中国科学院大学计算机组成原理实验课

实 验 报 告

学号：2018K8009929021 姓名：袁欣怡 专业：计算机科学与技术

实验序号：prj5.5 实验名称：深度学习算法及硬件加速

一、 逻辑电路结构与仿真波形的截图及说明（比如关键 RTL 代码段{包含注释}及其对应的逻辑电路结构、相应信号的仿真波形和信号变化的说明等）

1. 卷积算法 convolution

```
for(no=0;no<conv_size.d1;++no){
//number of output pictures
  input_offset=0; //reset
  for(ni=0;ni<conv_size.d0;++ni){
  //number of input pictures
    for(y=0;y<conv_size.d2;++y){
      for(x=0;x<conv_size.d3;++x){
      //(x,y) in output picture
        padx=x*stride;
        pady=y*stride;
        temp=0;

        if (ni==0)
          out[output_offset+y*conv_size.d3+x]=weight[weight_offset]; //add bias

        for(ky=0;ky<weight_size.d2;++ky){
          for(kx=0;kx<weight_size.d3;++kx){
          //(kx,ky) in weight map
            if (padx+kx>=pad && padx+kx<input_fm_w+pad && pady+ky>=pad && pady+ky<input_fm_h+pad) //not in padding zone
              temp_in=(short int)in[input_offset+(pady+ky-pad)*input_fm_w+(padx+kx-pad)];//temp_in=in[padx+kx-pad,pady+ky-
pad]
            else temp_in=0; //in padding zone

            temp_w=(short int)(weight[weight_offset+ky*weight_size.d3+kx+1]); //filter[0,0] is bias, +1 to skip it
            temp+=(int)(temp_in*temp_w);
          }
        }
        out[output_offset+y*conv_size.d3+x]+=(short)(temp>>FRAC_BIT); //only need 10 decimal places
      } //complete one row
    } //complete one input picture

    input_offset+=input_fm_h*input_fm_w; //go to next point
    weight_offset+=weight_size.d2*weight_size.d3+1; //size of a filter = 1+k*k
  } //complete one output picture

  output_offset+=conv_size.d2*conv_size.d3; //go to next output picture
}
```

输入图像共有 conv_size.d0 个，每张图的尺寸为 input_fm_w * input_fm_h。

输出图像共有 conv_size.d1 个，每张图的尺寸为 conv_size.d2 * conv_size.d3。

Weight 数组中存储了一位 bias 值和 k*k 位权重值，因此在 weight 中移动时需要 weight_offset += weight_size.d2 * weight_size.d3 +1。

在处理定点小数的时候两个有十位小数的数相乘会产生二十位小数，因此需要右移 10 位以得到正确的结果。

2. 池化算法 pooling

```
for(no=0;no<conv_size.dl;++no){
  for(y=0;y<pool_out_h;++y){
    for(x=0;x<pool_out_w;++x){
    //(x,y) in pooling output
      ox=x*stride;
      oy=y*stride; //pooling stride
      max=-2147483648; //int min

      for(j=0;j<stride;++j){
        for(i=0;i<stride;++i){
        //(ox+i,oy+j) in pooling area
          if (ox+i>=pad && ox+i<input_fm_w+pad && oy+j>=pad && oy+j<input_fm_h+pad)
            //not in the padding zone
            temp=(short)out[input_offset+(oy+j-pad)*input_fm_w+(ox+i-pad)];
          else temp=0; //in the padding zone

          if (temp>max) max=temp;
        }
      }

      out[output_offset+y*pool_out_w+x] = max;
    }
  }

  input_offset += input_fm_h * input_fm_w; //change to another input
  output_offset += pool_out_h * pool_out_w; //change to another output
}
```

设置变量 max 用来存放最大值，初始化 max 位-2147483648（int 类型下界），最后在 max 中存放取出的最大值。

3. 硬件加速器控制访问

```
int main()
{
        //TODO: Please add your own software to control hardware accelerator
        unsigned long *base = (void *) 0x40040000;
        unsigned long val=0;
        volatile unsigned long *val1;
        unsigned long val2;

        Result res;
        bench_prepare(&res);

        printf("Start convolution\n");
        val=*base&0xfffffffe;
        *base=val+1;

        while(1){
                val1=(void *)0x40040008;
                val2=*val1 & 0x00000001;
        //      printf("%d\n",val1);

                if (val2>0) break;
        }

        bench_done(&res);

        printf("Cycles of sw: %u\n",res.msec);
        printf("Memory visit times: %u\n",res.mem_cycle);

        return 0;
}
```

4. Mips_core 实现 mul 指令

```
assign RF_wen=(state==`EX && (Jal||Jalr))?1:
              (state==`WB && Movn && B!=0)?1:
              (state==`WB && Movz && B==0)?1:
              (state==`WB && op==6'b000000 && !Jr && !Movn && !Movz)?1:
              (state==`WB && Load)?1:
              (state==`WB && (Addiu||Andi||Ori||Xori||Slti||Sltiu||Lui||Mul))?1:
              0;
```

mul 为高时拉高 RF_wen 信号。

```
    assign addr_sign=(Sll||Sllv||Sltu||Slt||And||Or||Xor||Nor||Addu||Sra||Srav||
Subu||Srl||Srlv||Movz||Movn||Jalr||Beq||Bgez||Mul);
    assign RF_waddr=(Jal)?5'b11111:(addr_sign)?rd:rt;
```

RF_waddr 为 rd。

```
assign RF_wdata=(RF_wen==1)?(
                        (Mul==1)?(A*B):
                        (Nor==1)?~Result:
                        (Srl==1)?(B>>shamt):
                        (Srlv==1)?(B>>A[4:0]):
                        (Sll==1)?(B<<shamt):
                        (Sllv==1)?(B<<A[4:0]):
                        ((Jal||Jalr)==1)?next+32'd4:
                        ((Movn||Movz)==1)?A:
                        (Lui==1)?{instruction[15:0],16'd0}:
                        (Load==1)?data1_load:
                        Result):
                    0;
```

RF_wdata 为 A*B，其中 A=r[rs]，B=r[rt]。

对其他信号的处理和 Addi 等信号类似。

二、 实验过程中遇到的问题、对问题的思考过程及解决方法（比如 RTL 代码中出现的逻辑 bug，仿真、本地上板及云平台调试过程中的难点等）

遇到的问题：在 conv:01 无法通过的时候，一开始难以确定是 mips 中 mul 指令出现问题还是 sw_conv 中。后来使用 mul(int a, int b)函数辅助 debug，很快就确定了问题在哪里。

同时发现，在 hw_conv 中，如果不是用 volatile 关键字，可能导致编译器将赋值语句优化掉，导致值不会更新。同时 volatile 关键字只能对指针

类型的变量使用，对普通变量无效。

三、 对讲义中思考题（如有）的理解和回答

1. 如果想在 main()函数中使用当前 C 源码文件中未定义声明的 bench_prepare()、bench_done()和 printf()，还要做什么？

   #include<xxx.h>

2. 加速前后的对比：

   加速前：

```
Starting xl2tpd (via systemctl): xl2tpd.service.
Remote target: root@172.16.15.58
Try to reboot 172.16.15.58
Waiting for target reboot...
Completed FPGA configuration
Evaluating convolution benchmark suite...
Launching hw_conv benchmark...
Initializing read image data
Initializing weights
Start convolution
Cycles of sw: 409165
Memory visit times: 1020
Hit good trap
pass 1 / 1
Stopping xl2tpd (via systemctl): xl2tpd.service.
2020年 06月 28日 星期日 18:08:31 CST
```

   加速后：

```
Starting xl2tpd (via systemctl): xl2tpd.service.
Remote target: root@172.16.15.62
Try to reboot 172.16.15.62
Waiting for target reboot...
Completed FPGA configuration
Evaluating convolution benchmark suite...
Launching sw_conv benchmark...
Initializing read image data
Initializing weights
starting convolution
starting pooling
Cycles of sw: 186009704
Memory visit times: 1390772
Hit good trap
pass 1 / 1
Stopping xl2tpd (via systemctl): xl2tpd.service.
2020年 06月 28日 星期日 17:00:02 CST
```

四、 在课后，你花费了大约 15 小时完成此次实验。

五、 对于此次实验的心得、感受和建议（比如实验是否过于简单或复杂，是否缺少了某些你认为重要的信息或参考资料，对实验项目的建议，对提供帮助的同学的感谢，以及其他想与任课老师交流的内容等）

在理解 2D 卷积算法之后不难书写，但是 PPT 中没有提供足够的资料，理解起来还有一些困难。而且由于程序本地仿真时间太长，在 debug 时会感觉无从下手。感谢蒋卓伦同学的帮助。