Pytorch ECG Heartbeat Categorisation

Project Team ==4==

Francisco Correia   Szymon Pajzert   Bijan Rad

École polytechnique

MAP583, March 2020

Outline

The problem of cardiac health monitoring

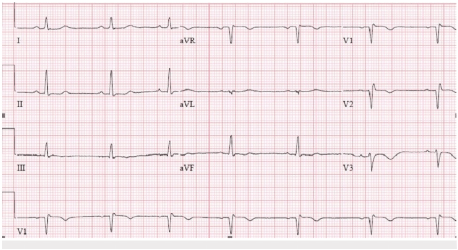
- Cardiovascular diseases (CVD) account for **1/3** of all deaths

The problem of cardiac health monitoring

- Cardiovascular diseases (CVD) account for **1/3** of all deaths
- ECG is the principal tool used by cardiologists to diagnose heart conditions

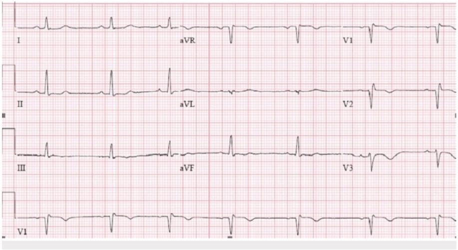
The problem of cardiac health monitoring

- Cardiovascular diseases (CVD) account for **1/3** of all deaths
- ECG is the principal tool used by cardiologists to diagnose heart conditions
- Problem: the difficulty of manually detecting and categorizing different wave-forms in the signal



The problem of cardiac health monitoring

- Cardiovascular diseases (CVD) account for **1/3** of all deaths
- ECG is the principal tool used by cardiologists to diagnose heart conditions
- Problem: the difficulty of manually detecting and categorizing different wave-forms in the signal



Aim: Develop an accurate model for diagnosis of arrhythmic heartbeats

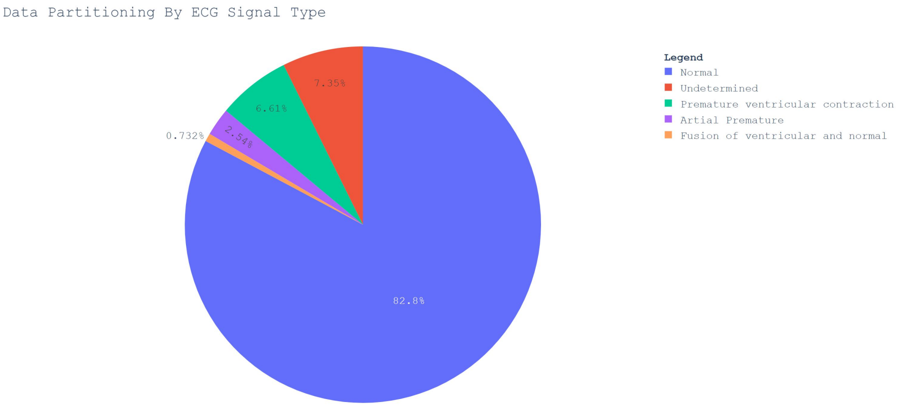
Problem formulation

1. **Create a binary classifier to distinguish normal ECG signals from anomalous ECG signals**

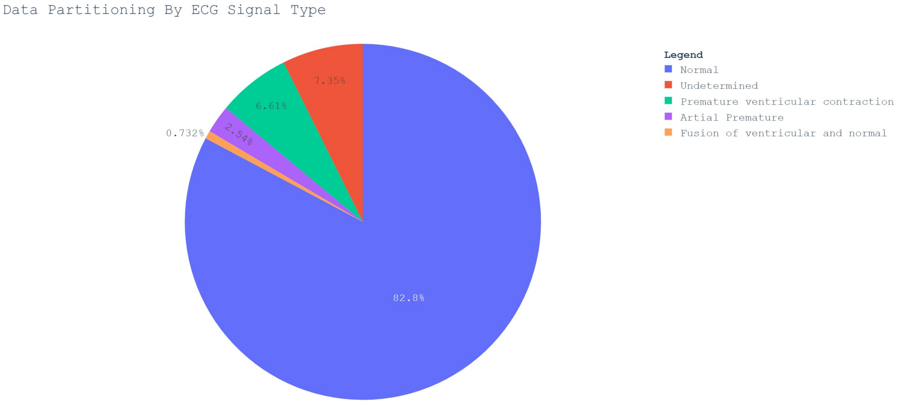
Problem formulation

1. **Create a binary classifier to distinguish normal ECG signals from anomalous ECG signals**
2. **Improve the classifier into a multi-class classifier to determine the different types of heartbeat conditions**

Data distribution

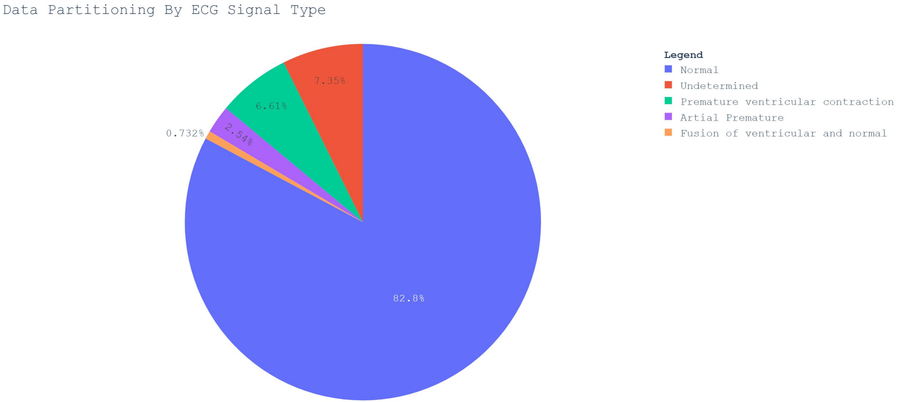


Data distribution



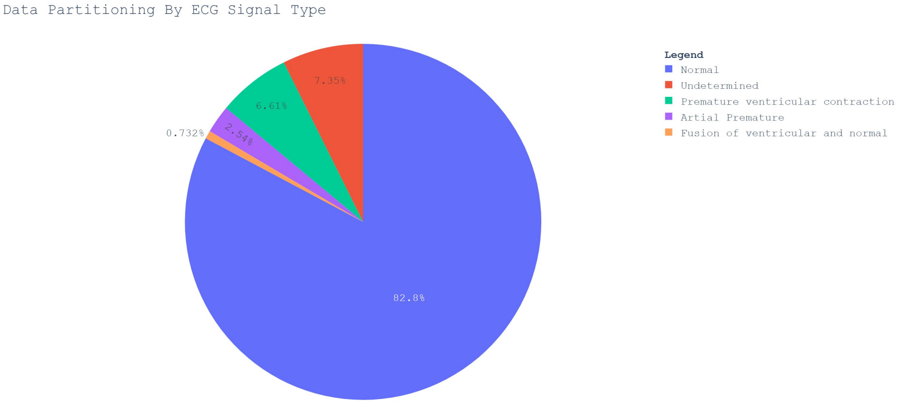
1. 5 classes

Data distribution



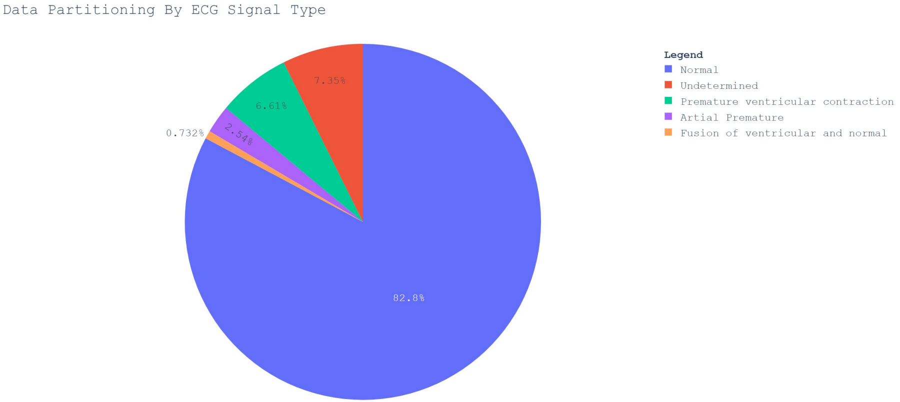
1. 5 classes
2. highly unbalanced - 82.8% normal signals

Data distribution



1. 5 classes
2. highly unbalanced - 82.8% normal signals
3. PV, PA, and Fusion signals are more severe

Data distribution



1. 5 classes
2. highly unbalanced - 82.8% normal signals
3. PV, PA, and Fusion signals are more severe
4. Undetermined is also dangerous

ECG signals

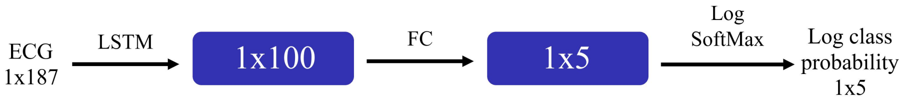
Example ECG Signal Categories
Legend
Normal
Artial Premature
Premature ventricular contraction
Fusion of ventricular and normal
Undetermined
Amplitude
Time (s)
0
0.2
0.4
0.6
0.8
1
1.2
1.4

Baseline Model

Baseline model: LSTM_1 - 1 long-short term memory layer + dense layer + log softmax

Baseline Model

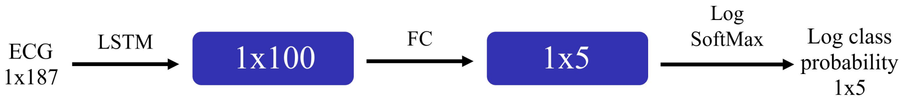
Baseline model: LSTM_1 - 1 long-short term memory layer + dense layer + log softmax



1. Input dimension = 1x187

Baseline Model

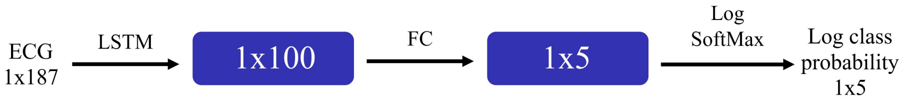
Baseline model: LSTM_1 - 1 long-short term memory layer + dense layer + log softmax



1. Input dimension = 1x187
2. Hidden layer dimension = 1x100

Baseline Model

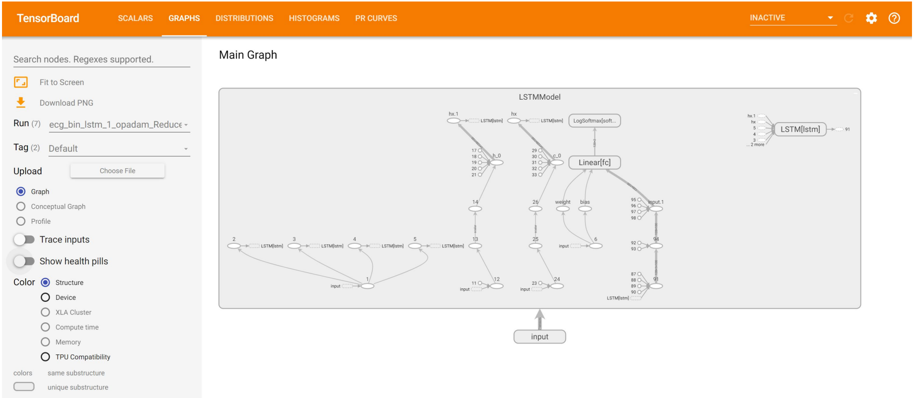
Baseline model: LSTM_1 - 1 long-short term memory layer + dense layer + log softmax



1. Input dimension = 1x187
2. Hidden layer dimension = 1x100
3. Result dimension = 1x5 (or 1x2 for binary classification)

Baseline Model

Baseline model: LSTM_1 - 1 long-short term memory layer + dense layer + log softmax



CHALLENGES

- 1 LSTM layer maybe not be sufficiently complex
- Problem of unbalanced data
- Other models may need to be aware of zero padding
- Hyper-parameter choices

CHALLENGES

- 1 LSTM layer maybe not be sufficiently complex
- Problem of unbalanced data
- Other models may need to be aware of zero padding
- Hyper-parameter choices

Up sampling/ down-sampling

Baseline Model

CHALLENGES

- 1 LSTM layer maybe not be sufficiently complex
- Problem of unbalanced data
- Other models may need to be aware of zero padding
- Hyper-parameter choices

- Up sampling/ down-sampling
- Weighted penalization for different classes

CHALLENGES

- 1 LSTM layer maybe not be sufficiently complex
- Problem of unbalanced data
- Other models may need to be aware of zero padding
- Hyper-parameter choices

- Up sampling/ down-sampling
- Weighted penalization for different classes
- Masking

CHALLENGES

- 1 LSTM layer maybe not be sufficiently complex
- Problem of unbalanced data
- Other models may need to be aware of zero padding
- Hyper-parameter choices

- Up sampling/ down-sampling
- Weighted penalization for different classes
- Masking
- Hyper-parameter tuning

Baseline Model

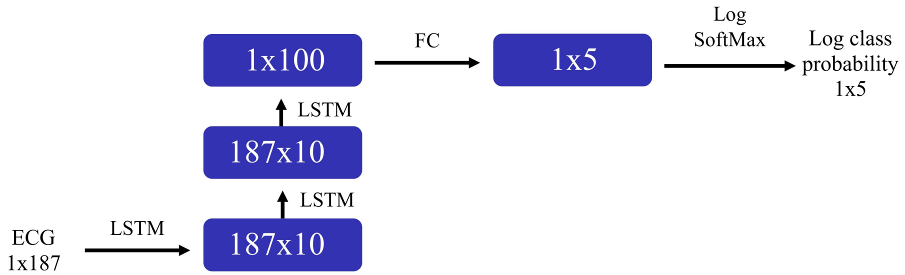
CHALLENGES

- 1 LSTM layer maybe not be sufficiently complex
- Problem of unbalanced data
- Other models may need to be aware of zero padding
- Hyper-parameter choices

- Up sampling/ down-sampling
- Weighted penalization for different classes
- Masking
- Hyper-parameter tuning
- Make more models!

LSTM_3

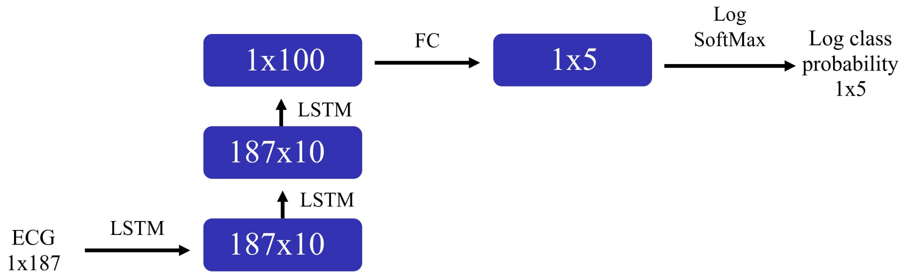
Model 2: LSTM_3 - 3 long-short term memory layer + dense layer + log softmax



1. 3 stacked LSTM layers

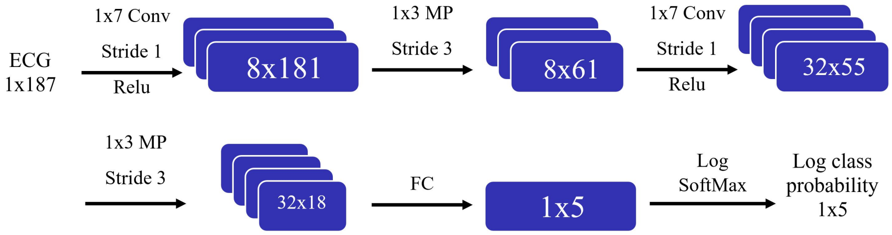
LSTM_3

Model 2: LSTM_3 - 3 long-short term memory layer + dense layer + log softmax



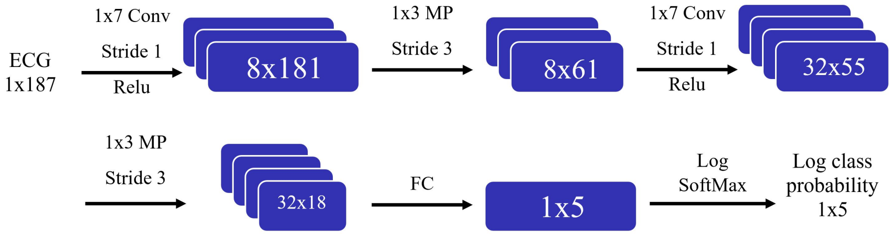
1. 3 stacked LSTM layers
2. Hidden layer dimension is tuned on Google Cloud (1x100 in image)

Model 3: 1D-CNN - 2 convolution layers + dense layer + log softmax



1D-CNN

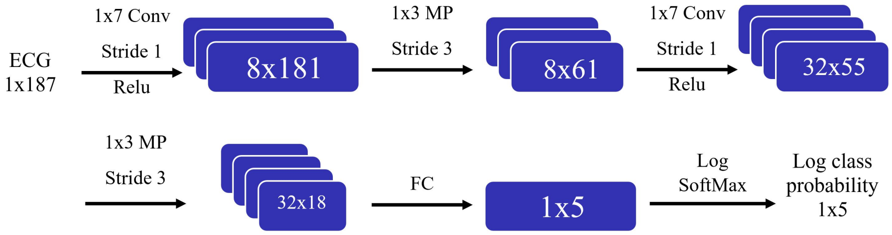
Model 3: 1D-CNN - 2 convolution layers + dense layer + log softmax



1. 2x (1D convolution layer (kernel size = 7) + Relu non-linearity + Max Pooling (3x down-sampling))

1D-CNN

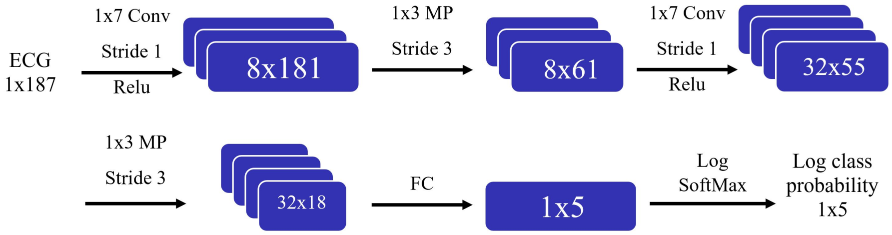
Model 3: 1D-CNN - 2 convolution layers + dense layer + log softmax



1. 2x (1D convolution layer (kernel size = 7) + Relu non-linearity + Max Pooling (3x down-sampling))
2. Masking can be activated by the flag *–masking*

1D-CNN

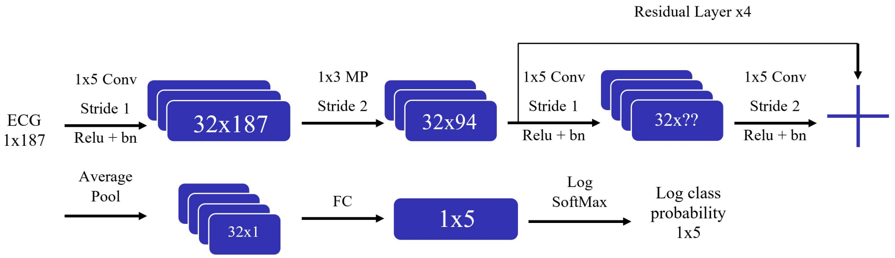
Model 3: 1D-CNN - 2 convolution layers + dense layer + log softmax



1. 2x (1D convolution layer (kernel size = 7) + Relu non-linearity + Max Pooling (3x down-sampling))
2. Masking can be activated by the flag *–masking*
3. It doesn't increase performance of the models

1D-RESNET

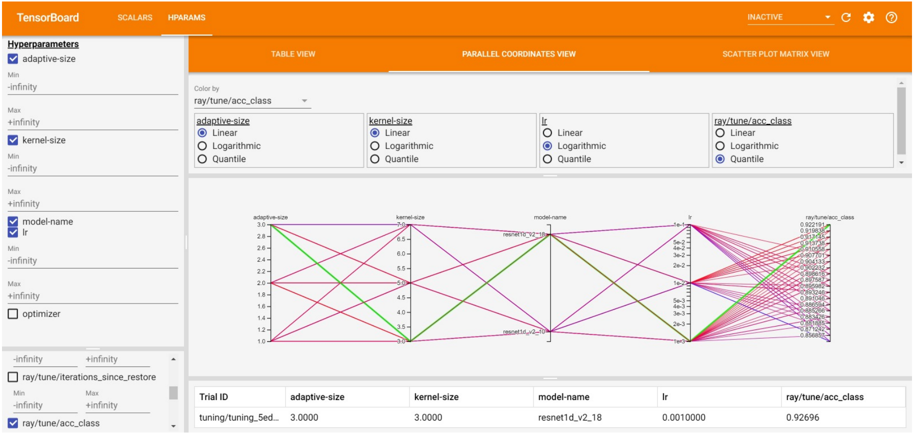
Model 4: 1D-RESNET - 1 convolution layer + 4 residual blocks + dense layer + log softmax



1. 1D convolution layer (kernel size = 5) + Relu non-linearity + Batch normalization + Max Pooling (2x down-sampling)
2. 4x (Residual block: 2x (1D convolution + Relu + Bn))
3. Average Pool + Fully connected layer

Hyperparameters tuning

Model 4: 1D-RESNET - 1 convolution layer + 4 residual blocks + dense layer + log softmax



All models

Table: Base model architectures and the principal variations that were tested[1].

Base Model	Variations
LSTM	1 LSTM layer 2 LSTM layers 3 LSTM layers
1D-CNN	2 convolutional layers
1D-RESNET	V1[2] - 2 convolutional layers per residual block V1 - 4 convolutional layers per residual block V2[3] - 2 convolutional layers per residual block V2 - 4 convolutional layers per residual block

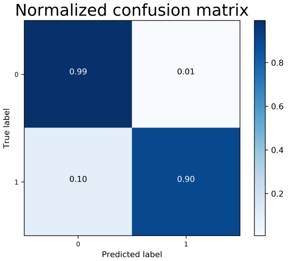
[1] Other hyper-parameters such as optimizer, learning rates, layer dimensions, weight decay and batch size are not shown but have also been experimented.

[2] Residual block has a fixed number of filters (32).

[3] Each residual block doubles the number of filters.

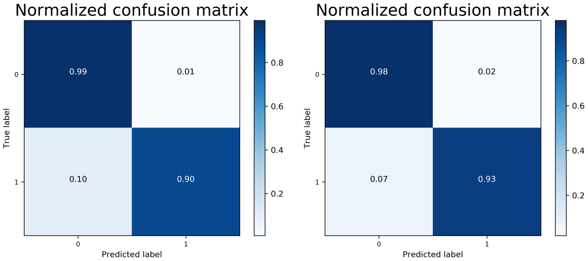
Binary classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



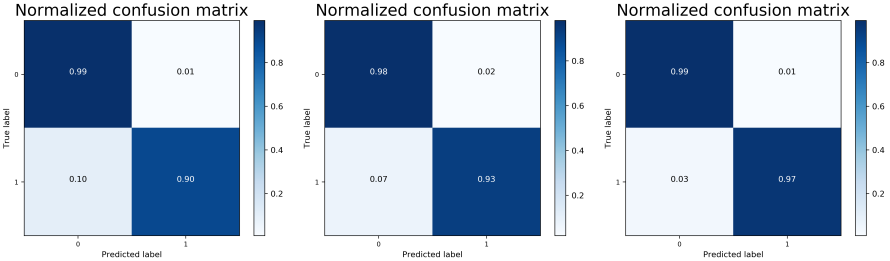
Binary classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



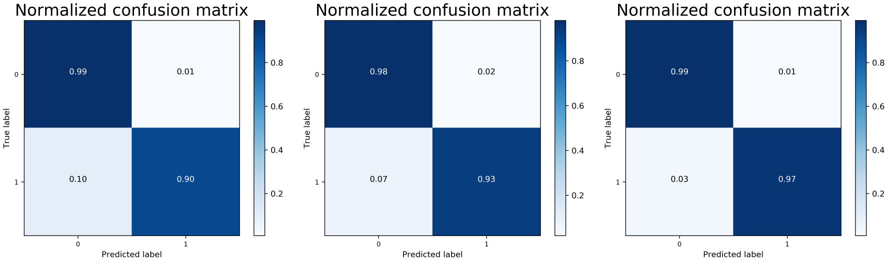
Binary classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



Binary classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



Performance is **very** good

Binary classification results

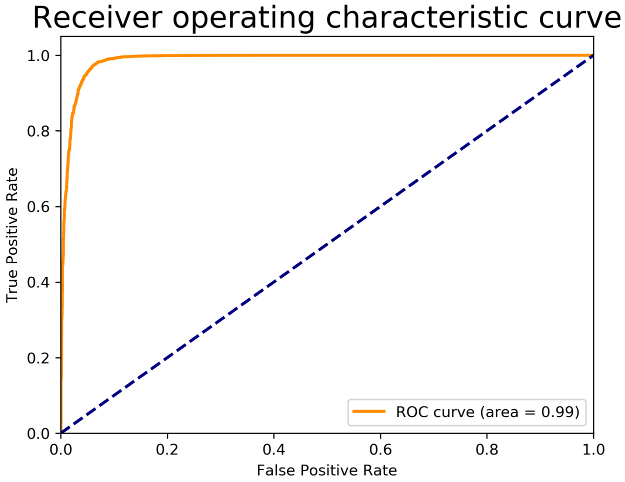


Figure: ROC curve for binary classification with 1D-RESNET_V2_10 with equal sampling

Multi-class classification results

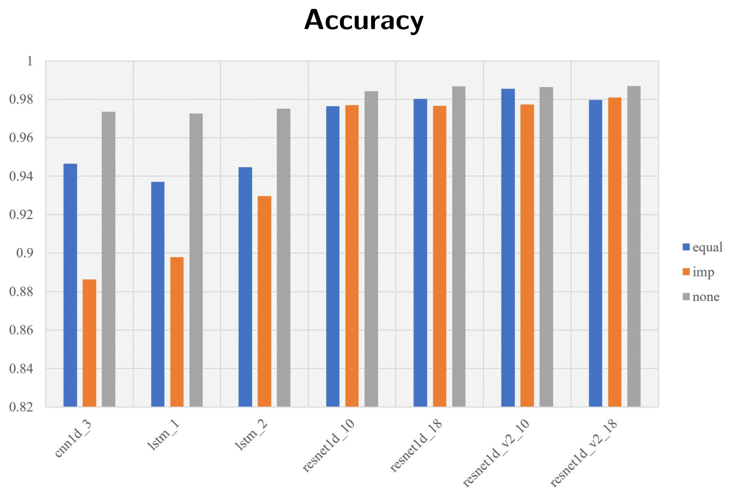


Figure: Accuracy metric on test set for multi-class classification

Multi-class classification results

Metric: Accuracy

	Sampling Type		
Model name	equal	importance	none
cnn1d_3	0.946	0.886	0.973
lstm_1	0.937	0.897	0.972
lstm_2	0.944	0.929	0.975
resnet1d_10	0.9764	0.976	0.984
resnet1d_18	0.980	0.976	0.987
resnet1d_v2_10	0.985	0.977	0.986
resnet1d_v2_18	0.980	0.981	0.987

Table: Accuracy metric on test set for multi-class classification

Multi-class classification results

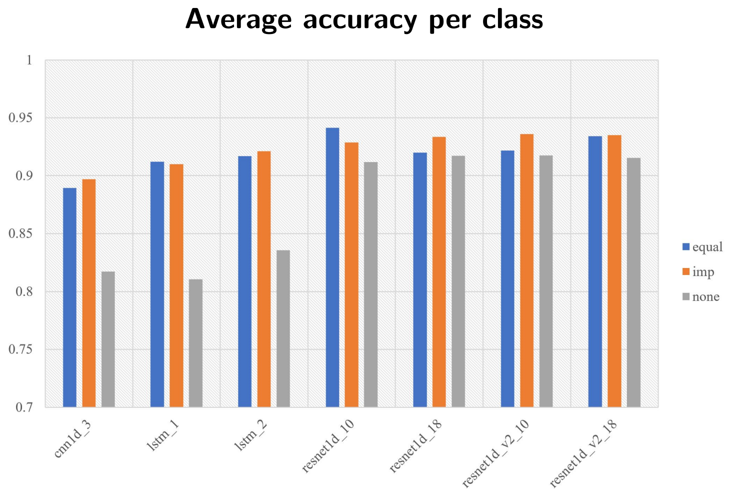


Figure: Average accuracy per class metric on test set for multi-class classification

Multi-class classification results

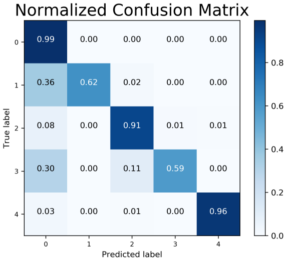
Metric: Average Accuracy per Class

	Sampling Type		
Model name	equal	importance	none
cnn1d_3	0.889	0.896	0.817
lstm_1	0.912	0.909	0.810
lstm_2	0.916	0.921	0.835
resnet1d_10	0.941	0.928	0.911
resnet1d_18	0.919	0.933	0.917
resnet1d_v2_10	0.921	0.936	0.917
resnet1d_v2_18	0.935	0.934	0.915

Table: Average accuracy per class metric on test set for multi-class classification

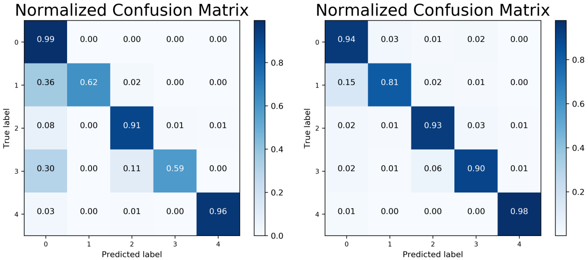
Multi-class classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



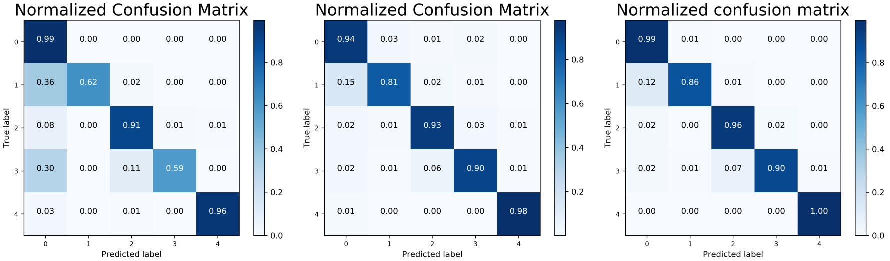
Multi-class classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



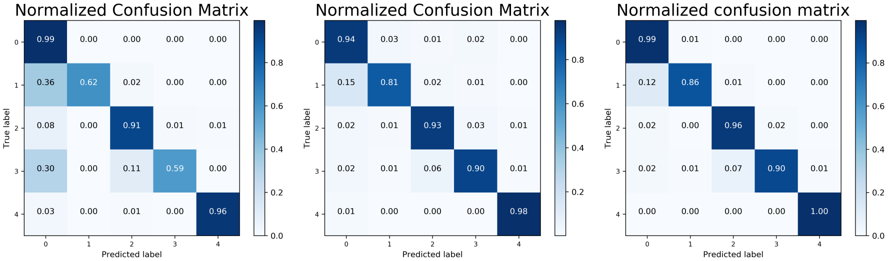
Multi-class classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



Multi-class classification results

Figure: Normalized confusion matrices for a) LSTM_1 with no sampling b) LSTM_1 with equal sampling c) 1D-RESNET_v2 with equal sampling



Performance is improving **significantly**

Classification Comparison

Work	Approach	Avg Acc (%)
Acharya et al[4] (2017)	Augmentation + CNN	93.5
Li et al[5] (2016)	DWT + random forest	**94.6**
Kachuee et al[6] (2018)	Deep residual CNN	93.4
Martis et al[7] (2013)	DWT + SVM	93.8
This work	Deep residual CNN	**94.1**

[4] Acharya, U
A deep convolutional neural network model to classify heartbeats.

[5] Li, Taiyong and Zhou, Min.
ECG Classification Using Wavelet Packet Entropy and Random Forests.

[6] Kachuee, Mohammad and Fazeli, Shayan and Sarrafzadeh, Majid.
ECG Heartbeat Classification: A Deep Transferable Representation.

[7] Martis.
Applications of Higher Order Cumulant Features for Cardiac Health Diagnosis using ECG Signals.

Resources

OUR WEBSITE:

`https://fv316.github.io/MAP583/`