Lab1-卷积神经网络

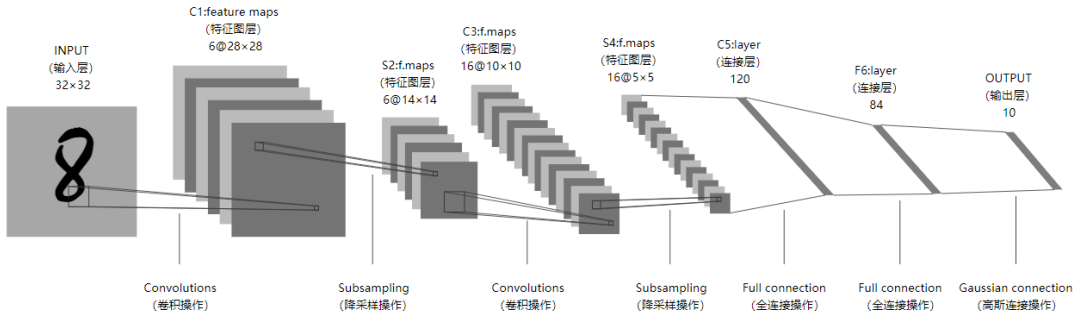
一、完成情况概述

尝试了 LeNet、AlexNet、ResNet18、MobileNet v1、VGG16 共 5 种神经网络模型，在 MNIST 数据集上训练并达到了 90%以上的准确率。

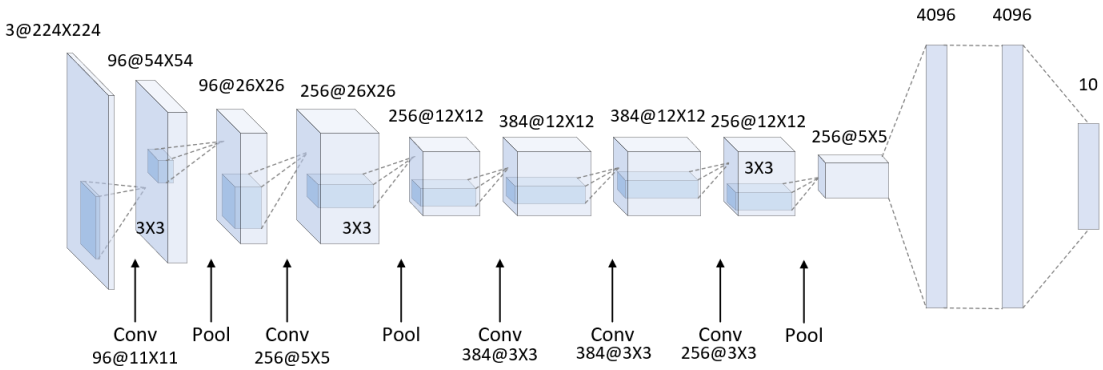
改进之处：将原来的带动量的 SGD 优化器改成了 AdaGrad，提升了模型的准确率，且在同样 epoch 次数下对比原来的模型具有更好的表现。

二、神经网络模型

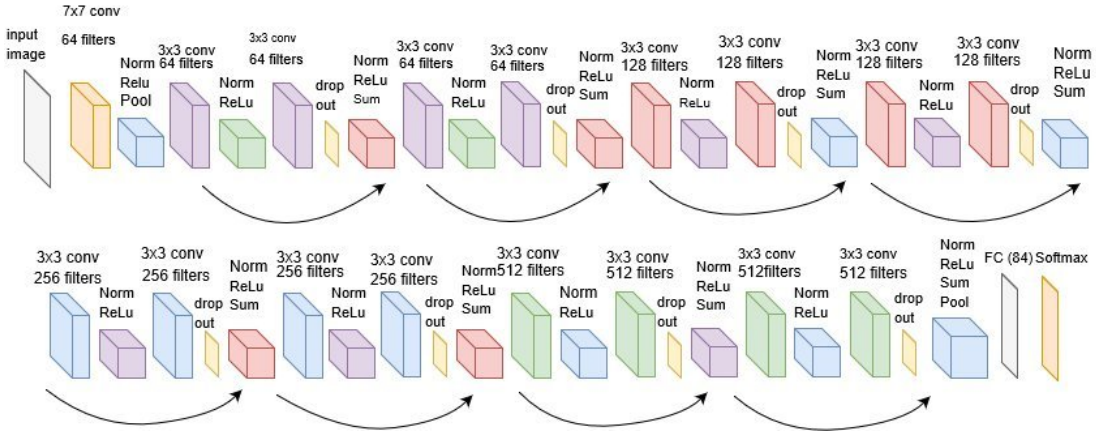
①LeNet



②AlexNet

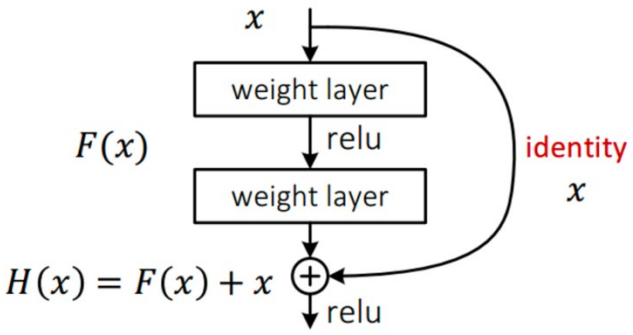


③ResNet18



ResNet 要解决的问题：随着网络深度的增加，模型精度并不总是提升，并且这个问题显然不是由过拟合（overfitting）造成的，因为网络加深后不仅测试误差变高了，它的训练误差竟然也变高了。

残差网络的提出者认为这可能是因为更深的网络会伴随梯度消失/爆炸问题，从而阻碍网络的收敛。虽然通过 Batch normalization 等方法，已经一定程度上缓解了这个问题，但依然不足以满足需求。于是残差网络的提出者想到了构建恒等映射（Identity mapping）来解决这个问题，即 ResNet 中的 shorcut 结构：

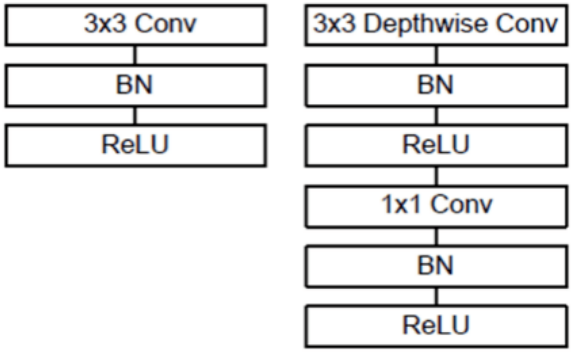


④MobileNet v1

Table 1. MobileNet Body Architecture

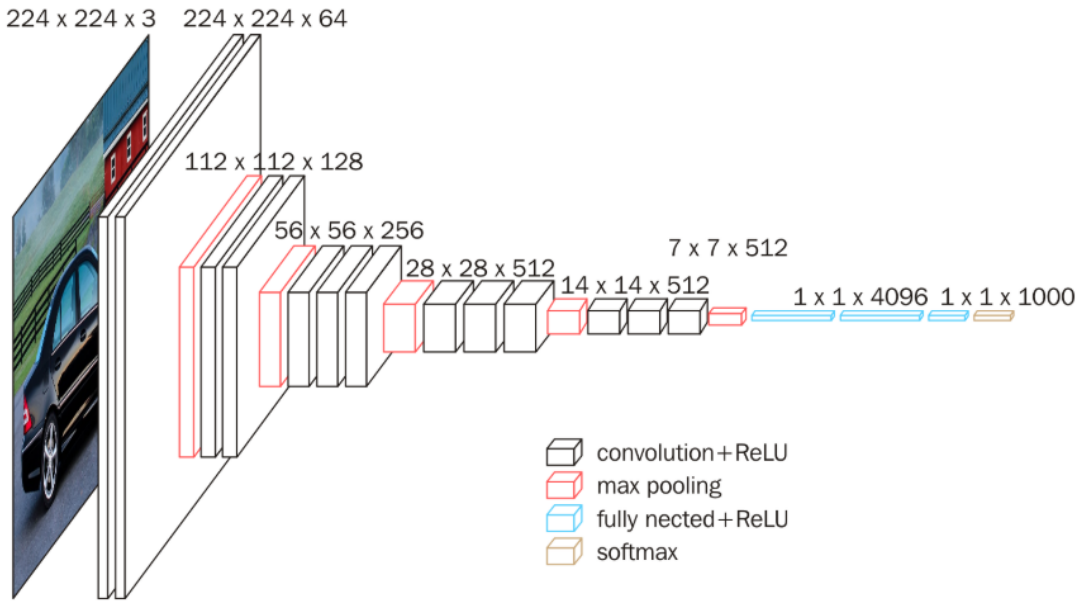
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

MobileNet v1 的核心思想是采用深度可分离卷积操作。在相同的权值参数数量的情况下，相较标准卷积操作，可以减少数倍的计算量，从而达到提升网络运算速度的目的。



深度可分离卷积就是将普通卷积拆分成为一个深度卷积和一个逐点卷积。如上图所示，一个常规的 3x3 Conv 被拆分成了两个部分： 3x3 Depthwise Conv（深度卷积层）和 1x1 Conv（逐点卷积层）。

⑤VGG16



VGG16 采用了固定 3*3 较小的卷积核，层数增加带来了更强的非线性，使模型的判别能力更强；而虽然层数增加，但较小的卷积核反而在卷积层减小了参数数量，与大卷积核相比相当于增加了正则化。

三、改进：优化器

①原来的优化器：带动量的 SGD

在随机梯度下降法的基础上，引入动量（Momentum）方法一方面是为了解决“峡谷”和“鞍点”问题；一方面也可以用于 SGD 加速，特别是针对高曲率、小幅但是方向一致的梯度。

算法描述：

Require：学习率ε，动量参数α

Require：初始参数θ，初始速度v

while 没有达到停止准则 do

从训练集中采包含m个样本$\{x^{(1)},\dots,x^{(m)}\}$的小批量，对应目标为$y^{(i)}$

计算梯度估计：$g \Leftarrow \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)};\theta),y^{(i)})$

计算速度更新：$v \Leftarrow \alpha v - \varepsilon g$

应用更新：$\theta \Leftarrow \theta + v$

end while

②改进之后：AdaGrad

AdaGrad 算法的思想是独立地适应模型的每个参数：具有较大偏导的参数相应有一个较大的学习率，而具有较小偏导的参数则对应一个较小的学习率。具体来说，每个参数的学习率会缩放各参数反比于其历史梯度平方值总和的平方根。

算法描述：

Require：全局学习率ε

Require：初始参数θ

Require：小常数δ，为了数值稳定大约设为10^{-7}

初始化梯度累计变量$r = 0$

while 没有达到停止准则 do

从训练集中采包含m个样本$\{x^{(1)}, \ldots, x^{(m)}\}$的小批量，对应目标为$y^{(i)}$

计算梯度：$g \Leftarrow \frac{1}{m} \nabla_\theta \sum_i L(f(x^{(i)}; \theta), y^{(i)})$

累计平方梯度：$r \Leftarrow r + g \odot g$

计算更新：$\Delta\theta \Leftarrow -\frac{\varepsilon}{\delta + \sqrt{r}} \odot \mathrm{g}$　　（逐元素地应用除和求平方根）

应用更新：$\theta \Leftarrow \theta + \Delta\theta$

end while

四、训练结果

①LeNet

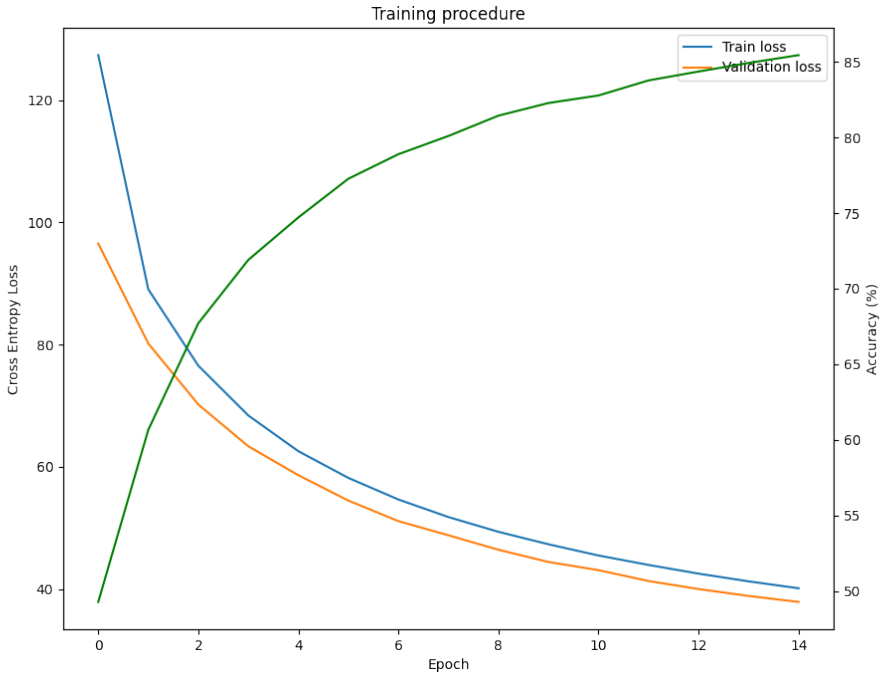


图 1-LeNet+带动量的 SGD

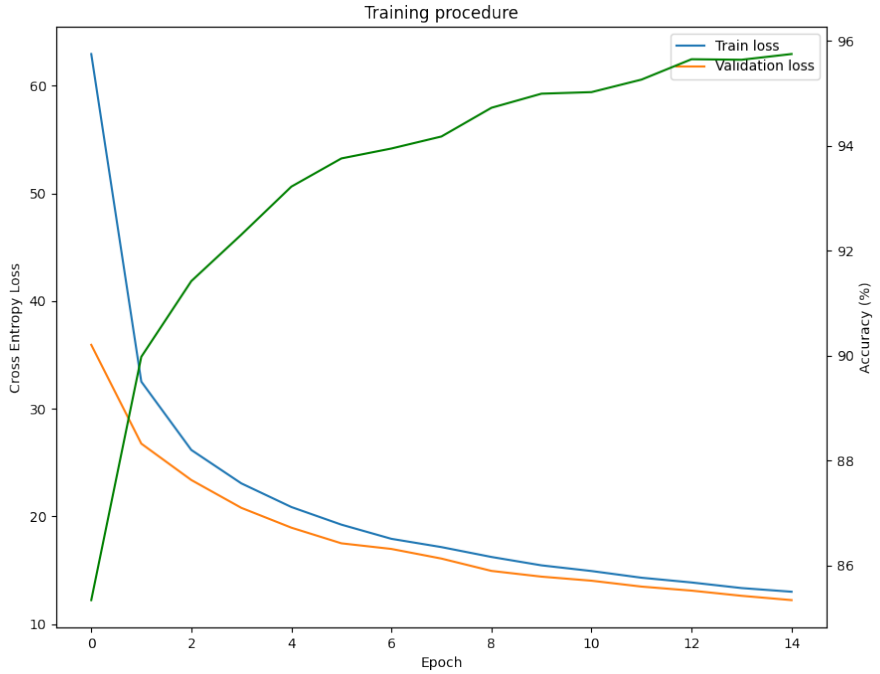


图 2- LeNet+AdaGrad

②AlexNet

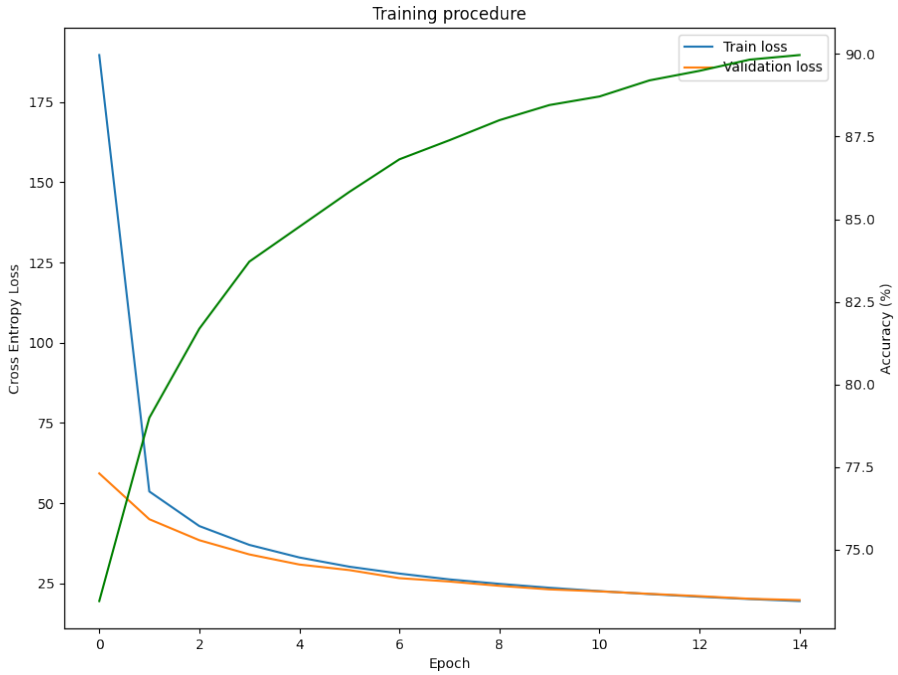


图 3-AlexNet+带动量的 SGD

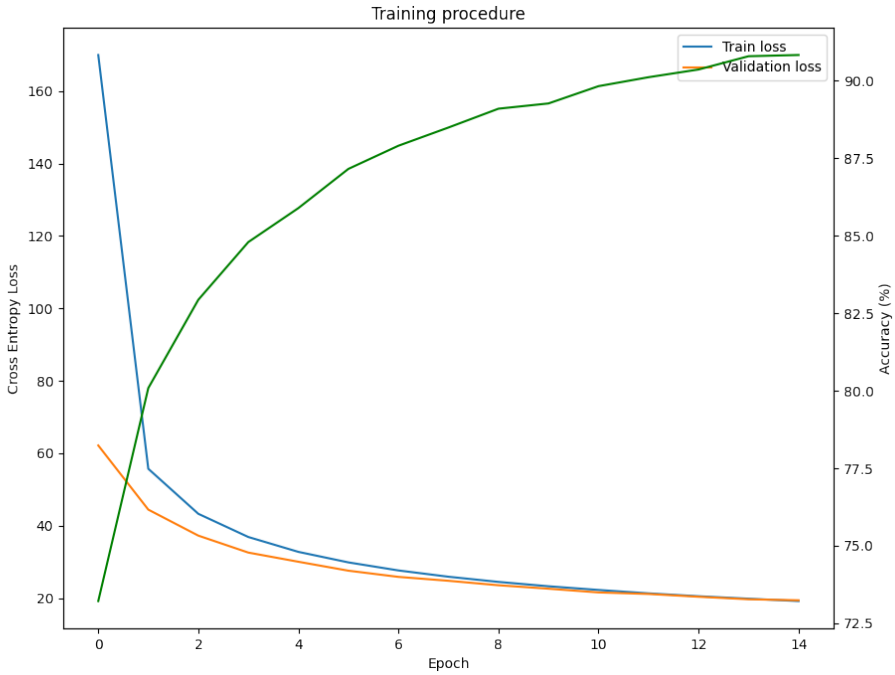


图 4-AlexNet+AdaGrad

③ResNet18

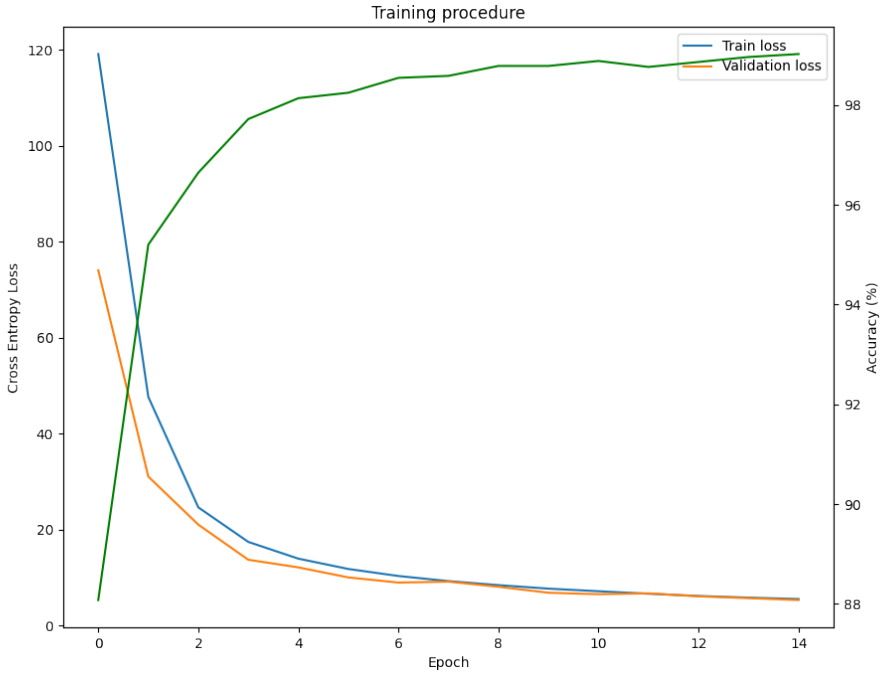


图 5-ResNet18+带动量的 SGD

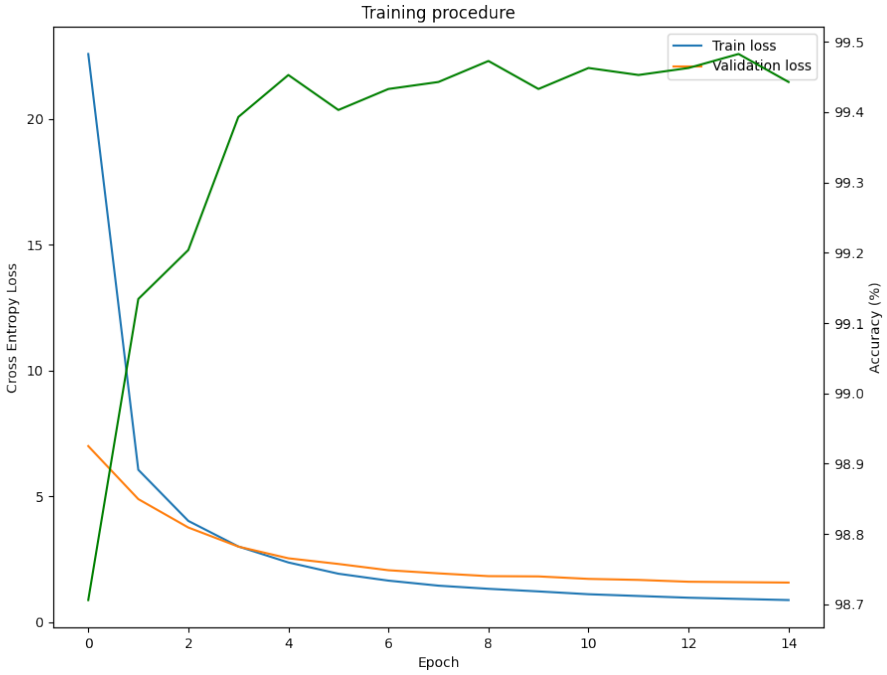


图 6-ResNet18+AdaGrad

④MobileNet

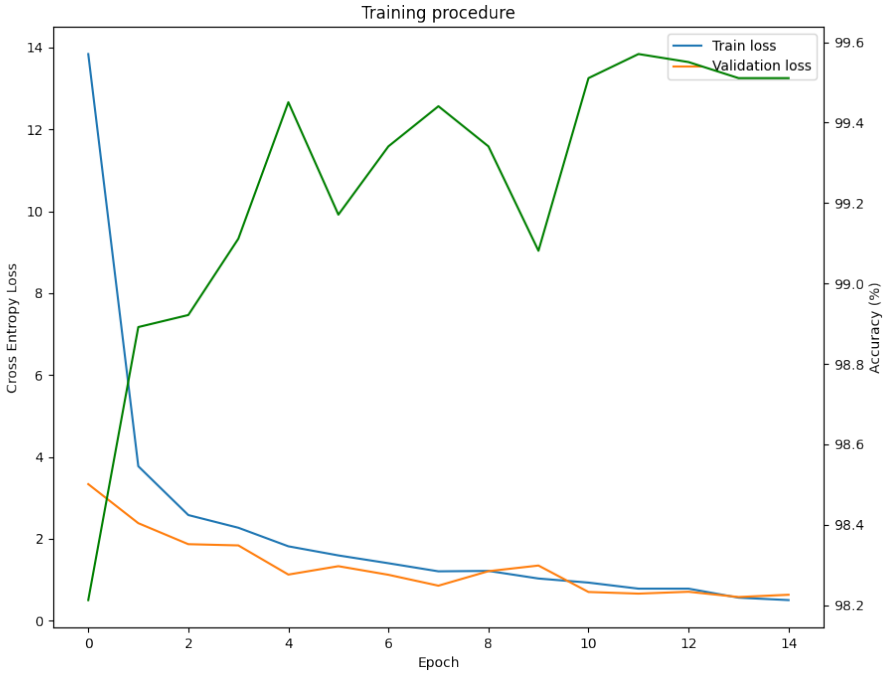


图 7-MobileNet+带动量的 SGD

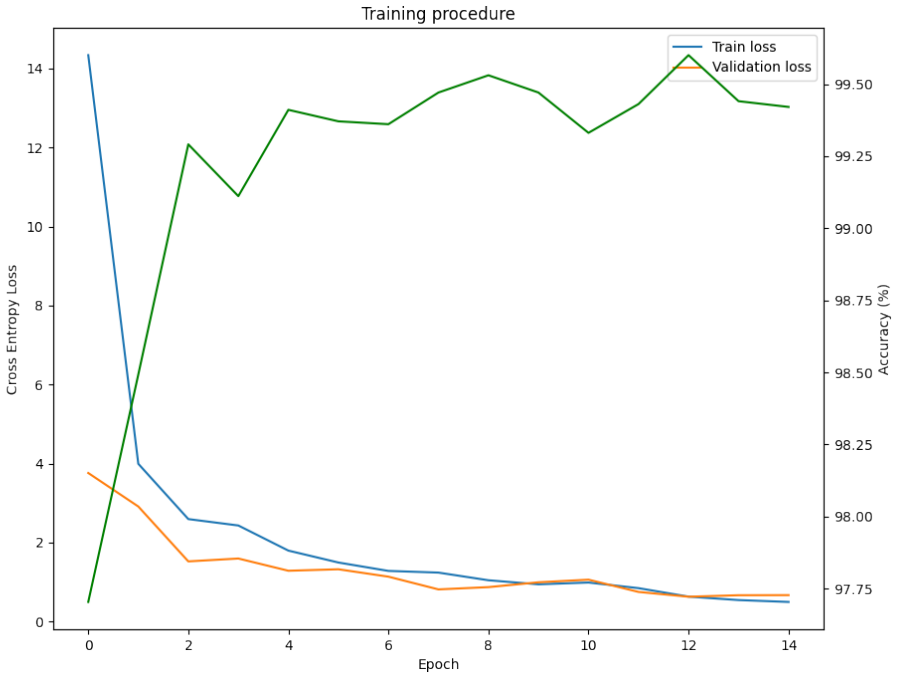


图 8-MobileNet+AdaGrad

⑤VGG16

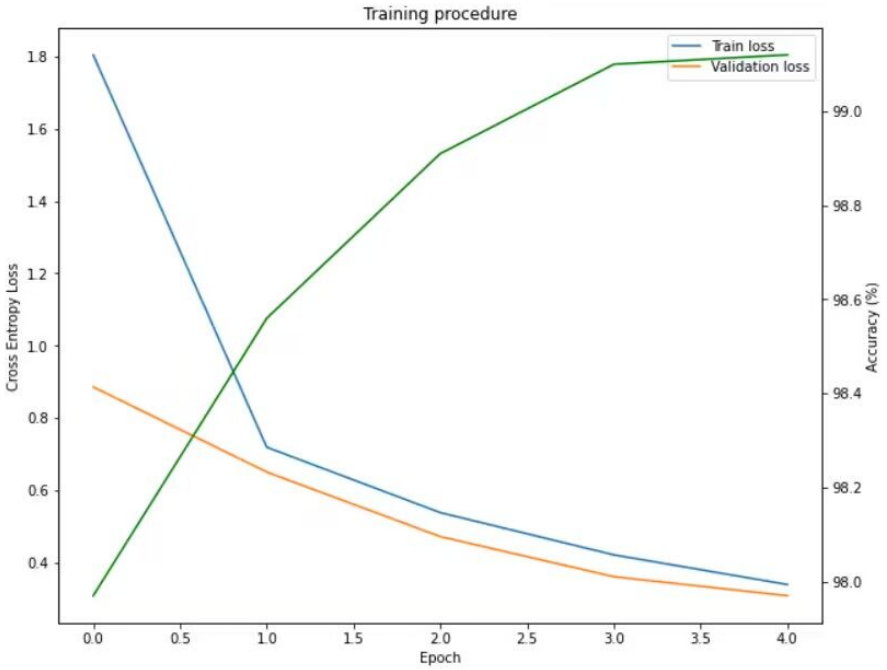


图 9-VGG16+带动量的 SGD

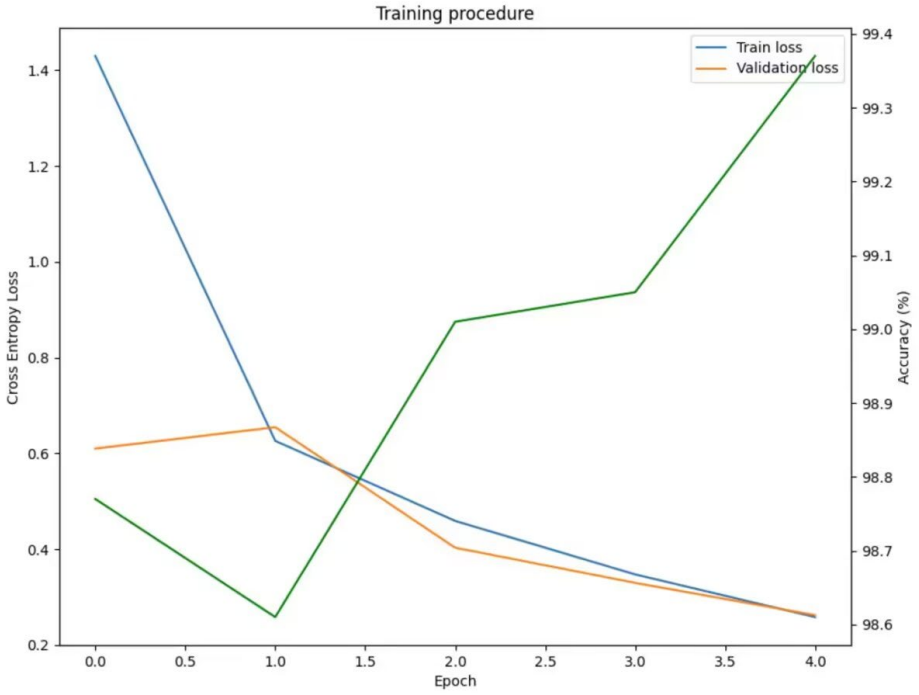


图 10-VGG16+AdaGrad