

## OFDMにおけるDNNを用いたデータ検出法の簡易化の検討

A-9 Study of Data Detection Method Using Simplified DNN in OFDM

井坂 開  
Kai Isaka宮嶋 照行  
Teruyuki Miyajima杉谷 栄規  
Yoshiki Sugitani茨城大学大学院理工学研究科  
Graduate School of Science and Engineering, Ibaraki University

## 1 まえがき

近年、無線通信分野へのディープラーニングの応用が盛んに研究されており [1], その一つに OFDM 信号検出への応用がある. Ye 等 [2] は受信機で周波数領域信号から送信データを検出するためにディープニューラルネットワーク (DNN) を用いている. この DNN は複数の通信路状態情報を事前に学習させておくことで, オンラインで通信路が未知の場合でも推定せずにデータの復調を可能とする. しかし, パイロットとデータシンボルから成る 2 ブロックを入力しており, 規模の大きい DNN を要するため, 必要な計算量が多くなる. そこで, 本稿では 1 ブロックのみを入力する DNN について検討し, 従来の DNN よりも構造が簡易で, 同等の BER 性能が得られることを示す.

## 2 提案法

## 2.1 システムモデル

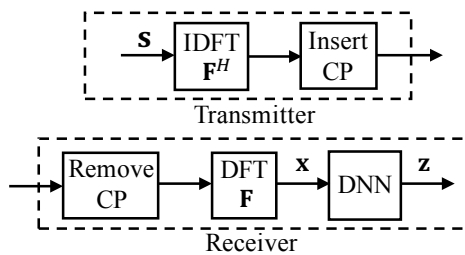


図1 OFDM システム.

$N$  個のサブキャリアをもつ図1の OFDM システムを考える. 送信機では  $P$  個のパイロットと  $(N - P)$  個のデータシンボルから成る周波数領域ブロック  $\mathbf{s} \in \mathbb{C}^{N \times 1}$  を送信する. このとき, パイロットは  $N/P$  個間隔で挿入する.  $\mathbf{s}$  は IDFT 処理後, CP を付加して送信する. 次数  $L_h$  の通信路を通過して得られた受信信号から CP を除去し, DFT 処理後の周波数領域信号は  $\mathbf{y} = \mathbf{F}\mathbf{H}_c\mathbf{F}^H\mathbf{s} + \mathbf{F}\mathbf{v} \in \mathbb{C}^{N \times 1}$  となる. ここで, DFT 行列を  $\mathbf{F} \in \mathbb{C}^{N \times N}$ , 通信路テプリッツ行列の巡回行列を  $\mathbf{H}_c$ , 雑音を  $\mathbf{v}$  とする.

受信機では,  $\mathbf{y}$  の実部と虚部を学習済み DNN への入力  $\mathbf{x} \in \mathbb{R}^{2N \times 1}$  とする. このとき, 学習済み DNN は複数用意し, 各 DNN は  $\mathbf{s}$  の一部のシンボルを 16 ビット分だけ復調するように学習させておく. 最終的に, 各 DNN からの出力  $\mathbf{z} \in \mathbb{R}^{16 \times 1}$  を継続して判定することで送信データを得る.

## 2.2 学習サンプルの作成手順と DNN の学習

学習サンプルの作成方法を説明する. 初めに  $\hat{\mathbf{H}}_c$  を  $M$  種類発生させ, 1 種類をランダムに選ぶ. 次に, その通信路とランダムに発生させた  $\hat{\mathbf{s}}$  と  $\hat{\mathbf{v}}$  から  $\hat{\mathbf{y}}$  を作成し,  $\hat{\mathbf{x}}$  を作成する. また,  $\hat{\mathbf{s}}$  から目標出力  $\hat{\mathbf{b}} \in \mathbb{R}^{16 \times 1}$  を作成し,  $(\hat{\mathbf{x}}, \hat{\mathbf{b}})$  を学習サンプルとする ( $\hat{\cdot}$  はオフラインで扱う信号を表す). 同様にして  $K$  個の学習サンプルを作成する. このとき, 全てのサンプルで挿入したパイロットは変えておらず, オンライン時においても同一のパイロットを挿入する.

DNN の学習は  $K$  個の学習サンプルを用いて行う. 学習サンプル  $(\hat{\mathbf{x}}_k, \hat{\mathbf{b}}_k), k = 0, \dots, K - 1$  において,  $\hat{\mathbf{x}}_k$  を入力して得られる  $\hat{\mathbf{z}}_k$  と  $\hat{\mathbf{b}}_k$  の平均自乗誤差を評価関数

$$J(\theta) = \frac{1}{K} \sum_k \|\hat{\mathbf{z}}_k - \hat{\mathbf{b}}_k\|^2, \quad (1)$$

とする. DNN の重み  $\theta$  は  $J$  が局所的最小となるように勾配法により更新していく. 学習済み DNN はオンライン時において, 未知の通信路を通過して得られた  $\mathbf{x}$  に対して,  $\mathbf{z}$  から正しくデータを復調できるようになる.

## 3 シミュレーション

変調方式を QPSK,  $N = 64, P = 8, L_h = 5, M = 10000, K = 4 \times 10^5$ , SNR を 40 [dB] とした. また, DNN 構造 (入力~出力層) は提案法を 128,300,200,100,16, 従来法を 256,500,250,120,16, 中間層活性化関数を ReLU, 学習係数を  $1 \times 10^{-5}$ , 勾配法を Adam, エポック数を 50 とした. 通信路は平均 0, 分散 1 でランダムに発生させた. このとき, 未知の通信路における BER 性能の比較を図 2 に示す. 縦軸は BER, 横軸はエポック数である. 図 2 より, 提案法においてエポック数が増えると, 未知の通信路における BER が従来法と同程度に減少していくことが分かる. また, 提案法は DNN の規模が小さいため, 16 ビットを得るために必要な計算量も減らすことができる.

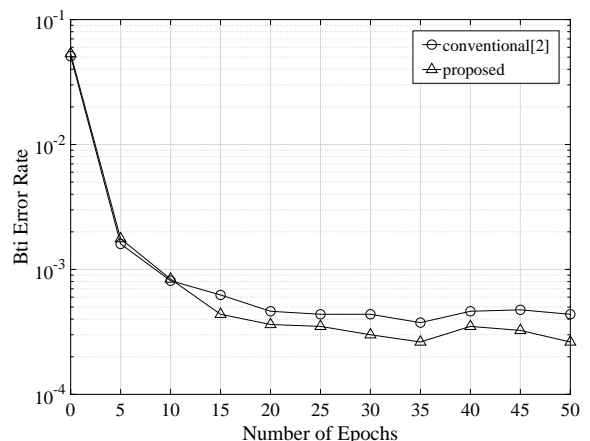


図2 未知の通信路における BER 性能の比較.

## 4 まとめ

従来法よりも構造が簡易な DNN を用いて, 同等の BER 性能が得られることを確認した.

## 参考文献

- [1] Q.Mao, et al., "Deep learning for...", *IEEE Commun. Surveys Tuts.*, vol.20, no.4, pp.2595–2621, 4<sup>th</sup> Quart., 2018.
- [2] H.Ye, et al., "Power of Deep Learning for...", *IEEE Wireless Commun. Lett.*, vol.7, no.1, pp.114–117, Feb. 2018.