

バッチ正規化（Batch Normalization）解説資料

1. バッチ正規化とは

バッチ正規化（BN）は、ニューラルネットワークの中間層において、**入力の分布を標準化（平均0、分散1に近づける）** する手法である。
これにより、学習を安定化させ、収束を早め、過学習を抑制する効果がある。

2. 処理の流れ

バッチ正規化は、ミニバッチ単位で以下の計算を行う。

1. バッチ平均の計算

$$\begin{aligned} &[\\ &\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \\ &] \end{aligned}$$

- m はミニバッチ内のサンプル数
- (x_i) は i 番目の入力値

2. バッチ分散の計算

$$\begin{aligned} &[\\ &\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \\ &] \end{aligned}$$

3. 正規化

3. 学習と推論時の違い

- 学習時

- 各ミニバッチの平均・分散を用いる
- 同時に「移動平均 (running mean/var)」を更新する

- 推論時

- 学習時に記録した「移動平均・分散」を使って正規化する

4. 効果

1. 内部共変量シフト（Internal Covariate Shift）の低減

- 各層への入力分布の変化を抑えることで学習を安定化

2. 勾配消失・爆発の抑制

- 活性化関数の入力が極端な値になるのを防ぐ

3. 正則化効果

- ミニバッチごとの統計量を利用するため、軽いノイズ効果が生まれ過学習抑制につながる

5. PyTorchでの実装例

```
import torch.nn as nn

# チャンネル数32の2次元バッチ正規化
bn = nn.BatchNorm2d(num_features=32)

# モデルに組み込み
self.conv1 = nn.Conv2d(3, 32, kernel_size=3, padding=1)
self.bn1 = nn.BatchNorm2d(32)
```

6. まとめ

- バッチ正規化は「平均0・分散1」に正規化しつつ、スケールとシフトを学習する層
- 学習を安定させ、収束を早め、精度改善や過学習抑制の効果がある
- CNNやRNNを含む多くのモデルで活用されている