DeepPhy

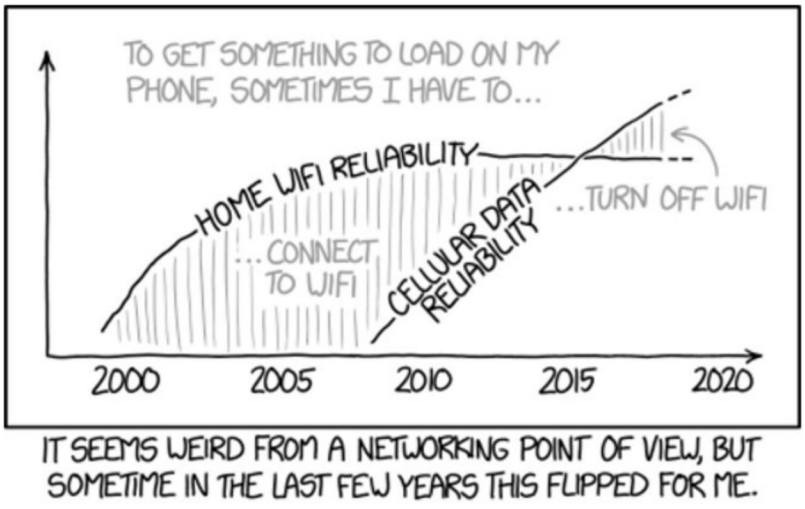
Exploration of Deep Learning Application to Modem UE Physical Layer Design

Nate Chizgi

CS230: Deep Learning, Winter 2018

Stanford University, CA

nchizgi@stanford.edu



[16]

ABSTRACT

Modem companies, such as Qualcomm, Intel and Samsung, continue to follow an approach of splitting signal processing tasks into a chain of independent and isolated routines, each individually optimized based on foundations within information and communication theory for their purpose. However, there are non-linearities within the system (ADCs, non-linear power amplifiers, finite duration sequences, RF & IC imperfections), which cannot be accurately captured by these standard processing algorithms. For this reason, machine learning (end-to-end deep learning in particular), can be used to improve or optimize the overall system performance since it removes both the assumptions of an independent modular construction being the best approach, as well as there being a mathematically determined solution.

However, applications of deep learning to embedded systems has proven costly in ASIC area and power, which has kept most applications in this field purely within an academic setting. There have been recent advances in neural network compression and sparse processing which has reduced the complexity of inference by several orders of magnitude, and additional work has been put into specialized ASIC solutions which reduce the power requirements of such networks further, putting them within reach to physical layer design.

This project is focused on summarizing existing research in power and complexity reduction techniques for deep neural network inference for embedded applications, as well as discussing applications of deep learning on modem baseband physical layer design (3G, 4G, and 5G centric discussion) and corresponding recommended neural network architectures, input and output training data-sets, and evaluation techniques.

I. INTRODUCTION TO MODEM CONSTRAINTS

Deep learning is proving to revolutionize many fields such as healthcare (ECG analysis, X-Ray based diagnosis, prescription recommendations, life expectancy diagnosis, etc), autonomous driving (object classification and detection, control decisions), and mobile applications (face & fingerprint recognition, targeted advertising, routing, user preferences, etc). Much of this has been through deep learning advancements in computer vision, natural language processing and speech recognition. Within the wireless physical layer realm, there has been ongoing research and application to network side optimizations (self-organizing networks, scheduling algorithms, beamforming, etc), however the user equipment (UE) domain has largely not yet been influenced by deep learning due to the impact on device power consumption, silicon area, latency and flexibility needed for rapid technological advancement.

A. Modem Power

Smartphones are often classified as having some sized battery with a certain duration of voice or data operation. During a voice call for example, the modem has been shown to consume upwards of 75% the overall device power. Similarly, for gigabit-LTE, the data connection is often the dominant power consumer, even higher than the display and application processor. An example spec from the Galaxy S9+ directly lists the following specs which are predominantly determined by the modem power consumption:

- Battery: 3500 mAh
- Talk Time: Up to 25 hours
- Internet Use (4G): Up to 15 hours

Physical layer processing for 3G and 4G frames is on the order of 1-2ms, and for 5G on the order of 125us for tasks such as channel estimation (CE), channel state information (CSI) feedback, interference cancellation (IC), or signal classification. Many of these have tight processing deadline requirements or are in the critical end to end processing path and require simplified or hardened control and data flow.

A typical voice power target for modem baseband + RF may be on the order of 25-100mW, whereas peak data rate LTE or 5G may be on the order of 500-2500mW.

B. Modem Area

The modem industry is highly competitive, with several large players vying to be the predominant supplier for the most popular smartphone OEMs around the world. Once a certain scale is achieved, the primary cost is no longer the upfront R&D or non-recurring engineering cost, but rather the recurring bill of materials (BoM). Often, there is small operating margin on the chips shipped, and a company with a 25% larger solution may sell at cost or even a small loss to be competitive with other companies.

For this reason, modem companies are often at the forefront of technology nodes. Despite the vast challenges of being the first to mass produce at a new technology node, it often pays off richly down the line with a reduced BoM. Total package sizes for modem baseband on the order of 100 mm^2 are common for even the most advanced technologies such as gigabit LTE or 5G, which largely limits on-chip memory sizes.

C. Modem Advancement

In addition to the above, modem technologies continue to advance at a rapid rate, with new standards released continuously to allow for higher throughput, ultra-reliable connections, device-to-device networking, automotive assistance, and other purposes. To keep on the cutting edge of technology, companies have resorted to multiple highly complex chipsets shipped per year, sometimes even before standards are completed.

This has two major effects in the architecture and design of a modem: 1) much of the control and datapath, outside of core kernel operations are not hardened or are heavily parameterizable to allow for flexible configuration based on actual network deployment and performance seen in the field, and 2) the hardware accelerators implemented tend to be low level and tied together through complex control software. This is done to de-risk the system architecture and allow for future feature-set expansion without a fresh tape-out.

II. EMBEDDED DEEP LEARNING ADVANCEMENTS

Having deep learning applications flourish within the modem physical layer requires optimizations to standard deep networks to fit within the modem constraints. There has been related research in this field over the past few years in how to manage the memory requirements of a deep network, and how to reduce the computational overhead of such networks.

A. Memory Requirements

Advancements in deep learning (DL) have largely come about through larger neural network architectures, more advanced detection algorithms and big data. It is not uncommon for a neural network (NN) to have more than 1M floating-point parameters, and some of the best networks have more than 100M parameters. Some examples of modern deep networks used primarily for image classification are AlexNet (61M parameters), and VGG-16 (138M parameters). These can require more than 256MB of memory just to store parameters for inference, orders of magnitudes larger than allowed on a modem physical layer solution. Due to the short timescales and large data involved with many physical layer applications, it is required to run the inference algorithms (and in some cases limited online training), directly on chip.

There has been research in NN compression algorithms which bring this within reach. One such compression algorithm is Deep Compression [4], which has achieved between 30x – 50x reduction in parameter storage requirement without loss in performance. The AlexNet network for example had parameters compressed from 240MB to 6.9MB – which is very close to fitting within the modem domain.

Another advancement in reducing NN architecture size without sacrificing performance has been with DenseNets [7]. These have additional connections between earlier and later layers rather than a simply connected network. This architecture outperformed other state of the art architectures with less than half the number of free parameters. Additionally, signal processing related DL algorithms tend to be on quantized 1D input IQ streams rather than 2D images or 3D image pairs, thus reducing the dimensionality of the input and potentially the overall network size.

B. Processing Requirements

Another problem with applying deep neural networks to physical layer is the processing power and latency introduced through the serial layer processing which takes of the form: $f^L(f^{L-1}(\ldots f^1(x)))$. Many state of the art NN's have 10 or more layers, with the most complex ones having more than 130 layers. Even with DenseNets and compressed parameters, there is still a lot of serialized processing required.

Recent research has been done to improve the latency and energy efficiency of running inference and training on neural networks. An example of this is the Sparse CNN (SCNN) accelerator architecture [8] which exploits zero-valued weights arising from pruning during training and from the ReLU operation (max(0,value)) during inference. During inference, it is not uncommon for 70% or more of the activations to be clamped to zero from the ReLU operation. The SCNN architecture has shown between 2.5x – 3.5x speedup in processing. There are other architecture papers ([12], [13]) mostly targeting removing multiplications with zero-valued activations or weights and storing locally only the compressed activation values.

III. MODEM PHYSICAL LAYER APPLICATIONS

There are a wide variety of potential applications of DL to physical layer. Below is a sampling of ideas based on discussions with experts in the field, and a review of research papers. I have broken the topics into three categories based on the primary operating modes of the UE modem baseband: (a) idle/acquisition, (b) voice, (c) data. Discussion is focused on physical layer functionality only and does not review applications through the stack, although some papers exist on the topic, e.g. [15]. Additionally, I have focused on practical DL applications to the existing 3G, 4G and 5G specifications, instead of proposing new theoretical specifications which has been done in other papers, for example those related to end-to-end autoencoders [1].

A. Idle/Acquisition

In idle mode or acquisition, the modem is primarily responsible for

- Initial frequency scan: finding/choosing a technology (2G/3G/4G/5G) and frequency band to camp on
- Idle mode camping & measurement: Measuring and retaining a strong link over time
- Cell selection: retaining a strong connection to a cell or handing off to a different cell when necessary
- Paging channel demodulation: Decoding periodic paging indicator channel from the network
- Interference classification: Classifying interfering signal from raw input IQ

For initial frequency scan, often an overall power is estimated from input baseband IQ samples on each frequency of interest. However, if a 3G waveform is being searched, a WiFi signal or a 2G or 4G signal may show as high 3G in-band power. This would often lead to the UE going through a full search on this frequency of interest, only to determine that there is no actual 3G signal. A deep learning technique could instead attempt to classify the technology type from the raw input IQ stream whenever sufficient in-band power is detected.

Similarly, for cell selection, detection of poorly configured cells, or making decisions on when to hand off to another cell can be learned through deep learning techniques. Outside information such as positioning information, mobile speed and prior cell statistics can be used to form inputs to a system and do real time training for handoff and selection. Specific application to 4G vehicular cellular technology (V2X) is discussed in [14] where reinforcement learning is used to address resource allocation and management.

B. Voice

Voice connection is unique in that it is an 'always on' state of the modem with relatively low data-rate. Statistics such as call-drop rate and voice quality are important to the user experience, whereas channel power is traditionally important to the network side resource allocation. Additionally, in voice mode of operation, the display is typically turned off, and relatively few non-essential applications are running concurrently. Therefore, the modem dominates the overall power consumption of the device.

An example of a potential DL application to voice is the SIR target determination. The basic problem here is that to achieve 1% or 0.5% block error rate (BLER) target on a channel transmitted every 20 ms, you need a long history of frame performance. At call establishment, this is typically overestimated to guarantee performance and it can take several seconds to settle down to the proper targeted SIR. Additionally, temporal variations in the channel may require fast changes in the SIR target, e.g. upon exiting an elevator or going around a building corner. Applying DL by taking the channel realization, tracking loop behavior, past statistics and other inputs to decide a proper SIR target more rapidly can allow network resources to be better utilized.

Another example related to power savings in voice systems relates to early frame decoding. In 3G, the voice frames usually are 20ms in duration with 1/3 code rate. With a 1% BLER target, it is often possible to successfully decode the voice frame prior to the 20ms frame boundary and pass CRC. In addition to the CRC-encoded voice frame, there are often non-CRC encoded frames which need to have a reasonably low (non-zero) BER. DL can be applied to determine: (a) at what point within the frame to attempt the early decode, (b) applying a sufficiency condition on the non-CRC encoded channels which does not impact voice quality.

Mean opinion score (MOS) determination and evaluation is also an area where deep learning algorithms could be applied. Many power features attempt to leave the RF and baseband portion of the modem in a disabled state for as long as possible. Often this comes at the expense of reduced performance at the boundary of silence to voice transition, either due to insufficient tracking loop training, incorrect voice frame transition detection, or other related factors.

C. Data

At the heart of the smartphone revolution is high speed mobile broadband – the latest commercially available modem solutions achieve peak throughputs of more than 1 Gbps and development is underway on peak 4G data-rates of nearly 4 Gbps and second generation 5G rates topping 8 Gbps. Within the realm of 3G, 4G and 5G data processing, deep learning has many applications.

Channel State Feedback (CSF) determination for 4G consists of Channel Quality Information (CQI), Rank Indication (RI) and Precoding Matrix Information (PMI). Currently this is determined based on traditional information theory models, however there is no closed form computation for this. Applying DL techniques to this problem by taking the channel realization as an input and directly outputting the CSF information could provide a better and more complete solution than traditional techniques which rely on a variety of separately computed parameters (SNR, doppler estimate, channel type classification, tracking errors, etc).

Decoder design has also been studied in the context of DL and recent work has presented RNN architectures which have better results and reduced complexity relative to traditional belief propagation methods [2]. Additional work can be done here in specific context of 3G, 4G and 5G 3GPP specification decoders such as LDPC, Turbo Decoder (TDEC), Polar Decoder, and Viterbi Decoder. DL may be used to help inform decoder design by tuning control parameters based on real world and simulated channel models. Determination of the "quality of failure" is also a useful learned result when deciding on number of decoder iterations to run or compression that can be done on the Hybrid Automatic Repeat Request (HARQ) soft bits, etc. Additionally, evaluation of the optimal storage of HARQ soft bits can help reduce overall bus traffic if this information is stored off-chip, with minimal loss of overall performance.

IV. SPECIFIC MODEM EXAMPLE

A more detailed discussion on deep learning frameworks, architectures and applications are discussed below using an example idle mode application. This specific topic does not appear to have prior publicly released research papers. Due to data availability constraints, these networks have not been implemented beyond a trivial proposed training and inference implementation. It remains a future item to collect sufficient dataset and fine-tune the architecture and hyper-parameters of the network.

A. Paging Inidicator Channel (PICH) demodulation

A user equipment (UE) device is paged by the network in 3G when another user initiates a phone call, or there is a data push notification from the network. This notification comes in two parts, a paging indication (PI = 1 or 0), followed conditionally by a paging message if the PI is 1. UEs on the network are required to always decode this periodic PI, in case there is a page from the network.

The periodicity of this decode is configured typically between 640ms and 2.4s, and accounts for a large percentage of the UE power consumption when a device is sitting idle. A missed decode leads directly to a call establishment failure which negatively impacts the user experience, whereas false positives lead to a long 'awake' time to demodulate a paging message which doesn't exist which negatively impacts battery life.

To successfully decode PICH in as short a time as possible, typically there are multiple independent stages run: a) Run searcher to get path profile & timing information, b) Estimate and correct frequency error, c) Track timing error, d) Obtain cleaned channel estimate (e.g. RAKE or Equilizer), e) get paging channel symbols, and f) make decision on whether UE was paged. This process can be run on IQ samples collected in real time, or offline on a set of samples, collected when RF was enabled to minimize the duration which RF is enabled on.

The DL algorithm proposed is a binary classification algorithm which takes as input a 5120-length (roughly 0.66ms) segment of complex 8-bit IQ samples which overlap the full duration of the PI, and outputs a '1' or '0' based on the paging indication. The complex valued IQ inputs have been interleaved and flattened which doubles the input size and convolution output since specific OVSF and scrambling sequences need to be learned for the QPSK coded common pilot and paging indicator channel

Layer	Layer Parameters			Output Shape		# parameters	Parameter	Compressed
	f	s	h	# chan	x		Size (kB)	Size (kB)
Input (Flattened 8-bit I/Q samples)				1	10240	0	0	0
Convolution (f=1x32) + ReLU	128	1		128	10209	8192	64	4
Max Pooling (f=1x2)		2		128	5104	0	0	0
Convolution (f=1x64) + ReLU	256	1		256	5041	16384	128	8
Max Pooling (f=1x2)		2		256	2520	0	0	0
Convolution (f=1x64) + ReLU	256	1		256	2457	16384	128	8
Max Pooling (f=1x2)		2		256	1228	0	0	0
Convolution (f=1x64) + ReLU	256	1		256	1165	16384	128	8
Max Pooling (f=1x2)		2		256	582	0	0	0
Flatten					148992	0	0	0
Fully Connected + ReLU			512		512	76283905	595968	9312
Fully Connected + ReLU			128		128	65537	512	8
Binary Classification			1		1	129	1	0
				TOTAL (M)		76.4M	582.9MB	9.1MB

V. CONCLUSIONS

The modem user equipment physical layer domain appears to be an area not yet explored in the context of modern deep learning techniques. Recent advancements in neural network compression, techniques to minimize processing power, and specialized ASIC designs have made it feasible to incorporate deep learning into commercial 3G, 4G and 5G solutions. A couple of simple examples have been briefly explored in the context of 3G idle mode processing, and dozens of other ideas have been highlighted across idle, voice and data processing. The more promising applications appear to be within the 4G and 5G domains, specifically as a tool to influence the demodulation and modulation design, however this is left as future work to explore.