

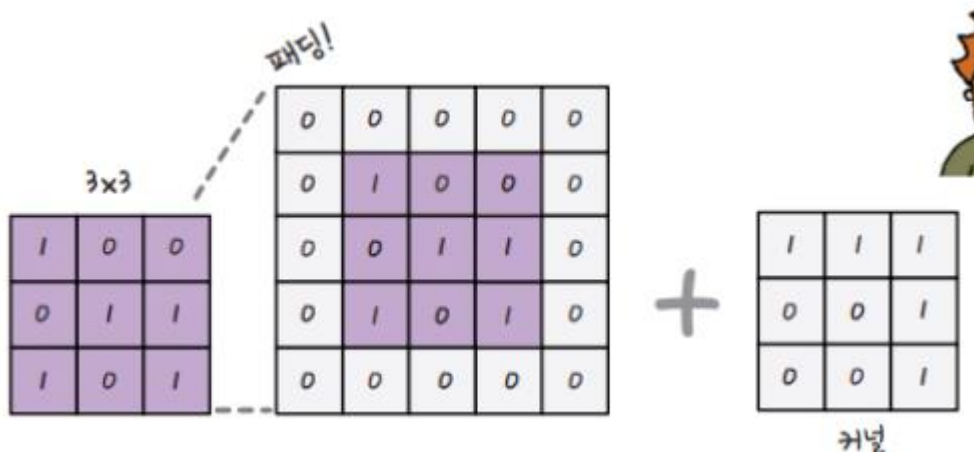
새로운 그림을 생성하는 신경망:

기존의 패턴을 기반으로 그림 생성하는 것:

GAN 만든 사람:

GAN은 ( )와 ( )로 구성됨

연산하세요



커널의 이동 간격:

- 기본값:

```
# 모델 이름을 discriminator로 정하고 Sequential() 함수를 호출합니다.
discriminator = Sequential()
discriminator.add(Conv2D(64, kernel_size=5, strides=2, input_shape=(28,28,1), padding='same')) ..... ①
discriminator.add(Activation('leaky_relu')) ..... ②
discriminator.add(Dropout(0.3)) ..... ③
discriminator.add(Conv2D(128, kernel_size=5, strides=2, padding='same')) ..... ④
discriminator.add(Activation('leaky_relu')) ..... ⑤
discriminator.add(Dropout(0.3)) ..... ⑥
discriminator.add(Flatten()) ..... ⑦
discriminator.add(Dense(1, activation='sigmoid')) ..... ⑧
discriminator.compile(loss='binary_crossentropy', optimizer='adam') ..... ⑨
discriminator.trainable = False ..... ⑩
```

판별자 활성화 함수:

```

generator = Sequential() # 모델 이름을 generator로 정하고 Sequential() 함수를 호출
generator.add(Dense(128*7*7, input_dim=100, ) ) ---- ①
generator.add() ---- ②
generator.add(Reshape((7, 7, 128))) ---- ③
generator.add() ---- ④
generator.add(Conv2D(64, kernel_size=5, ) ) ---- ⑤
generator.add() ---- ⑥
generator.add((0.2))) ---- ⑦
generator.add() ---- ⑧
generator.add(Conv2D(1, kernel_size=5, padding='same', ) )
---- ⑨

```

오토인코더 사용 이유:

# 생성자 모델 만들기

```
autoencoder = Sequential()
```

# 인코딩 부분

```

autoencoder.add((16, kernel_size=3, 'same', input_shape=(28,
28,1), activation='relu')) ---- ①
autoencoder.add((=2, padding='same')) ---- ②
autoencoder.add(Conv2D(8, =3, 'relu', padding='same'))
---- ③
autoencoder.add((pool_size=2, padding='same')) ---- ④
autoencoder.add(Conv2D(8, =3, strides=2, padding='same',
activation='relu')) ---- ⑤

```

# 디코딩 부분

```

autoencoder.add(Conv2D(8, kernel_size=3, padding='same',
activation='relu')) ---- ⑥
autoencoder.add() ---- ⑦
autoencoder.add(Conv2D(8, kernel_size=3, padding='same',
activation='relu')) ---- ⑧
autoencoder.add(UpSampling2D()) ---- ⑨
autoencoder.add(Conv2D(16, kernel_size=3, activation='relu')) ---- ⑩
autoencoder.add(UpSampling2D()) ---- ⑪
autoencoder.add(Conv2D(1, kernel_size=3, padding='same',
activation=) ) ---- ⑫

```

# 전체 구조 확인

```
autoencoder.summary() ---- ⑬
```

미리 학습한 모델을 가져와서 사용하는 방법:

```
transfer_model = (weights=, =False, input_shape=(150,150,3))
transfer_model. = False
transfer_model.summary()
```

Total params: 14,714,688

Trainable params: 0

: 14,714,688

```
finetune_model = models.Sequential()
finetune_model.
finetune_model.add(Flatten())
finetune_model.add(Dense(64))
finetune_model.add(Activation('relu'))
finetune_model.add(Dropout(0.5))
finetune_model.add(Dense(1))
finetune_model.add()
finetune_model.summary()
```

설명 가능한 AI 제안한 학자:

CNN 제작자:

인공지능의 결과를 알 수 없는 모델:

설명가능한 AI 약자:

중간 과정에서 나온 특징 맵 활용: ,

```
explainer = ()
output = explainer. (데이터, 모델, 클래스)
explainer. (output, 저장될 폴더, 저장될 이름)
```

대부분의 인공지능망은 ( ) 모델이다

CNN에서 부위별로 가려가며 설명가능한 모델 만드는 것: ,

두가지 XAI 기법: 오클루전