Deep Learning for Wireless Interference Segmentation and Prediction

Sandeep Chinchali
csandeep@stanford.edu

Sameep Tandon
sameep@stanford.edu

Abstract

The proliferation of wireless devices ranging from smartphones to medical implants has led to unprecedented levels of interference in shared, unlicensed spectrum. Modern devices face the challenging task of estimating such interfering signals from received transmissions in order to effectively cancel interference in analog circuitry. In an effort to solve this problem, we use deep learning to segment corrupted wireless transmissions into a desired signal of interest and interference estimate. We exploit cyclostationary repeating patterns unique to manmade wireless signals to extract sparse feature representations of signals using autoencoders and k-means clustering. We then train neural networks and linear models to predict wireless interference from received signals. Our results indicate that neural networks and linear models trained with features extracted from deep learning are extremely effective at predicting interference, allowing upto about an 18dB gain in signal to noise ratio (SNR). Further, our neural network architecture is much more robust than conventional schemes such as adaptive filtering with the least means squares (LMS) algorithm, which rely on accurate estimates of interference. We demonstrate our results on synthetic traces of wireless interference and experiments from signal generator traces.

1 Introduction

As wireless communication becomes ubiquitous, a plethora of devices ranging from smartphones to medical implants are increasingly competing for use of unlicensed spectrum. Radios on these devices interfere with each other, limiting the real-world throughput and capacity of today's networks. To cope with this challenge, wireless devices must estimate the nature of interference and predict future time samples to cancel unwanted components of a corrupted received signal.

If not accounted for, interference can lead to corruption of data that leads to unnecessary retransmission of corrupted packets. Since this results in a degradation of network performance, solving the problem of wireless interference prediction will yield networks with robust performance even while multiple devices contend for limited, shared spectrum.

Solving the wireless interference problem is challenging because a wireless signal can be corrupted by a variety of other ambient wireless signals, such as Bluetooth, Wi-Fi, or Long Term Evolution (LTE) protocol transmissions. These signals can be further attenuated by the environment, such as walls and obstacles. The net randomness can make determining wireless interference a challenging endeavor. At the same time, wireless signals may exhibit specific periodic structure and cyclostationary features [3]. Modern wireless architectures have not fully realized the potential of utilizing this data to segment and predict wireless interference.

We present a data-driven approach to predicting and segmenting wireless signal interference. We use deep learning and stacked autoencoders to extract key features of wireless signals. Then, we use these learned features in neural networks to predict future time samples of wireless signals. We also use neural networks for segmentation in order to separate a composite corrupted signal into a signal of interest and ambient interference. To demonstrate the efficacy of our approach, we show our segmentation results lead to significant gains in signal to noise ratio (SNR) compared to conventional techniques such as adaptive filtering with the least means squares (LMS) algorithm.

2 Related Work

Prior research in the Stanford Networked Systems Group has led to Degrees of Freedom (DOF), an algorithm that identifies cyclostationary repeating patterns in manmade wireless signals to classify interference as belonging to WiFi, Zigbee, or Bluetooth protocols [3]. Support vector machines (SVMs) were trained to classify signals. However, the full potential of this work to forecast future interference samples guided by knowledge of signal type has not been realized.

In [5], Lee *et. al* use convolutional deep belief networks for audio classification by learning features for a spectogram of various musical and speech time series. In a similar vein to our goal of wireless signal segmentation, Kolter *et. al* use discriminative sparse coding to segment an aggregate energy usage profile into the usage data of different appliances [4]. Our work differs from [5] since it addresses time sample prediction while it differs from [4] since we use neural networks and deep learning for wireless interference estimation.

3 Methods

Interference segmentation and prediction may be cast in the standard learning framework. Formally, let $\mathbf{x} = (x_t, x_{t+1}, \ldots, x_{t+N-1})$ and $\mathbf{y} = (y_t, y_{t+1}, \ldots, y_{t+N-1})$ be a vector of N contiguous samples of the composite and interference signals respectively. The composite signal $\mathbf{x}$ is the summation of the desired signal of interest $\mathbf{s}$, the interference signal $\mathbf{y}$, and additive Gaussian noise $\mathbf{n}$. We denote the protocol associated with the interference by a subscript so the composite measurement of a WiFi packet corrupted with Zigbee interference is given by $\mathbf{x} = \mathbf{s_{wifi}} + \mathbf{y_{zig}} + \mathbf{n}$.

For segmentation, the goal is to predict the *current* interference $\hat{\mathbf{y}}$ given the current composite signal $\mathbf{x}$. For prediction, the goal is to predict the *future* interference given the current composite signal $\mathbf{x}$. Formally, we will overload notation and denote the future interference as $\mathbf{y} = (y_{t+N}, y_{t+N+1}, \ldots, y_{t+2N-1})$ (the disambiguation of notation depends on the problem being solved; e.g. segmentation or prediction). We obtain samples $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ to generate a training and testing data set from a Matlab synthetic packet generator script and lab experiments.

3.1 Interference Segmentation and Prediction

At a high level, our system performs the following steps:

1. Preprocess the data using a PCA Whitening Scheme.
2. Execute an Unsupervised Learning Algorithm to learn a sparse feature representation
3. Train a learning algorithm to predict interference.

3.1.1 Preprocessing

The data is preprocessed using PCA Whitening. First, each $x^{(i)}$ is normalized by subtracting off the mean. We then run PCA on the entire data set, removing dimensions with variance less than a specified threshold. In practice, we found that the threshold did not matter much. The resulting dimensionality-reduced data is fed into an unsupervised learning algorithm.

3.1.2 Unsupervised Learning Algorithms

We evaluated two unsupervised learning algorithms for learning a sparse feature vector representation: K-means and Sparse Autoencoders. These algorithms were chosen because of their success in practice; e.g. [2].

Formally, denote the feature vector as $f(x) : \mathbb{R}^N \to \mathbb{R}^K$ where x is the input after preprocessing.

In brief, K-means clustering is an iterative algorithm that finds K centroids and assigns each data point $x^{(i)}$ to a cluster $c_k,\ k = 1, \dots, K$. The corresponding sparse feature representation is $f_k(x^{(i)}) = 1\{x^{(i)}$ assigned to cluster k$\}$. Since this feature representation is overly sparse, a "soft" K-means assignment is used [2]. In particular, this representation is

$$f_k(x) = \max\{0, \mu(d) - d_k\}$$

where $d_k = \|x - c_k\|$ and $\mu(d) = \frac{1}{k}\sum_{i=1}^{K} z_k$. In practice, this representation is typically half-sparse.

The second algorithm we use is a Sparse Autoencoder with K hidden units. This algorithm learns weights $W \in \mathbb{R}^{K\times N}$, biases $b \in \mathbb{R}^N$, such that $f(x^{(i)}) = g(Wx^{(i)} + b)$ where g is a sigmoid function. The autoencoder has a reconstruction term (using the feature vector representation) to reconstruct the input and a sparsity target term on the feature vector. If the number of hidden units is small, the autoencoder will try to learn a sparse, dimensionality-reduced representation, using the underlying structure of the data [1].

3.1.3 Supervised Learning Step

Using the feature representation, we considered two supervised learning algorithms: linear regression and feedforward neural networks. These algorithms were trained using input $f(x^{(i)})$ and output $y^{(i)}$. As noted above, $y^{(i)}$ is the observed interference for the segmentation problem or the future observed interference for the prediction problem.

3.2 Adaptive Filtering: Least Mean Squares

As a baseline comparison for our segmentation approach, we explored the conventional technique of adaptive filtering. Adaptive filtering uses the least means square (LMS) filter for adaptive noise cancellation (ANC) [6]. In adaptive filtering, a corrupted signal $\mathbf{x}$ is input to the LMS filter along with a reference interference signal $\mathbf{y_{ref}}$ which may differ from the actual interference $\mathbf{y}$. The LMS filter adjusts its impulse response via the Widroff-Huff learning rule discussed in class to estimate the actual interference $\mathbf{y}$ that corrupts the signal of interest.

4 Experiments

4.1 Validation with Synthetic Data

To generate synthetic data, we built upon starter Matlab code from the Stanford Networked Systems Group which generates WiFi and Zigbee protocol packets at various channels (frequencies). We then attenuate the desired WiFi packet by a factor α with respect to the Zigbee interference and corrupt the interference with low power additive Gaussian noise $\mathbf{n}$ to generate a composite signal $\mathbf{x} = \alpha\mathbf{s_{wifi}} + \mathbf{y_{zig}} + \mathbf{n}$. Signal parameters are given for the experiments detailed in Figure 3. Each training and test example consisted of $N = 60$ consecutive time samples (sampled at 240 MHz for synthetic data). An amount $m = 2000$ training examples of size N were randomly selected from the first half of the full composite signal $\mathbf{x}$ and interference time series $\mathbf{y}$. The second half of the time series was reserved for testing data.

4.2 Measuring Interference for Training Data

To obtain real traces of wireless interference, we used a signal generator to create a signal of interest at a specified amplitude such as an 8 dBm Bluetooth trace at a 2.5 GHz center frequency. We simultaneously used another signal generator to create an interfering signal such as an 8 dBm WiFi trace. The signal of interest and interfering packet were added using a signal combiner and the resulting composite waveform was sampled at $25MHz$. Though we measured the composite signal directly, we could not simultaneously measure the signal of interest since we only had one spectrum analyzer. Hence, we followed the same approach as for the synthetic data of artificially combining a signal and interferer, except in these cases the signals were real packets from a signal generator.

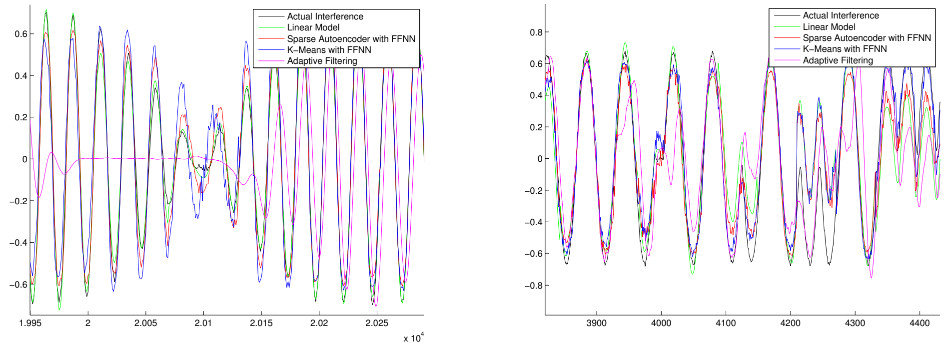

Figure 1: Left: Comparison of various schemes for *estimation* on synthetic data corresponding to experiment in col 2 of Figure 3. The Zigbee interference is estimated well by all schemes except conventional LMS since it is provided a reference at a different channel compared to the actual interference. Right: All schemes work worse with real data since it is more erratic and noisy. The plots correspond to col 4 of Figure 3.

4.3 Results

Each column of Table 3 indicates a specific interference experiment. Columns 1, 2, 3, 5, and 6 combined synthetic data packets while columns 4 and 7 correspond to packets measured on a signal generator that were artificially combined in Matlab. The autoencoders and all neural networks were trained for 5000 epochs, which was ample time for convergence. The autoencoder and neural network training and setup was done with the Matlab Deep Learning Toolbox [**?**]. For each interference experiment, we compared the performance of the following models.

1. **k-means NN:** k-means with $k = 20$ centroids is used in feature selection for a neural network (NN) with 1 hidden layer of 50 units.
2. **SAE NN:** An autoencoder with 20 units was used for feature selection and then a neural network with 1 hidden layer of 20 units was used to refine the weights.
3. **LM:** Linear Regression using PCA whitened data.
4. **LMS:** Least means square (LMS) filter predicts the interference given a reference signal.

To compare the efficacy of various models, we compute the signal to noise ratio (SNR) in decibels (dB). For segmentation problems, the SNR is computed as the true signal vs the segmented signal (composite minus interference) and for prediction problems, the SNR is computed as the predicted interference vs the actual interference. We also report the SNR for the case when the composite signal is stored without any cancellation as a baseline. Clearly, the linear model outperformed the others while the neural network approaches performed similarly. In nearly all cases, all models provided significant SNR improvement over the case when no cancellation was done. We see that LMS filtering performs badly since it is not robust to errors in reference selection. For example, when the reference estimate of the interference is of the correct protocol but at a wrong channel, adaptive filtering is quite off while the other schemes still predict interference well.

5 Conclusion

Overall, our experiments illustrate that simple linear models operating on data pre-processed with PCA are extremely useful for predicting interference followed by neural networks trained on features selected by k-means and stacked autoencoders. The results are promising since these methods worked with actual signals measured in the lab and provided extremely large SNR gains compared to the case when no cancellation or adaptive LMS filtering were enacted. Further validation comes from the observation that linear fitting and neural networks performed better than conventional LMS filtering because the adaptive filter needs an accurate interference reference. In many wireless coexistence scenarios, 10dB gains from cancellation are considered quite large. Since we significantly

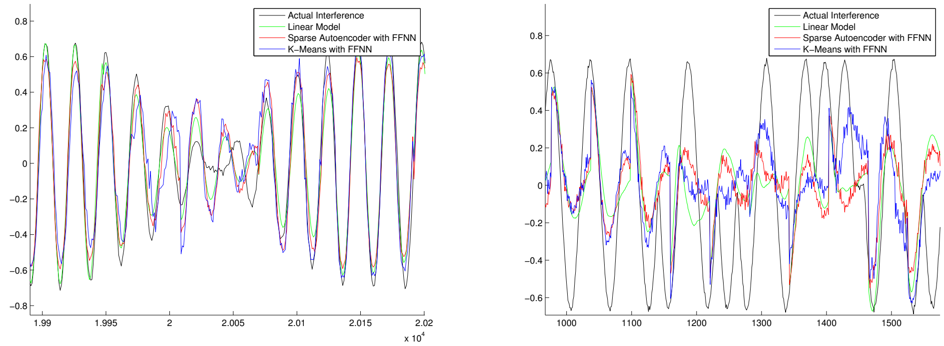

Figure 2: Left: Comparison of various schemes for *prediction* on synthetic data corresponding to experiment in col 5 of Figure 3. Right: All schemes work worse with real data since it is more erratic and noisy. The plots correspond to col 7 of Figure 3.

Method	S1	S1*	S2	S3	P1	P2	P3
k-means NN	2.59	2.59	1.00	-7.57	9.18	8.12	0.93
SAE NN	2.12	2.12	-0.26	-7.57	9.20	8.46	1.10
LM	7.42	7.42	1.15	-9.46	9.26	8.98	1.35
ANC	-2.49	-14.56	-3.16	-11.35	–	–	–

Figure 3: Results. Reported are SNR (dB) results for (S)egmentation and (P)rediction tests. For segmentation problems, the SNR is computed as the true signal vs the segmented signal and for prediction problems, the SNR is computed as the predicted interference vs the actual interference. The number represents the WiFi channel evaluated on (hence "S1" corresponds to segmentation test on Wifi channel 1). S1* is an example of a poor reference provided for ANC. For all the tests, the linear model performs the best, followed by the neural networks, and trailed by ANC.

pass this threshold in many experiments, our work may be incorporated into an actual cancellation circuit which could pass the interference estimate from a neural network or linear model to an adaptive filter for further refinement in a two-stage pipeline.

Without doubt, society's insatiable demand for wireless connectivity will make current challenges for wireless coexistence and interference management more acute. Our work illustrates that machine learning techniques can serve an integral role in a holistic solution for interference cancellation that will hopefully bring us closer to realizing ideal, cooperative wireless devices.

6 Acknowledgements

We thank Professor Sachin Katti, Kiran Joshi, and Dinesh Bharadia from the SNSG lab for advice.

References

[1] Unsupervised learning and deep learning tutorial. http://ufldl.stanford.edu/wiki/, 2011.

[2] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. *Journal of Machine Learning Research*, 15:215–223, 2011.

[3] S. Hong and S. Katti. Dof: A local wireless information plane. In *ACM SIGCOMM*, 2011.

[4] J.Z. Kolter, S. Batra, and A.Y. Ng. Energy disaggregation via discriminative sparse coding. In *Neural Information Processing Systems*, 2010.

[5] H. Lee, Y. Largman, P. Pham, and A.Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *NIPS*, 22:1096–1104, 2009.

[6] B. Widrow, J.R. Glover, Jr., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, Jr. E. Dong, and R.C. Goodlin. 'adaptive noise cancelling: Principles and applications. In *Proceedings of the IEEE, 63(12):1692-1716*, December 1975.