

第 1 章

考察

1.1 「漏斗型」ニューラルネットワークにおけるパラメータ数の計算

実装の節??において、隠れ層のユニット数を順方向に $n_{\text{out}}a^N, n_{\text{out}}a^{N-1}, \dots, n_{\text{out}}a^2, n_{\text{out}}a$ のように指数関数的に減らしていく「漏斗型」ニューラルネットワークを考案した。ただし N は隠れ層の層数で、 n_{out} は出力層のユニット数、 a は自然数である。 n_{in} を入力層のユニット数として、このネットワークでの層パラメータの数 N_{params} を求める。

全結合型のネットワークであるとすれば、図 1.1 のようにユニット数 x の層 1 の出力がユニット数 y の層 2 の入力として通るとすると、重みの数は xy だけある。さらに層 2 の各ユニットはバイアスを 1 つ持っているので、このとき総パラメータ数は $xy + y$ だけある。

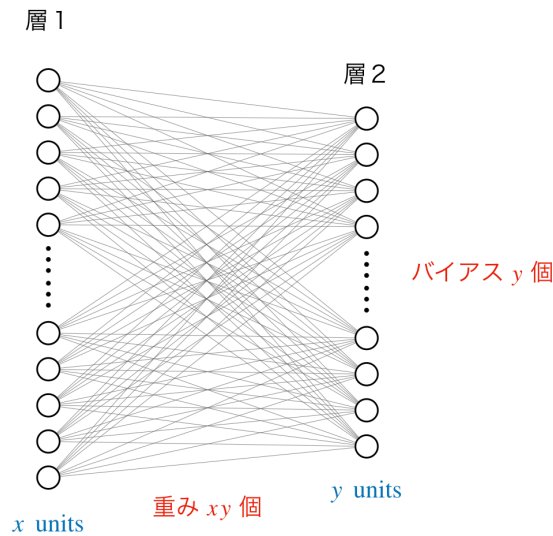


図 1.1 パラメータの数は $xy + y$

同様に考えて、まずユニット数 n_{in} の入力層とユニット数 $n_{\text{out}}a^N$ の第 1 隠れ層の間のパラメータ数は $n_{\text{in}} \times n_{\text{out}}a^N + n_{\text{out}}a^N$ であり、第 1 隠れ層と第 2 隠れ層の間のパラメータ数は $n_{\text{out}}a^N \times n_{\text{out}}a^{N-1} +$

$n_{\text{out}}a^{N-1}$ である。これを出力層まで辿って足してゆくと、

$$\begin{aligned} N_{\text{params}} &= (n_{\text{in}} \times n_{\text{out}}a^N + n_{\text{out}}a^N) + (n_{\text{out}}a^N \times n_{\text{out}}a^{N-1} + n_{\text{out}}a^{N-1}) \\ &\quad + \cdots + (n_{\text{out}}a^1 \times n_{\text{out}}a^0 + n_{\text{out}}a^0) \\ &= n_{\text{in}}n_{\text{out}}a^N + n_{\text{out}}^2 \sum_{k=1}^N a^{2k-1} + n_{\text{out}} \sum_{k=0}^N a^k \\ &= n_{\text{in}}n_{\text{out}}a^N + n_{\text{out}}^2 a \cdot \frac{a^{2N} - 1}{a^2 - 1} + n_{\text{out}} \frac{a^{N+1} - 1}{a - 1} \end{aligned}$$

となる。 $n_{\text{in}} = 784$, $n_{\text{out}} = 10$, $a = 2$, $N = 6$ として計算すると $N_{\text{params}} = 776030$ となり、確かに表??で得られた結果と一致する。

1.2 より精度の高いフィードフォワード・ニューラルネットワークの構築

第??章の結果から、フィードフォワード・ニューラルネットワークにおいては以下の状況のときに性能が良くなることがわかった。

- 活性化関数：ReLU
- コスト関数：交差エントロピー
- 最適化アルゴリズム：Adam
- エポック数：100
- ミニバッチのサイズ：100
- アーキテクチャ：隠れユニット数が出力層に向かって半減していくような「漏斗型」の構造

この設定のニューラルネットワークを作り同様に学習させたところ、次の結果を得た。

```
Test loss: 0.15242414196143
Test accuracy: 0.986600008010864
Computation time:434.38 sec
```

学習に 7.5 分かかったが、これまでのフィードフォワード・ニューラルネットワークで最も高い正解率を出している。

1.3 より精度の高い畳み込みニューラルネットワークの構築

節??の畳み込みニューラルネットワークに関する一連の実験により、以下の状況のときに性能が良くなることがわかった。

- フィルターのサイズ： 3×3 から 5×5 程度の小さいフィルターを用いるとよい
- プーリング
 - － 畳み込みと併用する
 - － 平均プーリングではなく最大プーリングのほうがよい
 - － プーリングカーネルのサイズは 2×2 など小さいものがよい
- ドロップアウト

- ドロップアウト率は $p = 0.25$ 程度がよい
- ユニット数が大きい（パラメータ数の大きい）層の間で用いるとよい
- ただしドロップアウトをしすぎると学習が遅くなることがある
- アーキテクチャ：畳み込みのフィルター数が出力に向けて大きくなってゆくような「末広がり」のネットワーク

以上のことを踏まえて、図 1.2 のような畳み込みニューラルネットワークとフィードフォワード・ネットワークを組み合わせたネットワークを構築し、学習をさせた。入力が 32×32 になるようにゼロパディングを施し、16 枚の 5×5 のフィルターを用いて畳み込み層 C1 を生成した。次に 2×2 の最大プーリングを行ってプーリング層 S2 を得た。同様に 32 枚の 5×5 の畳み込み、 2×2 の最大プーリングを行ったのち 800 次元のベクトルに展開した（Flatten 層 C5）。ここから先は通常のフィードフォワード・ネットワークであるが、ユニット数が 25 になるまでユニット数を半減させていった（C5～F10）。また F6-F7 の間と F7-F8 の間でドロップアウト率 $p = 0.25$ のドロップアウトを行なった。

活性化関数は出力層以外はすべて ReLU を使い、出力層はソフトマックス関数を用いた。コスト関数は交差エントロピー、最適化アルゴリズムには Adam を採用した。エポック数は 20、ミニバッチのサイズは 1000 とした。

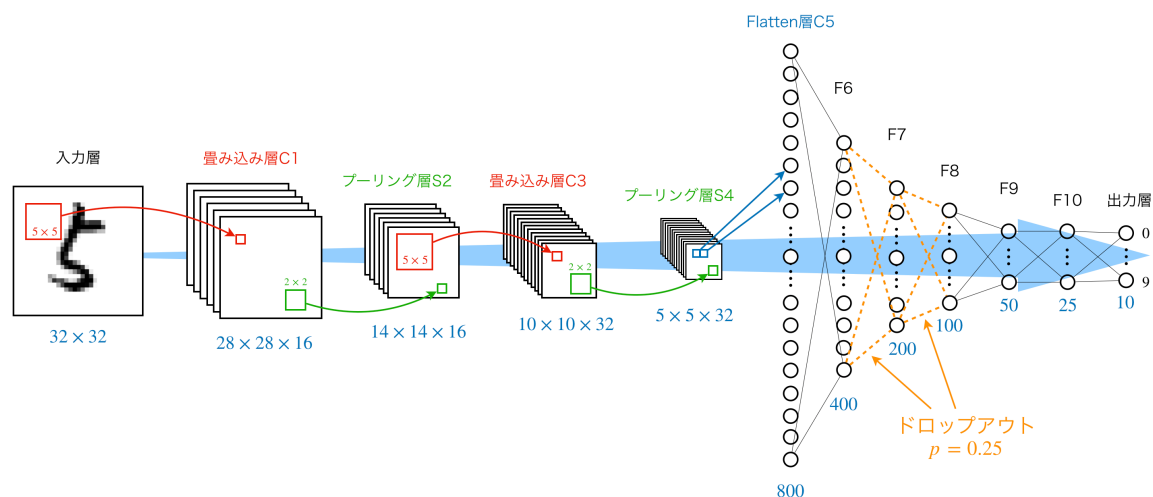


図 1.2 集大成のニューラルネットワーク

図 1.2 のネットワークに学習を行わせ、結果を表示すると次のようになった。

Test loss: 0.02598600876271721

Test accuracy: 0.9923

Computation time: 455.795 sec

実行時間は 7.5 分と短くはないがそこまで長くはない。また正解率は 99.23% となり、これまでで最高の正解率を得た。念のため学習曲線を表示すると図 1.3 となった。これにより過学習はしておらず、正しく学習がなされていることがわかる。

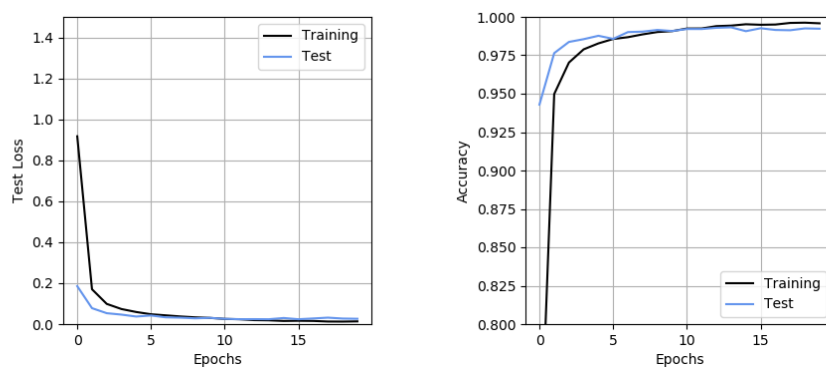


図 1.3 「集大成のニューラルネットワーク」の学習曲線