第 1 章

脳のモデルとしての ニューラルネットワーク

1.1 Spiking モデル

ホジキンハクスレーの式

1.1.1 LIF model

$$\tau \frac{dV_m(t)}{dt} = -V_m(t) + RI(t)$$

ただし、$\tau = RC$ です。静止膜電位を考慮する場合は、定数項 V_{rest} を加えるとよいです。これをオイラー近似で離散化し、

$$V_{m,t} = \frac{\Delta t}{C} I_t + \left(1 - \frac{\Delta t}{\tau}\right) V_{m,t-1}$$

となります。V_m が一定の閾値 V_{th} を超えるとニューロンは発火し、膜電位はリセットされて静止膜電位に戻ります。この実装は $V_{m,t} > V_{\text{th}}$ なら出力 $y_t = 1$ のように if 文を用いたものが多くみられますが、ここでは if 文を用いない実装を紹介します。

この手法は $y_{t-1} = 1$ なら膜電位がリセットされるように $(1 - y_{t-1})$ を膜電位に乗じます。

$$\begin{aligned} V_{m,t} &= \frac{\Delta t}{C} I_t + \left(1 - \frac{\Delta t}{\tau}\right) V_{m,t-1} \cdot (1 - y_{t-1}) \\ y_t &= f\left(V_{m,t} - V_{\text{th}}\right) \end{aligned}$$

ただし、$f(\cdot)$ はステップ関数で、

$$f(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

となります。この実装は元々 Spiking Neural Unit (SNU) というアーキテクチャで考案されたものです。SNU は普通のニューラルネットワークのフレームワークで SNN を取り扱うためのユニットで、これを用いると簡単に誤差逆伝搬法で SNN を学習させることができます。厄介なのはステップ関数を微分するとデルタ関数となる点ですが、疑似勾配として tanh の微分を用いることで解決しています。

さて、それでは SNN を実装してみましょう。

SNN.py

```
1  /* ここにはソースコードを書く */
2  #include<stdio.h>
3
4  int main(void)
5  {
6    printf("Hello, World!\n");
7    return 0;
8  }
9  /* breakable を付けるとこんな感じで改行にも対応できる */
```

ここで入力としてポアソンスパイクを用いています。つまり入力ニューロンがポアソン過程に従って発火するという場合を考えています。λ を発火率とした場合、区間 $[t, t+dt]$ の間にポアソンスパイクが発生する確率は λdt ですが、このことを簡単に示しておきます。事象が起こる確率が強度 λ のポアソン分布に従う場合、時刻 t までに事象が n 回起こる確率は $P[N(t) = n] = \dfrac{(\lambda t)^n}{n!} e^{-\lambda t}$ となります。よって、微小時間 dt において事象が 1 回起こる確率は

$$P[N(dt) = 1] = \frac{\lambda dt}{1!} e^{-\lambda dt} \simeq \lambda dt + o(dt)$$

となります。ただし、$o(\cdot)$ はランダウの記号であり、$e^{-\lambda dt}$ についてはマクローリン展開による近似を行っています。また、微小時間 dt の間はスパイクが 2 回以上生じないという仮定をおいています。これらのことから、一様分布 $U(0, 1)$ に従う乱数 ξ を取得し、$\xi < \lambda dt$ なら発火 $(y = 1)$, それ以外では $(y = 0)$ となるようにすればポアソンスパイクを実装できます。

1.2 発火率モデル

いよいよ NN を見ていきます。今言われている典型的なニューラルネットワークは「発火率モデル」というものです。

まず、初めにマカロック-ピッツ (McCulloch-Pitts) から見てみます。ANN は脳のようであると言われているが実際にはそうではない。発火率モデルとして捉えられる。ので、ミクロに見ればスパイクを出してはいないが、少しマクロにニューロンの活動のダイナミクスを見ると一致している。

活性化関数

活性化関数はニューロンの発火率における I/O 関係を表している。要はゲイン。低頻度での発火であれば、ReLU で近似できる。

1.2.1　RNN

微分方程式で書くと、

$$\tau \frac{d\boldsymbol{r}}{dt} = -\boldsymbol{r} + f(W^{\mathrm{rec}}\boldsymbol{r} + W^{\mathrm{in}}\boldsymbol{u} + \boldsymbol{b}^{\mathrm{rec}} + \boldsymbol{\xi})$$

これを first-order Euler approximation を用いて離散化 (time step Δt) すると、

$$\boldsymbol{r}_t = (1-\alpha)\boldsymbol{r}_{t-1} + \alpha f(W^{\mathrm{rec}}\boldsymbol{r}_{t-1} + W^{\mathrm{in}}u_t + \boldsymbol{b}^{\mathrm{rec}} + \boldsymbol{\xi}_t)$$

Dale の原理を守った RNN

Dale の法則. この法則は現在は修正されていますが、それでも

1.3 スパイクモデルと発火率モデル

1.3.1　LIF から発火率モデルへの変換

これは数理的な話。LIF ニューロンの発火率が rate $\sim \left(\tau \ln \frac{RI}{RI - V_{\mathrm{th}}}\right)$ と近似できることを示します。つまり I/O の描画。

$t = t_1$ にスパイクが生じたとします。このとき、膜電位はリセットされるので $V_m(t_1) = 0$ です。$[t_1, t]$ における膜電位は LIF の式を積分することで得られます。ただし、一定な

入力が持続していることを仮定します。

$$\tau \frac{dV_m(t)}{dt} = -V_m(t) + RI(t)$$

の式を積分すると、

$$\int_{t_1}^{t} \frac{\tau dV_m}{RI - V_m} = \int_{t_1}^{t} dt$$
$$\ln(1 - \frac{V_m(t)}{RI}) = -\frac{t - t_1}{\tau} \quad (\because V_m(t_1) = 0)$$
$$\therefore \quad V_m(t) = RI \left[1 - \exp\left(-\frac{t - t_1}{\tau}\right)\right]$$

となります。$t > t_1$ における初めのスパイクが $t = t_2$ に生じたとすると、そのときの膜電位は $V_m(t_2) = V_{\text{th}}$ です（実際には閾値以上となっている場合もあるますが近似します）。$t = t_2$ を上の式に代入して

$$V_{\text{th}} = RI \left[1 - \exp\left(-\frac{t_2 - t_1}{\tau}\right)\right]$$
$$\therefore \quad T = t_2 - t_1 = \tau \ln \frac{RI}{RI - V_{\text{th}}}$$

となります。ここで T は 2 つのスパイクの時間間隔です。$t_1 \leq t < t_2$ におけるスパイクは $t = t_1$ 時の 1 つなので、発火率は $1/T$ となります。よって

$$\text{rate} \sim \left(\tau \ln \frac{RI}{RI - V_{\text{th}}}\right)$$

です。不応期 Δ_{abs} を考慮すると、持続的に入力がある場合は単純に Δ_{abs} だけ発火が遅れるので発火率は $1/(\Delta_{\text{abs}} + T)$ となります。

1.3.2　発火率モデルから LIF への変換

SNN は学習が難しいが。

1.4 STDP 則による SNN の学習

1.5 これは section

我輩は猫である[*1]。

[*1] こんな感じで脚注を書く

```
1  /* ここにはソースコードを書く */
2  #include<stdio.h>
3
4  int main(void)
5  {
6    printf("Hello, World!\n");
7    return 0;
8  }
9  /* breakable を付けるとこんな感じで改行にも対応できる */
```

```
## ここにはコマンドを書く
$ echo "Hello, World!"
```

図表はキャプションを付けたときに、先頭に「▲」や「▼」を付けるようにした。

▼ 表 1.1　表のサンプル

日本	hoge	fuga	piyo
アメリカ	foo	bar	baz

これはコラム

コラムも随時挟めるようにした。
tcolorbox は title を指定するといい感じにタイトル付きの枠で囲ってくれる。

参考文献

1 章

[1] S. Woniak, et al. “Deep learning incorporating biologically-inspired neural dynamics”. (2018). https://arxiv.org/abs/1812.07040

[2] Gerstner, W. and Kistler, W. M. (2002). Spiking Neuron Models. Single Neurons, Populations, Plasticity. Cambridge University Press. https://icwww.epfl.ch/~gerstner/BUCH.html