基于深度学习的通道解码[1]

刘畅

摘要

我们重新讨论了使用深度神经网络对随机和结构化代码 (如极化码) 进行一次性解码的想法。尽管对于代码族和短码字长度都有可能实现最大的后验 (MAP) 误码率 (BER) 性能，但我们观察到 (i) 结构化代码更容易学习，(ii) 神经网络能够泛化到它在结构化代码训练中从未见过的码字，而不是随机代码。这些结果提供了一些证据，证明神经网络可以学习解码算法，而不仅仅是简单的分类

关键词：深度学习；解码

1 引言

基于深度学习的通道解码的复杂度很高，比如对于一个长度为 100bit，编码率为 0.5 的数据，一共有 2^{50} 种解码的可能，这对于神经网络来说是个很大的样本空间。神经网络只有当它能从部分的编码中学习到编码的规律，并且能实现全部样本的解码，才能真正用于实际解码。但是，为了让神经网络能学习到解码的算法，这种编码算法需要有一定的结构，也就是基于简单的编码规则，比如卷积码或者代数编码。本篇论文探讨两个问题，一是相对于非结构化的编码，结构化的编码是否更容易学习？而是一个神经网络是否能在经过训练后，能正确解码它所没见过的样本。

需要强调的是本篇论文是基于较短的块长来实现的，比如 N<64, 这样能跟最大后验解码进行一个对比。虽然块长比较短，但是依然具有一定的实际意义，比如在一些物联网场景中，块长并不需要很长。本篇论文限制了块长来降低指数提升的训练复杂度。因此，采用神经网络解码目前并不能跟已经发展了十多年并且可以扩展到任意字长的最先进的解码算法相比。

但是，存在着一些编码结构能加快学习进程，本文发现结构化的编码相比于随机化的编码更容易进行学习，比如只需要更少的训练次数。另外，我们发现不管是对于结构化的编码还是随机化的编码，神经网络在只接受部分样本的情况下，都能泛化出对所有样本的解码能力。

1.1 深度学习基础

深度学习的原理是很复杂的。神经网络包含许多互相连接的神经元，每个神经元都拥有具有权重的输入，将其叠加并且加上一个偏移，作为一个结果并且通过一个非线性的激活函数进行传播，比如 sigmoid 函数或者 ReLU 函数，它们如 1. 定义。

$$g_{sigmoid}(z) = \frac{1}{1+e^{-z}} \qquad g_{ReLU}(z) = max\{0, z\} \tag{1}$$

我们将输入层通过隐藏层到输出层的计算，叫做正向传播，反过来的话则称为反向传播。在实际使用中，我们会有一个损失函数，用这个函数来表示神经网络的输出值与目标值之间的差距。神经网络迭代的过程，就是不断利用正向传播进行计算以及反向传播调整权重，从而使得损失函数不断减小的过程。利用梯度下降优化算法以及反向传播，神经网络的权重能在所有的训练集中找到合适的权重，使得损失函数最小。但是学习的目的是为了应对没见过的数据，因此为了量化神经网络的学习结果，我们使用测试集来作为神经网络的输入，并判断该神经网络是否具有一定的泛化能力。在学习的过程中，

我们可以通过合理的学习率来控制神经网络的学习速度，当学习率太大时，网络很难收敛；当学习率过低时，训练则要耗费较多的时间。

1.2 通道编码基础

通道编码是对信道进行编码以实现无错误的传输。通道编码对原有数据进行调制，也就是前向错误检测。在接收端，这些通道编码的 bit 将用于验证是否有错误，通过通道编码，能提高信号的信噪比。下面主要介绍极化码，这是一种常用的编码。

2008 年，Erdal Arikan 在国际信息论 ISIT 会议上首次提出了信道极化（Channel Polarization）的概念；2009 年在“IEEE Transaction on Information Theory”期刊上发表了一篇长达 23 页的论文更加详细地阐述了信道极化，并基于信道极化给出了一种新的编码方式，名称为极化码（Polar Code）。极化码具有确定性的构造方法，并且是已知的唯一一种能够被严格证明“达到”信道容量的信道编码方法。

从代数编码和概率编码的角度来说，极化码具备了两者各自的特点。首先，只要给定编码长度，极化码的编译码结构就唯一确定了，而且可以通过生成矩阵的形式完成编码过程，这一点和代数编码的常见思维是一致的。其次，极化码在设计时并没有考虑最小距离特性，而是利用了信道联合（Channel Combination）与信道分裂（Channel Splitting）的过程来选择具体的编码方案，而且在译码时也是采用概率算法，这一点比较符合概率编码的思想。

2 相关工作

在 1943 年，为了解决问题，McCulloch 跟 Pitts 对人结构脑进行建模[2]. 但是直到 45 年后向后传播算法[3]的出现才能让一些应用实现，比如手写 ZIP 代码识别。早期神经网络的形式是 HopField 网络。但是由于它的低存储容量，Hopfield 网络很快被能在噪声以及密文输入间学习的前向反馈神经网络取代。因为神经网络在学习过程中能学习匹配或者能提取通道数据，因此没有对通道的噪声做任何的假设[4]。关于使用神经网络进行解码的不同的思想出现在九十年代。但是神经网络解码一直没有取得大的突破，因为标准的训练过程不能让神经网络进行长密文的解码，因此，学界对神经网络解码兴趣下降了。近些年来，通过使用随机神经网络[5]或者减少权重的数量[6]，神经网络解码获得了一些轻微的提升。

在 2006 年，一种叫梯度下降微调的层层无监督的预训练[7]的新的训练技术，使得神经网络复兴，因为这种技术使得神经网络能训练更多层。带有许多隐藏层的网络叫做深度神经网络。如今，强大的硬件比如图形处理单元（GPU）能加速学习过程。在神经网络复兴的过程中，一些关于用神经网络解码的想法出现了。但是，相比于之前的工作，神经网络解码只是用来优化已经知道的解码原理。比如，在这篇论文[8]中，为了改进 BP 算法，对信念传播算法的 Tanner 图进行了权值分配，并通过神经网络技术进行学习，机器学习领域内的人似乎还没有适应学习解码的思想。

3 本文方法

对于结构化编码，本文采用的是 PolarCode，另一种则是采用 Random Code。主要过程的框架如图 1所示，首先先对 k bit 数据编码为 N bit 的数据，再经过调制，加入噪声，根据需要进行 LLR(对数似然比)，然后最后得到的数据作为神经网络的输入。本文采用的神经网络的结构为隐藏层为 3 层的全连接神经网络，对于输入层以及隐藏层，采用 Sigmoid 作为其激活函数，对于输出层，采用 Relu 作

为其输出函数，输入层的神经元个数为 N，输出层的神经元的个数为 k。

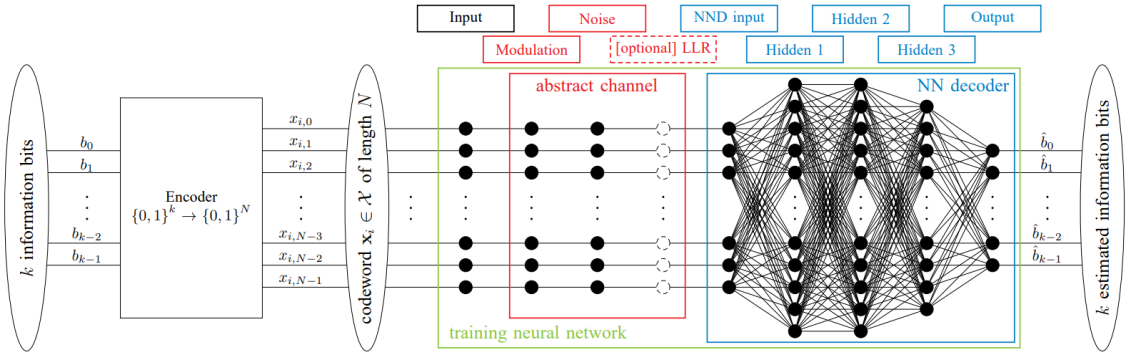

图 1: 框架示意图

开始时，先生成 k bit 能组合成的所有密文 (N bit)，然后依次调制、加入噪声，再送往神经网络进行训练。训练完毕后，再随机生成若干组密文，也是依次进行调制，加入噪声，最后经过神经网络进行解码，与原有的 k bit 数据进行对比，最后统计神经网络的学习效果。

其中在加入噪声时，需要考虑加入噪声的功率与原有信号的功率的比值，当噪声的功率远远大于信号的功率时，这时候信号的特征会被掩埋，从而使得神经网络的学习效果下降；当信号的功率远远大于噪声的功率时，训练出来的神经网络鲁棒性就会变差。

4 复现细节

原有的算法伪代码如下图的 Procedure 1 所示。其中分为两个部分，第一部分是神经网络的训练，首先生成所有的数据集，然后进行调制以及噪声的加入，再送入神经网络进行训练；第二部分是测试，随机生成数据集，同样经过调制，加入噪声，但测试时噪声的加入是需要根据不同的 SNR 来生成的，最后通过神经网络解码，得到结果图。

Procedure 1 Training and Testing Procedure

Input: data_len k, signal_len N, SNR, $isRandom$
Output: $Graph_{BER-E_b/N_0}$ G

// Creating data and Training
$data = \text{GreatAllData}(k)$
$signal = \text{Module}(data, N, isRandom)$
$x = \text{AddNoise}(signal, SNR)$
$\text{NN\_train}(x, data)$
// Testing
$data = \text{RandomGreatData}(k)$
$signal = \text{Module}(data, N, isRandom)$
$x = \text{AddNoise}(signal, SNR\_Start, SNR\_End, SNR\_Step)$
$y = \text{NN\_Decode}(x)$
$G = \text{Compare}(data, y)$

原本的开源代码有个问题，就是无法探讨神经网络是否能对神经网络未见过的数据进行解码。为此，我在原有算法的基础上，增加了这一功能，将数据集进行划分，分为测试集以及训练集，在测试时的数据都是训练时没见过的，这样就能探讨神经网络能否对未见过的编码进行解码这个问题了。其中训练集以及测试机的比重由一个参数 proportion 来决定。

Procedure 2 Training and Testing Procedure

Input: data_len k, signal_len N, SNR, $isRandom$, $proportion$

Output: $Graph_{BER-E_b/N_0}$ G

// Creating data and Training

$data = \text{GreatAllData}(k)$

$data\_train = \text{RandomPick}(proportion, data)$

$data\_test = data - data\_train$

$signal = \text{Module}(data\_train, N, isRandom)$

$x = \text{AddNoise}(signal, SNR)$

$\text{NN\_train}(x, data\_train)$

// Testing

$signal = \text{Module}(data\_test, N, isRandom)$

$x = \text{AddNoise}(signal, SNR\_Start, SNR\_End, SNR\_Step)$

$y = \text{NN\_Decode}(x)$

$G = \text{Compare}(data, y)$

5 实验结果分析

最终修改的实验结果如下图所示，可以发现，对于没有见过的编码，神经网络依然具有一定的解码能力，不过其解码的效果较其于用于训练的数据会比较差一些。另外，实验也发现，当测试集的比重变小时，神经网络的整体解码能力也会下降。

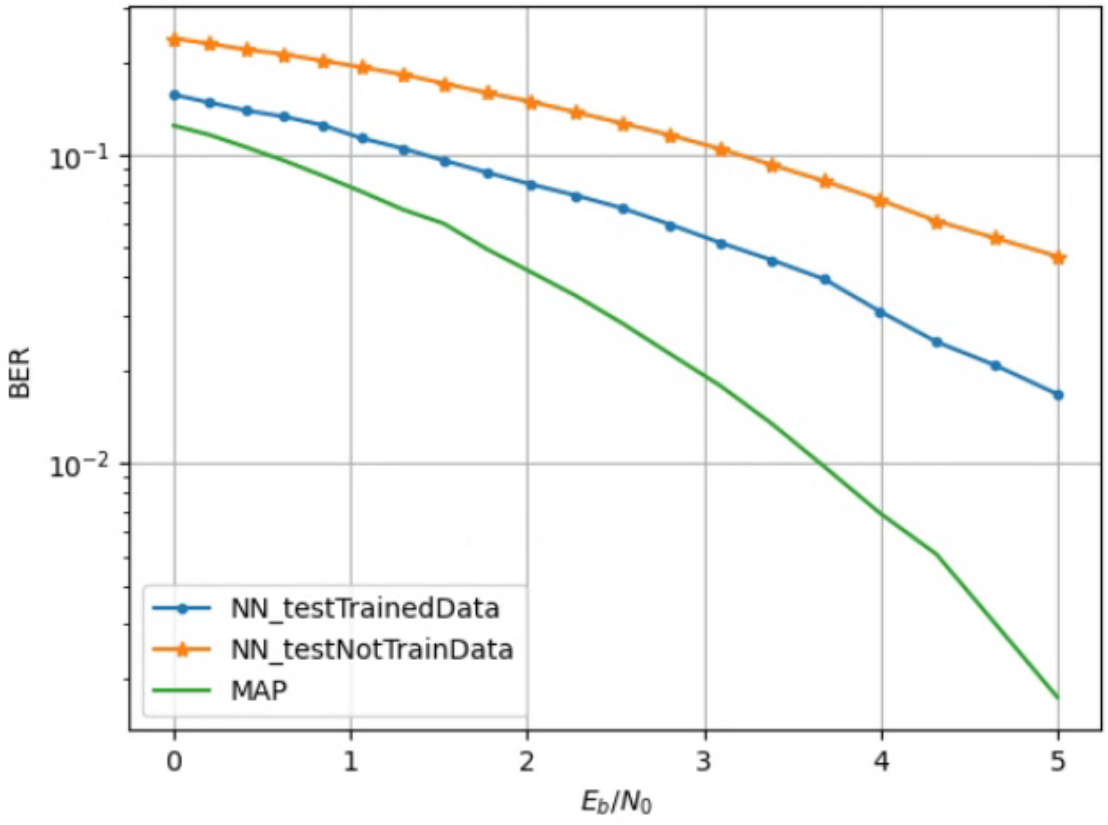

图 2: 实验结果示意

6 总结与展望

总结，本篇论文重点探讨使用神经网络进行解码的可行性，重点在于探讨两个问题，一是相对于随机化的编码，神经网络是否能更容易的学习结构化编码；二是对于没有见过的编码，神经网络是否依然能进行解码。源码对于第二个问题，并没有实现并且做出回答，我在源码的基础上，通过将数据

集进行划分，成功回答了这个问题。

未来可以改进的方向有很多，比如本文对于结构化编码，仅仅采用了极化码这种编码，可以采用其它的结构化编码从而得到更一般的结论。

参考文献

[1] GRUBER T, CAMMERER S, HOYDIS J, et al. On deep learning-based channel decoding[C/OL]//51st Annual Conference on Information Sciences and Systems, CISS 2017, Baltimore, MD, USA, March 22-24, 2017. IEEE, 2017: 1-6. https://doi.org/10.1109/CISS.2017.7926071. DOI: 10.1109/CISS.2017.7926071.

[2] WANG X, WICKER S B. An artificial neural net Viterbi decoder[J/OL]. IEEE Trans. Commun., 1996, 44(2): 165-171. https://doi.org/10.1109/26.486609. DOI: 10.1109/26.486609.

[3] ANDREW A M. *Brain Theory*, edited by Günther Palm and Ad Aertsen, Springer, Berlin, 1986 ix + 259 pp. (DM 96.)*Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. Vol 1: Foundations. David E. Rumelhart, James E. McClelland and the PDF Research Group. MIT Press, Cambridge, Mass., 1986, xx + 547 pp. (£27.50)*Parallel Distributed Processing: Explorations in the Microstructures of Cogition*. Vol 2: Psychological and Biological Models. Authorship, publisher, length and price all as for Vol 1[J/OL]. Robotica, 1987, 5(4): 344-345. https://doi.org/10.1017/S0263574700016404. DOI: 10.1017/S0263574700016404.

[4] CAID W, MEANS R. Neural network error correcting decoders for block and convolutional codes[C]//[Proceedings] GLOBECOM '90: IEEE Global Telecommunications Conference and Exhibition. 1990: 1028-1031 vol.2. DOI: 10.1109/GLOCOM.1990.116658.

[5] ABDELBAKI H, GELENBE E, EL-KHAMY S E. Random neural network decoder for error correcting codes[C/OL]//International Joint Conference Neural Networks, IJCNN 1999, Washington, DC, USA, July 10-16, 1999. IEEE, 1999: 3241-3245. https://doi.org/10.1109/IJCNN.1999.836175. DOI: 10.1109/IJCNN.1999.836175.

[6] WU J, TSENG Y, HUANG Y. Neural Network Decoders for Linear Block Codes[J/OL]. Int. J. Comput. Eng. Sci., 2002, 3(3): 235-256. http://www.worldscinet.com/ijces/03/0303/S1465876302000629.html.

[7] HINTON G E, OSINDERO S, TEH Y W. A Fast Learning Algorithm for Deep Belief Nets[J/OL]. Neural Comput., 2006, 18(7): 1527-1554. https://doi.org/10.1162/neco.2006.18.7.1527. DOI: 10.1162/neco.2006.18.7.1527.

[8] NACHMANI E, BE'ERY Y, BURSHTEIN D. Learning to decode linear codes using deep learning [C/OL]//54th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2016, Monticello, IL, USA, September 27-30, 2016. IEEE, 2016: 341-346. https://doi.org/10.1109/ALLERTON.2016.7852251. DOI: 10.1109/ALLERTON.2016.7852251.