

심층 신경망

Deep Neural Network (DNN)

MNIST를 이용한 NN 실습

-데이터 셋 소개-

MNIST Dataset 소개



학습데이터: 60000x784

학습데이터 라벨: 60000x10

테스트데이터: 10000x784

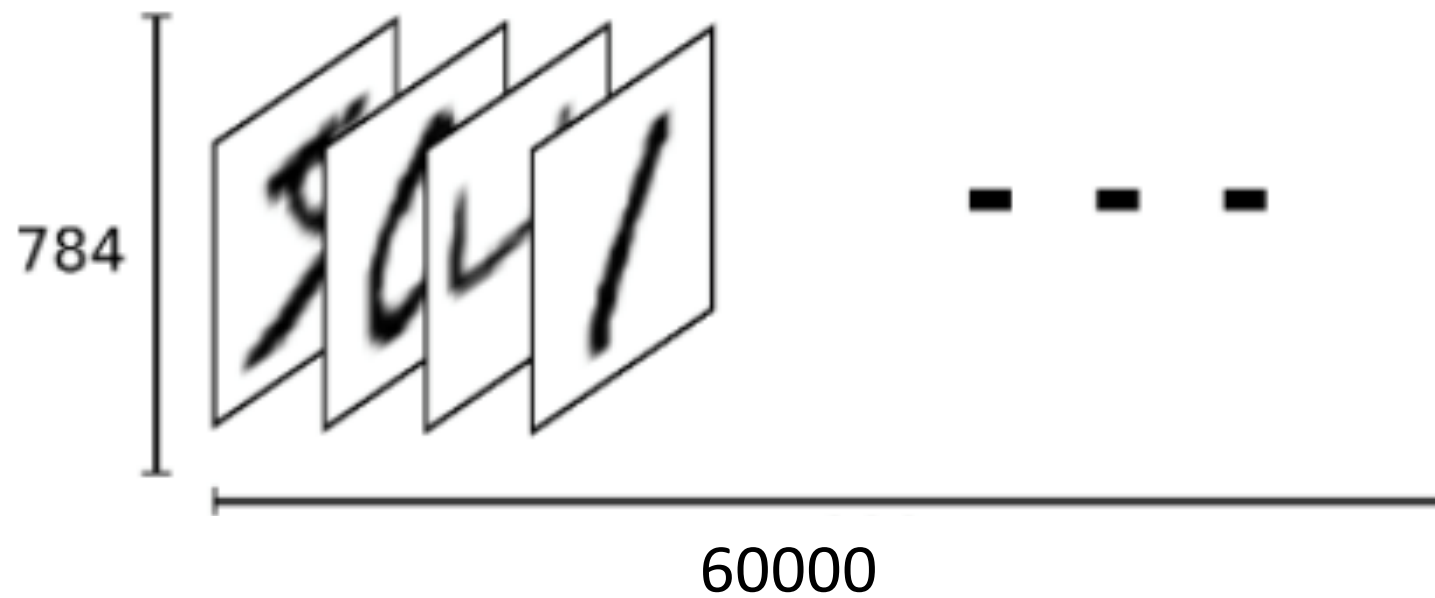
테스트데이터 라벨: 10000x10

MNIST Dataset 소개

[illegible]

MNIST Dataset 소개

mnist.train



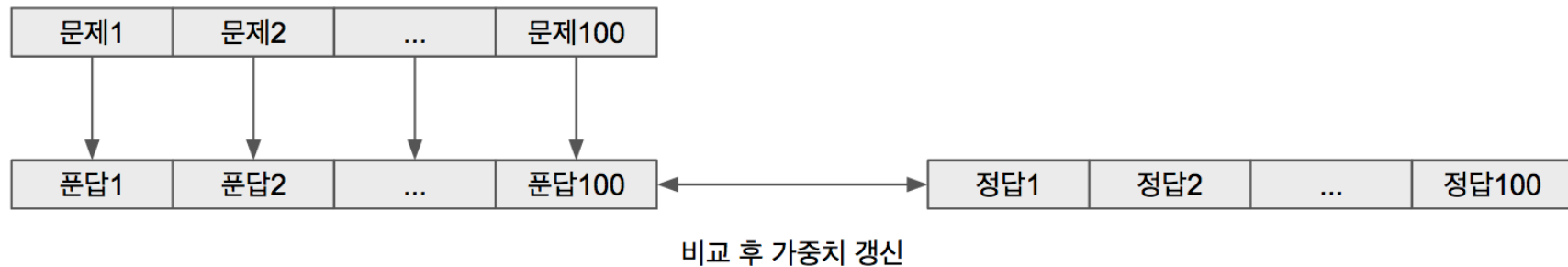
MNIST를 이용한 NN 실습

-사전 지식 쌓기-

Batch_size (배치사이즈)

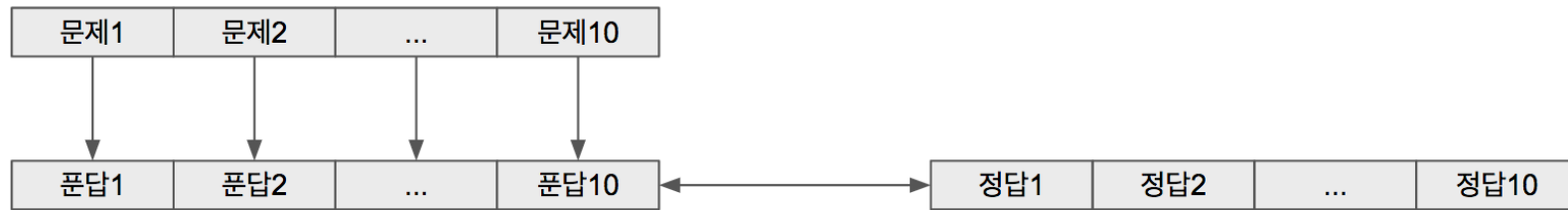
- 배치사이즈는 몇 문항을 풀고 해답을 맞추는 지를 의미
- 100문항일 때, 배치사이즈가 100이면 전체를 다 풀고 난 뒤에 해답을 맞춰보는 것
- 우리가 해답을 맞춰볼 때 ‘아하, 이렇게 푸는구나’ 라고 느끼면서 학습하는 것처럼 모델도 이러한 과정을 통해 가중치가 갱신

전체 문제를 푼 뒤 해답과 맞추므로 이 때 가중치 갱신은 한 번만 일어납니다.



Batch_size (배치사이즈)

배치사이즈가 10이면 열 문제씩 풀어보고 해답 맞춰보는 것입니다.
100문항을 10문제씩 나누어서 10번 해답을 맞추므로 가중치 갱신은 10번 일어납니다.



비교 후 가중치 갱신



비교 후 가중치 갱신

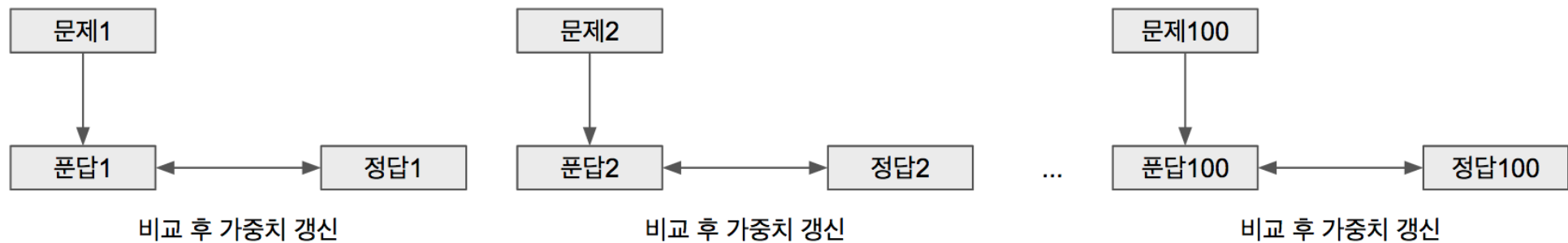
...



비교 후 가중치 갱신

Batch_size (배치사이즈)

배치사이즈가 1이면 한 문제 풀고 해답 맞춰보고 또 한 문제 풀고 맞춰보고 하는 것입니다.
한 문제를 풀 때마다 가중치 갱신이 일어나므로 횟수는 100번입니다.

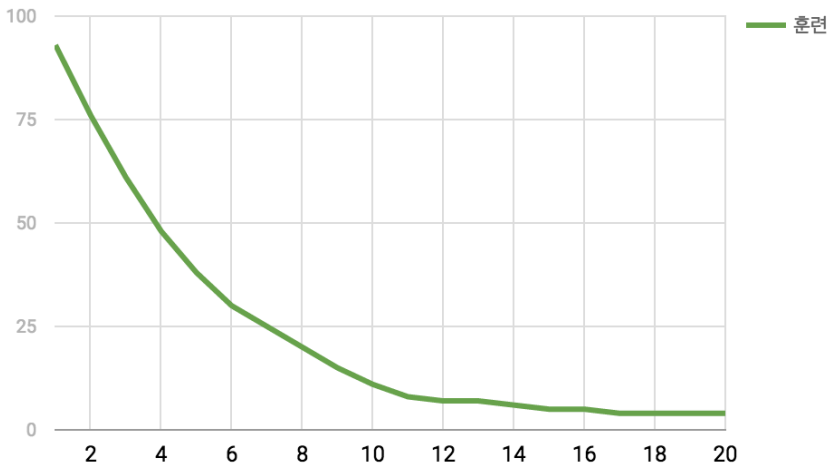


Batch_size = 100 vs Batch_size = 1

- **100문제 다 풀고 해답을 맞히는 것과 1문제씩 풀고 해답을 맞히는 것은 어떤 차이가 있을까요?**
 - 언뜻 생각해서는 별 반 차이가 없어 보이지만 모의고사 1회분에 비슷한 문항이 있다고 가정했을 때, 배치사이즈가 100일 때는 다 풀어보고 해답을 맞춰보기 때문에 한 문제를 틀릴 경우 이후 유사 문제를 모두 틀릴 경우가 많습니다.
 - 배치사이즈가 1인 경우에는 한 문제씩 풀어보고 해답을 맞춰보기 때문에 유사문제 중 첫 문제를 틀렸다고 하더라도 해답을 보면서 학습하게 되므로 나머지 문제는 맞추게 됩니다.
- **자 그럼 이 배치사이즈가 어떨 때 학습효과가 좋을까요?**
 - 사람이 학습하는 것이라 비슷합니다. 100문항 다 풀고 해답과 맞추어보려면 문제가 무엇이었는지 다 기억을 해야 맞춰보면서 학습이 되겠죠? 기억력(용량)이 커야합니다. 즉, GPU 메모리가 엄청 커야 합니다.
 - 1문항씩 풀고 해답 맞추면 학습은 꼼꼼히 잘 되겠지만 학습하는데 시간이 너무 걸리겠죠?

Epochs (에포크)

- 에포크는 모의고사 1회분을 몇 번 풀어볼까를 의미함
- 에포크가 20이면 모의고사 1회분을 20번 푸는 것.
- 우리가 같은 문제집을 여러 번 풀면서 점차 학습되듯이 모델도 같은 데이터셋으로 반복적으로 가중치를 갱신하면서 모델이 학습됨



그래프에서 세로축이 100문항 중 틀린 개수이고, 가로축이 모의고사 풀이 반복횟수를 의미함.

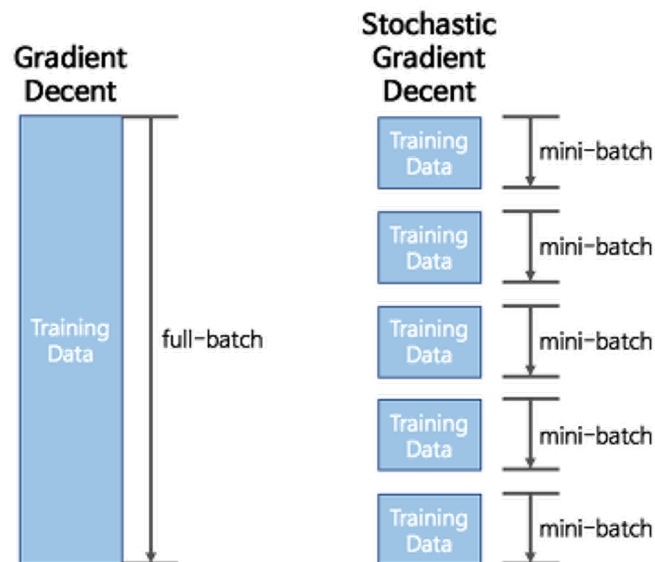
풀이를 반복할수록 틀린 개수가 적어지는 것을 보수 있음.

처음에는 틀린 개수가 확 줄어들지만, 반복이 늘어날수록 완만하게 틀린 오답 수가 줄어듬.

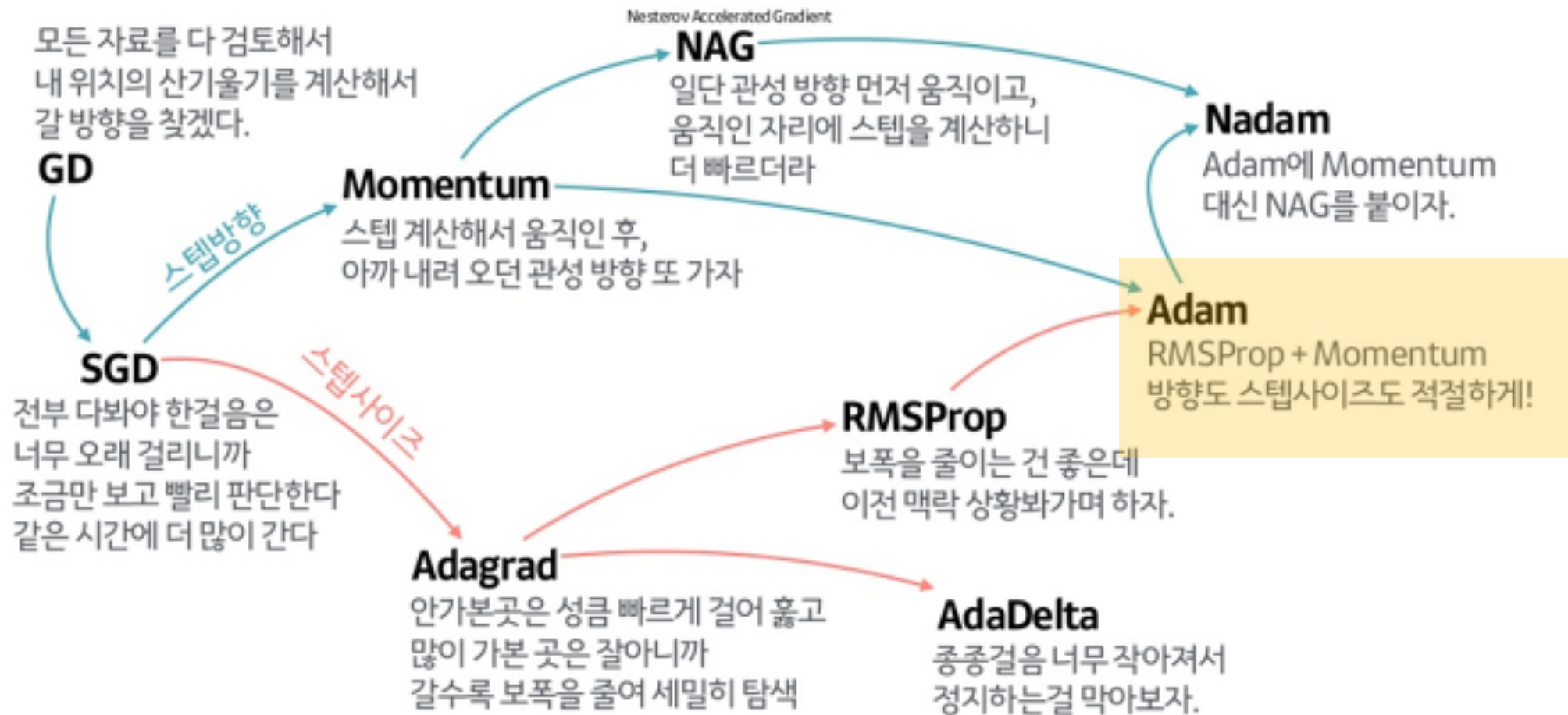
우리가 공부할 때도 낮은 점수에서는 공부를 조금하면 점수가 확 오르지만, 높은 점수에서 1~2점 올리는 것이 쉽지 않은 것과 비슷

Optimizer: 기본

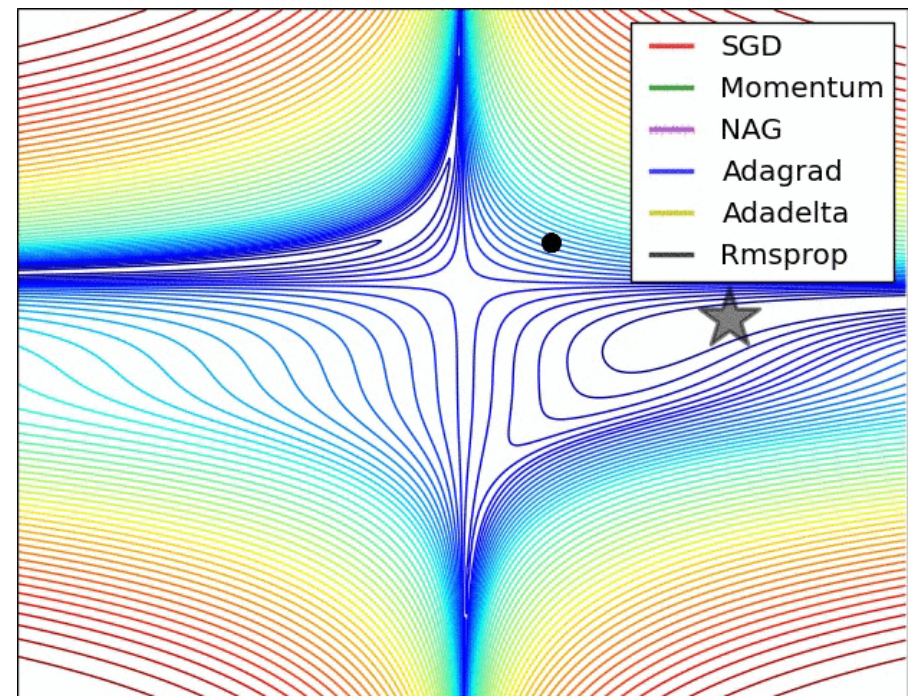
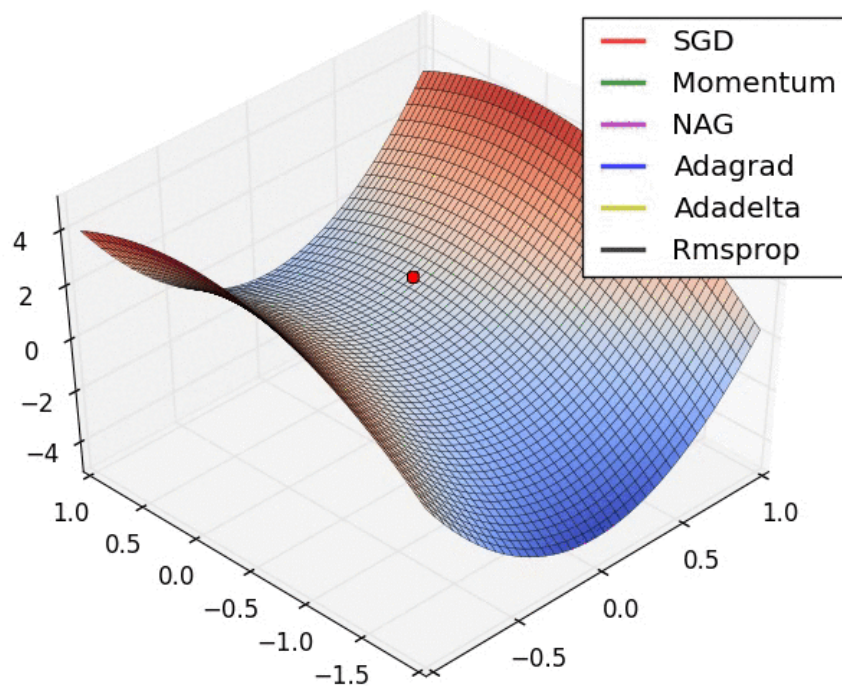
- **Gradient Descent (GD)**
 - 학습데이터 전체를 사용하는 최적화
- **Stochastic Gradient Descent (SGD)**
 - 학습데이터 일부(mini-batch)를 사용하는 최적화
 - BGD보다 빠르게 수렴
 - SGD를 여러 번 반복할 수록 BGD와 유사한 결과로 수렴



Optimizer : 발전 방향



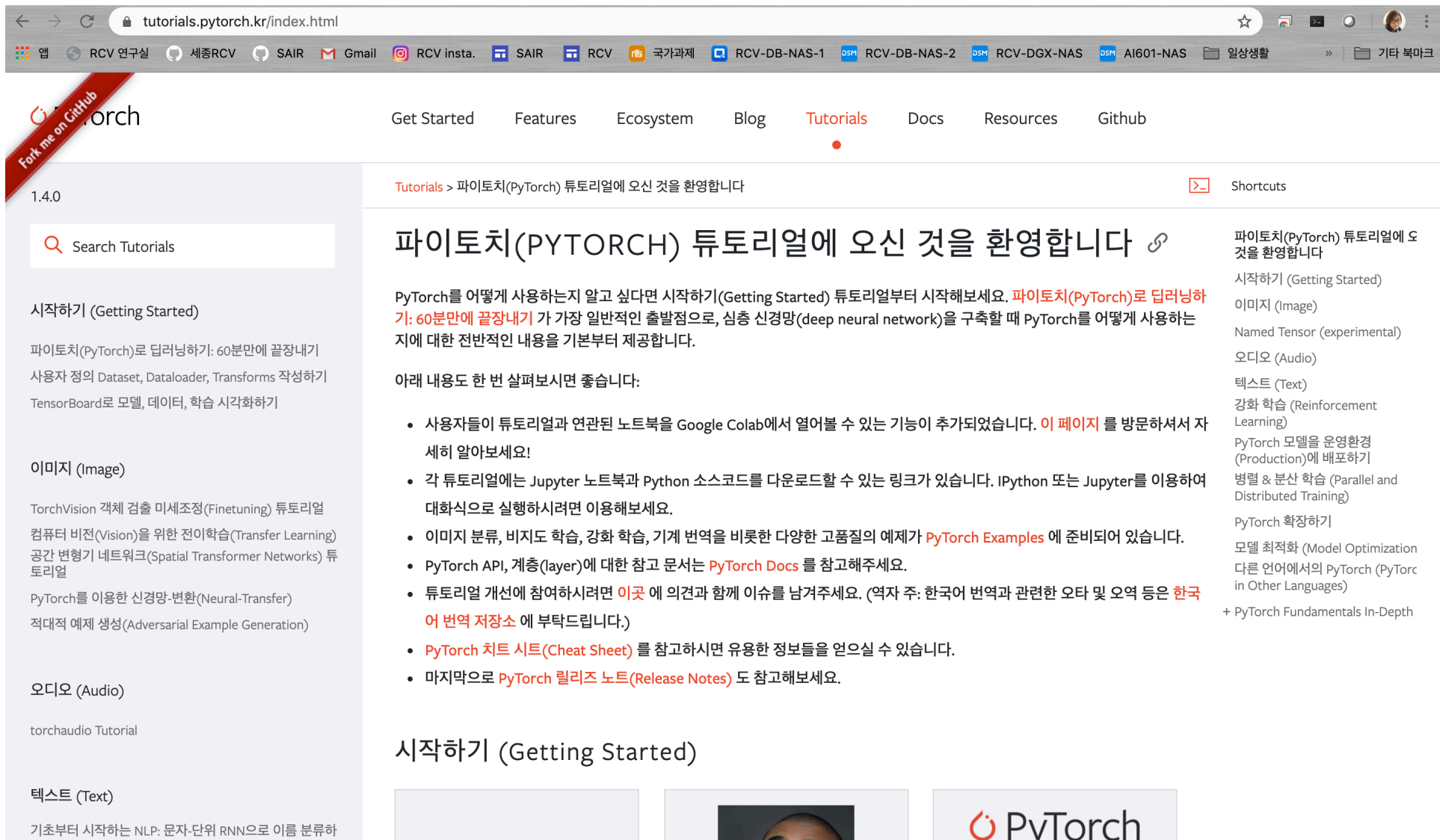
Optimizer: 비교



MNIST를 이용한 NN 실습

- 파이토치와 토치비전-

Pytorch (파이토치)



torchvision



pytorch torchvision



전체

이미지

동영상

지도

뉴스

더보기

설정

도구

검색결과 약 172,000개 (0.37초)

pytorch.org › docs › stable › torch... ▾ 이 페이지 번역하기

torchvision — PyTorch 1.5.0 documentation

The **torchvision** package consists of popular datasets, model architectures, and common image transformations for computer vision. Package Reference.

[Torchvision.models](#) · [Torchvision.datasets](#) · [Torchvision.transforms](#) · [Torchvision.utils](#)

함께 검색한 항목



torchvision이란	Torchvision vgg16
Torchvision github	Pytorch torchvision datasets mnist
Pytorch grayscale	Torchvision install

github.com › pytorch › vision ▾ 이 페이지 번역하기

pytorch/vision: Datasets, Transforms and Models ... - GitHub

<https://travis-ci.org/pytorch/vision.svg?> <https://pepy.tech/badge/torchvision> ... #645 [Master Issue]

Add more models to **torchvision** Opened by fmassa over 1 year ...

torchvision

- The [torchvision](https://pytorch.org/docs/stable/torchvision/index.html) package consists of popular datasets, model architectures, and common image transformations for computer vision.
- <https://pytorch.org/docs/stable/torchvision/index.html>

Package Reference

- `torchvision.datasets`
 - MNIST
 - Fashion-MNIST
 - KMNIST
 - EMNIST
 - QMNIST
 - FakeData
 - COCO
 - LSUN
 - ImageFolder
 - DatasetFolder
 - ImageNet
 - CIFAR
 - STL10
 - SVHN
 - PhotoTour
 - SBU
 - Flickr
 - VOC
 - Cityscapes

다양한 실험

0.3 / 0.9809

- ① Random Init / NN Layer #1 (784= \rightarrow 10) / **SGD** / CrossEntropy
 - <https://colab.research.google.com/drive/15ArrVdJELyV-PeuATH5poe2uZoHm6lh8>
- ② Random Init / NN Layer #1 (784= \rightarrow 10) / **Adam** / CrossEntropy
 - <https://colab.research.google.com/drive/1bimNZtout48XzSy7VoWbmdpOHQ-SAkHX>
- ③ Random Init / **NN Layer #3** (784= \rightarrow 256= \rightarrow 10) / Adam / CrossEntropy
 - <https://colab.research.google.com/drive/1jsP6lkoSZW5roMip53QSPJjiVP6Ej3Zk>
- Xavier** Init / NN Layer #3 (784= \rightarrow 256= \rightarrow 10) / Adam / CrossEntropy
 - <https://colab.research.google.com/drive/1CI11CA5otqB7-RsQLKak3LJW3xeCUr9q>
- Xavier Init / **DNN Layer #5** (784= \rightarrow 256= \rightarrow 10) / Adam / CrossEntropy
 - https://colab.research.google.com/drive/1y9qF3D4vbQVhX_dZCgplRuw_EcgW8Sg2
- Xavier Init / DNN Layer #5 (784= \rightarrow 256= \rightarrow 256= \rightarrow 256= \rightarrow 10) / Adam / CrossEntropy / **dropout (0.3)**
 - <https://colab.research.google.com/drive/15Q5GSAPgsHeR0Y70zaiigHBBo4kwJ86o>

SGD	Adam	NN	Xavier	DNN	Dropout
0.42	0.77	0.94	0.9791	0.9824	0.9776

END
