Sound Event Detection using AudioSet strongly labelled data

Overview:

The goal of the project was to create a CRNN based system to detect temporal sound events with 1s frames. Also, our goal was to create strong labelled dataset based on Audioset dataset. We are also testing and comparing our approach with weak labelled dataset, which doesn't contain temporal information. The used audio features are from Audioset features dataset. Those features are created from 10s sound clips with VGG-inspired acoustic model and quantized to 128-dim int8 format with 1Hz. The corresponding target labels are from Audioset Temporally Strong Labels. The original Audioset dataset contains 527 different labels where we selected 10 labels to use in our implementation.

Work organization:

Dataset creation and analysis: Kimmo and Viljami

Model design and training: Charitha and Prabu

Dataset creation and analysis:

The goal of the dataset creation was to create 1s frame level presentations of the labels, with strong labeled timestamps, for the provided Audioset VGG feature records. The original records included the "weak" labels for the whole 10s video features, but not frame level separation as in strong labels. We selected 10 labels to use in our dataset. Those labels where one of the most common ones out of the +500 original labels.

The strong labeled dataset was created by iterating over the feature records, where 10x128dim features were presented for the 10s video clip. Then the corresponding 10x10dim target was created for that clip, where each row/frame contains the "one hot encoded" label info based on strong labels timestamps. Those timestamps were mapped to corresponding frame indices with floor and ceil functions. The records that didn't include any of our labels were excluded. Now the strong labeled dataset provides 10,128dim feature input and 10,10dim label target for the model.

The creation of weak labeled dataset, for reference purposes, was done with same kind of above-mentioned steps. Since the weak labels don't have the frame level presentations, basically all of the 10 frames were associated with same labels. With that association, also the weak labeled dataset provides (10,128) dim feature input and (10,10) dim label target.

Strongly labeled data										
Labels	Music	Speech	Vehicle	Car	Animal	Engine	Boat/Water Vehicle	Train	Siren	Dog
Training										
Total training samples used: 851, Total frames: 8510										
Samples	651	54	11	42	39	14	5	32	8	26
%	73.8	6.12	1.25	4.76	4.42	1.59	0.57	3.63	0.91	2.95
Frames	5442	138	109	362	137	137	50	313	80	147
%	78.7	2.00	1.58	5.23	1.98	1.98	0.72	4.53	1.16	2.13
Atleast a frame labeled -> whole sample labeled with a label: 59.64%										
Evaluation										
Total evaluation samples used: 3291, Total frames: 32910										
Samples	2475	498	129	101	179	134	20	30	42	45
%	67.8	13.6	3.53	2.76	4.90	3.67	0.55	0.82	1.15	1.23
Frames	22294	2817	1128	878	1044	1188	168	285	328	249
%	73.4	9.27	3.71	2.89	3.44	3.91	0.55	0.94	1.08	0.82
Atleast a frame labeled -> whole sample labeled with a label: 64.71%										
Total										
Total samples used: 4142, Total frames: 41420										
Samples	3126	552	140	143	218	148	25	62	50	71
%	68.9	12.2	3.09	3.15	4.81	3.26	0.55	1.37	1.10	1.57
Frames	27736	2955	1237	1240	1181	1325	218	598	408	396
%	74.4	7.92	3.32	3.32	3.17	3.55	0.58	1.60	1.09	1.06
Atleast a frame labeled -> whole sample labeled with a label: 63.73%										

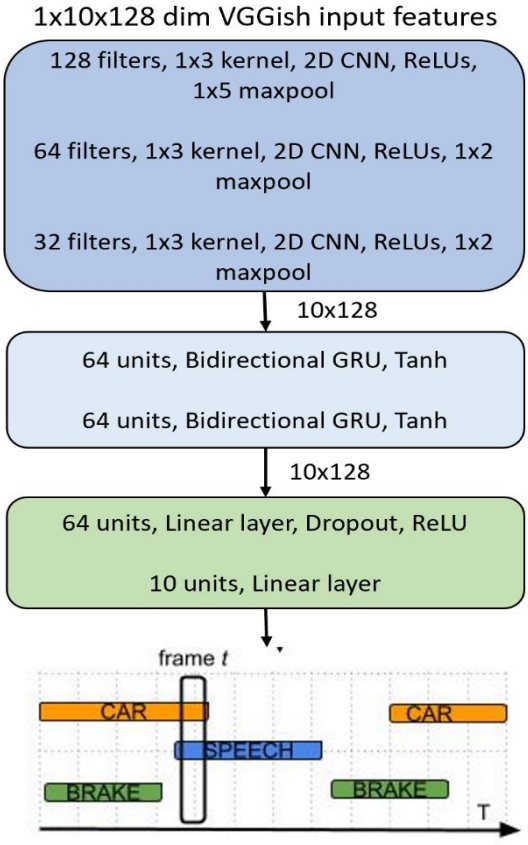
The above table has data on the strongly labeled data set used. With this data set it is evident that the label “Music” is overrepresented which might spell trouble for the model. Interesting to note that almost 2/3 of the time when a frame is labeled the whole sample is labeled with that label which in turn means that in the strongly labeled data the corresponding weakly labeled data would be the same for that label.

Weakly labeled data										
Labels	Music	Speech	Vehicle	Car	Animal	Engine	Boat/Water Vehicle	Train	Siren	Dog
Training										
Total training samples used: 4340										
Samples	2270	2023	303	113	264	82	74	58	75	104
%	42.3	37.7	5.65	2.11	4.92	1.53	1.38	1.08	1.40	1.94
Evaluation										
Total evaluation samples used: 3928										
Samples	2016	1862	279	102	259	75	91	59	88	90
%	41.0	37.8	5.67	2.07	5.26	1.52	1.85	1.20	1.79	1.83
Total										
Total samples used: 8268										
Samples	4286	3885	582	215	523	157	165	117	163	194
%	41.7	37.8	5.66	2.09	5.08	1.53	1.60	1.14	1.58	1.89

Looking at the table above it is looks like the labels “Music” and “Speech” dominate our weakly labeled data set although not as strongly as with the strongly labeled data. The other labels seem to be on a similar however lower level.

Model design:

Model design is designed with convolutional layers followed by gated recurrent unit (GRU) in recurrent neural network. Convolutional layers are used to precisely narrow the features from input features. Initially, the (1,128) dim of input features are passed through 3 convolutional layers and then passed through GRUs. And for multi-label classification, linear layer is used. The flowchart of the model is as follows:



Model training:

We used the CRNN model described in the 'Model' section for training and prediction, Where the input shape is (10,128) and the prediction output shape is 10,10. Since our dataset is multi-class and multi-label, we use sigmoid as our final activation unit which produces the individual probability(ish) for each class and the losses are calculated with Binary Cross entropy loss function which also calculates the loss on individual classes probability. We ran our training process up to 200 epochs with 30 epochs as a patience counter. The best model is then chosen to predict the classes of the test dataset. We use 0.5 as the arbitrary threshold to calculate the presence or absence of the class. i.e., if we get 0.3 probability for a class in a frame then we assume that class is not present in the given frame, on other hand if we get a probability for a class more than 0.5, we assume it is present in the given frame. Though the threshold is arbitrarily chosen, the result can be further improved with optimizing the threshold value by some mechanism.

Parameters related to training:

Loss function: binary cross entropy

Optimizer: Adam

Learning Rate:0.001

We chose to train two different models. One model is trained with strong labelled dataset, and another with weak labelled dataset. With these models we can observe the possible impact of the strong labelled dataset over weak labelled dataset.

Results:

Below we can see the table related to the prediction results of the testing datasets. There we could see that the model has classified all the testing dataset between Music and Speech while training with weak labelled dataset and between Music and Train while training with strong labelled dataset. (Refer True positive false positive in both tables).

Prediction stats for the model trained with weak dataset and tested on the weak dataset:

classes	Occurrence of Label	True positive	True negative	False positive	False negative	Precision	Recall	Accuracy	F1_sore
Music	20160	0.428691	0.453590	0.03317	0.08454684	0.92848091	0.83752485	0.8834949	0.88066059
Speech	18620	0.369425	0.449389	0.07658	0.10460794	0.82851228	0.77582122	0.81772959	0.80130148
Vehicle	2790	0	0.928971	0	0.07102851	Nan	0	0.92882653	Nan
Car	1020	0	0.974032	0	0.02596741	Nan	0	0.97397959	Nan
Animal	2590	0	0.934063	0	0.06593686	Nan	0	0.93392857	Nan
Engine	750	0	0.980906	0	0.01909369	Nan	0	0.98086735	Nan

Boat /water Vehicle	910	0	0.976833	0	0.02316701	Nan	0	0.97704082	Nan
Train	590	0	0.984980	0	0.01502037	Nan	0	0.98494898	Nan
Siren	880	0	0.977597	0	0.02240326	Nan	0	0.97755102	Nan
Dog	900	0	0.977088	0	0.02291242	Nan	0	0.97704082	Nan
Average	4921	0.079812	0.863745	0.01098	0.0454684	Nan	0.1613346	0.9435408	Nan

Prediction stats for the model trained with strong dataset and tested on the Strong dataset:

classes	Occurrence of label	True positive	True negative	False positive	False negative	precision	recall	Accuracy	F1_sore
Music	22294	0.561228	0.263537	0.05904	0.116196	0.905948	0.83023	0.82634146	0.866435
Speech	2817	0	0.914402	0	0.085597	Nan	0	0.91411585	Nan
Vehicle	1128	0	0.965724	0	0.034275	Nan	0	0.96597561	Nan
Car	878	0	0.973321	0	0.026679	Nan	0	0.97329268	Nan
Animal	1044	0	0.968277	0	0.031723	Nan	0	0.96871951	Nan
Engine	1188	0	0.963901	0	0.036099	Nan	0	0.96408537	Nan
Boat /water Vehicle	168	0	0.994895	0	0.00510	Nan	0	0.99487805	Nan
Train	285	0.002704	0.985141	0.006198	0.005956	0.290196	0.2691	0.98835366	0.2792453
Siren	328	0	0.990033	0	0.009967	Nan	0	0.99030488	Nan
Dog	249	0	0.992433	0	0.007566	Nan	0	0.99271341	Nan
average	3037.9	0.0563932	0.006523	0.035916	0.035916	Nan	0.1099	0.95788	Nan

We get around 95% overall accuracy in the strong labelled data and 94% accuracy in the weak labelled data and we also ended up classifying all the testing dataset between two classes, either it is between Music and Speech in the weak labelled data, and it is between Music and Speech in the strong labelled data. We Inferred the following from statements from the result analysis.

1. Our 95 % accuracy (94% in weak labelled) is attributed towards the skewness of the labels in both training and testing dataset where just two labels combinedly occupies around 80% of the total number of labels. So, our model learned to predict only those two labels
2. Our Accuracy difference from weak labelled dataset to strong labelled dataset is merely around 1%, we infer that this is because we have 128 embeddings for each second and we only have 10 seconds of information. So, our feature does not have sufficient information in the time domain to make use of the strong labels effectively.

Below are the tables for the "cross dataset" testing results.

Prediction stats for the model trained with strong dataset and tested on the weak dataset:

classes	Occurrence of Label	True positive	True negative	False positive	False negative	precision	recall	Accuracy	F1_sore
Music	20160	0.41174	0.40774	0.07902	0.10150	0.83898	0.80223	0.81948	0.82019
Speech	18620	0	0.52597	0	0.474033	Nan	0	0.52597	0.46426
Vehicle	2790	0	0.92897	0	0.071029	Nan	0	0.92897	Nan
Car	1020	0	0.97403	0	0.02597	Nan	0	0.97403	Nan
Animal	2590	0	0.93406	0	0.065937	Nan	0	0.93406	Nan
Engine	750	0	,0.9809	0	0.01909	Nan	0	0.98090	Nan
Boat /water Vehicle	910	0	0.97683	0	0.02317	Nan	0	0.97683	Nan
Train	590	0.00736	0.97566	0.00932	0.00766	0.44122	0.48983	0.98301	Nan
Siren	880	0	0.97760	0	0.02240	Nan	0	0.97760	Nan
Dog	900	0	0.97709	0	0.02291	Nan	0	0.97709	Nan
Average	4921	0.041909	0.86588	0.008834	0.083370	Nan	0.12921	0.90779	Nan

Prediction stats for the model trained with weak dataset and tested on the strong dataset:

classes	Occurrence of label	True positive	True negative	False positive	False negative	precision	recall	Accuracy	F1_sore
Music	22294	0.57441	0.27277	4.98025e-02	0.103008	0.920216	0.84794	0.84718	0.57441
Speech	2817	0.06952	0.67116	2.4324e-01	0.016074	0.222286	0.81221	0.740686	0.06952
Vehicle	1128	0	0.96569	3.0386e-05	0.034275	Nan	0	0.965694	Nan
Car	878	0	0.973321	0	0.026679	Nan	0	0.97332	Nan
Animal	1044	0	0.968277	0	0.031723	Nan	0	0.968277	Nan
Engine	1188	0	0.963901	0	0.036099	Nan	0	0.96390	Nan
Boat /water Vehicle	168	0	0.994895	0	0.00510	Nan	0	0.99489	Nan
Train	285	0	0.994895	0	0.008659	Nan	0	0.99134	Nan
Siren	328	0	0.990033	0	0.009966	Nan	0	0.99003	Nan
Dog	249	0	0.992433	0	0.007566	Nan	0	0.992433	Nan
average	3037.9	0.06439	0.87838	0.029307	0.02791	Nan	0.1660	0.94277	Nan

The model that is trained on strong data and tested on weak data, has less accuracy (0.90779) than the model that is trained on weak data and tested on strong data (0.94277). This might be from the fact that weak labels don't have actual frame level "10,10" output that would be based on label's presence. So, even though strong model might predict "correctly" presence between frames i.e., 3-8, the weak labeled data punishes from that. On the other hand, strong data has a lot of samples that are present in all frames, so the prediction from weak model isn't that much "wrong" in strong data.

The model that is trained with strong data and tested on strong data, has slightly higher accuracy (0.95788) than the model that is trained on weak data and tested on strong data (0.94277). However, the difference is very small. It basically proves the above-mentioned fact that, if label is present on the strong dataset, it is present on most of the frames.

Conclusions:

From these results we could conclude that skewness in the training and testing data does not affect accuracy, it heavily affects the classification of the other classes. We can also conclude that if we do not have sufficient information in the time domain, the strong labelled data cannot produce better results than the weak labelled data.

Before model training, we should have been giving more thought on selecting evenly distributed labels, or at least to process the dataset to be more balanced. Now the models are basically making predictions between two labels. Unfortunately, we realized these facts a bit too late, and should do all the training and analysis again to fix them. But the most important part is that we learned a lot during the whole process. And despite the outcome isn't "perfect", we are still satisfied of the skeleton and analysis that we created for this solution.