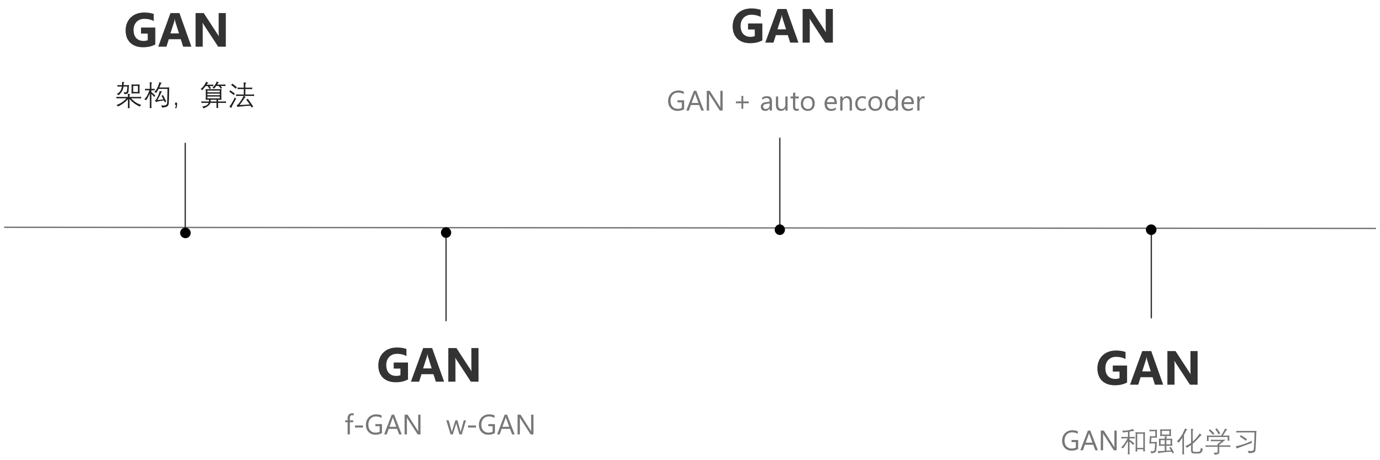
每周总结



GAN

架构，算法

GAN

f-GAN w-GAN

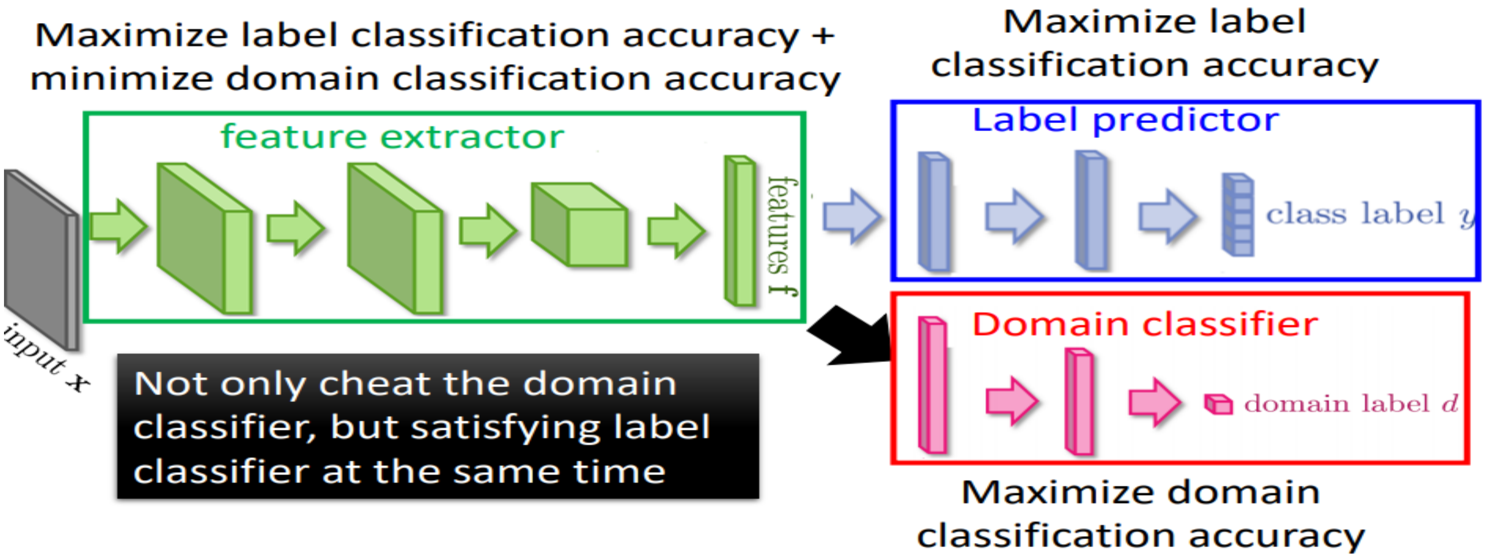
GAN

GAN + auto encoder

GAN

GAN和强化学习

模型结构



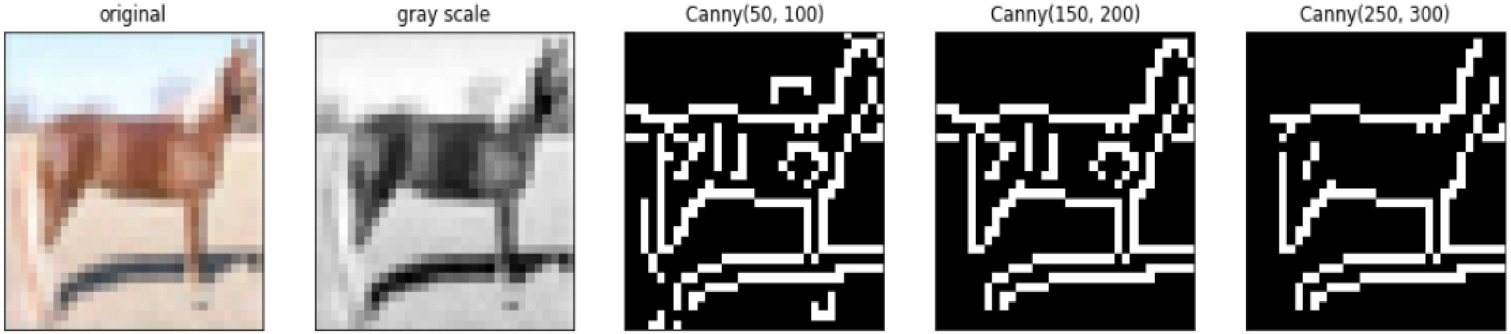
Maximize label classification accuracy + minimize domain classification accuracy

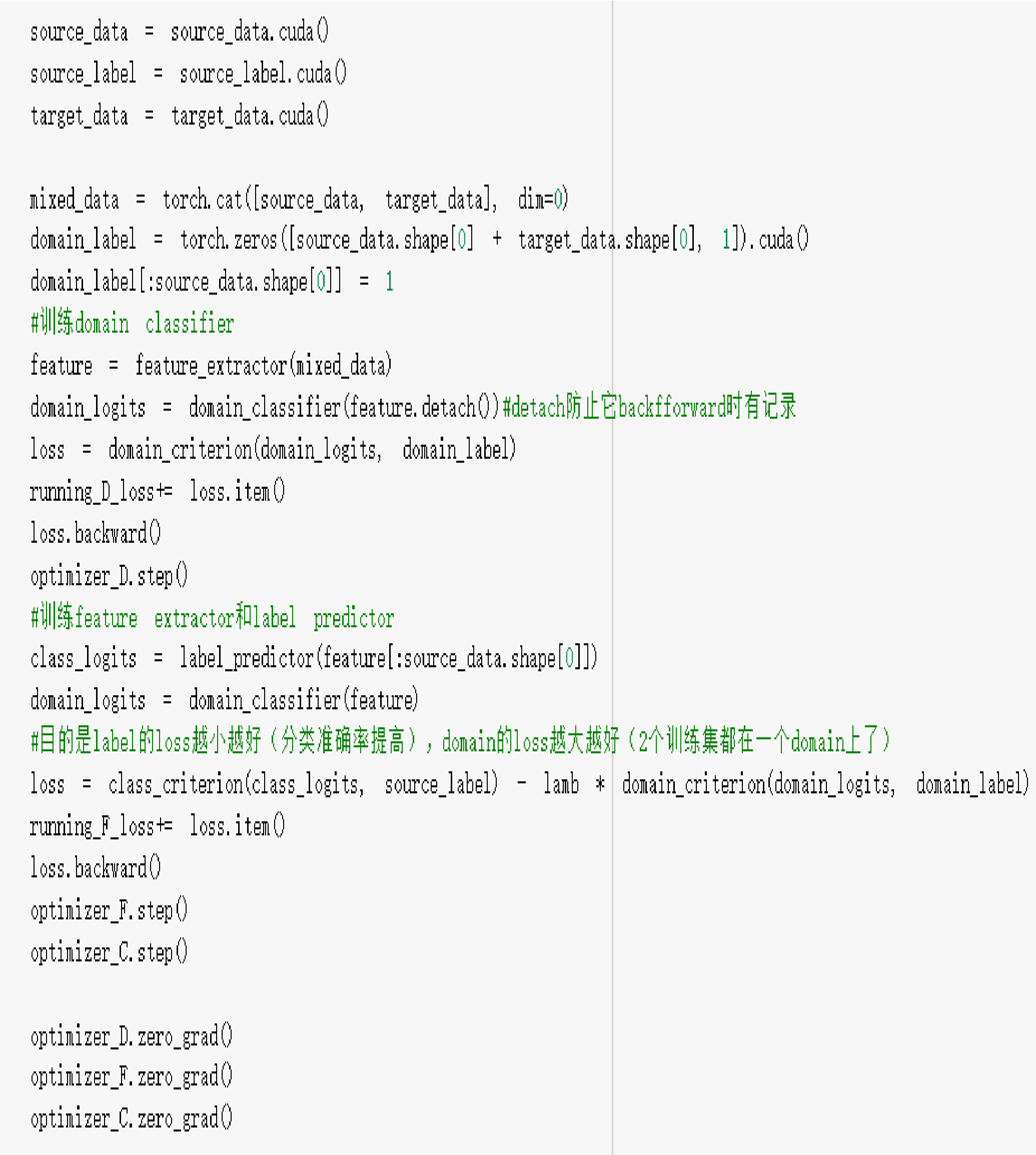
Maximize label classification accuracy

Not only cheat the domain classifier, but satisfying label classifier at the same time

Maximize domain classification accuracy

数据集：有标签的训练集和无标签的目标集





```
source_data = source_data.cuda()
source_label = source_label.cuda()
target_data = target_data.cuda()

mixed_data = torch.cat([source_data, target_data], dim=0)
domain_label = torch.zeros([source_data.shape[0] + target_data.shape[0], 1]).cuda()
domain_label[:source_data.shape[0]] = 1
#训练domain classifier
feature = feature_extractor(mixed_data)
domain_logits = domain_classifier(feature.detach())#detach防止它backfforward时有记录
loss = domain_criterion(domain_logits, domain_label)
running_D_loss+= loss.item()
loss.backward()
optimizer_D.step()
#训练feature extractor和label predictor
class_logits = label_predictor(feature[:source_data.shape[0]])
domain_logits = domain_classifier(feature)
#目的是label的loss越小越好（分类准确率提高），domain的loss越大越好（2个训练集都在一个domain上了）
loss = class_criterion(class_logits, source_label) - lamb * domain_criterion(domain_logits, domain_label)
running_F_loss+= loss.item()
loss.backward()
optimizer_F.step()
optimizer_C.step()

optimizer_D.zero_grad()
optimizer_F.zero_grad()
optimizer_C.zero_grad()
```

步骤一：

将训练集和目标集和在一起训练，训练集标1，目标集标0用于步骤二。

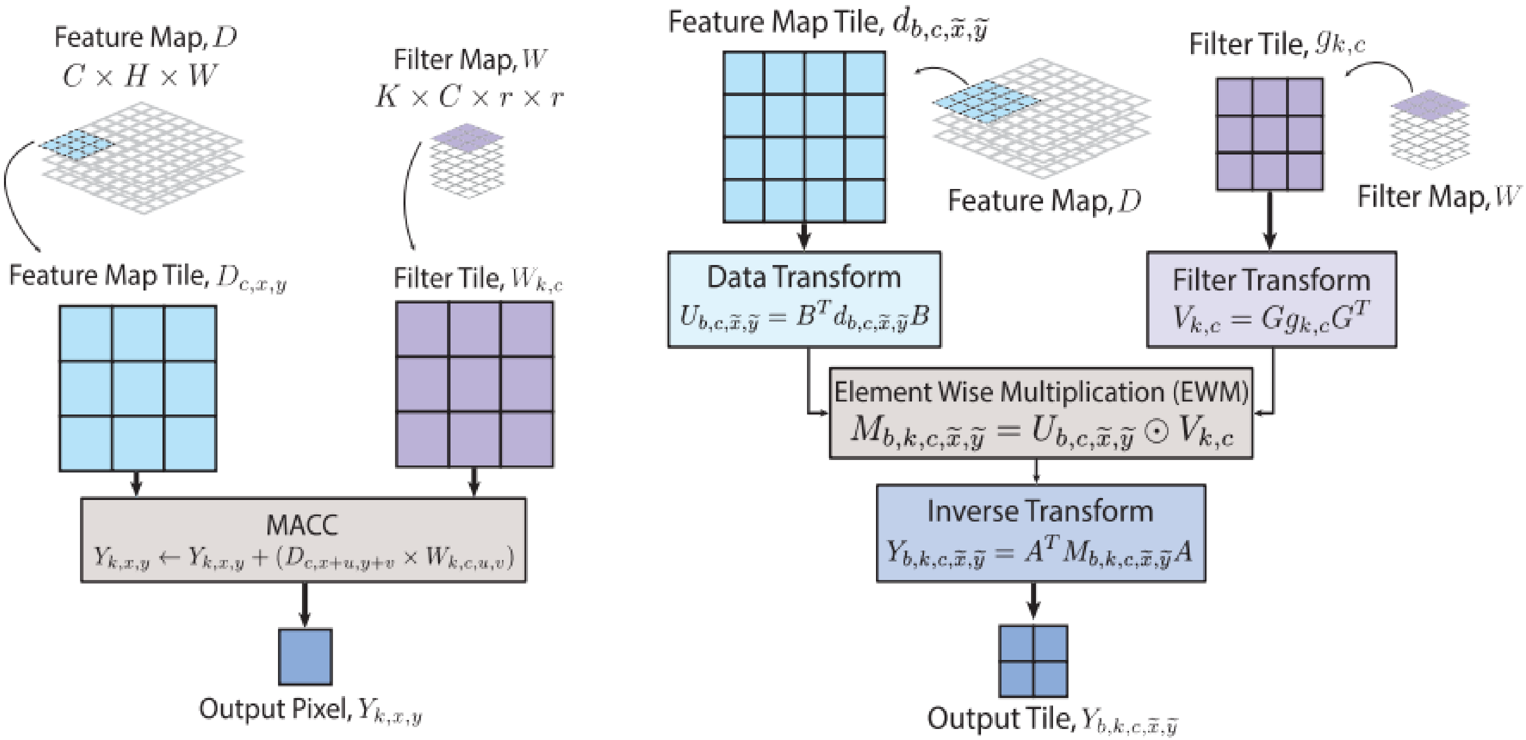
步骤二：

训练分类器，尽可能将训练集和目标集分离。

步骤三：

训练主网络。

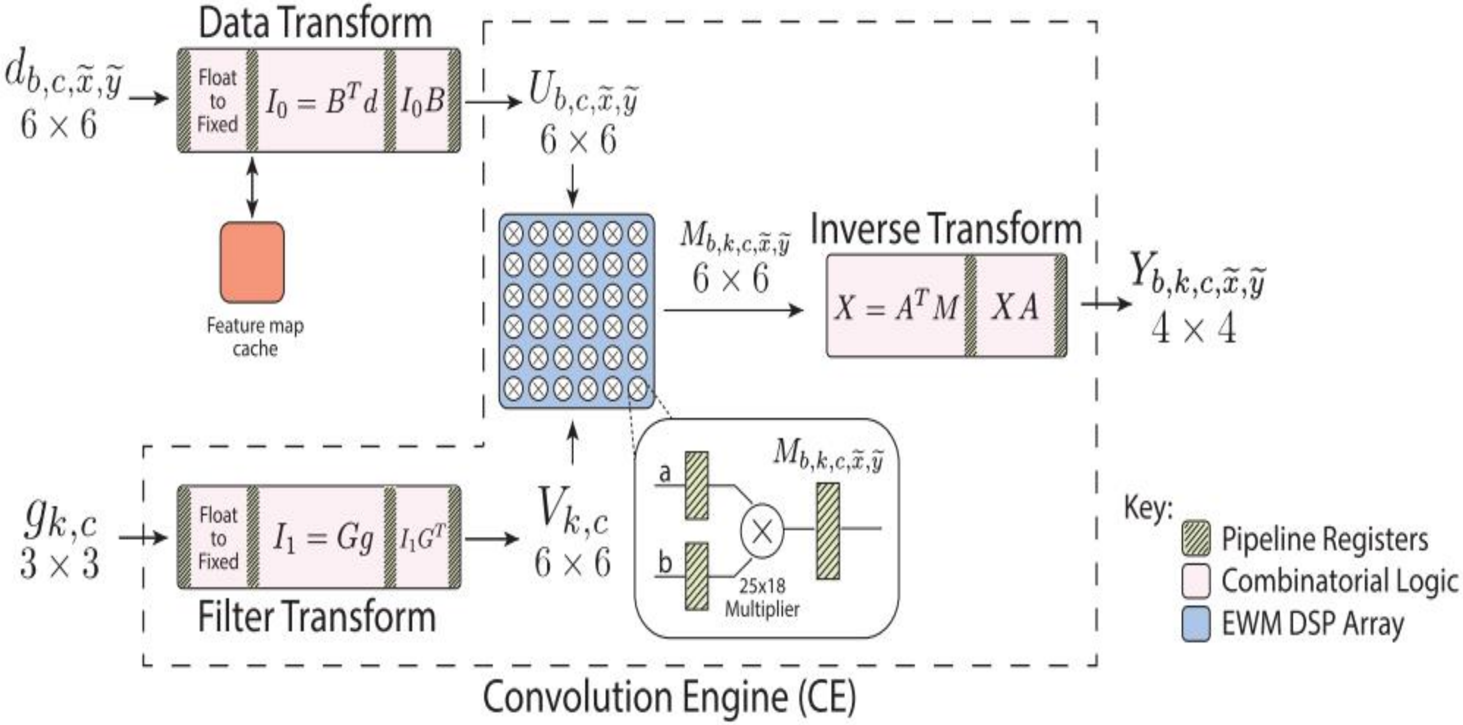
FFConv: An FPGA-based Accelerator for Fast Convolution Layers in Convolutional Neural Networks



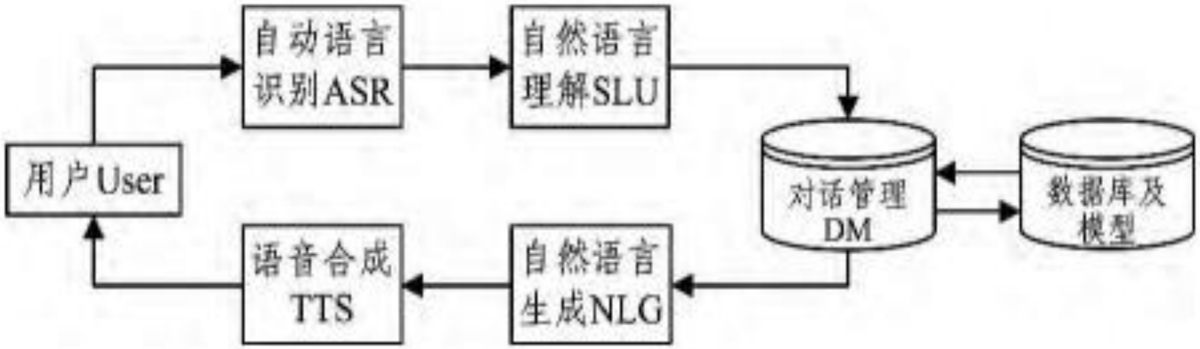
Feature Map, D
$C \times H \times W$
Filter Map, W
$K \times C \times r \times r$
Feature Map Tile, $D_{c,x,y}$
Filter Tile, $W_{k,c}$
MACC
$Y_{k,x,y} \leftarrow Y_{k,x,y} + (D_{c,x+u,y+v} \times W_{k,c,u,v})$
Output Pixel, $Y_{k,x,y}$
Feature Map Tile, $d_{b,c,\tilde{x},\tilde{y}}$
Feature Map, D
Filter Tile, $g_{k,c}$
Filter Map, W
Data Transform
$U_{b,c,\tilde{x},\tilde{y}} = B^T d_{b,c,\tilde{x},\tilde{y}} B$
Filter Transform
$V_{k,c} = G g_{k,c} G^T$
Element Wise Multiplication (EWM)
$M_{b,k,c,\tilde{x},\tilde{y}} = U_{b,c,\tilde{x},\tilde{y}} \odot V_{k,c}$
Inverse Transform
$Y_{b,k,c,\tilde{x},\tilde{y}} = A^T M_{b,k,c,\tilde{x},\tilde{y}} A$
Output Tile, $Y_{b,k,c,\tilde{x},\tilde{y}}$

$$\frac{m^2 r^2}{(m+r-1)^2} = \frac{2^2 \times 3^2}{(2+3-1)^2} = 2.25 \times .$$

F(4x4,3x3):卷积核大小为3x3，输出特征矩阵大小为4x4
则输入矩阵为6x6（4+3-1=6）



$$Y_{b,k,c,\tilde{x},\tilde{y}} = A^T[(B^T d_{b,c,\tilde{x},\tilde{y}} B) \odot (G g_{k,c} G^T)]A,$$



基于有限状态机
基于框架（填槽法）
基于实例
基于规划
基于贝叶斯网络
。。。。。。
基于强化学习：MDP，POMDP

下周计划

1.学习强化学习相关视频
深度强化学习，Q-learning，A3C等

2.完成GAN课后作业

3.继续阅读论文