

宿題 2

MNIST を識別する MSE 識別器を実装する。

学習は LMS 法によって行った。学習率を 0.007, ミニバッチサイズを 100, エポックを 500 とした。訓練データのうち 10,000 個を validation 用に分け, 学習状況の確認に利用した。

混同行列は表 1 のようになり, また, 各カテゴリごとの正解率等は表 2 のようになった。

表 1: MSE に対する混同行列

	0	1	2	3	4	5	6	7	8	9
0	957	0	0	2	0	6	7	2	6	0
1	0	1101	3	1	3	1	5	1	20	0
2	33	51	802	27	21	1	32	20	40	5
3	9	18	24	875	7	19	4	19	25	10
4	1	16	7	1	895	3	8	1	10	40
5	34	16	3	74	25	656	16	16	36	16
6	40	9	12	0	26	17	843	0	11	0
7	5	39	11	7	26	1	0	890	4	45
8	20	43	7	23	31	34	13	13	771	19
9	19	11	2	12	85	1	0	83	6	790

表 2: MSE に対する各カテゴリごとの結果

Category	#Data	#Correct	Accuracy
0	980	957	0.977
1	1,135	1101	0.970
2	1,032	802	0.777
3	1,010	875	0.866
4	9,82	895	0.911
5	8,92	656	0.735
6	9,58	843	0.880
7	1,028	890	0.866
8	9,74	771	0.792
9	1,009	790	0.783
All	10,000	8,580	0.858

プログラムは??ページの Listing ??に示した。その説明を以下に簡単に記す。なお、同じプログラムで宿題 3 の実行もできる。

- `load_data`
MNIST データを読み込んで返す関数。04-23 の課題で用いた `mnread` モジュールを用いている。
- `add_augment_axis`
入力を拡張ベクトルにする関数
- `normalize`
入力を正規化する関数。今回は `[0, 256]` のグレースケールが対象だったため、簡単に、各要素に対して 128 を引いてから 256 で割る操作を行っている。
- `split`
入力を与えられたバッチサイズごとに分ける関数
- `train`
訓練を行う関数。`mode` 引数で `MSE` か `MLP` かを切り替え可能にしている。ミニバッチに対して順伝搬を行い損失を計算したのち、勾配を計算してパラメータを更新する。`MLP` を用いる場合は、勾配の計算に誤差逆伝搬法が用いられる。
- `test`
テストデータに対する結果を求める関数
- `print_result_in_TeX_tabular_format`
混同行列と各ラベルに対する精度を、`TeX` の表の形式で出力する関数
- `compute_loss`
損失を計算する関数
- `Linear`
`MSE` 用の線形モデルを表したクラス。勾配計算とパラメータの更新を行うメソッドを持つ。