深層学習

推論の信頼性

会津大学 コンピュータ理工学研究科 コンピュータ情報システム学専攻 髙橋輝

推論の不確かさ

入力$\mathbf{x}$から, 何らかの量yを予測する問題において, どのようド結果を信頼できるかを定量的に図ることができればうれしい.

そのためには, 確率分布$p(y|\mathbf{x})$を予測するようにすればよい.
$\mathrm{argmax_y p(y|\mathbf{x})}$を選べば一つの値を確定することができる.

多クラス分類では, 元々確率分布$p(\mathcal{C}_k|\mathbf{x})$を予測するようになっている.
だが, 回帰では, $\mathbf{x}$からyの値一つを予測するようになっている.
この場合, $p(y|\mathbf{x})$をパラメトリック(いくつかのパラメータを定めることで再現できる)確率分布でモデル化することで, 推論の信頼性を考えやすくなる.

問題点

このようにして得られた分布の一つの予測でしかないため, その推論がどれだけ信頼できるものかは不透明.

不確かさの種類

不確かさを発生原因に基づいて分類

- 偶然性による不確かさ
- 知識の欠如による不確かさ

例

1. ある関数 $y = f(x)$ に従って生成された集合 $\mathcal{D} = \{(x_n, y_n)\}_{n=1,\cdots,N}$ が与えられ, $\mathcal{D}$から元の関数を推定したいとする.
2. 今, 真の関数が$y = \sin(x)$であるとして, 私たちが観測できるのは, それにノイズを加えた$y = \sin(x) + \epsilon_x$となる.
   (ここで, ϵ_xは, xに依存し, 正規分布$N(0, 0.005x^2)$に従うとする.)
3. 観測されるデータ$\mathcal{D}$は, 入力の範囲$0 < x < 2\pi$についてのみ得られる.

1-3の条件下で得られる点集合は, 図 8.1(a)のようになる.
このようなデータ$\mathcal{D}$が与えられたとき, yとx間の関係を表す適当なモデルを導入し,
これをデータにフィットさせる.
このとき, 図 8.2(b)のような予測が行われる(青色の領域は予測の精度)
このような予測の不確かさが, 偶然性による不確かさである.
次に, 知識の欠如による不確かさについて考える.
この種類の不確かさが起こる第一の要因として, 訓練データ$\mathcal{D}$が$0 < x < 2\pi$の範囲でしか与えられていないことがある(1).
この範囲の外側, $x < 0$もしくは$x > 2\pi$の領域ではデータが存在しないので, 不確かな予測しかできない.

もう一つの要因は, モデルが十分な表現能力を持っているかということだ. モデルの表現能力が足りなければ, 予測は正確にならない(2).
文献[1]では, (1)の不確かさを"近似由来の不確かさ", (2)を"モデル由来の不確かさ"と呼んでさらに区別している.

クラス分類の場合でも, 回帰分析の場合と同様に, "偶然性による不確かさ"と, "知識の欠如による不確かさ"の区別が可能.

- 偶然性による不確かさ
  2つのクラスが重なり合って存在しているとき, 自信をもって境界線を定めることができない.
- 知識の欠如による不確かさ
  サンプルがまばらにしか存在しないとき, 予測はやはり不確かになる. これはデータが十分でないことに由来する予測の不確かさである.

偶然性による不確かさは, どう頑張っても減らすことができないものととらえることができ, 条件設定が同じ限り, どう頑張っても減らすことができない.
知識の欠如による不確かさは, データの追加や, モデルの表現能力の改善によって減少させることができる.

とはいっても, 特徴量を追加することで, 偶然性による不確かさを減らすこともできる.
(特徴量エンジニアリングが大事な理由)

不確かさの数理モデル

ベイジアンニューラルネットワーク

知識の欠如による不確かさの確率分布による評価

知識の欠如による不確かさが$\mathcal{D}$の不足によるものならば, 与えられた$\mathcal{D}$からパラメータ$\mathbf{w}$をどれだけ曖昧さなく決定できるかに関係する.
$p(\mathbf{w}|\mathcal{D})$の分布の形が鋭い形を持つなら, $\mathbf{w}$は曖昧さなく決定できる.
$\rightarrow p(\mathbf{w}|\mathcal{D})$は知識の欠如による不確かさを表現した確率分布となっている.

ここで, 以前に$p(y|\mathbf{x})$を予測することで, 推論の不確かさを捉えるアプローチを紹介したが, この確率分布では, 一つの$\mathbf{w}$を一つ指定して, 入力に対して出力が決まるため, 知識の欠如による不確かさは含んでいない.
そこで, $\mathcal{D}$から決定できる$\mathbf{w}$のばらつきを考慮し, それぞれの$\mathbf{w}$での予測結果を平均(周辺化)すると, "$\mathcal{D}$を前提にあらゆる$\mathbf{w}$の可能性を織り込んだyの予測"を

$$p(y|\mathbf{x}, \mathcal{D}) = \int_{\mathbf{w}} p(y|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathcal{D}) d\mathbf{w}$$

のように表現できる.
これは, 2種類の不確かさを統合したものを表す.
ベイジアンニューラルネットワークは, $p(\mathbf{w}|\mathcal{D})$を使って, $\mathbf{w}$を周辺化することによって, 推論の不確かさを定量化しようとする試みだ.

最尤推定との関係

知識の欠如による不確かさは$p(\mathbf{w}|\mathcal{D})$で捉えられるが, 具体的に求めるのは簡単ではない.
今, "仮に", $p(\mathbf{w}|\mathcal{D})$が手元にあったとする.
このとき, 最適なパラメータ$\mathbf{w}$を求めるためには,

$$\hat{\mathbf{w}} = \mathrm{argmax}_{\mathbf{w}} \log \mathrm{p}(\mathbf{w}|\mathcal{D})$$

のように$p(\mathbf{w}|\mathcal{D})$を最大化すればよい.
ベイズの定理により,

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathbf{w}, \mathcal{D})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

と書き直せるので,

上記の最大化は, 以下のように書き直せる.

$$\hat{\mathbf{w}} = \mathrm{argmin}_{\mathbf{w}}(-\log\ p(\mathcal{D}|\mathbf{w}) - \log\ p(\mathbf{w}) + \mathrm{const.})$$

ここで, $p(\mathbf{w})$は$\mathbf{w}$の事前分布である.
$-\log p(\mathcal{D}|\mathbf{w})$を$\mathbf{w}$について最小化することは, $\mathbf{w}$の最尤推定と一致する.
$-\log p(\mathbf{w})$は, 正則化を行うものと考えることができる.

ベイズ推定の立場から見れば, 重みの減衰項による正則化は, $\mathbf{w}$の事前分布$p(\mathbf{w})$に正規分布 $p(\mathbf{w}) \propto \exp(-\lambda/2||\mathbf{w}||^2) + \mathrm{const.}$を指定するのと同じ.

$$\log p(\mathbf{w}) = -\lambda/2||\mathbf{w}||^2\ + \mathrm{const.}$$

不確かさの予測

回帰の場合

分布$p(y|\mathbf{x}, \mathbf{w})$を予測するようにするには, ネットワークの損失関数を設計しなおす必要がある.
$p.3$で考慮したように, 予測すべき$p(y|\mathbf{x}, \mathbf{w})$を, 何らかのパラメトリックな確率分布, たとえば正規分布$y \sim N(\mu, \sigma^2)$でモデル化する. 正規分布は, μ, σで表現できるので, ネットワークはこれらを予測するように設計する.
(式は, 本に書いてあるものそのまま。口頭で解説します.)

学習は, これまで同様, 訓練データ$\{(\mathbf{x}_n, y_n)\}_{n=1,\cdots,N}$の尤度を最大化することで行う.
なお, σはσ^2の形でしか尤度の式で現れないこと, またσ^2が非負であることを考慮して, $s(\mathbf{x};\mathbf{w}) \equiv \log \sigma^2$を新たに定義し, σのかわりに, これをネットワークの出力とする.

クラス分類の場合

ネットワークは元々各クラスkの事後確率$p\left(y=\mathcal{C}_k\middle|\mathbf{x},\mathbf{w}\right)$を予測するように設計されており, そのままそれを偶然性による不確かさの予測に利用できる.
また、分類はy_kが最大となるクラスになされるが, その最大値$\max_k y_k$は, 分類したクラス数の事後確率の予測値なので, この推論の確信度として解釈できる.
しかし, 確信度は本来あるべき値より高くなる傾向があることが, 経験的に知られている.
確信度が, "その分類結果が実際に正解である確率"であると考えると, ある分類結果の確信度がその分類が正しい確率と一致する条件は,

$$p\left(\mathrm{argmax}_k y_k=\bar{k}\,\middle|\,\max_k y_k=q\right)=q$$

と表現できる.
ただし, $\bar{k}$は正しいクラスとする.

このような一致・不一致の度合いを測る尺度に, **期待校正誤差**(expected calibration error, ECE)を用いる.

ECEの計算方法

N個のテストサンプルがある時, まずそれら各々の分類結果の確信度の度数分布を作る.
確信度の範囲$[0, 1]$をM分割したとき, その各ビン$(m = 1, \cdots, M)$に含まれるサンプルの集合B_mについて, その平均正答率$\mathrm{acc}(B_m)$を算出する. B_mに含まれるサンプルの確信度は, およそ$\mathrm{conf}(B_m) = (m - 1/2)/M$となるはずである.
ECEは, これらの差を

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$$

のように求めたものである.

クラス分類の予想が自信過剰となるネットワークでは, ECEの値は大きくなる.
これを修正するために, ソフトマックス関数に**温度スケーリング(temperature scaling)** を施した下記の式を用いる.

$$y_k = \frac{exp(u_k/T)}{\sum_{j=1}^{K} exp(u_j/T)}$$

これが, Validation Data上で最小になるように温度Tを決定する. これらの操作をソフトマックス関数の**校正(calibration)** と呼び, ソフトマックスの出力を鋭くなるようにしたり, 逆によりフラットにしたりすることができる.

モデルのアンサンブル

同一のタスクと訓練データ$\mathcal{D}$に対し, 何らかの方法で複数のモデルを得て, それらの推論結果を統合すると, 一般に推論の精度を向上させることができる.
たとえば, 初期重みのみを変えた同一構造のネットワークを複数訓練し, 複数モデルを得て, 同一入力に対するこれらの推論結果を統合することが考えられる.(Random seed average)

今までやってきた不確かさの予測方法を用いても, 知識の欠如による不確かさを推定することはできない. 以前の議論により, $\mathbf{w}$を$\int p(y|\mathbf{x},\mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$と周辺化すれば, 2つの不確かさを統合した確率分布を得られるが, この計算は簡単には実行できない.
そのため, モデルのアンサンブルを使って, 上の積分の近似を求める.
クラス分類の場合, クラスの確率の平均を与える.
回帰の場合, 正規分布の平均と分散を予測させ, 各モデルの予測した分布を平均する.

この平均を, $\int p(y|\mathbf{x},\mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$の近似と考える.
直感的には, 各モデルについて十分な学習が行われており, どのネットワークもほぼ同じ結果を与えるなら, 分布の平均はピークが高く, 予想は確かなものとみなせる.
一般に, 比較的少数のモデル(5つ程度)で良い結果が得られるが, それでも単一モデルと比べると計算コストがかかる.
改善策として, MC-Dropoutという手法を使うことができる.
これは, ドロップアウトを学習だけでなく推論においても行う方法である.
ただし, ドロップアウトのハイパーパラメータをうまく選ぶ必要があり, 常にアンサンブルの代わりになるとは言えない.

不確かさの尺度

クラス分類の不確かさは, $p(y|\mathbf{x})$の広がりを示すクロスエントロピーでも測れる.
エントロピーは, クラスの事後確率$y_k = p(\mathcal{C}_k|\mathbf{x})$を用いて,

$$H[p(\mathcal{C}_k|\mathbf{x})] = -\sum_{k=1}^{K} y_k \log y_k$$

と計算される. 回帰の場合, エントロピーは次のように計算される.

$$H[p(y|\mathbf{x})] = -\int p(y|\mathbf{x}) \log p(y|\mathbf{x}) dy$$

アンサンブルを使うときはこの他にも, 各モデルのソフトマックス出力のばらつきや, 各モデルが独立に予測したクラスがどれだけそろっているかを測る**変動比**が使われることがある.

$$v = 1 - \frac{m}{I}$$

ここでmは, I個の予測クラスのうち, 最大多数となったクラス以外の数である.

2つの不確かさの分離

モデルのアンサンブルを用いれば, 知識の欠如による不確かさは捉えられるが, そこで得られるのは, 偶然性による不確かさと知識の欠如による不確かさをトータルした不確かさだ.

2つの不確かさを分離して評価する方法の一つは, 偶然性による不確かさを求め, それを差し引くことだ.

$\mathbf{w}$を1つ指定したときの偶然性による不確かさは, $p\left(y|\mathbf{x}, \mathbf{w}\right)$についてのエントロピーで表現できる.

$\mathbf{w}$の取りうる変動を表す$p(\mathbf{w}|\mathcal{D})$が与えられ, このエントロピーの$p(\mathbf{w}|\mathcal{D})$に関する期待値

$$\mathbb{E}_{p(\mathbf{w}|\mathcal{D})} H\big[p(y|\mathbf{x}, \mathbf{w})\big]$$

が計算できるなら, トータルの不確かさ$H\big[p(y|\mathbf{x})\big]$からこれを引いた,

$$H\big[p(y|\mathbf{x})\big] - \mathbb{E}_{p(\mathbf{w}|\mathcal{D})} H\big[p(y|\mathbf{x}, \mathbf{w})\big]$$

が, 知識の欠如による不確かさを表すと考えられる.

分布外入力の検出

概要

$\mathcal{D}$の入力の集合$\{\mathbf{x}\}$が従う確率分布を$p(\mathbf{x})$とするとき, $p(\mathbf{x})$とは異なる分布から生成された$\mathbf{x}$のことを, **分布外(out-of-distribution, OOD)のサンプル**と呼ぶ.
逆に, $p(x)$から生成された$\mathbf{x}$のことを, **分布内(in-distribution, ID)のサンプル**と呼ぶ.
これらを区別する最も自然な方法は, 予測の不確かさを使う方法だ.
入力$\mathbf{x}$に対する出力$y_k = p(\mathcal{C}_k|\mathbf{x}) = \mathrm{softmax}_k(u_1, \cdots, u_K)$から, 予測の不確かさを確信度で測るとすると, $\max_k y_k$に対して, 適用な閾値τを設定し,

$$\max_{k=1,\ldots,K} y_k < \tau$$

となる場合にIDと判定する.
(基本はこれなのであとは口頭で説明)

敵対的事例

DNNは極めて高精度な推論を行える一方, 入力を少し細工するだけでその推論を誤らせることが可能.
あるCNNが正しく認識できる画像に, 後述する方法で計算したノイズを加算すると, 同じCNNが誤った推論結果を与えるようになる.
このような細工がなされた入力のことを, **敵対的事例(adversarial example)** と呼ぶ.
一般にクラス分類だけではなく, 回帰を含む様々な事例で, 一般に敵対的事例を作ることができる.

Toy Example

https://kennysong.github.io/adversarial.js/

標識画像の認識タスクにおいて, 敵対的事例を作成する.

これはノイズを付与することで, 認識を誤らせるものなので, 現実世界での攻撃には使えない.

Tesla Model Sに対する敵対的事例

https://youtu.be/6QSsKy0I9LE?t=71

路面上に配置された非常に小さなステッカーにより, Tesla Model Sのレーンディテクタが車線情報を誤認し, 反対車線に侵入した.

敵対的事例の生成

入力$\mathbf{x}$に微小な摂動δを加算した$\mathbf{x}+\delta$をネットワークに入力したとき, $\mathbf{x}$の正解クラスとは違うクラスを予測するように, δを決定することを考える.
また, ここで, 対象とするネットワークの構造とパラメータは完全にわかってるものとすると, この条件の下での攻撃を, **ホワイトボックス攻撃**という.
まず, ネットワークは, $E\left(\mathbf{w}, \mathbf{x}, k_{true}\right)$を最小化するいつも通りの訓練を行ったものとする.

このとき, 入力$\mathbf{x}+\delta$に対する予測クラスを, 誤った予測クラスk_{adv}に分類させたいので,

$$\min_{\delta,\lambda} E(\mathbf{w}, \mathbf{x}+\delta, k_{adv}) + \lambda|\delta|_1$$

の最小化を考える.
なお, $\mathbf{x}+\delta$が画像の濃淡の範囲に収まる条件も一緒に課す.
このようにして求めたδを用いることにより, CNNに誤認させることが可能となる.

推論を誤らせるだけなら, 損失が最も増加する方向に摂動を加える, **FGSM(Fast Gradient Sign Method)** と呼ばれる方法を用いることができる.
これは, $\mathbf{x}+\delta$を

$$\mathbf{x} + \epsilon \mathrm{sign}(\nabla_{\mathbf{x}} E(\mathbf{w}, \mathbf{x}, k_{true}))$$

と定めることによってできる

反復的FGSM or 射影勾配降下法(Projected Gradient Descent, PGD)

$$\mathbf{x}^{t+1} = \mathrm{Clip}_{\mathbf{x},\epsilon}[\mathbf{x}^t + \alpha \mathrm{sign}(\nabla_{\mathbf{x}} E(\mathbf{w}, \mathbf{x}^t, k_{true}))]$$

より絶対値の小さな摂動($\pm\epsilon$)を用いて, 推論を誤らせることができる.
この方法は, 攻撃者が入力に対する損失の勾配を知りうる条件の下では, 最も強力なもの.

転移可能性

一つのネットワークに対して成功した攻撃事例は, ネットワーク構造が近ければ近いほど, ほかのネットワークに対しても成功する可能性が高くなるということ

.

防御

PGDが最も強力な攻撃であるという前提に立ち, 敵対的事例に欺かれにくいネットワークのパラメータを取る学習方法を考える.
通常の学習では, 単純に損失関数をパラメータ$\mathbf{w}$のみについて最小化していたが, このままでは,入力の摂動に対して弱いネットワークが生成されてしまう.
この代わりに, 摂動を加えた場合の損失の最大値を考え, その最大値を最小化することを考える.

つまり,

$$\min_{\mathbf{w}} \rho(\mathbf{w}), \text{where} \quad \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\delta\in\mathcal{S}} E(\mathbf{w}, \mathbf{x}_n+\delta, d_n)\right]$$

この式の最適化をどのように行うかだが, 一つの方法として, PGDを使って, $\mathcal{D}$の各サンプルに対して敵対的事例を生成し, それを訓練データに追加して学習を行うことができる.
この方法を**敵対的学習(adversarial training)**と呼ぶ.

品質保証の取り組み

~~ちからつきた~~

NNの推論について, 品質を保証しようという取り組みがある.
e.g. ある一定の範囲内で変動する任意の入力について, 推論結果の正しさを保証する.

深層学習で扱うデータは, 画像や言語のように高次元空間にあり, 網羅的に入力を変えてテストを行うことは難しい.

一例

入力に摂動を加えた時, 推論がどうなるかを予想する.
画像に微小な幾何学的変化を加えた時に, 推論結果が変化しない頃を証明することができれば, 実用上の価値がある.
$\rightarrow$推論結果が変わらない範囲を特定しようという試み.
DeepPoly, DeepG

ニューロン網羅率　$\rightarrow$ どれだけニューロンが発火したか.
↑
これに何の意味があるというのかね...