How to predict ICU mortality with digital health data

Max Pumperla
Munich, October 11th 2017
COMPANY OVERVIEW

- **Deeplearning4j**
  Build, train, and deploy neural networks on JVM

- **ND4J**
  High performance linear algebra CPU and GPU libraries

- **DataVec**
  Data ingestion, normalization, and vectorization

**Founded**
2014

**Clients**
14 Enterprises
3,500 GH Forks, 7,200 Stars
300,000+ DL4J downloads/mo.

**Team**
~35; mostly engineers; 6 PhDs
HEALTHCARE AT SKYMIND

- Part of “Healthy China 2030” initiative
  - Expand health service industry to $2.35 trillion
  - Pilot project with 3600 hospitals in Fuzhou
  - 20 leading industry partners around CEC
  - Fatty liver disease detection
  - Bone fracture detection
  - Other use cases to come
- Intensive care unit (ICU) mortality prediction
DIGITAL HEALTH DATA ADOPTION

“The patient is data”

Randall Wetzel, Virtual PICU, Children’s Hospital LA

Figure 1: Percent of non-Federal acute care hospitals with adoption of at least a Basic EHR with notes system and possession of a certified EHR: 2008-2015.
DATA RECORDED IN EHR IN ICU

- Patient-level info (e.g., age, gender)
- Physiologic measurements (e.g., heart rate)
- Lab results (e.g., glucose)
- Clinical assessments (e.g., glasgow coma scale)
- Medications and treatments (one treatment: mechanical ventilation)
- Clinical notes
- Diagnoses
- Outcomes (one outcome: mortality)
PHYSIONET DATA

- Data from 12000 ICU stays published 2012
  - Only 4000 labeled records publicly available
  - 4000 unlabeled records used for tuning during competition (we didn’t use)
  - 4000 test examples not available
- Binary outcome: in-hospital survival or mortality (~13% mortality)
- Sequences vary in length from hours to weeks
- Observations begin at time of admission, not at onset of illness

General descriptors
- ID
- age
- gender
- weight
- height
- ICU type (cardiac, medical, surgical, trauma)

Time-series data
- blood pressure
- blood glucose
- etc.
**Task:** Predict mortality from first 48 hours of data

**Goal:** Advances toward accurate patient-specific predictive models

**Challenges:**
- Longterm dependencies
- Temporal outcome imbalances
- Treatment effects
- Missing data
- Irregularly sampled
PREPROCESSING & DATA IMPUTATION

1. Flatten data: (timestamp, variable name, value)
2. One-hot encoding of categorical variables
3. Rescaling to range [0,1] of real-values variables
4. Missing values:
   a. Carry forward imputation (suggested here)
   b. “Missing” flag to indicate imputation
5. Replicate descriptors across time
6. Do not resample to fixed timestamps

<table>
<thead>
<tr>
<th>Time</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00</td>
<td>RecordID,</td>
<td>132539</td>
</tr>
<tr>
<td>00:00</td>
<td>Age</td>
<td>54</td>
</tr>
<tr>
<td>00:00</td>
<td>Gender</td>
<td>0</td>
</tr>
<tr>
<td>00:00</td>
<td>Height</td>
<td>-1</td>
</tr>
<tr>
<td>00:00</td>
<td>ICUType</td>
<td>4</td>
</tr>
<tr>
<td>00:00</td>
<td>Weight</td>
<td>-1</td>
</tr>
<tr>
<td>00:07</td>
<td>GCS</td>
<td>15</td>
</tr>
<tr>
<td>00:07</td>
<td>HR</td>
<td>73</td>
</tr>
<tr>
<td>00:07</td>
<td>NIDiasABP</td>
<td>65</td>
</tr>
<tr>
<td>00:07</td>
<td>NIMAP</td>
<td>92.33</td>
</tr>
<tr>
<td>00:07</td>
<td>NISysABP</td>
<td>147</td>
</tr>
<tr>
<td>00:07</td>
<td>RespRate</td>
<td>19</td>
</tr>
<tr>
<td>00:07</td>
<td>Temp</td>
<td>35.1</td>
</tr>
<tr>
<td>00:07</td>
<td>Urine</td>
<td>900</td>
</tr>
<tr>
<td>00:37</td>
<td>HR</td>
<td>77</td>
</tr>
<tr>
<td>00:37</td>
<td>NIDiasABP</td>
<td>58</td>
</tr>
<tr>
<td>00:37</td>
<td>NIMAP</td>
<td>91</td>
</tr>
<tr>
<td>00:37</td>
<td>NISysABP</td>
<td>157</td>
</tr>
<tr>
<td>00:37</td>
<td>RespRate</td>
<td>19</td>
</tr>
</tbody>
</table>
RECURRENT NEURAL NETWORKS

- Recurrent neural networks (RNNs) great for modeling temporal structure in data
- Very flexible in input & output data

- Predict next word (*language modeling*)
- Translate from English to French (*machine translation*)
- Predict patient outcome from sequence (today)
- Generate beer review from category, score

http://karpathy.github.io/2015/05/21/rnn-effectiveness/
RNNS FOR TIME-SERIES DATA

Sequence of measurements in order they arrive, one step at a time
RNNS FOR TIME-SERIES DATA

Sequence of measurements in order they arrive, one step at a time
RNNS FOR TIME-SERIES DATA

Sequence of measurements in order they arrive, one step at a time
RNNS FOR TIME-SERIES DATA

Sequence of measurements in order they arrive, one step at a time

Target, e.g., in-hospital mortality
RNNS FOR TIME-SERIES DATA

Sequence of measurements in order they arrive, one step at a time

Target, e.g., in-hospital mortality

Replicate static inputs at every time step (conditional RNN)
RNNS FOR TIME-SERIES DATA

Sequence of measurements in order they arrive, one step at a time

Replicate static inputs at every time step (conditional RNN)

For irregularly sampled data, feed in elapsed time at each step

Target, e.g., in-hospital mortality
RNNS FOR TIME-SERIES DATA

Sequences of measurements in order they arrive, one step at a time

Replicate static inputs at every time step (conditional RNN)

For irregularly sampled data, feed in elapsed time at each step

Target, e.g., in-hospital mortality

Target replication
BUILDING A MODEL WITH DL4J

- Using long short-term memory networks (LSTMs, Hochreiter & Schmidhuber, 1997)
- Training done with mini-batch stochastic gradient descent

```java
MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
    .seed(RANDOM_SEED)
    .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
    .learningRate(LEARNING_RATE)
    .weightInit(WeightInit.XAVIER)
    .updater(Updater.ADM)
    .dropOut(0.25)
    .list()
    .layer(0, new GravesLSTM.Builder().nIn(NB_INPUTS).nOut(lstmLayerSize).activation(Activation.TANH).build())
    .layer(1, new RnnOutputLayer.Builder(LossFunctions.LossFunction.MCXENT).activation(Activation.SOFTMAX)
        .nIn(lstmLayerSize).nOut(nClasses).build())
    .pretrain(false).backprop(true)
    .build();
```
DISTRIBUTED TRAINING WITH SPARK

- DL4J scale-out module using Spark
- Data-parallel training procedure
- Parameter averaging on master
- Deployment on CDH cluster (Spark on YARN)

```java
TrainingMaster tm = new ParameterAveragingTrainingMaster.Builder(1)
    .averagingFrequency(3)
    .workerPrefetchNumBatches(2)
    .batchSizePerWorker(BATCH_SIZE)
    .build();

SparkDl4jMultiLayer sparkNet = new SparkDl4jMultiLayer(sc, conf, tm);
```
# EVALUATION & EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
<th>AUC*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kale, et al., AMIA 2015</td>
<td>SVM using hand-engineered features + features learned by MLP</td>
<td>0.8450</td>
</tr>
<tr>
<td>Skymind + Cloudera</td>
<td>LSTM with raw time series + missing data indicators</td>
<td>0.8520</td>
</tr>
<tr>
<td>Johnson, et al., CinC 2012</td>
<td>Bayesian ensemble with hand-engineered features</td>
<td>0.8602</td>
</tr>
</tbody>
</table>
REFERENCES

- Blog article on cloudera blog
- O’Reilly AI course on deep learning for time series
- Strata NYC 2017 Tutorial: Securely building deep learning models for digital health data
- Lipton, et al.: A Critical Review of RNNs
- Harutyunyan, et al. Multitask Learning and Benchmarking with Clinical Time Series Data.

DEEP LEARNING FOR ENTERPRISE

max@skymind.io
twitter: maxpumperla
github: maxpumperla