ncnn 在 arm-linux 平台的编译

https://github.com/Tencent/ncnn

在原代码基础上修改部分代码

1. **增加 arm-gcc.toolchain.cmake 文件**

```
set( ARM-LINUX True )          #自定义的名字
set(CMAKE_C_COMPILER     "/usr/bin/arm-linux-gnueabihf-gcc")      #编译工具所在路径
set(CMAKE_CXX_COMPILER    "/usr/bin/arm-linux-gnueabihf-g++")    #编译工具所在路径
```

2. **build.sh**

   添加

```
##### gcc-arm armv7
mkdir -p build-arm-gcc-armv7
pushd build-arm-gcc-armv7
cmake -DCMAKE_TOOLCHAIN_FILE=../arm-gcc.toolchain.cmake ..
make
make install
popd
```

3. **src/CMakeLists.txt**

```
31    message("WITH_LAYER_${name} = ${WITH_LAYER_${name}}")
32
33    if(WITH_LAYER_${name})
34        list(APPEND ncnn_SRCS "${CMAKE_CURRENT_SOURCE_DIR}/layer/${name}.cpp")
35
36        # look for arch specific implementation and append source
37        # optimized implementation for armv7 aarch64
38        if((ANDROID AND ("${CMAKE_SYSTEM_PROCESSOR}" STREQUAL "armv7-a"))
39            OR (ANDROID AND ("${CMAKE_SYSTEM_PROCESSOR}" STREQUAL "aarch64"))
40            OR (ARM-LINUX)
41            OR (IOS AND ("${CMAKE_OSX_ARCHITECTURES}" STREQUAL "armv7"))
42            OR (IOS AND ("${CMAKE_OSX_ARCHITECTURES}" STREQUAL "arm64"))
43            OR (IOS AND ("${CMAKE_OSX_ARCHITECTURES}" STREQUAL "armv7;arm64")))
44            if(EXISTS "${CMAKE_CURRENT_SOURCE_DIR}/layer/arm/${name}_arm.cpp")
45                list(APPEND ncnn_SRCS "${CMAKE_CURRENT_SOURCE_DIR}/layer/arm/${name}_
46                set(WITH_LAYER_${name}_arm 1)
47            endif()
48        else()
49            if(EXISTS "${CMAKE_CURRENT_SOURCE_DIR}/layer/x86/${name}_x86.cpp")
50                list(APPEND ncnn_SRCS "${CMAKE_CURRENT_SOURCE_DIR}/layer/x86/${name}_
51                set(WITH_LAYER_${name}_x86 1)
52            endif()
53        endif()
54    endif()
```

   保证能跑到${name}_arm.cpp 文件

4. **CMakeLists.txt**

```
add_definitions(-fvisibility=hidden -fvisibility-inlines-hidden)

if(ARM-LINUX)
    list(APPEND CMAKE_C_FLAGS "-Wall -Wno-unknown-pragmas -fPIC -fexceptions")
    list(APPEND CMAKE_C_FLAGS " -marm -mfloat-abi=hard -mfpu=neon ")

    list(APPEND CMAKE_CXX_FLAGS "${CMAKE_C_FLAGS} -Wno-reorder -Wno-sign-compare -std=c++11 ")
    string(REGEX REPLACE ";" " " CMAKE_C_FLAGS ${CMAKE_C_FLAGS})
    string(REGEX REPLACE ";" " " CMAKE_CXX_FLAGS ${CMAKE_CXX_FLAGS})
endif()

if(ANDROID)
    # disable shared library on android
    set_property(GLOBAL PROPERTY TARGET_SUPPORTS_SHARED_LIBS FALSE)
    set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fno-rtti -fno-exceptions")
elseif(ARM-LINUX)
    # disable shared library on arm-linux
    set_property(GLOBAL PROPERTY TARGET_SUPPORTS_SHARED_LIBS FALSE)
    set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fno-rtti -fno-exceptions")
elseif(IOS)
    # disable shared library on xcode ios
    set_property(GLOBAL PROPERTY TARGET_SUPPORTS_SHARED_LIBS FALSE)
    set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fno-rtti -fno-exceptions")
endif()
```

添加编译选项，这里以 Raspberry Pi2 为例，编译选项是-mfloat-abi=hard，其他平台可能大部分是 softfp，注意修改。

5. **检查生成的 libncnn.a**

readelf libncnn.a –a，查看生成的库，有红色标注的才是正确的。

```
No version information found in this file.
Attribute Section: aeabi
File Attributes
  Tag_CPU_name: "7-A"
  Tag_CPU_arch: v7
  Tag_CPU_arch_profile: Application
  Tag_ARM_ISA_use: Yes
  Tag_THUMB_ISA_use: Thumb-2
  Tag_FP_arch: VFPv3
  Tag_Advanced_SIMD_arch: NEONv1
  Tag_ABI_PCS_wchar_t: 4
  Tag_ABI_FP_number_model: Finite
  Tag_ABI_align_needed: 8-byte
  Tag_ABI_align_preserved: 8-byte, except leaf SP
  Tag_ABI_enum_size: int
  Tag_ABI_HardFP_use: Deprecated
  Tag_ABI_VFP_args: VFP registers
  Tag_ABI_optimization_goals: Aggressive Speed
  Tag_CPU_unaligned_access: v6
```