神经网络和层数理解手册

1. 神经网络的理解和使用 ...................................................................................................2二. 神经网络结合实例的落实和理解 ....................................................................................4

三. 神经网络模型框架介绍 ......................................................................................................5

四. 海思平台优化细则介绍 ....................................................................................................11

一.神经网络的理解和使用

别人的经验篇：

把神经网络“当成一个万能函数拟合器。

把卷积理解为滤波和特征提取。

把全连接层理解为加权求和。

把反向传播理解为从最后一层逐步调整权重参数。

把设计各种模型当成搭积木

了解这些就入门了，再

多看论文多思考，发篇论文毕业足够了

自主理解和思考：

1. 神经网络结合实例的落实和理解

这里我们使用的主要是将yolov5移植到海思平台下的一些实例，是基于有限硬件资源和算力下的优化方案和理解。

1.首先，确认我们主要移植需要实现的事情其实就是模型转化：

.pt(初始yolov5产生)->.onnx（转化成onnx模型）->caffemodel&.prototxt（转化成caffe模型产生）->.WK（使用NNIE工具将caffe转化成.wk产生）

然后，我们需要基于实现的算法功能进行实际的优化，这里需要我们了解一些算法的基本框架，文件结构等基础知识，在下面会讲到，最后我们需要根据HiSVP指南中的指导原则进行模型的优化和部署从而实现整个功能。

关于整个环境搭建使用到的工具和资源，我的安排如下，文档直接上传github，文件资源上传到百度网盘进行存储备份

1. 关于模型转化的联系和细节陈述

.pt（PyTorch 模型文件）：

初始模型文件：YOLOv5 通常在 PyTorch 中开发和训练，因此原始文件格式是 .pt，包含网络结构和训练后的权重参数。

初步调优：在训练 YOLOv5 时，超参数调优（例如学习率、batch size 等）和模型结构调整通常在 .pt 文件中完成。

.onnx（Open Neural Network Exchange 格式）：

标准化中间模型格式：将 .pt 模型转换为 .onnx，提供一个开放且兼容性更强的格式，用于在不同框架和硬件上部署。

连接其他框架：通过 .onnx 文件可以更方便地转换为其他框架，例如 Caffe、TensorFlow 等，使模型兼容更广泛的硬件平台。

调优与优化：在 .onnx 阶段，可以使用一些框架（如 ONNX Runtime）优化模型性能。

.caffemodel 和 .prototxt（Caffe 格式）：

转换为 Caffe 支持的格式：使用转换工具将 .onnx 模型转为 .caffemodel 和 .prototxt 文件。.prototxt 定义网络结构，.caffemodel 存储训练好的权重。

调优：在 Caffe 中，我们可以通过修改 .prototxt 文件对层次结构进行优化，同时在 .caffemodel 中保留优化的权重。

可视化和验证：通过 Netron 等工具可以验证 Caffe 模型结构和参数，确保转换正确。

.wk（NNIE 工具生成的模型文件）：

用于硬件加速：将 Caffe 模型转换为 .wk 格式，使模型能够在特定硬件加速器（如华为的 NNIE 上）高效运行。

优化配置：NNIE 的模型转换工具会对 .prototxt 和 .caffemodel 进行进一步优化，以提高性能。

特定硬件上的调优：在生成 .wk 文件的过程中，NNIE 工具会对模型执行量化和加速优化，确保在目标设备上运行效率最高。

文件和框架之间的关系

转换链：.pt -> .onnx -> .caffemodel/.prototxt -> .wk 是模型从开发到部署的转换链，保证模型在不同框架和硬件上的兼容性。

调优链：在每个转换阶段可以进行不同的优化。比如在 PyTorch (.pt) 阶段调优网络结构和训练参数，在 ONNX 阶段可以进行跨框架优化，在 Caffe (.prototxt 和 .caffemodel) 阶段调优具体层次配置，最后在 .wk 阶段针对硬件执行量化和加速优化。

用自己的话说其实就是：.pt在pytorch深度学习框架进行训练，使用yolov5对象检测模型，onnx其实就是一个更开放的模型格式，适合用于不同框架之间的格式转化，caffe是一个深度学习框架，.wk实际部署在硬件的模型文件，只适配于海思平台的文件。转化成caffe纯属是因为对caffe支持性更好

后续补充：

1. 神经网络模型框架介绍

Yolov5介绍

YOLOv5 是 You Only Look Once（YOLO）系列中的第五个版本，是一个用于实时对象检测的 模型架构。YOLOv5 主要用于图像和视频中的对象检测，广泛应用于自动驾驶、安防监控、智能零售等领域。

模型框架：YOLOv5 是在 PyTorch 框架上开发和训练的，支持灵活的训练和模型优化。

特点：速度与精度：YOLOv5 在保持高精度的同时具有较快的推理速度，非常适合实时检测应用。

模块化和可扩展：支持多种模型大小（如 YOLOv5s、YOLOv5m 等），可以根据不同应用场景选择合适的模型，适应资源紧张的嵌入式平台或高性能计算环境。

应用：通过转换为其他格式（如 ONNX、Caffe），可以在不同的硬件和框架上进行部署。YOLOv5 提供的 .pt 文件可以用于模型的训练、推理以及导出。

用自己的话说：其实就是一个实时对象检测的模型

onnx介绍

ONNX（Open Neural Network Exchange） 是一种 开放的模型文件格式，主要用于在不同的深度学习框架间进行模型的互操作和迁移。

模型格式：ONNX 是一种 中间模型格式，支持在多种框架（如 PyTorch、TensorFlow、Caffe2）之间转换模型，方便不同框架或平台的使用。

特点：开放和标准化：ONNX 使得模型的跨框架迁移更加方便，无需从头重写或重新训练模型。

硬件兼容性：ONNX 支持多种硬件加速器，便于在嵌入式设备、服务器、移动设备等不同硬件上进行推理。

广泛支持：ONNX 被许多深度学习框架和硬件供应商支持，具有良好的可扩展性和兼容性。

应用：ONNX 可以用于导出 YOLOv5 等模型，然后在不同平台上推理。ONNX 模型常被进一步转换为 Caffe 模型或直接部署在支持 ONNX 的框架上。

ONNX（Open Neural Network Exchange） 是一种 开放的模型文件格式，主要用于在不同的深度学习框架间进行模型的互操作和迁移。

模型格式：ONNX 是一种 中间模型格式，支持在多种框架（如 PyTorch、TensorFlow、Caffe2）之间转换模型，方便不同框架或平台的使用。

特点：开放和标准化：ONNX 使得模型的跨框架迁移更加方便，无需从头重写或重新训练模型。

硬件兼容性：ONNX 支持多种硬件加速器，便于在嵌入式设备、服务器、移动设备等不同硬件上进行推理。

广泛支持：ONNX 被许多深度学习框架和硬件供应商支持，具有良好的可扩展性和兼容性。

应用：ONNX 可以用于导出 YOLOv5 等模型，然后在不同平台上推理。ONNX 模型常被进一步转换为 Caffe 模型或直接部署在支持 ONNX 的框架上。

用自己的话说其实就是工具人转化格式，被很多深度学习框架支持

Caffe介绍

由caffemodel和.prototxt组成

我们在使用nnie工具时会用到这两个文件

1.Prototxt文件介绍

关于prototxt，其实就是基于文本的网络结构和参数的配置文件

他的基本文件结构如下：  
在 .prototxt 文件中，每一层的配置由 layer 块定义，每个 layer 块有以下主要字段：

name：层的名称，用于标识该层。

type：层的类型，例如 Convolution、Pooling、ReLU、FullyConnected 等，表示该层的具体操作类型。

bottom：当前层的输入，引用前一层的 top（输出）。可以有多个。

top：当前层的输出，其他层可以通过 bottom 引用。也可以有多个。

param：包含该层的参数（如卷积核数、步幅等），不同层会有不同的参数字段。

实例：

layer {

name: "conv1"

type: "Convolution" （就是这里对应的type）

bottom: "data"

top: "conv1"

convolution\_param {

num\_output: 32

kernel\_size: 3

stride: 1

pad: 1

}

}

在 Caffe 网络定义中（也就是.prototxt），每一层的 top 和 bottom 参数确实用于定义层与层之间的连接关系

用自己的话讲就是：其实这个层的计算就是根据这个top和bottom来定的是吗？ 比如说我的top是279，下一层接收bottom是279就能匹配上并进行接收，就是按照这个东西来进行计算的

关于各个层的介绍：

· **Input**：输入层，定义输入数据的形状。

· **Convolution**：卷积层，用于提取特征。

· **ReLU**：激活层，使用 ReLU 激活函数。

· **Pooling**：池化层，用于下采样。

· **FullyConnected**（InnerProduct）：全连接层，用于分类或回归任务的输出。

· **Softmax**：Softmax 层，通常用于分类问题的输出。

还有一个Reshape 层，他的作用是：

调整特征图形状：让上一层的输出满足检测层或其他下游层的输入要求。

格式转换：将特征图转换为适合目标检测输出的形状，便于解析检测框的输出。

这里需要注意，我们实际使用中出现问题就是这个层，他要求上一层和下一层输入输出元素是一致的，也就是其中的：  
layer {

name: "Reshape\_151"

type: "Reshape"

bottom: "265"

top: "277"

reshape\_param {

shape {

dim: 0

dim: 3

dim: 9

dim: 1600

}

}

}

这个dim的值要求上一层top（279）和这一层bottom(279)的输入输出计算一致，也就是需要修改这个dim值

1. caffemodel文件介绍

.caffemodel 文件 是 Caffe 的模型文件，它存储了经过训练的网络的 权重和偏置 参数。与 .prototxt 文件不同，.prototxt 文件只是定义网络结构，.caffemodel 文件才包含了实际的模型数据（即训练好的参数），因此是模型部署的核心文件之一。

.caffemodel 文件存储了：

卷积层、全连接层等的权重：这些是模型在训练过程中学习到的参数。

偏置值：对应层的偏置参数，结合权重一起用于神经网络推理。

在推理时，Caffe 加载 .prototxt 文件定义的网络结构，然后用 .caffemodel 文件加载训练好的参数，从而进行前向计算和预测。

用自己话来说就是：他是核心模型文件，用于参数的存储和实际执行的主程序，而.prototxt则像是.ini文件用于加载层的计算顺序

1. 关于模型性能查看和优化（后面再说，记着先）

性能调优与可视化工具

可视化工具：许多深度学习框架支持模型的性能调优和可视化，包括：Netron：一个开源的模型可视化工具，支持 .onnx、.caffemodel 和 .prototxt 文件。可以用来检查网络层级关系、参数设置和维度。

TensorBoard 或其他第三方工具：虽然 TensorBoard 本身是为 TensorFlow 设计的，但一些转换工具可以使其他框架兼容，帮助我们查看 Caffe 网络的图表和层输出。

调优方法：Caffe 中的 .prototxt 文件用于网络结构定义，可以调优模型的层数、卷积核大小、激活函数等；.caffemodel 文件包含训练好的权重，需要在训练完成后通过更改网络架构或模型参数进行微调。

性能调优与可视化工具

可视化工具：许多深度学习框架支持模型的性能调优和可视化，包括：Netron：一个开源的模型可视化工具，支持 .onnx、.caffemodel 和 .prototxt 文件。可以用来检查网络层级关系、参数设置和维度。

TensorBoard 或其他第三方工具：虽然 TensorBoard 本身是为 TensorFlow 设计的，但一些转换工具可以使其他框架兼容，帮助我们查看 Caffe 网络的图表和层输出。

调优方法：Caffe 中的 .prototxt 文件用于网络结构定义，可以调优模型的层数、卷积核大小、激活函数等；.caffemodel 文件包含训练好的权重，需要在训练完成后通过更改网络架构或模型参数进行微调。

WK介绍

.wk 文件 是基于嵌入式硬件的优化模型文件，通常用于 NNIE 或其他硬件加速器上进行推理。这是一个在嵌入式平台上部署的二进制模型文件格式。

文件格式：.wk 是为嵌入式硬件（如华为的 HiSilicon 芯片）量身定制的模型文件格式，包含了模型的推理权重和架构信息。

特点：硬件加速优化：.wk 文件格式经过优化，适合在特定硬件上高效推理，充分利用硬件的并行计算能力。

定制化：生成 .wk 文件的过程会依据硬件的特性对模型进行定制化优化，以适配设备资源限制并提高推理速度。

生成与应用：通常通过工具链（如 NNIE Mapper）从 Caffe 模型（.caffemodel 和 .prototxt 文件）生成 .wk 文件，之后可以直接部署在硬件上执行推理任务。

.wk 文件 是基于嵌入式硬件的优化模型文件，通常用于 NNIE 或其他硬件加速器上进行推理。这是一个在嵌入式平台上部署的二进制模型文件格式。

文件格式：.wk 是为嵌入式硬件（如华为的 HiSilicon 芯片）量身定制的模型文件格式，包含了模型的推理权重和架构信息。

特点：硬件加速优化：.wk 文件格式经过优化，适合在特定硬件上高效推理，充分利用硬件的并行计算能力。

定制化：生成 .wk 文件的过程会依据硬件的特性对模型进行定制化优化，以适配设备资源限制并提高推理速度。

生成与应用：通常通过工具链（如 NNIE Mapper）从 Caffe 模型（.caffemodel 和 .prototxt 文件）生成 .wk 文件，之后可以直接部署在硬件上执行推理任务。

用自己的话说，就是可执行文件，经过多轮转化被硬件支持加速计算的部署文件

4.关于优化和准则：

一般能在一层计算完就计算完，然后整个的计算顺序如下，一般使用xxx层进行计算效果比较好