Challenges

- A common challenge in healthcare today is that physicians have access to massive amounts of data on patients, but have short time to analyze all of them.

- One limitation is that hospitals without robust computational systems for processing, storing and drawing conclusions requires to outsource the clinical tasks and that is a risk for privacy clinical data.

**Developing a Green Intelligence Medical System to derivate a patient representation for predict general medical targets and improving the computational resources usage.**

Motivation: Health Care Decision-Making

Motivation: Health Care Decision-Making

Predictive Modeling

EHRs

Data Mining & Feature Extraction

General Patient Phenotype Representation

DiagnoseNET

Green Intelligence Medical System

Provides several high-level features:
1) A framework for extracting the desired features from EHRs and encoding them
2) A framework to build full learning workflow (mainly related to DeepLearning)
3) A distributed processing building learning model on Jetson TX2 Mini-Clusters/Array
4) An energy-monitoring tool for workload characterization.
DiagnoseNet theoretical workflow

EHR
- Demography
- Diagnosis
- Procedures
- Medication
- Lab Test
- History
- Signal analysis

Database federation
- Linguistic analysis
- Imaging and signals medical

Minutes of the medical meetings
- Auto-encoding (unsupervised)

Auto-encoding (unsupervised)
- Dense information

Classification / Regression / Clusterisation

Deploy Execute

Dense information

General medical target prediction

Fragment of medical databases
- Cohorts
- Additional data on a case

Case Study: Predict the Medical Future of Hospitalized Patients

Hospital Admissions
- Inpatient Diagnoses Flow

Predict Care Purpose
- ICD-10 Codes
- Billing Codes

Predict Length of Stay
- Destination
- Weeks / Days

Inpatient Home

Clinical Descriptors
- Classify Medication Procedures
- C04M Codes

Diagnosis-related Groups (ICD-10 Codes)

<table>
<thead>
<tr>
<th>Diagnosis-related Group</th>
<th>ICD-10 Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 1</td>
<td>Morbidity Principal: Cardiology</td>
<td>B410</td>
</tr>
<tr>
<td>Medical Target</td>
<td>Care Purpose</td>
<td>Z51.1</td>
</tr>
<tr>
<td>Label used</td>
<td>Clinical Major Category</td>
<td>20</td>
</tr>
<tr>
<td>Patient 2</td>
<td>Morbidity Principal: Pulmonology</td>
<td>B150</td>
</tr>
<tr>
<td>Medical Target</td>
<td>Care Purpose</td>
<td>Z51.4</td>
</tr>
<tr>
<td>Label used</td>
<td>Clinical Major Category</td>
<td>80</td>
</tr>
</tbody>
</table>

Data-mining: Feature Extraction From Electronic Health Records

Serialized each patient record in a clinical document architecture schema

<table>
<thead>
<tr>
<th>Patients</th>
<th>x1 demographics</th>
<th>x4 physical dependence</th>
<th>x7 related diagnoses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>gender</td>
<td>age</td>
<td>feeding</td>
</tr>
<tr>
<td>Patient 1</td>
<td>2</td>
<td>61</td>
<td>4</td>
</tr>
<tr>
<td>Patient 2</td>
<td>2</td>
<td>65</td>
<td>4</td>
</tr>
<tr>
<td>Patient m</td>
<td>1</td>
<td>95</td>
<td>1</td>
</tr>
</tbody>
</table>

Build a binary patient phenotype representation from their features selected

<table>
<thead>
<tr>
<th>Patients</th>
<th>x1 demographics</th>
<th>x4 physical dependence</th>
<th>x7 related diagnoses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[male]</td>
<td>[female]</td>
<td>[60-74]</td>
</tr>
<tr>
<td>Patient 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Patient 2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Patient m</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Mining Electronic Health Records
As input, we are using result features that describe the patient clinical descriptors to predict the medical targets.

PMSI-PACA Clinical Dataset

Unsupervised Patient Phenotype Representation

The task:
+ From a binary patient representation \( \{X\} \) derive a latent patient representation \( \{Z\} \).
+ Using the general representation plus a supervised learning algorithm for predict different medical targets, eventually with different methods.

Methodology: Unsupervised Patient Phenotype Representation

Unsupervised Learning Representation
Number of Gradient Updates as Factor to Early Model Convergence.

1) Number of Gradient Updates as Factor to Early Model

- Network convergence using batch partitions of [20000, 1420, 768] records to generate [4, 59, 110] gradient updates by epoch respectively.

Number of Gradient Updates as Factor to Early Model Convergence.

<table>
<thead>
<tr>
<th>Batch size</th>
<th>20000</th>
<th>1420</th>
<th>768</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epochs</td>
<td>100</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Gradient updates</td>
<td>400</td>
<td>1050</td>
<td>1100</td>
</tr>
<tr>
<td>Times</td>
<td>2176 s</td>
<td>476 s</td>
<td>266 s</td>
</tr>
<tr>
<td>Energy</td>
<td>137.65 KJ</td>
<td>41.26 KJ</td>
<td>21.87 KJ</td>
</tr>
<tr>
<td>Power</td>
<td>63.25 W</td>
<td>86.61 W</td>
<td>82.21 W</td>
</tr>
</tbody>
</table>

Power consumption in a window of 6 minutes

- BF: 20000
  - 100 Epochs
  - EC: 137.65 KJ

- BF: 1420
  - 40 Epochs
  - EC: 41.26 KJ

- BF: 768
  - 20 Epochs
  - EC: 21.87 KJ

63.35 Watts in average to process 68 gradient updates in 17 epochs.
86.61 Watts in average to process 885 gradient updates in 15 epochs.
82.21 Watts in average to process 1540 gradient updates in 14 epochs.
Model Dimensionality as Factor to Generate Quality Latent Representation

Comparison of different model dimensionality using relu as function to generate the latent representation.

Autoencoders:

End to End:

Supervised Learning
Medical Target 1: Care Purpose Description Labels

- Placement expectation, 299
- Grouping impossible, 16
- Cardiovascular / Respiratory, 18299
- Rheumatics and orthopedic, 19080
- Nutritional, 9240
- Visceral Medical Care, 13074
- Neuro-muscular, 5375
- Sensory and skin, 8273
- Amputations, 741
- Palliative care, Rehabilitation, 2261
- Mental health, 2929
- Other situations, 4489

Machine Learning Algorithm: Random Forest

Random Forest: F1-Score for Different Number of Features Scales

Random Forest: Execution Time vs Number of Features Scales
**Random Forest: F1-Score for Different Number of Features Scales**

**Experiments: Feed-forward Multilayer Perceptron**

- For the same number of neurones in the hidden layers (here 9,192 neurones):
  - When the number of layer is increasing
  - The F1 score decrease
  - And the energy consumption decrease also

<table>
<thead>
<tr>
<th>Architecture</th>
<th>F1 Score</th>
<th>Exec. time sec</th>
<th>Energy Kj</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 units on 2 layers</td>
<td>0.92</td>
<td>686</td>
<td>59.97</td>
</tr>
<tr>
<td>2048 units on 8 layers</td>
<td>0.91</td>
<td>654</td>
<td>66.49</td>
</tr>
<tr>
<td>8192 units on 2 layers</td>
<td>0.92</td>
<td>1,108</td>
<td>238.06</td>
</tr>
</tbody>
</table>

For similar F1 score, generally:
- The energy consumption is increasing
- when the number of units increase

<table>
<thead>
<tr>
<th>No units: 9,192</th>
<th>F1 Score</th>
<th>Exec. time sec</th>
<th>Energy Kj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed on 2 layers</td>
<td>0.92</td>
<td>1,108</td>
<td>238.06</td>
</tr>
<tr>
<td>4 layers</td>
<td>0.85</td>
<td>934</td>
<td>161.74</td>
</tr>
<tr>
<td>8 layers</td>
<td>0.72</td>
<td>793</td>
<td>124.04</td>
</tr>
<tr>
<td>16 layers</td>
<td>0.75</td>
<td>693</td>
<td>90.74</td>
</tr>
</tbody>
</table>
The lower value of F1, is generally for bigger number of layer.

The strategy for saving energy

• Choose the good number of neurones in hidden layer by using many layer (8 or 16)
• When the number of neurones are fixed choose the number of layer to have a good compromise between F1 score and energy consumed
**2) Preliminary Results to Scale the Unsupervised Representation Learning**

Preliminary results using 10,000 records and 11,466 features.

- Execution time for processing one epoch
- Execution time until convergence point

**3) Number of Workers and Task Granularity as Factor to Early Model Convergence**

- Early convergence comparison between different groups of workers and task granularity for distributed training with 10,000 records and 11,466 features.

- 1.30 mins in average for processing one epoch on 1 PS: 3 workers.
- 1.17 mins in average for processing one epoch on 1 PS: 6 workers.
- 566 secs in average for processing one epoch on 1 PS: 8 workers.
Preliminary Results to Scale the Feed-forward Multilayer Perceptron

- **Jetson**: 8 Gb of disk, direct transfer from disk to GPU memory
- **Network architecture**: 8-Layers Model with 256 neurons per layer
- **Task**: classification from binary representation → 6 Go
  (116,851 patients / 14,637 features)

![Graph showing exec times in minutes to reach targeted F1-score](image)

**Conclusion**

Latent representation:
- Reduces the number of sparse features without loss of precision in future classification
- Reduces training time (41%)

Use the unsupervised embedding stage to create a new lower dimensional patient representation, reduces the number of sparse features to classify at stage 3. In which, the execution time for training is minimized by 41% with regard to BPPR and the precision to classify the first medical target is almost equal.

Data partitioned on different Jetsons + small batch =
  - frequent gradient number update
  - early model convergence
  - minimizes energy consumption

DiagnoseNET: Green Intelligence System

**Process**

1. Select optimal computational resources and make good mapping of task granularity for training one model in less time and less power consumption give a mini-batch size factor.

2. Minimize the number of different trained models to converge the optimal generalization-accuracy model.

3. Management the queue of models to training and determine optimal combination of computational resources to use in each model training.

Next work:
- Distribute others kind of DL architecture (CNN or recurrent neurones) or random forest architecture
- Compare several architecture:
  - multi-GPU (share memory)
  - vs Cluster (distributed memory)
  - vs Array (hybrid memory)
- For several task
  - **MT-1**: Predict the ‘Major Clinical Category’ of patients (coarse grain CMC / fine grain GHJ) from inpatients features recorded at the admission time
  - **MT-2**: Predict the ‘Clinical Procedures’ from inpatients features recorded at the admission time and the Primary Morbidity
  - **MT-3**: Predict the ‘Inpatient Destination’ (home, transfer, death) and length of hospitalization stay from inpatients features recorded at the admission time and Primary Morbidity and Clinical Procedures
Future Work

Evaluate the DNN approaches using the different platform such as, cluster Jetson TX2, a multiGPU Node with 8 GPUs and the array Node with 24 Jetson TX2.

1. Port the framework DiagnoseNET to array Node.
2. Integrate the communication measures with the energy monitor on distributed and Hybrid platform.
3. Perform the different experiments to evaluate the case studies on the different platform.