

Neural Network

김효관

1. 딥러닝 개발환경 설치

설치 절차

1	Visual C++	텐서플로우는 실제 C기반	vc_redist.x64.exe
2	CUDA <small>* nvidia cuda 지원 그래픽카드 없는경우 스킵</small>	분산처리 플랫폼 (nvidia)	cuda_9.0.176_win10_network.exe
3	cuDNN	소프트웨어 개발 Kit / cuda lib	cudnn-9.0-windows10-x64-v7.4.2.24.zip
4	라이브러리 설치	딥러닝 라이브러리	

CUDA : Compute Unified Device Architecture

윈도우 10 기준 설치 파일은 www.sparkkorea.com AI강의안 자료실 내
01. 프로그램 (64비트운영체제) -> 6. 텐서플로우 cuda 에서 한번에 다운 가능함

1. 딥러닝 개발환경 설치

설치 절차

1	Visual C++	텐서플로우는 실제 C기반	vc_redist.x64.exe
2	CUDA * nvidia cuda 지원 그래픽카드 없는경우 스킵	분산처리 플랫폼 (nvidia)	cuda_9.0.176_win10_network.exe
3	cuDNN	소프트웨어 개발 Kit / cuda lib	cudnn-9.0-windows10-x64-v7.4.2.24.zip
4	라이브러리 설치	딥러닝 라이브러리	

CUDA : Compute Unified Device Architecture

윈도우 10 기준 설치 파일은 www.sparkkorea.com AI강의안 자료실 내
01. 프로그램 (64비트운영체제) -> 6. 텐서플로우 cuda 에서 한번에 다운 가능함

Nvidia 확인

설치 준비



1-1. Visual C++

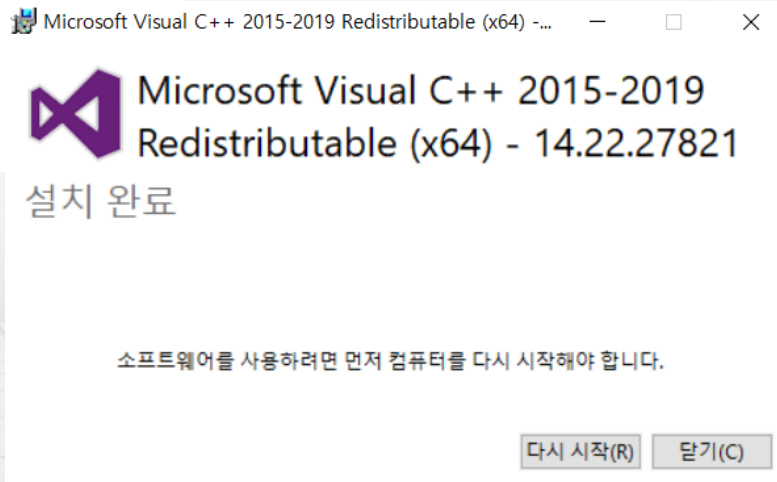
설치 방법

링크 : <https://support.microsoft.com/ko-kr/help/2977003/the-latest-supported-visual-c-downloads>

Visual Studio 2015, 2017 및 2019

Visual Studio 2015, 2017 및 2019용 Microsoft Visual C++ 재배포 가능 패키지를 다운로드합니다. 다음 업데이트는 Visual Studio 2015, 2017 및 2019용으로 지원되는 최신 Visual C++ 재배포 가능 패키지입니다. Universal C Runtime의 기본 버전이 포함되어 있습니다. 자세한 내용은 MSDN을 참조하세요.

- x86: [vc_redist.x86.exe](#)
- x64: [vc_redist.x64.exe](#)
- ARM64: [vc_redist.arm64.exe](#)



1-2. CUDA

설치 방법

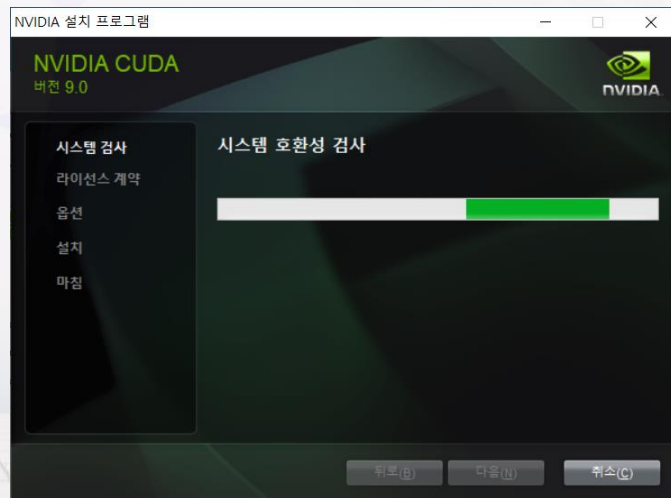
링크 : <https://developer.nvidia.com/cuda-90-download-archive>

Select Target Platform ⓘ

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX
Architecture ⓘ	x86_64		
Version	10	8.1	7
		Server 2016	Server 2012 R2
Installer Type ⓘ	exe [network]	exe [local]	

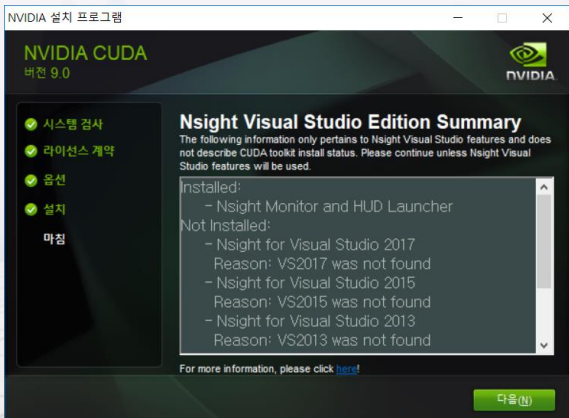
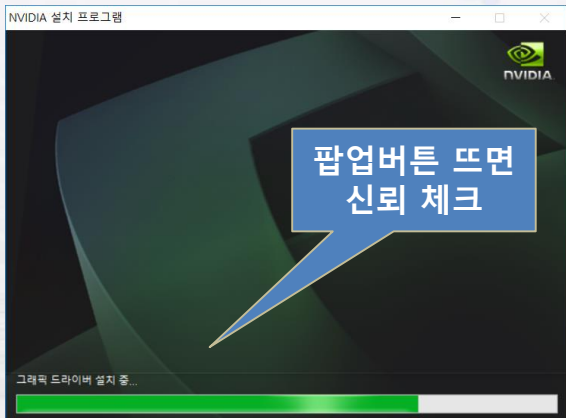
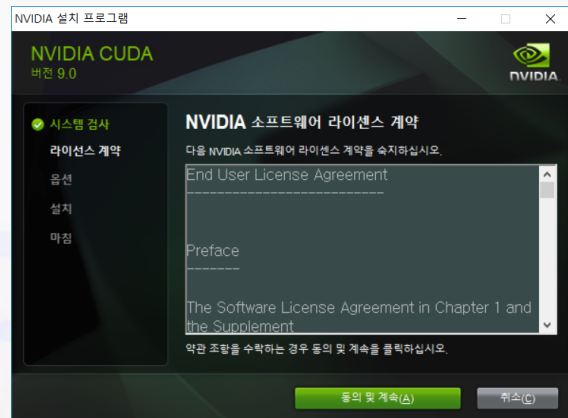
빠른설치 선택



설치참조: <https://pythonkim.tistory.com/137>

<http://twinstarinfo.blogspot.com/2018/12/tensorflow-gpu-install-nvidia-cuda.html>

1-2. CUDA

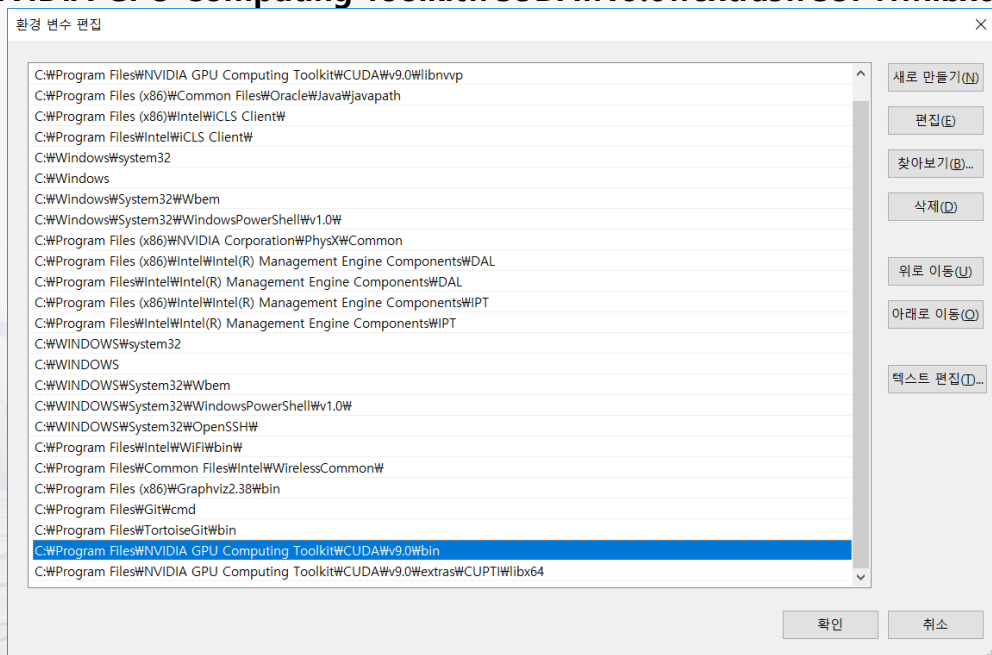


1-2. CUDA

환경변수 등록 (개인별 cuda 설치 폴더 다를 수 있음)

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\extras\CUPTI\libx64



1-3. cuDNN

설치 방법

링크 : <https://developer.nvidia.com/cudnn>

NVIDIA cuDNN

The NVIDIA CUDA® Deep Neural Network library (cuDNN) is a GPU-accelerated implementation of standard routines such as forward and backward convolution, pooling, and fully connected layer operations.

Deep learning researchers and framework developers worldwide rely on cuDNN to accelerate their neural networks and developing software applications rather than spending time on implementing these operations. cuDNN is integrated into the frameworks, including Caffe, Caffe2, Chainer, Keras, MATLAB, MxNet, PyTorch, and TensorFlow, that has cuDNN integrated into the frameworks, visit [NVIDIA cuDNN](#).

Download cuDNN >

Introductory Webinar >

cuDNN Archive | NVIDIA Developer

<https://developer.nvidia.com/rdp/cudnn-archive>

[Download cuDNN v7.4.2 \(Dec 14, 2018\), for CUDA 9.0](#)

Library for Windows, Mac, Linux, Ubuntu and RedHat/Centos (x86_64 architecture)

[cuDNN Library for Windows 7](#)

[cuDNN Library for Windows 10](#)

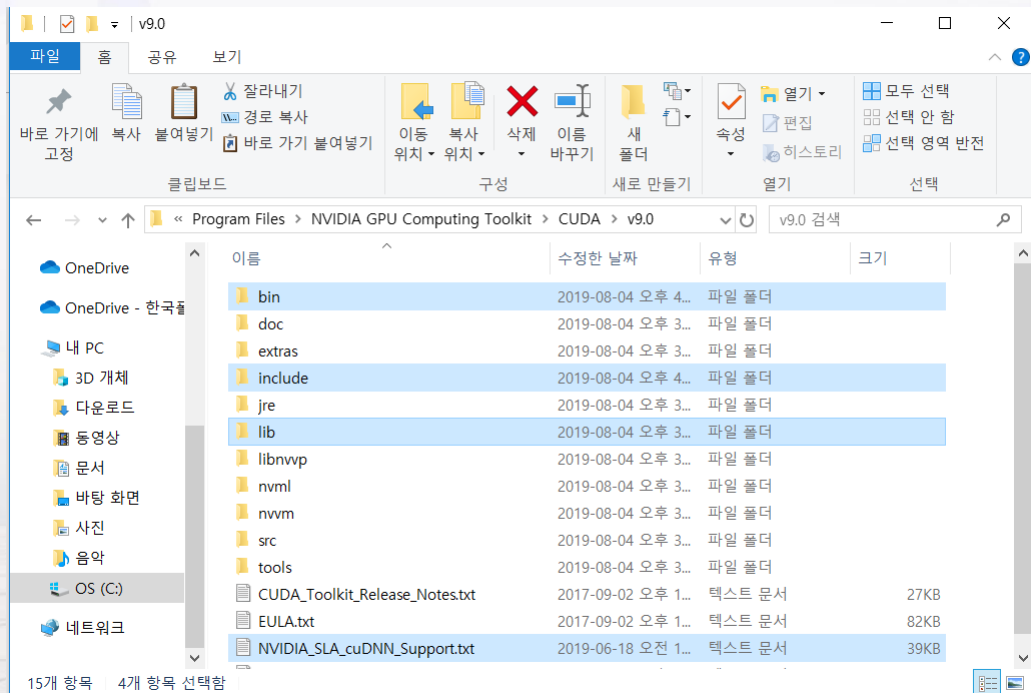
[cuDNN Library for Linux](#)

회원가입 필요

1-3. cuDNN

설치파일 복사

cuDNN 압축해제 후 아래 폴더 내용을 CUDA 설치 폴더 내 복사



1-4. 라이브러리 설치

라이브러리 호환성 맞춰 설치

텐서플로우와 호환성이 맞는 파이썬 3.6 downgrade 및 tensorflow 호환 버전 설치

아나콘다 downgrade

conda install python=3.6

tensorflow 설치

pip install tensorflow-gpu==1.12

pip install keras==2.2.4

conda update conda 실행 후 진행 "requests"..error
io_loop error 시 pip install tornado==4.5.3

가상환경 생성 후 진행해도 무방
conda create -n haiteam pip python=3.6
conda activate haiteam

```
(base) C:\Users\Whk>
(base) C:\Users\Whk>pip install tensorflow-gpu==1.12
Collecting tensorflow-gpu==1.12
  Using cached https://files.pythonhosted.org/packages/88/73/13e4071739df8d5ee7a27780d66bc98a516125
21ad7e5a1e468d9507087c/tensorflow_gpu-1.12.0-cp36-cp36m-win_amd64.whl
Collecting protobuf>=3.6.1 (from tensorflow-gpu==1.12)
  Using cached https://files.pythonhosted.org/packages/74/74/44ec96740ed10ae6d0508efc083c6b7e605c50
9bc32136e9aea840d09daf/protobuf-3.9.0-cp36-cp36m-win_amd64.whl
```

pip install --upgrade sklearn
pip install --upgrade scipy

1-4. 라이브러리 설치

라이브러리 확인

import tensorflow as tf

from tensorflow.python.client import device_lib

device_lib.list_local_devices()

```
import tensorflow as tf

from tensorflow.python.client import device_lib

device_lib.list_local_devices()

[name: "/device:CPU:0"
 device_type: "CPU"
 memory_limit: 268435456
 locality {
 }
 incarnation: 18268563969281400942, name: "/device:GPU:0"
 device_type: "GPU"
 memory_limit: 1472089292
 locality {
   bus_id: 1
   links {
   }
 }
 incarnation: 6618853039916440327
 physical_device_desc: "device: 0, name: GeForce 940MX, pci bus id: 0000:01:00.0, compute capability: 5.0"]
```

1-4. 라이브러리 설치

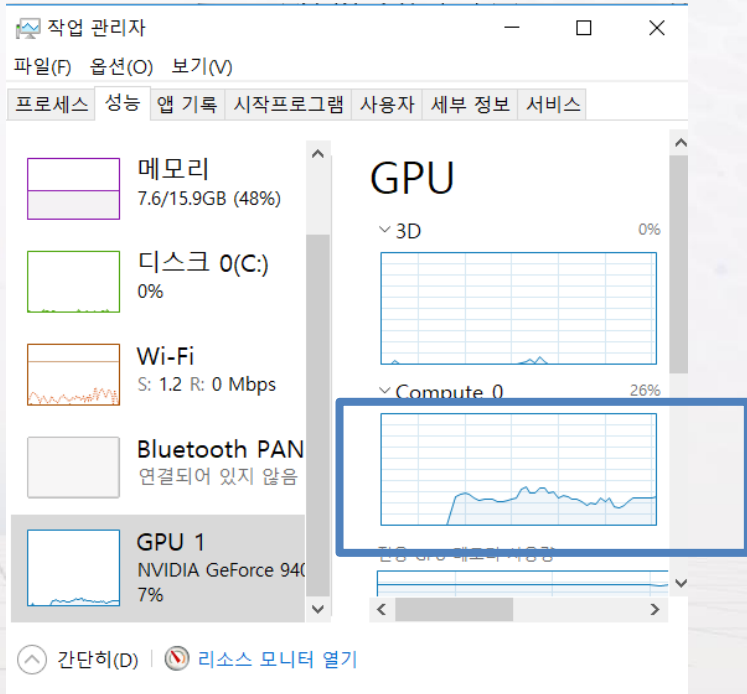
테스트모델 생성 및 확인

```
model.summary()
```

Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 64)	320
dense_8 (Dense)	(None, 64)	4160
dense_9 (Dense)	(None, 1)	65
Total params: 4,545		
Trainable params: 4,545		
Non-trainable params: 0		

```
model.compile(loss='mean_squared_error',  
              optimizer='adam',  
              metrics=['mean_absolute_error', 'mean_squared_error'])
```

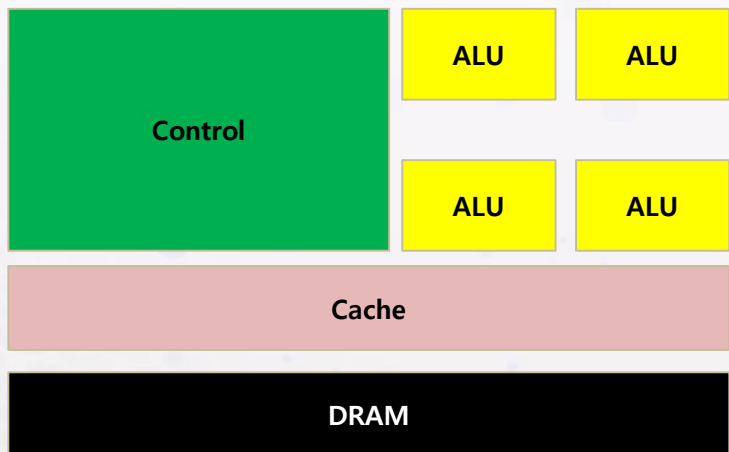
```
from keras.callbacks import EarlyStopping  
#set early stopping monitor so the model stops training when it won't improve anymore  
early_stopping_monitor = EarlyStopping(patience=2000)  
EPOCHS = 2000  
#train model  
history = model.fit(normed_train_data, train_labels, validation_split=0.2,  
                    epochs=EPOCHS, callbacks=[early_stopping_monitor])
```



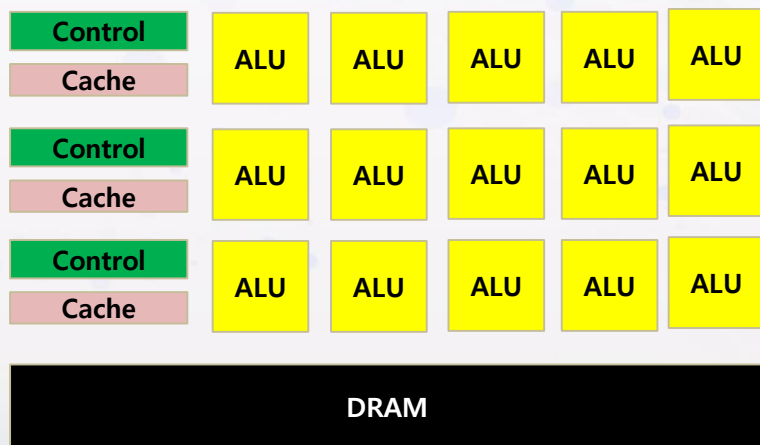
참조. CPU vs GPU

CPU vs GPU

CPU



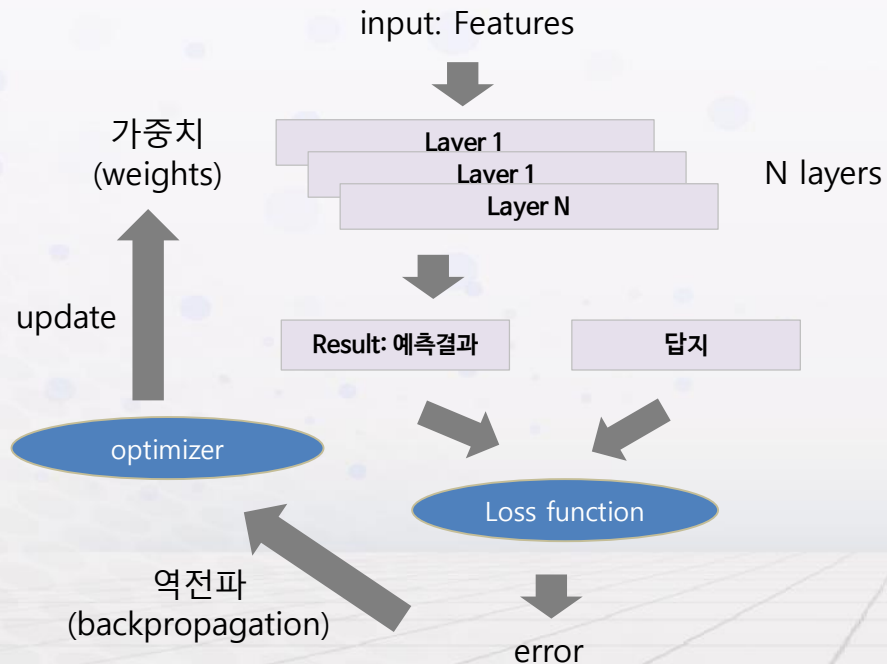
GPU



GPU는 코어가 많음
CPI는 복잡한 계산을 빠르게 하지만 직렬 처리, GPU는 많은 연산을 병렬로 처리함

참조. 딥러닝 작동 원리

딥러닝 프로세스



주요용어	내용
답지	알고있는 Label
가중치	가장 효율적인 식 $y=wx+b$
손실함수 (Loss function)	예측결과와 실제값의 차이를 계산
역전파 (Backpropatation)	손실함수의 결과 개선을 위해 가중 치 수정과정 (optimizer 담당)

감사합니다.

