利用深度残差学习进行图像分类

伍炜臻
华南师范大学
mattwaizuen@outlook.com

摘要

更深的卷积神经网络会更难训练, 而采用改进的深度残差网络可以明显提高模型的表现. 我们搭建了用于图像分类的深度残差网络, 利用 CIFAR-10 数据集的 50,000 张图片来训练网络的参数, 并在测试数据上达到了 91% 的准确率.

1 引言

在近几年内深度卷积神经网络为图像分类领域带来了重大突破. 卷积神经网络最重要的两个特性是局部连接和权值共享. 因此, 和标准的前馈神经网络相比, 卷积神经网络具有非常少的连接和参数, 并且非常容易训练. 此外, 局部连接和权值共享分别满足了两个假设, 图像中距离近的像素之间的相关性远大于距离远的像素之间的相关性, 图像中的特征属性是和位置无关的.

卷积神经网络中的卷积层利用卷积核来从图像中提取特征, 而卷积核就是需要训练的参数. 一般观点认为更深的网络可以从图像中提取出更高层次的更抽象的特征, 因而拥有更好的泛化能力. 但在实践当中使用深层的卷积神经网络时, 由于信号的逐层衰减, 输入信息不能有效地从网络浅层传递到深层. 同样地在误差反向传播阶段中, 误差梯度无法有效地从深层传递到浅层. 因此网络的深层和浅层部分的的训练速率不一致, 容易导致深层部分发生过拟合而浅层的部分却训练不充分, 这种现象被称为梯度弥散 (gradient vanish)[1].

基于归一化的方法, 例如 Batch Normalization[5], 可以有效解决梯度弥散问题, 提高深层神经网络的表现, 加速训练的收敛速度. 但随着网络的加深, 模型退化的现象依然发生. 有研究在卷积神经网络的基础上加入了 “shortcut connections“结构, 提出了深度残差网络, 有效解决了网络过深导致模型退化的问题, 即使深达 1000 层的网络也能进行有效训练并收敛.

2 卷积神经网络的图像分类应用

在基于卷积神经网络的图像分类任务中, 模型通常是端对端 (end-to-end) 的, 即网络的输入就是图像本身或经过预处理的图像, 而输出就是图像属于各个类别的概率. 在这样的任务当中, 我们并不需要手工地设计特定的规则和特征提取器, 而是直接将图像以及标记类别的标签输入到网络中, 通过训练网络来得分类器. 经过端对端的训练后, 网络本身就包含了特征提取器和分类器.

在图像 $\boldsymbol{x}$ 可能属于 K 个类别的分类任务当中, 用于分类预测的卷积神经网络 f 可以表达为

$$\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\Theta}) \tag{1}$$

其中图像 $\boldsymbol{x}$ 是一个形状为 (height × width × channels) 的三阶张量, height 和 width 分别表示图像的行数和列数, channels 表示图像的颜色通道数, 例如 RGB 图片有 3 个颜色通道. 卷积神经网络的输出值 $\boldsymbol{y}$

是 K 维的概率向量, 其第 k 个元素 y_k 表示预测得图像 $\boldsymbol{x}$ 属于第 k 类的概率 $p(\boldsymbol{x} \in \mathcal{C}_k; \boldsymbol{\Theta})$. $\boldsymbol{\Theta}$ 表示卷积神经网络中所有可训练的参数, 包括权值和偏置等.

在预测得到 $\boldsymbol{x}$ 的类别概率向量 $\boldsymbol{y}$ 后, 求得

$$j = \arg\max_k y_k, \tag{2}$$

之后, 我们把 $\boldsymbol{x}$ 分到 $\mathcal{C}_j$ 类中.

2.1 网络参数的最大似然估计

为了方便表示, 我们用 one-hot 编码的类别向量 $\boldsymbol{t} = (t_1, \ldots, t_K)^T$ 来表示图像 $\boldsymbol{x}$ 所属类别, 满足

$$t_k = 1\{\boldsymbol{x} \in \mathcal{C}_k\} \tag{3}$$

其中 $1\{\cdot\}$ 为指示函数, $1\{True\} = 1, 1\{False\} = 0$.

对于图像 $\boldsymbol{x}$, 我们利用式 (1) 预测它属于各个类别的概率 $\boldsymbol{y}$, 在这个基础上 $\boldsymbol{x} \in \mathcal{C}_k$ 的条件概率为

$$p(\boldsymbol{x} \in \mathcal{C}_k | \boldsymbol{x}; \boldsymbol{\Theta}) = y_k, \quad k = 1, \ldots, K \tag{4}$$

由于

$$p(\boldsymbol{x} \in \mathcal{C}_k | \boldsymbol{x}; \boldsymbol{\Theta}) = p(t_k = 1 | \boldsymbol{x}; \boldsymbol{\Theta}) \tag{5}$$

类别向量 $\boldsymbol{t}$ 的条件分布为

$$p(\boldsymbol{t} | \boldsymbol{x}; \boldsymbol{\Theta}) = \prod_{k=1}^{K} y_k^{t_k} \tag{6}$$

对于一组已知类别的图像数据 $D = \{(\boldsymbol{x_1}, \boldsymbol{t_1}), \ldots, (\boldsymbol{x_M}, \boldsymbol{t_N})\}$, $\boldsymbol{x}$ 为图像, $\boldsymbol{t}$ 为类别标签. 我们希望用这些数据来训练网络, 可以直接写出似然函数

$$\begin{aligned} \mathcal{L}(\boldsymbol{\Theta}|D) &= \prod_{m=1}^{M} p(\boldsymbol{t_m} | \boldsymbol{x_m}; \boldsymbol{\theta}) \\ &= \prod_{m=1}^{M} \prod_{k=1}^{K} y_{mk}^{t_{mk}} \end{aligned} \tag{7}$$

取似然函数的负对数

$$-\log L(\boldsymbol{\Theta}|D) = -\sum_{m=1}^{M} \sum_{k=1}^{K} t_{mk} \log y_{mk} \tag{8}$$

我们得到被称为交叉熵的损失函数

$$E(\boldsymbol{\Theta}|D) = -\sum_{m=1}^{M} \sum_{k=1}^{K} t_{mk} \log y_{mk} \tag{9}$$

最大化似然函数等价于最小化交叉熵损失函数, 网络的训练目的就是通过最小化交叉熵损失函数来求解网络的所有可训练参数 $\boldsymbol{\Theta}$

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\Theta} E(\boldsymbol{\Theta}|D). \tag{10}$$

参数是通过基于梯度的优化方法来迭代求解的, 最常见的方法是梯度下降. 每次的迭代将依次进行以下三个步骤

- 前向传播, 利用当前的参数 $\boldsymbol{\Theta}$ 和数据计算交叉熵损失函数

$$E(\boldsymbol{\Theta}|D)$$

- 反向传播, 计算损失函数关于所有可训练参数 $\boldsymbol{\Theta}$ 的梯度

$$\frac{\partial E(\boldsymbol{\Theta}|D)}{\partial \boldsymbol{\Theta}}$$

- 梯度下降, 利用预先设定的学习速率 η 来更新参数

$$\boldsymbol{\Theta} := \boldsymbol{\Theta} - \eta \frac{\partial E(\boldsymbol{\Theta}|D)}{\partial \boldsymbol{\Theta}}$$

在适当的学习速率 η 下, 损失函数 E 总能收敛到某个局部极小值点. 实际我们并不需要求出全局最小值点, 因为这可能意味着模型过拟合.

2.2 卷积神经网络

到目前为止, 我们已经建立了模型的优化框架, 但并没有设计一个具体预测函数 f, 即卷积神经网络. 网络的结构不是唯一的, 可以根据需要来设计. 一个卷积神经网络是由不同结构的模块串联而成的, 一般包含卷积层, 降采样 (池化) 层, Batch Normalization, 激活函数, 输出层等

一个图像的特征信息在网络中有两种格式, 第一种格式是特征图 (feature maps), 是形状为 (height$\times$ width $\times$ channels) 的三阶张量, 每个 channel 就是一个特征图. 第二种格式是特征向量 (feature vector), 是一个向量, 向量的长度就是特征的维数. 特征图一般出现在卷积层和池化层, 特征向量一般出现在全连接层.

2.2.1 卷积层

描述图像的卷积层的时候, 我们需要先引入 2 维的离散卷积 $*$. 设 f, h 分别是数字图像和卷积核, 定义 $g = f * h$, 则图 g 在坐标 (x, y) 处的像素值为

$$g(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n) \tag{11}$$

现在考虑卷积神经网络中的卷积层 $l.$, 则 l 层中的每个特征图 $\boldsymbol{X}_j^l$ 是由卷积核 $\boldsymbol{k}_{ji}^l$ 对前一层的各个特征图 $\boldsymbol{X}_i^{l-1}$ 卷积再加上偏置项 b_j^l 得到的

$$\boldsymbol{X}_j^l = \sum_{i=1}^{D_{l-1}} \boldsymbol{X}_i^{l-1} * \boldsymbol{k}_{ji}^l + b_j^l, \quad j = 1, \ldots, D_l \tag{12}$$

其中卷积核 $\boldsymbol{k}$ 和偏置项 b 为可训练的参数. 第 $l-1$ 层和 l 层分别有 D_{l-1} 和 D_l 个特征图, 这是由超参数控制的. 此外, 卷积运算 $*$ 还有多种变体, 此处不展开描述.

2.2.2 降采样层

降采样层也称为池化层, 其作用是在尽可能保留特征信息的前提下减小特征图的尺寸, 从而较少网络的运算量, 提高特征的抽象性. 降采样层的每个特征图对应前一层的一个特征图, 降采样层的每个单元从前一层相应特征图中的一个对应的接收区域中计算采样值. 采样的方式包含最大采样, 平均采样和按概率采样, 前两种是最常见的. 接收区域也称为采样窗. 相邻采样窗的间隔称为采样步长. 采样层中特征图的计算可以表示为

$$\boldsymbol{X}_j^l = down(\boldsymbol{X}_{\boldsymbol{j}}^{l-1}), \quad j = 1, \ldots, D_{l-1} \tag{13}$$

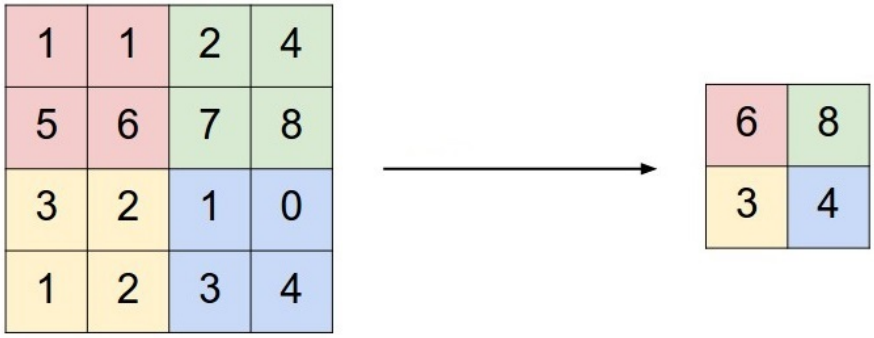

图 1: 最大采样, 采样窗大小为 2×2, 采样步长为 2×2.

2.2.3 全连接层

全连接层 l 的每个单元与前一层的所有单元是完全连接的, 这和普通的前馈神经网络是一致的. 表示为

$$x_j^l = \sum_{i=1}^{D_{l-1}} w_{ji}^l x_i^{l-1} + b_j, \quad j = 1, \ldots, D_l \tag{14}$$

2.2.4 激活函数

激活函数层 l 的作用是对前一层的每个单元进行非线性变换, 这个变换被称为激活函数 g. 为了方便表示, 我们把激活函数 g 扩展为向量形式

$$g\left(\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}\right) = \begin{pmatrix} g(x_{11}) & \cdots & g(x_{1n}) \\ \vdots & & \vdots \\ g(x_{m1}) & \cdots & g(x_{mn}) \end{pmatrix}. \tag{15}$$

激活函数层对前一层的特征图的激活可以表示为

$$\boldsymbol{X}_j^l = g(\boldsymbol{X}_j^{l-1}), \quad j = 1, \ldots, D_{l-1} \tag{16}$$

常见的激活函数有

$$ReLu(x) = \max(0, x) \tag{17}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{18}$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{19}$$

其中 Rectified Linear Units(ReLu) 是深度学习领域中最常用的激活函数, 因为它不会饱和, 当输入值大于 0 时, 导函数始终为 1, 这不仅能有效缓解梯度弥散问题, 而且能加速训练的收敛.

2.2.5 输出层

在分类任务中, 假设类别数量为 K, 通常会在卷积神经网络的最后一层 L 设置为含有 K 个单元的全连接层, 每个单元的输出值 $x_j^L, j = 1, \ldots, K$ 表示相应类别的得分, 得分的相对大小反应了图像属于各个类别的可能性. 通常我们在最后的这层加入一个 softmax 函数, 将全连接层的 K 个输出值转换为概率值

$$y_j = \frac{e^{x_j^L}}{\sum_{k=1}^K e^{x_k^L}}, \quad j = 1, \ldots, K, \tag{20}$$

向量化的表示为

$$\boldsymbol{y} = \frac{1}{\sum_{k=1}^K e^{x_k^L}} \begin{pmatrix} e^{x_1^L} \\ \vdots \\ e^{x_K^L} \end{pmatrix} \tag{21}$$

概率向量 $\boldsymbol{y}$ 就是网络的输出值.

除了上述的模块, 本文不再对 Batch Normalization, Locally connected 等模块作描述. 一个卷积神经网络就由这些模块按照一定的顺序串联得到的. 图 2是 LeNet5[8] 的结构示意图. 可以看到, 卷积神经网络将原始图像的像素值逐层地进行变换, 最终得到类别的概率值.

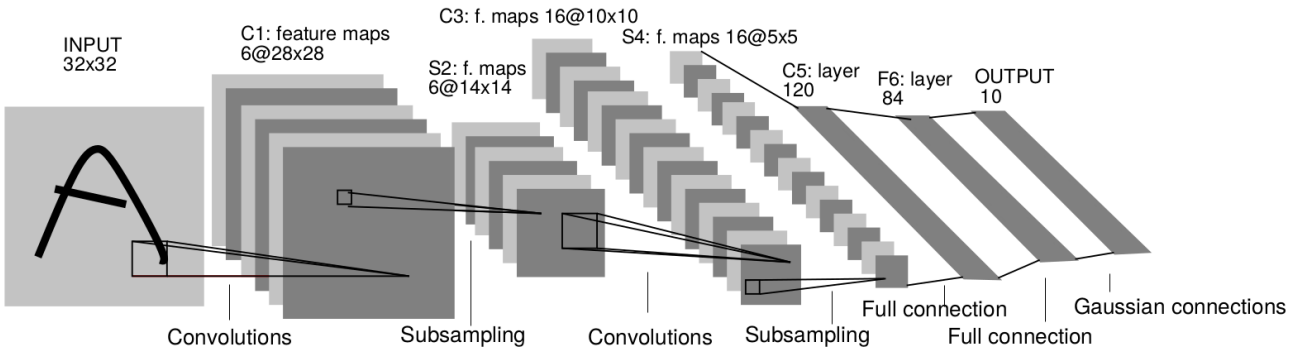

图 2: 用于手写数字分类卷积神经网络 LeNet5 的结构示意图.

2.3 残差网络

我们知道, 典型卷积神经网络通常是多个模块按照一定的顺序串联起来的, 这样看起来就像一个扁平的网络, 代表有 AlexNet 和 VGG[7, 10]. 图 3 (a) 是这种扁平网络中的一部分. 深度残差网络 [3, 4]

是在网络中特定位置加入了 shortcut connection 结构, 这可以让信号更容易地在网络中传播. 深度残差网络中的基本模块是残差单元, 其中卷积核的大小都是 3×3 的. 图 3 (b) 是残差单元的结构图, 一个深度残差网络可能包含数十上百个这样的残差单元. 一个残差单元的前向传播可以表示为

$$\boldsymbol{Y} = \mathcal{F}(\boldsymbol{X}, \boldsymbol{W}) + \boldsymbol{X} \tag{22}$$

其中 $\mathcal{F}(\boldsymbol{X}, \boldsymbol{W})$ 表示信号 $\boldsymbol{X}$ 沿着主路径传播的输出值, $\boldsymbol{W}$ 表示主路径中的权值参数. $\mathcal{F}(\boldsymbol{X}, \boldsymbol{W})$ 再加上捷径的信号 $\boldsymbol{X}$, 就得到这个残差单元的输出信号 $\boldsymbol{Y}$.

有时候残差单元的输入和输出的维数不一致, 例如特征图的尺寸或通道数不同. 这导致 $\mathcal{F}(\boldsymbol{X}, \boldsymbol{W})$ 和 $\boldsymbol{X}$ 不能直接相加. 当特征图尺寸不相同时, 可以在捷径中设置一个采样层来对 $\boldsymbol{X}$ 进行采样, 从而调整捷径输出的特征图大小. 当特征图通道数不同的时候, 可以在捷径中设置一个 1×1 的卷积层来对 $\boldsymbol{X}$ 进行卷积, 从而改变捷径输出的特征图通道数, 可以表达为

$$\boldsymbol{Y} = \mathcal{F}(\boldsymbol{X}, \boldsymbol{W}_1) + h(\boldsymbol{X}, \boldsymbol{W}_2) \tag{23}$$

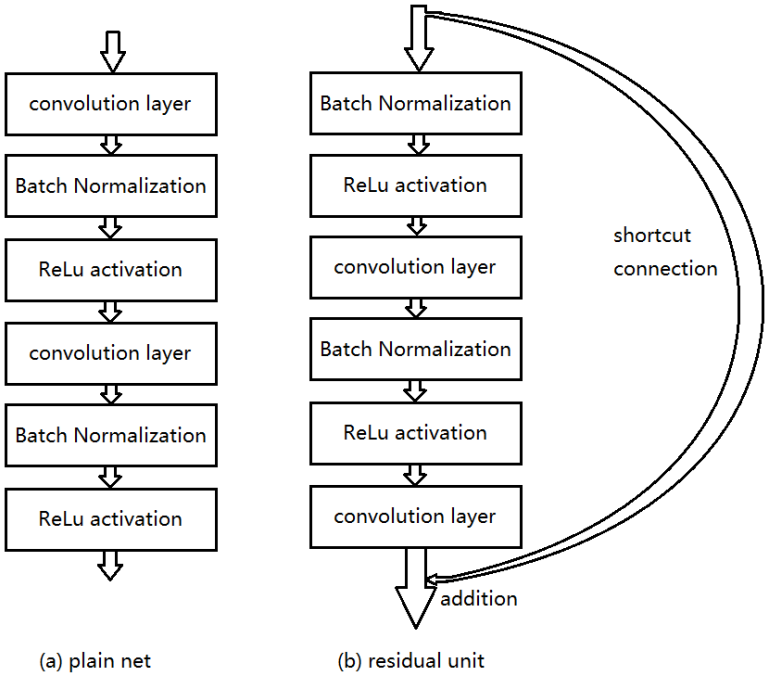

图 3: (a) 是典型的卷积神经网络中的一部分, 信号从浅层到深层逐层传播. (b) 是一个残差单元, 信号沿着主路径逐层传播的同时还会通过捷径传播. 深度残差网络通常可以包含数十上百个这样的残差单元.

3 实验

3.1 CIFAR-10 图像数据集

我们的实验是在 CIFAR-10 图像数据集 [6] 上进行, 这个数据集包含 50,000 张用于训练的图片和 10,000 张用于测试的图片. 每张图片为 $32 \times 32 \times 3$ 的 RGB 格式彩色图片, 并且属于十个类别中的一类. 在搭建用于分类的深度残差网络后, 我们训练数据来训练网络的参数, 然后用测试数据来测试网络的准确率.

3.2 网络结构和训练策略

网络依次由 1 个卷积层和 15 个残差单元, 其中每 5 个残差单元构成一组, 3 组单元输出的特征图的形状分别为 $(32 \times 32 \times 16)$, $(16 \times 16 \times 32)$ 和 $(8 \times 8 \times 64)$, 随后是一个全局平均池化层 (global average pooling)[9], 然后是一层 10 个单元的全连接层, 最后添加了一个 softmax 函数作为输出. 详细的结构见图 4. 此外需要注意的是, 网络的第一个残差单元中, 在捷径的路径上还设置了一个 Batch Normalization 层和 ReLu 层.

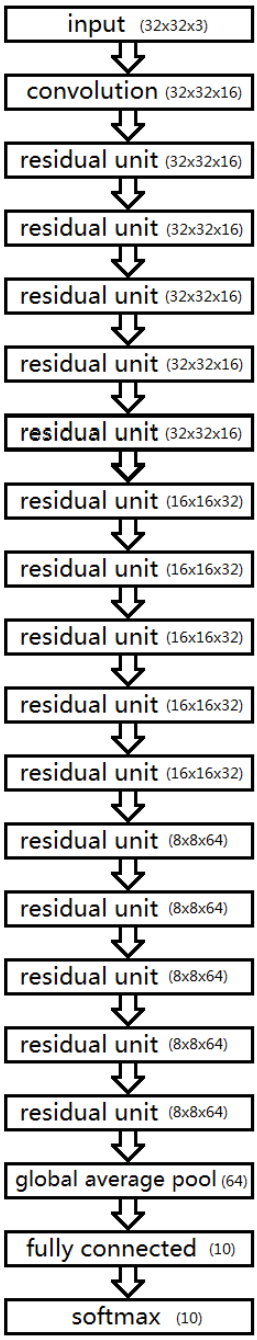

图 4: 本文中采用的网络的结构图. 在第一个残差单元中, 在捷径的路径上还设置了一个 Batch Normalization 层和 ReLu 层.

在训练阶段中, 首先使用 [2] 中的方法来初始化网络参数. 每次迭代的 batch-size 为 128, 采用系数为 0.9 的 Momentum 梯度优化. $L2$ 正则化系数为 0.0001. 在前 500 次迭代的时候使用 0.01 的学习速率来进行预热, 随后将学习速率调高到 0.1. 当训练达到第 32,000 和 48,000 和 64,000 次迭代的时候将学习速率除以 10. 训练在第 80,000 次迭代后停止. 数据增加的具体操作是, 在图像四周分别补上 4 个像素宽度的 0 像素, 使图像的形状变为 $(36 \times 36 \times 3)$, 然后随机裁剪成 $(32 \times 32 \times 3)$, 再经过 50% 概率的水平翻转后就得到增加后的图像. 由于像素的取值范围是 0 至 255, 这里的预处理方法是将每个像素乘以 2 后除以 255, 最后再减去 1, 这使得预处理后像素值的取值范围是 -1 至 1.

在测试阶段中, 不需要对数据进行补零和随机裁剪, 但需要进行相同的归一化预处理.

3.3 结果

使用 Intel i5 - 7200u cpu 和 NVIDIA GeForce 940mx gpu 进行计算, 训练经过 20 小时后达到迭代次数上限. 最后交叉熵损失函数值约为 0.015. 网络在测试数据上的预测准确率为 91.1%.

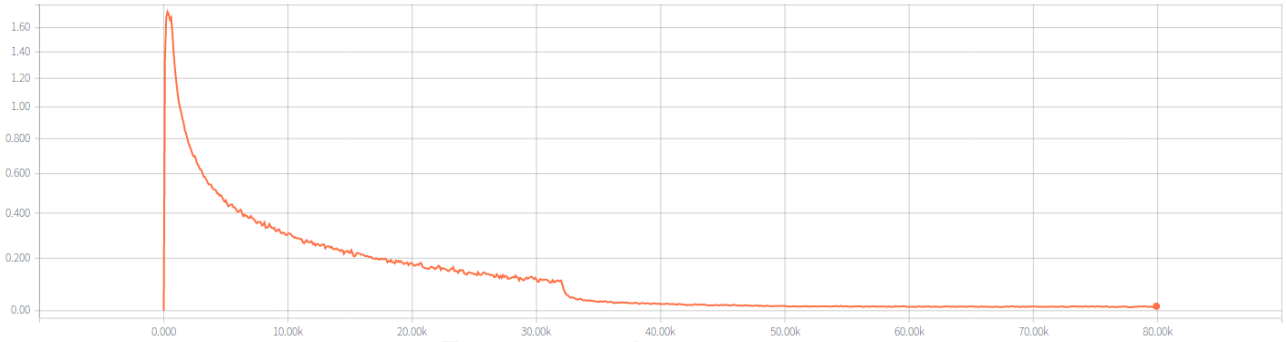

图 5: 随着训练迭代次数增长, 交叉熵损失函数的变化曲线.

参考文献

[1] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[6] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[8] Yann Lecun, L Eon Bottou, Yoshua Bengio, and Patrick Haaner. Gradient-based learning applied to document recognition. 1998.

[9] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.