JDLA E資格認定プログラム

全人類がわかるE資格コース

音声処理分野

目次

1）音声処理について

2）WaveNetとは

3）WaveNetの応用

【Chapter12】音声処理分野

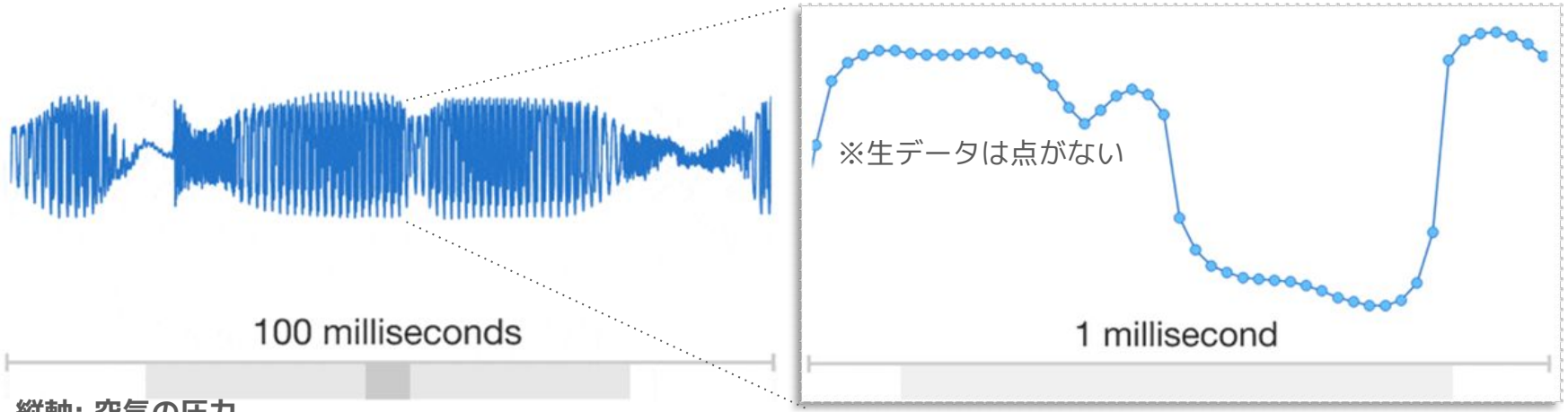
音声処理について

- **音声データ**
  - **WaveNetにおける前処理**
  - フーリエ変換
- 音声処理とは
- 音声処理分野のタスク

音声データ

音声データは波形である

すなわち、アナログ情報（時系列的に連続的な変化を伴う情報）である

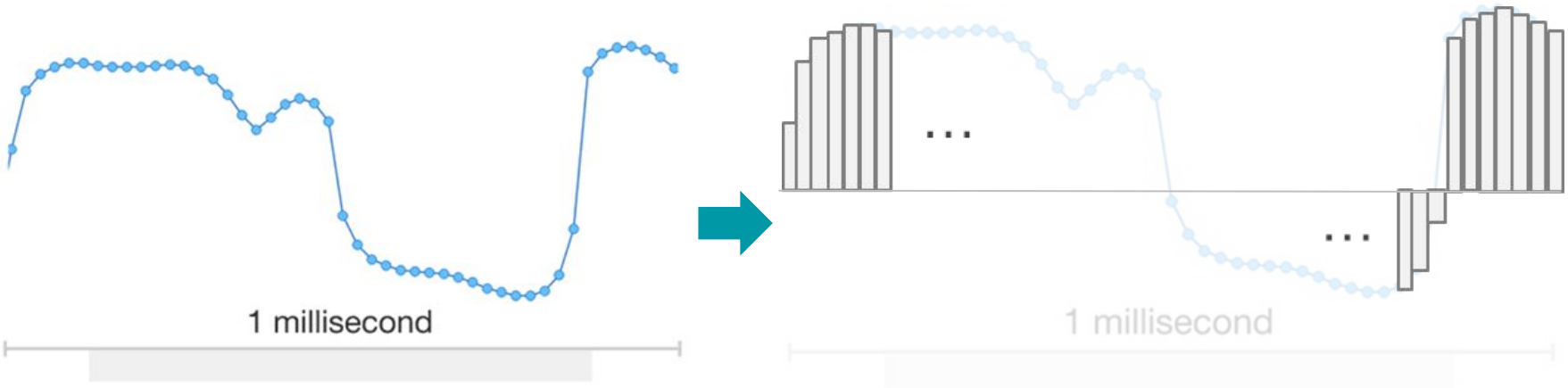


縦軸: 空気の圧力
※実際にはマイクなどを使って空気の圧力を電圧へ変換した値
横軸: 時間

音声データ

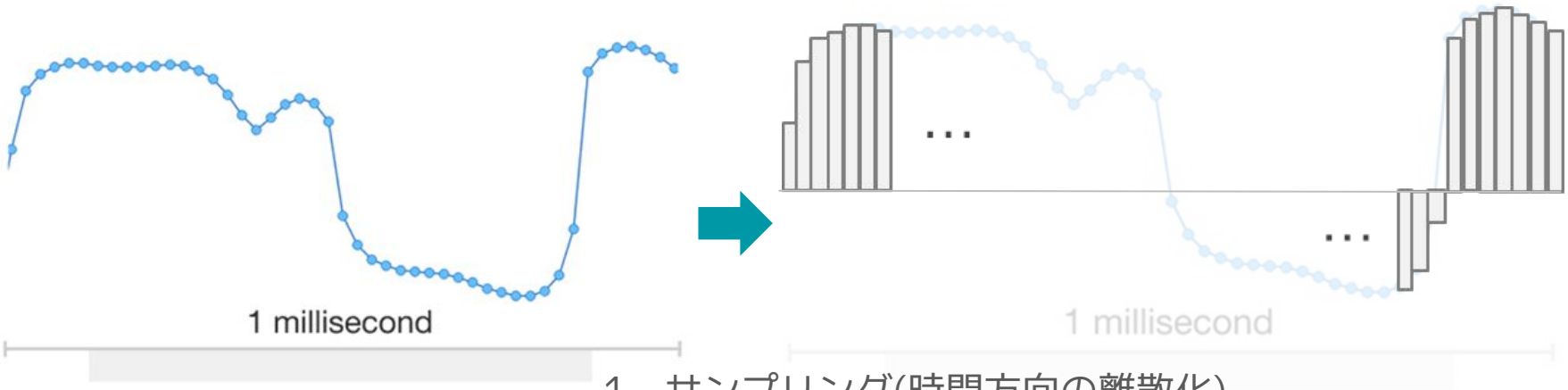
コンピュータが処理しやすいデジタル信号へ変換する必要がある

時系列的に離散的な値



音声データ前処理

アナログからデジタルへの変換プロセス



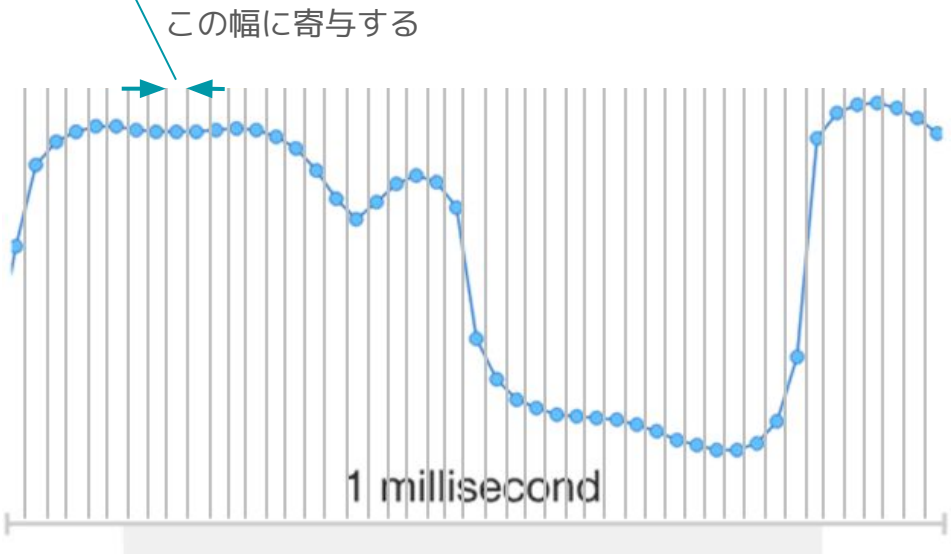
1. サンプリング(時間方向の離散化)
2. 量子化(電圧方向の離散化)
3. 符号化

サンプリングとは、時間方向に離散化すること

サンプリングレート
…1s間にどれほどの標本を取得するか

例) サンプリングレート=8000Hz
1s間で均等に8000本の標本を取得してきている

サンプリングレートの例
- 電話：8,000Hz
- CD：44,100Hz



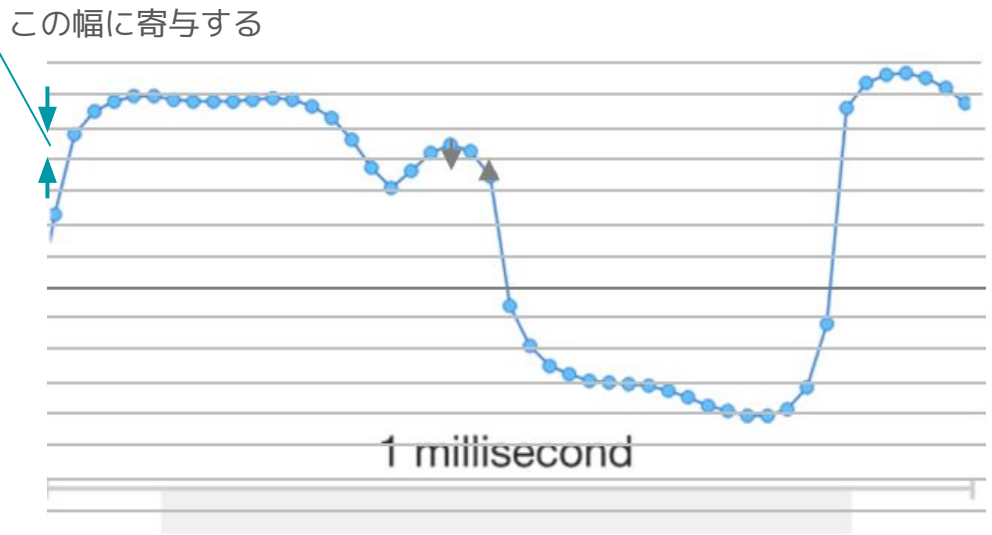
この処理で点がプロットされる

量子化とは、電圧方向に離散化すること

量子化ビット数
…何段階の数値で区切るか

例) 量子化ビット数=4bit
2^4 =16段階で区切る

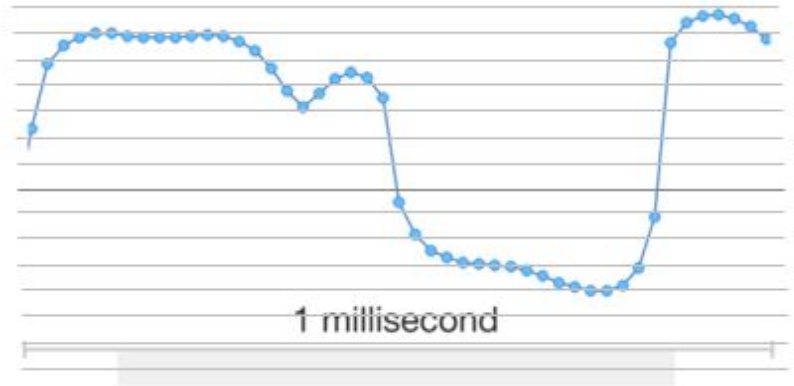
量子化の実例
- 電話: 8ビット(2^8=256段階)
- CD: 16ビット(2^{16}=65,536段階)



各点は一番近い線の値となる

量子化には2種類ある

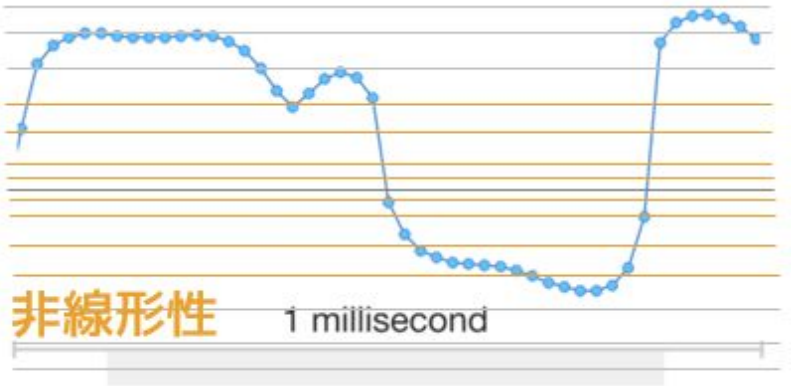
線形量子化
等間隔で量子化



例) CD: 16bit線形量子化

【狙い】0付近を細かく量子化すれば
量子化による誤差が減らせる

非線形量子化
非線形的な変換を施した量子化



例) VoIP: 8bit非線形量子化
WaveNetの前処理で行われている

非線形量子化の例: μ-law量子化

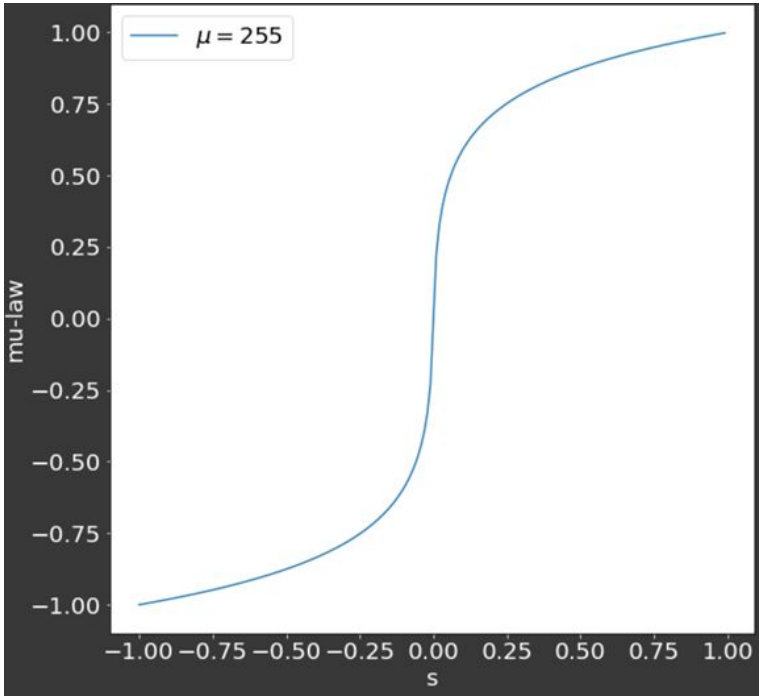
WaveNetの前処理で使われている非線形量子化

$$f(s) = \mathrm{sgn}(s)\frac{\ln(1+\mu|s|)}{\ln(1+\mu)}$$

ここで、s は**正規化された**標本化値

$$-1 \leq s \leq 1$$

WaveNetでは $\mu = 255$としている



【参考】メル尺度

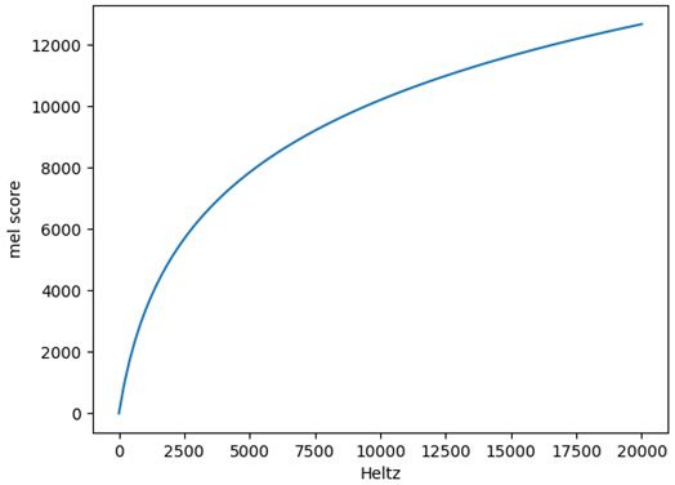
人間の音高知覚を考慮した周波数(1秒間の音波の振動回数)の尺度

周波数f→メル尺度mへの変換式

$$m = m_0 \log(\frac{f}{f_0} + 1)$$

メル尺度m→周波数fへの変換式

$$f = f_0(e^{\frac{m}{m_0}} - 1)$$



※f_0とm_0はパラメータで、 $f_0 = 700$と$m_0 = 2595$がよく使われる

量子化された値を二進数で表す操作

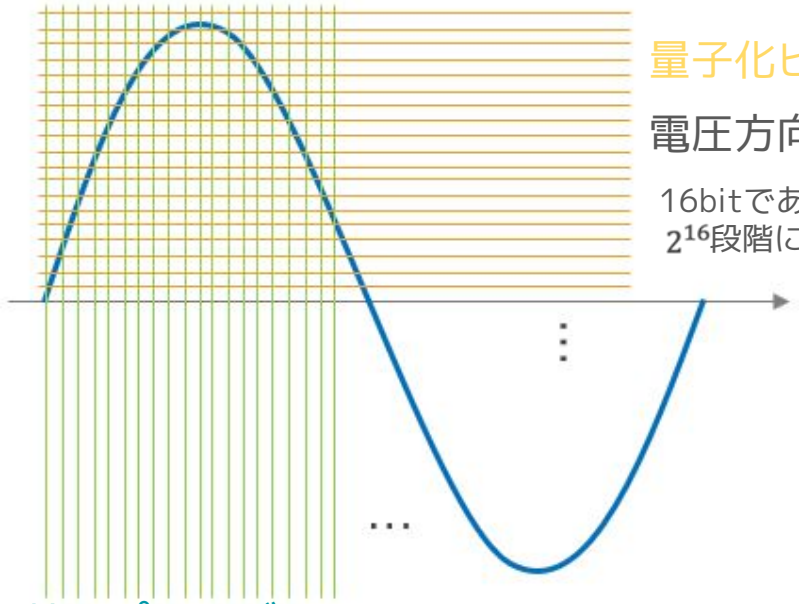
例） 8bit μ-law量子化された値は256種類となる
これを二進数で表す

10進数		符号付き2進数（8bit）
0.25	→	0000.0100
-0.875	→	1111.0010

これによって、256種類の10進数が2進数に符号化される

WaveNet前処理まとめ

波形(アナログ信号)をデジタル信号に変換し、コンピュータ処理しやすくする



量子化ビット数

電圧方向に離散化

16bitであれば
2^{16}段階に分割

サンプリングレート

時間方向に離散化

44.1kHzであれば
1sあたり44,100回の標本化

手順

1.サンプリング

2.量子化

- 線形量子化
- 非線形量子化
  - μ-law量子化

3.符号化

2進数へ

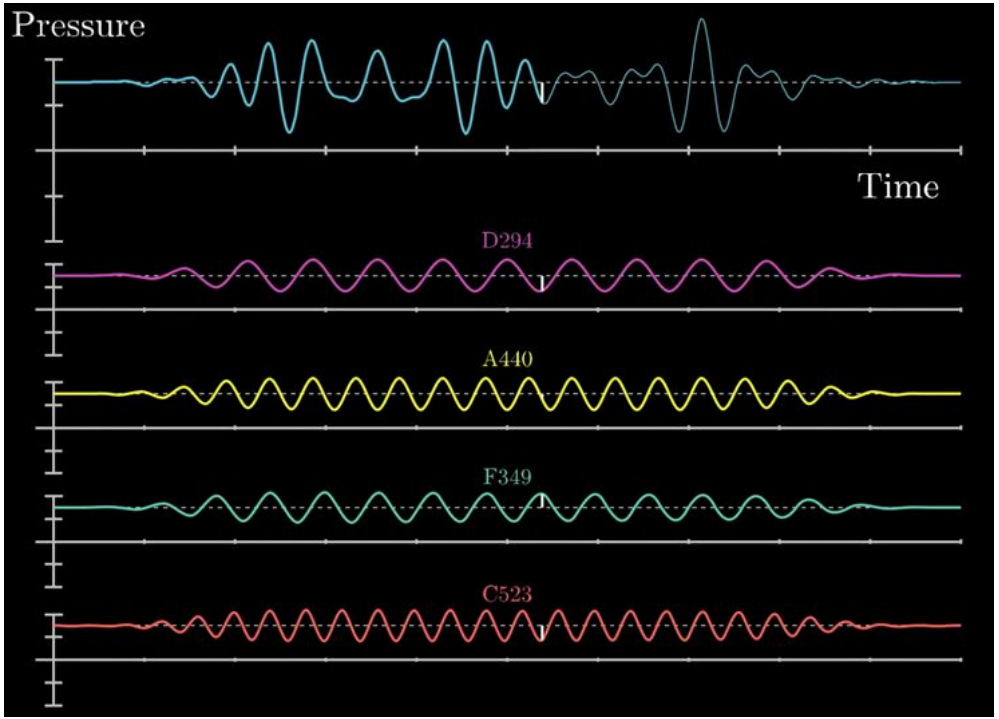
【Chapter12】音声処理分野

音声処理について

- **音声データ**
  - WaveNetにおける前処理
  - **フーリエ変換**
- 音声処理とは
- 音声処理分野のタスク

フーリエ変換(Fourier Transform)

音声データは、それぞれ異なる周波数を持つ正弦波波形に分割できる



＝　紫＋黄＋緑＋赤

294Hzのsin波＊ A_1

＋

440Hzのsin波＊ A_2

＋

349Hzのsin波＊ A_3

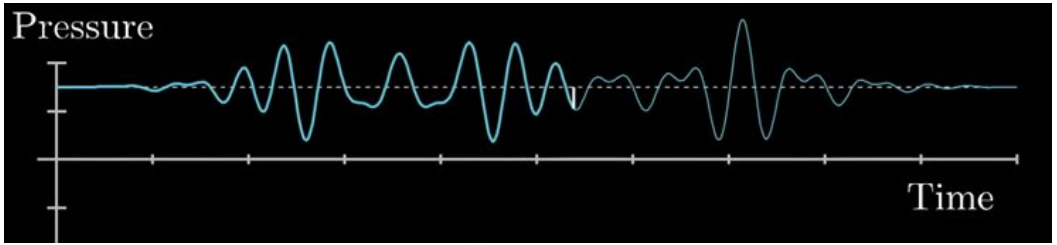
＋

523Hzのsin波＊ A_4

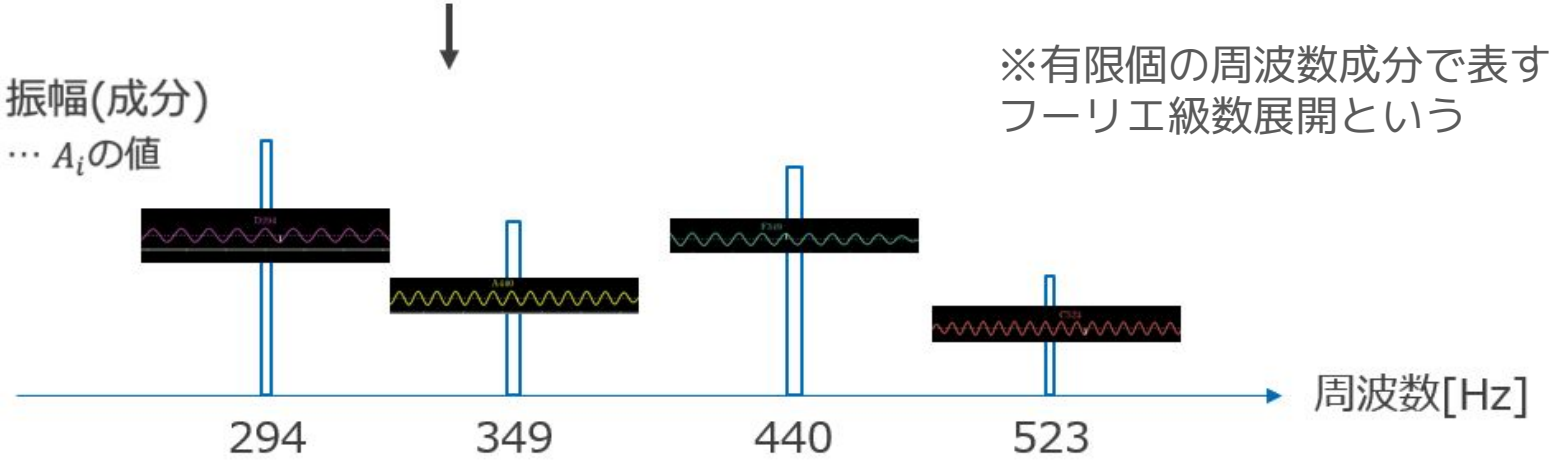
A_i：振幅

フーリエ変換(Fourier Transform)

グラフにすると…



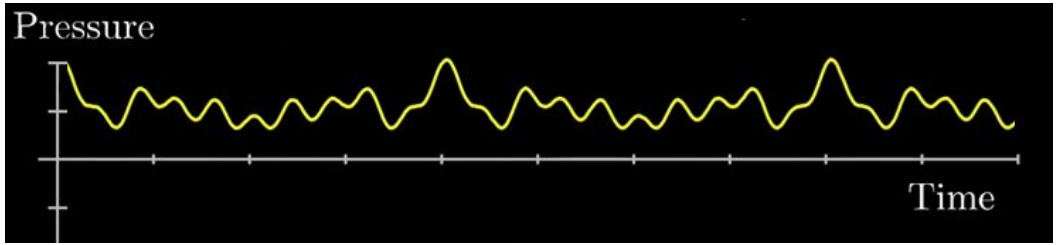
周期性のある波



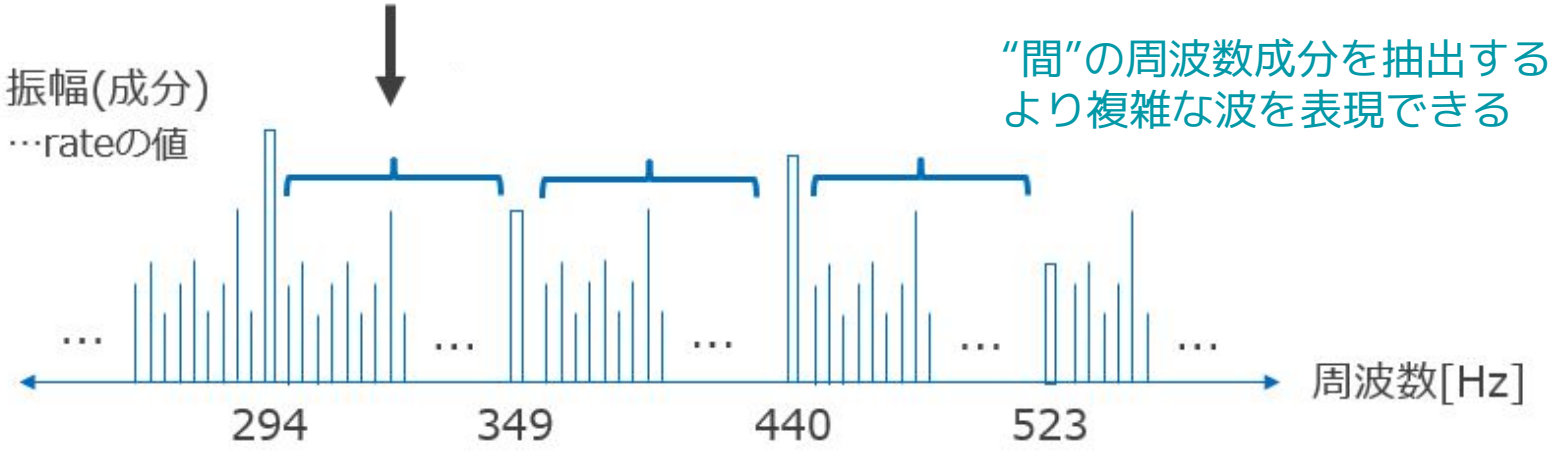
※有限個の周波数成分で表すことを
フーリエ級数展開という

フーリエ変換(Fourier Transform)

フーリエ変換とは、無数の周波数を持つ正弦波波形に分割すること



周期性のない波

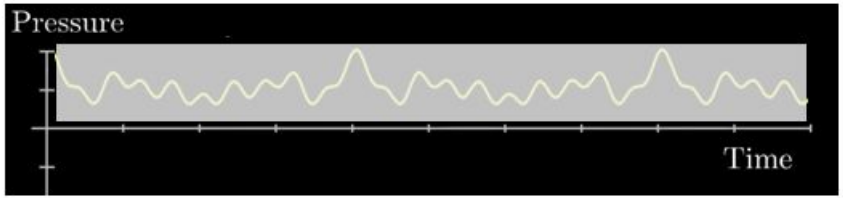


“間”の周波数成分を抽出することで
より複雑な波を表現できる

フーリエ変換の課題

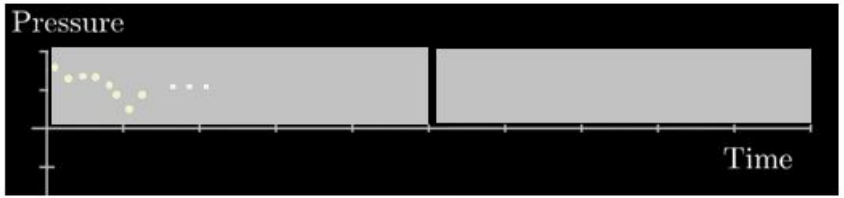
実データを現実的に処理するためには、フーリエ変換は望ましくない

フーリエ変換が適用できる条件



▶被積分関数は連続関数
▶被積分関数は無限時間で定義されている

実処理ではフーリエ変換適用不可

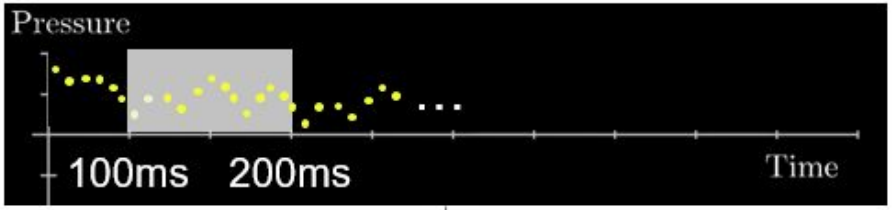


▶音声データは離散信号
▶有限時間のデータしかとれない

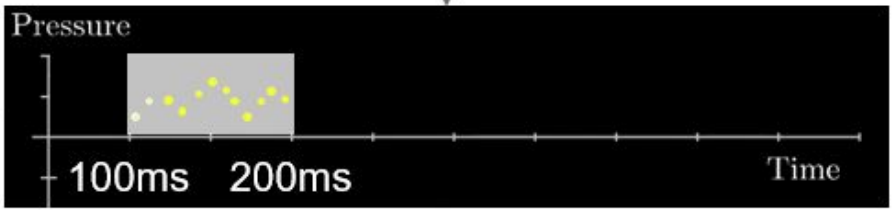
そこで、二段階の手順を踏む
1．窓関数
2．離散フーリエ変換

窓関数とは、音声データを有限時間で抽出する関数

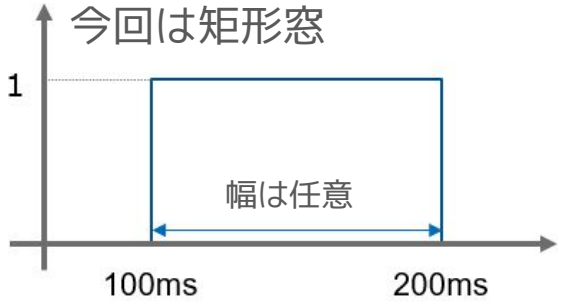
生データ
音声信号



窓関数を
適用した
音声信号



窓関数
多くの種類がある
今回は矩形窓



200ms-300ms, 300-400ms, …の区間で同じことを繰り返して、いくつもデータを作る

手順2. 離散フーリエ変換(Discrete FT)

離散信号を入力とするフーリエ変換

窓関数で切り出した領域からサンプリングした

$$F(t) = \sum_{x=0}^{N-1} f(x) e^{-i\frac{2\pi t x}{N}}$$

- N：データ総数
- t：周波数
- $f(x)$：振幅
- x：データの添え字

これによって、音声データの周波数構造を直接的にモデルへ入力できる

高速フーリエ変換(Fast FT)

高速で離散フーリエ変換を処理するアルゴリズム

【計算容量(乗算回数)】
通常：N^2
高速フーリエ交換：$\frac{N}{2}(\log_2 N - 1)$

複素数の対称性を利用することで、乗算回数を大幅に削減

【適用例】
サンプリングレート: 16kHz
FFTを適用するサンプル数: 1024
このとき、 0Hzから8kHzまでの、1025個の等分点における周波数情報が得られる

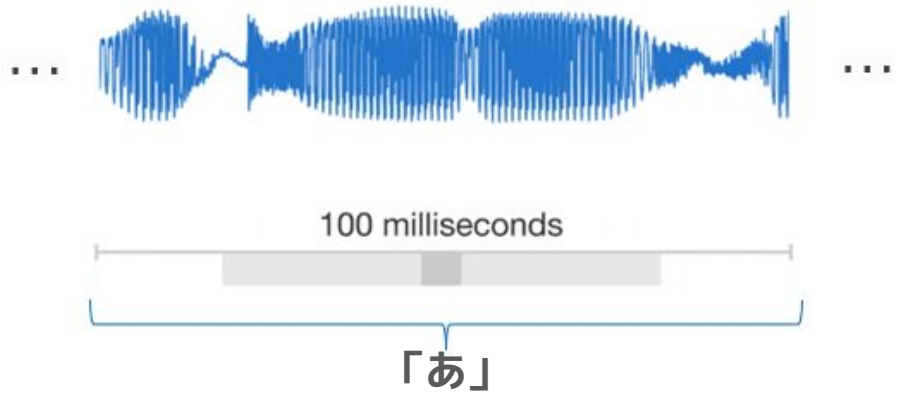
【用途】
機器や機械の異常検出、品質管理、振動観測など

【Chapter12】音声処理分野

CTC

CTC(Connectionist Temporal Classification)

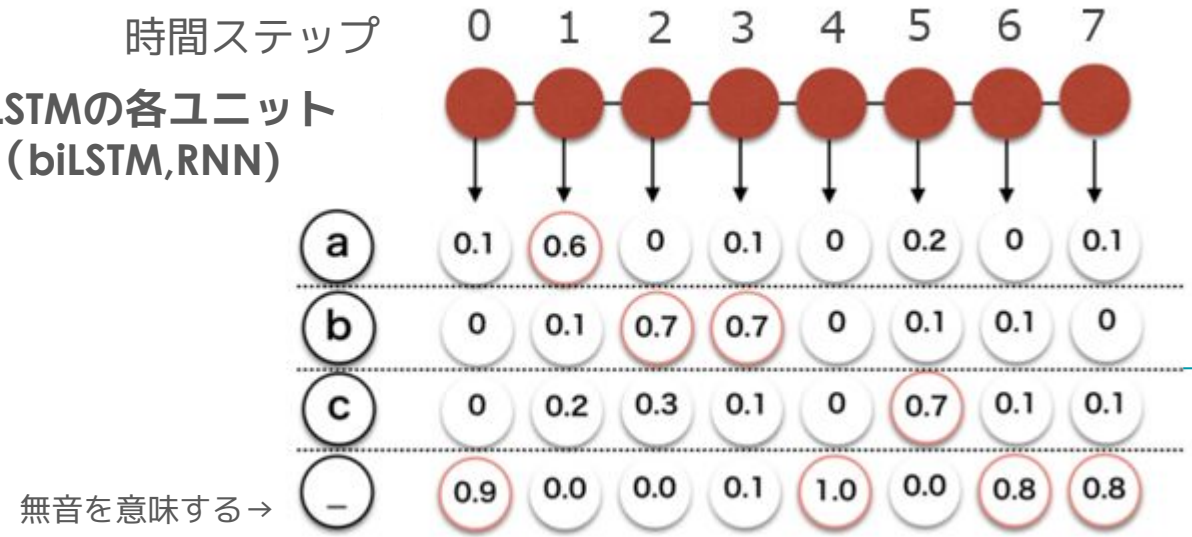
【背景】
音声認識において、入力音声の複数時間フレームが出力文字列の1文字に対応するという、入出力間の時間構造の違いがある。
たとえば「OO:XX:△△から0.1sの間に“あ”と発音した」というアノテーションが必要になる。これは言うまでもなくアノテーションコストが高い。



このようなアノテーションを必要としない学習方法としてCTCがある

CTC概要

【出力例】”_”を使うことで、入出力間の時間構造の違いを緩和



→ _abb_c__

※この文字列の出現確率は、赤丸の値を全てかけたもの

http://www.thothchildren.com/chapter/5c0b599041f88f26724a6d63

【学習時】損失関数の計算方法

ネットワーク出力例
・_abb_c___
・a__bbb_c
・aab____c_

写像B → **abc**

1. 【重複音素の削除】連続した同一音素は、一つの文字にする
2. 【無音の削除】ブランク文字"_"を全て削除する

"abc"の尤度＝上記操作により"abc"となる**すべての**文字列 π の出現確率の和

$$p(\text{“}abc\text{”}) = \sum_{\pi \in \boldsymbol{B}^{-1}(\text{“}abc\text{”})} p(\pi) = \sum_{\pi \in \boldsymbol{B}^{-1}(\text{“}abc\text{”})} \prod_{t=0}^{T-1} y_{\pi_t}^t$$

(文字列abcを認識するための)**損失関数** $Loss = -\log p(\text{“}abc\text{”})$

【学習時】前向き後ろ向きアルゴリズム

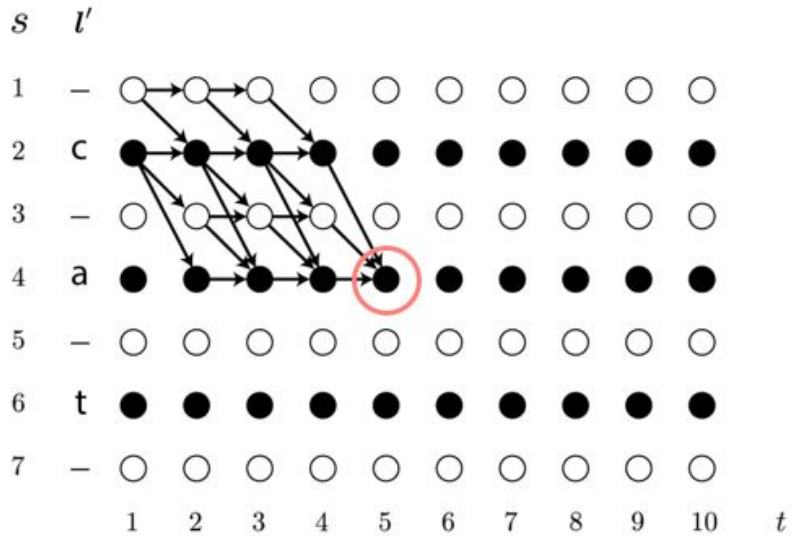
$p(\text{“}abc\text{”})$を計算するために、愚直にすべての$p(\pi)$を計算することは非効率
そこで、動的計画法の一つである**前向き後ろ向きアルゴリズム**を用いる

例)"cat"となるすべての$p(\pi)$の計算方法
手順1．前向き確率を求める
手順2．後ろ向き確率を求める
手順3．前向き確率と後ろ向き確率を用いて、全ての$p(\pi)$を算出する

【学習時】前向き後ろ向きアルゴリズム

前向き確率

たとえば、赤丸に到達するパスは右図のように与えられる。



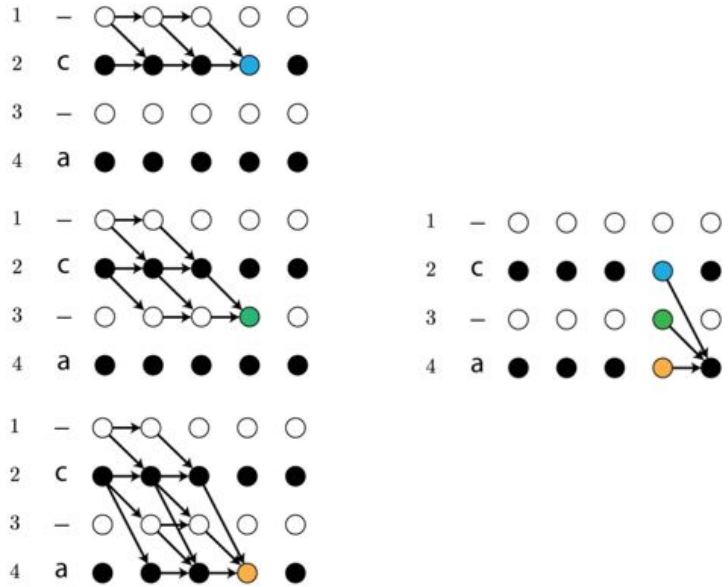
【学習時】前向き後ろ向きアルゴリズム

前向き確率

ここで、赤丸に到達する確率は

左の3パターンの足し算×右の確率

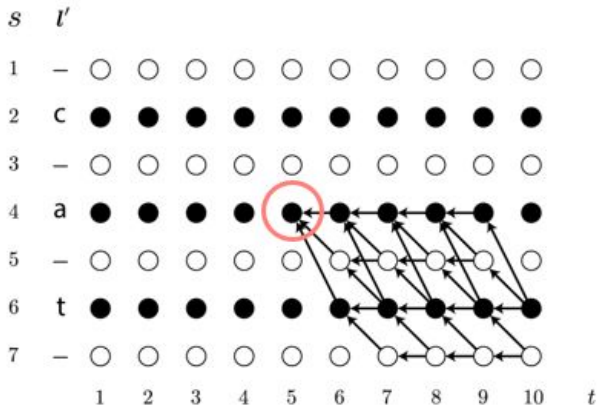
と再帰的に求められる

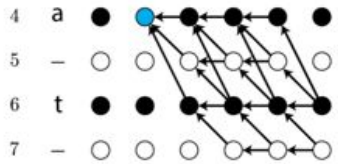


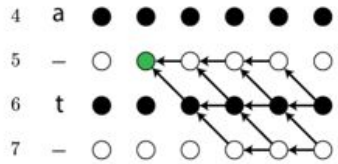
【学習時】前向き後ろ向きアルゴリズム

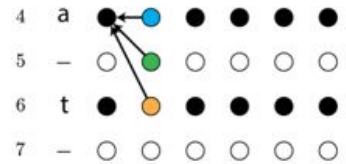
後ろ向き確率

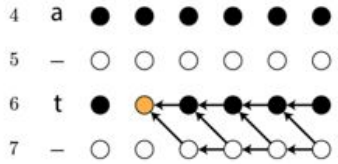
前向き確率と同様に、後ろ向きに再帰的に赤丸の確率を求める









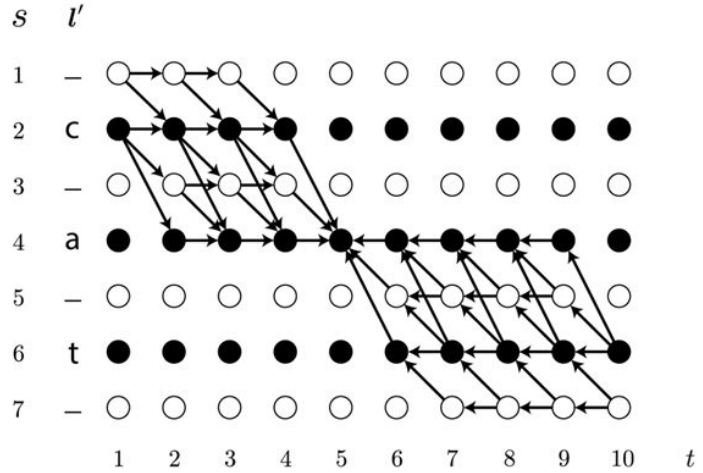


http://musyoku.github.io/2017/06/16/Connectionist-Temporal-Classification/

【学習時】前向き後ろ向きアルゴリズム

“cat”となる全パスの確率

1. 先ほどの前向きと後ろ向きの確率をかける

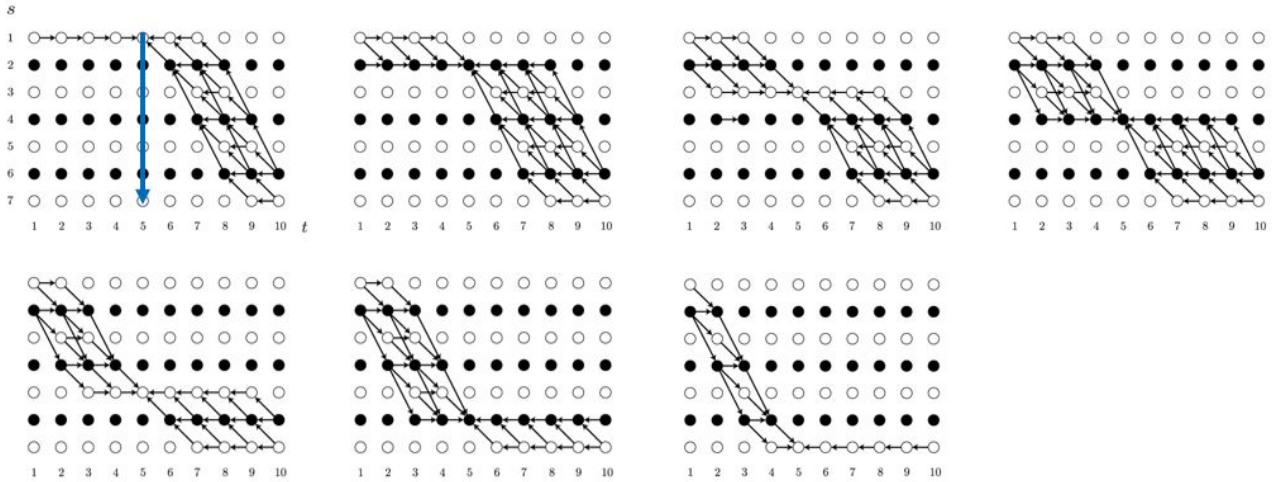


http://musyoku.github.io/2017/06/16/Connectionist-Temporal-Classification/

【学習時】前向き後ろ向きアルゴリズム

“cat”となる全パスの確率

2. t(時間方向)を固定してs(文字方向)を動かすとすべてのパスが網羅される

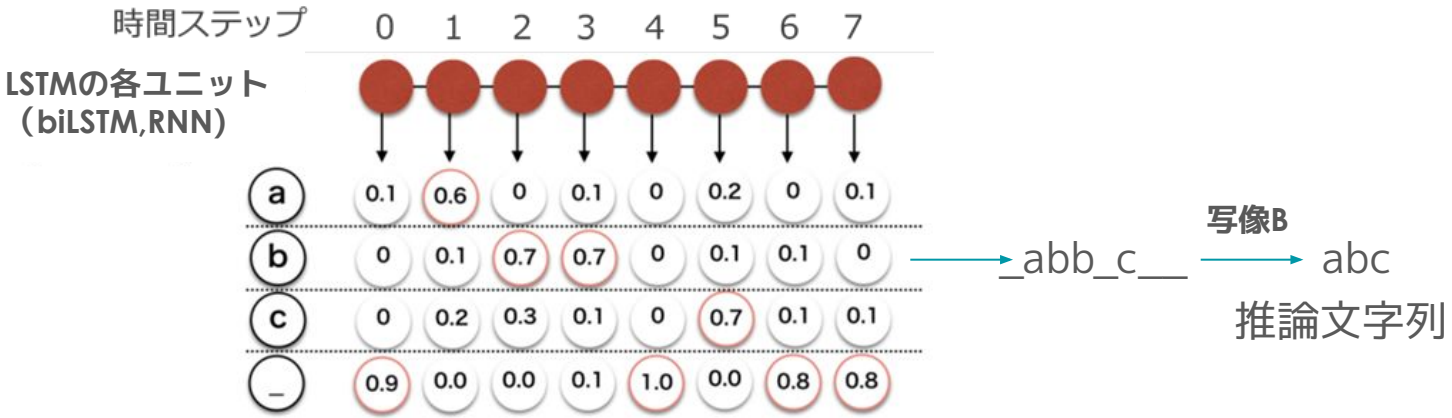


http://musyoku.github.io/2017/06/16/Connectionist-Temporal-Classification/

【推論時】Best Path Decoding

推論時には、正解文字列が与えられないので候補ごと(“abc”など)に尤度を計算して比較するのが自然な発想である。しかし、パス数が多くなると計算量が多くなるため、best path decoding が用いられる。

方法はとてもシンプルで、**各時刻で最も確率の高い文字を採用する**



【Chapter12】音声処理分野

音声処理について

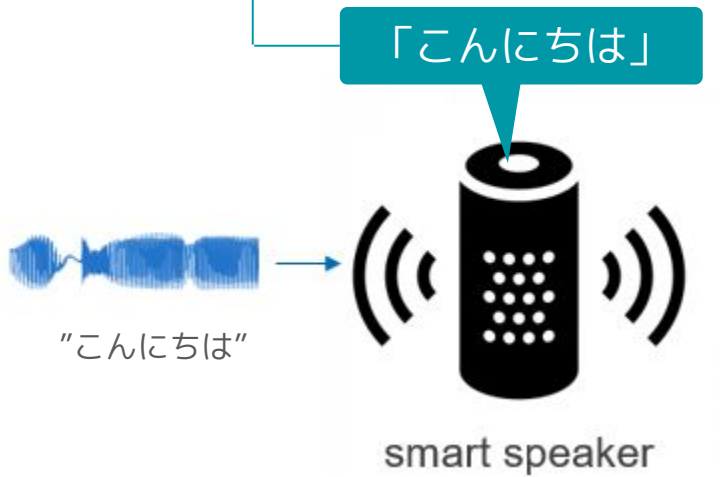
- 音声データ
- **音声処理とは**
- 音声処理分野のタスク

音声処理とは

**音声処理とは、音声データを分析して種々の特徴量を抽出し、
それに基づいて合成や認識などを行うこと**

音声認識

音声信号から言語情報を抽出する技術



音声合成

ある情報を入力として音声波形を合成する技術

テキスト音声合成(後述)の例

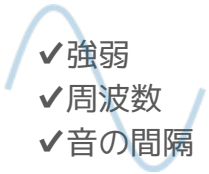
音声認識とは、音声信号から言語情報を抽出する技術

音響分析
音の特徴量を抽出

入力音声

”ありがとう”

特徴量



✔強弱
✔周波数
✔音の間隔

音響モデル
音素マッチング

学習済みモデル

音素

a-r-i-g-a-t-o-u

発音辞書
音の組み合わせをピックアップ

データベース

音の組み合わせ

a-ri-ga-to-u

単語化

言語モデル
直後の単語の出現確率をモデル化

文章=単語の組み合わせ

文章であれば、経験的により流暢な文章として認識する

音声合成とは、ある情報を入力として音声波形を合成する技術

入力の種類によって分類

Text-To-Speech(TTS)

テキストを音声に変換する手法

テキスト音声合成とも呼ばれる

“こんにちは”

音声変換 Voice Conversion

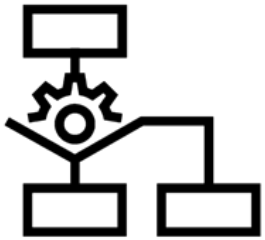
入力音声の性質を変換する手法

歌声に変換

声質変換

例)キャラクターの音声に変える

音声合成の手法



規則合成

音声生成に関する知識をもとに定めたルールに基づいて音声を合成

波形接続型音声合成

録音された音声の素片を連結して音声を合成

統計的パラメトリック音声合成

統計的に学習した生成モデルの出力を元に音声を合成

中でもニューラルネットワーク音声合成は表現力が高い

WaveNetはその一種である

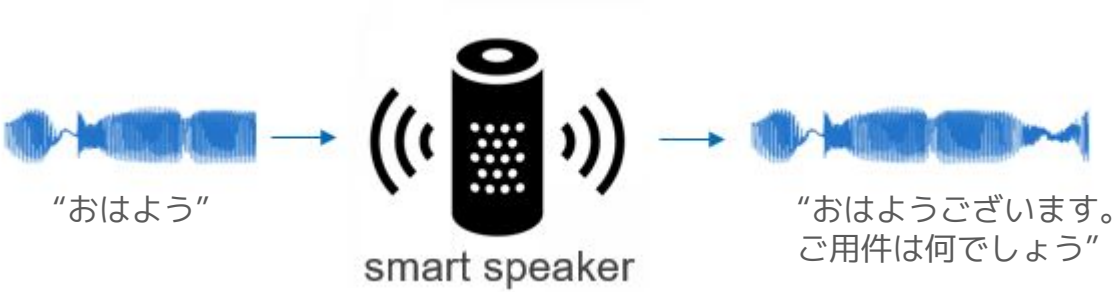
【Chapter12】音声処理分野

音声処理について

- 音声データ
- 音声処理とは
- **音声処理分野のタスク**

音声処理分野のタスク

音声認識と音声合成を組み合わせて、スマートスピーカーのような複雑なシステムができる



WaveNetは3つのタスクで検証されている

✓多話者音声合成

多様性を持つ複数の話者の音声の特徴を学習し、話者のイントネーションなどを再現

✓テキスト音声合成(TTS)

入力テキストから得られた言語的特徴と周波数の特徴を学習し、音声を合成

✓音楽

音楽音声の特徴を学習し、調和的な音声を合成

多話者音声合成

タスク内容

多様性を持つ複数の話者の音声の特徴を学習し、話者のイントネーションなどを再現

データ概要

異なる109人の話者の計44時間分の音声データ
入力: 音声の波形データ; 話者ID
出力: 音声の波形データ

結果・評価

▶人間の言語によく似た単語をリアルで滑らかなイントネーションで生成
▶109人の特徴を１つのモデルで表現

特徴

▶1人の話者よりも人数を増やした方が精度向上
▶音響や録音品質のような話者そのもの以外の情報を取得

テキスト音声合成(TTS)

タスク内容

入力テキストから得られた言語的特徴と基本周波数の特徴を学習し、音声を合成

データ概要

1人の話者の24.6時間分の北米英語と34.8時間分の北京語データ
入力：対数基本周波数 $(\log F_0)$;単語の言語的特徴（音節・発音など）
出力：対数基本周波数 $(\log F_0)$;音節の長さ

結果・評価

▶MOS(“Bad”,…,”Excellent”の5段階スコア(人為的)の平均値)で音声の滑らかさを評価
▶被験者に「2つの音声のどちらが好みか」を選択してもらう評価方法

特徴

▶外部モデルは各言語に対する言語的特徴から対数基本周波数値と電話持続時間を予測
▶各言語において従来の統計的音声合成モデル(LSTMやHMM)を上回る精度

音楽の合成

タスク内容

音楽音声の特徴を学習し、調和的な音声を合成

データ概要

2つの音楽データセット

✓約200時間分の音楽データ

　各29sのクリップに「ジャンル」「楽器」「テンポ」「ムード」など188個のタグ付与

✓YouTubeの動画から得られた約60時間のピアノ独奏曲

結果・評価

▶論文に詳細が記述されていないので割愛

特徴

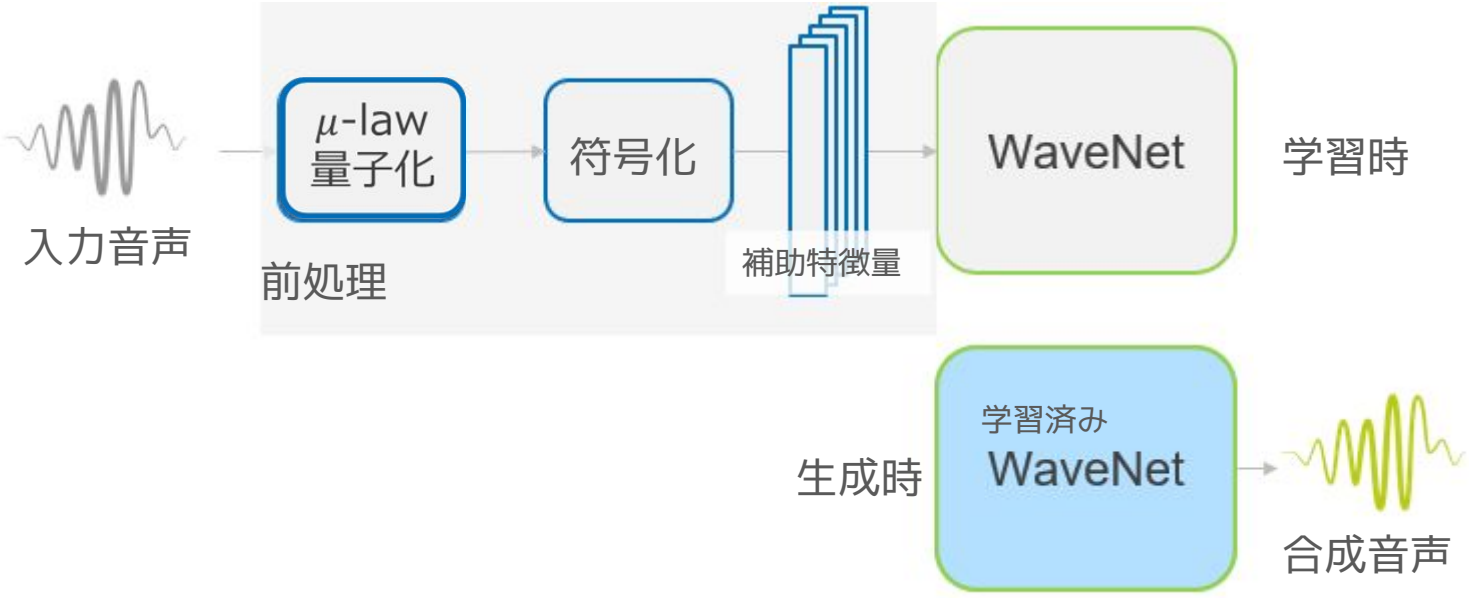
▶条件付き音楽モデルを用いて、ジャンルや楽器など指定されたタグの音楽を生成できる

WaveNetとは

- **WaveNetとは**
- 生成された音声の例

WaveNet 全体像

WaveNetとは、(AlphaGoでお馴染みの)DeepMind社が開発を進めている音声合成ディープニューラルネットワーク



WaveNetの前に… 「補助特徴量」はどのような形式なのか

WaveNet 前処理

デジタル変換

NNの入力用変換

入力音声

μ-law 量子化

符号化

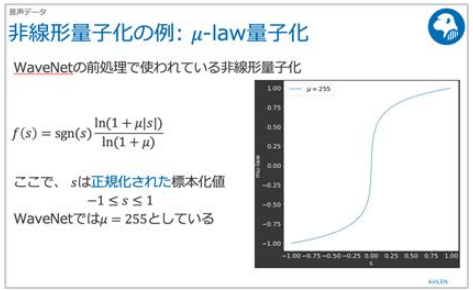
WaveNetへ

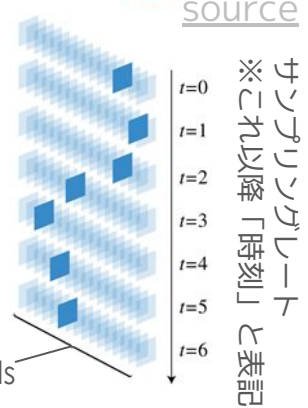
補助特徴量

✓サンプリングレート
16kHz（16000samples/s）
✓情報量
16bit(2^{16}段階)

✓サンプリングレート
16kHz（16000samples/s）
✓情報量
8bit($2^8 = 256$ 段階)

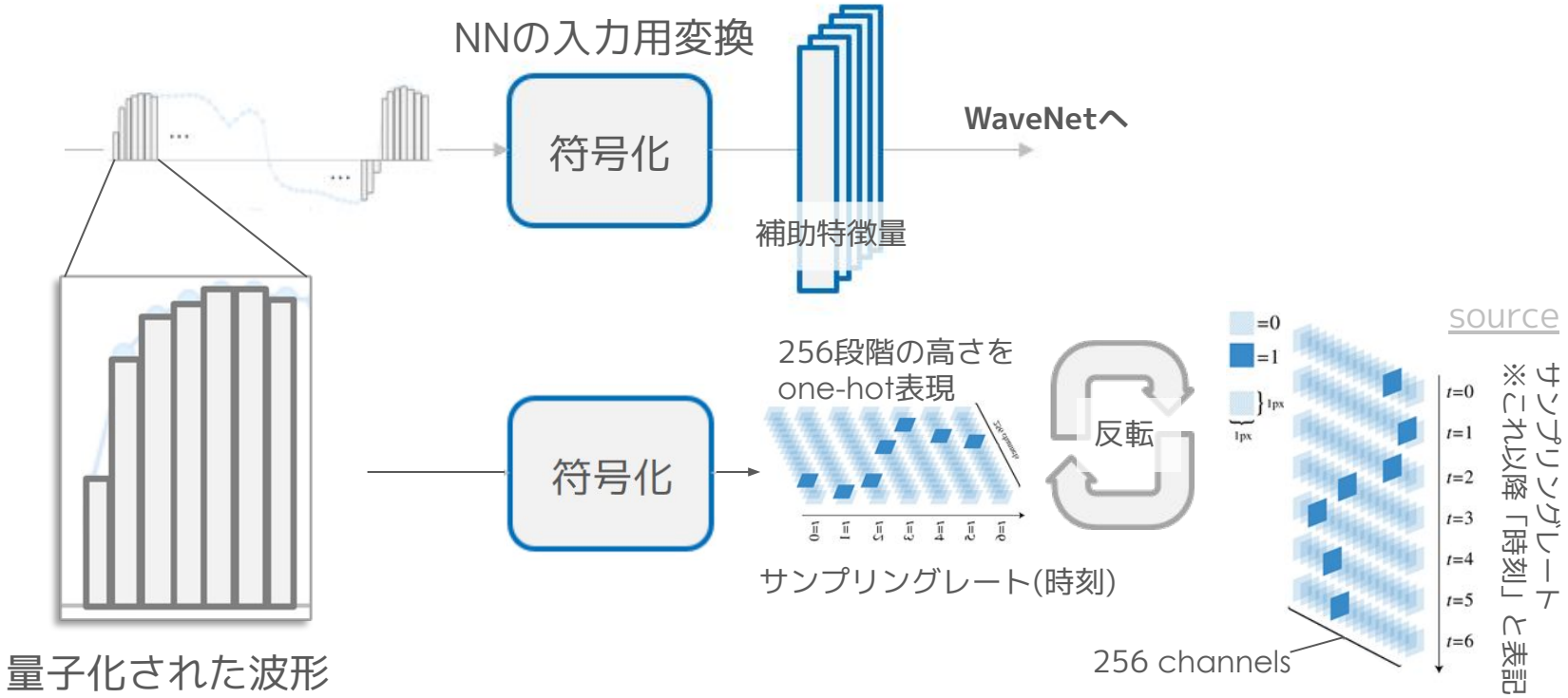
source





(1, サンプリングレート, チャンネル数)

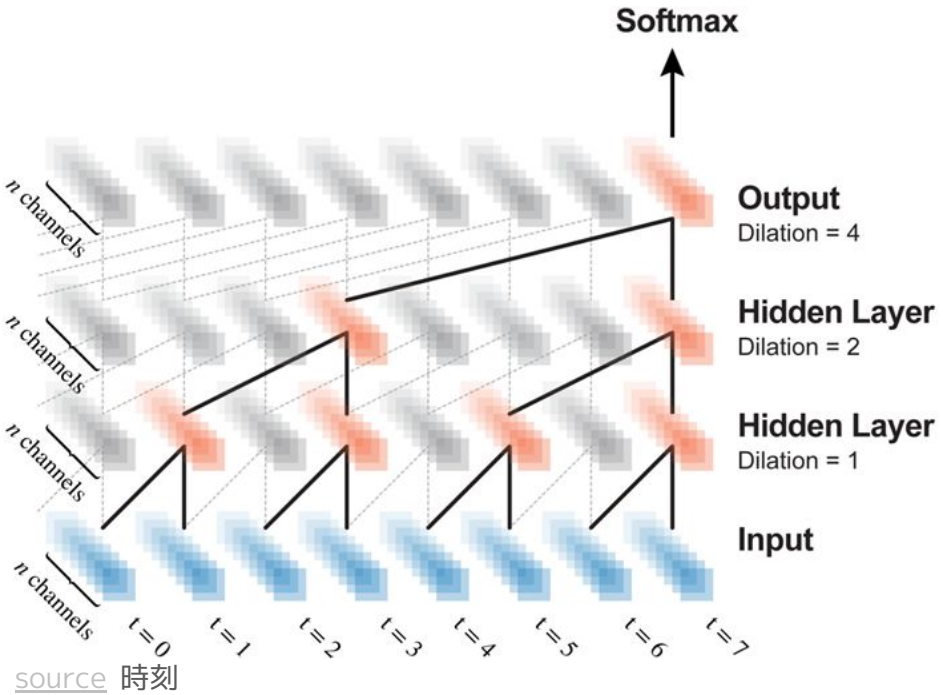
WaveNet 前処理



いよいよWaveNetのアーキテクチャへ

WaveNetとは

CNNを用いた自己回帰モデルの応用



source

モデルを表す数式

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_1, \cdots, x_{t-1})$$

x_tの出現確率は、x_{t-1}までの情報に基づいて定まる

基本情報

▶プーリング層はない

▶$-\log p(x)$を最小化するように学習

重要なテクニック

✓Dilated Causal Convolutions

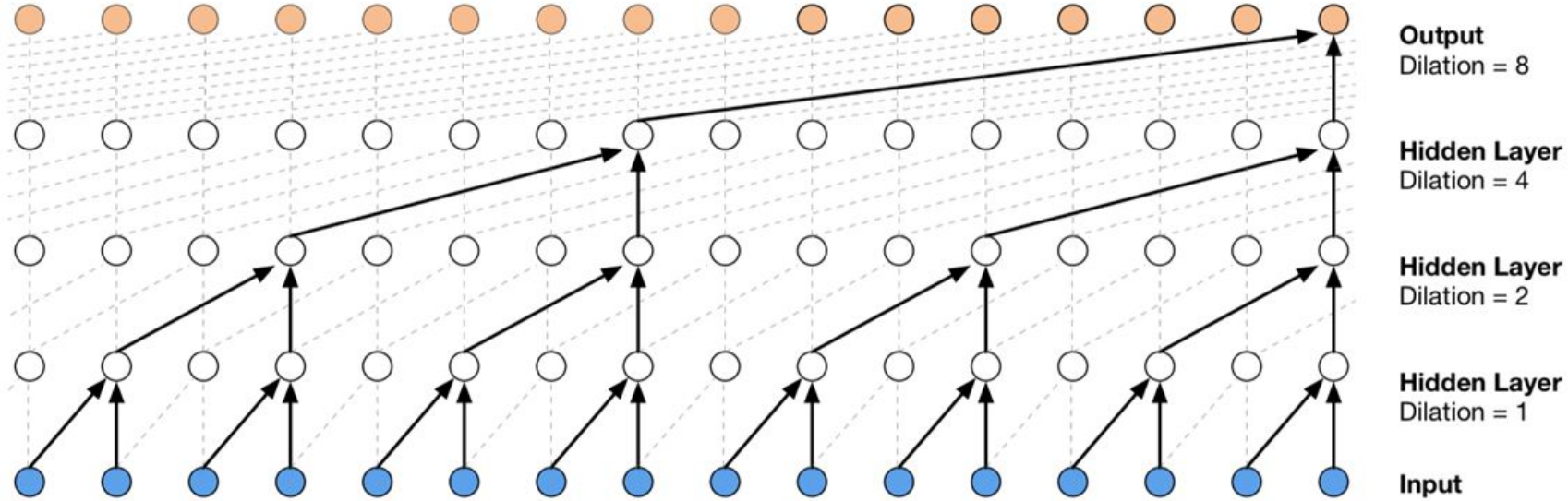
✓Residual & Skip Connections

✓Gated Activation Units

Dilated Causal Convolutions

過去のみの情報に基づいて、飛び飛びで畳み込む処理

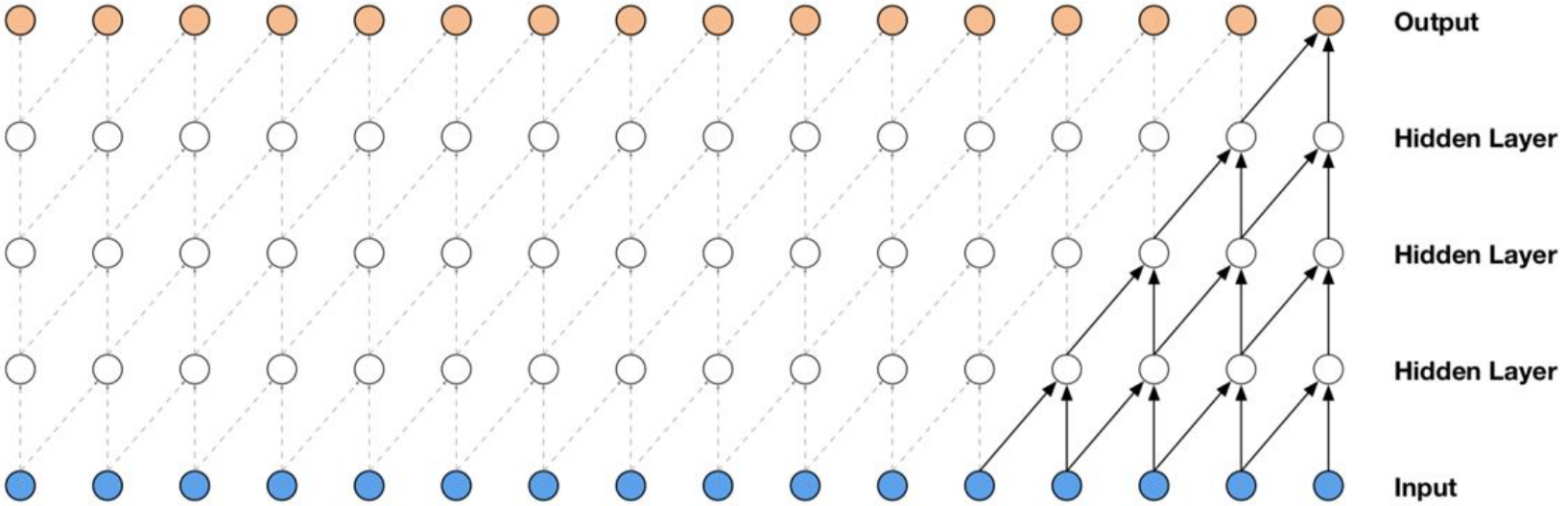
Dilated Causal Convolutions = Dilated Conv + Causal Conv



WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

Causal Convolutions

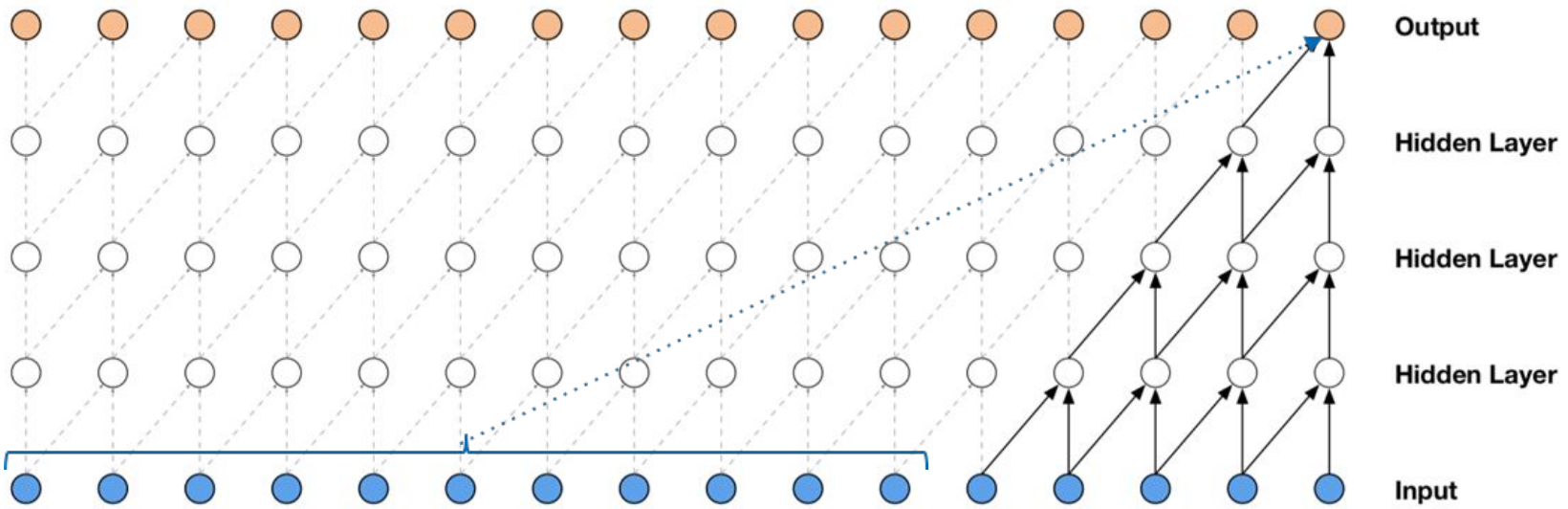
通常の畳み込みを試みると未来の情報を参照しようとしてしまう
しかし音声処理において、これは当然できない。
そこで、過去の情報のみを参照するCausal Convolutionsが提案された



WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

Causal Convolutionsのみの問題点

通常のCausal Convolutionsでは
受容野を増やすために膨大な数のレイヤやフィルタが必要になる



これらの要素も出力に影響する可能性がある　しかし、考慮できていない

カーネルの要素間を0パディングすることで、広い受容野を畳み込むことができる

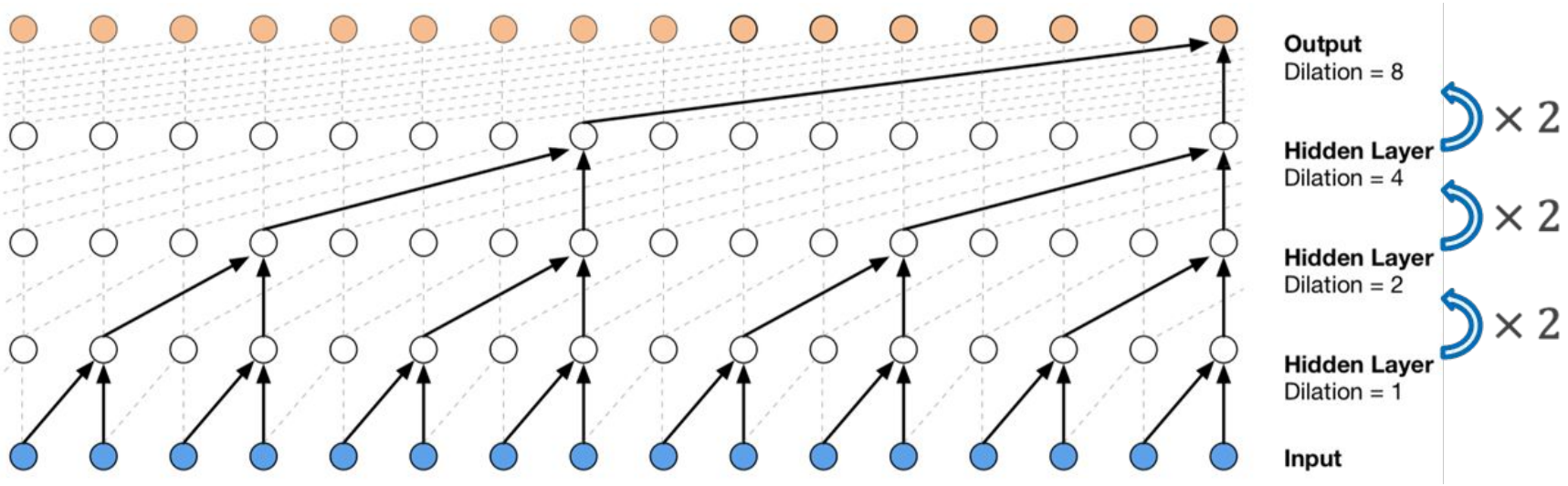
souce:https://github.com/vdumoulin/conv_arithmetic#transposed-convolution-animations

【再掲】Dilated Causal Convolutions

過去のみの情報に基づいて、飛び飛びで畳み込む処理

層を深くすれば、カバーできる受容野が指数関数的に増大する

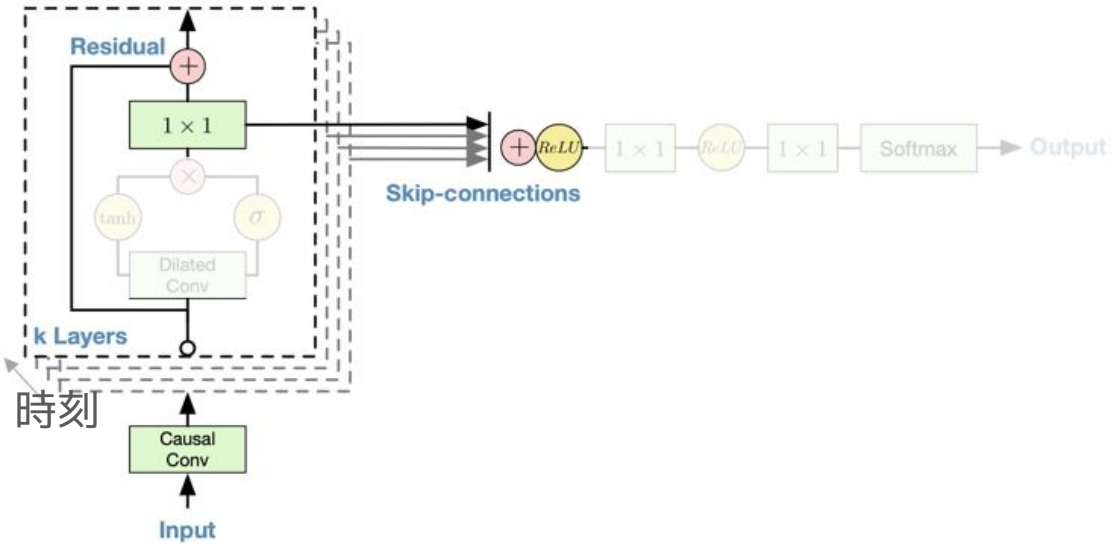
Dilation=〇個飛ばし



Residual & Skip Connections

入力をDilated Conv. の出力に接続することで、その残差を学習する

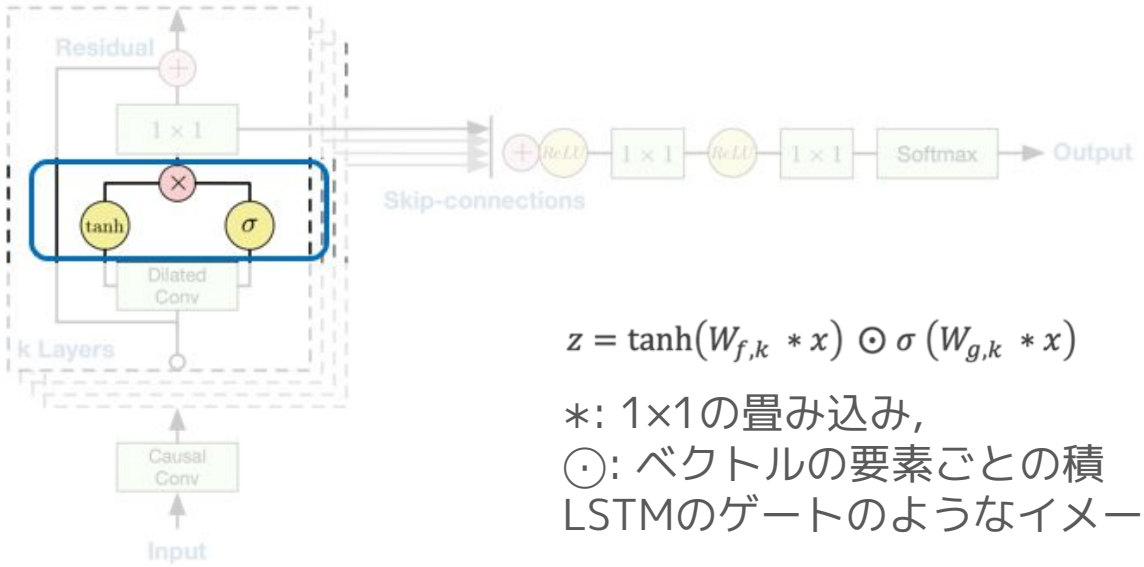
勾配消失問題を回避する目的で使われる



WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

Gated Activation Units

residual Block中の青枠部分



$$z = \tanh(W_{f,k} * x) \odot \sigma(W_{g,k} * x)$$

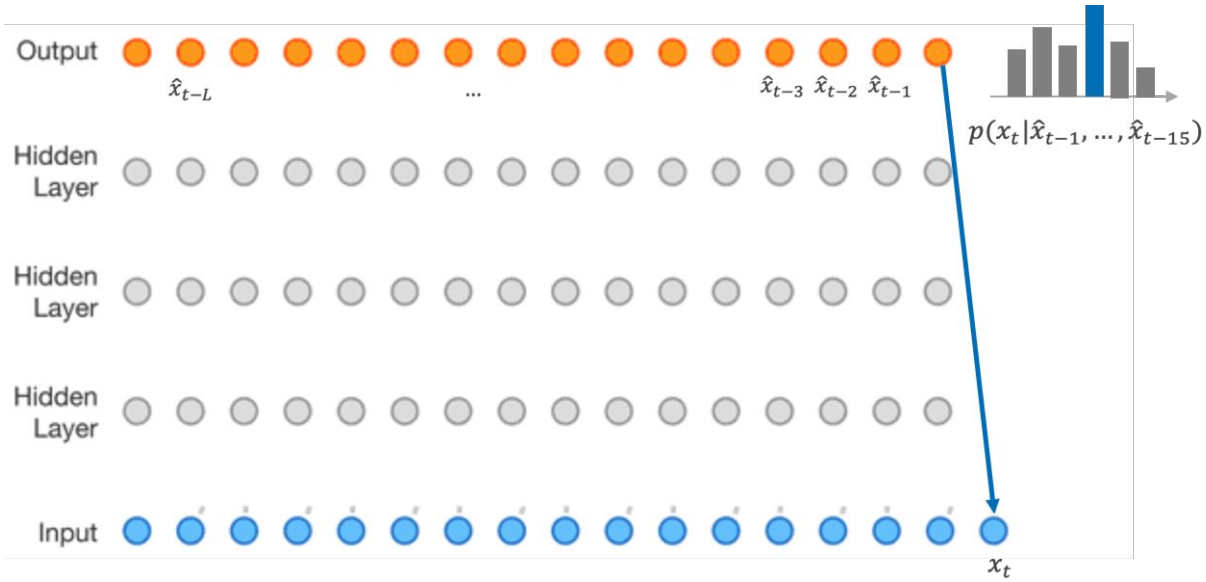
$*$: 1×1の畳み込み,
$\odot$: ベクトルの要素ごとの積
LSTMのゲートのようなイメージ

WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

生成時のWaveNetは？

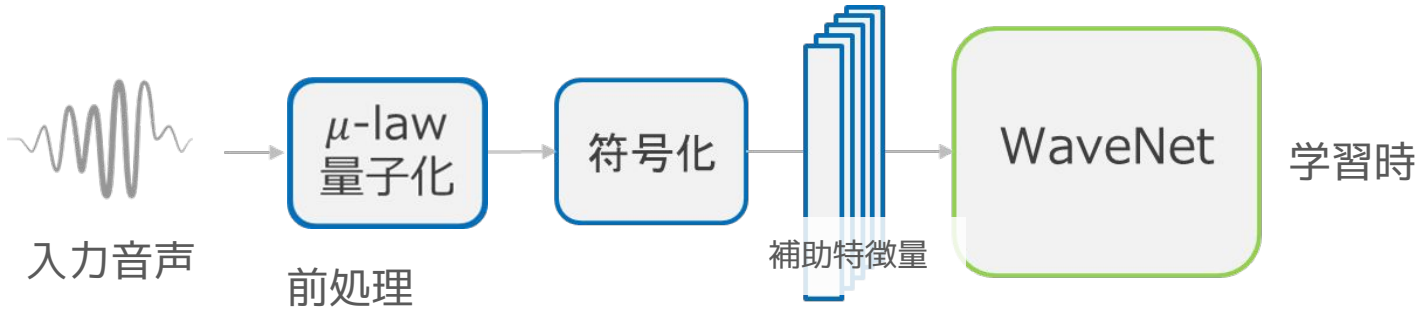
WaveNetの生成処理

生成された出力例 $(\hat{x}_{t-1}, \ldots, \hat{x}_{t-L})$を用いて次の入力$x_t$を予測



WaveNet launches in the Google Assistant

音声波形を生成するためのディープニューラルネットワークの一つ



WaveNetの特徴

✓Dilated Causal Convolutions
過去のみの情報に基づいて、飛び飛びで畳み込む処理
✓Residual & Skip Connections
入力をDilated Conv.の出力に接続する
✓Gated Activation Units
LSTMのゲートのようなイメージ

WaveNetとは

- WaveNetとは
- **生成された音声の例**

US English Voice I

従来手法

WaveNet

US English Voice Ⅱ

従来手法

WaveNet

US English Voice Third Party Voice

従来手法

WaveNet

生成された音声

Japanese Voice

従来手法

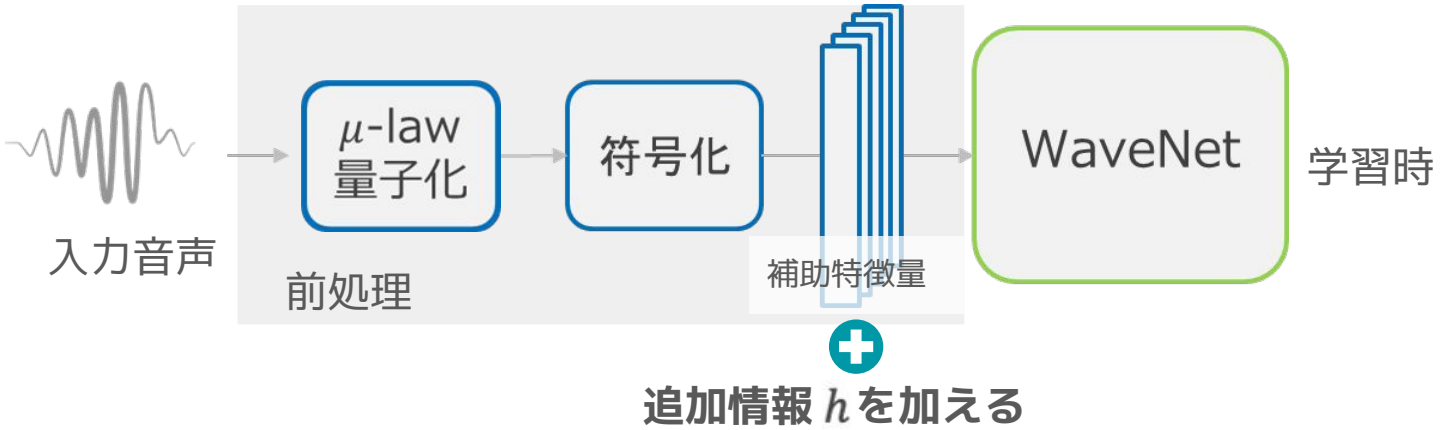
WaveNet

【Chapter12】音声処理分野

WaveNetの応用

conditional WaveNet

さらに生成された音声の特徴を特定するため
入力h(例：Aさんの声、Bさんの声、…のように”誰かの声”という情報）を加える



これにより複数の話者が含まれるデータセットにおいて、
複数の話者の中から特定の話者を選択することができる

テキスト音声合成の場合、入力hを”テキストに関する情報”として与える

$$p(x|h) = \prod_{t=1}^{T} p(x_t|x_1, \ldots, x_{t-1}, h)$$

条件付けの方法は2種類ある

Global Conditioning

大局的な条件付け

例) h は「TTSにおける話し手」を表す

Local Conditioning

局所的な条件付け

例) h は「TTSにおける言語的な特徴」を表す

Global Conditioning

全てのタイムステップに渡って出力の分布に与える影響を条件付け
例)テキスト音声合成に埋め込められた話者など

活性化関数は(Gated Activation Unitsの式に新たな項を追加)

$$z = \tanh\left(W_{f,k} * x + V_{f,k}^{T} h\right) \odot \sigma\left(W_{g,k} * x + V_{g,k}^{T} h\right)$$

ここで
$*$:1×1の畳み込み
$\odot$:ベクトルの要素ごとの積
$V_{*,k}$:学習可能な線形射影
ゆえに、**ベクトル$V_{*,k}^{T} h$は時間次元に渡ってブロードキャストされる**

hは音声シグナルより低いサンプリング周波数であり、局所的な条件付け
例)テキスト音声合成における言語的な特徴など

transposed convolutional network(Deconvolution)によって、現在の時系列を(音声シグナルと同じ解像度の)新しい時系列　$y = f(h)$ 写像する

活性化関数は(Gated Activation Unitsの式の新たな項を追加)

$$z = \tanh\left(W_{f,k} * x + V_{f,k} * y\right) \odot \sigma(W_{g,k} * x + V_{g,k} * y)$$

ここで
＊:1×1の畳み込み
⊙:ベクトルの要素ごとの積

ちなみに$V_{f,k} * h$ を使う方法も考えたが、実験においてあまり良くなかったらしい

この章で学習したこと

1）音声処理について

2）WaveNetとは

3）WaveNetの応用

AVILEN