[2018] ArcFace: Additive Angular Margin Loss for Deep Face Recognition #24

New issue

Open **yoheikikuta** opened this issue on 7 Mar · 10 comments

yoheikikuta commented on 7 Mar • edited — Owner

論文リンク

https://arxiv.org/abs/1801.07698

公開日（yyyy/mm/dd）

2018/01/23

概要

大規模顔認識で使える特徴量学習において、超球面上の幾何学的な距離とマージンを考えることで、分類問題として学習するだけでより discriminative な特徴量を学習できるという手法を提案した。
具体的には、通常の softmax に使う特徴量に対して、工夫を施す。まずは特徴量を l2-normalize してそれと列ごとに l2-normalize した weight を掛けて、それぞれの要素が $\cos\theta_i$ として表せる特徴量を作る。この要素数は最終的な予測クラス数と同じになっていて、GroundTruth のラベルに対応する要素には $\cos(\theta + m)$ とマージンを入れることでクラス間をより分離するようにして、この特徴量を softmax につなぐということをする。
かなり色々な実験で先行手法、特に SphereFace と CosFace、と比較して良い性能を示した。性能がめちゃくちゃ上がるという感じでもないが、シンプルな単一の loss で高性能であることに加えて、先行手法を包括的に整理しているところが良い点。

yoheikikuta commented on 7 Mar — Author Owner

最近ちょっと tripletloss などを扱っていて、なかなかコントロールが難しいという実感を持っていた。
それとは別に顔認識系の最近の発展を調べたりもしていて、この辺の論文の存在を知った。
どんなもんなのか真面目に読んでちゃんと理解してみようというモチベーションで読んでみた。

yoheikikuta commented on 7 Mar — Author Owner

まず、顔認識のやり方として典型的には {通常の分類として softmax を使って解く, tripletloss で特徴量学習を使う} の二種類がある。こいつらそれぞれのダメなところを挙げると以下。

- softmax での分類
  - クラス数が多いとパラメタ数が多くなる（出力がクラス数の fully connected layer を使うので）
  - 学習後に使う際に open set なデータに弱い
- tripletloss での特徴量学習
  - データが大きいと triplet pair の組み合わせが大きくなる
  - semi-hard sample mining （学習する際に適度な難しさの pair を選ぶ）が必要でこれが簡単ではない

yoheikikuta commented on 7 Mar — Author Owner

最近の傾向は分類をより discriminative にするためにマージンを工夫するという流れになっている。
例えば center loss はあるサンプルの feature がそいつが属するクラスの中心から離れないようにペナルティを加えるもので、SphereFace は最終層の線形変換が角度空間でのクラス中心の表現として使えることを仮定してそこで feature 間の角度にペナルティを加えるようになっている。
後者はこの段階では何言ってるか分からんが、とりあえずどうやって discriminative にするために典型的に使うアプローチを図示する。

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

1 participant

る。これはクラス内が凝集してクラス間が離れるのを同時に実現しようとするものであり、今回の ArcFace もこれを使う。
Intra と Inter はそれぞれクラス内を小さくクラス間を大きくするように働くもの。
Triplet-Loss はやりたいことは Margin-Loss と同じだが、クラス中心という概念は持ち出さずにサンプル同士を引っ張ってきてペアを作るようになっている。

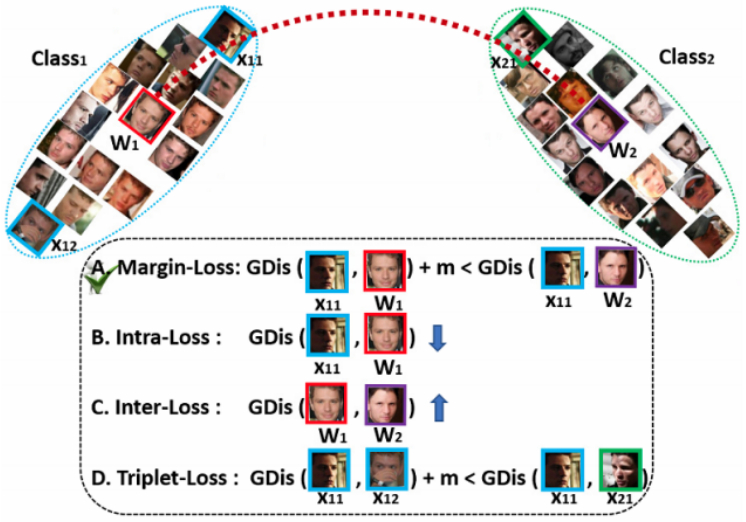


Figure 1. Based on the centre [18] and feature [37] normalisation, all identities are distributed on a hypersphere. To enhance intra-class compactness and inter-class discrepancy, we consider four kinds of Geodesic Distance (GDis) constraint. (A) Margin-Loss: insert a geodesic distance margin between the sample and centres. (B) Intra-Loss: decrease the geodesic distance between the sample and the corresponding centre. (C) Inter-Loss: increase the geodesic distance between different centres. (D) Triplet-Loss: insert a geodesic distance margin between triplet samples. In this paper, we propose an Additive Angular Margin Loss (ArcFace), which is exactly corresponded to the geodesic distance (Arc) margin penalty in (A), to enhance the discriminative power of face recognition model. Extensive experimental results show that the strategy of (A) is most effective.

1

yoheikikuta commented on 7 Mar • edited Author Owner

ということで提案手法の説明。
論文の Figure 2 を眺めてこれを理解しよう。

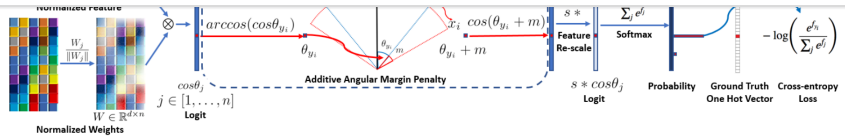


Figure 2. Training a DCNN for face recognition supervised by the ArcFace loss. Based on the feature x_i and weight W normalisation, we get the $\cos\theta_j$ (logit) for each class as $W_j^T x_i$. We calculate the $arccos\theta_{y_i}$ and get the angle between the feature x_i and the ground truth weight W_{y_i}. In fact, W_j provides a kind of centre for each class. Then, we add an angular margin penalty m on the target (ground truth) angle θ_{y_i}. After that, we calculate $\cos(\theta_{y_i} + m)$ and multiply all logits by the feature scale s. The logits then go through the softmax function and contribute to the cross entropy loss.

Algorithm 1 The Pseudo-code of ArcFace on MxNet

Input: Feature Scale s, Margin Parameter m in Eq. 3, Class Number n, Ground-Truth ID gt.

```
x = mx.symbol.L2Normalization (x, mode = 'instance')
W = mx.symbol.L2Normalization (W, mode = 'instance')
fc7 = mx.sym.FullyConnected (data = x, weight = W, no_bias = True, num_hidden = n)
original_target_logit = mx.sym.pick (fc7, gt, axis = 1)
theta = mx.sym.arccos (original_target_logit)
marginal_target_logit = mx.sym.cos (theta + m)
one_hot = mx.sym.one_hot (gt, depth = n, on_value = 1.0, off_value = 0.0)
fc7 = fc7 + mx.sym.broadcast_mul (one_hot, mx.sym.expand_dims (marginal_target_logit - original_target_logit, 1))
fc7 = fc7 * s
```

Output: Class-wise affinity score $fc7$.

まず画像分析のモデルから特徴量を持ってくる。ここでは `fc7` となっているが、これはモデルに依存する部分。とにかく softmax につなぐ前の fully connected の出力になっているもの。

この特徴量を x_i ∈ R^d とする。分類問題の学習においては、これは GroundTruth のラベルとセットなので {x_i, y_i} となっている。
こいつは適当な特徴量の次元 d なので、最終的に解きたい分類問題のクラス数 n を得るために行列 W ∈ R^{d*n} を掛けて W^T x_i で次元 n のベクトルを得たいと考える。
ポイントはここで単純に積を取るのではなく、特徴量を l2-normalize しておき、行列 W も列毎に l2-normalize する（すなわち W^T の各行が l2-normalize されている）という点である。
W^T x_i により得られる要素はそれぞれ l2-normalize されたベクトル同士の積なので cos θ と表現できる。

この cos θ は真面目にベクトルとして書くと、cos θ_j (j = 1,2,...,n) というものになっている。
マージンのことを思い出してみると、正解のクラスに該当する要素にマージンを加えてより discrminataive にしたいのだった。
すなわち、正解のクラス y_i に対応する cos θ_{y_i} を取り出して、こいつにマージンを付与する。マージンの入れ方は一意ではないと思うが、ここでは arccos(cos θ_{y_i}) + m として単純に角度に m を足す。

こうして GroundTruth に該当する要素だけマージンを加えた特徴量ができるので、これを適当にスケールして（これは cos だと [-1,1] が値域で softmax を計算するときに違いが出づらいのでスケールしてるものと思われる）、あとは通常通り softmax につなぐ。
以下のような softmax が得られる。

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}.$$

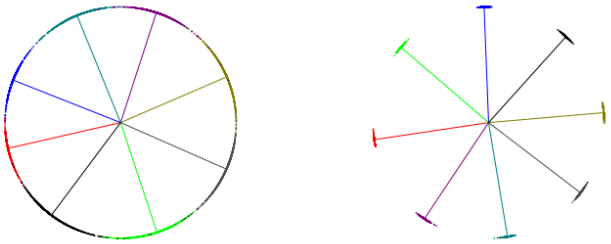
これの効果を定性的に考えてみる。
正しいクラスに対応する softmax の要素は大きい値を持って欲しいので、正しいクラスに対応する W^T_{y_i} と x_i はなるべく alignment が揃って角度は 0 に近く、cos の値は 1 に近くなって欲しい。ここにマージンが入ってくるので、何となく揃うというのは許容されにくくなってクラス毎に alignment が揃いやすくなる。
また、W^T_{y_i} に関しては、同じクラスであれば同じベクトルが使われるので、こいつに揃うように学習が進んでいくという意味でクラスの中心として働いていると理解できる。ということで上で書いた話がちゃんと理解できる。

> 例えば center loss はあるサンプルの feature がそいつが属するクラスの中心から離れないようにペナルティを加えるもので、SphereFace は最終層の線形変換が角度空間でのクラス中心の表現として使えることを仮定してそこで feature 間の角度にペナルティを加えるようになっている。
> 後者はこの段階では何言ってるか分からんが、とりあえずどうやって discriminative にするために典型的に使うアプローチを図示する

こういうのはアイデアが難しくないので、先行研究とかを辿らずともちょっと考えれば分かるので読むのがラクだ。
何にせよ、これでコアの部分の理解はできた。

yoheikikuta commented on 7 Mar Author Owner

feature は normalize されてるので半径が同じで円状に配置されるようになっている。



(a) Softmax (b) ArcFace

Figure 3. Toy examples under the softmax and ArcFace loss on 8 identities with 2D features. Dots indicate samples and lines refer to the centre direction of each identity. Based on the feature normalisation, all face features are pushed to the arc space with a fixed radius. The geodesic distance gap between closest classes becomes evident as the additive angular margin penalty is incorporated.

yoheikikuta commented on 7 Mar Author Owner

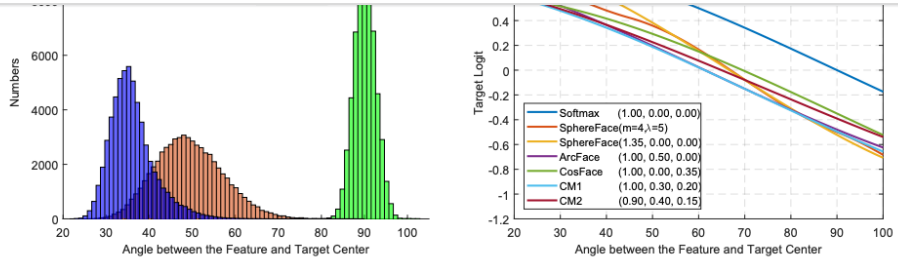
先行研究として、SphereFace と CosFace があるのでそれらとの違いを比較する。

あまり細かい違いには興味がないので詳しくは調べていないが、提案手法になぞらえて表現をすると以下のようにまとめて書けるとのこと。

$$L_4 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

ここで、m1 が SphereFace で m2 が ArcFace で m3 が CosFace の寄与となっている。
まあまあ些細な違いに見えなくもないが、SphereFace は学習が不安定で学習に結構工夫が必要だったりするのでちゃんと提案手法が良いとのこと。ただし後の実験の結果を見ても CosFace と比べてめちゃくちゃ優れているということはなさそう。

ということで比較をしてみたのが下図。
左側は ArcFace のみの結果で学習とともに {W_{y_i}, x_i} のなす角度の分布がどう変わっていったのかを示している。つまり正しいクラス中心となす角度の分布である。最初は W をランダムに選ぶので直交するようになっていて、学習とともに角度分布が移り変わっていく様子がみて取れる。0° まではいかずに 30° 前後くらいで収束しているようだ。
右側は target logit とクラス中心とのなす角をプロットしたもの。これはモデルの性能を表しているものではないので、まあ numerical にはそこまで違いはないですよ、くらいのもの。



(a) θ_j Distributions (b) Target Logits Curves

Figure 4. Target logit analysis. (a) θ_j distributions from start to end during ArcFace training. (2) Target logit curves for softmax, SphereFace, ArcFace, CosFace and combined margin penalty $(\cos(m_1\theta + m_2) - m_3)$.

マージンがどのように取られるかは手法によって振る舞いが異なり、ArcFace は角度そのものにマージンを加えているので、下図のように線形にマージンが確保されている。その方が良い、という類のものではないが、一応幾何学的な解釈としてはこのような違いがある、というもの。

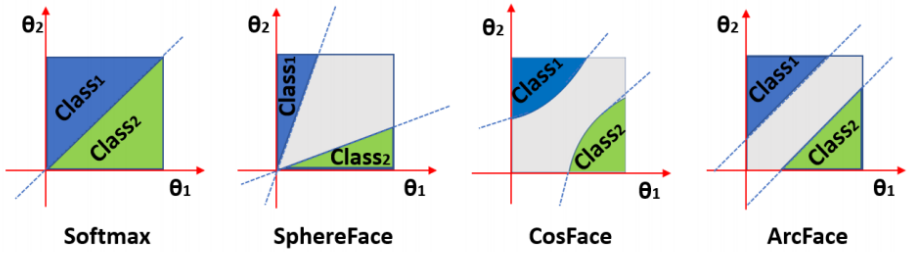


Figure 5. Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.

1

yoheikikuta commented on 7 Mar Author Owner

この論文は比較実験はたくさんしている。

SphereFace や CosFace などの先行研究の他にも、マージンなしで構築した loss function に以下の三つの要素を合わせたモデルも比較している。

- Intra-Loss
  クラス内で距離が大きくなるとペナルティになる項を足す。直感的にはクラス内の θ の平均値を足す、ということをすればよいし、実際にそうしている。
- Inter-Loss
  クラス間で距離が小さくなるとペナルティになる項を足す。GroundTruth の W_{y_i} とそうでないクラスの W_j のなす角度 (arccos(W_{y_i} W_j)) の和を loss から引くということにしてこれを実現している。
- Triplet-Loss
  普通の Triplet-Loss を角度空間で実施する。

yoheikikuta commented on 7 Mar Author Owner

実験を網羅的に説明するのはあまりやりたくないのでいくつか抜粋。

まずは典型的な Labeled Faces in the Wild (LFW) などの結果。
確かに提案手法が良い性能を示している。しかし CosFace なども同程度の性能を出しており、この辺が単体のモデルでの state-of-the-art と思っておくのがよいということだろう。

Loss Functions	LFW	CFP-FP	AgeDB-30
ArcFace (0.4)	99.53	95.41	94.98
ArcFace (0.45)	99.46	95.47	94.93
ArcFace (0.5)	**99.53**	**95.56**	**95.15**
ArcFace (0.55)	99.41	95.32	95.05
SphereFace [18]	99.42	-	-
SphereFace (1.35)	99.11	94.38	91.70
CosFace [37]	99.33	-	-
CosFace (0.35)	99.51	95.44	94.56
CM1 (1, 0.3, 0.2)	99.48	95.12	94.38
CM2 (0.9, 0.4, 0.15)	99.50	95.24	94.86
Softmax	99.08	94.39	92.33
Norm-Softmax (NS)	98.56	89.79	88.72
NS+Intra	98.75	93.81	90.92
NS+Inter	98.68	90.67	89.50
NS+Intra+Inter	98.73	94.00	91.41
Triplet (0.35)	98.98	91.90	89.98
ArcFace+Intra	99.45	95.37	94.73
ArcFace+Inter	99.43	95.25	94.55
ArcFace+Intra+Inter	99.43	95.42	95.10
ArcFace+Triplet	99.50	95.51	94.40

Table 2. Verification results (%) of different loss functions ([CASIA, ResNet50, loss*]).

それ以外にも MegaFace dataset はラベルの付け間違いが多かったので手動で付け直してその結果も載せたりしている。いやー頑張りますね。

その他にも色々なデータセットで比較しているが、ほとんどの場合で ArcFace が確かに良い性能を示している、という結果になっている。いいじゃん。

yoheikikuta commented on 7 Mar Author Owner

最後の方には大規模データに対して並列学習ができるかとか 512d の特徴量で十分に見分けられるか等を議論していて、どちらも結論は yes となっている。
ただこれは ArcFace そのものとはそこまで関係ない話なので割愛。論文のアイデア自体がめちゃくちゃ新しいものでもないし、いろんな実験などをしっかりとして論文として仕上げているという感じ。

yoheikikuta commented on 7 Mar Author Owner

ということで読み終わり。
純粋に分類問題として解く際に class 間をちゃんと区別するように工夫をしているという話（それ自体は SphereFace とか CosFace が取り組んできた内容ではあるけど）。

以前 closed set での予測に苦労したりもしてたのでそういうのにも使えそうだし試してみたい感じもする。
また、あくまで {x_i, y_i} という通常の分類問題の枠組みで使える話になっている。Triplet-Loss は例えば EC で画像検索してクリックされたもの同士を近くに配置するように学習する、などができるわけでそういう違いはあるよな。