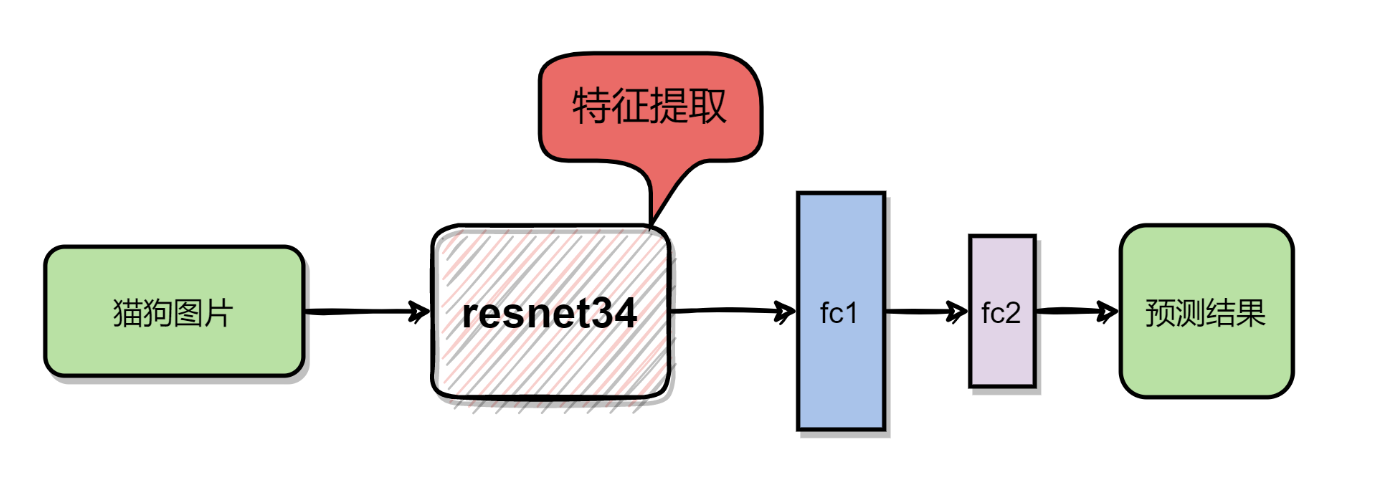
猫/狗分类任务

## **数据增强**

为了提高模型的能力，可以使用pytorch自带的Transforms对图片进行处理变换。在训练时，可以对图片进行一定的剪裁，旋转

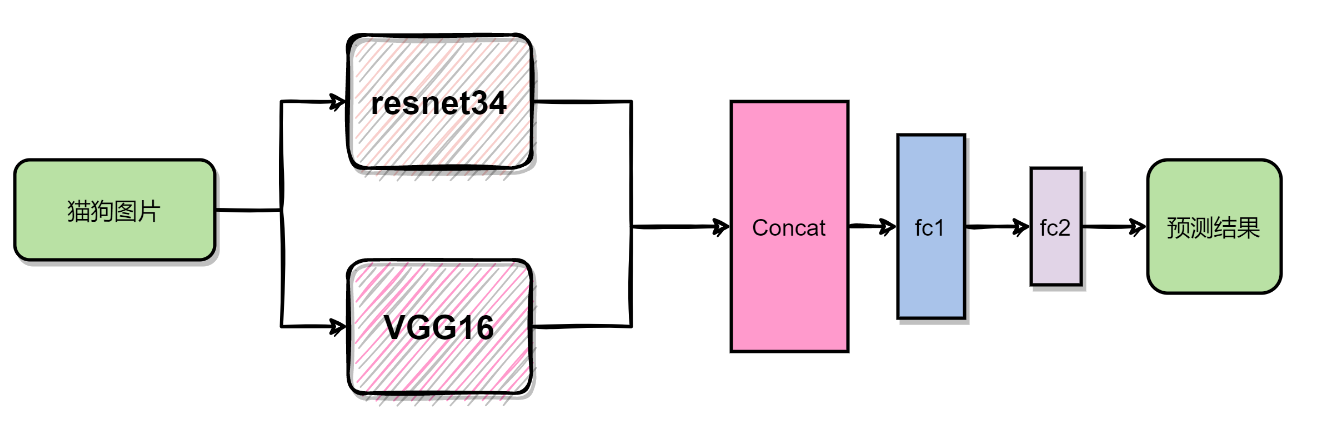
## **使用resnet34做特征提取**

resnet34使用的是torchvision中自带的模型，去除最后一层的全连接层，将前面的**卷积层**用于特征提取。然后将特征提取的结果进行Flatten，输入到全连接层，最终输出预测结果



* 使用resnet34做特征提取：93.6%93.6%。

## **resnet34&vgg16做特征提取**



* 使用resnet34和VGG16做特征提取：94.88%94.88%。

使用feature\_extract.py提取特征。

采用迁移学习的思想，使用Pytorch预训练的模型“GoogLeNet”、“ResNet”和“ResNeXt”提取图像特征。

选择预训练模型的全局平均池化层的输出为新的特征，注意到对于每张图像，GoogLeNet提取到1024维特征；ResNet和ResNeXt提取到2048维特征；最后组合成5120维特征。

将多个不同的网络输出的特征向量先保存下来，以便后续的训练，这样做的好处是我们一旦保存了特征向量，即使是在普通笔记本上也能轻松训练。

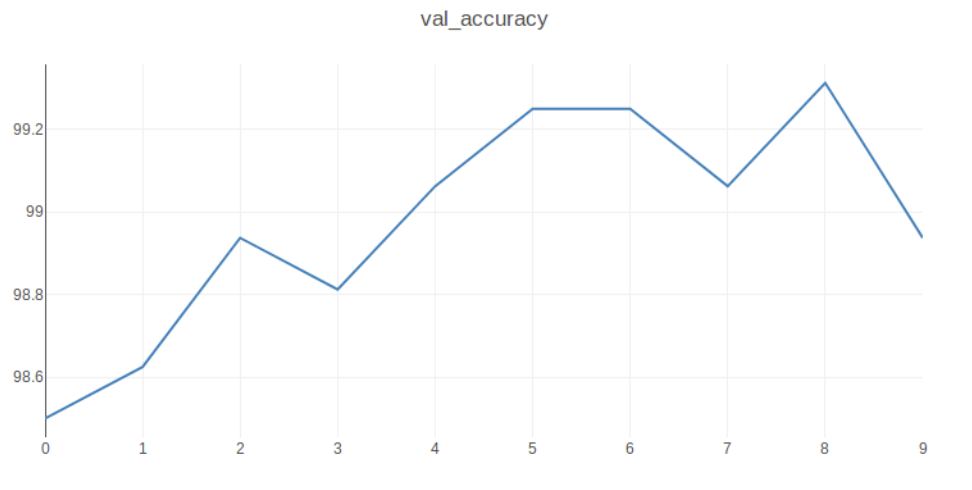
### tep2：模型训练

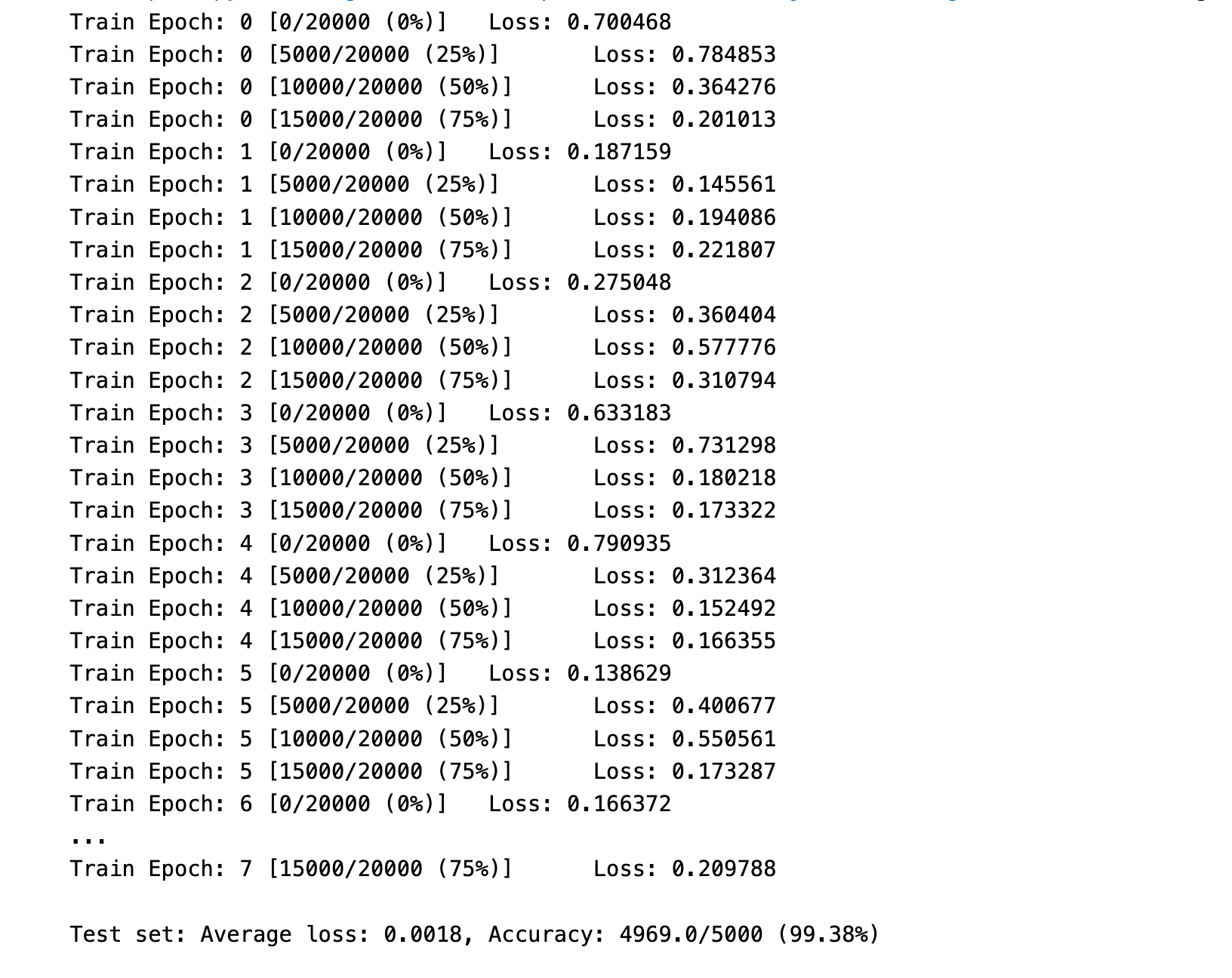
使用train\_test.py训练模型。

使用提取的特征作为输入进行二分类，直接用一个全连接层，输入5120维，输出2维(Softmax分类)。

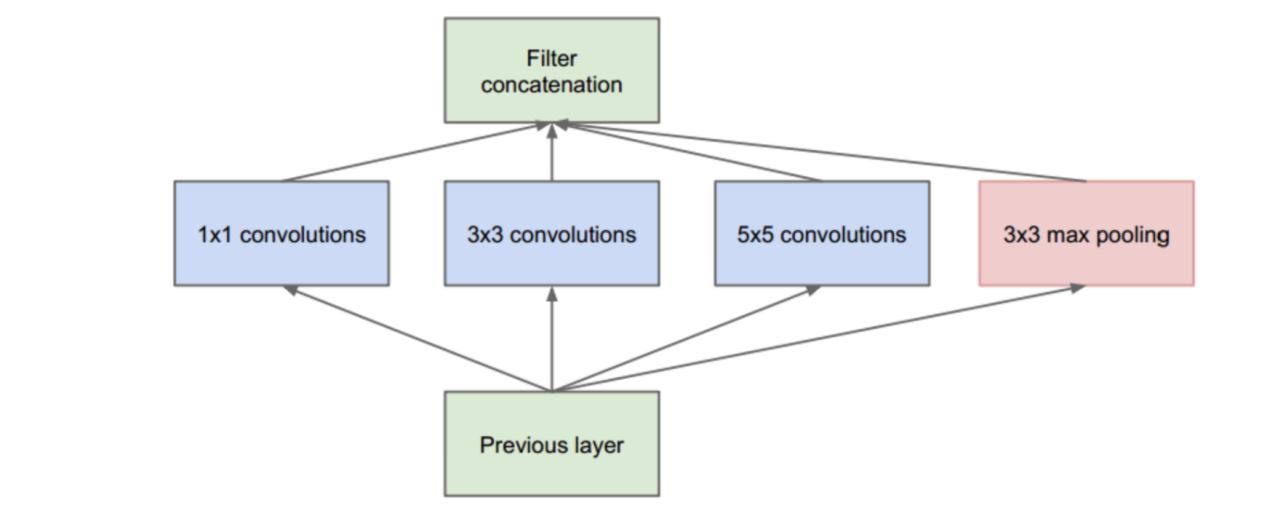
使用Dropout，设置p=0.5。

训练速度相对于使用raw image就很快了，CPU上几秒完成





GoogLeNet是谷歌在2014年提出的一种CNN深度学习方法前一层用三个单独的卷积核扫描，并加一个池化操作，然后把这四个操作的输出串联到一起作为下一层的输入



对于猫狗分类任务，您可以尝试使用以下分类器来训练和预测：

1. 支持向量机（SVM）：SVM是一种常见的分类算法，适用于二分类问题。它可以通过在高维空间中找到一个最优的超平面来进行分类。您可以使用Scikit-learn库中的SVM实现进行训练和预测。

2. 随机森林（Random Forest）：随机森林是一种集成学习算法，由多个决策树组成。它通过投票或平均预测结果来进行分类。Scikit-learn库中的RandomForestClassifier可以用于训练和预测。

3. K最近邻（K-Nearest Neighbors，KNN）：KNN算法根据最近的K个邻居来进行分类。在训练过程中，它会存储训练样本的特征向量和标签，然后根据测试样本的特征向量找到最近的K个邻居，并根据它们的标签进行分类。Scikit-learn库中的KNeighborsClassifier可用于训练和预测。

4. 多层感知机（Multilayer Perceptron，MLP）：MLP是一种人工神经网络模型，具有多个隐藏层。它可以用于解决复杂的非线性分类问题。您可以使用Keras或PyTorch等深度学习库来构建和训练MLP模型。

这些分类器都是常见的用于图像分类任务的算法，您可以根据数据集的大小、复杂性和计算资源的可用性选择合适的分类器进行实验和训练。另外，这些分类器还可以通过调整超参数、进行特征选择等方式进行优化，以提高分类性能。

自定义cnn

<https://github.com/JerrybroDu/cats_vs_dogs/tree/main>

600星的将不同模型单独跑一便

* gap\_ResNet50.h5
* gap\_InceptionV3.h5
* gap\_Xception.h5

然后进行结合

每个模型的准确率

<https://github.com/mrgloom/kaggle-dogs-vs-cats-caffe>

微调过程详细

<https://github.com/aleksas/keras-fine-tune-inception>

正确97

https://github.com/RomanKornev/dogs-vs-cats-redux