深層学習day4

Section 1 強化学習
◯強化学習とは
・教師あり/教師なし/強化学習の分類
・長期的に報酬を最大化できるようなエージェントを学習させる
・報酬をもとに行動を決定する原理を学習していく

・探索と利用のトレードオフ
　：過去のデータを利用しすぎるとベストな行動を見つけられない（探索が足りない）
　　未知の行動をとり続けると過去の経験が活かせない（利用が足りない）

・方策関数、行動価値関数の二つを学習させる

・強化学習では優れた方策を見つけることが目標
　cf.) 教師あり/なし学習

・Q学習：行動価値関数を、行動するごとに更新することにより学習を進める方法
・関数近似法：価値関数や方策関数を関数近似する手法

◯価値関数
・状態価値関数V(s)：状態をもとに価値を決める
・行動価値関数Q(s, a)：状態と行動の組み合わせをもとに価値を決める

◯方策関数Π(s) = a
・ある状態でどのような行動を取るのかの確率を与える関数
・方策勾配法：方策をモデル化して最適化する手法

Section 2 AlphaGo
・AlphaGo Lee, AlphaGo Zeroの二種類

◯AlphaGo Lee
・PolicyNet : 方策関数
　盤面特徴入力（19×19×48チャンネル）を入力
　19×19マスの着手予想確率を出力
・石、取れる石の数、着手履歴などの情報を合計して48チャンネル

・ValueNet : 行動価値関数
　盤面特徴入力（19×19×49チャンネル）を入力
　現局面の勝率を-1 ~ 1で表したものを出力

◯学習のステップ
・1. 教師あり学習によるRolloutPolicyとPolicyNetの学習
　過去の人間同士棋譜から教師データを用意し学習
・2. 強化学習でPolicyNetの学習
・3. 強化学習でValueNetの学習

※RollOutPolicy：NNではなく線形の方策関数
　探索中に高速に着手確率を出すために使用

・モンテカルロ木探索：強化学習の探索手法

◯AlphaGo Zero
・教師あり学習を一切行わず、強化学習のみでモデルを作成
・特徴量のチャンネルからヒューリスティックな要素を排除し、石の配置のみにした
・PolicyNetとValue Netを一つに統合
　ー一つのネットワークの途中で枝分かれしPolicyとValueの出力を生成
・Residual Networkを適用(×39)
　など

・Residual Network
　モデル構造にショートカット構造を適用したもの
　層を深くしても勾配消失を抑える、アンサンブル学習ができる等の利点

Section 3 軽量化・高速化技術
◯全体像
・分散深層学習：並列的にニューラルネットワークを構築し、効率の良い学習を行う。
　データ並列化、モデル並列化、GPUによる高速技術
・モデル高速化、軽量化

◯モデル並列
・親モデルを各ワーカーに分割し、それぞれのモデルを学習させる。
　全てのデータで学習が終わった後で、一つのモデルに復元。
・モデルが大きいときはモデル並列化、データが大きいときはデータ並列化をするとよい。

・モデルのパラメータ数が多いほど計算効率も上がる

◯データ並列
・親モデルを各ワーカーに子モデルとしてコピー
　データを分割し、各ワーカーごとに計算させる
　計算させるPCを増やす、演算器を増やすなど

・パラメータの反映させ方
：同期型
　ー各ワーカーの勾配の平均を計算し、パラメータを更新する
：非同期型
　ー各ワーカーがお互いの計算を待たずに、各モデルごとに更新を行う
　　学習が完了したらパラメータサーバにPush, 新しく学習を始めるときに、
　　パラメータサーバからPopしたモデルで学習を行う。
・非同期型の方が計算が早いが、同期型の方が精度が良いことが多い

◯GPU

・GPU:比較的低性能なコアが多数。簡単な並列処理が得意
・CPU:高性能なコアが少数。複雑で連続的な処理が得意。
・GPGPU(General-purpose on GPU)
　：元々の使用目的であるグラフィック以外の用途で使用されるGPUの総称。
・GPGPU開発環境
　ーCUDA、OpenCL

◯量子化
・大量のパラメータに伴い、学習・推論に多くのメモリと演算処理が必要
　→通常のパラメータの64bit浮動小数点を32bitなど下位の精度に落とし、メモリと演算処理の削減

・8byte(64bit) × 1024(KB) × 1024(MB) × 1024(GB)を4byte(32bit)、2byte(16bit)に変える
・計算の高速化、省メモリ化するが、モデルの表現力が低下し精度が下がる
　実際の問題では倍精度(64bit)を単精度(32bit)にしてもそれほど精度に影響しない

◯蒸留
・規模の大きなモデルの知識を使い、軽量なモデルの作成を行う
・教師モデル：予測精度の高い高性能なモデル
・生徒モデル：教師モデルをもとに作られる軽量なモデル
・学習済み教師モデルと生徒モデルの誤差を使い生徒モデルを更新していく

◯プルーニング
・モデルの精度に寄与が少ないニューロンを削減し高速化する
・重みが閾値以下の場合ニューロンを削減し再学習を行う

Section 4 応用モデル
◯MobileNet
・モデルの軽量化、高速化、高精度化を実現したモデル
・従来の計算量：H×W(画像サイズ)×K×K(カーネルサイズ)×C(チャネル数)×M(フィルタ数)
・Depthwise Convolution, Pointwise Convolutionの組み合わせで計算量を削減した

・Depthwise Convolution
　フィルタを1に固定(H*W*K*K*C)

・Pointwise Convolution
　1×1畳み込み
　カーネルを1×1に固定(H*W*M*C)

・DepthwiseとPointwiseを組み合わせることでパラメータを作成
　(H*W*K*K*C) + (H*W*M*C)

◯DenseNet
・Dense block

出力層に前層までの入力を足し合わせる
結合する過程で特徴マップのサイズが増えていく
増やしたチャネルをtransition layerで特徴マップのサイズを減らす

・成長率をハイパーパラメータとして持つ

◯BatchNorm
・Batch Normalization：データの分布をミニバッチ単位で正規化
ミニバッチ数に影響を受けてしまう（デバイスによってミニバッチ数が変わる等）
ミニバッチ数が少ないと学習が収束しないことがある。

・Layer Normalization
一つの各sample, 画像に対し正規化
・Instance Normalization
一つの各チャネルに対し正規化

◯WaveNet
・生の音声波形を生成する深層学習モデル
・時系列データに対して畳み込み(Dilated convlolution)を適用する
一層が深くなるにつれて畳み込むリンクを離す

Section 5 Transformer
◯RNN×言語モデル
・それまでの単語の並びに対して尤度（文章として自然か）を確率で評価
・時刻t-1までの情報で、時刻tの事後確率を求めることが目標
・言語モデルを再現するようにRNNの重みが学習されていれば、ある時点の次の単語を予測できる

◯Transformer
・Seq2seqでは文の長さに弱い
→Attention：単語間の関連度を利用
・Transformer:RNNを使わずAttentionのみを使用
計算量が少なく当時のSOTAを実現

・Source Target AttentionとSelf-Attentionの二種類

Section 6 物体検知・セグメンテーション
◯鳥瞰図
・入力：画像
・出力：分類、物体検知(Bounding Box)、意味領域分割、個体領域分割
・物体の位置、個々の物体の認識へとタスクの難易度が変わっていく

・代表的なデータセット
VOC12, ILSVRC17, MS COCO18, OICOD18など
Instance Annotationが含まれる

一枚あたりいくつ物体があるか(Box/画像)の指標も重要
（部分的な重なり等や日常生活のコンテキストに近くなる）

◯評価指標
・分類：confusion Matrix
・物体検出ではクラスラベル＋物体位置の予測精度（IoU)
・IoU = Area of Overlap / Area of Union
※IoUは値の直感的解釈が難しい
・物体検知では、confidence + IoUで閾値を決めて評価する
・Average Precision : confidenceの閾値を変化させる
・Flames per Second : 検出速度の指標

◯物体検知の大枠
・2012年：AlexNetを皮切りにDCNNへ
・物体検知は大きく1段階検出、2段階検出に分けられる
・2段階検出：候補領域とクラス推定を別々に行う。精度が高い傾向
・1段階検出：候補領域とクラス推定を同時に行う。計算量が小さく推論も早い傾向

◯SSD(single shot detector)
・1段階検出器
・default BOXを定義→default BOXを変形し出力
・VGG16をベースにしたモデル

◯Semantic Segmentation
・層を経るごとに解像度が落ちてしまう
→UpSamplingで解像度を戻す
・Poolingの必要性：正しく認識するために受容野にある程度大きさが必要

・Deconvolution/Transposed convolution
：UpSampling手法
※逆演算ではなく、poolingで失われた情報が復元される訳ではない

・Dilated Convolution：Convolutionの段階で受容野を広げる工夫

◯GAN
・生成器と識別器を競わせて学習する生成&識別モデル
・Generator : 乱数からデータを生成
・Discriminator : 入力が真データ（学習データ）であるかを識別
・価値関数Vに対し、Dが最大化、Gが最小化を行う

◯DCGAN(Deep Convolutional GAN)
・GANを利用した画像生成モデル
・Generator : Pooling層の代わりに転置畳み込み層を使用、活性化関数を変更
・Discriminator : Pooing層の代わりに畳み込み層を使用、活性化関数を変更
・共通事項：中間層に全結合層を使わない、Batch Normalizationを適用