EasyChair Preprint

№ 6062

Bird-Species Audio Identification, Ensembling 1D + 2D Signals

Gyanendra Das and Saksham Aggarwal

July 13, 2021

Bird-Species Audio Identification, Ensembling 1D + 2D Signals

Gyanendra Das[1], Saksham Aggarwal[1]

[1]*Indian Institue of Technology, Dhanbad, India*

Abstract

In this paper, a method for recognizing bird species in audio recordings is described. We have two main models: 1) A binary classifier for predicting if BirdCall is present in the audio or not; 2) A multiclass classifier for predicting which bird is present. Combining 1D and 2D signals gives strong results. We also experiment on ATDemucs which extends Demucs [1], replacing the BiLSTM with self-attention. In the waveform dimension, we first do source separation of multiple birds along with noise separation as Universal Source Separation [2]. Then we classify each source, both using a 1D waveform model (ReSE-Multi [3], but adding self-attention) and a 2D spectrogram model. We also discussed how we handle different thresholds for different models by a postprocessing technique. Ensembling techniques like Voting and Scaling describe in Section8 gave us a good boost in our results. Our combined architecture including 1D and 2D signals achieves 0.619 micro-averaged F1 in the task that asked for classification of 347 bird species.

Keywords

Deep Learning, Bird Species Classification, Transfer Learning, Attention Mechanism, Sound Detection, Audio Source Detection, Demucs, Resnet 50, Efficient Net, Ensembling, Multi Domain Meta Training

1. Introduction

There are about 10,000 different bird species in the world, and they play an important role in the natural world. They serve as good indicators of declining habitat quality and pollution. It is often easier to hear birds than it is to see them. BirdCLEF 2021 - Birdcall Identification is a Kaggle competition [4] organized by The Cornell Lab of Ornithology whose challenge is to identify which birds are calling in long recordings, given training data generated in meaningfully different contexts. This paper is structured in a way that it first gives details of the competition and the given data so that there is a clear understanding of the challenges posed by the train and test data. Also, we provide a detailed solution to the approaches we used for this challenge including data preparation, augmentations, model building, training procedure, and post-processing techniques.

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania
gyanendralucky9337@gmail.com (G. Das); sakshamaggarwal20@gmail.com (S. Aggarwal)
https://luckygyana.github.io/Portfolio/ (G. Das); https://github.com/saksham20aggarwal (S. Aggarwal)

2. Data

This section gives a brief overview of the data provided in the competition. Training on the data posed a lot of challenges since the train and test data were of different types.

2.1. Training Data

The training data is mainly comprised of two types of audio recordings:

Train short audio: The bulk of the training data consists of short recordings of individual bird calls generously uploaded by users of xenocanto.org. These files have been down-sampled to 32 kHz where applicable to match the test set audio and converted to the ogg format. Information of 397 unique species has been given. Along with audio files, metadata is also provided which consists of primary label, secondary labels, type, latitude, longitude, scientific name, common name, author, date, filename, license, rating, time, and URL.

Train Soundscapes: There is a distinct shift in acoustic domains between the training and test set. So, some examples of soundscape recordings from the test set have been provided for training and validation purposes. These 20 recordings represent 2 of the 4 test recording locations and are of length 10 minutes each. In the metadata, information has been given as to which birds are present in each of the 5 second timestamps in the training soundscapes.

All labels for train short audio had to be considered as *weak labels* since we did know which species is audible in the recording, but we did not know the exact timestamps of the vocalizations. Training with weakly labeled data was one of the core challenges of this competition. Secondary label lists the number of audible background species as annotated by the author. These lists might be incomplete and not very reliable. Also, the training data had a long-tail distribution making the dataset highly imbalanced as the head classes contained some species having train sequences more than 500 whereas some in the tail region had around mere 10 -20 sequences.

2.2. Test Data

It has approximately 80 audio recordings similar to train soundscapes. They are of 10 minutes each. We need to identify the birds present in each of the 5 second timestamps throughout the audio. These recordings are from 4 locations .

3. Our Approach

We used 4 different approaches to train our model

- Model on Spectrograms
- Model on Waveform Domain
- Multi-Domain Meta Training
- ATDemucs

4. Model on Spectrograms

In this approach, we trained the model on Mel-Spectrograms. We trained 2 types of models- model A and model B. Model A was trained to predict whether a bird is present or not in an audio clip i.e. it was a binary classification model. To train the model, we used an external data-set freefield1010 [5] along with competition data. Model B was trained to classify the birds' species. Official competition data [4] was used for this model and we tried not to input any case of nocall making use of the weak labels generated by Model A when run on the competition dataset 1.

4.1. Data Preparation

- Resample the dataset to 22050 Hz sampling rate.
- Let $time_d$ be the accepted minimum duration of anaudio sample. We choose a random $time_d$ length of chunk from audio sample
- Let min_s be the accepted minimum duration of the subimage. If the duration is less than min_s, then we convert it back to same length of min_s by padding.
- Compute three Mel-Spectrogram $M_i(x)$ with window sizes W_i (128, 512, 2048).
- Concatenate the three $M_i(x)$ into one 3 channels RGB multiscale image I

4.2. Model Building

Transfer learning from State of The Art Image-net Models to Sound Classification.

For Model type A we took 3 pretrained models i.e. Efficient B0 [6], Resnet50 [7] and Densenet [8]. We noticed that SpecAugment [9] was not giving good results, but SpecChannelShuffle increased the model performance by 0.07. We got the highest score of 0.91 by EFFB0 and by blending three models we get 0.93 F1 Score.

For Model type B we experimented with many pretrained models including Efficient B0,B1,B2,B3,B4, Resnet50,Nfnet [10] and Resnet WSL . We mention the result of this in result section 9. Here SpecAgument worked very well.

4.3. Augmentation

We executed data augmentation during the training stage.

- SpecAugment: SpecAugment is a popular augmentation technique applied on spectrogram. The spectrogram is transformed by warping it in the time direction, masking blocks of consecutive frequency channels, and masking blocks of utterances in time. We noticed that SpecAugment increased model performance without requiring any further model or training parameter tweaks.
  - TimeMasking: In time masking, t consecutive time steps [t_0, t_0 + t) are masked where t is chosen from a uniform distribution from 0 to the time mask parameter T, and t_0 is chosen from [0, t) where is the time steps.

  - FrequencyMasking: In frequency masking, frequency channels [f_0, f_0 + f) are masked where f is chosen from a uniform distribution from 0 to the frequency mask parameter F, and f_0 is chosen from (0, f) where is the number of frequency channels.

- SpecChannelShuffle: Shuffle the channels of a multichannel spectrogram (channels last). This can help combat positional bias.
- MixUp[11]: We did mixup according to primary labels that is we combined the mel-spectrograms according to a parameter alpha which had been taken from beta distribution and also took weighted average of the target label according to the same alpha. Mixup helps in reducing memorization of corrupt labels and acts as a good regularizer during training.

$$Image_i = \alpha * Image_i + (1 - \alpha) * Image_j$$
$$Target_i = \alpha * Target_i + (1 - \alpha) * Target_j$$

  Here Image represents the raw input image array and target represents the label (one-hot encodings) of the corresponding image.

4.4. Training Procedure

The training procedure used for both the models is as follows:

Model A: The model was fed with both Freefield1010 as well as Competition data and the above augmentations were applied on them. Smaller models were trained for 15 epochs while larger models were trained for 8 epochs. We used linear learning rate for the first few epochs to provide warmup and after reaching its peak i.e. *0.002*, it was linearly reduced. Adam [12] optimizer was giving the best result for this model.

Model B: The model was fed with competition data only and augmentations similar to that of Model A were applied. Smaller models were trained for 40 epochs while larger models were trained for 25 epochs. A similar strategy was used for learning rate scheduler as that of Model A. The optimizer used was Adam. While training we froze all the layers but the last few for the initial few epochs to help the model converge faster. Then all the layers were unfrozen and trained for the remaining epochs.

5. Model on Waveform Domain

In this approach, we trained the model on raw audio sample in Waveform domain. Here also we train 2 type models- model A and model B 1. Model A was trained to predict whether a bird is present or not in an audio clip i.e. it was a binary classification model. Model B was trained to classify the birds' species.

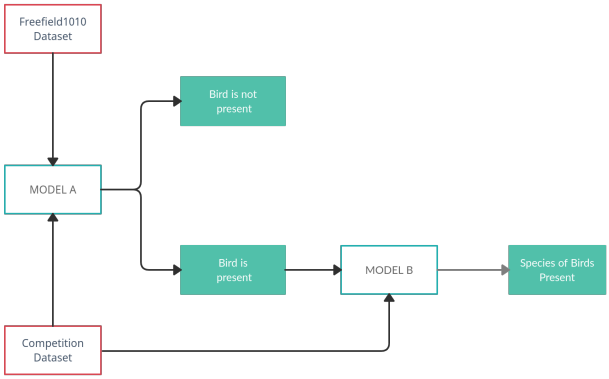

Figure 1: Pipeline Of Spectograms And WaveForm Domain Model Training.

5.1. Data Preparation

We resampled the Raw Wave to 16000 Hz sampling rate. Then we let max_l be the max length of the audio. If the length was less than max_l, we padded it with 0 at one end whereas if the length was greater than max_l, we cut the audio from both the side.

5.2. Model Building

This Model is highly motivated by *ReSE-2-Multi* [3]. With this frame-level raw waveform input, the bottom layer filters should learn all conceivable phase variations of (pseudo-)periodic waveforms that are likely to be present in audio signals. This has hampered the usage of raw waveforms as input over spectrogram-based representations in which the phase fluctuation within a frame is taken into account (i.e. time shift of periodic waveforms) is removed by taking merely the magnitude. So we added an Attention layer between two FC (Fully Connected Layer) 2. It's a simple Convolutional Long short-term memory Deep Neural Network (CLDNN) Model [13], with residual Connections which will impact high level features of Audio data.

5.3. Augmentation

- AddImpulseResponse: Convolve the audio with a random impulse response.
- TimeMask: Make a randomly chosen part of the audio silent.
- AddGaussianSNR: Add gaussian noise to the samples with random Signal to Noise Ratio (SNR) [14]
- AddGaussianNoise: Add gaussian noise to the samples
- We add pink noise at variable volumes, as well as random soundscape
- We also used a Butterworth filter with stochastic cutoffs (randomly lowpass, highpass, bandpass, bandstop).

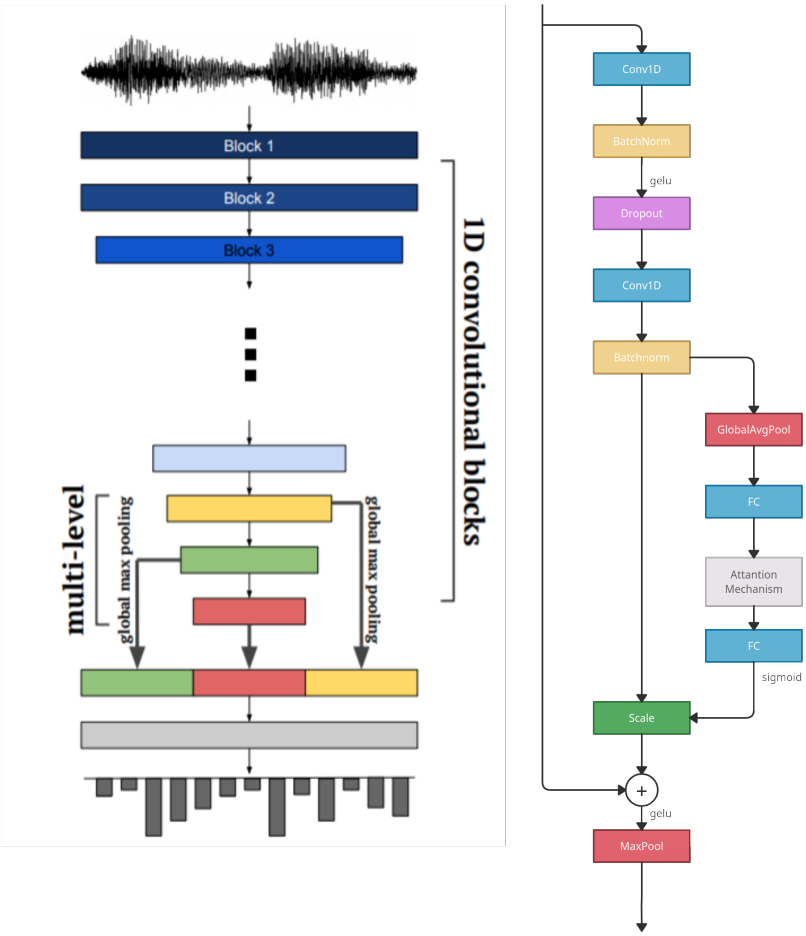

Figure 2: ReSE-2-Multi With Attention for WaveForm Domain Model Training

5.4. Training Procedure

Model A used both Freefield1010 data as well as competition data for training whereas Model B used competition data. the augmentations already stated above were applied to these raw audio samples.The rest of the training procedure is very similar to that of Wave-gram model.

6. Multi-Domain Meta Training

After training the whole dataset in Spectrograms domain and waveform domain. We check our hypotheses of combining the result from both domains so that both the models have other model's domain knowledge. For training, we froze all layers but the last 5 layers for Wave-gram Training Model M_g, and in case of Wave-Form Domain Training Model M_f, we froze all the layers except the last 3 layers. We calculated the loss using the below method which would

back-propagate through both the models.

$$O_{gi} = M_g(Spec(X_i))$$

$$O_{fi} = M_f(X_i)$$

$$Loss_i = Criterion(O_{gi}, T_i) + Criterion(O_{fi}, T_i)$$

We got the boost by 0.05 in Cross Validation Score with this technique.

7. ATDemucs

7.1. Motivation

In test set and train soundscapes, the audio file contains different types of birds. We thought about separating them and then training the classification models. We decided to introduce the music source separation concept in the multi-class classification task and experimented on it. The model is highly motivated by Demucs [1]. We provide the code in our GitHub repository A.

7.2. Data Preparation

We discovered that an audio sample in Train Sound Scapes data typically contained a maximum of 5 birds. So we took a hyper-parameter Sep_{No} to mix Sep_{No} short_audios of birds. We did another experiment of mixing the nocall data from freefield1010 and considered nocall as another bird that needs to be separated. We did the same steps for data preparation as in the Wave Form Domain data-set. We took different Sep_{No} of Short_Audio of Data and mixed it according to $\sum_{n=1}^{Sep_{No}} A_i$ For second stage training of this model, we prepared the train_soundscapes data by dividing it into chunks of data of length max_l and trained with the pseudo labels predicted by the first-stage model.

7.3. Model Building

What is the difference between Demucs and ATDemucs? In Demucs there is downsample block and then a BiLSTM layer and then upsample block. In ATDemucs Figure 3 there is attention in the LSTM layer and upsample block. In our method, we did cross attention between downsample output and upsample output.

Downsample Block: The Down Sample block is made up of a convolution with kernel size K=8, stride S=4, C_{i-1} input channels, C_i output channels, and ReLU activation, followed by a 1x1 convolution with GLU activation. We doubled the number of channels in the 1x1 convolution since the GLU outputs C/2 channels with C channels as input.
Horizontal Trans Block: We replace the Bi-LSTM Layer with Self Attention [15] Layer consisting of 8 heads and Dropout 0.2 and hidden size C_L. This block outputs $2C_L$ channels per time position. We use a 1x1 convolution with ReLU activation to take that number down to C_L.
Upsample Block: The Upsample Block is nearly symmetrical to the Downsample Block. It is made up of a convolution with kernel size 3 and stride 1, as well as input/output channels C_i and

a ReLU [16] activation. By eliminating simple concatenation like Demucs, we introduce a cross attention layer in which we take a query from the downsample block and a key and value from the upsample block. In addition, return the number of channels C_i by doing a 1x1 convolution using GLU activation. Finally, we employ a transposed convolution with K = 8 kernel size and S = 4 stride, C_{i-1} outputs, and ReLU [17] activation. Instead of using an activation function, we output $4C_0$ channels for the final layer.

$$Attention(Q_D, K_U, V_U) = Softmax(Q_D K_U^T / \sqrt{d_k}) V_U$$

Where Q_D is Corresponding Upsample layers value, K_U is Down-sample layers value and V_U is Downsample layers value.

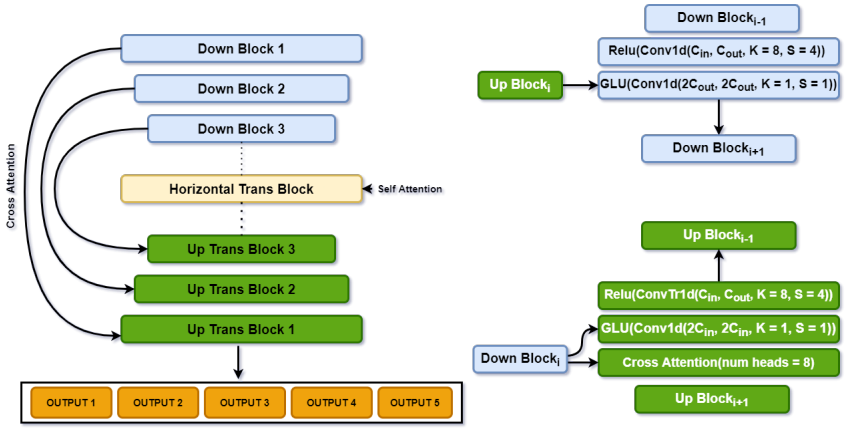

Figure 3: ATDemucs (Attention + Demucs). It consists of Three types of blocks- Downblock, HorizontalTransBlock and UpTransBlock. As the name, suggest We use Attention in HorizontalTransBlock and UpTransBlock.

7.4. Augmentation

- Shift: Randomly shift audio in time by up to ‘shift‘ samples.
- FlipChannels: Flip left-right channels.
- FlipSign: Random sign flip.
- Remix: Within a batch, shuffle the sources. Each batch is divided into groups of size *group size*, and shuffling is done separately inside each group.

7.5. Training Procedure

We trained this model in two-stage.

First Stage: First we trained the model on mixing short_audio. In this, we have training data input as a combination of 5 different birds’ short audios. We train our model to differentiate between these different recordings and separated them. We trained the model for 150 epochs with a learning rate 0.003. We used cosine annealing as the LR Scheduler which starts with

a large learning rate which is relatively rapidly decreased to a minimum value before being increased rapidly again. AdamW [18] optimizer gave good results than others.

Second Stage: In the train_soundscapes, we were given primary labels for the audio recordings at each timestamp of 5 seconds. So, after the first stage we took inference of our model on the train_soundscapes and did pseudo labeling so as to finetune the model. We trained the model for 5 epochs with a low learning rate taking AdamW as optimizer. During training, we froze some of the initial layers.

7.6. Classification After Separation

Once our model has been trained to separate the different bird sounds from the main audio recording, we run a classification model on the separated audios so as to classify which bird species it is. For this, we used Resnet50 model with pre-trained weights. We trained the model for approx 20 epochs and got a Cross Validation Score of 0.62.

8. Post Processing And Inference

We used two Post Processing Technique

Scaling Method: We noticed different models have different best thresholds. So we decide to take them into some scale then add the logits. Let Min_{Th} be the minimum threshold of all the models to be ensembled. Then we convert all the logits *0 to* Min_{Th}. Then We average all the logits and predict all the birds which have more probability than Min_{Th}.

Voting Ensemble: Let Min_C is the minimum count of bird should present in all models *N*. We predict all those birds which have $\bigcap_{i=1}^{N} Model_i > Min_C$

We submit three type of inference models

- Spectrograms Model + Waveform Model: We ensemble all the models by above scaling method, Which gave us Cross Validation Score of 0.732 and LeaderBoard Score of 0.6179.
- Multi-Domain Meta Trained Model: We optimize the best threshold for the CV and get Cross Validation Score of 0.745 and LeaderBoard Score of 0.6167 by 0.15 threshold.
- ATDemucs: We get the Cross Validation Score score of 0.623 and LeaderBoard Score of 0.59. There are many whereabouts to increase the model accuracy.

9. Results

Table 9 Shows Cross Validation Score of Spectrograms Based Model (Type Model B). After Scaling All the models, we ensemble with a threshold of 0.20 and We get 0.716 accuracy. While Direct Averaging Method giving 0.708 and Voting Classifier giving 0.699 accuracy.

Model	Best Threshold	Scaling method	Direct Averaging
EFF B3	0.1	0.6664166667	0.6687777778
EFF B2	0.25	0.6765555556	0.6789166667
NFNET	0.35	0.6669305556	0.6680416667
EFF B4	0.09	0.6618611111	0.6662777778
EFF B1	0.45	0.6633333333	0.6643055556
Resnext101 WSL	0.4	0.6762083333	0.6756527778
Resnest50 32x4D	0.35	0.686625	0.6831527778
Resnet 50	0.25	0.6908194444	0.6906805556
EFF B0	0.3	0.6675277778	0.6668333333

10. Conclusion and future work

We compose several approaches, specifically a spectrogram architecture, a raw-waveform architecture, and multi-domain meta training. In the spectrogram model as well as the raw waveform model, we used two downstream modules: one for predicting whether a bird is present or not and the other for multi-label classification of the birds. We then combined both these approaches using a loss method that back-propagates through both the models. Also, we experimented with the Demucs model and extended the model architecture by adding an attention layer in upsampling block. Ensembling methods including voting and scaling methods helped achieve better results than any individual model. Multi-domain meta training model gave us the best single model score on Cross Validation Score as well as Leaderboard Score. The spectrogram model along with scaling and downstream modules gave us the best result on the Private Leaderboard which helped us reach 67th position in the competition.

References

[1] A. Défossez, N. Usunier, L. Bottou, F. Bach, Demucs: Deep extractor for music sources with extra unlabeled data remixed, 2019. arXiv:1909.01174.

[2] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. L. Roux, J. R. Hershey, Universal sound separation, 2019. arXiv:1905.03330.

[3] J. Lee, T. Kim, J. Park, J. Nam, Raw waveform-based audio classification using sample-level cnn architectures, 2017. arXiv:1712.00866.

[4] S. Kahl, T. Denton, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of birdclef 2021: Bird call identification in soundscape recordings, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, 2021.

[5] D. Stowell, M. D. Plumbley, freefield1010 - an open dataset for research on audio field recording archives, in: Proceedings of the Audio Engineering Society 53rd Conference on Semantic Audio (AES53), Audio Engineering Society, 2014.

[6] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015. arXiv:1512.03385.

[7] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. arXiv:1905.11946.

[8] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely connected convolutional networks, 2018. arXiv:1608.06993.
[9] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, Q. V. Le, Specaugment: A simple data augmentation method for automatic speech recognition, Interspeech 2019 (2019). URL: http://dx.doi.org/10.21437/Interspeech.2019-2680. doi:10.21437/interspeech.2019-2680.
[10] A. Brock, S. De, S. L. Smith, K. Simonyan, High-performance large-scale image recognition without normalization, 2021. arXiv:2102.06171.
[11] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, 2018. arXiv:1710.09412.
[12] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.
[13] T. N. Sainath, O. Vinyals, A. Senior, H. Sak, Convolutional, long short-term memory, fully connected deep neural networks, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4580–4584. doi:10.1109/ICASSP.2015.7178838.
[14] N. Elkum, M. Shoukri, Signal-to-noise ratio (snr) as a measure of reproducibility: Design, estimation, and application, Health Services and Outcomes Research Methodology 8 (2008) 119–133. doi:10.1007/s10742-008-0030-2.
[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.
[16] Y. N. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks, 2017. arXiv:1612.08083.
[17] A. F. Agarap, Deep learning using rectified linear units (relu), 2019. arXiv:1803.08375.
[18] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. arXiv:1711.05101.

A. Online Resources

The sources for the code for this Paper available via

- GitHub,