Performance Comparison between Machine Learning based LTE Downlink Grant Predictors

Abdullah Nisar¹, Jonathan Ah Sue¹, Jürgen Teich²
¹Intel Deutschland GmbH, Munich, Bayern, Germany
²Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract - Deep learning has been finding applications in many fields. In particular, recent investigations show the potential to use machine learning techniques to improve the power consumption of cellular mobile devices. The motivation of this paper is to compare the performance between a recurrent neural network algorithm called long short-term memory and a feed forward neural network for downlink grant prediction in LTE modems. Based on this prediction, specific modem sub-components can be turned off in order to save a significant amount of power. However, as the computational complexity of the algorithms induces additional power consumption, a trade-off between the complexity and the prediction accuracy of the predictor is needed. Therefore, we evaluate the two algorithms in terms of number of floating point operations and area under the receiver operating characteristics curve. Analysis carried out in this paper have shown an average improvement of around 4.7% by using long short-term memory instead of feed forward neural network.

Keywords: Long Short-Term Memory, Feed Forward Neural Network, LTE Downlink Scheduling, Machine Learning, Grant Prediction

1 Introduction

Long Term Evolution (LTE) was started as a standardization project by the Third Generation Partnership Project (3GPP) body in 2004 [1]. LTE provisions enhanced data rates up to the Gigabit range. In LTE, packet scheduling, in the physical layers is exercised every Transmission Time Interval (TTI) which is 1 ms. Downlink grant refers to the allocation of resources by the network to communicate with the User Equipment (UE). The UE has no information whether it will receive a grant in the current TTI or not. In LTE mobile devices, the modem power consumption is crucial and substantial amount of power is used to monitor control channels for grant status although no valuable information is in there (in continuous reception mode). Grant allocation is a decided by the base station and from the UE perspective it is not deterministic. Predicting the content of this control channel can save a significant amount of power as shown in [2]. As grant prediction is a time series prediction task, we propose to use long short-term memory (LSTM), a recurrent neural network (RNN) that shows significant improvements for classifying time series sequences [3]. However, this implementation should be computationally dense, and it will have an impact on the total power saving outcome compared to predictor that uses a normal feed forward neural network (FFNN). Therefore, we propose a comparison between LSTM in terms of number of floating point operations (FLOPs) and area under the receiver operating characteristics curve (AUC) for different architectures. Section 2 summarizes the basic LTE functionalities, architecture and downlink resource allocation. Section 3 summarizes the basics about FFNN, RNN and LSTM. Section 4 gives an overview about the test setup, experimentation and results. Finally, in Section 5 conclusion will be drawn.

2 Long Term Evolution

The fast growth of mobile traffic volume was driving the standardization and development of the 4th Generation of Mobile Communication. Due to the increase in data driven applications, the number of LTE subscribers have crossed 1 billion worldwide [4].

The sections below describe the basic LTE Architecture, downlink scheduling and algorithms, and input parameters for the predictor.

2.1 LTE Architecture

LTE has a flat, IP based architecture. Primarily it is comprised of three main components [5].

- The User Equipment (UE).
- The Evolved Packet Core (EPC)

![Basic LTE Architecture](image)

Figure 1: Basic LTE Architecture [5]

The UE is an equipment that contains a universal subscriber identity module (USIM) e.g. a cellphone, tablet or notebook etc. and communicates with the E-UTRAN.
The E-UTRAN is responsible for handling all the radio communications between the UE and the EPC and is comprised of only the evolved base stations, called eNodeB or eNB. Each eNB manages all radio related functions in the fixed part of the system and works as bridge between UE and the EPC. In LTE, a specific component of the eNB, called Medium Access Control (MAC) scheduler, is responsible to schedule the radio resources in time and frequency.

The Evolved Packet Core (EPC) is LTE network framework for providing converged voice and data. It contains the network operator's subscriber database, communicates with other Packet Data networks (PDN), acts as router for data forwarding and is responsible for initiating paging and authentication of the mobile device. Downlink refers to the data transfer from the network to the UE and uplink refers to the data transfer from the UE to the eNB.

2.2 LTE Downlink Scheduling

Scheduling in LTE is done dynamically by the eNB every TTI. LTE frame structure is given in Figure 2. The smallest unit of resources that can be allocated to a user is called resource block (RB).

![LTE Frame Structure](image)

Figure 2: LTE Frame Structure [6]

Since LTE is a packet switched network, UE spends a significant amount of time decoding the Physical Downlink Control Channel (PDCCH), a physical channel that contains the scheduler signaling, to inform the UEs about the downlink resource block assignments to Physical Downlink Shared Channel (PDSCH). The PDCCH will be sent in each subframe shortly before PDSCH starts. If the grant is present in PDCCH, UE then receives the PDSCH and decodes the information. If there is no grant the UE switches off right after PDCCH decoding. In the latter case the UE’s power is wasted when it receives and decodes the PDCCH. The MAC Scheduler assigns resource blocks to different UEs.

2.3 LTE Downlink Scheduling Algorithms

The 3GPP standards do not outline specific scheduling algorithms. Therefore, the eNB manufacturers define their own algorithm to optimize metrics like fairness or throughput. However, these algorithms are not publicly available as they provide differentiation between eNB manufacturers.

In order to evaluate the parameters that impact the scheduling decision and that should consequently be used as predictor input, we provide below a review of some of the common scheduling algorithms available in the literature [7].

2.3.1 Max Throughput

This algorithm allocates RB to the user which experiences the best channel conditions, thus providing the maximum overall throughput i.e., the UE experiencing the best channel quality will always be scheduled. The priority metric for \( i \)th user on \( k \)th RB can be expressed as

\[
 p_i^k = \max_i \left( \frac{d_i^k(t)}{R_i(t-1)} \right) ; 1 \leq i \leq N, \tag{1}
\]

where \( N \) is the number of users in a network and \( d_i^k(t) \) is the expected data rate for \( i \)th user at time \( t \) on \( k \)th RB. The goal of this algorithm is to maximize the cell throughput. But cell edge users might suffer as they may not always have good channel conditions. It provides unfairness in resource sharing among UEs.

2.3.2 Proportional Fair (PF)

This algorithm provides balance between fairness and spectral efficiency among UEs. The priority metric used for PF for \( i \)th user on \( k \)th RB can be expressed as

\[
 p_i^k = \max_i \left( \frac{d_i^k(t)}{R_i(t-1)} \right) ; 1 \leq i \leq N, \tag{2}
\]

where \( N \) is the number of users, \( d_i^k(t) \) is the expected data rate for \( i \)th user at time \( t \) on \( k \)th RB and \( R_i(t-1) \) is the past average throughput up to time slots \( t-1 \). The priority metric is maximized when a user experiences bad channel conditions, which allows resources to be allocated to those poor channel users. The past average throughput can be calculated as follow:

\[
 R_i(t) = \left(1 - \frac{1}{T}\right) R_i(t-1) + \frac{1}{T} \eta_i(t) \tag{3}
\]

where \( T \) is the fairness window and would be equal to one TTI. \( \eta_i(t) \) is the data rate achieved by the \( i \)th user at time \( t \). The main goal of this algorithm is to achieve balance between fairness and highest cell throughput.

2.3.3 Round Robin

This algorithm allows users to take turns in using the shared resources, without taking the instantaneous channel conditions into account. Therefore, it offers great fairness among the users in radio resource assignment, but degrades the overall throughput [8].

2.4 Predictor Input Parameters

Based on these algorithms and their inputs, we propose the following list of predictor input parameters:

- DL Transport Block Size Code Word 0
- DL Transport Block Size Code Word 1
- DL Modulation and Coding Scheme Code Word 0
- DL Modulation and Coding Scheme Code Word 1
- DL Negative Acknowledgement (NACK)
Previous DL Grants

This is probably not the complete list of parameters that are used by the eNodeB but they give some indications on the past grant history. Even though the information used as input is only a small part of the metrics used by the eNodeB MAC scheduler, it is believed that the proposed parameters contain valuable information that can be extracted by the predictor to infer the UE’s future grant allocation by learning from past patterns.

3 Neural Network

A neural network is an interconnection of nodes or neurons that can receive, process and send signals to one another. The neurons can either be fully connected or partially connected, which can be achieved by defining the weight matrices. In this work, we focus on two types of neural networks known as feed forward neural networks (FFNN) and long short-term memory (LSTM) which is a type of RNN, which are mostly used for applications that have time series dependence. The simplest type of neural network architecture is the FFNN that allows information to move strictly from input to output. In contrast, recurrent neural networks are considered to be more sophisticated, including feedback loops and time dependence.

3.1 Feed Forward Neural Network

In FFNN, input $X_i$ with a particular weight and activation is passed to the nodes of the first hidden layer $H_i$, and the output of each node in the hidden layer acts as an input to the nodes of the subsequent hidden layer, until the output of the nodes of the last hidden layer is passed to the nodes of the output layer $Y_i$. The weighted connections, known as the weights, between nodes of different layers are denoted as $W_i$ and $B_i$ is the bias of each layer. $f(·)$ denotes the activation function (they can be same or different) that is selected. Table 1 summarizes some of the commonly used activation functions. The activation function maps the output of a particular neuron between the desired limits.

<table>
<thead>
<tr>
<th>Function</th>
<th>Equation</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sigmoid</td>
<td>$\frac{1}{1 + e^{-x}}$</td>
<td><img src="#" alt="Sigmoid" /></td>
</tr>
<tr>
<td>Hyperbolic Tangent (Tanh)</td>
<td>$\frac{2}{1 + e^{-2x}} - 1$</td>
<td><img src="#" alt="Tanh" /></td>
</tr>
<tr>
<td>Rectified Linear Unit (ReLU)</td>
<td>$\text{Max}(0,x)$</td>
<td><img src="#" alt="ReLU" /></td>
</tr>
</tbody>
</table>

Table 1: Activation Functions

The output of the FFNN presented in Figure is calculated as follows:

$$Y = f(W_0f(W_hX + B_0) + B_1)$$  \hspace{1cm} (4)

3.2 Long Short-Term Memory

In comparison, a recurrent neural network incorporates feedback connections that take into account output from previous time steps $H_{t-1}$ and/or $Y_{t-1}$. RNNs have a feedback loop at every node, which allows information to move in both directions. Recurrent neural networks can identify temporal patterns [3].

In a LSTM cell, there are three extra gates, i.e., forget, input, and output gates which decide the signals that are going to be forwarded to other nodes. $U$ is the recurrent connection between the previous hidden layer and current hidden layer. $W$ is the weight matrix that connects the inputs to the hidden layer. $C_t$ is a candidate hidden state that is computed based on the previous hidden state and current input. $C_t$ is the internal memory of the unit, which is a combination of the newly computed hidden state, multiplied by the input gate and previous memory, multiplied by the forget gate. The equations that describe the behavior of all gates in the LSTM cell are given below and LSTM cell is illustrated in the Figure 5. Operations in boxes are equivalent to classical layers (matrix multiplication and transfer function) and operations in circles refer to point-wise operations.
The equation governing the LSTM cell are given as follows:

\[ i_t = \sigma(x_t W_i + h_{t-1} U_i) \]  
\[ f_t = \sigma(x_t W_f + h_{t-1} U_f) \]  
\[ o_t = \sigma(x_t W_o + h_{t-1} U_o) \]  
\[ C_t = \tanh(x_t W_c + h_{t-1} U_c) \]  
\[ C_t = \sigma(f_t \cdot C_{t-1} + i_t \cdot C_t) \]  
\[ h_t = \tanh(C_t) \cdot o_t \]

4 Performance Metrics

4.1 Area under the Receiver Operating Characteristics Curve

The first performance metric is the area under the receiver operating characteristics curve (AUC). Definitions of different terminologies and a brief overview about AUC is given below.

4.1.1 True Positive Rate and False Positive Rate

True positive rate (TPR) and false positive rate (FPR) can be defined with the help of confusion matrix [10].

\[ TPR = \frac{TP}{TP + FN} \]  
\[ FPR = \frac{FP}{FP + TN} \]

4.1.2 Receiver Operating Characteristic

Receiver operating characteristic (ROC) is a curve between the TPR on the y-axis and the FPR on the x-axis.
4.1.3 Area under the ROC Curve

Area under the ROC curve (AUC) is a performance metric that quantifies how much the model is capable to distinguish the classes. The higher the AUC, the better the model is at predicting 1s as 1s and 0s as 0s. By analogy, the higher the AUC, the better the model is at distinguishing between TTIs with grant and without grant. If the AUC is equal to 100%, this means that there exists a threshold such that TPR = 1 and FPR = 0.

4.2 Number of FLOPs

The second metric is the number of FLOPs required to predict the content of one TTI. Table 2 shows the complexity of the basic arithmetic operations used to evaluate the overall number of FLOPs.

Table 2: FLOPs for Standard Arithmetic Operators [2]

<table>
<thead>
<tr>
<th>Arithmetic Operation</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition (+)</td>
<td>1 FLOP</td>
</tr>
<tr>
<td>Subtraction (-)</td>
<td>1 FLOP</td>
</tr>
<tr>
<td>Multiplication (×)</td>
<td>2 FLOPs</td>
</tr>
<tr>
<td>Division (/)</td>
<td>4 FLOPs</td>
</tr>
<tr>
<td>Exponential (e)</td>
<td>8 FLOPs</td>
</tr>
</tbody>
</table>

4.2.1 Feed Forward Neural Network

The FFNN used is a 2-layer network and the size of the input is 60 (6 features × 10 time step). Table 3 summarizes the layer and its dimension for each computational stage of the FFNN. H1 and H2 are the number of neurons in hidden layer 1 and hidden layer 2 for the FFNN.

Table 3: Architecture of FFNN

<table>
<thead>
<tr>
<th>Layer</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>1 × Features</td>
</tr>
<tr>
<td>Hidden Layer 1</td>
<td>Features × H1</td>
</tr>
<tr>
<td>Hidden Layer 2</td>
<td>H1 × H2</td>
</tr>
<tr>
<td>Output Layer</td>
<td>H2 × 1</td>
</tr>
</tbody>
</table>

4.2.2 Long Short-Term Memory

The LSTM used is a single-layer network with LSTM unrolled in 10 time steps with 6 features at each time step. Table 4 summarizes the layer and its dimension for each computational stage of the LSTM.

Table 4: Architecture of LSTM

<table>
<thead>
<tr>
<th>Layer</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>1 × Features</td>
</tr>
<tr>
<td>LSTM Layer</td>
<td>W matrix size is Features × Units</td>
</tr>
<tr>
<td></td>
<td>U matrix size is Units × Units</td>
</tr>
<tr>
<td></td>
<td>Layer Output 1 × Units</td>
</tr>
<tr>
<td>Dropout</td>
<td>1 × Units</td>
</tr>
<tr>
<td>Dense</td>
<td>Units × 1</td>
</tr>
</tbody>
</table>

The total number of FLOPs based on the dimensions of the matrices for LSTM and FFNN are specified in Table 5.

Table 5: Total FLOPs for FFNN and LSTM

<table>
<thead>
<tr>
<th>Operation</th>
<th>FLOPs for FFNN</th>
<th>FLOPs for LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>H1 × (10F+H2)+H2+H1+H2</td>
<td>4U²+4FU+2U</td>
</tr>
<tr>
<td>-</td>
<td>H1+H2+1</td>
<td>2U</td>
</tr>
<tr>
<td>×</td>
<td>H1 × (10F+1+H2)+2×H2+1</td>
<td>4U²+4FU+7U</td>
</tr>
<tr>
<td>/</td>
<td>H1+H2+1</td>
<td>5U</td>
</tr>
<tr>
<td>e</td>
<td>H1+H2+1</td>
<td>5U</td>
</tr>
</tbody>
</table>

5 Experimental Setup and Results

The input data is composed of 10 traces containing modem events logged during a HTTP GET request of 2 MB of data. On average, downloading 2 MB of data from the application server takes x seconds and generates therefore times series of x × 1000 time steps. The traces are recorded from several locations in Munich using the Intel® XMM™ 7480 modem for LTE-Advanced services [12]. For the algorithm, the previous 10 time steps of 6 features are observed and based on this information, the grant status for the next time step is predicted. Grants are represented using a binary time series where a 1 indicates a DL grant for the current TTI and a 0 indicates the absence of grant.
The experimental setup is divided into two flows: the global approach and divide approach. In the global approach, the model is trained on all the traces, tested on individual traces and the results averaged. Whereas in the divide approach, each model is trained just on the specific training set for trace, tested on that trace and results averaged. The two approaches are depicted in Figure 10.

At a given number of FLOPs, the global model has a better AUC than the divide model. This is because global approach is a generic predictor that is trained using the entire training set containing multiple traces recorded in different contexts, thereby it is able to extract more information and patterns from data for grant prediction. For LSTM, the results of both approaches are depicted in Figure 11.
5.2 Comparison between LSTM and FFNN Based Predictors

We observe that, at a given computational complexity, LSTM performs better than FFNN for DL grant prediction (Figure 12 and 13), this is because the LSTM can remember long time dependencies, and the grant allocation by the eNB is dependent on the previous grants and other channel conditions.

![Figure 12: Comparison between LSTM and FFNN Global Approach](image)

![Figure 13: Comparison between LSTM and FFNN Divide Approach](image)

6 Conclusion

In this work, we propose to compare a LSTM based grant predictor with a FFNN one for LTE mobile devices. To illustrate our approach, we divided our analysis into 2 parts. The global approach and the divide approach. Both NNs were trained on these approaches and the AUC and FLOPs were observed. LSTM performs better than FFNN and global approach performs better than the divide approach, at a given computational complexity. LSTM based grant predictor has on average 4.97% better AUC then FFNN for global approach and 4.47% for divide approach. If we average the 2 approaches LSTM performs 4.72% better than FFNN.

In order to enhance the significance of the results, further studies shall perform data analysis to identify other variables of interest that could be exploited to accurately predict the grants in downlink as well as in the uplink. The power consumption analysis will be carried out to assess the advantage of using this algorithm.

7 References


