# RECORD

## 技巧对比

Smooth f2-score：效果不佳，本质上来说smooth f2-score优化的是基于sample的f2-score，但是本次评估是macro f2-score

Original vs Segmented：Original 普遍由于Segmented，但是速度稍微慢

对小标签进行上采样（其他标签会连带采样）：对0，3，6，12同时采样15倍，效果并不好，除了12效果勉强可用。之后尝试单独对12进行50倍的上采样。

大标签下采样：

色相augment：效果明显，可用提升1~2个百分点，部分标签可提高数个百分点。

使用blue-》red色相变化：效果一般

四阶段学习（最后一个阶段打开所有层）：效果不如三阶段

1、使用自己编写的crop代替keras自带的crop：效果提升1个百分点。自己编写的crop是先裁剪再resize（可能对图像进行轻微的横向/纵向的拉伸，且有轻微缩放效果）

2、正方形crop：每次crop的时候都先crop为一个正方形然后再resize（去掉长宽的变形以及图像缩放）， 效果并不好

## Baseline

实现了一个简单的卷积神经网络。

Train loss可以接近为0，Train f2-score可用达到99.5%

用于验证算法的正确性（loss函数、代码框架等）

## ResNet50

### Model-1~6

分阶段迁移学习：一开始只训练最后两层，然后逐步打开预训练层的权重

阈值搜索：新增了基于BFGS的搜索和Greedy搜索，可以大幅提高准确率，但是不知道阈值的泛化能力如何（参考Kaggle比赛中的两支冠军队伍，均是在Validation集上搜索阈值）

图片随机增强：horizontal flip、random shift、random rotation

结果：最好可达到87.4%（训练12个Epoch达到最优，然后下降）

### Model-7

1. 取消了图像的mean、std预处理: 之前使用基于像素的图像mean、std预处理，从生成的图片来看有问题（Keras动态生成图片导出），后面可尝试改为基于通道的均值。
2. 使用include top=False预训练参数

#### Evaluate

weights.007.hdf5  
####### Smooth F2-Score is 0.752615 #######  
####### F2-Score with threshold 0.2 is 0.861973 #######  
####### F2-Score with threshold 0.1 is 0.860249 #######  
####### Basonhopping F2-Score is 0.872198 #######  
####### Greedy F2-Score is 0.873286 #######  
  
weights.009.hdf5  
####### Smooth F2-Score is 0.741987 #######  
####### F2-Score with threshold 0.2 is 0.848438 #######  
####### F2-Score with threshold 0.1 is 0.850239 #######  
####### Basonhopping F2-Score is 0.872164 #######  
####### Greedy F2-Score is 0.874792 #######  
  
weights.010.hdf5  
####### Smooth F2-Score is 0.794807 #######  
####### F2-Score with threshold 0.2 is 0.869044 #######  
####### F2-Score with threshold 0.1 is 0.872603 #######  
####### Basonhopping F2-Score is 0.876129 #######  
####### Greedy F2-Score is 0.877365 #######  
  
weights.011.hdf5  
####### Smooth F2-Score is 0.789265 #######  
####### F2-Score with threshold 0.2 is 0.860144 #######  
####### F2-Score with threshold 0.1 is 0.862353 #######  
####### Basonhopping F2-Score is 0.870077 #######  
####### Greedy F2-Score is 0.873587 #######  
  
weights.012.hdf5  
####### Smooth F2-Score is 0.800313 #######  
####### F2-Score with threshold 0.2 is 0.869751 #######  
####### F2-Score with threshold 0.1 is 0.874894 #######  
####### Basonhopping F2-Score is 0.878213 #######  
####### Greedy F2-Score is 0.879970 #######

[label 0] smooth-f2=0.153020 BFGS-f2=0.404412[0.062066] greedy-f2=0.401460[0.060000]

[label 1] smooth-f2=0.498293 BFGS-f2=0.634196[0.054479] greedy-f2=0.633372[0.070000]

[label 2] smooth-f2=0.945747 BFGS-f2=0.980983[0.019290] greedy-f2=0.981002[0.020000]

[label 3] smooth-f2=0.222902 BFGS-f2=0.324074[0.082953] greedy-f2=0.320122[0.080000]

[label 4] smooth-f2=0.598736 BFGS-f2=0.816145[0.077905] greedy-f2=0.815349[0.080000]

[label 5] smooth-f2=0.551574 BFGS-f2=0.741818[0.140408] greedy-f2=0.741306[0.140000]

[label 6] smooth-f2=0.185681 BFGS-f2=0.265957[0.337496] greedy-f2=0.265957[0.340000]

[label 7] smooth-f2=0.284397 BFGS-f2=0.438641[0.040893] greedy-f2=0.438287[0.040000]

[label 8] smooth-f2=0.943996 BFGS-f2=0.979409[0.013240] greedy-f2=0.979199[0.010000]

[label 9] smooth-f2=0.363826 BFGS-f2=0.613208[0.080324] greedy-f2=0.613213[0.090000]

[label 10] smooth-f2=0.400007 BFGS-f2=0.715000[0.054791] greedy-f2=0.712041[0.040000]

[label 11] smooth-f2=0.306027 BFGS-f2=0.496162[0.054027] greedy-f2=0.493492[0.050000]

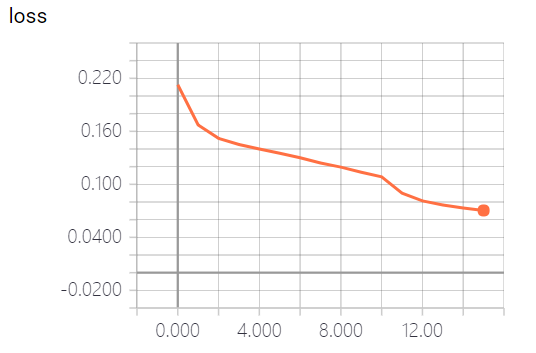
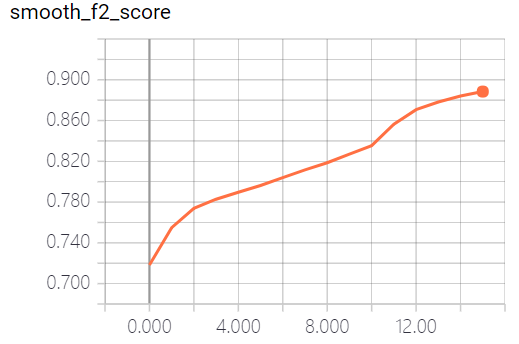
[label 12] smooth-f2=0.238251 BFGS-f2=0.348028[0.139986] greedy-f2=0.348028[0.140000]

####### Greedy F2-Score by single label is 0.877764 #######

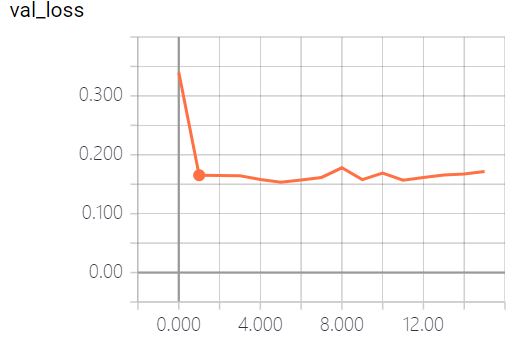
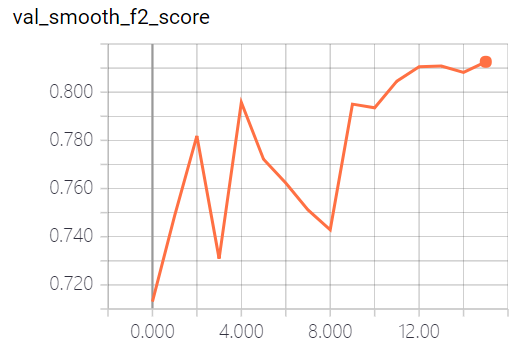
标签的评估结果存在极大的不均匀  
  
weights.013.hdf5  
####### F2-Score with threshold 0.2 is 0.867209 #######  
####### F2-Score with threshold 0.1 is 0.873079 #######  
####### Basonhopping F2-Score is 0.876897 #######  
####### Greedy F2-Score is 0.878053 #######  
  
weights.014.hdf5  
####### Smooth F2-Score is 0.804727 #######  
####### F2-Score with threshold 0.2 is 0.866394 #######  
####### F2-Score with threshold 0.1 is 0.871515 #######  
####### Basonhopping F2-Score is 0.875239 #######  
####### Greedy F2-Score is 0.876984 #######  
  
weights.015.hdf5  
####### Smooth F2-Score is 0.802782 #######  
####### F2-Score with threshold 0.2 is 0.863338 #######  
####### F2-Score with threshold 0.1 is 0.870111 #######  
####### Basonhopping F2-Score is 0.874347 #######  
####### Greedy F2-Score is 0.876150 #######

较之前模型，最优值提高约0.5%。均值提高约1%

#### Curve

Loss、smooth f2-Score在epoch=10时有一个加速，此时是打开了所有的层的权重进行训练，并降低学习率到0.00001

Train loss最低值为0.07，而Val 最低值为0.15。说明存在泛化问题。而Val smooth f2-score前期的波动性可能是由于只训练最后两层导致的。

Val loss前期降低非常快，然后趋于饱和，在epoch=10左右，开始出现上升倾向，可能是过拟合了。

结合Evaluation的结果，epoch 7~16变化不大，其中12最优，之后逐渐下降基本符合曲线中12之后loss缓慢上升的趋势。

还有一点是，从7~16，smooth f2-score 增加了几个百分点，但是阈值搜索后的f2-score只变化了零点几个百分点。

### Model-8

1. 去掉图像旋转：AlexNet、ResNet、DenseNet等论文中均为做图像旋转
2. 增大Val batch size为512：希望减少Val smooth f2-score的抖动

结果：

1. 取消掉image rotation几乎没有影响
2. Val batch size增大后，Val smooth f2-score曲线的确变得平滑了

#### Evaluate

weights.011.hdf5

####### Smooth F2-Score is 0.803181 #######

####### F2-Score with threshold 0.2 is 0.863644 #######

####### F2-Score with threshold 0.1 is 0.863083 #######

####### Greedy F2-Score is 0.874964 #######

weights.012.hdf5

####### Smooth F2-Score is 0.802437 #######

####### F2-Score with threshold 0.2 is 0.869768 #######

####### F2-Score with threshold 0.1 is 0.873220 #######

####### Greedy F2-Score is 0.879088 #######

weights.013.hdf5

####### Smooth F2-Score is 0.806060 #######

####### F2-Score with threshold 0.2 is 0.867854 #######

####### F2-Score with threshold 0.1 is 0.871998 #######

####### Greedy F2-Score is 0.877431 #######

weights.014.hdf5

####### Smooth F2-Score is 0.807866 #######

####### F2-Score with threshold 0.2 is 0.865808 #######

####### F2-Score with threshold 0.1 is 0.870474 #######

####### Greedy F2-Score is 0.876441 #######

weights.015.hdf5

####### Smooth F2-Score is 0.808137 #######

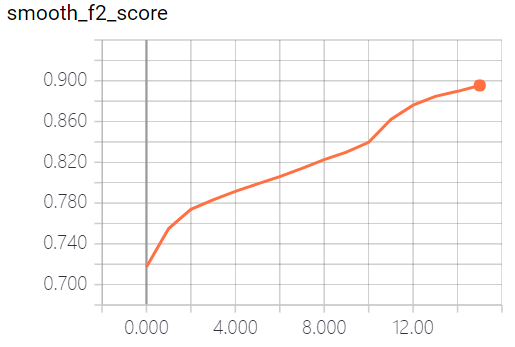
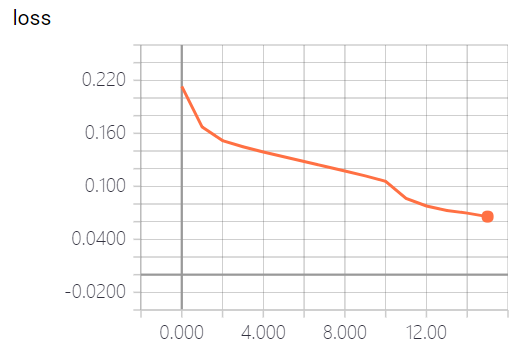
####### F2-Score with threshold 0.2 is 0.864625 #######

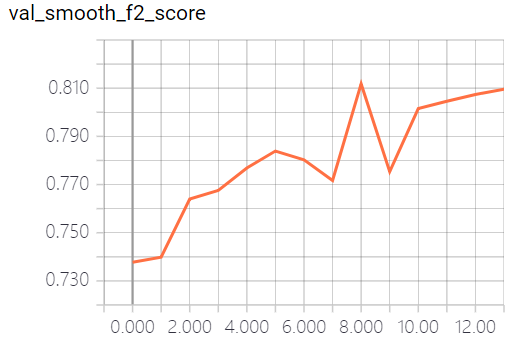
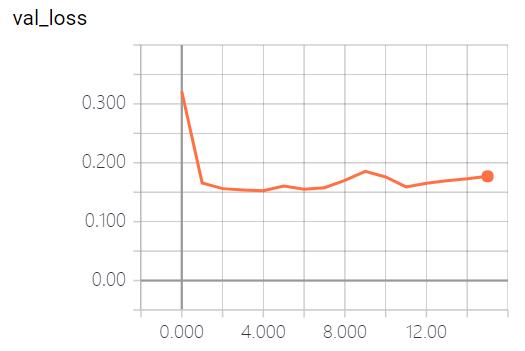
####### F2-Score with threshold 0.1 is 0.869519 #######

####### Greedy F2-Score is 0.876153 #######

与Model 7相比，差异不大

#### Curve





和Model 7差异不大

## InceptionResNetV2

## DensNet121

### Model 1

使用分级transfer learning

1：仅训练最后两层，lr=0.001

2~6：训练100层之后的权重，lr=0.0001

7~20：训练所有层，lr=0.00001

#### Evaluate

weights.006.hdf5

####### predict 10981 images spend 99 seconds ######

####### search greedy threshold spend 15 seconds ######

####### Smooth F2-Score is 0.775151 #######

####### F2-Score with threshold 0.2 is 0.871815 #######

####### F2-Score with threshold 0.1 is 0.870598 #######

####### Greedy F2-Score is 0.877627 #######

weights.007.hdf5

####### Smooth F2-Score is 0.786137 #######

####### F2-Score with threshold 0.2 is 0.880634 #######

####### F2-Score with threshold 0.1 is 0.877516 #######

####### Greedy F2-Score is 0.885350 #######

weights.008.hdf5

####### Smooth F2-Score is 0.793202 #######

####### F2-Score with threshold 0.2 is 0.880199 #######

####### F2-Score with threshold 0.1 is 0.879286 #######

####### Greedy F2-Score is 0.885232 #######

weights.009.hdf5

####### Smooth F2-Score is 0.791803 #######

####### F2-Score with threshold 0.2 is 0.879592 #######

####### F2-Score with threshold 0.1 is 0.878887 #######

####### Greedy F2-Score is 0.885013 #######

weights.010.hdf5

####### Smooth F2-Score is 0.791281 #######

####### F2-Score with threshold 0.2 is 0.878118 #######

####### F2-Score with threshold 0.1 is 0.878272 #######

####### Greedy F2-Score is 0.883909 #######

weights.011.hdf5

####### Smooth F2-Score is 0.791499 #######

####### F2-Score with threshold 0.2 is 0.875889 #######

####### F2-Score with threshold 0.1 is 0.877018 #######

####### Greedy F2-Score is 0.882867 #######

weights.012.hdf5

####### Smooth F2-Score is 0.799399 #######

####### F2-Score with threshold 0.2 is 0.876372 #######

####### F2-Score with threshold 0.1 is 0.876772 #######

####### Greedy F2-Score is 0.882706 #######

Load image mean: ['[0.8372988 0.84012526 0.8637541 ]']

Load image std: ['[0.00499635 0.00508581 0.0047855 ]']

weights.013.hdf5

####### Smooth F2-Score is 0.797504 #######

####### F2-Score with threshold 0.2 is 0.874816 #######

####### F2-Score with threshold 0.1 is 0.877803 #######

####### Greedy F2-Score is 0.883175 #######

weights.014.hdf5

####### Smooth F2-Score is 0.798624 #######

####### F2-Score with threshold 0.2 is 0.874295 #######

####### F2-Score with threshold 0.1 is 0.876805 #######

####### Greedy F2-Score is 0.882275 #######

weights.015.hdf5

####### Smooth F2-Score is 0.799785 #######

####### F2-Score with threshold 0.2 is 0.873373 #######

####### F2-Score with threshold 0.1 is 0.876170 #######

####### Greedy F2-Score is 0.881740 #######

weights.016.hdf5

####### Smooth F2-Score is 0.802412 #######

####### F2-Score with threshold 0.2 is 0.872065 #######

####### F2-Score with threshold 0.1 is 0.874351 #######

####### Greedy F2-Score is 0.879751 #######

weights.017.hdf5

####### Smooth F2-Score is 0.802762 #######

####### F2-Score with threshold 0.2 is 0.871442 #######

####### F2-Score with threshold 0.1 is 0.875009 #######

####### Greedy F2-Score is 0.879854 #######

weights.018.hdf5

####### Smooth F2-Score is 0.799736 #######

####### F2-Score with threshold 0.2 is 0.870077 #######

####### F2-Score with threshold 0.1 is 0.873717 #######

####### Greedy F2-Score is 0.879753 #######

weights.019.hdf5

####### Smooth F2-Score is 0.797830 #######

####### F2-Score with threshold 0.2 is 0.869414 #######

####### F2-Score with threshold 0.1 is 0.873306 #######

####### Greedy F2-Score is 0.879036 #######

weights.020.hdf5

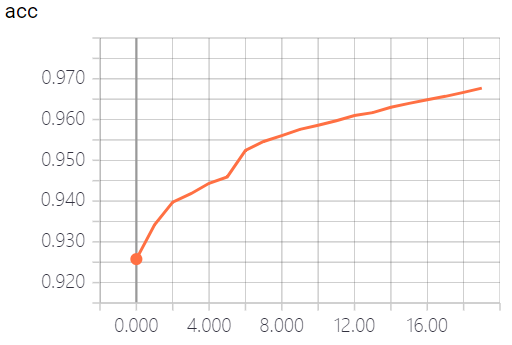
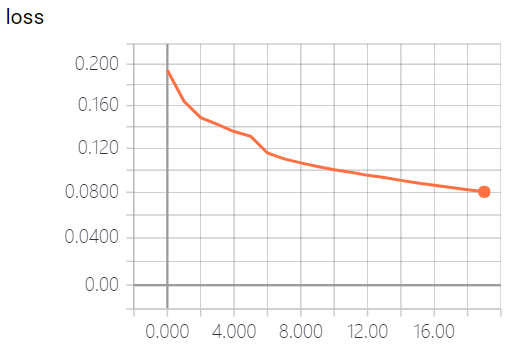
####### Smooth F2-Score is 0.801643 #######

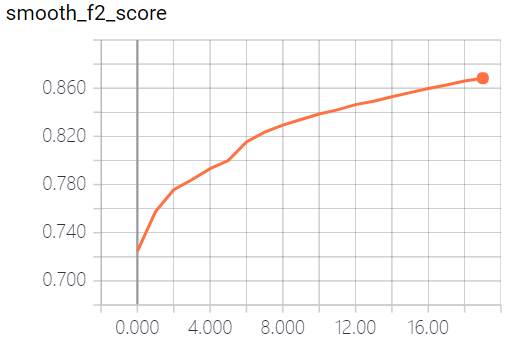
####### F2-Score with threshold 0.2 is 0.870481 #######

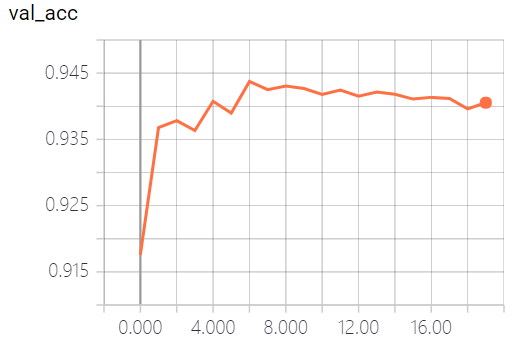
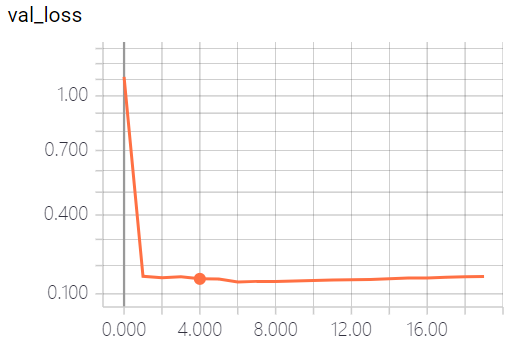
####### F2-Score with threshold 0.1 is 0.874482 #######

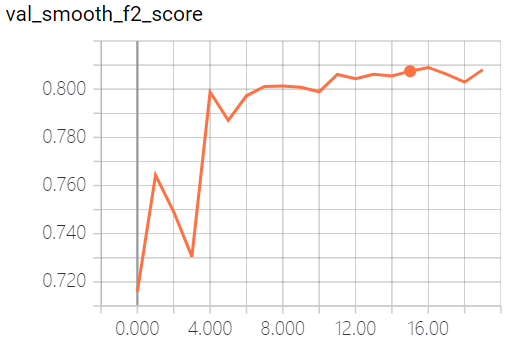
####### Greedy F2-Score is 0.879349 #######

#### Curve





## InceptionV3

### Model 1

baseline

weight: F:\Projects\competition\model\inceptionv3\record\1/val1\['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12']weights.008.hdf5  
Smooth F2-Score: 0.793786  
F2-Score with threshold 0.2: 0.877642  
Greedy F2-Score is: 0.882831  
Greedy threshold: 0.15,0.09,0.05,0.68,0.15,0.17,0.56,0.26,0.04,0.19,0.11,0.21,0.37  
[label 0] smooth-f2=0.117504 BFGS-f2=0.326087[0.149977] greedy-f2=0.326087[0.150000]  
[label 1] smooth-f2=0.494151 BFGS-f2=0.658755[0.087541] greedy-f2=0.657845[0.090000]  
[label 2] smooth-f2=0.939215 BFGS-f2=0.981257[0.048427] greedy-f2=0.981236[0.050000]  
[label 3] smooth-f2=0.206820 BFGS-f2=0.299625[0.237136] greedy-f2=0.300752[0.240000]  
[label 4] smooth-f2=0.599563 BFGS-f2=0.813178[0.107909] greedy-f2=0.812963[0.110000]  
[label 5] smooth-f2=0.514483 BFGS-f2=0.738950[0.150655] greedy-f2=0.738368[0.150000]  
[label 6] smooth-f2=0.101948 BFGS-f2=0.253807[0.130110] greedy-f2=0.253807[0.130000]  
[label 7] smooth-f2=0.253383 BFGS-f2=0.433778[0.064291] greedy-f2=0.434124[0.060000]  
[label 8] smooth-f2=0.936758 BFGS-f2=0.979768[0.052406] greedy-f2=0.979671[0.050000]  
[label 9] smooth-f2=0.383380 BFGS-f2=0.616801[0.111405] greedy-f2=0.616158[0.120000]  
[label 10] smooth-f2=0.522957 BFGS-f2=0.750955[0.101422] greedy-f2=0.751383[0.110000]  
[label 11] smooth-f2=0.296374 BFGS-f2=0.491124[0.109981] greedy-f2=0.491124[0.110000]  
[label 12] smooth-f2=0.154647 BFGS-f2=0.278970[0.099650] greedy-f2=0.280172[0.100000]

参数：epoch=[1, 6, 12], lr=[0.001, 0.0001, 0.00001], freeze\_layers=[-1, 0.5, 0]

base\_model = keras.applications.InceptionV3(include\_top=False, input\_shape=model\_config.image\_shape, pooling="max")  
x = base\_model.output  
x = Dense(128, use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)  
x = Dense(128, use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)

优化算法：adam，默认参数

### Model 2

两个Dense中间层

weight: F:\Projects\competition\model\inceptionv3\record\2/val1\['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12']weights.007.hdf5  
Smooth F2-Score: 0.792085  
F2-Score with threshold 0.2: 0.878309  
Greedy F2-Score is: 0.882348  
Greedy threshold: 0.12,0.11,0.06,0.28,0.2,0.13,0.16,0.14,0.07,0.17,0.1,0.18,0.31  
[label 0] smooth-f2=0.043846 BFGS-f2=0.246305[0.043544] greedy-f2=0.231481[0.040000]  
[label 1] smooth-f2=0.468800 BFGS-f2=0.677114[0.108839] greedy-f2=0.677507[0.110000]  
[label 2] smooth-f2=0.941623 BFGS-f2=0.981047[0.150765] greedy-f2=0.981047[0.150000]  
[label 3] smooth-f2=0.204768 BFGS-f2=0.324859[0.052084] greedy-f2=0.323034[0.050000]  
[label 4] smooth-f2=0.603267 BFGS-f2=0.815606[0.137213] greedy-f2=0.815347[0.140000]  
[label 5] smooth-f2=0.494641 BFGS-f2=0.733818[0.120094] greedy-f2=0.733748[0.120000]  
[label 6] smooth-f2=0.050684 BFGS-f2=0.232558[0.114944] greedy-f2=0.229885[0.110000]  
[label 7] smooth-f2=0.235283 BFGS-f2=0.477941[0.100105] greedy-f2=0.477941[0.100000]  
[label 8] smooth-f2=0.940460 BFGS-f2=0.979482[0.072911] greedy-f2=0.979464[0.100000]  
[label 9] smooth-f2=0.364089 BFGS-f2=0.609496[0.087372] greedy-f2=0.608727[0.080000]  
[label 10] smooth-f2=0.451833 BFGS-f2=0.742058[0.068494] greedy-f2=0.741142[0.080000]  
[label 11] smooth-f2=0.270869 BFGS-f2=0.506166[0.059495] greedy-f2=0.506427[0.060000]  
[label 12] smooth-f2=0.111038 BFGS-f2=0.309051[0.090200] greedy-f2=0.308370[0.090000]

epoch=[1, 6, 12], lr=[0.001, 0.0001, 0.00001], freeze\_layers=[-1, 0.5, 0]

base\_model = keras.applications.InceptionV3(include\_top=False, input\_shape=model\_config.image\_shape, pooling="max")  
x = base\_model.output  
x = Dense(128, use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)  
x = Dense(128, use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)

optimizer=keras.optimizers.adam(lr=lr),

### Model 3

使用global avg pooling

weight: F:\Projects\competition\model\inceptionv3\record\3/val1\['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12']weights.007.hdf5  
Smooth F2-Score: 0.796726  
F2-Score with threshold 0.2: 0.878855  
Greedy F2-Score is: 0.883117  
Greedy threshold: 0.15,0.09,0.06,0.59,0.18,0.15,0.81,0.14,0.06,0.19,0.13,0.13,0.34  
[label 0] smooth-f2=0.054002 BFGS-f2=0.241379[0.072766] greedy-f2=0.236486[0.070000]  
[label 1] smooth-f2=0.491883 BFGS-f2=0.661998[0.092652] greedy-f2=0.659974[0.090000]  
[label 2] smooth-f2=0.941480 BFGS-f2=0.981435[0.063898] greedy-f2=0.981368[0.110000]  
[label 3] smooth-f2=0.218803 BFGS-f2=0.271318[0.340651] greedy-f2=0.362319[0.060000]  
[label 4] smooth-f2=0.610572 BFGS-f2=0.818948[0.099791] greedy-f2=0.818679[0.100000]  
[label 5] smooth-f2=0.505000 BFGS-f2=0.733995[0.145668] greedy-f2=0.733333[0.100000]  
[label 6] smooth-f2=0.167433 BFGS-f2=0.269231[0.082156] greedy-f2=0.268817[0.070000]  
[label 7] smooth-f2=0.264206 BFGS-f2=0.486753[0.125964] greedy-f2=0.486284[0.130000]  
[label 8] smooth-f2=0.939920 BFGS-f2=0.979965[0.075256] greedy-f2=0.979892[0.090000]  
[label 9] smooth-f2=0.400085 BFGS-f2=0.614368[0.143252] greedy-f2=0.611696[0.140000]  
[label 10] smooth-f2=0.503343 BFGS-f2=0.745656[0.088132] greedy-f2=0.744365[0.120000]  
[label 11] smooth-f2=0.308339 BFGS-f2=0.528190[0.114235] greedy-f2=0.526631[0.120000]  
[label 12] smooth-f2=0.168030 BFGS-f2=0.341981[0.110247] greedy-f2=0.341176[0.110000]

epoch=[1, 6, 12],  
lr=[0.001, 0.0001, 0.00001],  
freeze\_layers=[-1, 0.5, 0])

base\_model = keras.applications.InceptionV3(include\_top=False, input\_shape=model\_config.image\_shape, pooling="avg")  
x = base\_model.output  
x = Dense(256, use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)

optimizer=keras.optimizers.adam(lr=lr),

### Model 4

bn之间进行relu，之后也进行relu（之前由于手误，有几个模型均写成了这样，因此要测一测有没有负面效果）

weight: F:\Projects\competition\model\inceptionv3\record\4/val1\['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12']weights.007.hdf5  
Smooth F2-Score: 0.787900  
F2-Score with threshold 0.2: 0.878091  
Greedy F2-Score is: 0.882635  
Greedy threshold: 0.21,0.13,0.04,0.38,0.17,0.15,0.38,0.21,0.1,0.16,0.17,0.12,0.23  
[label 0] smooth-f2=0.085655 BFGS-f2=0.298507[0.073037] greedy-f2=0.287770[0.070000]  
[label 1] smooth-f2=0.479988 BFGS-f2=0.659932[0.121100] greedy-f2=0.659351[0.130000]  
[label 2] smooth-f2=0.935598 BFGS-f2=0.980852[0.043845] greedy-f2=0.980717[0.040000]  
[label 3] smooth-f2=0.227292 BFGS-f2=0.333333[0.034021] greedy-f2=0.342466[0.040000]  
[label 4] smooth-f2=0.596918 BFGS-f2=0.816955[0.149739] greedy-f2=0.816641[0.150000]  
[label 5] smooth-f2=0.501536 BFGS-f2=0.742761[0.130869] greedy-f2=0.742331[0.130000]  
[label 6] smooth-f2=0.060773 BFGS-f2=0.071429[0.379579] greedy-f2=0.237154[0.060000]  
[label 7] smooth-f2=0.240253 BFGS-f2=0.468561[0.121398] greedy-f2=0.467431[0.120000]  
[label 8] smooth-f2=0.932896 BFGS-f2=0.979619[0.090968] greedy-f2=0.979672[0.100000]  
[label 9] smooth-f2=0.375479 BFGS-f2=0.621563[0.116307] greedy-f2=0.620429[0.140000]  
[label 10] smooth-f2=0.483160 BFGS-f2=0.742857[0.099408] greedy-f2=0.742374[0.100000]  
[label 11] smooth-f2=0.300951 BFGS-f2=0.517435[0.095080] greedy-f2=0.515670[0.100000]  
[label 12] smooth-f2=0.121701 BFGS-f2=0.338809[0.063498] greedy-f2=0.336538[0.090000]

epoch=[1, 6, 12],  
lr=[0.001, 0.0001, 0.00001],  
freeze\_layers=[-1, 0.5, 0])

base\_model = keras.applications.InceptionV3(include\_top=False, input\_shape=model\_config.image\_shape, pooling="max")  
x = base\_model.output  
x = Dense(256, activation="relu", use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)

optimizer=keras.optimizers.adam(lr=lr)

### Model 5

修改第二阶段的freeze layer为20%  
weight: F:\Projects\competition\model\inceptionv3\record\5/val1\['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12']weights.007.hdf5  
Smooth F2-Score: 0.793022  
F2-Score with threshold 0.2: 0.878182  
Greedy F2-Score is: 0.882797  
Greedy threshold: 0.09,0.1,0.08,0.46,0.18,0.15,0.44,0.13,0.07,0.16,0.12,0.1,0.2  
[label 0] smooth-f2=0.074758 BFGS-f2=0.350877[0.095209] greedy-f2=0.344828[0.090000]  
[label 1] smooth-f2=0.474030 BFGS-f2=0.665142[0.097773] greedy-f2=0.664309[0.100000]  
[label 2] smooth-f2=0.941529 BFGS-f2=0.981228[0.096892] greedy-f2=0.981221[0.110000]  
[label 3] smooth-f2=0.222799 BFGS-f2=0.306513[0.292819] greedy-f2=0.363924[0.080000]  
[label 4] smooth-f2=0.604956 BFGS-f2=0.815478[0.142467] greedy-f2=0.814896[0.170000]  
[label 5] smooth-f2=0.478857 BFGS-f2=0.735607[0.122292] greedy-f2=0.734788[0.140000]  
[label 6] smooth-f2=0.107889 BFGS-f2=0.256410[0.127443] greedy-f2=0.260417[0.130000]  
[label 7] smooth-f2=0.234356 BFGS-f2=0.460449[0.118643] greedy-f2=0.458384[0.130000]  
[label 8] smooth-f2=0.940307 BFGS-f2=0.979597[0.070602] greedy-f2=0.979648[0.110000]  
[label 9] smooth-f2=0.379974 BFGS-f2=0.619441[0.086713] greedy-f2=0.617157[0.090000]  
[label 10] smooth-f2=0.455315 BFGS-f2=0.741185[0.094029] greedy-f2=0.741056[0.110000]  
[label 11] smooth-f2=0.271743 BFGS-f2=0.519935[0.063369] greedy-f2=0.519062[0.090000]  
[label 12] smooth-f2=0.155711 BFGS-f2=0.322581[0.183669] greedy-f2=0.322997[0.130000]

epoch=[1, 6, 12],  
lr=[0.001, 0.0001, 0.00001],  
freeze\_layers=[-1, 0.2, 0])

base\_model = keras.applications.InceptionV3(include\_top=False, input\_shape=model\_config.image\_shape, pooling="max")  
x = base\_model.output  
x = Dense(256, use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)

optimizer=keras.optimizers.Adam(lr=lr),

### Model 6

修改第三阶段的freeze layer为5

weight: F:\Projects\competition\model\inceptionv3\record\6/val1\['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12']weights.007.hdf5  
Smooth F2-Score: 0.793766  
F2-Score with threshold 0.2: 0.879008  
Greedy F2-Score is: 0.883701  
Greedy threshold: 0.14,0.09,0.06,0.23,0.17,0.19,0.46,0.2,0.05,0.16,0.17,0.17,0.45  
[label 0] smooth-f2=0.107232 BFGS-f2=0.312500[0.058250] greedy-f2=0.300000[0.140000]  
[label 1] smooth-f2=0.484697 BFGS-f2=0.666978[0.093012] greedy-f2=0.666109[0.090000]  
[label 2] smooth-f2=0.939191 BFGS-f2=0.981241[0.062705] greedy-f2=0.981144[0.060000]  
[label 3] smooth-f2=0.217992 BFGS-f2=0.332168[0.170418] greedy-f2=0.331010[0.170000]  
[label 4] smooth-f2=0.613982 BFGS-f2=0.818334[0.152417] greedy-f2=0.818588[0.160000]  
[label 5] smooth-f2=0.507319 BFGS-f2=0.737508[0.123577] greedy-f2=0.736700[0.130000]  
[label 6] smooth-f2=0.103734 BFGS-f2=0.244361[0.074378] greedy-f2=0.238971[0.070000]  
[label 7] smooth-f2=0.273451 BFGS-f2=0.474359[0.154584] greedy-f2=0.473485[0.150000]  
[label 8] smooth-f2=0.937561 BFGS-f2=0.979430[0.051636] greedy-f2=0.979450[0.080000]  
[label 9] smooth-f2=0.361231 BFGS-f2=0.619656[0.117596] greedy-f2=0.618818[0.120000]  
[label 10] smooth-f2=0.499927 BFGS-f2=0.743038[0.083957] greedy-f2=0.740784[0.080000]  
[label 11] smooth-f2=0.287664 BFGS-f2=0.507070[0.046050] greedy-f2=0.505506[0.050000]  
[label 12] smooth-f2=0.135111 BFGS-f2=0.313199[0.089065] greedy-f2=0.313953[0.100000]

epoch=[1, 6, 12],  
lr=[0.001, 0.0001, 0.00001],  
freeze\_layers=[-1, 0.5, 5])

base\_model = keras.applications.InceptionV3(include\_top=False, input\_shape=model\_config.image\_shape, pooling="max")  
x = base\_model.output  
x = Dense(256, use\_bias=False)(x)  
x = BatchNormalization()(x)  
x = Activation("relu")(x)

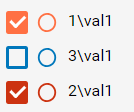
optimizer=keras.optimizers.Adam(lr=lr),

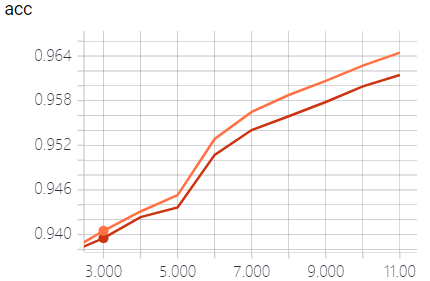
### 模型对比

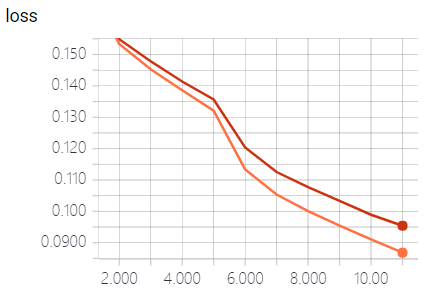
#### 1 vs 2

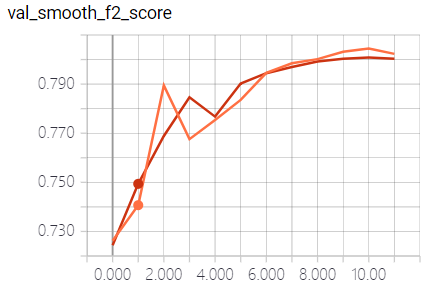
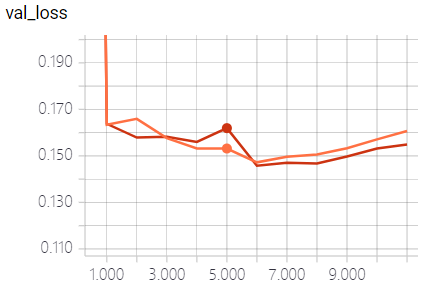
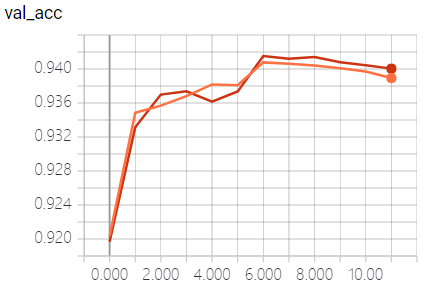
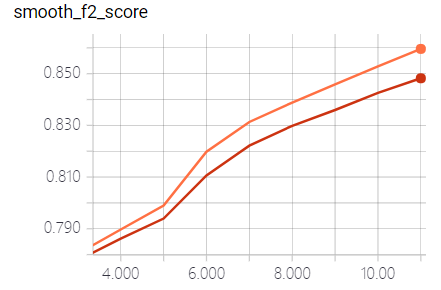
2增加了一个隐藏层，但是loss反而更高？

2的泛化能力更好一点点，但不明显可能是噪声。从greedy f2来看，1更优



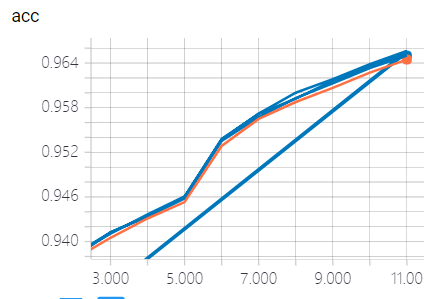


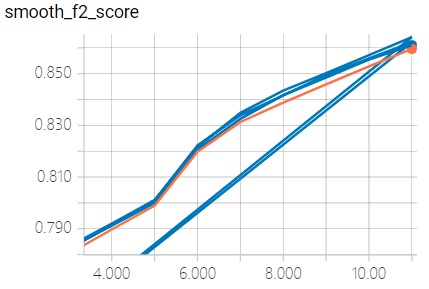
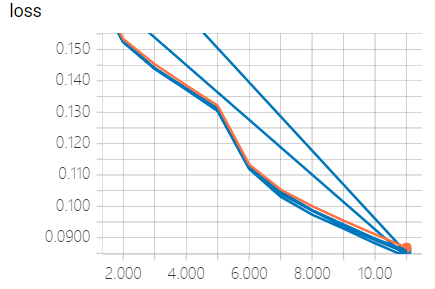


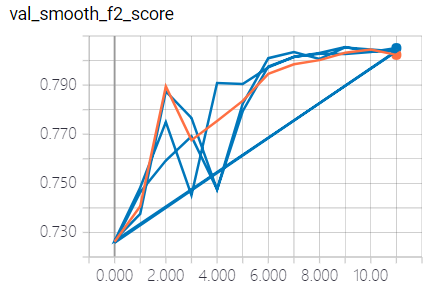
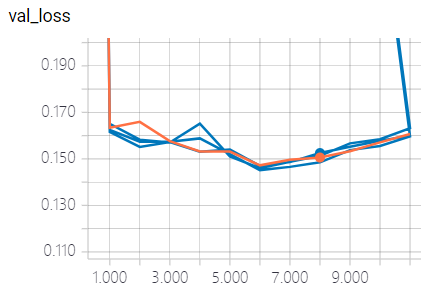
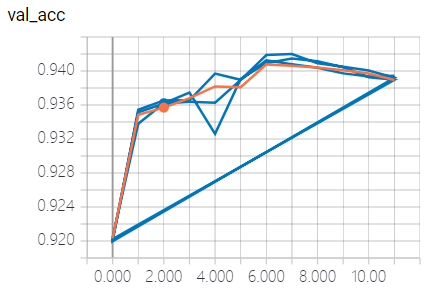


#### 1 vs 3

3从bias、variance来说都更优秀一点，greedy f2 score只高0.05%

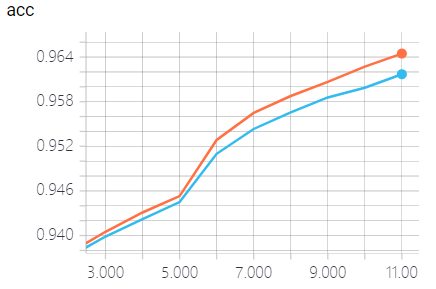


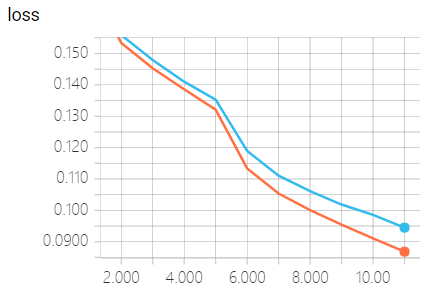


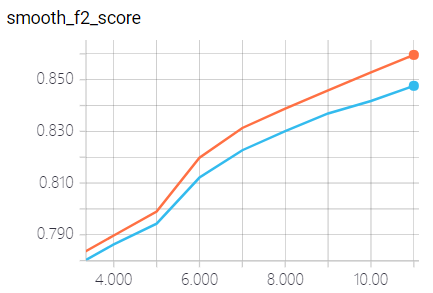


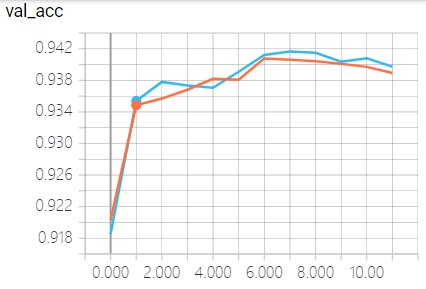
#### 1 vs 4

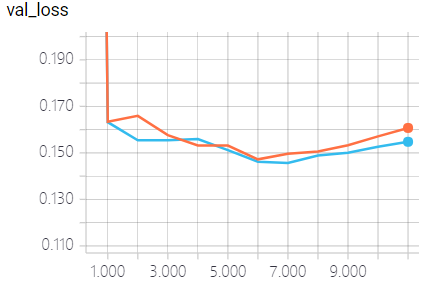
4在各方面都更差一点，证明4这种模型是有负面影响的，尽管greedy f2 只低了0.02%，但其他方面差距相对较大。

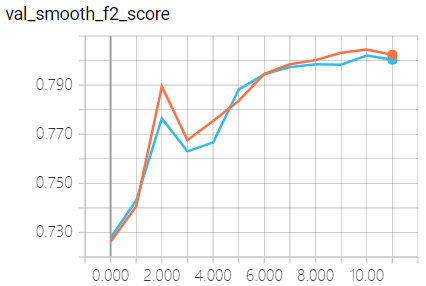






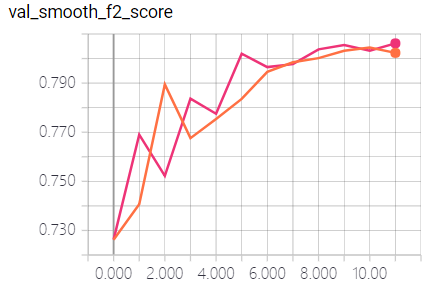
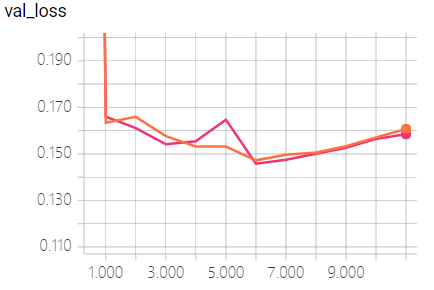
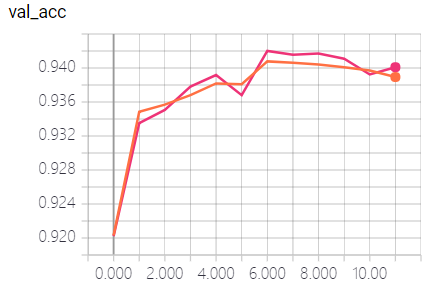
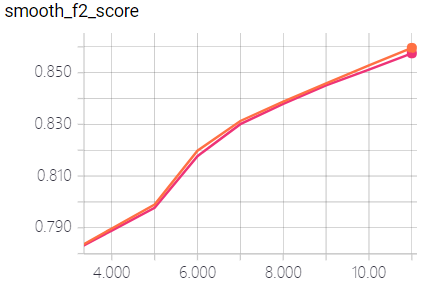
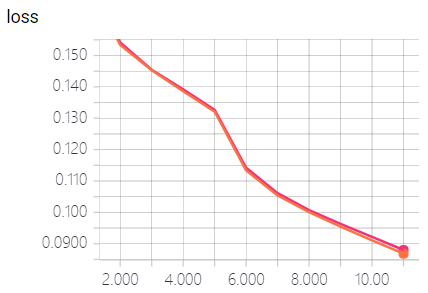
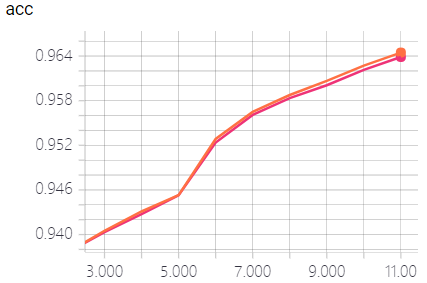






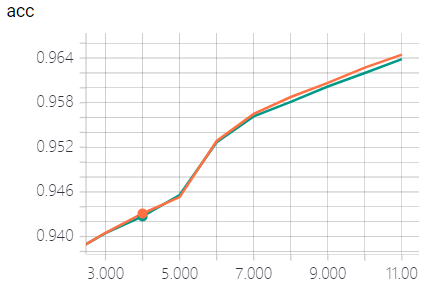
#### 1 vs 5

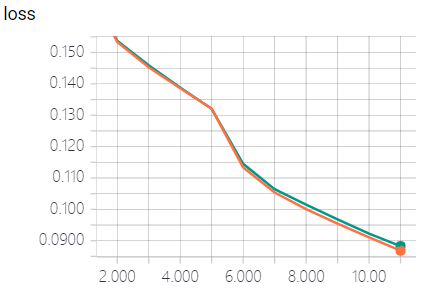
1 更好一些，可能是因为数据较小，因此少训练一些层效果更好，模型6也得出相同的结果。

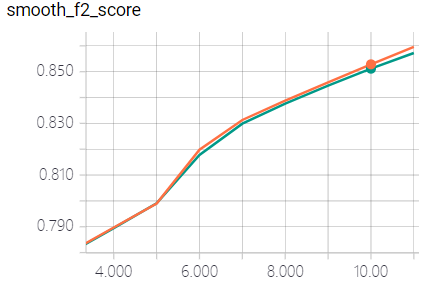


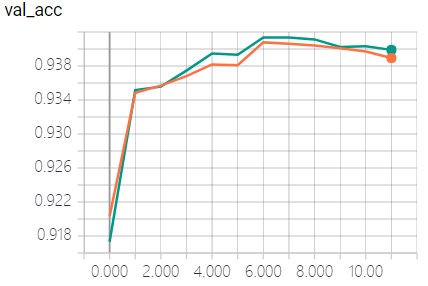
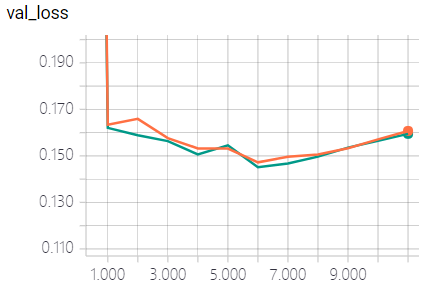
#### 1 vs 6

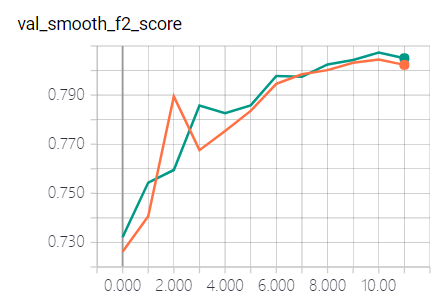
6具有更好的泛化能力，greedy f2要好0.1%











#### 整体分析：

1. 模型在8个Epoch左右，开始过拟合（曲线出现拐点，且Greedy f2 在这附近最优）。
   1. 数据量不够，在打开了所有的层数

