## Ch01 DL導入

自動微分の種類

Define and Run：グラフの作成と計算の工程を完全に分離する（画像向）

Define by Run：グラフの作成と計算の工程を同時に行う（系列データ向）

DLライブラリ

TensorFlow：計算グラフを自分で定義する必要がある, Define and Run

Chainer：日本のPreferred Networksが開発, Define by Run

TensorFlow2：動的なグラフ作成、シンプル, Define by Run

Pytorch：伸びてる、Numpyに似ている, Define by Run

Mxnet：AWS, 動的・静的グラフの併用可能

Caffe2：画像認識が得意

Cognitive Toolkit：音声認識の改善を目的に開発開始

Theano：モントリオール大学, Define and run

DL高度API

Keras：TensorFlow2からは標準のAPI担った

GLUON：Mxnet用のAPI, 独立したライブラリはない

GPGPU：行列演算を高速に処理できるプロセッサ

FPGA：ユーザが集積回路レベルで構成を設計できるユニット

TPU：Googleが開発した機械学習に特化した集積回路ユニット

## Ch02 MLP基礎

## Ch03 バッチ処理編

回帰：二乗和誤差関数

二値分類：バイナリークロスエントロピー関数

多値分類：クロスエントロピー誤差関数

最適化関数

SGD：確率的勾配降下法

モメンタム：ボールが転がるようにパラメタ更新

AdaGrad：過去の勾配の二乗和を全て記憶し学習率を調整する

RMSProp：古い情報は忘れて新しい勾配情報が反映されるように記憶し学習率を調整する

Adam：勾配の二乗を記憶したボールが関数平面上を転がる

→イテレーション数tでバイアス補正することで、初期段階の不安定性を解消

## Ch04

過学習対策

バイアス-バリアンス分解

損失関数の期待値を分解すると「バリアンス項」「バイアス項」「ノイズ項」

バリアンス項：予測値の分散であり、この値が大きければ過学習をしている

バイアス項：予測値の期待値と目的変数の期待値の差であり、未学習状態である

ノイズ項：データのノイズであり、機械学習でフィッティングできない

ノルムペナルティ

ラッソ回帰：L1ノルムにおけるノルムペナルティ

リッジ回帰：L2ノルムにおけるノルムペナルティ

アーリーストッピング

アンサンブル学習：複数モデルを組み合わせて、学習→汎化性能を向上

バギング：同時多数学習モデル　varianceを下げる

ブースティング：段階的に改善していくモデル　biasを下げる

パラメタ拘束：類似したタスクを学習する2つのモデルのパラメタにノルムペナルティを課す方法

パラメタ共有：モデルのあるパートのパラメタを同じ値にする

→パラメタに制限をかけることで、モデルの過剰な表現力を抑える事ができ、汎用性の高いモデルを作ることができる

ノイズによる正則化

入力値にノイズ：data augmentation　→学習データに多様性が生まれること

隠れ層にノイズ：dropout, dropconnect　→ノードの接続を断つ

目的変数にノイズ：label smoothing　→ラベルの平滑化

Dropout:ノードをステップごとに消す　→　部分的なネットワークのアンサンブル学習と捉えられる

※評価時には全てのノードを使う, 実用性が高い

Dropconnect：指定された割合のパラメタをランダムに選んで0にするテクニック

ラベル平滑化：分類タスクにおける正則化のために目的変数にノイズを加える手法

内部共変量シフト：各層の入力時の分布が学習を進める中で変化してしまう現象

分散処理

モデル並列：ネットワーク上のパラメタを複数のプロセスに分割し訓練する

データ並列：同一のネットワークの複製を複数用意し、それぞれのレプリカに対して訓練データを適用

→学習情報の共有に時間がかかる

同期型：勾配を更新する前に集約し、平滑したものを使う

非同期型：各レプリカ間で同期を取らない　※古い学習時の勾配によりサーバ内の平均勾配が悪化

蒸留：学習済みの教師モデルの入出力を用いて、より軽量なモデル（生徒モデル）で学習

枝刈：寄与の小さい重みを0にする（データの流れを切る）

量子化：浮動小数点で表されるネットワークのパラメタ等を低bitで表現

## Ch05 CNN

Batch Normalization：チャンネル全体の入力を正規化する

Layer Normalization：同じ層のニューロン間で入力を正規化する手法

Instance Normalization：各チャンネルで独立に画像の縦横方向について平均・分散を取る

Group Normalization：チャンネルをG個のグループにグループ化して、中間的な処理を行う

## Ch06 RNN

BPTT

勾配クリッピング

Skip connection

Leaky接続

LSTM

Forget gate

Input gate

Output gate

GRU

双方向RNN

## Ch07 生成モデル

エンコーダ：高次元データから潜在変数を推測するモデル

デコーダ：潜在変数からデータを生成するモデル

潜在変数：潜在的な特徴を表す変数

AE：

Encoder, DecoderはNN。入力と同じものを出力するように学習

隠れ層のノード数は入力層・出力層より少なくする

入力と同じものを入力の次元より少ない次元の潜在変数から出力さ→効果的な潜在変数が作られる

応用例：画像のノイズ除去, クラスタリング

Denoising AE：

入力にノイズを加え、ノイズがないデータを生成することを目指すモデル

ノイズ：

種類

マスキングノイズ：入力の一部を0にする

ソルト&ペッパーノイズ：入力の一部を0か1にする

ガウシアンノイズ：入力に平均0分散σ^2ウス分布からサンプルした値を加える

VAE：

実在しないデータを生み出したい

潜在変数にランダム性と連続性を持たせている

→正規分布を潜在変数に導入

理論が整っているがややぼやける。その原因はまだわかっていない。

Reparameterization Trick：順伝播のときに得たεを記憶しておけば誤差逆伝播が可能

Posterior collapse：表現力があるDecoderを使うときに、潜在変数を無視した生成が行われる現象

VQ-VAE：潜在変数をVAEの時とは異なり、離散的なベクトルとして扱う

Encoderからの出力をEmbedding Spaceで受ける

GAN：

DはV(D,G)を最大化しようとする

GはV(D,G)を最小化しようとする

Unrolled GAN：GよりDの方の学習が進みすぎてしまう問題に対する解決策の1つ

→Generatorに質のいい勾配情報を与える＝学習のバランスが取りやすくなる

Minibatch Discrimination：mode collapseへの対応策

Dにヒントを与える（これは生成された画像fakeの可能性が高い）

Wasserstein GAN

通常のGANがJSDを使って近づいていく→分布の裾が小さいときに全く異なる分布同士でも近づく

EMDという距離を使う→Gの目的を補助してくれる

JSD

通常のGANはJSD(イェンセンシャノンダイバーシティ)を最小にすることで、真のデータ分布を学習する

EMD：「質量」を最適に輸送するコストを考える

## Ch08 強化学習

エージェント：打手（人間やコンピュータなど）

環境：盤面情報、対戦相手のレベル、相手の行動など

状態：自分の手番のときの盤面情報

行動：自分の手番でどこに石を置くか

状態遷移確率：自分の手番での一手が次の自分の手番の盤面にどう影響するか

報酬：現在の盤面からある行動を起こしたときに得られる勝敗に寄与した価値

マルコフ決定過程（MDP）：

状態のみでなく、行動や報酬などを追加した確率過程を考える

マルコフ性：次の状態が現在の状態のみによって決定する性質のこと

方策：マルコフ決定過程への入力となる行動の選択ルールを規定する関数を定義する

累積報酬和：

価値関数：強化学習では目的関数のことを価値関数といい、最大化したい

→ある状態sから方策πに従って動いた際に、今後どれくらい報酬を得ることができるかの指標

価値関数ベースの手法：価値を基準にし、行動の選択は別のアルゴリズムによって決定する

価値反復法：報酬と状態遷移の関係が完全にわかっている

モンテカルロ法：実際に複数回プレイして見て、各1手の報酬をサンプリングし、価値を推定する

有限の時間ステップ数で終了する場合（エピソディック）に有効である

TD法：終了まで待たず、それまでに得た報酬から価値関数V(s)を更新していく手法

初期の価値が学習されにくいなどのデメリットがある

ε-greedy方策：確率0<a<1でランダムに行動し、局所解へ陥るのを防ぐ

ソフトマックス方策：行動価値をエネルギとみなし、ボルツマン分布に従い行動を選択する

SARSA：実際の方策（ε-greedy）に従って価値を更新する

→価値関数の更新にランダム性を取り入れているため局所解に陥りにくい

Q学習：実際に進む行動（方策）と価値観数の更新に用いる行動（方策）が異なる＝方策オフ型 →学習が早い（Q関数の収束が早い）

DQN：Q学習のたくさんの状態・行動に対して処理できないという問題点からDNNを用いて、パラメトリックに価値関数を表現するDQNという方法がある

方策勾配定理：Q値を用いて累積報酬を増加させる方策の勾配が求まる

方策勾配ベースの手法

REINFORCEアルゴリズム：実際に得られた報酬の平均を使ってQ値を近似

Actor-Critic：Critic(評価器)を学習させながら、Criticの推定をベースにしてActor(行動器)を改善

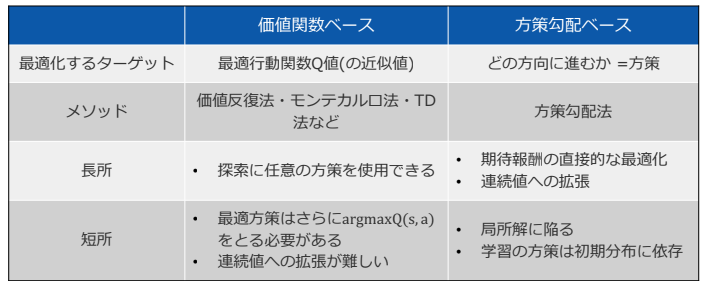
A3C：学習の安定化を図るAsynchronousを導入

A2C：A3Cの学習方法である非同期分散処理から同期分散処理を可能にしたモデル

ACER：A3Cをベースに方策オフ型にして、Experience Replay(経験再生)を利用するモデル

UNREAL：A3Cをベースに補助タスクも同時に学習させるモデル

強化学習まとめ



## Ch09 DL実用

事前学習（Pretraining）：目的とするモデルを訓練する前に予め他のタスクを訓練させておく戦略

転移学習：事前学習済みモデルを引き継いで、追加した層を用いて学習。この時学習済みモデルの重みは固定

→すでに学習済みモデルを流用するので、学習コスト少なく精度も高い

ファインチューニング：転移学習と違って重み全体を学習する。新しい層も追加して再学習

ワンショット学習：一つのラベル付きデータが有れば特徴空間上で周りにあるデータのラベルを推論できる

ゼロショット学習：全てのデータがラベルなし、訓練データに存在しない未知の物体のクラスを識別学習

半教師あり学習：教師あり学習を拡張。一部のデータのみラベルありのデータセットを学習。

アノテーションにかかるコストの削減を目指す（ラベルありデータに偏りがあるとうまく学習できない）

自己教師あり学習：データ間の関係性に対して自動で教師信号を定義して学習する

マルチタスク学習：1つのモデルで複数のタスクを同時にこなせるよう学習することでメインタスクの汎化性能を向上させる

距離学習：

Siamese Network：データセットから2つの入力のペアセットを作る→損失を計算

Triplet Network：ランダムに3つの入力ペアを事前に作成する

メタ学習：Networkのよりよい学習方法を学習する手法である

MAML：少ないデータを使ったタスクをM個用意し、最も良さそうなモデルの初期値を採用し、学習する

Explainable AI

CAM：Global Average Poolingが実装されているモデル（ResNetなど）で使える。

最終的なクラスの出力値に関わる重みwからどこに注力したかのMAPを得る

GAP：対象となるChannelの全ての値から1つの平均を求め出力とする

Grad-CAM：GAPがないNetworkでも動作し、現在では最も主流な可視化手法である

Guided Grad-CAM：負の値を0に置き換える

LIME：影響度合を測るために予測モデルを別のモデルで局所的に近似、局所モデルの係数から寄与率を見る

SHAP：シャープレイ値で各特徴量の貢献度を表せる、かつ近似的に少ない計算で行えるようにした

Relational GCN：N個の関係を持った有向グラフを想定し、関係ごとに畳み込みを行って特徴を抽出する

## Ch10画像分類モデル

AlexNet：【火付け役】CNN層×5 + 全結合層×3

VGG：【シンプル】フィルタは3×3　→　多層化

GoogLeNet：【多重化】Inception moduleの導入

ResNet：【残差】Residual Blockの導入

WideResNet：【残差】層の拡大

DenseNet：【N個の残差】Dense Blockの導入

MobileNet：【計算量削減】Depthwise Separable Conv.の採用

EfficientNet【効率的Network】：パラメータ数の最適化

## Ch11 自然言語処理

CBOW：前後のいくつかの単語の間の単語を予測する

Skip-gram：一単語からその前後の単語を予測する

Bi-LSTM：

RBMT（Rule-Based Machine Translation）：登録済みのルールを適応することで訳文を出力する手法

SMT（Statistical Machine Translation）：統計モデルを学習させることで訳文を出力する手法

Perplexity：次の単語として絞り込めた候補単語数を測る評価指標

BLEU：予測文と正解文の類似度を測る評価指標

N-gram：ある文章を連続するn文字、またはn単語の塊に分割する手法

n-gram：分割された塊のこと

Sequence-to-Sequence：系列データを入力として系列データを出力するアーキテクチャ

Peeky：覗き見

Attention：デコーダがエンコーダの出力に注意を払う仕組み

加法注意：Attentionの重みaを求めるために、隠れ層1つのFFNを用いる

内積注意：Attentionの重みaを求めるために、内積を用いる

AttentionつきBiLSTM：

Google Neural Machine Translation(GNMT)

Transformer：

Positional encoders

Masked Multi-Head Attention

OpenAI GPT

BERT

Masked Language Model(MLM)

Next Sentence Prediction(NSP)

XLNet

ALBERT

## Ch13 生成分野

GAN：24ピクセル　GとDの学習バランスの調整が難しい

LAPGAN：256×256ピクセル　複数のGを用意する必要があり、計算コストが高い

DCGAN：64×64ピクセル　CNNフィルタを可視化する→解釈性を与える, GPUを有効活用

PGGAN：1024×1024ピクセル

sinGAN：ほとんど任意　高解像度化, 絵から画像へ, 編集, 調和

クラス指定画像生成のGAN

cGAN：28×28 MNIST生成　→　クラス情報をGとDに与える

InfoGAN：クラス情報不要　潜在変数を調整することで画像の微調整ができる

ACGAN：128×128　Dは生成画像の真偽判定とクラス分類タスクを担う

SAGAN：Self-Attentionを用いることで画像のより大局的な特徴に注目し、計算コストを抑える

BigGAN：SAGAN + Truncation Trick ノイズzの閾値を設けることで多様性と画質のバランスを調整

StyleGAN：Styleによる影響はどの解像度の画像に組み込むかで求める

画像変換のGAN

Pix2pix：1ペアの画像の関連性を学習する

CycleGAN：ウマ↔シマウマのように画像のドメインを変換するGAN

StarGAN：単一モデルでの複数のドメイン変換を可能にしたGAN

StackGAN：テキストから画像を生成するGAN

AnoGAN：画像内の異常検知を行うGAN

## Ch14 強化学習

TD法：「ひとつ先の報酬と行動価値」と「現在の行動価値」との差を用いて最適な行動価値観数へと更新する

Q学習：TD法のアイデアで最適行動価値関数を収束させる学習

DQN：2つのNNを用いて、最適行動価値関数を求める手法

ApeX：1つのLearnerと複数のActorがReplay memoryを共有する分散学習

R2D2：ApeXにLSTMを導入した

R2D3：demonstrations（人間の熟練者の経験データ）を投入

AlphaGo

AlphaGo Fan：MCTS+2つのDNNで構成される

AlphaGo Lee：イ・セドルに勝利

AlphaGo Master：オンライン上で60戦全勝

AlphaGo Zero：MCTS+1つのDNNで構成され、Fan比べてシンプルなアーキテクチャ