인공지능의 기초

6, 7주차

Today

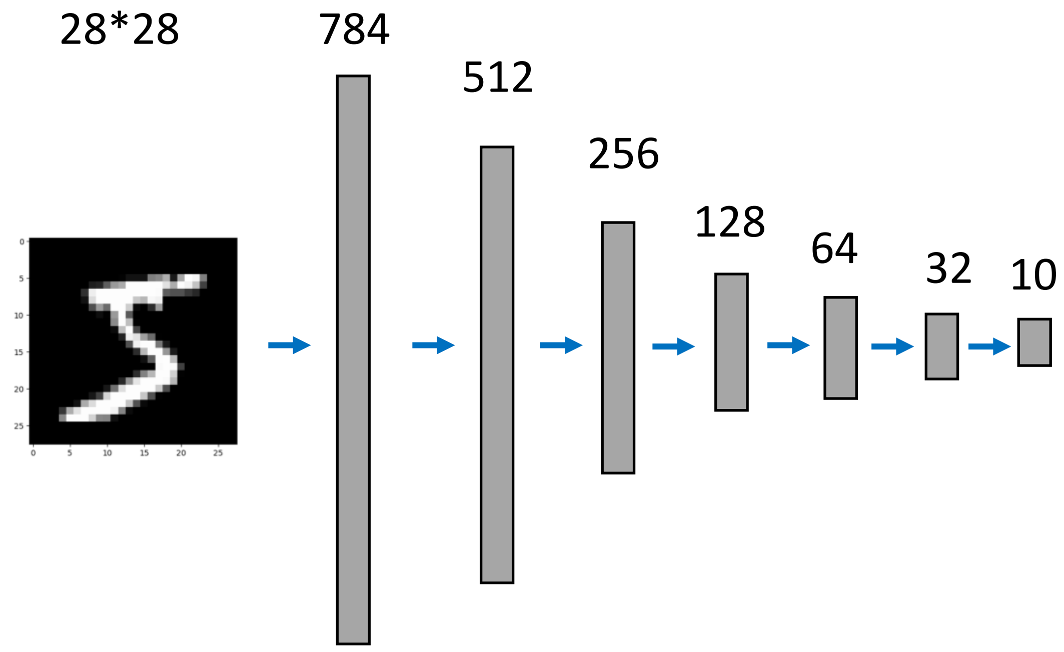
- CNN
- Mnist classification using CNN(실습)

- RNN/ Transformer
- LLM
- GAN/ Diffusion

- Hugging Face

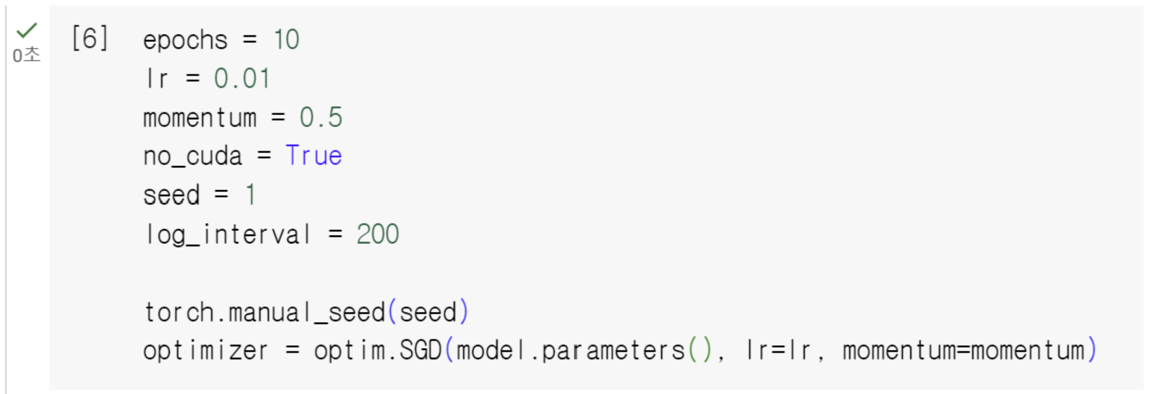
Review

- Mnist classification using DNN
- 최대 accuracy 70

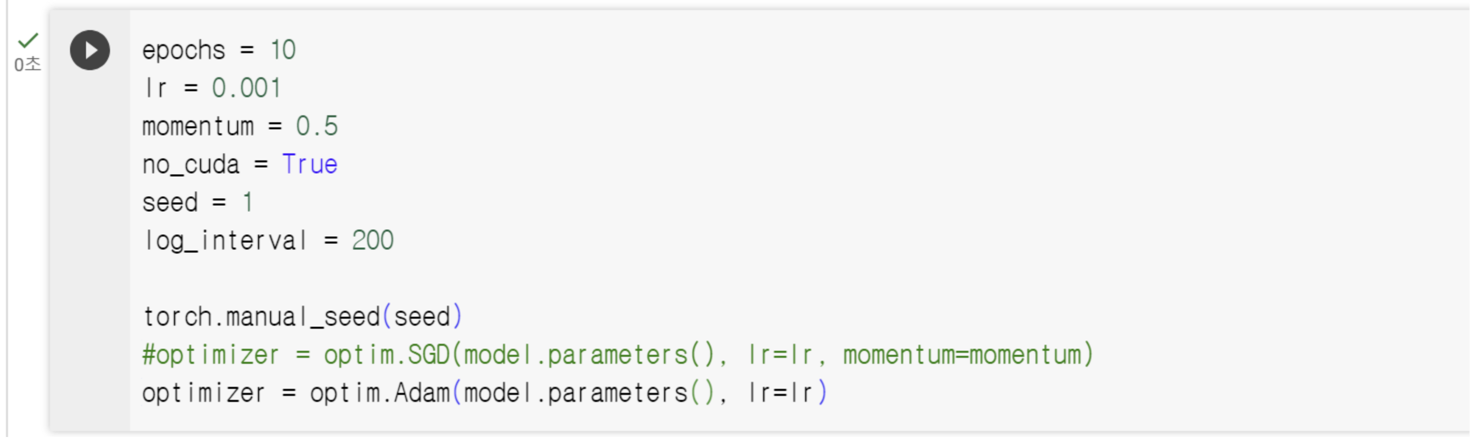


Review

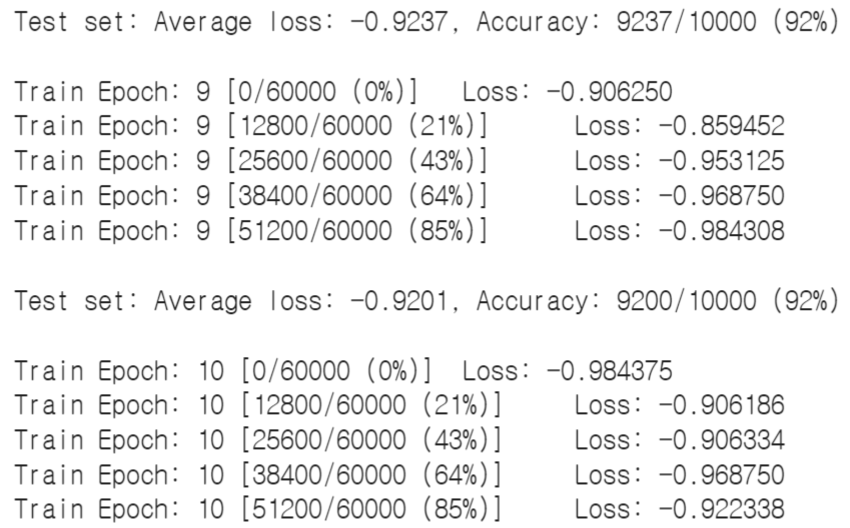
- 최대 accuracy 70 -> 92



```
[6]  epochs = 10
     lr = 0.01
     momentum = 0.5
     no_cuda = True
     seed = 1
     log_interval = 200

     torch.manual_seed(seed)
     optimizer = optim.SGD(model.parameters(), lr=lr, momentum=momentum)
```



```
epochs = 10
lr = 0.001
momentum = 0.5
no_cuda = True
seed = 1
log_interval = 200

torch.manual_seed(seed)
#optimizer = optim.SGD(model.parameters(), lr=lr, momentum=momentum)
optimizer = optim.Adam(model.parameters(), lr=lr)
```

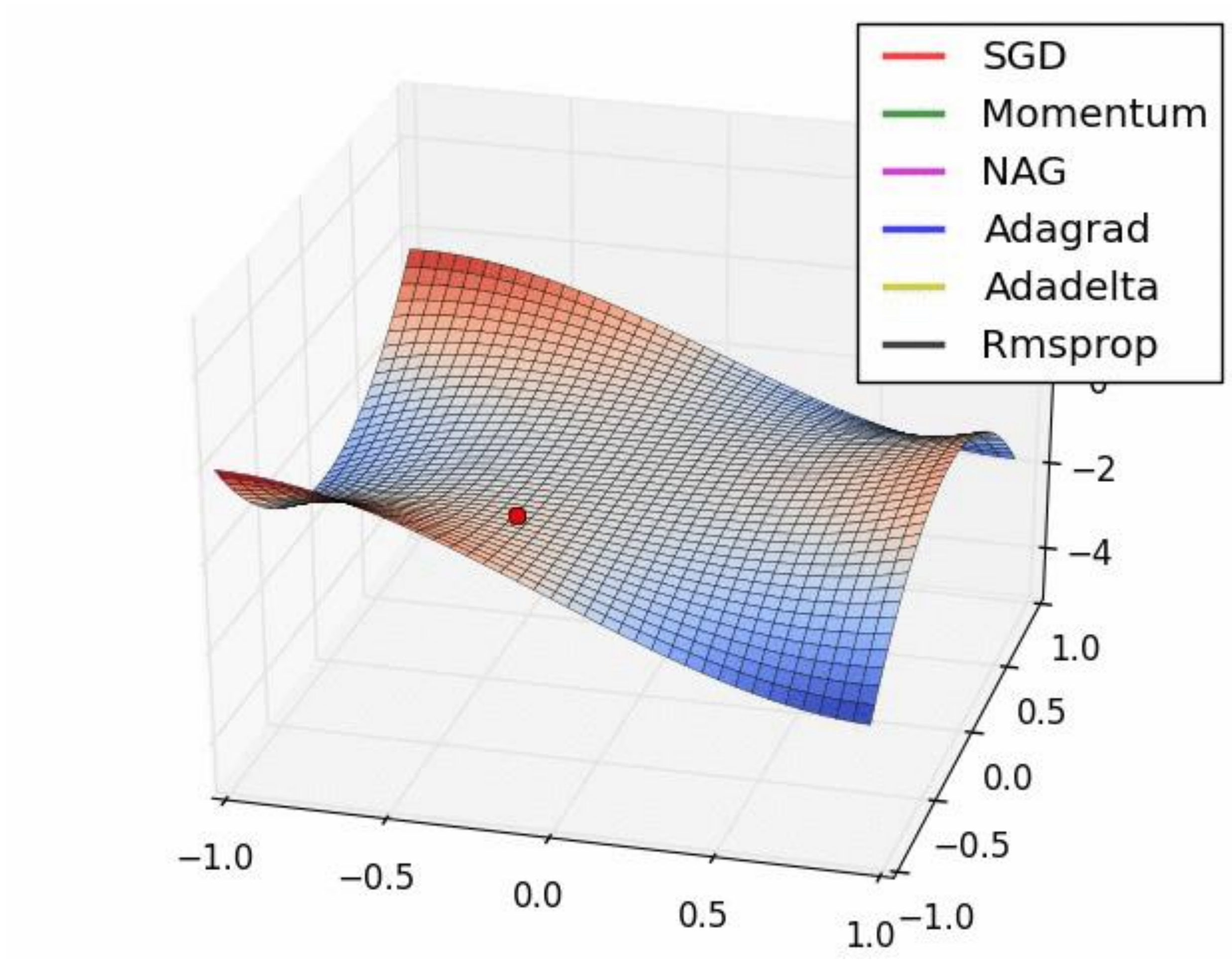


```
Test set: Average loss: -0.9237, Accuracy: 9237/10000 (92%)

Train Epoch: 9 [0/60000 (0%)]   Loss: -0.906250
Train Epoch: 9 [12800/60000 (21%)]      Loss: -0.859452
Train Epoch: 9 [25600/60000 (43%)]      Loss: -0.953125
Train Epoch: 9 [38400/60000 (64%)]      Loss: -0.968750
Train Epoch: 9 [51200/60000 (85%)]      Loss: -0.984308

Test set: Average loss: -0.9201, Accuracy: 9200/10000 (92%)

Train Epoch: 10 [0/60000 (0%)]  Loss: -0.984375
Train Epoch: 10 [12800/60000 (21%)]     Loss: -0.906186
Train Epoch: 10 [25600/60000 (43%)]     Loss: -0.906334
Train Epoch: 10 [38400/60000 (64%)]     Loss: -0.968750
Train Epoch: 10 [51200/60000 (85%)]     Loss: -0.922338
```

Review

- Optimizer 변경
- Learning rate 최적화

-> hyperparameter tuning

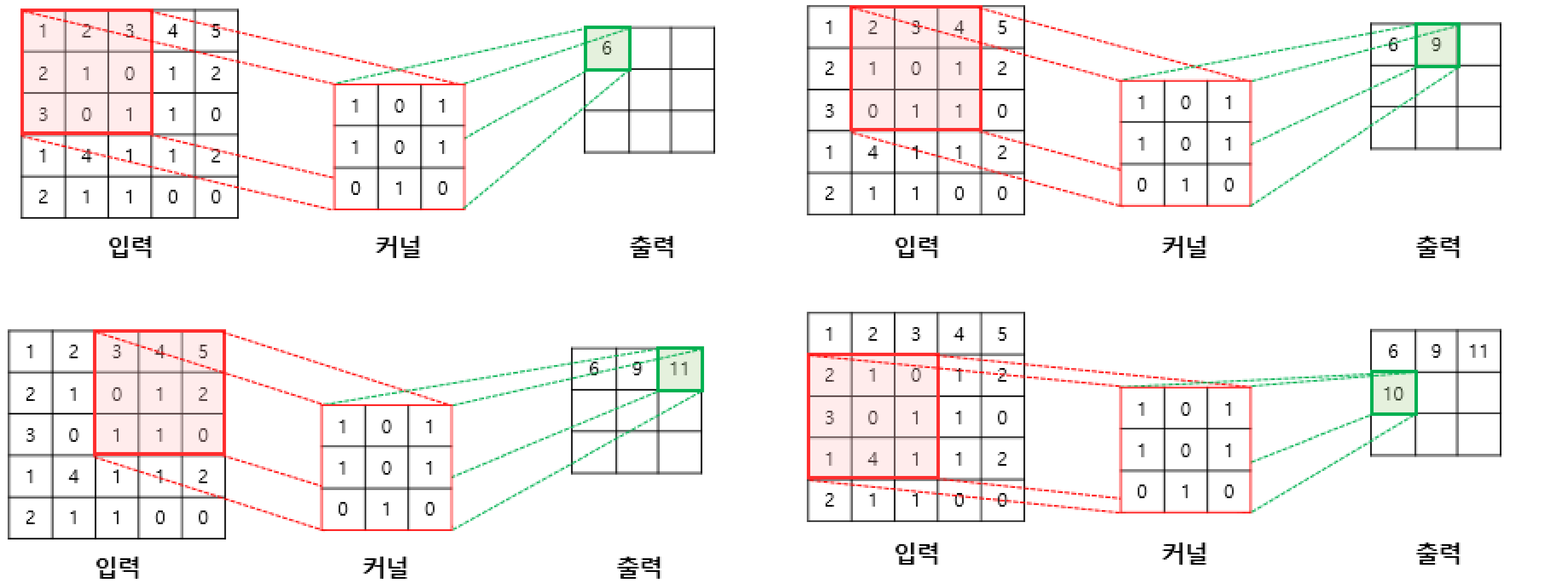


CNN

- Convolution Neural Network
- Image 특화

CNN

- Convolutional layer

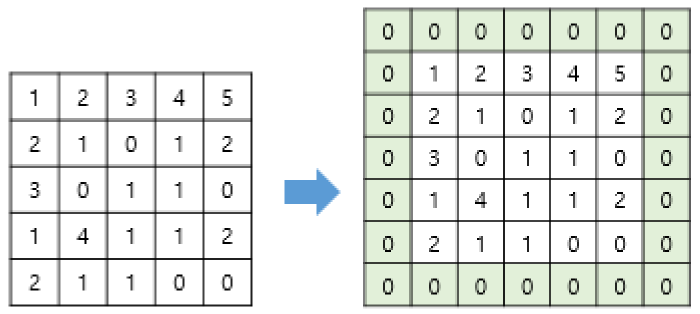


6	9	11
10	4	4
7	7	4

특성 맵(feature map)

CNN

- Padding layer
- Pooling

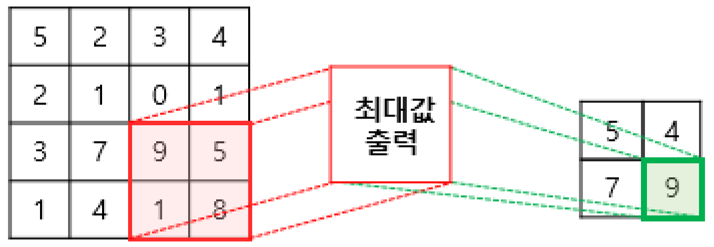


1	2	3	4	5
2	1	0	1	2
3	0	1	1	0
1	4	1	1	2
2	1	1	0	0

패딩 전

0	0	0	0	0	0	0
0	1	2	3	4	5	0
0	2	1	0	1	2	0
0	3	0	1	1	0	0
0	1	4	1	1	2	0
0	2	1	1	0	0	0
0	0	0	0	0	0	0

패딩 후

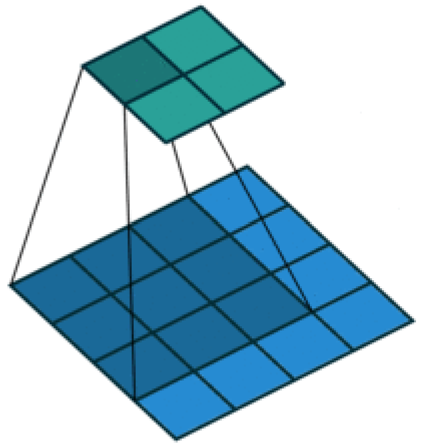
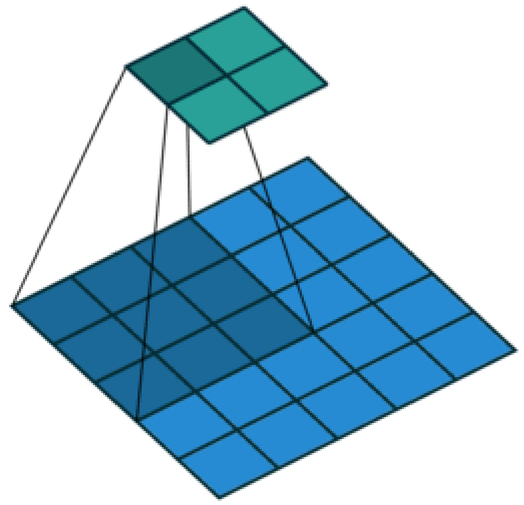
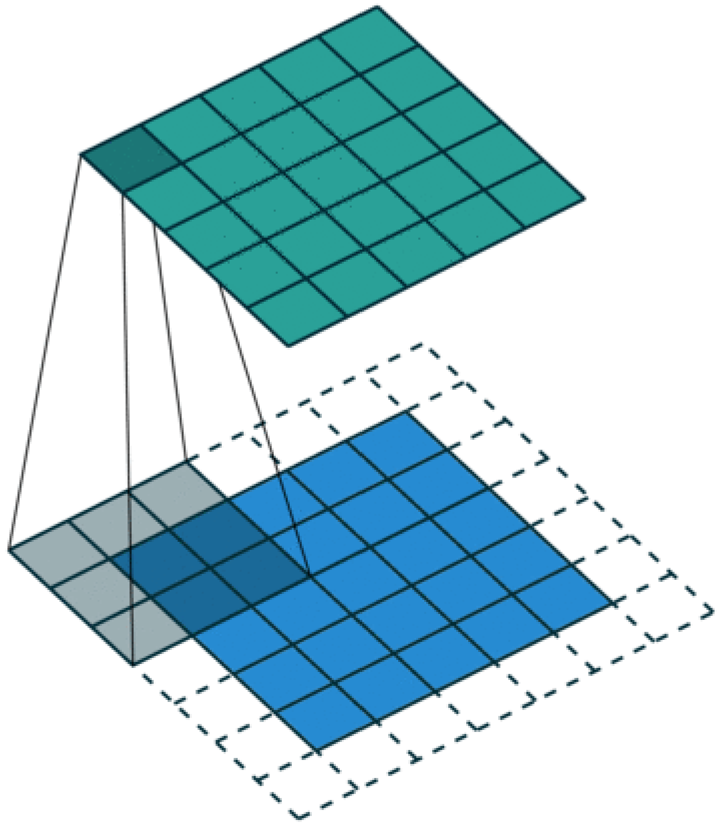


5	2	3	4
2	1	0	1
3	7	9	5
1	4	1	8

5	4
7	9

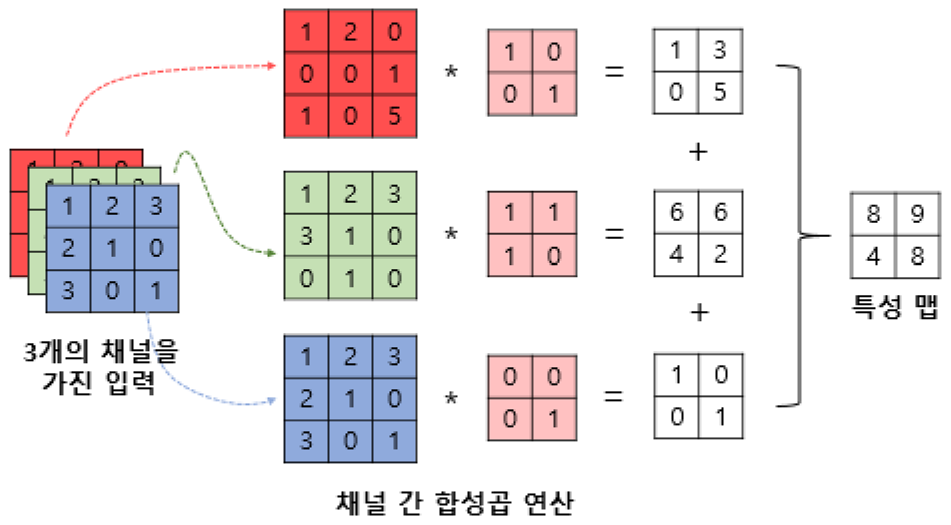
CNN

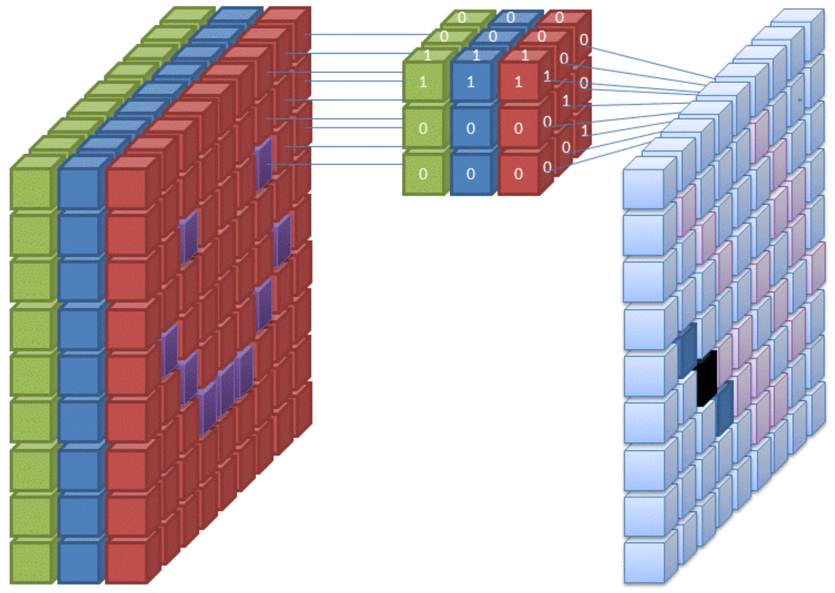
- No Padding
- No Padding, Stride=2
- Same Padding

CNN

- 3 Channel

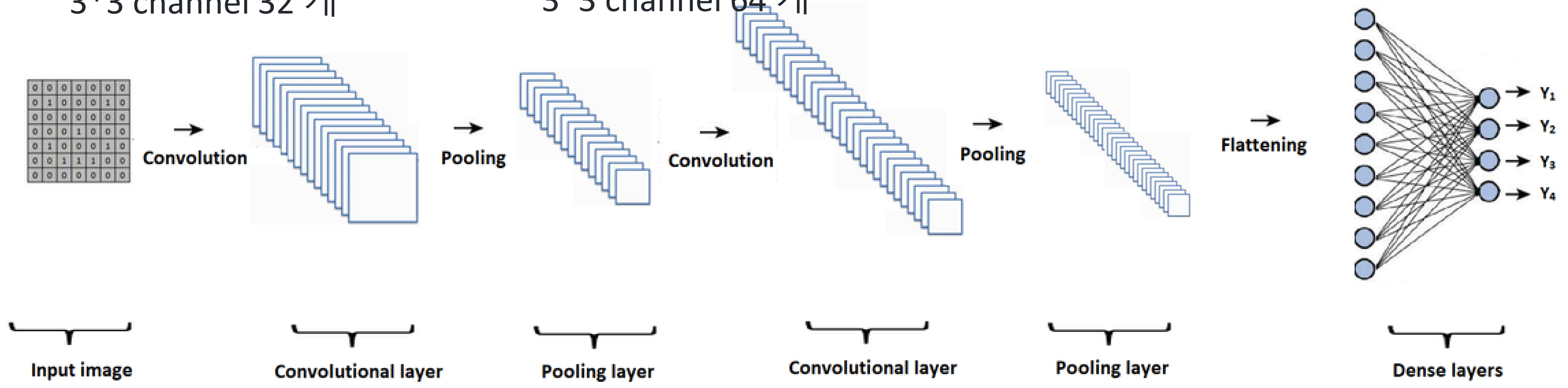




CNN

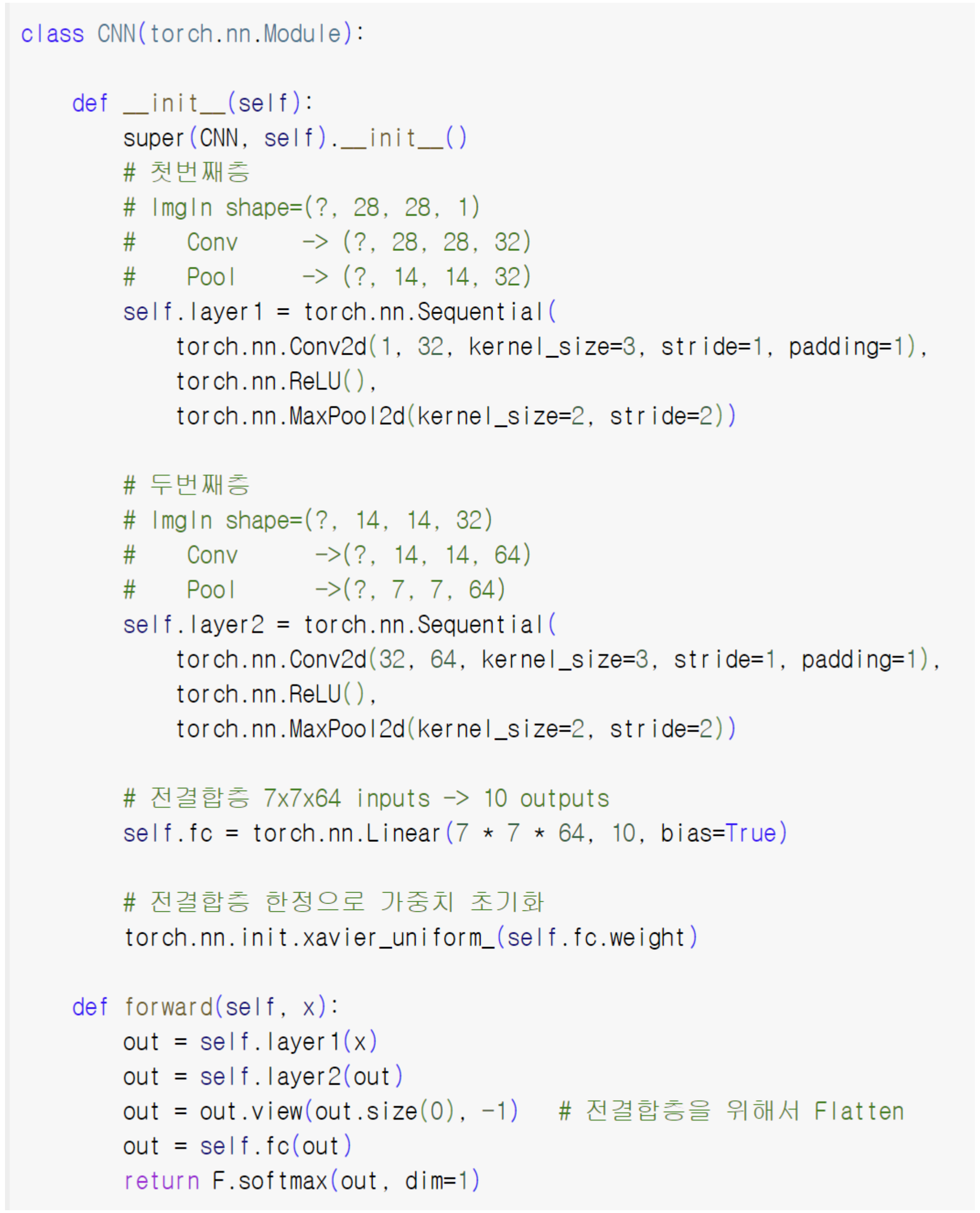
3*3 channel 32 개

3*3 channel 64 개



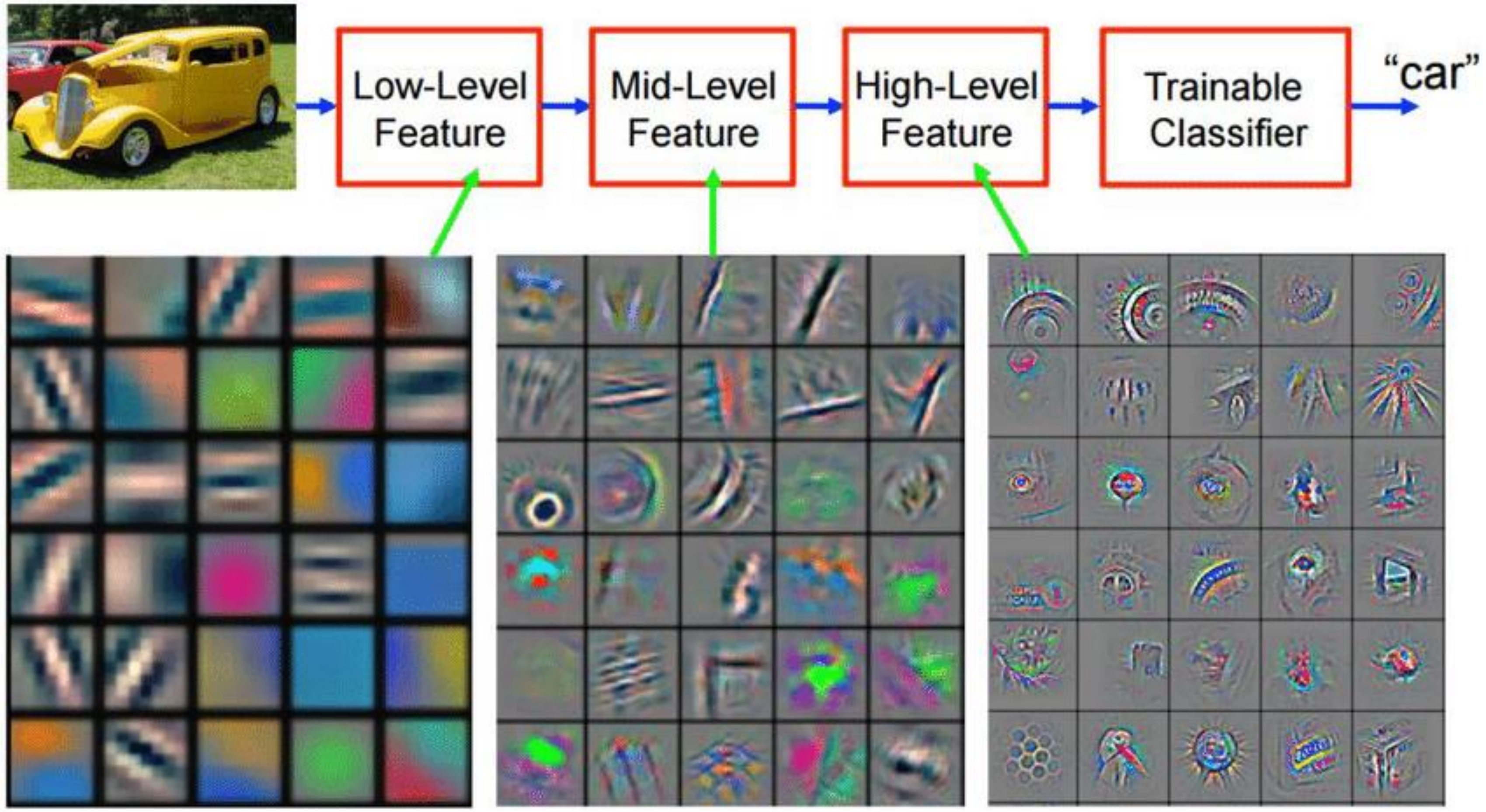
28, 28, 1 | 28, 28, 32 | 14, 14, 32 | 14, 14, 64 | 7, 7, 64 | 7*7* 64 | 10

CNN



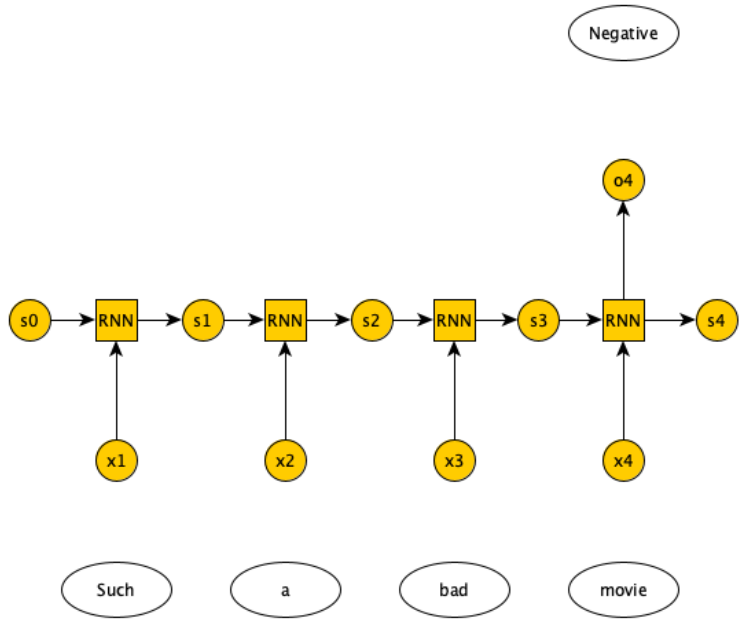
```python
class CNN(torch.nn.Module):

    def __init__(self):
        super(CNN, self).__init__()
        # 첫번째층
        # ImgIn shape=(?, 28, 28, 1)
        #    Conv     -> (?, 28, 28, 32)
        #    Pool     -> (?, 14, 14, 32)
        self.layer1 = torch.nn.Sequential(
            torch.nn.Conv2d(1, 32, kernel_size=3, stride=1, padding=1),
            torch.nn.ReLU(),
            torch.nn.MaxPool2d(kernel_size=2, stride=2))

        # 두번째층
        # ImgIn shape=(?, 14, 14, 32)
        #    Conv      ->(?, 14, 14, 64)
        #    Pool      ->(?, 7, 7, 64)
        self.layer2 = torch.nn.Sequential(
            torch.nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),
            torch.nn.ReLU(),
            torch.nn.MaxPool2d(kernel_size=2, stride=2))

        # 전결합층 7x7x64 inputs -> 10 outputs
        self.fc = torch.nn.Linear(7 * 7 * 64, 10, bias=True)

        # 전결합층 한정으로 가중치 초기화
        torch.nn.init.xavier_uniform_(self.fc.weight)

    def forward(self, x):
        out = self.layer1(x)
        out = self.layer2(out)
        out = out.view(out.size(0), -1)   # 전결합층을 위해서 Flatten
        out = self.fc(out)
        return F.softmax(out, dim=1)
```

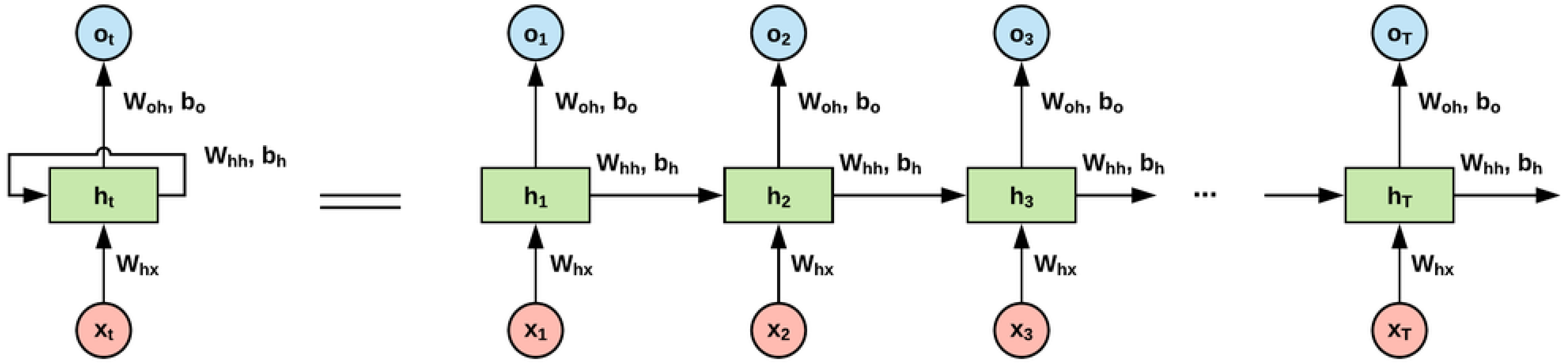
CNN


Low-Level Feature
Mid-Level Feature
High-Level Feature
Trainable Classifier
"car"

RNN/ Transformer(GPT)

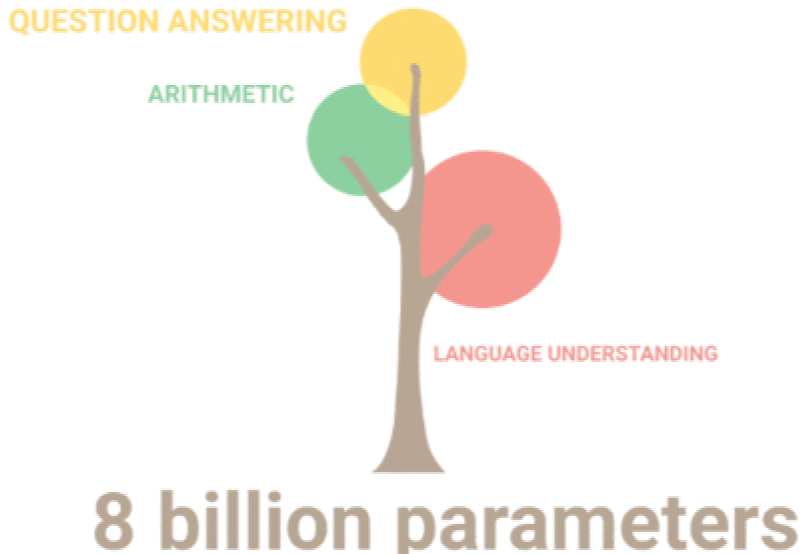
- Sequence data (언어, 음악)





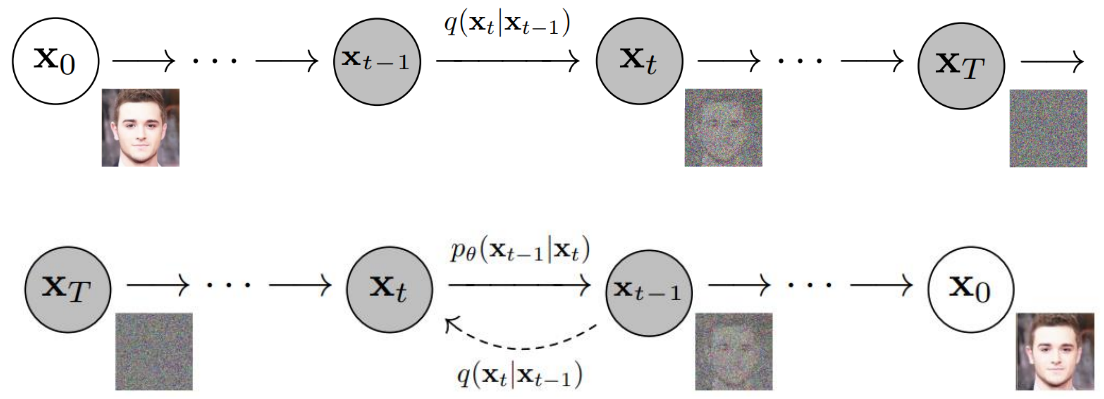
LLM(Large Language Model)

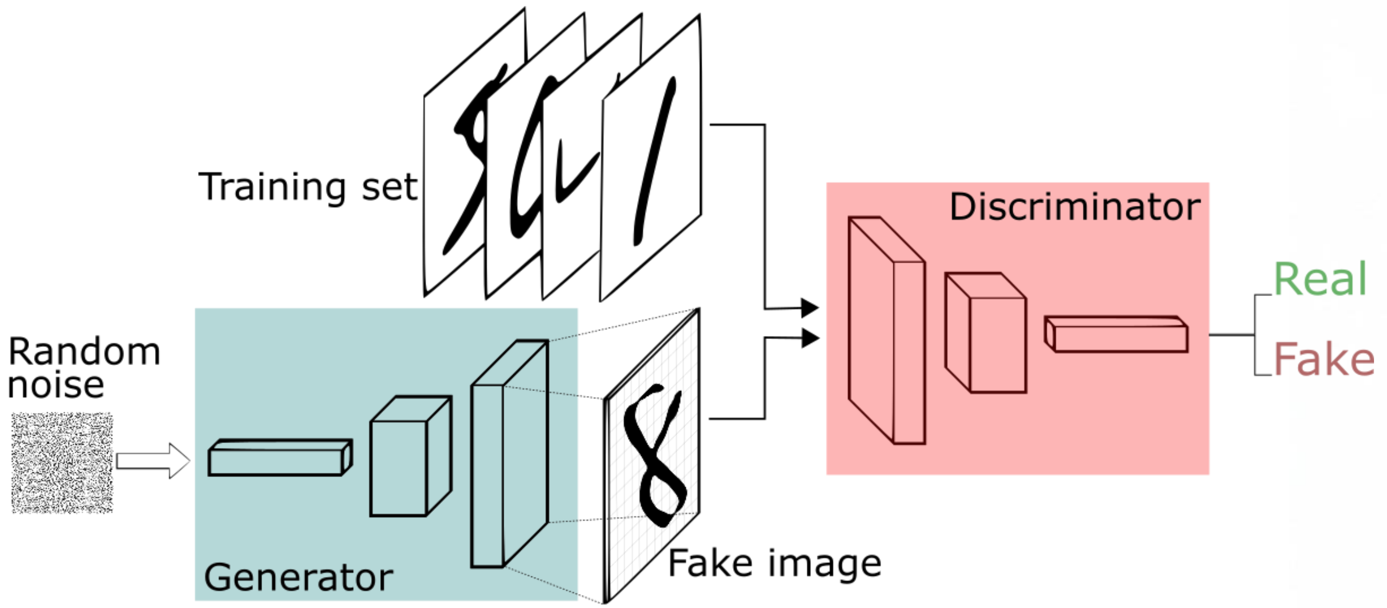
- 추론 능력
- 세계에 대한 이해

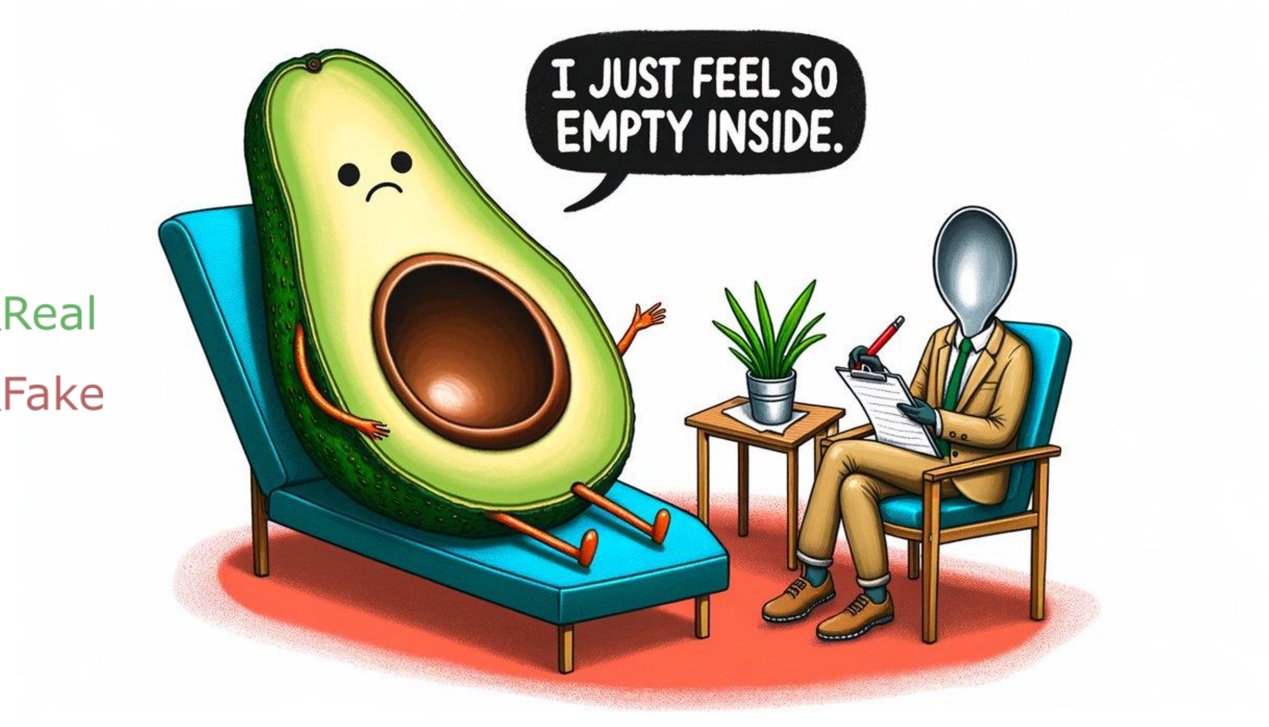


GAN/ Diffusion

- Image generation


$\mathbf{x}_0$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_t$
$\mathbf{x}_T$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$





Hugging Face

인공지능의 기초 세미나

- 딥러닝 학습 과정
- 인공신경망
- Backpropagation
- Dataloader
- Mnist classification using DNN
- CNN
- AI의 흐름 소개

감사합니다

출처

- https://wikidocs.net/63565
- https://commons.wikimedia.org/wiki/Category:Convolutional_neural_networks
- https://www.researchgate.net/figure/Representation-of-the-architecture-of-a-convolutional-neural-network-CNN_fig2_339278442