 214–223 (2017).

[72] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A. C.: Improved Training of Wasserstein GANs, *Advances in Neural Information Processing Systems (NIPS)*, pp. 5767–5777 (2017).

[73] Peyré, G. and Cuturi, M.: Computational Optimal Transport, *arXiv preprint arXiv:1803.00567* (2018).

[74] Willard, S.: *General Topology*, Addison-Wesley Series in Mathematics, Dover Publications (1970).

[75] D. Aliprantis, C. and C. Border, K.: *Infinite Dimensional Analysis: A Hitchhiker's Guide*, Springer-Verlag, third edition (2006).

[76] Lucic, M., Kurach, K., Michalski, M., Gelly, S. and Bousquet, O.: Are GANs Created Equal? A Large-Scale Study, *arXiv preprint arXiv:1711.10337* (2017).

[77] Smith, S. L., Kindermans, P.-J. and Le, Q. V.: Don't Decay the Learning Rate, Increase the Batch Size, *International Conference on Learning Representations (ICLR)*, pp. 1–11 (2018).

[78] Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y.: Spectral Normalization for Generative Adversarial Networks, *International Conference on Learning Representations (ICLR)*, pp. 1–26 (2018).

[79] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going Deeper with Convolutions, *Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9 (2015).

[80] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv preprint arXiv:1409.1556* (2014).

[81] Gatys, L. A., Ecker, A. S. and Bethge, M.: Image Style Transfer Using Convolutional Neural Networks, *Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423 (2016).

[82] Li, Y., Wang, N., Liu, J. and Hou, X.: Demystifying Neural Style Transfer, *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2230–2236 (2017).

[83] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *International Conference on Learning Representations (ICLR)*, pp. 1–15 (2015).

[84] Luong, T., Pham, H. and Manning, C. D.: Effective Approaches to Attention-based Neural Machine Translation, *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1412–1421 (2015).

[85] Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S. and Kavukcuoglu, K.: Efficient Neural Audio Synthesis, *arXiv preprint arXiv:1802.08435* (2018).

[86] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y. and Wu, Y.: Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP), to appear* (2018).

[87] Oyamada, K., Kameoka, H., Kaneko, T., Tanaka, K., Hojo, N. and Ando, H.: Generative Adversarial Network-based Approach to Signal Reconstruction from Magnitude Spectrograms, *arXiv preprint arXiv:1804.02181* (2018).

[88] Vinyals, O., Toshev, A., Bengio, S. and Erhan, D.: Show and Tell: A Neural Image Caption Generator, *Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164 (2015).

[89] Boulanger-Lewandowski, N., Bengio, Y. and Vincent, P.: Audio Chord Recognition with Recurrent Neural Networks, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 335–340 (2013).

[90] Vogl, R., Dorfer, M., Widmer, G. and Knees, P.: Drum Transcription via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 150–157 (2017).

[91] Jansson, A., Humphrey, E. J., Montecchio, N., Bittner, R. M., Kumar, A. and Weyde, T.: Singing Voice Separation with Deep U-Net Convolutional Networks, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 745–751 (2017).

[92] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D. and Simonyan, K.: Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders, *International Conference on Machine Learning (ICML)*, pp. 1068–1077 (2017).

[93] Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J., Oore, S. and Eck, D.: Onsets and Frames: Dual-Objective Piano Transcription, *arXiv preprint arXiv:1710.11153* (2017).

[94] Roberts, A., Engel, J., Raffel, C., Hawthorne, C. and Eck, D.: A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music, *arXiv preprint arXiv:1803.05428* (2018).

[95] Schmidhuber, J.: Deep Learning in Neural Networks: An Overview, *Neural Networks*, Vol. 61, pp. 85–117 (2015).

[96] Goldberg, Y.: A Primer on Neural Network Models for Natural Language Processing, *Journal of Artificial Intelligence Research*, Vol. 57, pp. 345–420 (2016).

[97] Klapuri, A.: Multipitch Analysis of Polyphonic Music and Speech Signals Using an Auditory Model, *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 2, pp. 255–266 (2008).

[98] McVicar, M., Santos-Rodriguez, R., Ni, Y. and Bie, T. D.: Automatic Chord Estimation from Audio: A Review of the State of the Art, *IEEE/ACM Transactions on Audio, Speech and Language Processing*, Vol. 22, No. 2, pp. 556–575 (2014).

[99] Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J. J., Speck, J. A. and Turnbull, D.: Music Emotion Recognition: A State of the Art Review, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 255–266 (2010).