（強化学習のための？）深層学習メモ

菱沼　徹

2020 年 3 月 6 日

強化学習をやる際，深層学習についてだいたい次が分かればいいらしい [1] ので，それぞれメモしていく．

- architectures (MLP, vanilla RNN, LSTM, GRU, conv layers, resnets, attention mechanisms)
- common regularizers (weight decay, dropout)
- normalization (batch norm, layer norm, weight norm)
- optimizers (SGD, momentum SGD, Adam, others)
- reparameterization trick

表記の単純化のため，（できるだけ同じになるように努めるが）記号の定義は各節のみにおいて有効とする．

1　入門：多層パーセプトロン (MLP) を用いた回帰

1.1　２層パーセプトロン

■2 層 NN 順伝播．　n 番目のデータに対して，入力 $\boldsymbol{x}_n \in \mathbb{R}^{H_0}$，出力 $\boldsymbol{y}_n \in \mathbb{R}^D$，重みパラメータ $\boldsymbol{W}^{(1)} \in \mathbb{R}^{H_1 \times H_0}$，$\boldsymbol{W}^{(2)} \in \mathbb{R}^{D \times H_1}$，活性化関数 $\phi(\cdot)$，正規ノイズ $\boldsymbol{\epsilon}_n \mathbb{R}^D$ として，次のモデルを考える．

$$y_{n,d} = \sum_{h_1=1}^{H_1} w^{(2)}_{d,h_1} \phi \left(\sum_{h_0=1}^{H_0} w^{(1)}_{h_1,h_0} x_{n,h_0} \right) + \epsilon_{n,d}.$$

あるいはベクトル表記で

$$\boldsymbol{y}_n = \boldsymbol{W}^{(2)} \boldsymbol{\phi}.\left(\boldsymbol{W} \boldsymbol{x}_n\right) + \boldsymbol{\epsilon}_n$$

ここで，$\boldsymbol{\phi}.(\cdot)$ は要素ごとに活性化関数を適用してベクトルにしたものを意味する．2 層だと上の書き方で十分だが，多層に拡張する際には次の表記の方が使いやすい．

$$\begin{aligned}
\boldsymbol{z}^{(0)}_n &= \boldsymbol{x}_n \\
\boldsymbol{a}^{(1)}_n &= \boldsymbol{W}^{(1)} \boldsymbol{z}^{(0)}_n \\
\boldsymbol{z}^{(1)}_n &= \boldsymbol{\phi}.(\boldsymbol{a}^{(1)}_n) \\
\boldsymbol{a}^{(2)}_n &= \boldsymbol{W}^{(2)} \boldsymbol{z}^{(1)}_n \\
\boldsymbol{y}_n &= \boldsymbol{a}^{(2)}_n + \boldsymbol{\epsilon}_n
\end{aligned}$$

■2 層 NN 逆伝播．　パラメータの集合を $\boldsymbol{W}$，学習データ数を N とする．誤差関数を次で設計したとする．

$$E_n(\boldsymbol{W}) = \frac{1}{2} \sum_{d=1}^{D} (y_{n,d} - a^{(2)}_{n,d})^2$$

$$E(\boldsymbol{W}) = \sum_{n=1}^{N} E_n(\boldsymbol{W})$$

この時，ϕ' は活性化関数の微分として，次のように書ける．

$$
\begin{aligned}
\frac{\partial E_n(\boldsymbol{W})}{\partial w^{(2)}_{d,h_1}} &= \frac{\partial E_n(\boldsymbol{W})}{\partial a^{(2)}_{n,d}} \frac{\partial a^{(2)}_{n,d}}{\partial w^{(2)}_{d,h_1}} = \left[y_{n,d} - a^{(2)}_{n,d}\right] \left[z^{(1)}_{n,h_1}\right] = \left[\delta^{(2)}_{n,d}\right] \left[z^{(1)}_{n,h_1}\right] = \delta^{(2)}_{n,d} z^{(1)}_{n,h_1} \\
\frac{\partial E_n(\boldsymbol{W})}{\partial w^{(1)}_{h_1,h_0}} &= \left\{ \sum_{d=1}^{D} \frac{\partial E_n(\boldsymbol{W})}{\partial a^{(2)}_{n,d}} \frac{\partial a^{(2)}_{n,d}}{\partial a^{(1)}_{n,h_1}} \right\} \frac{\partial a^{(1)}_{n,h_1}}{\partial w^{(1)}_{h_1,h_0}} = \left\{ \sum_{d=1}^{D} \delta^{(2)}_{n,d} \frac{\partial a^{(2)}_{n,d}}{\partial a^{(1)}_{n,h_1}} \right\} \frac{\partial a^{(1)}_{n,h_1}}{\partial w^{(1)}_{h_1,h_0}} \\
&= \left\{ \sum_{d=1}^{D} \delta^{(2)}_{n,d} \left[\frac{\partial}{\partial a^{(1)}_{n,h_1}} \sum_{h'_1=1}^{H_1} w^{(2)}_{d,h'_1} z^{(1)}_{n,h'_1} \right] \right\} \left[\frac{\partial}{\partial w^{(1)}_{h_1,h_0}} \sum_{h'_0=1}^{H_0} w^{(1)}_{h_1,h'_0} z^{(0)}_{n,h'_0} \right] \\
&= \left\{ \sum_{d=1}^{D} \delta^{(2)}_{n,d} \left[w^{(2)}_{d,h_1} \frac{\partial z^{(1)}_{n,h_1}}{\partial a^{(1)}_{n,h_1}} \right] \right\} \left[z^{(0)}_{n,h_0} \right] = \left\{ \sum_{d=1}^{D} \delta^{(2)}_{n,d} \left[w^{(2)}_{d,h_1} \phi'(a^{(1)}_{n,h_1}) \right] \right\} \left[z^{(0)}_{n,h_0} \right] \\
&= \left\{ \phi'(a^{(1)}_{n,h_1}) \sum_{d=1}^{D} \delta^{(2)}_{n,d} w^{(2)}_{d,h_1} \right\} \left[z^{(0)}_{n,h_0} \right] \\
&= \left[\delta^{(1)}_{n,h_1} \right] \left[z^{(0)}_{n,h_0} \right] = \delta^{(1)}_{n,h_1} z^{(0)}_{n,h_0}
\end{aligned}
$$

■**更新.**　全てのパラメータに対する微分が上で計算できたので，勾配法で更新することができる．

$$
w^{(\ell)}_{i,j} - \eta \frac{\partial E(\boldsymbol{W})}{\partial w^{(\ell)}_{i,j}} = w^{(\ell)}_{i,j} - \eta \sum_{n=1}^{N} \frac{\partial E_n(\boldsymbol{W})}{\partial w^{(\ell)}_{i,j}}
$$

■**まとめ.**

- 順伝播で全ての $\boldsymbol{a}$ と $\boldsymbol{z}$ を計算.
- 逆伝播で全ての $\boldsymbol{\delta}$ を計算.
- 勾配法.

1.2　多層パーセプトロン

同様にして導出できる．

■**MLP 順伝播.**

$$
\begin{aligned}
\boldsymbol{z}^{(\ell)}_n &= \begin{cases} \boldsymbol{x}_n & \text{if } \ell = 0 \\ \boldsymbol{\phi}.(\boldsymbol{a}^{(\ell)}_n) & \text{if } \ell \in \{1, \cdots, L-1\} \end{cases} \\
\boldsymbol{a}^{(\ell)}_n &= \boldsymbol{W}^{(\ell)} \boldsymbol{z}^{(\ell-1)}_n, \quad \ell = 1, \cdots, L \\
\boldsymbol{y}_n &= \boldsymbol{a}^{(L)}_n + \boldsymbol{\epsilon}_n
\end{aligned}
$$

■**MLP 逆伝播.**

$$
\begin{aligned}
\frac{\partial E_n(\boldsymbol{W})}{\partial w^{(L)}_{d,h_1}} &= \left[\frac{\partial E_n(\boldsymbol{W})}{\partial a^{(L)}_{n,d}} \right] \left[z^{(L-1)}_{n,h_{L-1}} \right] = \delta^{(L)}_{n,d} z^{(L-1)}_{n,h_{L-1}} \\
\frac{\partial E_n(\boldsymbol{W})}{\partial w^{(\ell)}_{h_\ell,h_{\ell-1}}} &= \left[\phi'(a^{(\ell)}_{n,h_\ell}) \sum_{h_{\ell+1}=1}^{H_{\ell+1}} \delta^{(\ell+1)}_{n,h_{\ell+1}} w^{(\ell+1)}_{h_{\ell+1},h_\ell} \right] \left[z^{(\ell-1)}_{n,h_{\ell-1}} \right] = \delta^{(\ell)}_{n,h_\ell} z^{(\ell-1)}_{n,h_{\ell-1}}, \quad \ell = 1, \cdots, L-1
\end{aligned}
$$

2 最適化 (optimization)

Adam [13] の導入を目標にして，書いていく．説明が載っている和書は，[24] が良いと思う．でも，英語て [19] を読むのが一番分かりやすいと思う．

2.1 復習：最急勾配法

変数を $\boldsymbol{w}$，評価関数（滑らかとする）を $E(\boldsymbol{w})$ とする．Taylor 展開により，

$$\begin{aligned} E(\boldsymbol{w}+\eta\boldsymbol{d}) &= E(\boldsymbol{w}) + \eta\boldsymbol{d}^T \left.\frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w})\right|_{\boldsymbol{w}} + o(\eta) \\ &= E(\boldsymbol{w}) + \eta\boldsymbol{d}^T \nabla E(\boldsymbol{w}) + o(\eta) \end{aligned}$$

そのため，微小な係数 η の高次項を無視して，$||\boldsymbol{d}|| = 1$ の下で $E(\boldsymbol{w}+\eta\boldsymbol{d})$ を最小化する $\boldsymbol{d}$ は，次を満たす．

$$\boldsymbol{d} = -\frac{1}{||\nabla E(\boldsymbol{w})||}\nabla E(\boldsymbol{w})$$

最急降下法の基本的アイデアは，上式に基づき次式で $\boldsymbol{w}$ を更新していくことである．

$$\begin{aligned} \boldsymbol{d}^{(t)} &= -\frac{1}{||\nabla E(\boldsymbol{w}^{(t)})||}\nabla E(\boldsymbol{w}^{(t)}) = -\frac{1}{||\left.\frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w})\right|_{\boldsymbol{w}^{(t)}}||} \left.\frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w})\right|_{\boldsymbol{w}^{(t)}} \\ \boldsymbol{w}^{(t+1)} &= \boldsymbol{w}^{(t)} + \eta^{(t)}\boldsymbol{d}^{(t)} \end{aligned}$$

最急降下法に限らず，$\eta^{(t)}$ に相当する部分は学習係数やステップ幅と呼ばれ，様々な決め方が存在する．

2.2 確率的勾配降下 (SGD)

n 番目の入出力データ $(\boldsymbol{x}_n, y)$，与えられたデータを，$D = \{(\boldsymbol{x}_n, y)\}_{n=1}^N$ とする．回帰関数 $f(\boldsymbol{x}; \boldsymbol{w})$，パラメータ $\boldsymbol{w}$ とする．fitting の誤差関数を次のように設計したとする．

$$\begin{aligned} E_n(\boldsymbol{w}) &= \frac{1}{2}(y_n - f(\boldsymbol{x}_2; \boldsymbol{w}))^2 \\ E(\boldsymbol{w}) &= \sum_{n=1}^N E_n(\boldsymbol{w}) \end{aligned}$$

やりたいことは，誤差関数を小さくするような $\boldsymbol{w}$ を見つけることである．しかし，データの個数 N が大きいほど，勾配 $\nabla E(\boldsymbol{w})$ を直接得る事そのものが難しくなる．

確率的勾配降下のアイデアは，「確率的に取り出した $M < N$ となるデータの部分集合（ミニバッチと呼ばれる）に対して誤差関数を設定し，勾配を計算するための 1 回あたりの計算を減らす」というものである．具体的には，$\mathcal{S}$ をランダムに選択された M 個のインデックスとして，次の誤差関数を扱う．

$$E_{\mathcal{S}}(\boldsymbol{w}) = \frac{N}{M}\sum_{n\in M} E_n(\boldsymbol{w})$$

この期待値は $E(\boldsymbol{w})$ と等価である（ように $\mathcal{S}$ がランダムに選ばれる）場合，元々やりたい $E(\boldsymbol{w})$ の最適化をサンプル近似しながら解いているとラフに考えることができる．

学習係数のスケジューリングは，次の条件の下での $\eta^{(t)}$ を用いれば，確率 1 で $E(\boldsymbol{w})$ の停留点に収束する．

$$\sum_{t=1}^{\infty}\eta^{(t)} = \infty, \quad \sum_{t=1}^{\infty}[\eta^{(t)}]^2 < \infty$$

直感的には，$\sum_{t=1}^{\infty} \eta^{(t)} = \infty$ で実行可能領域全体へ到達することを保証し，$\sum_{t=1}^{\infty} [\eta^{(t)}]^2 < \infty$ で推定が収束することを保証している（和書なら [23] を参照）．これを満たすシンプルな方法は，正定数 c に対して $\eta^{(t)} = c/t$ である．

これ以降，バッチサイズが 1 である場合について書く．ステップ t におけるデータのインデックスを n^t とする．この時，誤差関数は $E_{n^t}(\boldsymbol{w})$ である．

2.3 Momentum SGD

局所的に微分係数がゼロに近くなる場合（プラトー，平坦域とも）や，鞍点が存在する場合には，SGD の更新が遅くなってしまう．この問題の解決策の一つが慣性項の利用である．慣性係数 $\mu \in (0, 1]$ として，次のような更新を扱う．

$$
\begin{aligned}
m_i^{(t+1)} &= \mu m_i^{(t)} + \eta \left. \frac{\partial}{\partial w_i} E_{n^t}(\boldsymbol{w}) \right|_{\boldsymbol{w}^{(t)}} \\
w_i^{(t+1)} &= w_i^{(t)} - m_i^{(t+1)}
\end{aligned}
$$

この方法では，微分がゼロに近い場合でも過去の勾配の方向に基づいて更新が進むため，平坦域を抜けることができる．また，この手法は，ランダムサンプリングに起因するパラメータ勾配の振動を平滑化する特性を持っている．

2.4 学習係数の調整

AdaGrad [7] では，パラメータの要素ごとに異なる学習係数を設定する．

$$
\begin{aligned}
g_i &= \left. \frac{\partial}{\partial w_i} E(\boldsymbol{w}) \right|_{\boldsymbol{w}^{(t)}} \\
v_i^{(t+1)} &= v_i^{(t)} + g_i^2 \\
w_i^{(t+1)} &= w_i^{(t)} - \frac{\eta}{\sqrt{v_i^{(t+1)} + \epsilon_c}} g_i
\end{aligned}
$$

ここで，ϵ_c はゼロ除算を防ぐための小さな定数である．

直感的な説明としては，大きな勾配を用いたパラメータ更新が続けば学習係数は早くなり，逆に小さな勾配を用いた更新が続けば現状の学習係数を維持してパラメータを更新する．（和書なら [24, 23] を参照）．

理論は（私には）難しいので省略（COLT で発表→ JMLR で完成版を発表という激強研究です）．

2.5 Adam

天下り的に書くと，Adam アルゴリズムは以下である．

$$
\begin{aligned}
g_i &= \left.\frac{\partial}{\partial w_i} E(\boldsymbol{w})\right|_{\boldsymbol{w}^{(t)}} \\
m_i^{(t+1)} &= \beta_1 m_i^{(t)} + (1-\beta_1) g_i \\
v_i^{(t+1)} &= \beta_2 v_i^{(t)} + (1-\beta_2) g_i^2 \\
\hat{m}_i^{(t+1)} &= \frac{1}{1-\beta_1^{m+1}} m_i^{(t+1)} \\
\hat{v}_i^{(t+1)} &= \frac{1}{1-\beta_2^{m+1}} v_i^{(t+1)} \\
w_i^{(t+1)} &= w_i^{(t)} - \frac{\eta}{\sqrt{\hat{v}_i^{(t+1)} + \epsilon_c}} \hat{m}_i^{(t+1)}
\end{aligned}
$$

Adam は，いくつかの勾配法のアイデアを組み合わせたものになっている．

- 慣性項の利用：$\beta_1 m_i^{(t)} + (1-\beta_1) g_i$ の部分．
- 学習係数のスケジューリング：$\frac{\eta}{\sqrt{\hat{v}_i^{(t+1)} + \epsilon_c}} \hat{m}_i^{(t+1)}$ の部分．
- 主に初期段階において大きくなるバイアスの修正：$\frac{1}{1-\beta^{m+1}}$ の部分．

これ以上の説明は，[19] などを参照すること．

■ノート． 最近，オリジナル版の収束性に関する問題（そもそも非凸で確率的最適化やっているので何であれ難しい）が提起され [18]，理論考察とそれに基づく変種が提案されたりしている（AdaBound [14] など）が，実用面・理論面の両方で adam をちゃんと超えた（と皆が納得できた）ものはまだ無い（という雰囲気がある）．

最近の強化学習実装（ICLR 2020 に通った研究）を眺めていると，Adam（あるいはそれより前の RMSProp など）を使っていれば今のところ文句は言われない（まあ，既存手法との比較をやる都合上，特に議論の対象にしたくない部分は既存手法の実験設定に合わせているだけなんだろうけど）．

3　再パラメータ化

変分ベイズをやる時に，解析的に得られない項がありサンプル近似をする必要がある．雑に言うと，再パラメータ化は，変数変換を用いてサンプル近似の性能を向上する方法である．

3.1　復習：変分ベイズ

入出力データを $(\boldsymbol{X}, \boldsymbol{Y})$，パラメータを $\boldsymbol{w}$ とする．確率モデル $p(\boldsymbol{Y}|\boldsymbol{w}, \boldsymbol{X})$ とする．事前分布 $p(\boldsymbol{w})$ として，Bayes 則に従って事後確率を次のように得る．

$$p(\boldsymbol{w}|\boldsymbol{Y}, \boldsymbol{X}) = \frac{P(\boldsymbol{Y}|\boldsymbol{w}, \boldsymbol{X})P(\boldsymbol{w})}{P(\boldsymbol{Y}|\boldsymbol{X})} = \frac{P(\boldsymbol{Y}, \boldsymbol{w}|\boldsymbol{X})}{P(\boldsymbol{Y}|\boldsymbol{X})}$$

一般に，$p(\boldsymbol{w}|\boldsymbol{Y}, \boldsymbol{X})$ は解析的な形にはならない．変分ベイズでは，$q(\boldsymbol{w}; \boldsymbol{\xi})$ を用いて $p(\boldsymbol{w}|\boldsymbol{Y}, \boldsymbol{X})$ を近似することを考える（$\boldsymbol{\xi}$ は変分パラメータと呼ばれる）．近似の評価指標を KL ダイバージェンスとすると，

$$\begin{aligned}
\mathrm{KL}\left[q(\boldsymbol{w}; \boldsymbol{\xi})||p(\boldsymbol{w}|\boldsymbol{Y}, \boldsymbol{X})\right] &= \int q(\boldsymbol{w}; \boldsymbol{\xi}) \frac{\ln q(\boldsymbol{w}; \boldsymbol{\xi})}{\ln p(\boldsymbol{w}|\boldsymbol{Y}, \boldsymbol{X})} d\boldsymbol{w} = \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln q(\boldsymbol{w}; \boldsymbol{\xi}) - \ln p(\boldsymbol{w}|\boldsymbol{Y}, \boldsymbol{X})\right] d\boldsymbol{w} \\
&= \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln q(\boldsymbol{w}; \boldsymbol{\xi}) - \ln p(\boldsymbol{Y}, \boldsymbol{w}|\boldsymbol{X}) + \ln p(\boldsymbol{Y}|\boldsymbol{X})\right] d\boldsymbol{w} \\
&= \ln p(\boldsymbol{Y}|\boldsymbol{X}) - \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln p(\boldsymbol{Y}, \boldsymbol{w}|\boldsymbol{X}) - \ln q(\boldsymbol{w}; \boldsymbol{\xi})\right] d\boldsymbol{w} \\
&= \ln p(\boldsymbol{Y}|\boldsymbol{X}) - \mathcal{L}(\boldsymbol{\xi})
\end{aligned}$$

ここで，$\mathcal{L}(\boldsymbol{\xi})$ は，変分下界（ELBO）として知られる量である．$\ln p(\boldsymbol{Y}|\boldsymbol{X})$ は $\boldsymbol{\xi}$ に依存しないことに注意すれば，近似指標 $\mathrm{KL}\left[q(\boldsymbol{w}; \boldsymbol{\xi})||p(\boldsymbol{w}|\boldsymbol{Y}, \boldsymbol{X})\right]$ を最小化する問題は，ELBO $\mathcal{L}(\boldsymbol{\xi})$ を最大化する問題と等価である．ELBO は，次のように変形できる．

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\xi}) &= \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln p(\boldsymbol{Y}, \boldsymbol{w}|\boldsymbol{X}) - \ln q(\boldsymbol{w}; \boldsymbol{\xi})\right] d\boldsymbol{w} \\
&= \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln p(\boldsymbol{Y}|\boldsymbol{w}, \boldsymbol{X}) + \ln p(\boldsymbol{w}) - \ln q(\boldsymbol{w}; \boldsymbol{\xi})\right] d\boldsymbol{w} \\
&= \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln p(\boldsymbol{Y}|\boldsymbol{w}, \boldsymbol{X})\right] d\boldsymbol{w} - \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln q(\boldsymbol{w}; \boldsymbol{\xi}) - \ln p(\boldsymbol{w})\right] d\boldsymbol{w} \\
&= \int q(\boldsymbol{w}; \boldsymbol{\xi}) \left[\ln p(\boldsymbol{Y}|\boldsymbol{w}, \boldsymbol{X})\right] d\boldsymbol{w} - \mathrm{KL}\left[q(\boldsymbol{w}; \boldsymbol{\xi})||p(\boldsymbol{w})\right] \\
&= \mathbb{E}_{\boldsymbol{w} \sim q(\boldsymbol{w}; \boldsymbol{\xi})} \left[\ln p(\boldsymbol{Y}|\boldsymbol{w}, \boldsymbol{X})\right] - \mathrm{KL}\left[q(\boldsymbol{w}; \boldsymbol{\xi})||p(\boldsymbol{w})\right]
\end{aligned}$$

基本的には，ELBO の最大化問題を勾配法で解いていくことになるので，上式の各項の $\boldsymbol{\xi}$ に関する微分が欲しいものである．ここで，$\mathrm{KL}\left[q(\boldsymbol{w}; \boldsymbol{\xi})||p(\boldsymbol{w})\right]$ の微分は，事前分布 $p(\boldsymbol{w})$ と変分分布 $q(\boldsymbol{w}; \boldsymbol{\xi})$ を $\boldsymbol{w}$ の正規分布などで与えれば，解析的に計算することができる．一方で，$\mathbb{E}_{\boldsymbol{w} \sim q(\boldsymbol{w}; \boldsymbol{\xi})} \left[\ln p(\boldsymbol{Y}|\boldsymbol{w}, \boldsymbol{X})\right]$ の微分は解析的には計算できない．

これを扱うシンプルな方法は，分布 $q(\boldsymbol{w}; \boldsymbol{\xi})$ から $\boldsymbol{w}$ のサンプルを生成してサンプル近似することであるが，実用上は高い分散を生じてしまうことが知られている．

3.2 reparameterization trick [6]

与えられた $q(\boldsymbol{w};\boldsymbol{\xi})$ に対して，$q(\boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon});\boldsymbol{\xi})d\boldsymbol{\xi} = p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}$ を満たす変数変換 $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon})$ と分布 $p(\boldsymbol{\epsilon})$ を定義できる場合には，次が成り立つ（途中の $(\cdot)_{i,j}$ は行列の成分表示，$(\cdot)_j$ はベクトルの成分表示を指す）．

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\xi}}\mathbb{E}_{\boldsymbol{w}\sim q(\boldsymbol{w};\boldsymbol{\xi})}\left[\ln p(\boldsymbol{Y}|\boldsymbol{w},\boldsymbol{X})\right] &= \frac{\partial}{\partial\boldsymbol{\xi}}\mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}\left[\ln p(\boldsymbol{Y}|\boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon}),\boldsymbol{X})\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}\left[\frac{\partial}{\partial\boldsymbol{\xi}}\ln p(\boldsymbol{Y}|\boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon}),\boldsymbol{X})\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}\left[\left(\frac{\partial\boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon})}{\partial\boldsymbol{\xi}}\right)\left(\frac{\partial}{\partial\boldsymbol{w}}\ln p(\boldsymbol{Y}|\boldsymbol{w},\boldsymbol{X})\Big|_{\boldsymbol{w}=\boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon})}\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}\left[\left(\frac{\partial g_j(\boldsymbol{\xi};\boldsymbol{\epsilon})}{\partial\xi_i}\right)_{i,j}\left(\frac{\partial}{\partial w_j}\ln p(\boldsymbol{Y}|\boldsymbol{w},\boldsymbol{X})\Big|_{\boldsymbol{w}=\boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon})}\right)_j\right]
\end{aligned}$$

よって，1 回のサンプル値 $\tilde{\boldsymbol{\epsilon}} \sim p(\boldsymbol{\epsilon})$ によって近似すると，

$$\frac{\partial}{\partial\boldsymbol{\xi}}\mathbb{E}_{\boldsymbol{w}\sim q(\boldsymbol{w};\boldsymbol{\xi})}\left[\ln p(\boldsymbol{Y}|\boldsymbol{w},\boldsymbol{X})\right] \approx \frac{\partial}{\partial\boldsymbol{\xi}}\ln p(\boldsymbol{Y}|\boldsymbol{g}(\boldsymbol{\xi};\tilde{\boldsymbol{\epsilon}}),\boldsymbol{X}) = \left(\frac{\partial g_j(\boldsymbol{\xi};\tilde{\boldsymbol{\epsilon}})}{\partial\xi_i}\right)_{i,j}\left(\frac{\partial}{\partial w_j}\ln p(\boldsymbol{Y}|\boldsymbol{w},\boldsymbol{X})\Big|_{\boldsymbol{w}=\boldsymbol{g}(\boldsymbol{\xi};\tilde{\boldsymbol{\epsilon}})}\right)_j$$

■具体例：正規分布の場合． $q(w;\boldsymbol{\xi})$ として正規分布が与えられたとする（従って，$\boldsymbol{\xi} = (\mu_\xi, \sigma_\xi)^T$ である）．

$$q(w;\boldsymbol{\xi}) = \mathcal{N}(w|\mu_\xi,\sigma_\xi^2) = \frac{1}{\sqrt{2\pi\sigma_\xi^2}}\exp\left(-\frac{(w-\mu_\xi)^2}{2\sigma_\xi^2}\right)$$

ここで，変数変換 $w = g(\boldsymbol{\xi};\epsilon)$ と分布 $p(\epsilon)$ を次のように定義する．

$$\begin{aligned}
g(\boldsymbol{\xi};\epsilon) = w &= \mu_\xi + \sigma_\xi\epsilon \\
p(\epsilon) = \mathcal{N}(0,1) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\epsilon^2}{2}\right)
\end{aligned}$$

$dw = \frac{dw}{d\epsilon}d\epsilon = \sigma_\xi d\epsilon$ であるため，次が成り立つ．

$$\begin{aligned}
q(w;\boldsymbol{\xi})dw &= \frac{1}{\sqrt{2\pi\sigma_\xi^2}}\exp\left(-\frac{((\mu_\xi+\sigma_\xi\epsilon)-\mu_\xi)^2}{2\sigma_\xi^2}\right)\sigma_\xi d\epsilon \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\epsilon^2}{2}\right)d\epsilon = p(\epsilon)d\epsilon
\end{aligned}$$

よって，この変数変換と分布は，$q(w;\boldsymbol{\xi})dw = p(\epsilon)d\epsilon$ を満たしている．そのため，再パラメータ化が適用できて，

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\xi}}\mathbb{E}_{w\sim q(w;\boldsymbol{\xi})}\left[\ln p(\boldsymbol{Y}|w,\boldsymbol{X})\right] &= \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\left(\frac{\partial g(\boldsymbol{\xi};\epsilon)}{\partial\xi_i}\right)_i\left(\frac{\partial}{\partial w}\ln p(\boldsymbol{Y}|w,\boldsymbol{X})\Big|_{w=g(\boldsymbol{\xi};\epsilon)}\right)\right] \\
&= \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\begin{pmatrix}\frac{\partial g(\boldsymbol{\xi};\epsilon)}{\partial\mu_\xi}\\ \frac{\partial g(\boldsymbol{\xi};\epsilon)}{\partial\sigma_\xi}\end{pmatrix}\left(\frac{\partial}{\partial w}\ln p(\boldsymbol{Y}|w,\boldsymbol{X})\Big|_{w=g(\boldsymbol{\xi};\epsilon)}\right)\right] \\
&= \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\begin{pmatrix}1\\ \epsilon\end{pmatrix}\left(\frac{\partial}{\partial w}\ln p(\boldsymbol{Y}|w,\boldsymbol{X})\Big|_{w=g(\boldsymbol{\xi};\epsilon)}\right)\right]
\end{aligned}$$

■ノート． 再パラメータ化の拡張や，効率化を与えるメカニズムの解析は，現在でも研究トピックの一つである（例えば [22]）．

4 正則化 (regularization)

4.1 weight decay

要するに L2 正則化であり，NN の分野ではこれを weight decay とも呼んでいる．過剰適合を防ぐ目的で，$\boldsymbol{w}$ の大きさに対してペナルティを置いてパラメータを最適化する．L2 正則化では，2 次のペナルティ項を用いる．

$$J(\boldsymbol{w}) = E(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

これは，事前分布を $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\boldsymbol{I})$ とする MAP 推定としても解釈できる（後述）．

■MAP 推定． 次の回帰モデルを考える．

$$y = f(\boldsymbol{x}; \boldsymbol{w}) + \epsilon$$

ここで，$\epsilon \sim \mathcal{N}(\epsilon|0, \sigma^2)$ とする．y の確率分布は，次のように得られる．

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - f(\boldsymbol{x}; \boldsymbol{w}))^2}{2\sigma^2}\right)$$
$$\ln p(y|\boldsymbol{x}, \boldsymbol{w}) = -\frac{(y - f(\boldsymbol{x}; \boldsymbol{w}))^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2)$$

したがって，対数尤度関数は次のように得られる．

$$\ln p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n; \boldsymbol{w}))^2 - \frac{N}{2}\ln(2\pi\sigma^2)$$

また，対数事前分布は次のように得られる．

$$\ln p(\boldsymbol{w}) = -\frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} - \frac{D}{2}\ln(2\pi\lambda^{-1})$$

ここで，D はパラメータ $\boldsymbol{w}$ の次元である．事後確率最大化は，次のように書くことができる．

$$\begin{aligned}
\boldsymbol{w}_{MAP} &= \arg\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) \\
&= \arg\max_{\boldsymbol{w}} \left[\ln p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) + \ln p(\boldsymbol{w})\right] \\
&= \arg\max_{\boldsymbol{w}} \left[-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n; \boldsymbol{w}))^2 - \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}\right] \\
&= \arg\min_{\boldsymbol{w}} \left[\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n; \boldsymbol{w}))^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}\right]
\end{aligned}$$

そのため，誤差関数 $\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n; \boldsymbol{w}))^2$ と，ペナルティ $\frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$ とした最適化問題と等価である．

4.2 dropout

4.2.1 実装

■MLP 順伝播（再掲）.

$$\begin{aligned}
\boldsymbol{z}_n^{(\ell)} &= \begin{cases} \boldsymbol{x}_n & \text{if } \ell = 0 \\ \boldsymbol{\phi}.(\boldsymbol{a}_n^{(\ell)}) & \text{if } \ell \in \{1, \cdots, L-1\} \end{cases} \\
\boldsymbol{a}_n^{(\ell)} &= \boldsymbol{W}^{(\ell)}\boldsymbol{z}_n^{(\ell-1)}, \quad \ell = 1, \cdots, L \\
\boldsymbol{y}_n &= \boldsymbol{a}_n^{(L)} + \boldsymbol{\epsilon}_n
\end{aligned}$$

dropout では，入力 $\boldsymbol{x}_n = \boldsymbol{z}_n^{(0)}$ と中間層 $\boldsymbol{z}_n^{(\ell)}$ に対してランダムにゼロを与えるような操作を行う．中間層の要素 $z_{n,i}^{(\ell)}$ にゼロを与えるかどうかを示すマスクを $m_i^{(\ell)} \in \{0,1\}$ とし，ベルヌーイ分布に従って得る．$m_i^{(\ell)}$ のベクトルを $\boldsymbol{m}^{(\ell)}$ とする．$\boldsymbol{m}^{(\ell)}$ の成分が対角項に並んだ対角行列を，$[\mathrm{diag}\{\boldsymbol{m}^{(\ell)}\}]$ とする．この時，dropout を用いた NN 順伝播は次のように書ける．

■MLP 順伝播 with dropout.

$$
\begin{aligned}
\boldsymbol{z}_n^{(\ell)} &= \begin{cases} \boldsymbol{x}_n & \text{if } \ell = 0 \\ \boldsymbol{\phi}.(\boldsymbol{a}_n^{(\ell)}) & \text{if } \ell \in \{1, \cdots, L-1\} \end{cases} \\
\tilde{\boldsymbol{z}}_n^{(\ell)} &= [\mathrm{diag}\{\boldsymbol{m}^{(\ell)}\}]\boldsymbol{z}_n^{(\ell)}, \quad \ell = 1, \cdots, L \\
\boldsymbol{a}_n^{(\ell)} &= \boldsymbol{W}^{(\ell)}\tilde{\boldsymbol{z}}_n^{(\ell-1)}, \quad \ell = 1, \cdots, L \\
\boldsymbol{y}_n &= \boldsymbol{a}_n^{(L)} + \boldsymbol{\epsilon}_n
\end{aligned}
$$

逆伝播も，$\boldsymbol{z}_n^{\ell}$ を $\tilde{\boldsymbol{z}}_n^{\ell}$ で置き換えて導出する．2 層 NN の場合は，1.1 節の導出の途中から書いていくと，次のようになる．

$$
\begin{aligned}
\frac{\partial E_n(\boldsymbol{W})}{\partial w_{h_1,h_0}^{(1)}} &= \left\{ \sum_{d=1}^{D} \delta_{n,d}^{(2)} \left[w_{d,h_1}^{(2)} \frac{\partial \tilde{z}_{n,h_1}^{(1)}}{\partial a_{n,h_1}^{(1)}} \right] \right\} \left[\tilde{z}_{n,h_0}^{(0)} \right] = \left\{ \sum_{d=1}^{D} \delta_{n,d}^{(2)} \left[w_{d,h_1}^{(2)} m_{h_1}^{(1)} \frac{\partial z_{n,h_1}^{(1)}}{\partial a_{n,h_1}^{(1)}} \right] \right\} \left[m_{h_0}^{(0)} z_{n,h_0}^{(0)} \right] \\
&= \left\{ \sum_{d=1}^{D} \delta_{n,d}^{(2)} \left[w_{d,h_1}^{(2)} m_{h_1}^{(1)} \phi'(a_{n,h_\ell}^{(1)}) \right] \right\} \left[m_{h_0}^{(0)} z_{n,h_0}^{(0)} \right] = \left\{ m_{h_1}^{(1)} \phi'(a_{n,h_\ell}^{(1)}) \sum_{d=1}^{D} \delta_{n,d}^{(2)} \left[w_{d,h_1}^{(2)} \right] \right\} \left[m_{h_0}^{(0)} z_{n,h_0}^{(0)} \right]
\end{aligned}
$$

同様にして，逆伝播は次のように書ける．

■MLP 逆伝播（再掲）.

$$
\begin{aligned}
\frac{\partial E_n(\boldsymbol{W})}{\partial w_{d,h_1}^{(L)}} &= \left[\frac{\partial E_n(\boldsymbol{W})}{\partial a_{n,d}^{(L)}} \right] \left[z_{n,h_{L-1}}^{(L-1)} \right] = \delta_{n,d}^{(L)} z_{n,h_{L-1}}^{(L-1)} \\
\frac{\partial E_n(\boldsymbol{W})}{\partial w_{h_\ell,h_{\ell-1}}^{(\ell)}} &= \left[\phi'(a_{n,h_\ell}^{(\ell)}) \sum_{h_{\ell+1}=1}^{H_{\ell+1}} \delta_{n,h_{\ell+1}}^{(\ell+1)} w_{h_{\ell+1},h_\ell}^{(\ell+1)} \right] \left[z_{n,h_{\ell-1}}^{(\ell-1)} \right] = \delta_{n,h_\ell}^{(\ell)} z_{n,h_{\ell-1}}^{(\ell-1)}, \quad \ell = 1, \cdots, L-1
\end{aligned}
$$

■MLP 逆伝播 with dropout.

$$
\begin{aligned}
\frac{\partial E_n(\boldsymbol{W})}{\partial w_{d,h_1}^{(L)}} &= \left[\frac{\partial E_n(\boldsymbol{W})}{\partial a_{n,d}^{(L)}} \right] \left[m_{h_{L-1}}^{(L-1)} z_{n,h_{L-1}}^{(L-1)} \right] = m_{h_{L-1}}^{(L-1)} \times \delta_{n,d}^{(L)} z_{n,h_{L-1}}^{(L-1)} \\
\frac{\partial E_n(\boldsymbol{W})}{\partial w_{h_\ell,h_{\ell-1}}^{(\ell)}} &= \left[m_{h_\ell}^{(\ell)} \phi'(a_{n,h_\ell}^{(\ell)}) \sum_{h_{\ell+1}=1}^{H_{\ell+1}} \delta_{n,h_{\ell+1}}^{(\ell+1)} w_{h_{\ell+1},h_\ell}^{(\ell+1)} \right] \left[m_{h_{\ell-1}}^{(\ell-1)} z_{n,h_{\ell-1}}^{(\ell-1)} \right] = m_{h_\ell}^{(\ell)} m_{h_{\ell-1}}^{(\ell-1)} \times \delta_{n,h_\ell}^{(\ell)} z_{n,h_{\ell-1}}^{(\ell-1)}, \quad \ell = 1, \cdots, L-1
\end{aligned}
$$

■勾配 with dropout (and 正則化).　dropout 付き順伝播における $\boldsymbol{a}_n^{(\ell)}$ を求める式は，次のようにも書ける．

$$
\boldsymbol{a}_n^{(\ell)} = \boldsymbol{W}^{(\ell)}[\mathrm{diag}\{\boldsymbol{m}^{(\ell)}\}]\boldsymbol{z}_n^{(\ell-1)} = \tilde{\boldsymbol{W}}^{(\ell)}\boldsymbol{z}_n^{(\ell-1)}, \quad \ell = 1, \cdots, L
$$

$\tilde{\boldsymbol{W}}^{(\ell)}$ の集合を $\tilde{\boldsymbol{W}}$ とする．$\boldsymbol{W}$ と $\boldsymbol{m}$ から $\tilde{\boldsymbol{W}}$ を得る操作を，$\tilde{\boldsymbol{W}} = \boldsymbol{g}(\boldsymbol{W}, \boldsymbol{m})$ と表す（具体的には，各層 ℓ で $\tilde{\boldsymbol{W}}^{(\ell)} = \boldsymbol{W}^{(\ell)}[\mathrm{diag}\{\boldsymbol{m}^{(\ell)}\}]$ をする）．$\boldsymbol{a}_n^{(L)} = \boldsymbol{f}(\boldsymbol{x}_n; \tilde{\boldsymbol{W}}) = \boldsymbol{f}(\boldsymbol{x}_n; \boldsymbol{g}(\boldsymbol{W}, \boldsymbol{m}))$ と表記すると，回帰モデルは次のように書ける．

$$
\boldsymbol{y}_n = \boldsymbol{f}(\boldsymbol{x}_n; \boldsymbol{g}(\boldsymbol{W}, \boldsymbol{m})) + \boldsymbol{\epsilon}_n
$$

ノイズ項の分布を正規分布 $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\boldsymbol{\epsilon}_n | \boldsymbol{0}, \sigma^2 \boldsymbol{I})$ で仮定すると，対数尤度関数は

$$
\ln p(\boldsymbol{y}_n | \boldsymbol{f}(\boldsymbol{x}_n; \boldsymbol{g}(\boldsymbol{W}, \boldsymbol{m}))) = -\frac{1}{2\sigma^2}(\boldsymbol{y}_n - \boldsymbol{f}(\boldsymbol{x}_n; \boldsymbol{g}(\boldsymbol{W}, \boldsymbol{m})))^T(\boldsymbol{y}_n - \boldsymbol{f}(\boldsymbol{x}_n; \boldsymbol{g}(\boldsymbol{W}, \boldsymbol{m}))) - \frac{1}{2}\ln(2\pi\sigma^2)
$$

ミニバッチのデータ数を M，インデックス集合を $\mathcal{S}$ とする．評価関数（最小化したい）を，二乗誤差プラス正則化項とする．

$$J(\boldsymbol{W}) = \frac{1}{2M}\sum_{n\in\mathcal{S}}(\boldsymbol{y}_n - \boldsymbol{f}(\boldsymbol{x}_n;\boldsymbol{g}(\boldsymbol{W},\boldsymbol{m})))^T(\boldsymbol{y}_n - \boldsymbol{f}(\boldsymbol{x}_n;\boldsymbol{g}(\boldsymbol{W},\boldsymbol{m}))) + \frac{1}{2}\sum_{\ell=1}^{L}\lambda_\ell||\boldsymbol{W}^{(\ell)}||$$
$$= -\left\{\sigma^2\sum_{n\in\mathcal{S}}\ln p(\boldsymbol{y}_n|\boldsymbol{f}(\boldsymbol{x}_n;\boldsymbol{g}(\boldsymbol{W},\boldsymbol{m}))) - \frac{1}{2}\sum_{\ell=1}^{L}\lambda_\ell||\boldsymbol{W}^{(\ell)}||\right\} + \text{const.}$$

勾配を取ると，

$$\nabla_{\boldsymbol{W}}J(\boldsymbol{W}) = -\left\{\sigma^2\sum_{n\in\mathcal{S}}\nabla_{\boldsymbol{W}}\ln p(\boldsymbol{y}_n|\boldsymbol{f}(\boldsymbol{x}_n;\boldsymbol{g}(\boldsymbol{W},\boldsymbol{m}))) - \frac{1}{2}\sum_{\ell=1}^{L}\lambda_\ell\nabla_{\boldsymbol{W}}||\boldsymbol{W}^{(\ell)}||\right\}$$

なお，マスク $\boldsymbol{m}$ は確率的に得られていることに注意する．

4.2.2 変分ベイズとしての解釈 [8]

■勾配 with dropout の記号の書き換え． 前小節で導出した勾配の記号を，$\boldsymbol{W} \to \boldsymbol{\xi}$ と $\boldsymbol{m} \to \tilde{\boldsymbol{\epsilon}}$ とと書き替える．

$$\nabla_{\boldsymbol{\xi}}J(\boldsymbol{\xi}) = -\left\{\sigma^2\sum_{n\in\mathcal{S}}\nabla_{\boldsymbol{\xi}}\ln p(\boldsymbol{y}_n|\boldsymbol{f}(\boldsymbol{x}_n;\boldsymbol{g}(\boldsymbol{\xi},\tilde{\boldsymbol{\epsilon}}))) - \frac{1}{2}\sum_{\ell=1}^{L}\lambda_\ell\nabla_{\boldsymbol{\xi}}||\boldsymbol{\xi}^{(\ell)}||\right\} \tag{1}$$

■変分ベイズ with 再パラメータ化勾配（3 節の導出をまとめると得られる）．

$$\nabla_{\boldsymbol{\xi}}\mathcal{L}(\boldsymbol{\xi}) = \nabla_{\boldsymbol{\xi}}\mathbb{E}_{\boldsymbol{w}\sim q(\boldsymbol{w};\boldsymbol{\xi})}\left[\ln p(\boldsymbol{Y}|\boldsymbol{w},\boldsymbol{X})\right] - \nabla_{\boldsymbol{\xi}}\mathrm{KL}\left[q(\boldsymbol{w};\boldsymbol{\xi})||p(\boldsymbol{w})\right]$$
$$\approx \nabla_{\boldsymbol{\xi}}\ln p(\boldsymbol{Y}|\boldsymbol{g}(\boldsymbol{\xi};\tilde{\boldsymbol{\epsilon}}),\boldsymbol{X}) - \nabla_{\boldsymbol{\xi}}\mathrm{KL}\left[q(\boldsymbol{w};\boldsymbol{\xi})||p(\boldsymbol{w})\right]$$

ここで，$\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{\xi};\boldsymbol{\epsilon})$ と $p(\boldsymbol{\epsilon})$ は再パラメータ化の条件を満たしていて，また $\tilde{\boldsymbol{\epsilon}}$ は $p(\boldsymbol{\epsilon})$ から得られたサンプルである．尤度関数を $p(\boldsymbol{Y}|\boldsymbol{w},\boldsymbol{X}) = \prod_{n\in\mathcal{S}}p(\boldsymbol{y}_n|\boldsymbol{f}(\boldsymbol{x}_n;\boldsymbol{w}))$ と書くと，

$$\nabla_{\boldsymbol{\xi}}\mathcal{L}(\boldsymbol{\xi}) \approx \sum_{n\in\mathcal{S}}\nabla_{\boldsymbol{\xi}}\ln p(\boldsymbol{y}|\boldsymbol{f}(\boldsymbol{x}_n;\boldsymbol{g}(\boldsymbol{\xi};\tilde{\boldsymbol{\epsilon}}))) - \nabla_{\boldsymbol{\xi}}\mathrm{KL}\left[q(\boldsymbol{w};\boldsymbol{\xi})||p(\boldsymbol{w})\right] \tag{2}$$

■比較． 定数倍を無視して式 (1) と (2) を見比べると，第 1 項についてはそれぞれ同じ形で表現できる．そのため，式 (1) の第 2 項（正則化項の勾配←データに依存しない）と，式 (2) の第 2 項（事前分布・変分分布間の KL ダイバージェンス←データに依存しない）を同じ形で表現できるなら，dropout は変分ベイズの枠組みの枠組みで解釈できる．これは，特定の $q(\boldsymbol{w};\boldsymbol{\xi})$ と $p(\boldsymbol{w})$ を選ぶことにより成立させることができる（条件は [8] とその関連研究を参照）．

これらをまとめると，

- dropout におけるパラメータ $\boldsymbol{W}$ は，変分ベイズにおけるハイパパラメータ $\boldsymbol{\xi}$ に相当する．
- dropout におけるマスク $\boldsymbol{m}$ のサンプリングは，再パラメータ化における $\boldsymbol{\epsilon}$ のサンプリングに相当する．
- dropout の背後には，暗に確率変数（式 (2) における $\boldsymbol{w}$ 相当）が存在しているが，周辺化されているので表には出てこない．

一般に，確率変数の周辺化は，確率変数の点推定と比較して overfitting に対してロバストである（その代償に計算量をより多く必要とする）と理解される（例：NN における重みパラメータの点推定よりも，Bayesian NN における重みパラメータの周辺化の方がロバストである）．上述の議論より，dropout はある種の周辺化を暗にやっていると解釈することができて，それが dropout において経験的に得られているロバスト性の理由だと考えられている．

5 正規化 (normalization)

5.1 batch normalization [12]

多層 NN のパラメータを更新する場合，前の層のパラメータの変化によって次の層に対する入力の分布が大きく変化してしまい，学習が不安定になりやすい．正規化のアイデアは，入力を標準正規（ガウス）分布を使って変換して用いることにより，学習を安定化させようというものである．

n 番目のデータの ℓ 層目の中間変数を，$\boldsymbol{z}_n^{(\ell)} = (z_{n,1}^{(\ell)}, z_{n,2}^{(\ell)}, \cdots)^T$ と表記する．ミニバッチに含まれるデータのインデックスの集合を $\mathcal{S}$ とする．具体的には，次の変換を用いる．

$$
\begin{aligned}
\mu_d^{(\ell)} &= \frac{1}{m}\sum_{n\in\mathcal{S}} z_{n,d}^{(\ell)} \\
{\sigma^2}_d^{(\ell)} &= \frac{1}{m}\sum_{n\in\mathcal{S}} (z_{n,d}^{(\ell)} - \mu_d^{(\ell)})^2 \\
\hat{z}_{n,d}^{(\ell)} &= \frac{z_{n,d}^{(\ell)} - \mu_d}{\sqrt{{\sigma^2}_d^{(\ell)} + \epsilon_c}} \\
\tilde{z}_{n,d}^{(\ell)} &= \gamma_d^{(\ell)} \hat{z}_{n,d}^{(\ell)} + \beta_d^{(\ell)}
\end{aligned}
$$

ここで，γ_d と β_d は学習可能な係数である．

■ノート． バッチ正則化が収束と汎化を改善する事，つまり学習安定化／効率化と overfitting 低減化をしている事は，多くの研究において経験的には得られている（だからみんな使っている）．また，理解の試みもなされている [15]．強化学習でも，汎化性能を改善するみたいな話（要検討）がある [5]．

5.2 layer normalization [2]

別の空間で正規化する．[3]（ICLR 2020 で興味深いと言われつつ検証不足で reject されたもの）を見てると，強化学習において効率的になるかは微妙な感じ？

5.3 weight normalization [20, 15]

別の空間で正規化する．いくつかの強化学習実装（例えば [4, 11]）でも見かける．

6 雑感と文献メモ

- 各項目について，[25] が和書でだいたい網羅してくれているので，まずはこれの該当箇所を読めばいいと思う．
- 活性化関数は，ここ最近の強化学習実装（例えば SAC [9]）ではほぼ ReLU 一択なのが現状．ちょっと古い強化学習実装（例えば TRPO [21] とか）だと，tanh とかも見かける（ReLU vs tanh の比較は [10] を見ると良い）．強化学習に限らず，swish [17] とか mish [16] とか提案されてはいるものの，実用面は ReLU 一択で決着がついた印象．
- アーキテクチャに関するのは，(1) mujoco などで状態が関節角なら MLP，(2) 画像を状態とするなら CNN，(3) 完全状態観測ではなく履歴が欲しいなら（あるいは動的な特徴量が欲しいなら）RNN や LSTM，を使っている印象．最新の強化学習でもまだ最近の深層学習アーキテクチャを使いこなしていない（あるいは手が回っていない）雰囲気なので，MLP さえ押さえておけば我々にとっては十分だと思う．

参考文献

[1] Joshua Achiam. Spinning Up in Deep Reinforcement Learning. *GitHub repository*, 2018. `https://github.com/openai/spinningup`.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Aditya Bhatt, Max Argus, Artemij Amiranashvili, and Thomas Brox. Crossnorm: Normalization for off-policy td reinforcement learning. *arXiv preprint arXiv:1902.05605*, 2019.

[4] Ignasi Clavera, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning*, pages 617–629, 2018.

[5] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289, 2019.

[6] P Kingma Diederik and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

[10] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018.

[11] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *The IEEE International Conference on Computer Vision*, 2019.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by

reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[14] Liangchen Luo, Yuanhao Xiong, and Yan Liu. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2019.

[15] Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. In *International Conference on Learning Representations*, 2019.

[16] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[17] Quoc V. Le Prajit Ramachandran, Barret Zoph. Searching for activation functions, 2018.

[18] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[19] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[20] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pages 901–909, 2016.

[21] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[22] Ming Xu, Matias Quiroz, Robert Kohn, and Scott A Sisson. Variance reduction properties of the reparameterization trick. In *International Conference on Artificial Intelligence and Statistics*, pages 2711–2720, 2019.

[23] 鈴木 大慈. **確率的最適化** = *Stochastic optimization*. MLP 機械学習プロフェッショナルシリーズ. 講談社, 2015.

[24] 原田 達也. **画像認識** = *Image recognition*. MLP 機械学習プロフェッショナルシリーズ. 講談社, 2017.

[25] 須山 敦志. **ベイズ深層学習** = *Bayesian deep learning*. MLP 機械学習プロフェッショナルシリーズ. 講談社, 2019.