

嵌入式系統軟體設計與實作 實驗報告十三

資工三 許耘熙 411410054

實驗名稱：

從原始碼建置 TVM 並在嵌入式裝置上部署深度學習模型

實驗目的：

從原始碼編譯並安裝 TVM，在樹梅派上編譯並部署 TVM Runtime，設置 RPC 伺服器以支援遠端推論。透過範例程式，實作在資源受限裝置上使用 TVM RPC 執行 ResNet-18 模型推論。評估不同最佳化等級(opt_level)對編譯時間、執行時間與記憶體使用的影響，以及遠端與本地執行效能差異。

實驗步驟：

1. 在本機設置相關環境以及安裝套件並編譯 TVM

```
sudo apt-get install -y python3 python3-dev python3-setuptools \
gcc git libtinfo-dev zlib1g-dev build-essential cmake libedit-dev \
libxml2-dev llvm-dev llvm
wget https://dlcdn.apache.org/tvm/tvm-v0.18.0/apache-tvm-src-
v0.18.0.tar.gz
tar zxvf apache-tvm-src-v0.18.0.tar.gz
mv apache-tvm-src-v0.18.0 tvm
cd tvm
rm -rf build && mkdir build && cd build
cp ../cmake/config.cmake .
echo "set(USE_LLVM ON)" >> config.cmake
echo "set(HIDE_PRIVATE_SYMBOLS ON)" >> config.cmake
echo "set(USE_CUDA OFF)" >> config.cmake
echo "set(USE_METAL OFF)" >> config.cmake
echo "set(USE_VULKAN OFF)" >> config.cmake
echo "set(USE_OPENCL OFF)" >> config.cmake
echo "set(USE_CUBLAS OFF)" >> config.cmake
echo "set(USE_CUDNN OFF)" >> config.cmake
echo "set(USE_CUTLASS OFF)" >> config.cmake
echo "set(USE_GRAPH_EXECUTOR ON)" >> config.cmake
echo "set(USE_PROFILER ON)" >> config.cmake
echo "set(CMAKE_BUILD_TYPE Release)" >> config.cmake
cmake .. -G Ninja
```

```
ninja
echo "export TVM_HOME=~/.tvm" >> ~/.bashrc
echo "export PYTHONPATH=$TVM_HOME/python:$PYTHONPATH" >> ~/.bashrc
source ~/.bashrc
sudo -s
apt install python3-pip
python3 -m venv ./lab13
source ./lab13/bin/activate
pip install numpy decorator attrs typing-extensions psutil scipy \
packaging Pillow torchvision tornado
python -c "import tvm;print(tvm.__version__)"
```

2. 在樹莓派上設置相關環境以及安裝套件並編譯 TVM

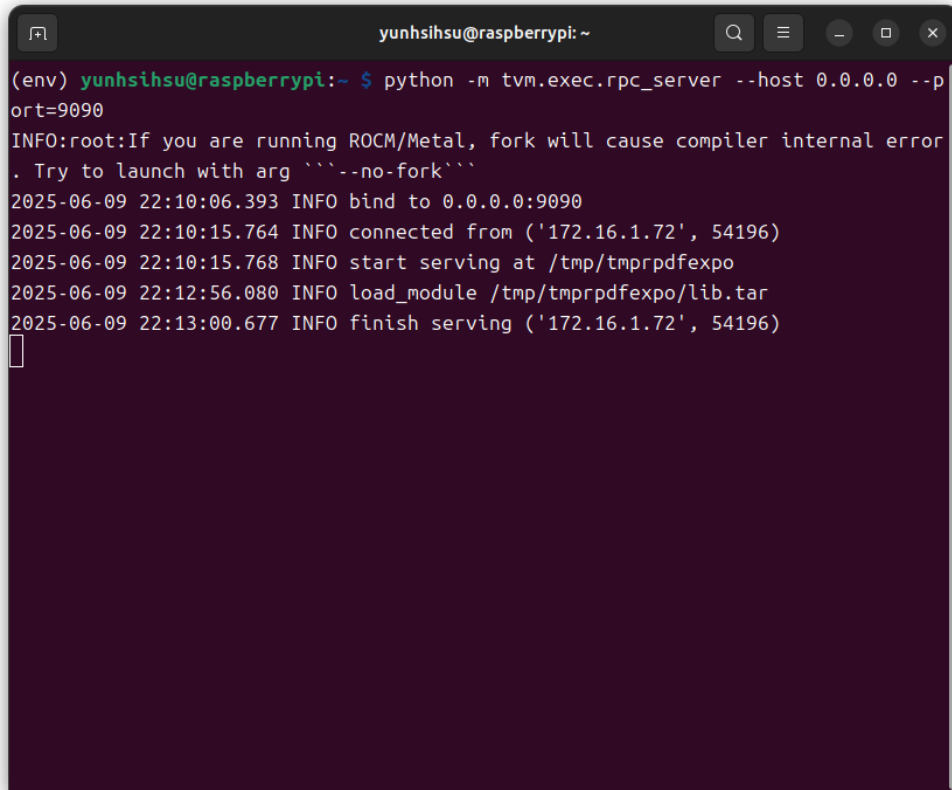
```
ssh -X yunhsihsu@172.16.1.53
wget https://d1cdn.apache.org/tvm/tvm-v0.18.0/apache-tvm-src-
v0.18.0.tar.gz
tar xzvf apache-tvm-src-v0.18.0.tar.gz
mv apache-tvm-src-v0.18.0 tvm
cd tvm
rm -rf build && mkdir build && cd build
cp ../cmake/config.cmake .
sudo fallocate -l 4G /swapfile
sudo chmod 600 /swapfile
sudo /mkswap /swapfile
sudo swapon /swapfile
echo "set(USE_GRAPH_EXECUTOR ON)" >> config.cmake
cmake .. -G Ninja
ninja runtime -v -j3
echo "export TVM_HOME=~/.tvm" >> ~/.bashrc
echo "export PYTHONPATH=$TVM_HOME/python:$PYTHONPATH" >> ~/.bashrc
source ~/.bashrc
python3 -m venv ./env
source ./env/bin/activate
pip install numpy decorator attrs typing-extensions psutil tornado \
packaging torchvision cloudpickle
cd ..
python -m tvm.exec.rpc_server --host 0.0.0.0 --port=9090
```

3. 在本機上執行程式碼

```
pip install pytest
python test.py
```

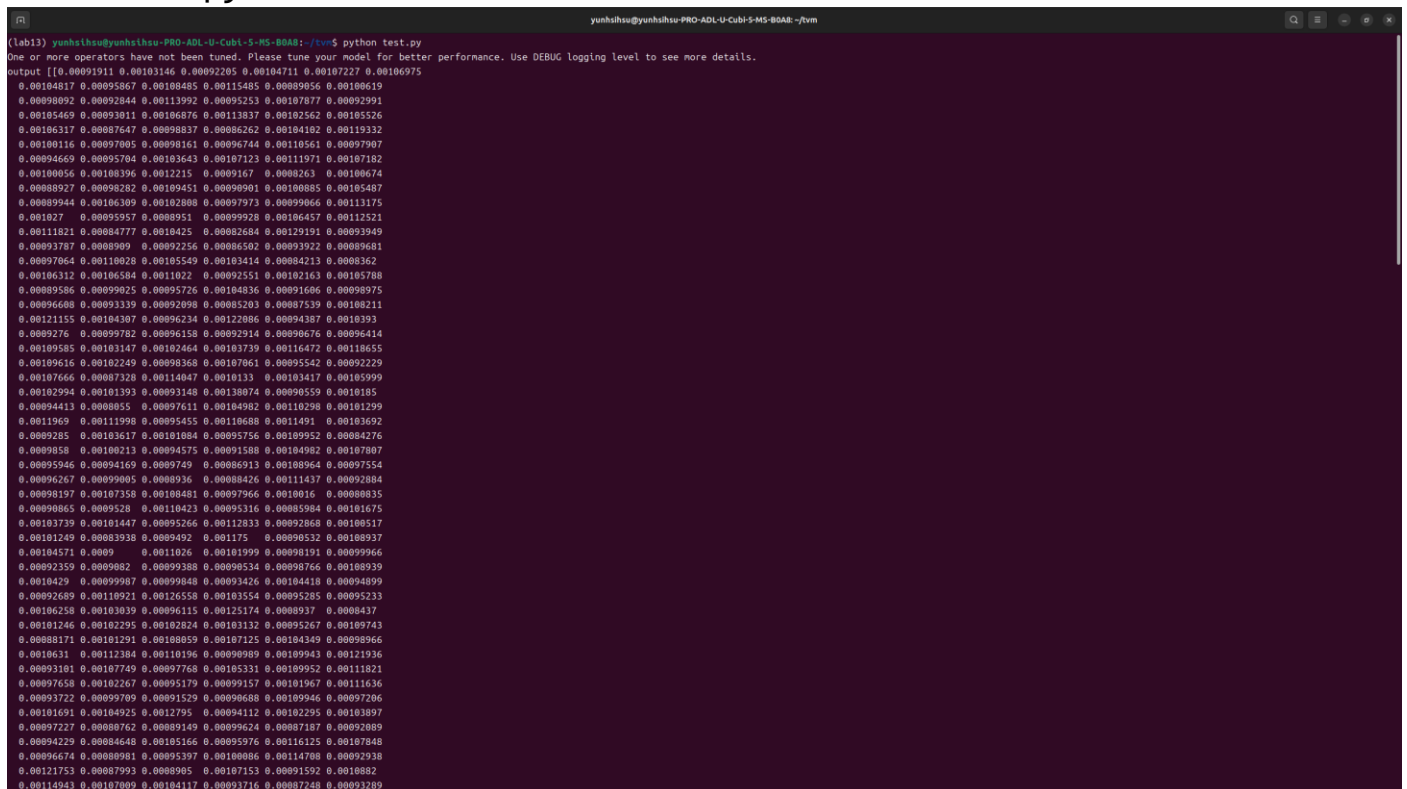
問題與討論：

1. 請截圖成功執行程式的結果。



```
yunhsihsu@raspberrypi: ~  
(env) yunhsihsu@raspberrypi:~ $ python -m tvm.exec.rpc_server --host 0.0.0.0 --port=9090  
INFO:root:If you are running ROCM/Metal, fork will cause compiler internal error  
. Try to launch with arg ``--no-fork``  
2025-06-09 22:10:06.393 INFO bind to 0.0.0.0:9090  
2025-06-09 22:10:15.764 INFO connected from ('172.16.1.72', 54196)  
2025-06-09 22:10:15.768 INFO start serving at /tmp/tmpprpdfexpo  
2025-06-09 22:12:56.080 INFO load_module /tmp/tmpprpdfexpo/lib.tar  
2025-06-09 22:13:00.677 INFO finish serving ('172.16.1.72', 54196)
```

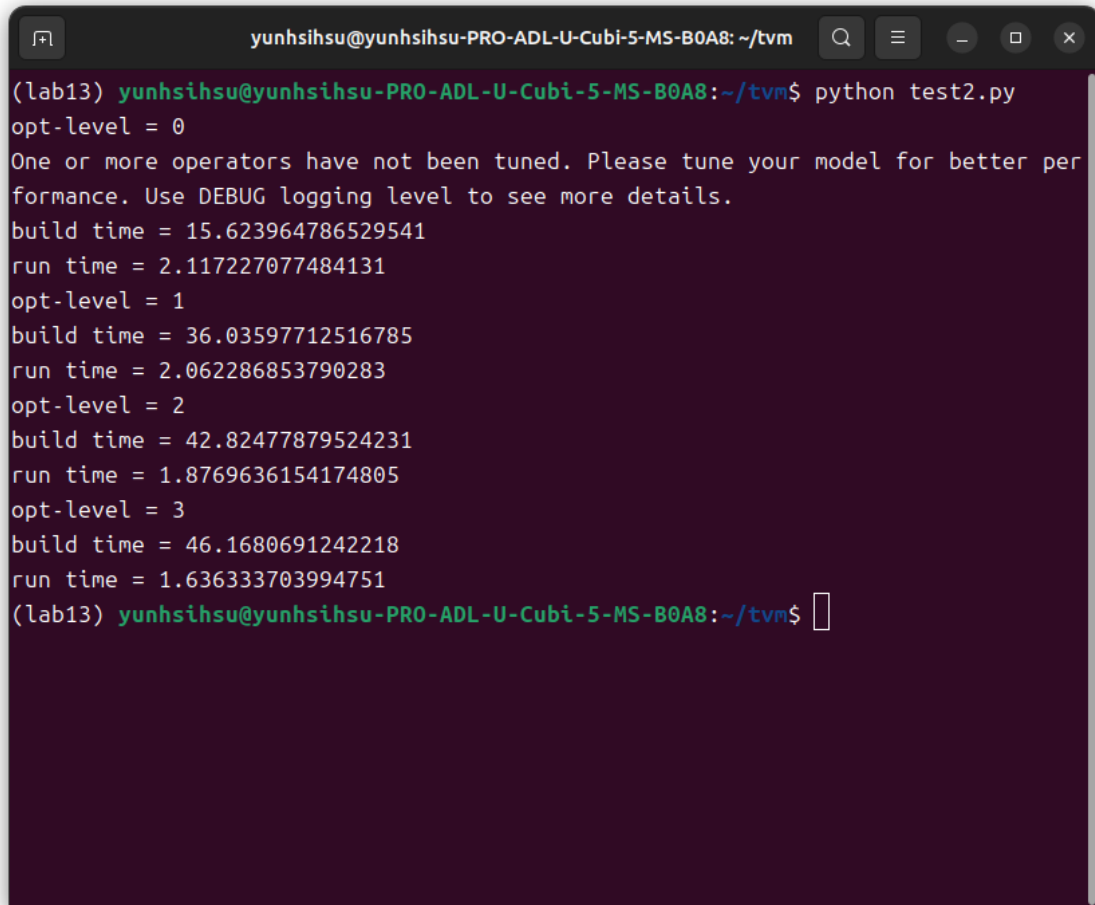
執行 test.py



```
[lab13] yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0AB: ~/tvm$ python test.py  
One or more operators have not been tuned. Please tune your model for better performance. Use DEBUG logging level to see more details.  
output [[0.00091911 0.00103146 0.00092205 0.00104711 0.00107227 0.00106975  
0.00104817 0.00095867 0.00108485 0.00115485 0.00089856 0.00100619  
0.00090992 0.00092844 0.00113992 0.00095253 0.00107877 0.00092991  
0.00105469 0.00093811 0.00106876 0.00113837 0.00102562 0.00105526  
0.00106317 0.00087647 0.00098837 0.00086262 0.00104102 0.00119332  
0.00100116 0.00097805 0.00098161 0.00096744 0.00110561 0.00097907  
0.00094669 0.00095704 0.00103643 0.00107123 0.00111971 0.00107182  
0.00100056 0.00108396 0.0012215 0.0009167 0.0008263 0.00100674  
0.00088927 0.00098282 0.00109451 0.00090901 0.00100885 0.00105487  
0.00089944 0.00106309 0.00102808 0.00097573 0.00099866 0.00113175  
0.001027 0.00095957 0.0008951 0.00099928 0.00106457 0.00112521  
0.00111821 0.00084777 0.0010425 0.00082684 0.00129191 0.00093949  
0.00093787 0.0008909 0.00092256 0.00086502 0.00093922 0.00089681  
0.00097044 0.00110028 0.00105549 0.00103114 0.00084213 0.0008362  
0.00106312 0.00106584 0.0011022 0.00092551 0.00102163 0.00105708  
0.0009586 0.00098025 0.00095726 0.00104836 0.00091006 0.00080975  
0.00096608 0.00093339 0.00092098 0.00085203 0.00087539 0.00108211  
0.00121155 0.00104307 0.00096234 0.00122086 0.00094387 0.0010393  
0.0009276 0.00099782 0.00096158 0.00092914 0.00090676 0.00096414  
0.00109585 0.00103147 0.00102464 0.00103739 0.00116472 0.00118655  
0.00109616 0.00102249 0.00098368 0.00107061 0.00095542 0.00092229  
0.00107666 0.00087328 0.00114047 0.0010133 0.00103417 0.00105999  
0.00102994 0.00101393 0.00093148 0.00138074 0.00090559 0.00101815  
0.00094413 0.0008855 0.00097611 0.00104902 0.00110298 0.00101299  
0.0011069 0.00111998 0.00095455 0.00110688 0.0011491 0.00103692  
0.0009285 0.00103617 0.00101804 0.00095756 0.00109952 0.00084276  
0.0009858 0.00100213 0.00094575 0.00091588 0.00104082 0.00107807  
0.00095946 0.00094169 0.0009749 0.00086913 0.00108964 0.00097554  
0.00096267 0.00099005 0.0008936 0.00088426 0.00111437 0.00092884  
0.00089197 0.00107358 0.00108481 0.00097966 0.0010016 0.00088035  
0.00090865 0.0009528 0.00110423 0.00095316 0.00085984 0.00101675  
0.00103739 0.00101447 0.00095266 0.00112833 0.00092868 0.00100517  
0.00101249 0.00083938 0.0009492 0.001175 0.00090532 0.00108937  
0.00104571 0.0009 0.0011026 0.00101999 0.00098191 0.00099966  
0.00092359 0.0009080 0.00099388 0.00090534 0.00098766 0.00100939  
0.0010429 0.00099897 0.00099848 0.00093426 0.00104418 0.00094899  
0.00092689 0.00110921 0.00126558 0.00103554 0.00095285 0.00095233  
0.00106258 0.00103039 0.00096115 0.00125174 0.0008937 0.0008437  
0.00101246 0.00102295 0.00102824 0.00103132 0.00095267 0.00109743  
0.00088171 0.00101291 0.00108059 0.00107125 0.00104349 0.00098966  
0.0010631 0.00112384 0.00110196 0.00090989 0.00109943 0.00121936  
0.00093101 0.00107749 0.00097768 0.00105331 0.00109952 0.00111821  
0.00097658 0.00102267 0.00095179 0.00099157 0.00101967 0.00111636  
0.00093722 0.00099709 0.00091529 0.00090688 0.00109946 0.00097206  
0.00101691 0.00104925 0.0012795 0.00094112 0.00102295 0.00103897  
0.00097227 0.00088762 0.00089149 0.00099624 0.00087187 0.00092889  
0.00094229 0.00084648 0.00105166 0.00095976 0.00116325 0.00107848  
0.00096674 0.00080981 0.00095397 0.00100806 0.00114708 0.00092938  
0.00121753 0.00087993 0.0008905 0.00107153 0.00091592 0.00108882  
0.00114943 0.00107009 0.00104117 0.00093716 0.00087248 0.00093289
```

2. 比較不同 TVM `opt_level` 的效能 (`opt_level=0,1,2,3`)，使用 Python 所提供的 `time.time` 分別計算 `relay.build(mod,target,params)` 的編譯時間與 `module.run()` 的執行時間。請嘗試說明每個 `opt_level` 對模型進行了哪些優化，並觀察 `build` 時間與 `run` 時間的差異。

加入測量時間的變數，並執行



```
(lab13) yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8: ~/tvm$ python test2.py
opt-level = 0
One or more operators have not been tuned. Please tune your model for better performance. Use DEBUG logging level to see more details.
build time = 15.623964786529541
run time = 2.117227077484131
opt-level = 1
build time = 36.03597712516785
run time = 2.062286853790283
opt-level = 2
build time = 42.82477879524231
run time = 1.8769636154174805
opt-level = 3
build time = 46.1680691242218
run time = 1.636333703994751
(lab13) yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8: ~/tvm$
```

`opt_level 0`：不進行任何優化，直接把 Relay IR 轉為原始程式碼，僅作必要的型別與形狀檢查。

`opt_level 1`：基本優化，常數折疊與死代碼消除

`opt_level 2`：中等優化，算子融合與資料布局轉換

`opt_level 3`：高度優化，向量化、迴圈展開、SIMD 使用等，並做線程排程與記憶體預取。

優化越多，編譯時間越長，執行時間越短

3. 比較 TVM `opt_level` 為 0 與 3 時，在不同優化下的記憶體使用情況。請透過下列程式碼觀察記憶體變化，並簡要分析 `build` 階段與 `run` 階段的記憶體使用差異。

加入測量空間的變數，並執行

因為 `opt_level 3` 的優化較多，因此也會使用較多空間。

```
yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8: ~/tvm
(lab13) yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8:~/tvm$ python test3.py
opt-level = 0
One or more operators have not been tuned. Please tune your model for better performance. Use DEBUG logging level to see more details.
Build memory usage: 201.58 MB
Run memory usage: 0.00 MB
opt-level = 3
Build memory usage: 347.87 MB
Run memory usage: 0.00 MB
(lab13) yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8:~/tvm$
```

4. 修改 `local_demo` 變數為 `True`，在本地執行推論，

```
yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8: ~/tvm
(lab13) yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8:~/tvm$ python test4.py
opt-level = 0
One or more operators have not been tuned. Please tune your model for better performance. Use DEBUG logging level to see more details.
build time = 4.2553324699401855
2025-06-09 22:48:09.451 INFO load_module /tmp/tmp15p_3c0k/lib0.tar
run time = 0.09830164909362793
opt-level = 1
build time = 4.986120223999023
2025-06-09 22:48:19.665 INFO load_module /tmp/tmp15p_3c0k/lib1.tar
run time = 0.08733129501342773
opt-level = 2
build time = 11.918745040893555
2025-06-09 22:48:36.648 INFO load_module /tmp/tmp15p_3c0k/lib2.tar
run time = 0.07294440269470215
opt-level = 3
build time = 21.347509145736694
2025-06-09 22:49:02.953 INFO load_module /tmp/tmp15p_3c0k/lib3.tar
run time = 0.0708770751953125
(lab13) yunhsihsu@yunhsihsu-PRO-ADL-U-Cubi-5-MS-B0A8:~/tvm$
```

因為本地的效能比較高，因此執行時間降低很多。

5. 實驗心得：

在本次實驗中，我遇到最大的困難是在樹莓派上面操作時，常常會當機，造成很多時候都在重複操作相同內容，光是樹莓派的編譯致少花費了 5 天，後續的操作就不會太困難了。這次的實驗我了解了 TVM 的建置流程與可客製化的最佳化選項，並實際操作了跨平台部署與遠端推論機制，充分的加深了對嵌入式裝置上深度學習部署挑戰的認識。