

AI CUP 2023 春季賽

多模態病理嗓音分類競賽報告 (TEAM_3037)

隊伍：TEAM_3037

隊員：陳在民（隊長）、龐力元

Private leaderboard：0.600346 / Rank 9

壹、環境

abs1-py==1.4.0
alabaster==0.7.13
alumentations==1.2.1
altair==4.2.2
anyio==3.6.2
appdirs==1.4.4
argon2-cffi==21.3.0
argon2-cffi-bindings==21.2.0
array-record==0.2.0
arviz==0.15.1
astropy==5.2.2
astunparse==1.6.3
attrs==23.1.0
audioread==3.0.0
autograd==1.5
Babel==2.12.1
backcall==0.2.0
beautifulsoup4==4.11.2
bleach==6.0.0
blis==0.7.9
blosc2==2.0.0
bokeh==2.4.3
branca==0.6.0
build==0.10.0
CacheControl==0.12.11
cached-property==1.5.2
cachetools==5.3.0
catalogue==2.0.8
certifi==2022.12.7
cffi==1.15.1
chardet==4.0.0
charset-normalizer==2.0.12

chex==0.1.7
click==8.1.3
cloudpickle==2.2.1
cmake==3.25.2
cmdstanpy==1.1.0
colorcet==3.0.1
colorlover==0.3.0
community==1.0.0b1
confection==0.0.4
cons==0.4.5
contextlib2==0.6.0.post1
contourpy==1.0.7
convertdate==2.4.0
cryptography==40.0.2
cufflinks==0.17.3
cupy-cuda11x==11.0.0
cvxopt==1.3.0
cvxpy==1.3.1
cyclor==0.11.0
cymem==2.0.7
Cython==0.29.34
dask==2022.12.1
datascience==0.17.6
db-dtypes==1.1.1
dbus-python==1.2.16
debugpy==1.6.6
decorator==4.4.2
defusedxml==0.7.1
distributed==2022.12.1
dlib==19.24.1
dm-tree==0.1.8
docutils==0.16
dopamine-rl==4.0.6
duckdb==0.7.1
earthengine-api==0.1.350
easydict==1.10
ecos==2.0.12
editdistance==0.6.2
en-core-web-sm==3.5.0
entrypoints==0.4
ephem==4.1.4
et-xmlfile==1.1.0
etils==1.2.0
etuples==0.3.8

exceptiongroup==1.1.1
fastai==2.7.12
fastcore==1.5.29
fastdownload==0.0.7
fastjsonschema==2.16.3
fastprogress==1.0.3
fastrlock==0.8.1
filelock==3.12.0
firebase-admin==5.3.0
Flask==2.2.4
flatbuffers==23.3.3
flax==0.6.9
folium==0.14.0
fonttools==4.39.3
frozendict==2.3.7
fsspec==2023.4.0
future==0.18.3
gast==0.4.0
GDAL==3.3.2
gdown==4.6.6
gensim==4.3.1
geographiclib==2.0
geopy==2.3.0
gin-config==0.5.0
glob2==0.7
google==2.0.3
google-api-core==2.11.0
google-api-python-client==2.84.0
google-auth==2.17.3
google-auth-httpplib2==0.1.0
google-auth-oauthlib==1.0.0
google-cloud-bigquery==3.9.0
google-cloud-bigquery-storage==2.19.1
google-cloud-core==2.3.2
google-cloud-datastore==2.15.1
google-cloud-firestore==2.11.0
google-cloud-language==2.9.1
google-cloud-storage==2.8.0
google-cloud-translate==3.11.1
google-colab==1.0.0
google-crc32c==1.5.0
google-pasta==0.2.0
google-resumable-media==2.5.0
googleapis-common-protos==1.59.0

googledrivedownloader==0.4
graphviz==0.20.1
greenlet==2.0.2
grpcio==1.54.0
grpcio-status==1.48.2
gsread==3.4.2
gsread-dataframe==3.0.8
gym==0.25.2
gym-notices==0.0.8
h5netcdf==1.1.0
h5py==3.8.0
hijri-converter==2.3.1
holidays==0.23
holoviews==1.15.4
html5lib==1.1
httpimport==1.3.0
httplib2==0.21.0
humanize==4.6.0
hyperopt==0.2.7
idna==3.4
imageio==2.25.1
imageio-ffmpeg==0.4.8
imagesize==1.4.1
imbalanced-learn==0.10.1
imgaug==0.4.0
importlib-resources==5.12.0
imutils==0.5.4
inflect==6.0.4
iniconfig==2.0.0
intel-openmp==2023.1.0
ipykernel==5.5.6
ipython==7.34.0
ipython-genutils==0.2.0
ipython-sql==0.4.1
ipywidgets==7.7.1
itsdangerous==2.1.2
jax==0.4.8
jaxlib==0.4.7+cuda11.cudnn86
jieba==0.42.1
Jinja2==3.1.2
joblib==1.2.0
jsonpickle==3.0.1
jsonschema==4.3.3
jupyter-client==6.1.12

jupyter-console==6.1.0
jupyter_core==5.3.0
jupyter-server==1.24.0
jupyterlab-pygments==0.2.2
jupyterlab-widgets==3.0.7
kaggle==1.5.13
keras==2.12.0
kiwisolver==1.4.4
korean-lunar-calendar==0.3.1
langcodes==3.3.0
lazy_loader==0.2
libclang==16.0.0
librosa==0.10.0.post2
lightgbm==3.3.5
lit==16.0.5
llvmlite==0.39.1
locket==1.0.0
logical-unification==0.4.5
LunarCalendar==0.0.9
lxml==4.9.2
Markdown==3.4.3
markdown-it-py==2.2.0
MarkupSafe==2.1.2
matplotlib==3.7.1
matplotlib-inline==0.1.6
matplotlib-venn==0.11.9
mdurl==0.1.2
miniKanren==1.0.3
missingno==0.5.2
mistune==0.8.4
mizani==0.8.1
mkl==2019.0
ml-dtypes==0.1.0
mlxtend==0.14.0
more-itertools==9.1.0
moviepy==1.0.3
mpmath==1.3.0
msgpack==1.0.5
multipledispatch==0.6.0
multitasking==0.0.11
murmurhash==1.0.9
music21==8.1.0
natsort==8.3.1
nbclient==0.7.4

nbconvert==6.5.4
nbformat==5.8.0
nest-asyncio==1.5.6
networkx==3.1
nibabel==3.0.2
nltk==3.8.1
notebook==6.4.8
numba==0.56.4
numexpr==2.8.4
numpy==1.22.4
oauth2client==4.1.3
oauthlib==3.2.2
opencv-contrib-python==4.7.0.72
opencv-python==4.7.0.72
opencv-python-headless==4.7.0.72
openpyxl==3.0.10
opt-einsum==3.3.0
optax==0.1.5
orbax-checkpoint==0.2.1
osqp==0.6.2.post8
packaging==23.1
palettable==3.3.3
pandas==1.5.3
pandas-datareader==0.10.0
pandas-gbq==0.17.9
pandocfilters==1.5.0
panel==0.14.4
param==1.13.0
parso==0.8.3
partd==1.4.0
pathlib==1.0.1
pathy==0.10.1
patsy==0.5.3
pexpect==4.8.0
pickleshare==0.7.5
Pillow==8.4.0
pip==23.1.2
pip-tools==6.13.0
platformdirs==3.3.0
plotly==5.13.1
plotnine==0.10.1
pluggy==1.0.0
polars==0.17.3
pooch==1.6.0

portpicker==1.3.9
prefetch-generator==1.0.3
preshed==3.0.8
prettytable==0.7.2
proglog==0.1.10
progressbar2==4.2.0
prometheus-client==0.16.0
promise==2.3
prompt-toolkit==3.0.38
prophet==1.1.2
proto-plus==1.22.2
protobuf==3.20.3
psutil==5.9.5
psycpg2==2.9.6
ptyprocess==0.7.0
py-cpuinfo==9.0.0
py4j==0.10.9.7
pyarrow==9.0.0
pyasn1==0.5.0
pyasn1-modules==0.3.0
pycocotools==2.0.6
pyparser==2.21
pyct==0.5.0
pydantic==1.10.7
pydata-google-auth==1.7.0
pydot==1.4.2
pydot-ng==2.0.0
pydotplus==2.0.2
PyDrive==1.3.1
pyerfa==2.0.0.3
pygad==3.0.1
pygame==2.3.0
Pygments==2.14.0
PyGObject==3.36.0
pymc==5.1.2
PyMeeus==0.5.12
pymystem3==0.2.0
PyOpenGL==3.1.6
pyparsing==3.0.9
pyproject_hooks==1.0.0
pyrsistent==0.19.3
PySocks==1.7.1
pytensor==2.10.1
pytest==7.2.2

python-apt==0.0.0
python-dateutil==2.8.2
python-louvain==0.16
python-slugify==8.0.1
python-utils==3.5.2
pytz==2022.7.1
pytz-deprecation-shim==0.1.0.post0
pyviz-comms==2.2.1
PyWavelets==1.4.1
PyYAML==6.0
pyzmq==23.2.1
qdldl==0.1.7
qudida==0.0.4
regex==2022.10.31
requests==2.27.1
requests-oauthlib==1.3.1
requests-unixsocket==0.2.0
requirements-parser==0.5.0
rich==13.3.4
rpy2==3.5.5
rsa==4.9
scikit-image==0.19.3
scikit-learn==1.2.2
scipy==1.10.1
scs==3.2.3
seaborn==0.12.2
Send2Trash==1.8.0
setuptools==67.7.2
shapely==2.0.1
six==1.16.0
sklearn-pandas==2.2.0
smart-open==6.3.0
sniffio==1.3.0
snowballstemmer==2.2.0
sortedcontainers==2.4.0
soundfile==0.12.1
soupsieve==2.4.1
soxr==0.3.5
spacy==3.5.2
spacy-legacy==3.0.12
spacy-loggers==1.0.4
Sphinx==3.5.4
sphinxcontrib-applehelp==1.0.4
sphinxcontrib-devhelp==1.0.2

sphinxcontrib-htmlhelp==2.0.1
sphinxcontrib-jsmath==1.0.1
sphinxcontrib-qthelp==1.0.3
sphinxcontrib-serializinghtml==1.1.5
SQLAlchemy==2.0.10
sqlparse==0.4.4
srsly==2.4.6
statsmodels==0.13.5
sympy==1.11.1
tables==3.8.0
tabulate==0.8.10
tblib==1.7.0
tenacity==8.2.2
tensorboard==2.12.2
tensorboard-data-server==0.7.0
tensorboard-plugin-wit==1.8.1
tensorflow==2.12.0
tensorflow-datasets==4.9.2
tensorflow-estimator==2.12.0
tensorflow-gcs-config==2.12.0
tensorflow-hub==0.13.0
tensorflow-io-gcs-filesystem==0.32.0
tensorflow-metadata==1.13.1
tensorflow-probability==0.20.1
tensorstore==0.1.36
termcolor==2.3.0
terminado==0.17.1
text-unidecode==1.3
textblob==0.17.1
tf-slim==1.1.0
thinc==8.1.9
threadpoolctl==3.1.0
tiffio==2023.4.12
tinycss2==1.2.1
toml==0.10.2
tomli==2.0.1
toolz==0.12.0
torch==2.0.1+cu118
torchaudio==2.0.2+cu118
torchdata==0.6.1
torchsummary==1.5.1
torchtext==0.15.2
torchvision==0.15.2+cu118
tornado==6.3.1

tqdm==4.65.0
traitlets==5.7.1
triton==2.0.0
tweepy==4.13.0
typer==0.7.0
types-setuptools==67.7.0.2
typing_extensions==4.5.0
tzdata==2023.3
tzlocal==4.3
uritemplate==4.1.1
urllib3==1.26.15
vega-datasets==0.9.0
wasabi==1.1.1
wcwidth==0.2.6
webcolors==1.13
webencodings==0.5.1
websocket-client==1.5.1
Werkzeug==2.3.0
wheel==0.40.0
widgetsnbextension==3.6.4
wordcloud==1.8.2.2
wrapt==1.14.1
xarray==2022.12.0
xarray-einstats==0.5.1
xgboost==1.7.5
xlrd==2.0.1
yellowbrick==1.5
yfinance==0.2.18
zict==3.0.0
zipp==3.15.0

貳、演算方法與模型架構

We hypothesize that our previous network architecture, MetforNet, can be generalized to predict pathological voices using sound. The backbone architecture of MetforNet is based on our works [1], where we demonstrated that the backbone architecture of MetforNet can extract and predict the patterns of biomedical signals of electrocardiography (ECG), which brings us the world championship of The China Physiological Signal Challenge 2018. In our following studies, we further showed that the backbone architecture of MetforNet is strong with transferability in other applications of ECG, such as long-term monitoring for portable/wearable devices and prediction of cardiovascular mortality [2, 3].

參、創新性

Our preliminary result suggests the MetforNet could also be applied to different biomedical signals, such as DNA sequences [4], rather than just ECG signals. Therefore, we hypothesized that MetforNet could also be applied to learn the patterns of pathological voices from sound signals. The motivation of our approach is to prove that our MetforNet could be a general end-to-end solution to predict patterns from all kinds of biomedical signals.

肆、資料處理

To utilize the sound records, our previous network architecture, MetforNet, was selected as the backbone model to predict the patterns of pathological voices. It was built on a combined architecture of five CNN blocks, followed by a bidirectional gated recurrent unit (GRU), an attention layer [5, 6], and a fully connected layer. In our architecture, each CNN block contained two convolution layers that were followed by a convolution-pooling layer, which is a convolution-based pooling layer, to reduce the number of parameters and computation in the network and to control overfitting [7]. All convolution layers share the same kernel number of 12 and kernel size of three, except for the five convolution-pooling layers with kernel sizes of 24, 24, 24, 24, and 48, sequentially. Furthermore, between these CNN blocks and other independent layers, including the one between the last CNN block and the bidirectional GRU layer, 20% of their connections were randomly dropped. Batch normalization was also used to adjust and rescale the results from the attention layer, which is a special mechanism proposed to generate importance-weighting vectors [8]. LeakyReLU activation functions were used in each layer other than the fully connected layer, where a sigmoid activation function was used for predicting five classes of pathological voices [9]. This architecture was built using the Keras package supported by TensorFlow in GPUs [10, 11]. To utilize the additional demographic and clinical data, the categorical and numerical data of them were processed by one-hot encoding and Z-score normalization, respectively. The processed additional demographic and clinical data then concatenate with the output vector of the attention layer above to go through another path of dense, batch normalization, LeakyReLU, and dropout layers to produce five additive values by a dense layer with a hyperbolic tangent activation function on the five output values of the sigmoid activation functions to generate another model output.

The sound records were subjected to Z-score normalization. They were randomly divided into 60 equal parts to set up an 58-1-1 training, validation, and test scheme for machine learning.

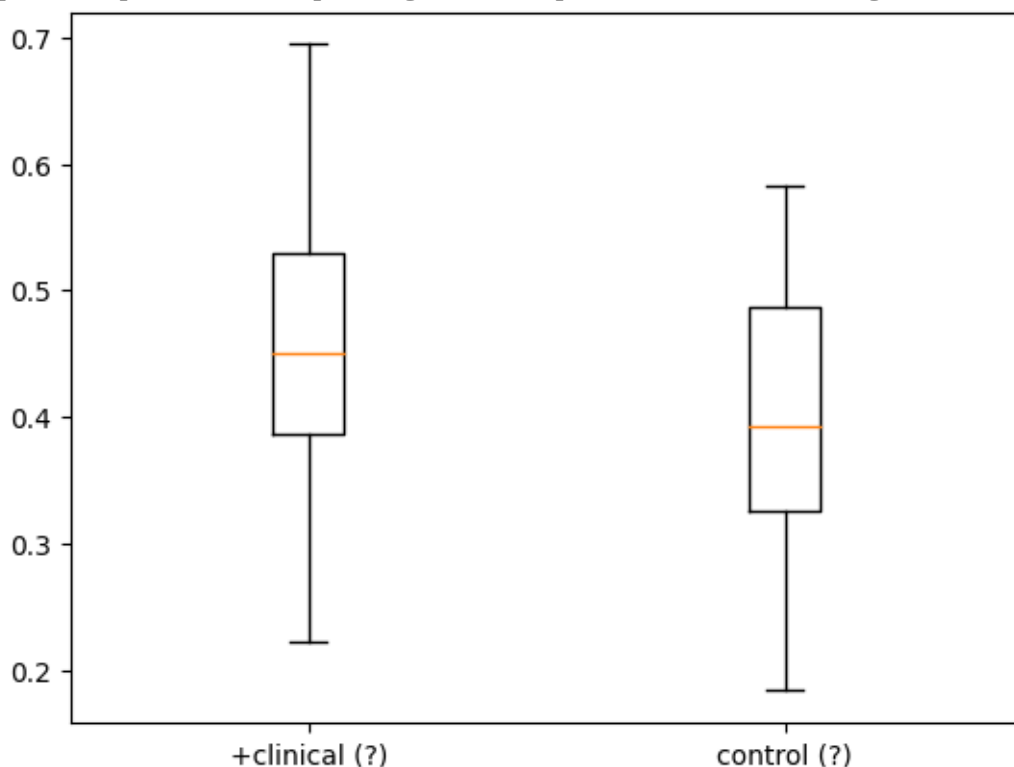
伍、訓練方式

Under such data splitting, the model was trained for 100 epochs (which refers to one complete cycle feeding training dataset) to generate 100 models (one model at each epoch), and the model with the best performance on the validation set was selected as the best model for this training process. Subsequently, the best model was used to compute the Unweighted Average Recall (UAR) score on the test set. The procedure was repeated 60 times to complete a 60-fold training; thus, the 60 best models were selected for each fold. The training process was implemented using the ADAM optimizer with a mean-square error loss func

tion [12]. Finally, the model output with additive values was selected for this task. The 60 best models were combined into one ensemble model for which the average of the output probabilities from the 60 models for each CA type was adjusted by a weight vector to produce the final probability for that pathological voice type. The vector's five weights, each for each of the five pathological voice types, were optimized by a genetic algorithm to produce the best median UAR score on the 60 test sets. In this GA optimization, a mating system consisting of 1000 genes, each of a DNA length of 5, and a population of 10 was set up, and the mating (optimization) process was followed for 10000 generations using a mating probability (DNA crossover) of 0.1 and a mutation probability also of 0.1. Given an input of a patient recording, the pathological voice type receiving the largest probability from the ensemble model would then be the type of pathological voice predicted for that patient recording. The ensemble model was our model submitted to the competition.

陸、分析與結論

The proposed end-to-end solution, MetforNet, can predict pathological voices from sound records. It suggests that a method to learn features of pathological voices from sounds is possible. Besides, the result shows that the additional demographic and clinical data could improve the performance of pathological voices prediction shown in the figure below.



The left boxplot indicates the 60-fold result with demographic and clinical data (+clinical) in comparison to the right boxplot without demographic and clinical data (control).

RecallScores +clinical:

Median Recall: 0.4506944444444444

Mean Recall: 0.45791666666666664

RecallScores control:

Median Recall: 0.39236111111111105

Mean Recall: 0.3966898148148148

柒、程式碼

Github連結：<https://github.com/chentsaimin/AI-CUP-2023-Multimodal-Pathological-Voice-Classification-Competition-Metformin-121/>

捌、使用的外部資源與參考文獻

- [1] T.-M. Chen, et al., "Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model." *Iscience* 23.3 (2020): 100886.
- [2] T.-M. Chen, et al., "SRECG: ECG Signal Super-resolution Framework for Portable/Weearable Devices in Cardiac Arrhythmias Classification." *IEEE Transactions on Consumer Electronics* (2023).
- [3] C.-M. Liu, et al., "Artificial intelligence-enabled electrocardiogram improves the diagnosis and prediction of mortality in patients with pulmonary hypertension." *JACC: Asia* 2.3_Part_1 (2022): 258-270.
- [4] T.-M. Chen, et al., "DREAM-2022-Gene-Expression-Metformin-121." GitHub repository <https://github.com/ChihHanHuang/DREAM-2022-Gene-Expression-Metformin-121> (2022).
- [5] Z. Yang, et al., "Hierarchical Attention Networks for Document Classification." *HLT-NAACL* (2016).
- [6] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks." *IEEE Transactions on Signal Processing* 45.11 (1997): 2673-2681.
- [7] J. Hearty, "Advanced Machine Learning with Python." Packt (2016).
- [8] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." presented at the Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research (2015).
- [9] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network." arXiv: 1505.00853.

[10] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." ArXiv e-prints 1603.

[11] P. W. D. Charles, "Project Title." GitHub repository <https://github.com/charlespwd/project-title> (2013).

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." ArXiv e-prints 1412.