

※自身が知らなかった知識を中心にレポートをまとめている

深層学習 day1

○Section1:入力層～中間層

- ・大きく識別モデルと生成モデルに分けられる
ex.) 犬と猫のデータを識別、犬らしい画像を生成
- ・識別モデル：高次元 → 低次元
生成モデル：低次元 → 高次元
ー応用例：画像の超解像など
- ・識別器の開発の3つのアプローチ
(「パターン認識と機械学習」での取り上げ方)
ー生成モデル
ー識別モデル：確率的な識別
ー識別関数：決定的な識別、確率や間違いの精度はわからない
- ・W重みとbバイアスがパラメータ
- ・ニューラルネットワークが主に扱う問題：回帰、分類

- ・入力 x , 重み w , バイアス b , 総入力 u , 活性化関数 f , 出力 z
- ・ w は傾き、 b は切片のイメージ

$$u = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$$
$$= Wx + b$$

○Section2:活性化関数

- ・線形関数
：加法性、斉次性を満たす
- ・中間層用の活性化関数
ーReLU関数：勾配消失問題、スパース化に貢献
ーシグモイド関数：勾配消失問題を引き起こすことがある
ーステップ関数など
- ・出力層用の活性化関数：
ーソフトマックス関数
ーシグモイド関数など

○Section3:出力層

- ・出力層の役割：確率を出力する、信号の比率をそのまま変換
Cf.) 中間層：次の入力に適切なもの、信号の強弱を調整
- ・ニューラルネットワークの学習
ーニューラルネットワークの出力と正解データを比較
→どのくらい合っていたかを誤差関数で表現し、その誤差からパラメータを学習する
- ・誤差関数：平均二乗誤差（二乗和誤差）などが使われる

微分した時に都合がよいように $1/2$ がついている

- ・ 分類問題ではクロスエントロピー誤差、
回帰問題では平均二乗誤差を用いることが多い
- ・ 出力層の活性化関数
- ・ 二値分類：シグモイド関数
多クラス分類：ソフトマックス関数（出力の合計が1になる）
回帰問題：恒等関数

○Sentence4:勾配降下法

- ・ 誤差関数を最小にするネットワークを作成
→ 勾配降下法を利用してパラメータを最適化
- ・ 学習率が大きすぎると、誤差関数の最小値にいつまでも辿り着かず発散する
小さい場合発散することはないが、収束するまでに時間がかかる
また、最小値ではなく極小値になることがある
- ・ 勾配降下法のアルゴリズム
→ Adam, Momentum など
- ・ 確率的勾配降下法 (SGD)
→ ランダムに抽出したサンプルの誤差を求める
- ・ 計算コストの削減、極小解に収束するリスクの軽減、オンライン学習ができる
- ・ オンライン学習
→ モデルに都度データを入れていく学習方法
→ 一度に全データを用意しなくてもよい

cf) バッチ学習

- ・ ミニバッチ勾配降下法
→ ランダムに分割したデータの集合（ミニバッチ）に属するサンプルの平均誤差
ex. 10万枚の画像を 500 枚ずつに分割し、小分け（ミニバッチ）にして学習させる
2000 個のバッチができるので、最後に 2000 で割って $(1/N)$ 平均誤差を求める
 - ・ 確率勾配降下法のメリットを損なわずに、計算機の計算資源を有効活用できる
ミニバッチごとに学習することで、計算の並列化ができる
(Single Instruction Multi Data:SIMD 並列化)
cf. 確率的勾配降下法だと、10万枚の画像からランダムに抽出する
- ※エポックを重ねるごとにパラメータに関する損失関数の最小値に近づいていく

○Section5:誤差逆伝播法

- ・ 勾配の計算方法：数値微分
各パラメータそれぞれに数値微分を行うと計算量が多くなる
→ 誤差逆伝播法を利用する

- ・ 誤差逆伝播法：パラメータの微分値を解析的に計算する手法
計算結果（誤差）から微分を逆算する（微分の連鎖率）
- ・ ディープラーニングの開発環境
： CPU, GPU, FPGA, ASIC(TPU)
- ・ 入力層の設計
 - ー入力データ、欠損値の扱い、データの結合、正規化正則化等の工夫
- ・ 入力として取るべきではないデータ
 - ： 欠損値が多い、出力を加工した情報、連続性のないデータなど
- ・ データセットの拡張：学習データが不足する時に人工的にデータを水増しする手法
- ・ 画像認識の分類タスクに効果が高い
- ・ ノイズ注入によるデータ活用
 - ： 中間層へのノイズ注入で様々な抽象化レベルでのデータ拡張が可能
- ・ 転移学習
 - ーファインチューニング：学習済みモデルの重みを含め再学習
 - ー転移学習：学習済みモデルの重みを固定