機械学習・AI

【物体検出】vol.3 ：YOLOv3の独自モデル学習の勘所

オリジナルで学習したモデルを使った物体検出

YOLOv3の環境構築が終わり、一通り学習済モデルで「おぉぉぉ」と興奮した後は、オリジナルモデルの学習に興味が沸いてきます。

YOLOv3の学習については、下記ような参考サイトで手順を確認できます。

参考にしたサイト

AlexeyAB/darknet

YOLOオリジナルデータの学習

YOLOオリジナルデータの学習その２（追加学習）

YOLO v3による顔検出：02.Darknetで学習

Windows 10上のDarknetでYolo v3をトレーニングしOpenCVから使ってみる

以下では、私が実際にやってみた「勘所」の部分を記載します。

どれくらいのデータを用意すれば良いか？

最低：1カテゴリに対して100枚

基準：1カテゴリ1000枚

推奨：1カテゴリ5000、10000枚（高い精度と検出率、差異が少ない対象を扱っている場合、汎化性能を求める場合）

1カテゴリ当たり100枚で「試しにやってみる」
⇒ PoC（Proof of Concept：仮説検証）

角度、大きさ、色相、明度の異なる複数のバリエーションを偏りなく集め、1カテゴリ1000枚を達成します。
⇒ 精度を向上します。

誤検知のデータを修正、データの水増しをしながらブラッシュアップを繰り返していくと、結果的に5000枚、10000枚になります。

どこまでいったら学習を止めるか？

カテゴリ毎にAPが90%、mAPで80%達成できたら、目標達成です。

1カテゴリ画像100枚、10カテゴリ（1000枚）を集めるためにはおよそ3日のデータ作成時間と、1日の学習、1日の評価時間が必要です。

データ量が10倍になればデータ作成時間は10倍(30人日)ものボリュームに。10000枚ならさらに10倍。どこまで費用をかけられるのか？が制限になります。

ハイパーパラメータの適正値は？

https://github.com/AlexeyAB/darknet に書かれていることですが、

バッチサイズ：64で試す→GPUメモリが足りなければ32に変更
イテレーション：（カテゴリ数*2000）以上
学習データと評価データの割合：70%、30%

例：
1カテゴリ100枚で10区分の場合、かつ学習データと評価データを7:3とすると、700枚が学習に使われます。バッチサイズ32枚とすると、700÷32≒20イテレーションで1エポックとなります。

推奨値である区分数*2000イテレーションを正とすると、20000イテレーション、1000エポックです。

1エポックで評価データとして300枚（1カテゴリ30枚）が使われます。APを求める母数として、わずか30枚に対する割合や精度だということに注意したほうが良いです。1枚の誤検知があると3%上下するということ。

評価指標について

何をもって、モデルの精度や検出率を評価するかというのは、「課題に対して異なる」というのが真のようですが、まず、YOLOで良く出て来る指標を押さえておきます。

Precision **適合率**：	検出結果の中に、適合しない文書が入っていない割合 Precision(P) = tp / (tp + fp)
Recall **再現率**：	すべてのデータのうち、どれだけ拾うことが出来たのか（漏れなく） Recall(R) = tp / (tp + fn)
TP **（TruePositive）**	検出すべきものを検出できた数
FP **（FalsePositive）**	検出すべきではないのに検出した数
LOSS：	正解とどれくらい離れているかを表す値
AP：	AP = Average Precision、平均適合率。適合率の平均
mAP：	MAP = Mean Average Precision、平均適合率の平均。複数のカテゴリのAPの平均。1イテレーションor1エポック単位で各カテゴリのAPの平均を出して、学習が収束しているかどうかの目安にする
iOU：	検出した枠が正確に対象を囲んでいる割合
OverFitting **過学習**：	学習データに最適化しすぎて、それ以外のデータではそれほどではない状態（専門バカ、教えたことしかできない子）

iOUは、50%で評価することが多いようです。面積で半分と言えば、縦横がどちらか半分くらいずれててもいいでしょう？というレベル。
75%の面積の一致とは、縦横のいずれも半分ずれていないよというレベル。

「50%合っていればいいんじゃないでしょうか？」と思いますが、位置決めがシビアな用途には、もっと精度が必要かもしれません。

…が、アノテーションデータの囲い方にもシビアな品質の追求が必要です。そこまで行ってしまうと、ちょっと現実的ではない気がしますね。

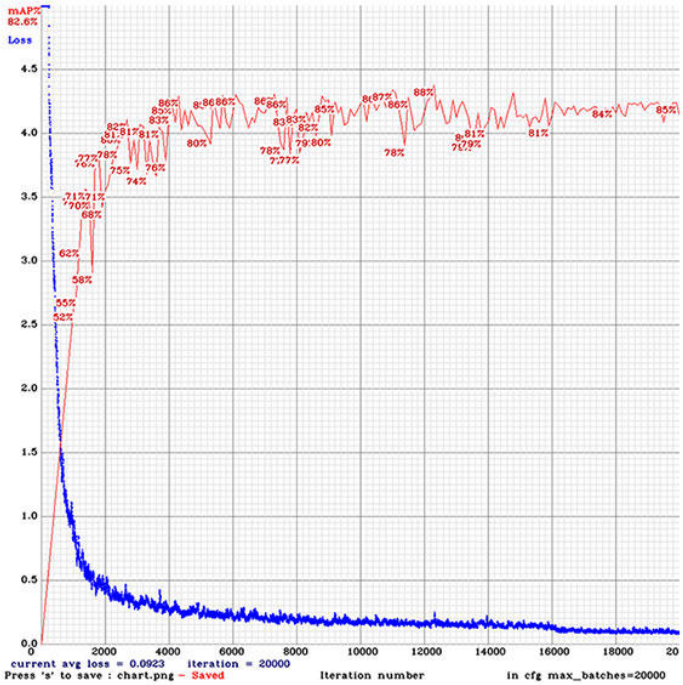
mAPは、完全にでたらめに予測をしても50%は出せるでしょうから、60%以上なければ、有意な検出では無いのでは？と想像できます。
（そして実際に、50%を越えなければ、まったく使い物になりません）

Precisionも感覚的には、60を超えると、「まぁまぁ分かっているな」。
70を超えると「少し間違えるかな」、80を超えると「間違えなくなってきたな」、90を超えると「過学習かな」と疑うレベルです。

イテレーションと各種評価指標の関連は？

評価は、iOU=50%で固定して評価します。

イテレーションが進むとLossが減り、mAPが向上します。
（Lossが少なく、mAPが高ければ高いほど良い。）
Loss　＜0.2
mAP　＞0.7



イテレーションとLoss、mAPの推移

実例：カテゴリごとのAP（TP、FP）から判断できる事

イテレーション：20000回時点での各区分のAP（正解率）

class_id = 0, name = 5_****, ap = 98.12%　(TP = 112, FP = 5)
class_id = 1, name = 3_****, ap = 65.53%　(TP = 17, FP = 5)
class_id = 2, name = 7_****, ap = 59.88%　(TP = 27, FP = 11)
class_id = 3, name = 4_****, ap = 89.49%　(TP = 8, FP = 4)
class_id = 4, name = 1_****, ap = 96.03%　(TP = 112, FP = 8)
class_id = 5, name = 6_****, ap = 73.17%　(TP = 63, FP = 20)
class_id = 6, name = 2_****, ap = 95.64%　(TP = 137, FP = 10)

class_id：1、2でAPが低いことが分かります。区分ごとに正解率が高いものと低いものがあることが分かります。（→原因が、データ量の偏りにあるのかどうかを確認すること！）

特に、class_id：2のFPが11と高いです。（FP/(TP+FP)=28.9%）。これは誤検知の割合を示し、検知した中に、本来検知されるべきではないものを示します。区分が紛らわしいか、特徴が他のカテゴリと被っていないかを確認する必要があります。

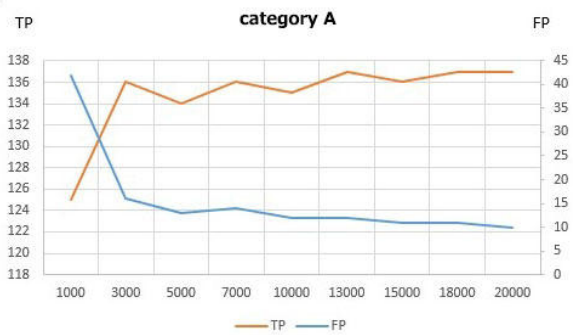
イテレーションを重ねると起こること

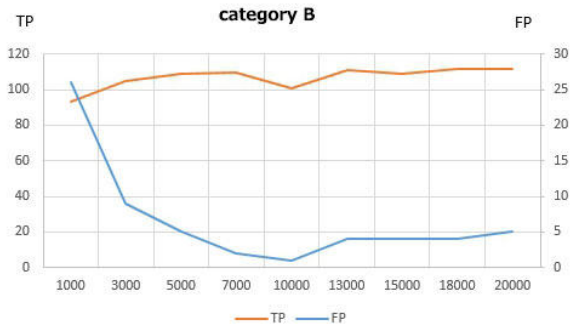
次に、これらの各イテレーション毎に、各カテゴリのTP、FPがどのように推移しているかを確認すると、各カテゴリごとに学習が完了したかどうかを推測できます。

⇒ 総検出数が減ります（余計な検出をしない）
⇒ 誤認識が減ります（間違えない）
⇒ バウンディングボックスの精度が上がります（より正確な矩形を描く）

実例

イテレーション毎にTP､FPをプロットしたグラフ：





TPがあるところで頭打ち、FPが下げ止まりしていれば学習が収束（完了）しています。FPが高い数値でそれ以上下がらない場合は、特徴を上手く捉えられていない可能性があります。

データが少ないか、他のカテゴリと特徴が被っている、未定義の紛らわしいものがあるか。カテゴリの戦略か、学習データを見直す必要があるかもしれません。

感覚的なこと

イテレーション毎の重みを使って、テストを実行した際、mAPやLossの数字だけではなかなかわからない、"体感"が確かにあります。

経験的に

Lossが0.1台になり、mAPが80%を超えるようになると、学習データを使った検証で誤認はほとんどありません。

では、それが絶対ベストな状態になっているかというと、実はそうでもなく途中のイテレーションのモノを採用した方が良い場合もあったりします。

例えば、

> 2つの物体が重なって存在しているときに、2つのカテゴリを正しく検出している

1つの物体に対して、2つの区分を回答してしまっていた部分が、正しく一つだけ検出するようになった

誤検出が減った

紛らわしく見えてしまうもの（人間でさえ誤認しそうなもの）を正しく選り分けている

実は、この辺りの都市伝説的なことは、「validationに使うデータが、本当に偏りなく、いろんなパターンを網羅しているのか？」が怪しいんじゃないかと思っています。もっともっとデータを増やした時に、徐々に部分最適解ではなく、全体最適解に収束していくのかもしれません。

どこまで行っても、学習データにおいて良く認識できていることと、汎化性能が確保されているかどうかは、学習時（及び学習時に分割したデータを使ったvalidationにおける）LossやmAPといった指標だけではわかりませんので、あとは色々な（学習や評価に使ったことがない）動画や、紛らわしく意地悪に加工したデータで検証します。

パフォーマンスは如何ほど？

これまでに紹介している当社のゲーミングPC（普通に20万円台で購入できるものです）において、

640*480ドットの荒い動画でも十分に認識できます。（所詮416ドットで切り出している。これより小さいとダメかもしれませんね）

米粒ほどの物体でも認識します。（あっているか間違っているかはわかりませんが）

2物体重なっていても正確に検出します。（3物体以上重なっている場合はダメのようです）

WindowsゲーミングPC（GeForce　GTX1070)で640x480ドット@30fpsの動画を、40-50fpsで認識できました。4K動画の場合で15fps程度です。

720pのUSBカメラでリアルタイムに物体検出させた場合、34.2fps出ました。

▼この記事を書いたひと

R&Dセンター
松井　良行

R&Dセンター 室長。コンピュータと共に35年。そしてこれからも！

おすすめの関連記事

【物体検出】vol.5 ：YOLOv3のファンクションと引数のまとめ（私家版）

【物体検出】vol.4 ：YOLOv3をWindows⇔Linuxで相互運用する

【物体検出】vol.2 ：YOLOv3をNVIDIA Jetson Nanoで動かす

【物体検出】vol.1 ：Windowsでディープラーニング！Darknet YOLOv3（AlexeyAB Darknet）

機械学習・AIの最新記事

【物体検出】vol.17 ：Darknet YOLOv4でRTX2080Superのベンチマーク（GTX1070の1.7倍！）

【物体検出】vol.16 ：Darknet YOLOv4の新機能 -save_labelsで"検出結果を学習データに活用する"

【物体検出】vol.15 ：Darknet YOLOv3→YOLOv4の変更点（私家版）

【物体検出】vol.14 ：YOLOv4 vs YOLOv3 ~ 同じデータセットを使った独自モデルの性能比較