

Machine Learning for Health Care: Project 1

Andrej Ivanov and Carlos Gomes

Visualization

We visualize 5000 sampled data points using the embeddings achieved by our baseline CNNs. Here we show the visualization obtained from applying PCA to extract the 2 most informative dimensions from these embeddings. We also use K-means to preform clustering, showing the results in the same representation (see figures in Appendix).

There is a nice separation of classes in the MITBIH dataset, with some overlap between “Normal” and “S” with the clustering being able to clearly identify the “Q” and “V” classes. There is some confusion in the “Normal” class, perhaps due to the overwhelming majority of the dataset that the class represents. For the PTBDB dataset, the separation is not as convincing, with a large overlap in the visualization and with a complete failure in the clustering. This indicates that the PTBDB dataset may be more challenging to classify. Further methods, including T-sne, and work on the raw data can be seen in the Visualization notebook.

Recurrent Neural Network

For the main task of the project, we developed RNN models for both datasets. The main results from our experiments are that CNNs are a better solution to this task: they are both easier to train and achieve better performance (see comparison table in next section). We found that GRU cells outperform LSTM cells in the recurrent layers, as all LSTM architectures we tested ended up suffering from vanishing gradients and not learning anything. It’s worth noting that the PTBDB task is particularly hard for an RNN because of the small number of samples. In fact, all RNN architectures we tested got stuck on predicting only the dominant class, and we only achieved decent results when we made the network bidirectional.

The best hyperparameter values and architectures were obtained through hyperparameter tuning (see notebooks in /rnn folder for details).

Comparison of models

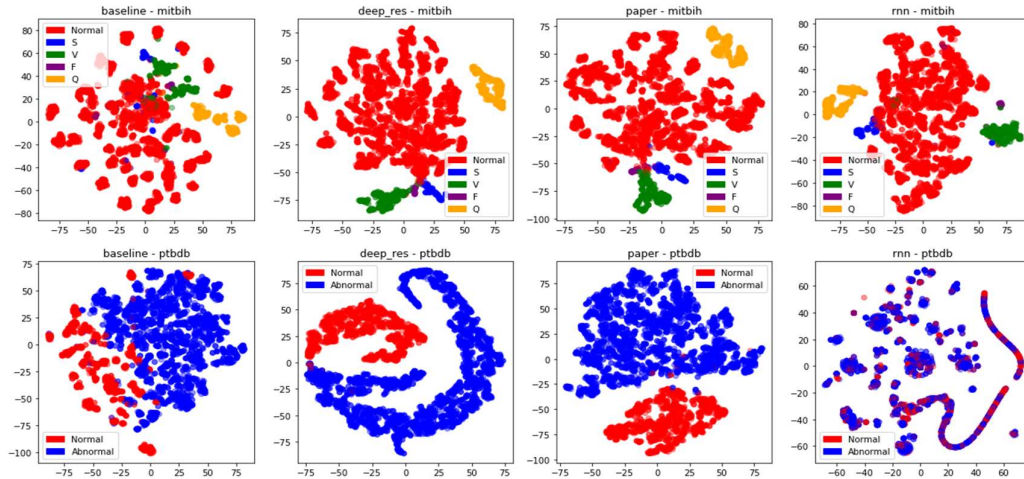
| | MITBIH | | PTBDB | | | |
|--------------------------|----------|----------|----------|----------|-------|-------|
| | Accuracy | F1-score | Accuracy | F1-score | AUROC | AUPRC |
| Baseline | 0.986 | 0.917 | 0.991 | 0.994 | 0.911 | 0.995 |
| Residual CNN | 0.987 | 0.927 | 0.993 | 0.995 | 0.993 | 0.996 |
| Deep Residual CNN | 0.987 | 0.923 | 0.995 | 0.996 | 0.993 | 0.998 |
| Embedding + SVM | 0.985 | 0.912 | 0.993 | 0.995 | 0.993 | 0.996 |
| Ensemble | 0.988 | 0.923 | 0.996 | 0.997 | 0.994 | 0.998 |
| Bi-RNN | 0.981 | 0.898 | 0.973 | 0.981 | 0.959 | 0.985 |
| RNN | 0.978 | 0.881 | 0.722 | 0.838 | 0.5 | 0.861 |
| Transfer learning | N/A | N/A | 0.862 | 0.909 | 0.785 | 0.927 |

Discussion

Overall, we implemented four distinct other methods to compare with the RNN and baselines:

1. A CNN with residual connections, as in [1].
2. A much deeper version of 1, with 50 layers compared to the original’s less than 20.
3. A bidirectional RNN.
4. A more traditional set of models (SVM and Random Forest) trained on the embeddings from the baseline models.

Figure 1: Visualization of embeddings from each of the networks. Each column shows a different model.



By far, the best performing models in our experiments are the CNNs. Our hypothesis for this is the CNNs' focus on more local patterns rather than long term dependencies. It is possible that, since an ECG reading is highly periodic, local dependencies, namely within the QRS complex of each heartbeat, play a larger role in classification than long term connections between different heartbeats. Furthermore, a CNN is able to analyze dependencies both in the past and in the future of each time step, whereas an RNN can only look at past time steps. The same advantage is shared by the Bidirectional RNN (see [/rnn/bilstm_ptbdb.ipynb](#)).

The fact that CNNs are much easier to train allowed us to make these models quite deep when compared to the RNNs, which may be another reason for their superior performance. We expected a larger performance gain from the deeper model, as this is usually the reason for the introduction of residual connections. This is reinforced by the visualization of each network's embeddings shown above. To us, the embeddings from the deeper network are very impressive. In both datasets, they show an extremely clear separation between classes, the likes of which are usually only seen in machine learning textbooks.

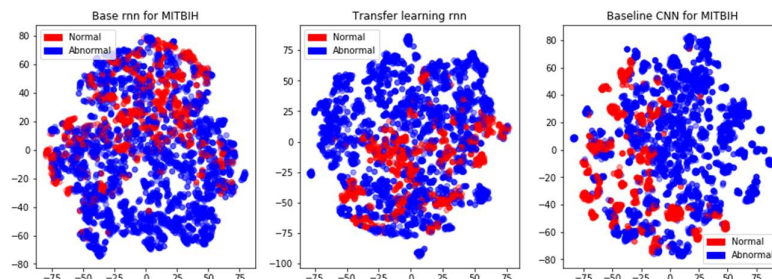
Ensembles of models

The ensemble of models was the best performing approach in our experiments (see table above). This result was expected, as it is known that theoretically ensembles lower the variance of the constituent estimators. From the models in this category, the best-performing one was the one where we simply added up the (softmax) outputs of 3 CNNs and 1 RNN and predicted the "highest output" class (see notebook [Ensemble.ipynb](#)).

Transfer Learning

The motivation for transfer learning in this setting is the lack of data: as we discussed previously, a regular RNN network with LSTM or GRU cells was not able to learn anything meaningful on the small PTBDB dataset. As both datasets consist of ECG data, it could be useful to transfer the "knowledge" from the larger dataset to the prediction task on the smaller dataset. In Figure 3, we show that even though they were trained on another prediction task, the embeddings created by a network on the MITBIH dataset are still useful for discriminating the classes in the PTBDB dataset. This is even more evident when compared to the plain RNN embeddings (Figure 3, bottom right corner), which show no separation power at all.

The evaluation results are in agreement with these visualizations: a simple RNN trained on top of MITBIH embeddings is significantly better than one trained from scratch (see [Transfer_learning.ipynb](#)).



Appendix

Top: Ground Truth; Bottom: K-means clusters(K=5)

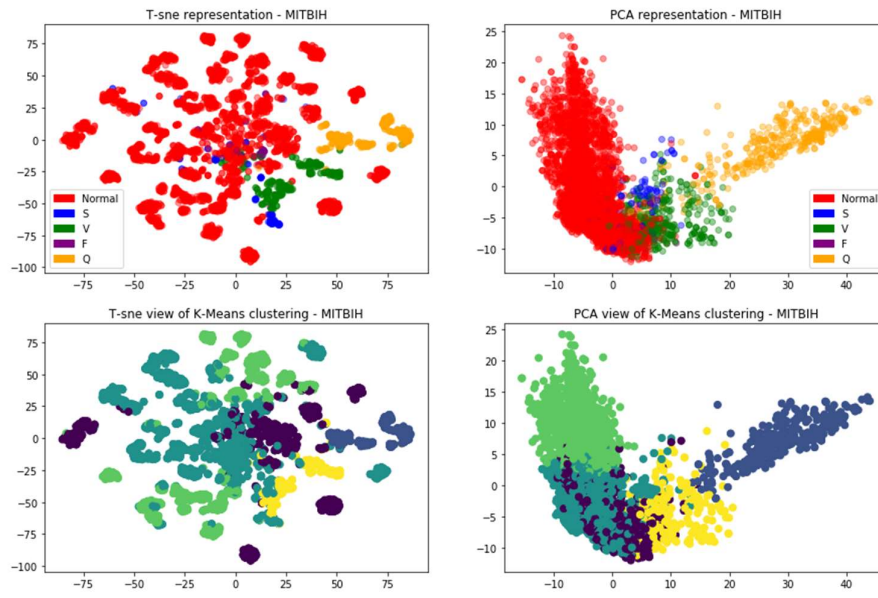


Figure 1: Visualization of embeddings on MITBIH dataset

Top: Ground Truth; Bottom: K-means clusters(K=2)

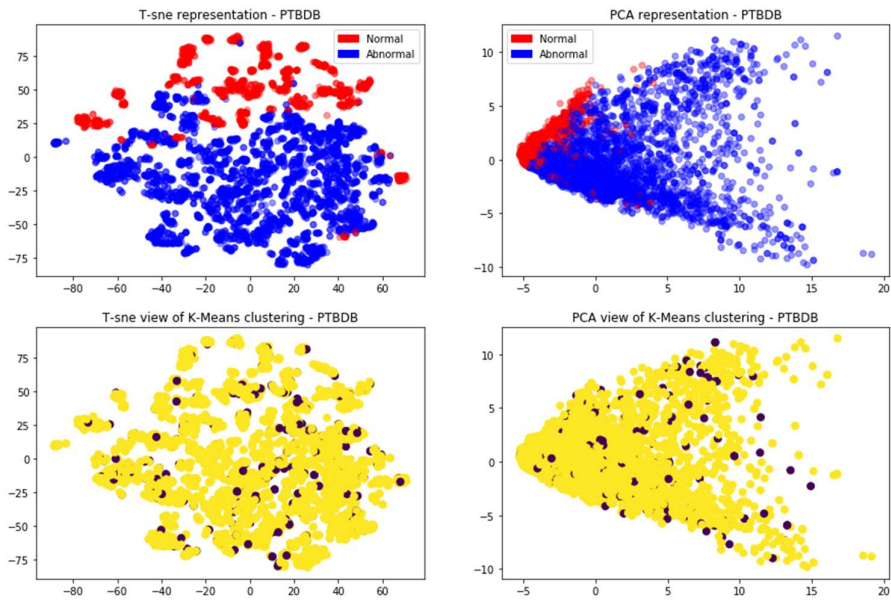


Figure 2: Visualization of embeddings on PTBDB dataset

References:

- [1]: ECG Heartbeat Classification: A Deep Transferable Representation, available at: <https://arxiv.org/abs/1805.00794>
- [2]: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, available at: <https://arxiv.org/abs/1803.01271>