텐서플로 걸음마

TensorFlow

( )
2 . .

2016. 11. 14.

발표 소개

- **“텐서플로 첫걸음”** 요약
- **MNIST CNN** 자세히
- 쉬운 딥러닝을 위한 **TFLearn**
- TensorBoard
- 일주일 동안 알게된 것들

TensorFlow는 어디에 쓰이나?

- 머신 러닝으로 작곡하는 프로젝트
- TensorFlow를 사용
- 7초 정도의 멜로디 생성하는데 성공
  (magenta.tensorflow.org에 영상 두 개 있음)

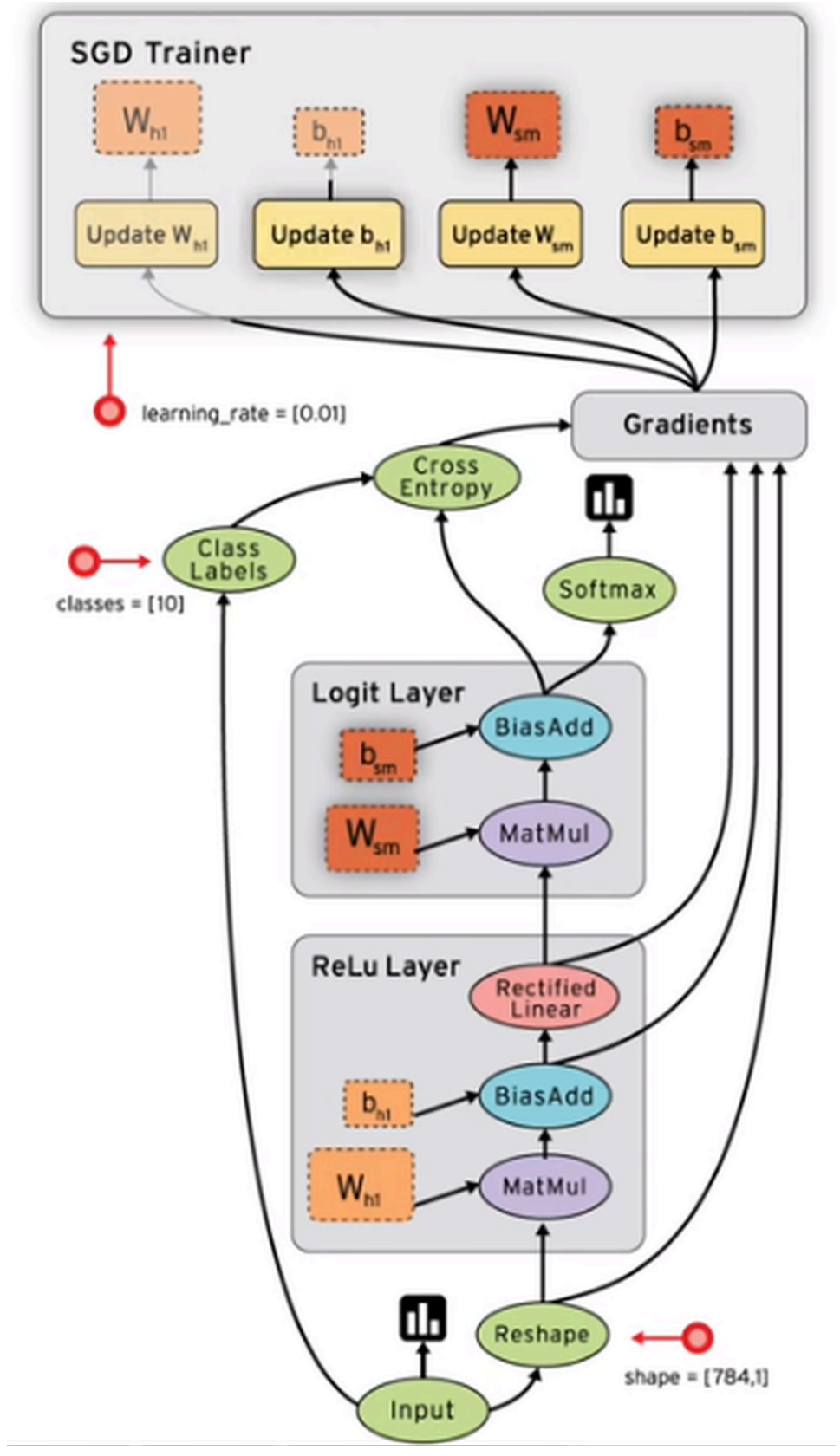
"TensorFlow™ is an open source software library for numerical computation using data flow graphs."

- 머신러닝 & 딥러닝을 **위한** 고성능 수치 계산 라이브러리
- 한 개 이상의 CPU 혹은 **GPU** 사용 가능
- Google 천재들 중의 천재들이 모인 **Google Brain** 팀이 머신 러닝과 딥러닝을 위해 개발

Data Flow Graph

- **Node**
  Mathematical operations

- **Edge**
  multidimensional data arrays (tensors)



GitHub Showcases: Machine Learning

tensorflow / **tensorflow**

Computation using data flow graphs for scalable machine learning

C++ ★ 36,820 16,618 Updated 43 minutes ago

독보적 1위

scikit-learn / **scikit-learn**

scikit-learn: machine learning in Python

Python ★ 14,674 8,314 Updated 5 hours ago

BVLC / **caffe**

Caffe: a fast open framework for deep learning.

C++ ★ 13,801 8,509 Updated 18 hours ago

TensorFlow의 자료형

- TensorFlow의 **모든 데이터는 tensor**로 표현됨

- Tensor를 n 차원 배열 혹은 리스트라 하여도 무방

- Tensor는 다음과 같은 속성들을 가짐:
  - **Rank**: tensor의 차원 수
  - **Shape**: 차원의 모양
  - Data types: e.g. int, float, double…

Rank, Shape: Example

Python Code	Rank	Shape	Math Entity
483	0	[]	Scalar
[1.1, 2.2, 3.3]	1	[3]	Vector
[[1, 2, 3], [4, 5, 6]]	2	[2, 3]	Matrix
[[[2], [4], [6]], [[8], [10], [12]]]	3	[2, 3, 1]	3-Tensor
...	n	[D0, D1, ⋯ , Dn-1]	n-Tensor

Tutorial 목차

- **1. Hello, TensorFlow!**
  3X4 예제
- **2. Linear Regression**
  이해하기 쉬운 linear regression을 통해 학습 과정 경험
- **3. MNIST with Single Layer NN**
  단일 계층 신경망으로 딥러닝 입문
- **4. MNIST with CNN**
  정확도가 99.3%인 CNN 구현
- **5. MNIST with CNN using TFLearn**
  동일한 CNN을 TFLearn으로 훨씬 쉽게 구현

1. Hello, TensorFlow!

- 초간단 3 X 4 예제

- TensorFlow는 **lazy evaluation** 사용

- 전체적인 TensorFlow 코드의 흐름은 다음과 같다:
  - Tensor들의 flow 구축 (알고리즘 설계)
  - Input을 넣고 구축한 tensor flow를 실행

2. Linear Regression

- **Hypothesis**
  `y = W*x + b`

- **Cost function**
  Mean Square Error(MSE)

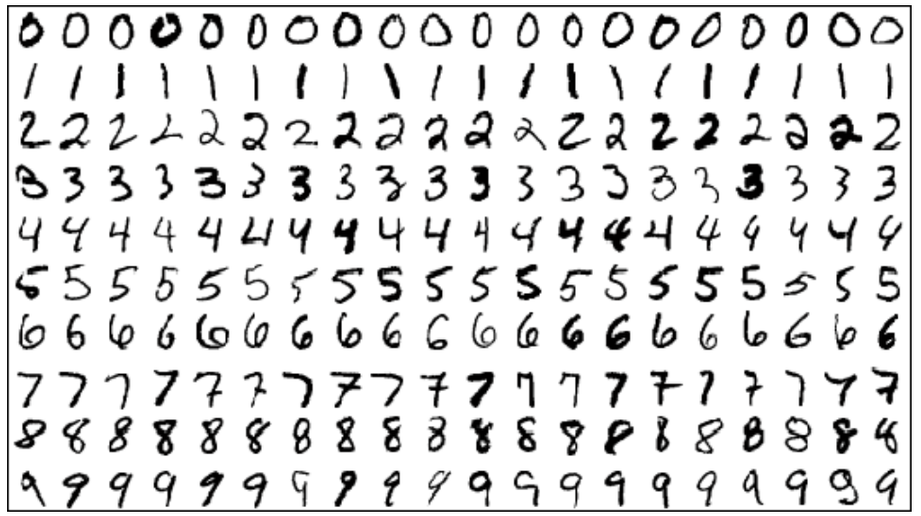
- **Optimizatoin Algorithm**
  Gradient Descent

- 1000개의 샘플 데이터를 랜덤 생성하여 사용

3. MNIST with Single Layer NN

MNIST

- 아라비아 **숫자 손글씨** (0 ~ 9)



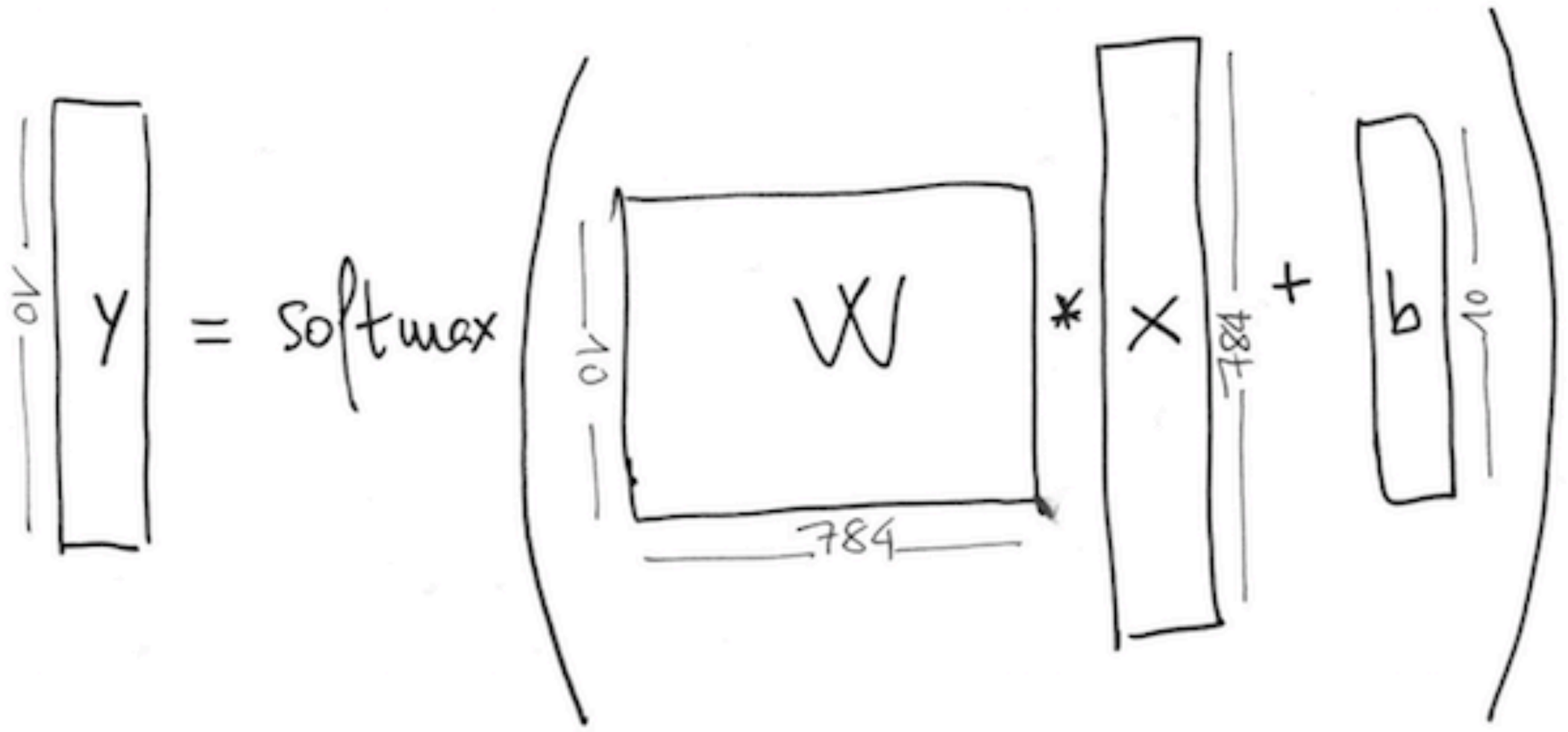
- 데이터 분포:
  - #Training = 55000
  - #Test = 10000

- 가로, 세로의 픽셀 수 = 28 (28X28 크기의 이미지)

- 데이터는 28X28 = **784 크기의 배열**

- 즉, training image set은 크기가 55000X784인 2차원 배열

Single Layer NN for MNIST



(주의) 실제 구현은 행과 열이 반대

3. MNIST with Single Layer NN

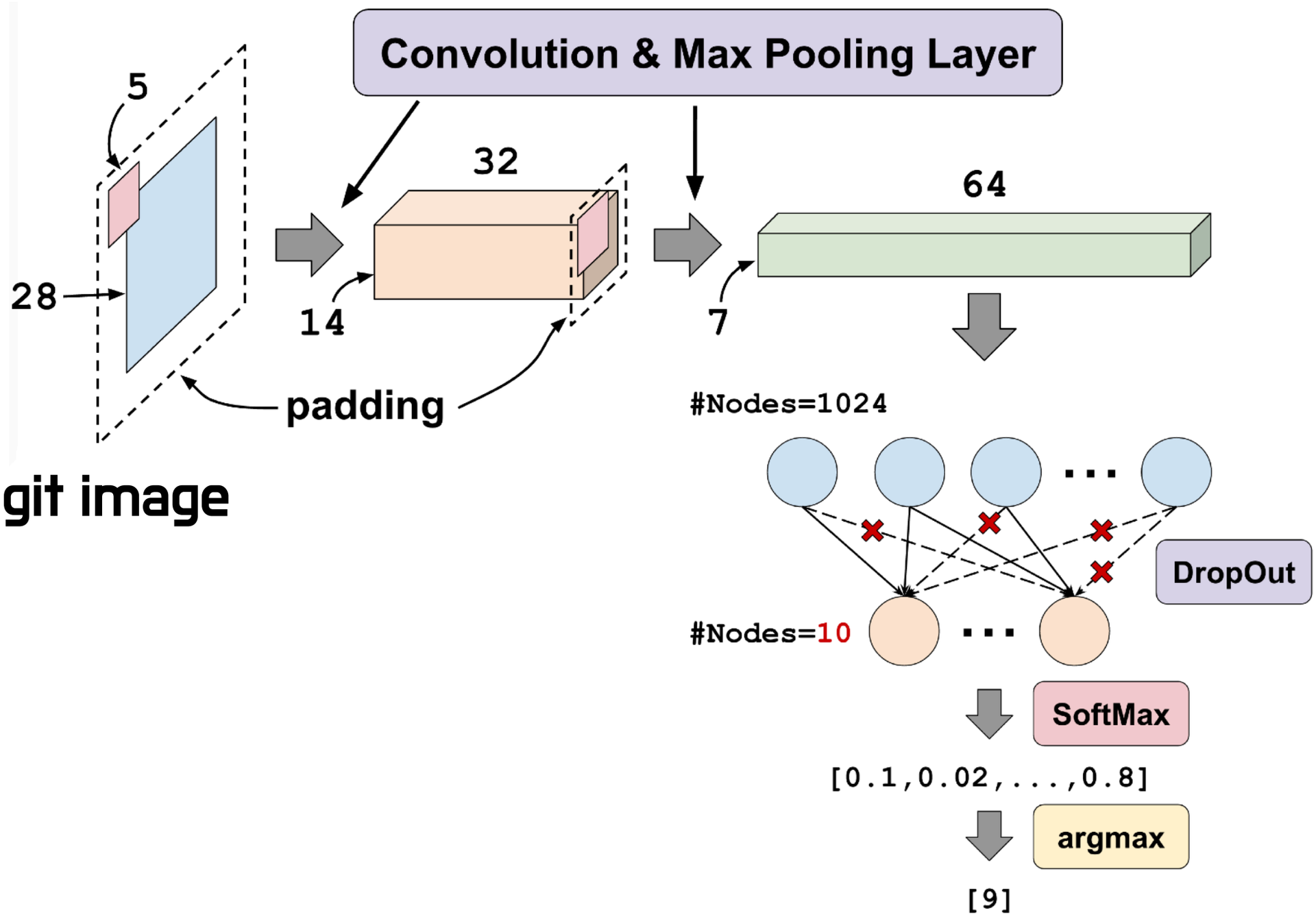
- 약 91%의 정확도
- Multi-class classification (#class = 10)
- **Cost function**
  Cross Entropy
- **Optimizer**
  Stochastic Gradient Descent
  - batch size = 100

4. MNIST with CNN

MNIST with CNN

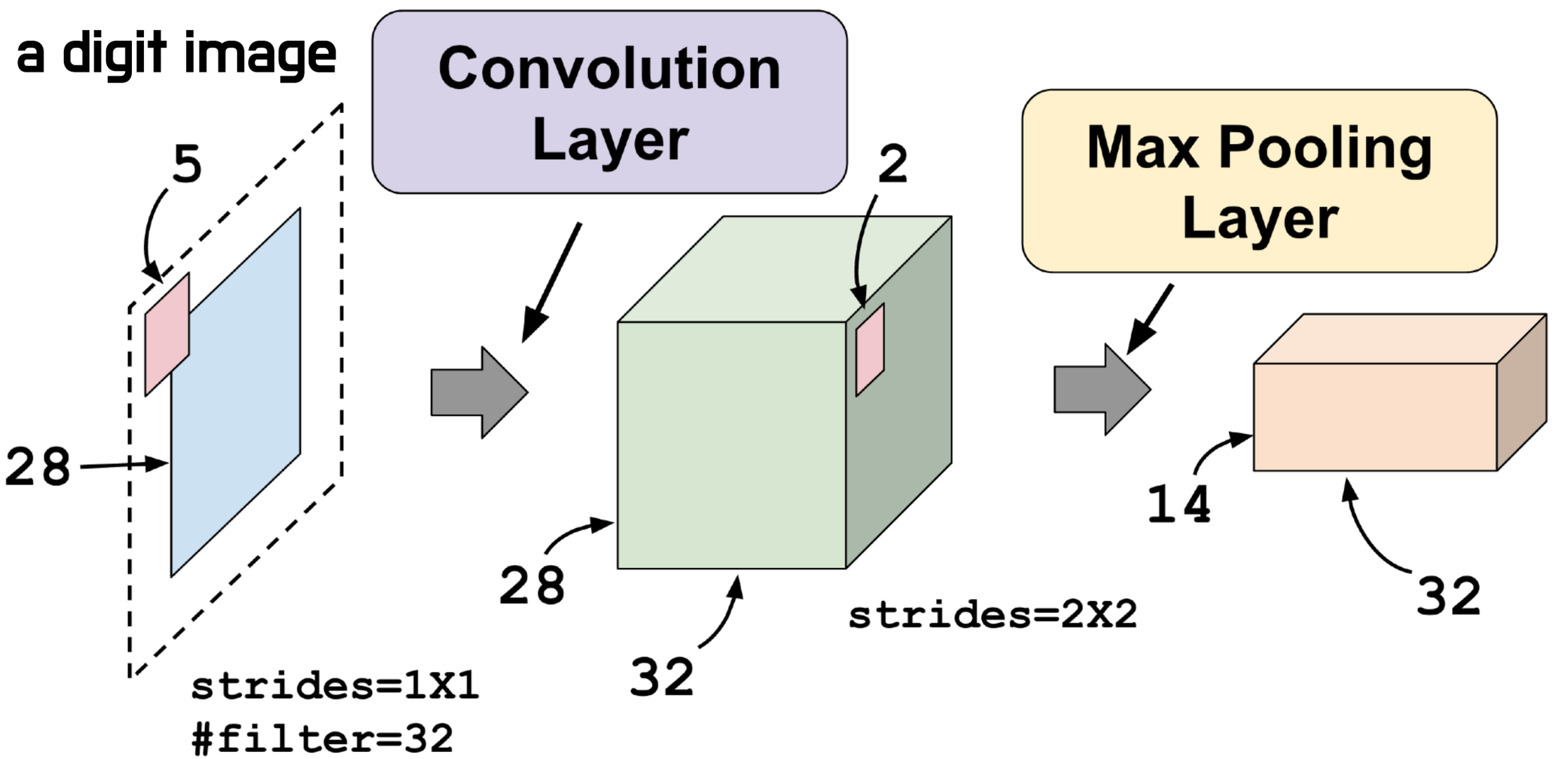
- Layers:
  1. input
  2. convolution & max pooling (#filter=32, size=5X5)
  3. convolution & max pooling (#filter=64, size=5X5)
  4. fully connected (#out_node=1024)
  5. drop out (keep_probability=0.5)
  6. fully connected (#out_node=10, activation='softmax')

NN for MNIST Overview



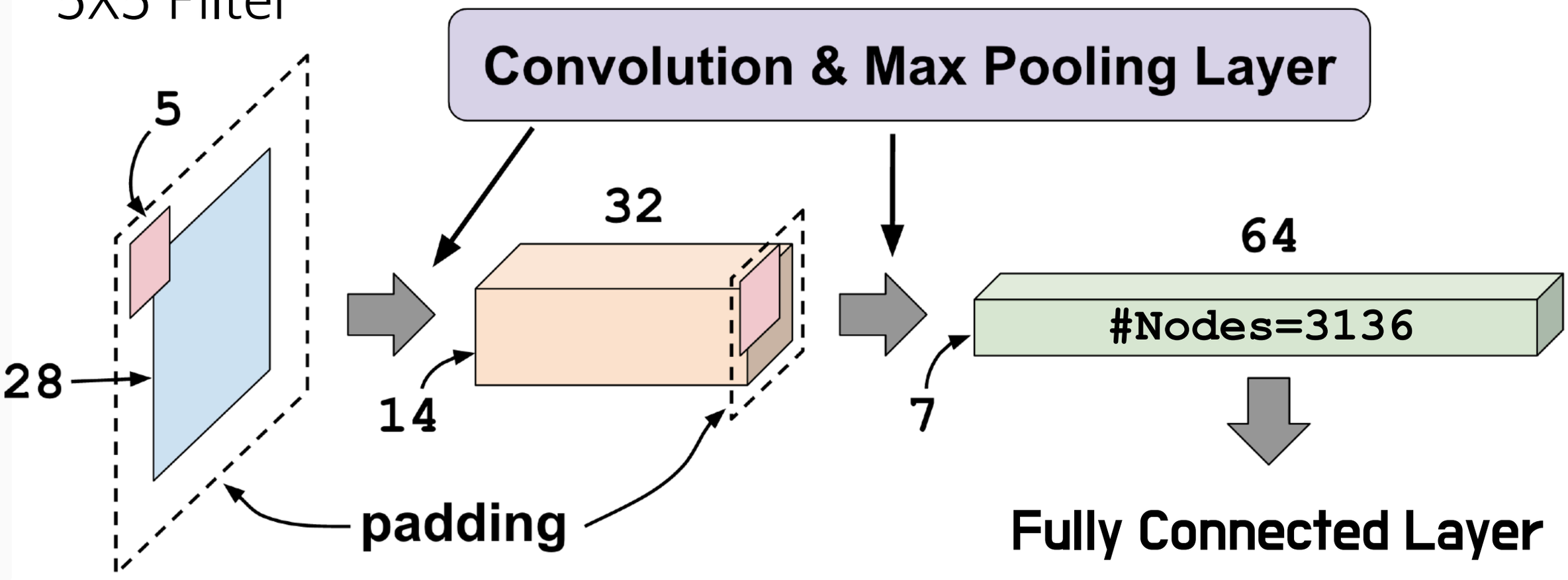
a digit image

Convolution & Max Pooling

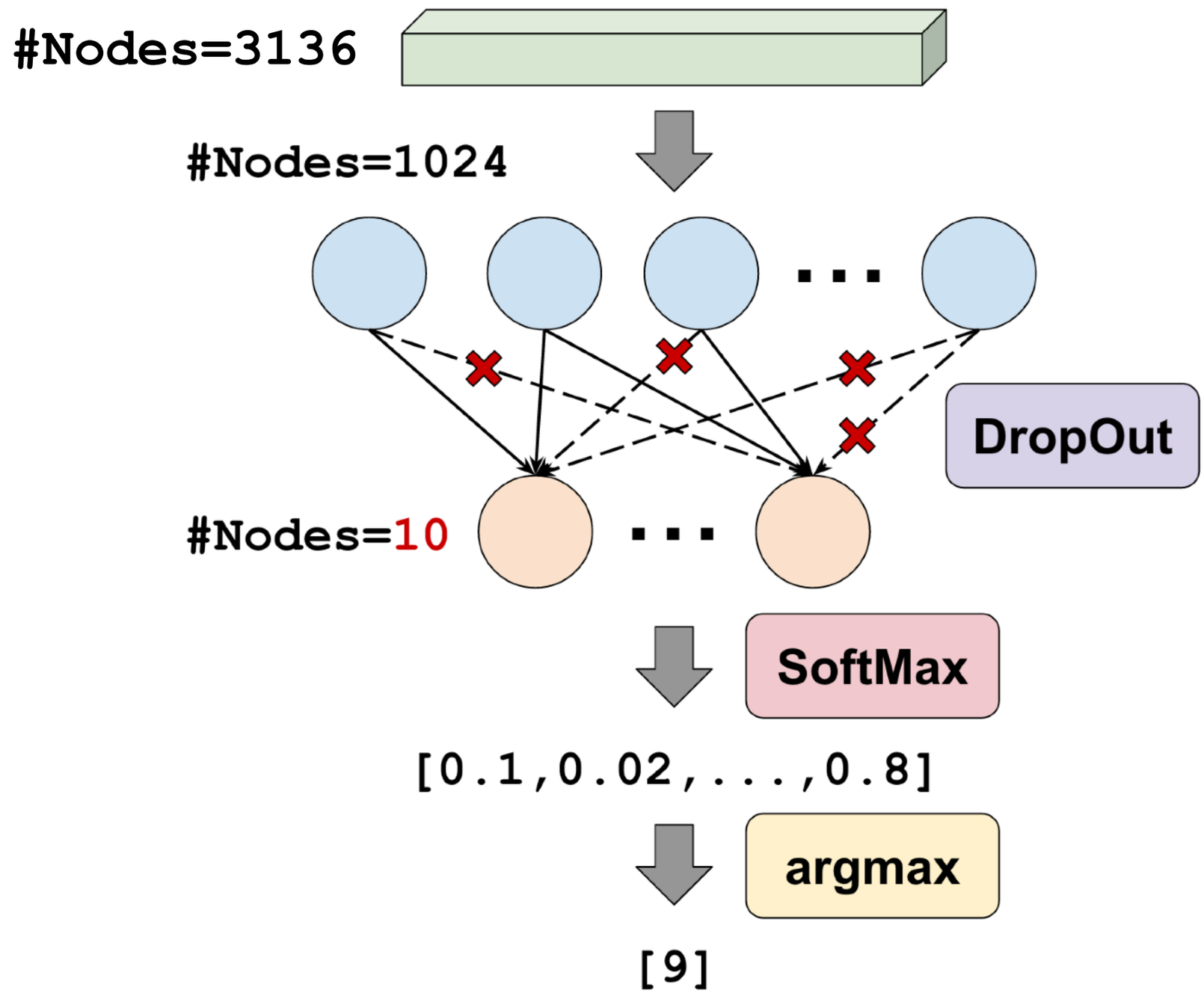


Hidden Layer

5X5 Filter

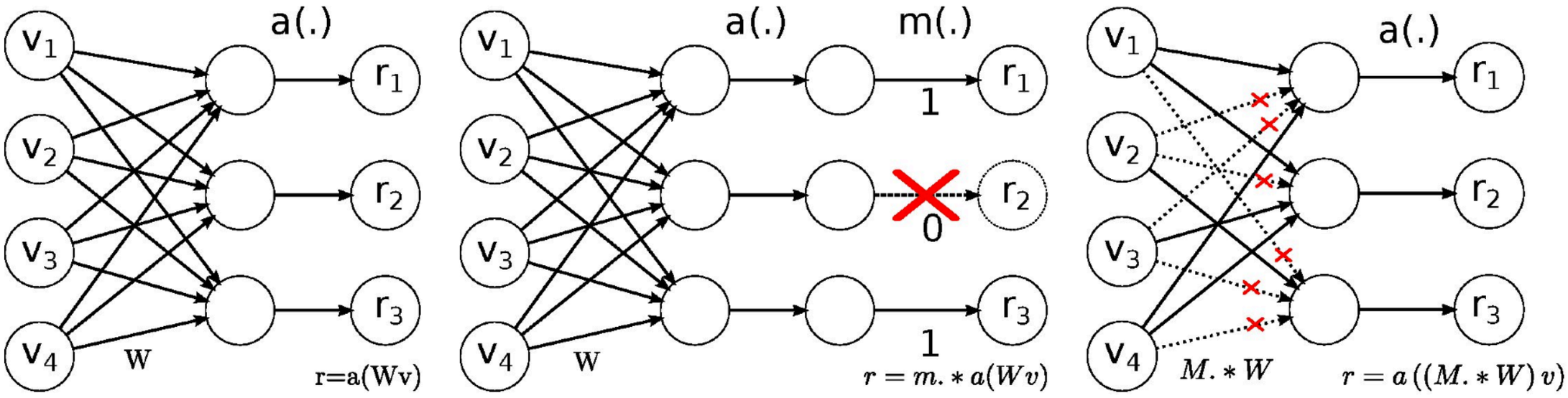


Fully Connected & DropOut



참고!

DropOut vs. DropConnect



No-Drop Network **DropOut Network** **DropConnect Network**

- MNIST 분류 정확도 1위는 DropConnect (99.79%)
- DropOut은 **layer 간**의 연결을 끊는 것
  DropConnect는 **layer 내의 cell 간**의 연결을 끊는 것

4. MNIST with CNN

- 약 99.3%의 정확도
- **Cost function**
  Cross Entropy
- **Optimizer**
  ADAM
  - batch size = 50
- **Regularization**
  DropOut
  - Keep Probability = 0.5
- **Convolution Layer**
  - Filter Size = 5
  - #Filters = 32
  - Padding = Same with input
  - Strides = [1, 1, 1, 1]
- **Max Pooling Layer**
  - Filter Size = 2
  - Strides = [1, 2, 2, 1]

Deep Learning을 위한 TensorFlow에 대한 제 생각

- TensorFlow는 정말 훌륭한 라이브러리이지만, 데이터 분석가에게는 복잡함

- TensorFlow 예제도 한단계 추상화하여 사용

- TensorFlow로 **딥러닝**을 쉽게하려면 **한 단계 더 추상화**할 필요가 있음

- 함수 하나로 layer 하나를 생성하여 레고처럼 layer들을 조립할 수 있으면 좋겠다!

5. MNIST with CNN using **TFLearn**

"TFLearn: Deep learning library featuring a higher-level API for TensorFlow."

- Layer를 레고 조립하듯이 이어 붙여나감
- 각 layer의 세부 설정을 option으로 줄 수 있음
- 직접 이전의 4. MNIST with CNN과 동일한 NN을 TFLearn을 사용하여 구현해봄

TensorBoard

Visualizing Learning

- TensorFlow 프로그램들을 쉽게 이해, 디버깅, 최적화하기 위한 시각화 도구

백문이불여일견, 일단 보자

TensorFlow 분산

- **TensorFrames**
  Spark + TensorFlow
  https://github.com/databricks/tensorframes

- **Distributed TensorFlow**
  TensorFlow가 자체적으로 만든 TensorFlow 분산 버전
  https://www.tensorflow.org/versions/r0.11/how_tos/distributed/index.html

TensorFlow 연구실 활용 방안

- 사용해보니, 2015년 맥북 프로 고급형에서 돌렸는데 매우 느림, CPU만으론 역부족

- 워크 스테이션급 서버에 GPU 2-3개 꽂고,

- JupyterHub(Multi-user server for Jupyter notebooks)를 설치하여 연구원 모두 함께 사용하자!

감사합니다

김태준(Jun Kim)
i2r.jun@gmail.com