

宇宙人コンペ振り返り

大阪大学医学部3年 安部政俊

data

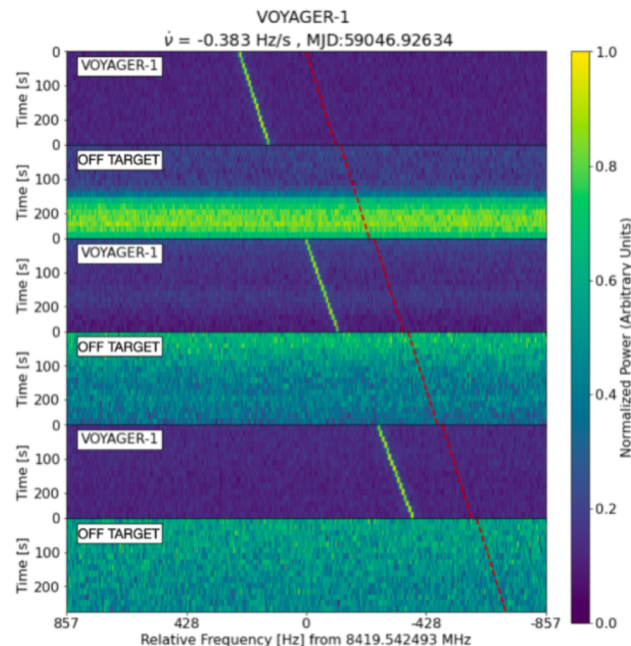
望遠鏡から得た信号に地球外からの信号を人工的に混ぜたデータ

5秒ごとに星A,B,A,C,A,Dに望遠鏡を向ける

Aに宇宙人がいれば5秒ごとに信号が

見えるはず(黄色：ボイジャー1号)

地球のものはすべての時間で観測(赤線)



task

信号データはFFTして正規化したものがnpzで与えられた。

































これを地球外からの信号があるか否かの2値分類する。

評価指標はAUC

提出形式はcsvの提出

...値の順番だけを見ている評価指標！！

自分のチームの取り組み

<div><div></div> In the money <div></div> Gold <div></div> Silver <div></div> Bronze</div>							
#	△...	Team Name	Notebook	Team Membe...	Score ?	Entries	Last
1	—	Watercooled		  	0.96782	93	8d
2	—	未知との遭遇			0.81206	85	9d
3	—	knj			0.80475	77	9d
4	▲ 2	Steven Signal		  	0.80428	92	8d
5	▲ 2	SETIの壁		  	0.80171	168	8d
6	▲ 4	James Howard			0.80072	13	8d
7	▼ 3	The Unforgiven	    	0.80036	108	9d	
8	▼ 3	Ilya Makarov			0.79945	123	8d
9	—	MPWARE Giba ...	    	0.79929	167	8d	
10	▲ 3	Aliens among us		   	0.79809	213	8d
11	▼ 3	SETIes	   	0.79806	126	8d	
12	—	A Speck in the Cosmos			0.79698	70	8d

序盤

信号のコンペなので過去の音系のコンペを確認した

評価指標がAUCなので過去のAUCを使ったコンペを確認した

公開notebookをnfnet→efnetB0_nsに変えてbaseline作成

<work>

mixup(>cutmix), GeM pooling, cutout, specaug, remove resize, large model

<not work>

focal loss($\alpha=3$), label smoothing($\alpha=0.05$), under sample, Bright contrast

specaug/mixup/GeM

specaugは周波数方向/時間方向にマスクをかける

mixupは決定領域をなめらかに&label noiseにも有効

GeMはGlobal avg/max poolingの一般化

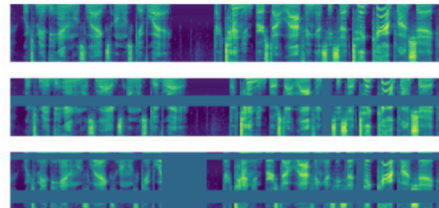


Figure 2: Augmentation policies applied to the base input. From top to bottom, the figures depict the log mel spectrogram of the base input with policies None, LB and LD applied.

GeM: avepoolとmaxpoolを一般化し、パラメーター p によって最適化できるようにしたもの、です。
 p はnn.Parameterとすることで初期から最適なものに学習していきます。

$p=1$ でmean, $p=\infty$ でmaxと等しい。論文では $p=3$ を推奨。
100%よくなるかはわからないが、頻繁に使われているようです。

```
F.avg_pool2d(x.clamp(min=eps).pow(p),  
(x.size(-2), x.size(-1))).pow(1./p)
```

Mixup is a data augmentation technique that generates a weighted combinations of random image pairs from the training data. Given two images and their ground truth labels: (x_i, y_i) , (x_j, y_j) , a synthetic training example (\hat{x}, \hat{y}) is generated as:

$$\hat{x} = \lambda x_i + (1 - \lambda) x_j$$

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j$$

where $\lambda \sim \text{Beta}(\alpha = 0.2)$ is independently sampled for each augmented example.

中盤(マージ後)

チームメイトのアイデア

1 : CNNの最初のcnv2dのstrideを2→1に

2 : ABACADと並べるのではなく、[AAA][BCD]とstackして2chに
discussionより

NMFしたらノイズ取れる/sizeが大きいほうがよい

<work>

2ch, stride1, large size(768*768), add NMF feature as 3rd ch

<not work> いっぱい

CNNのstrideの変更..alaskaコンペより

alaska 1stの解法から引用

”

弱い信号を捉えるために、モデルがより高い解像度に長く留まることが重要です。

オリジナルのSe-ResNeXtまたはDenseNetは、1/4解像度（最初に2つの連続したダウンサンプリング）から「深刻な」モデリングを開始するだけなので、収束が遅くなります。ステガナリシスでは、高解像度をモデル化することが重要です。Efficientnetsは1/2解像度でモデリングを開始するため、高速に収束できます。最初の2つの「ダウンサンプリング」(stride2とpooling)を削除したため、seresnet18も高速に収束しました。

ストライドとプーリングを削除すると、CNNは $4 \times 4 = 16$ 倍複雑になります。計算の複雑さは、pytorchのefficiencynetb5と同様です。

”

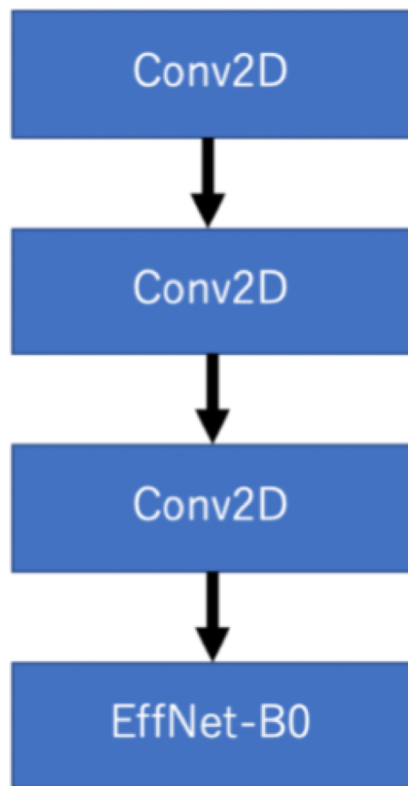
更に高解像度を維持するアイデア

backboneの前にconv2dをいくつか入れる

```
class CustomTimmModel(nn.Module):
    def __init__(self, backbone, out_dim=1, pool_type=None, pretrained=True):
        super().__init__()
        self.myconv0 = ConvBnRelu(1, 12, (27,1), (12,1), (0,0))
        self.myconv1 = ConvBnRelu(12, 18, 3, 1, 1)
        self.myconv2 = ConvBnRelu(18, 24, 3, 1, 1)
        self.myconv3 = ConvBnRelu(24, 32, 3, 1, 1)
        self.model = timm.create_model(backbone, pretrained=pretrained, in_chans=32)
        if 'efficientnet' in backbone or 'densenet' in backbone:
            in_ch = self.model.classifier.in_features
            self.model.conv_stem.stride = (1, 1)
            self.model.classifier = nn.Identity()

        self.dropout = nn.Dropout(0.5)
        self.myfc = nn.Linear(in_ch, out_dim)

    def forward(self, x):
        x = self.myconv0(x)
        x = self.myconv1(x)
        x = self.myconv2(x)
        x = self.myconv3(x)
        x = self.model(x)
        x = torch.stack([self.dropout(x) for _ in range(5)], 0).mean(0)
        x = self.myfc(x)
        return x
```



終盤

データリークが発見&修正された。

新しいtest,train,リークがある古いtrain+testが配布された。

CV/LBのgapが0.1ほどになって開いた。

discussionよりコンペの課題が**domain shift** と明らかになった。

..test dataにはtrainにはない信号のパターンがある/testとtrainは見た目が異なり、
testのほうがノイズが大きい

終盤

<work>

shift,sharpen,add old data(only positive),pretrained with old data

<not work>

arcface,DANN,gausse noise, resizeの方法を変える,疑似ラベル(soft/hard)

なぜarcfaceなのか

距離学習はCrossentropyでの学習と比べて未知クラスに頑強

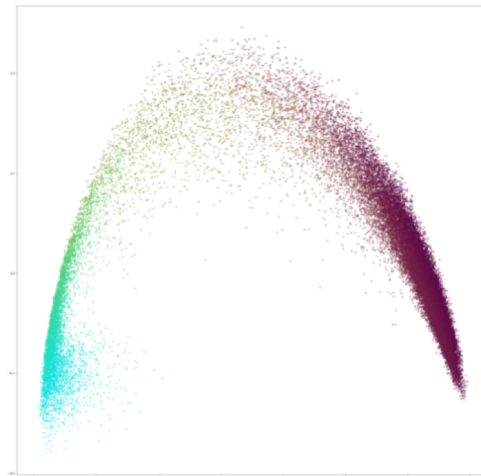
また、未知クラスは決定境界付近にあるはず！

→arcfaceで学習したモデルから得た埋め込みを可視化して分析すると未知クラスがわかるのでは？...スコア向上なし...

右図：testの埋め込み(PCAで圧縮)

明るい色ほどpredictが大きい

半円上に埋め込みが配置されている



距離学習後の埋め込み分析の注意

sheep氏より

TSNE is a visualization method and seems no one use its decomposition output to do further analysis, and It usually used in cell biology and medical.

非線形なTSNEではぱっと見の別れ方は良いが距離の情報が失われる のだそう

最終的な自分の提出

model:b3(pretrained with old data)

size:768*768*3

augmentation:h/vflip,specaug,cutout mixup($\alpha=1$)

epochs 20(after 15 epoch stop mixup),lr:adam ,CosineAnnealing($1e-3 \sim 1e-5$)

と

ssrとsharpenを加えたもの

と

B5(add old positive data)にしたもの を算術平均した

気になった解法

1st

① 同じCVでもLB違った..subを分析すると未知クラスを発見！

② target以外の背景がtrain/testで異なる。

psuedo label (add target~1すると、target=1がtest由来であると予測し始めた

→背景に過剰適合しているとわかった

→ノイズを除去してCV/LBgap埋めた.

③ ノルムのないモデルがわずかに優れたCV / LB相関を示すことを発見しました。これは、おそらくデータ分布の違いによって影響を受ける可能性のあるバッチノルムがないためです。

2nd

① mixupを論理ORにした

(チームメイトもやっていたらしい)

② 疑似ラベル使用時にnoisy studentを使用した

...疑似ラベルへの過剰適合を防ぐ

- The mixup target can be mixed by using the following expression to express a logical OR, which also supports soft targets when using pseudo labels.

```
y = y + y[index] - (y *  
y[index])
```

mixupの工夫

`lam=np.beta(alpha,alpha)`

`mix_y = y*lam+y[index]*(1-lam)` が普通のもの

工夫①`lam=np.random.uniform(clips[0], clips[1])`と一様分布を用いる

...ベンガルにて登場

工夫②`mix_y = y + y[index] - (y * y[index])` とする(論理OR)

...信号で有効

3rd/ 5th/8th

3rd

- ① To accelerate the training, replace Swish to ReLU
- ② Triplet Attention

5th

SHOT(<https://github.com/tim-learn/SHOT>)

8th

疑似ラベル(soft)でCV下がってもLBあがるかもだから提出！

反省

- 1 : 音コンペの解法を読み込めばmixup論理ORは序盤から試せたはず
- 2 : test見てたら未知クラスは目視で見つけることができたはず
- 3 : gause noiseは信念を持ってパラメータの探索をすべきだった
- 4 : 中盤にgradcamを見ていたが終盤も確認してfeedbackを得るべきだった
- 5 : 疑似ラベルを使いこなせなかった

感想

hydraは神

序盤金圏はモチベに非常に良い

twitterなどで界隈の人との知り合いが増えた

生活リズムが崩壊した

GMと交流できるのは現行コンペだけ！？

計算環境は非常に大事！！

初メダル嬉しいです

計算環境など

このコンペでは医学科学生用計算機(2080ti*2)を使用しました。
このコンペでは大阪大学データビリティフロンティア機構の計算環境を使用しました。(V100(16GB)*2,V100(32GB)*1,Quadro RTX 8000*3)

実験管理はhydraにconfig持たせて基本1実験 1 スクリプト
idea.mdにメモを取ってスプシにCV/LBと実験の差分を書いてました

参考

mixup: <https://paperswithcode.com/method/mixup>

<https://arxiv.org/pdf/1912.02911.pdf>

<https://www.kaggle.com/c/bengaliai-cv19/discussion/136025>

specaug: <https://arxiv.org/pdf/1904.08779.pdf>

GeM: <https://amaarora.github.io/2020/08/30/gempool.html>

stride 1: <https://www.kaggle.com/c/alaska2-image-steganalysis/discussion/168542>

<https://www.kaggle.com/c/alaska2-image-steganalysis/discussion/168548>

discussion : <https://www.kaggle.com/c/seti-breakthrough-listen/discussion>