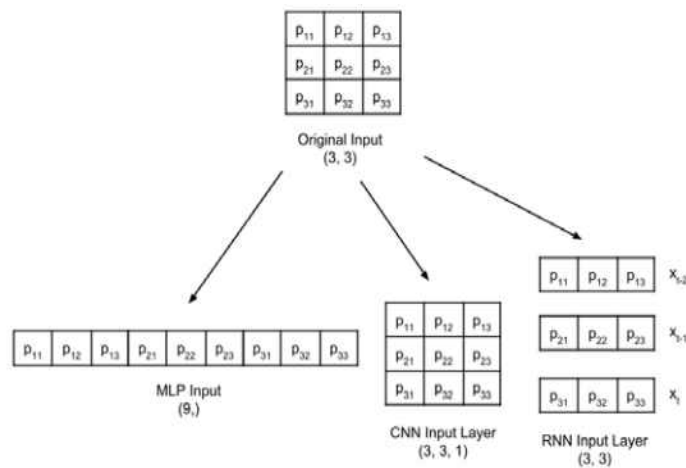


## Deep Learning (MLP, CNN, RNN)

Source: Rowel Atienza, Advanced Deep Learning with TensorFlow 2 and Keras(2nd eds), Packt Publishing Ltd., 2020.

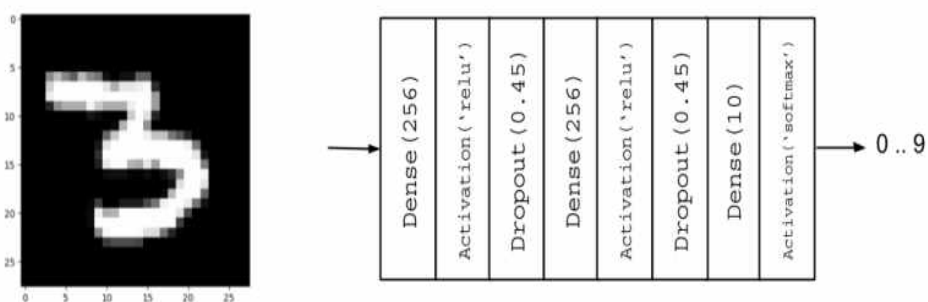
### 0. MNIST 데이터셋

- MNIST (Modified National Institute of Standards and Technology)
- 0에서 9까지의 범위를 갖는 필기 숫자 모음
- 60,000개의 이미지 training data, 10,000개의 이미지 test data
- 각 이미지는  $28 \times 28$  픽셀

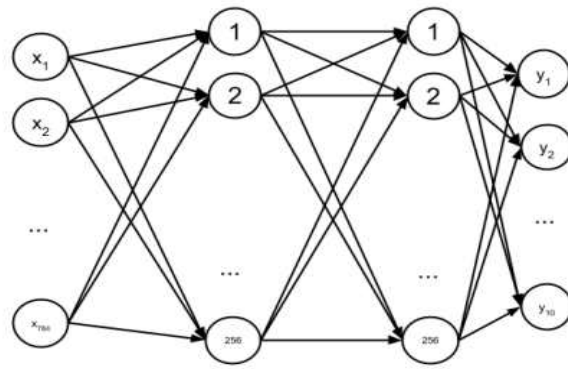


$3 \times 3$  회색조 이미지가 MLP, CNN, RNN 입력 레이어에 대해 어떻게 변형되는지 보여준다.

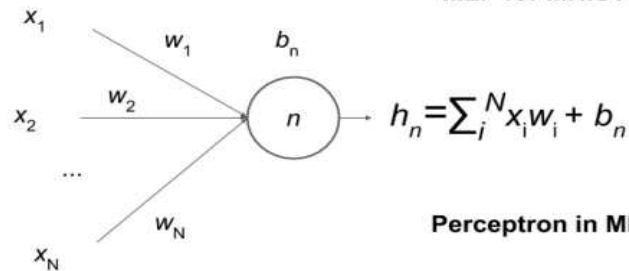
### 1. MLP (MultiLayer Perceptron)



MNIST 숫자 분류를 위해 사용된 MLP 모델



MLP for MNIST

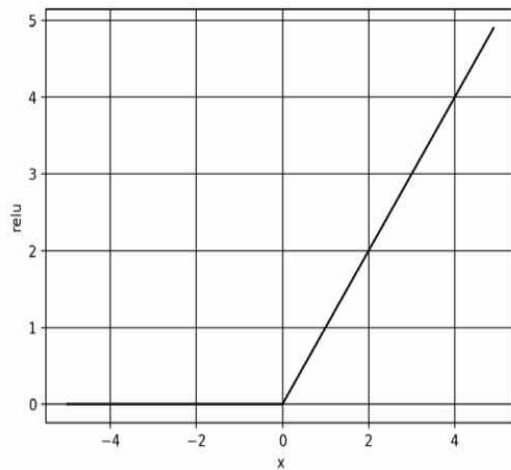


Perceptron in MLP

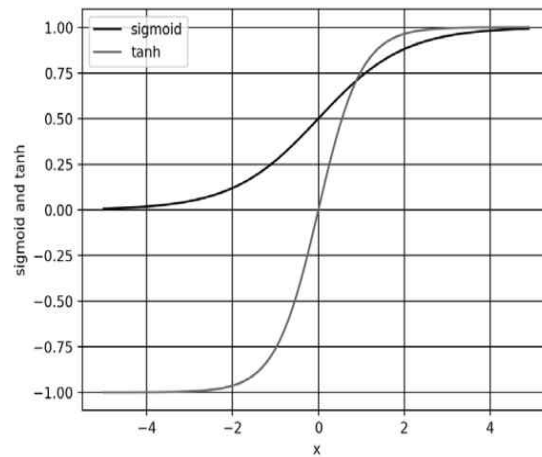
MLP 모델: 완전 연결 네트워크(Fully Connected Network)

relu	$\text{relu}(x) = \max(0, x)$	1.3.1
softplus	$\text{softplus}(x) = \log(1 + e^x)$	1.3.2
elu	$\text{elu}(x, a) = \begin{cases} x & \text{if } x \geq 0 \\ a(e^x - 1) & \text{otherwise} \end{cases}$ <p>where <math>a \geq 0</math> and is a tunable hyperparameter</p>	1.3.3
selu	$\text{selu}(x) = k \times \text{elu}(x, a)$ <p>where <math>k = 1.0507009873554804934193349852946</math> and  <math>a = 1.6732632423543772848170429916717</math></p>	1.3.4
sigmoid	$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$	1.3.5
tanh	$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	1.3.6

일반적인 비선형 활성화 함수들의 정의



Relu 함수



Sigmoid와 Tanh 함수

Loss Function	Equation
mean_squared_error	$\frac{1}{categories} \sum_{i=1}^{categories} (y_i^{label} - y_i^{prediction})^2$
mean_absolute_error	$\frac{1}{categories} \sum_{i=1}^{categories}  y_i^{label} - y_i^{prediction} $
categorical_crossentropy	$- \sum_{i=1}^{categories} y_i^{label} \log y_i^{prediction}$
binary_crossentropy	$-y_1^{label} \log y_1^{prediction} - (1 - y_1^{label}) \log(1 - y_1^{prediction})$

#### 일반적인 손실함수

- 표시된 손실함수는 하나의 출력에만 해당한다.
- 평균 손실 값은 전체 배치의 평균이다.
- 손실함수의 선택
  - 범주 분류의 경우 softmax 활성화 이후에 categorical\_crossentropy 또는 mean\_squared\_error
  - binary\_crossentropy 손실함수는 일반적으로 sigmoid 활성화 레이어 이후에 사용
  - mean\_squared\_error 손실함수는 tanh 출력에 대한 옵션

#### ※ 최적화(optimization)

- 손실함수를 최소화하는 것

: 손실이 허용가능한 수준으로 줄어든다면, 모델은 입력을 출력으로 매핑하는 함수를 간접적으로 학습했음을 뜻한다.

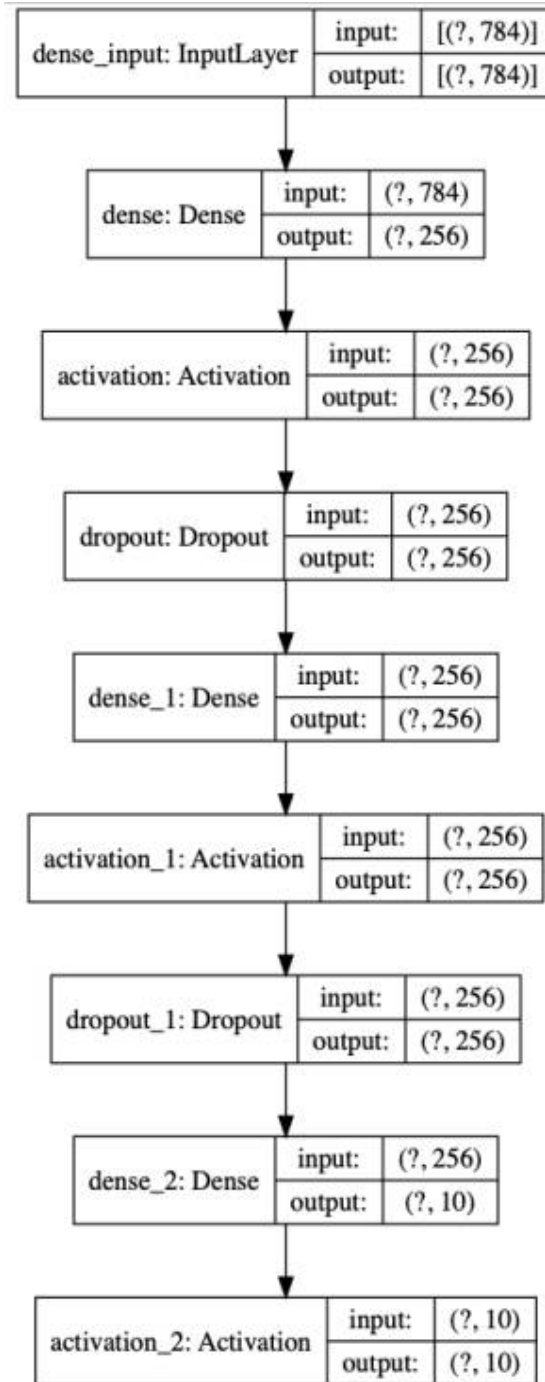
■ 가장 일반적으로 사용되는 최적화 방법

- 확률적 경사 하강법(Stochastic Gradient Descent; SGD)
- 적응적 모멘트(ADaptive Moments; Adam)
- 제곱 평균 제곱근 전파(Root Mean Squared Propagation; RMSprop)

→ 각 최적화 방법은 학습률(learning rate), 모멘텀(momentum), 감쇠(decay)와 같은 조정 가능한 파라미터들을 특징으로 갖는다.

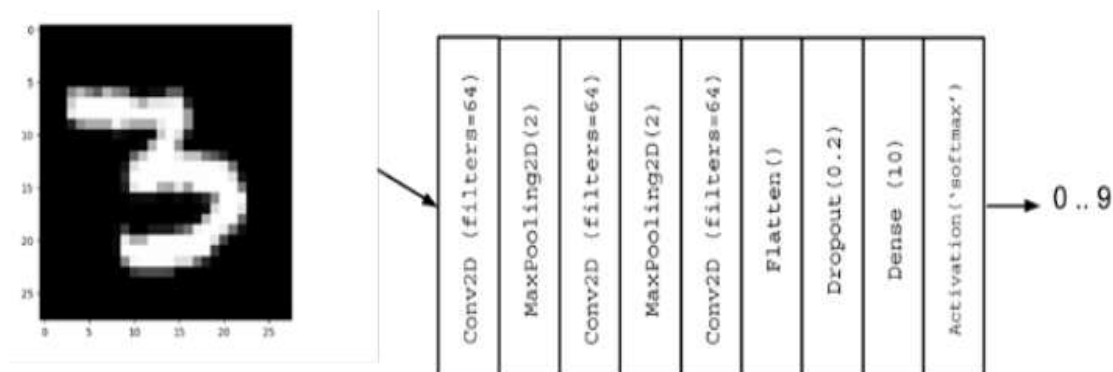
Layers	Regularizer	Optimizer	ReLU	Train Accuracy (%)	Test Accuracy (%)
256-256-256	None	SGD	None	93.65	92.5
256-256-256	L2(0.001)	SGD	Yes	99.35	98.0
256-256-256	L2(0.01)	SGD	Yes	96.90	96.7
256-256-256	None	SGD	Yes	99.93	98.0
256-256-256	Dropout(0.4)	SGD	Yes	98.23	98.1
256-256-256	Dropout(0.45)	SGD	Yes	98.07	98.1
256-256-256	Dropout(0.5)	SGD	Yes	97.68	98.1
256-256-256	Dropout(0.6)	SGD	Yes	97.11	97.9
256-512-256	Dropout(0.45)	SGD	Yes	98.21	98.2
512-512-512	Dropout(0.2)	SGD	Yes	99.45	98.3
512-512-512	Dropout(0.4)	SGD	Yes	98.95	98.3
512-1024-512	Dropout(0.45)	SGD	Yes	98.90	98.2
1024-1024-1024	Dropout(0.4)	SGD	Yes	99.37	98.3
256-256-256	Dropout(0.6)	Adam	Yes	98.64	98.2
256-256-256	Dropout(0.55)	Adam	Yes	99.02	98.3
256-256-256	Dropout(0.45)	Adam	Yes	99.39	98.5
256-256-256	Dropout(0.45)	RMSprop	Yes	98.75	98.1
128-128-128	Dropout(0.45)	Adam	Yes	98.70	97.7

여러 가지 MLP 네트워크 설정 및 성능 측정



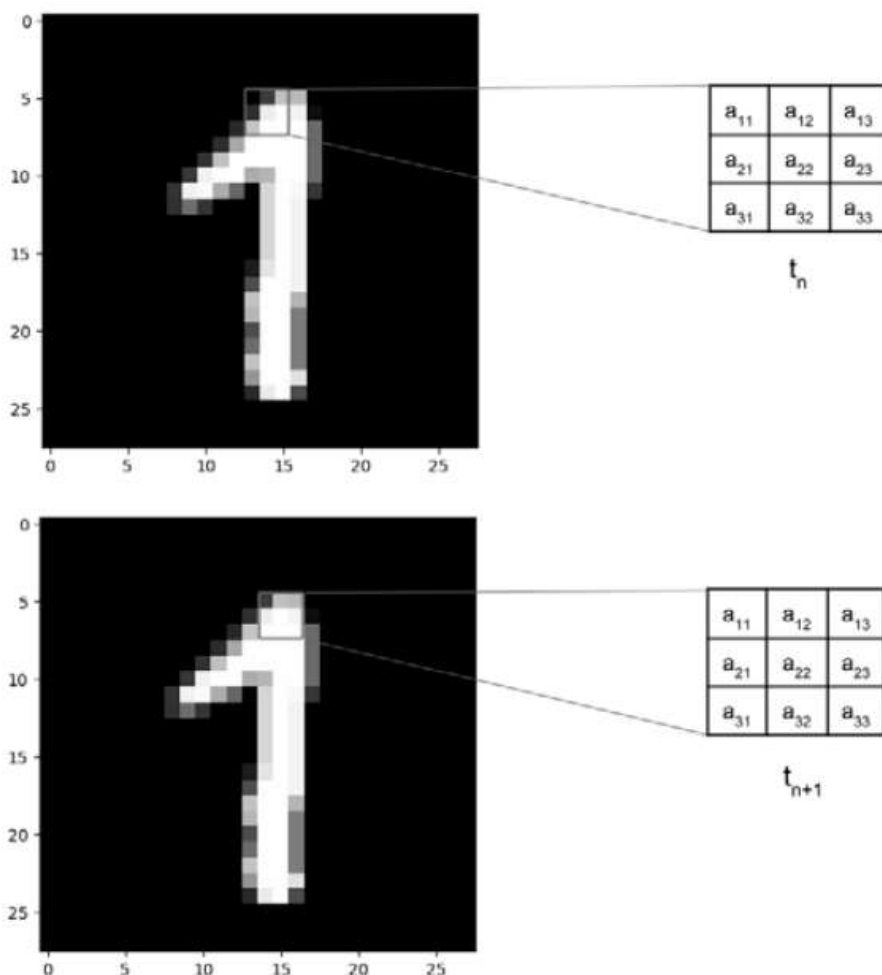
MLP MNIST 숫자 분류기의 그래픽 설명

## 2. CNN (Convolution Neural Network)



MNIST 숫자 분류기를 위한 CNN 모델

- 입력 텐서는 회색조 MNIST 이미지에 대해 (image\_size, image\_size, 1)=(28, 28, 1)을 갖는다.
- training과 test 이미지들의 크기를 조정하려면 입력 모양 요구사항을 준수해야 한다.



3×3 커널은 MNIST 숫자 이미지와 합성곱된다.

합성곱은 스텝  $t_n$  과 커널이 스트라이드(stride)에 의해 오른쪽으로 1 픽셀 이동한  $t_{n+1}$ 로 표시된다.

$p_{11}$	$p_{12}$	$p_{13}$	$p_{14}$	$p_{15}$
$p_{21}$	$p_{22}$	$p_{23}$	$p_{24}$	$p_{25}$
$p_{31}$	$p_{32}$	$p_{33}$	$p_{34}$	$p_{35}$
$p_{41}$	$p_{42}$	$p_{43}$	$p_{44}$	$p_{45}$
$p_{51}$	$p_{52}$	$p_{53}$	$p_{54}$	$p_{55}$

 $\otimes$ 

$a_{11}$	$a_{12}$	$a_{13}$
$a_{21}$	$a_{22}$	$a_{23}$
$a_{31}$	$a_{32}$	$a_{33}$

 $=$ 

$f_{11}$	$f_{12}$	$f_{13}$
$f_{21}$	$f_{22}$	$f_{23}$
$f_{31}$	$f_{32}$	$f_{33}$

$$f_{11} = p_{11}a_{11} + p_{12}a_{12} + p_{13}a_{13} + p_{21}a_{21} + p_{22}a_{22} + p_{23}a_{23} + p_{31}a_{31} + p_{32}a_{32} + p_{33}a_{33}$$

합성곱 연산은 특정 지도(feature map)의 하나의 요소가 계산되는 방법을 보여준다.

$$\text{MaxPooling2D}(2) \left( \begin{array}{|c|c|c|c|} \hline p_{11} & p_{12} & p_{13} & p_{14} \\ \hline p_{21} & p_{22} & p_{23} & p_{24} \\ \hline p_{31} & p_{32} & p_{33} & p_{34} \\ \hline p_{41} & p_{42} & p_{43} & p_{44} \\ \hline \end{array} \right) = \begin{array}{|c|c|} \hline f_{11} & f_{12} \\ \hline f_{21} & f_{22} \\ \hline \end{array}$$

$$f_{11} = \max(p_{11}, p_{12}, p_{21}, p_{22})$$

$$f_{12} = \max(p_{13}, p_{14}, p_{23}, p_{24})$$

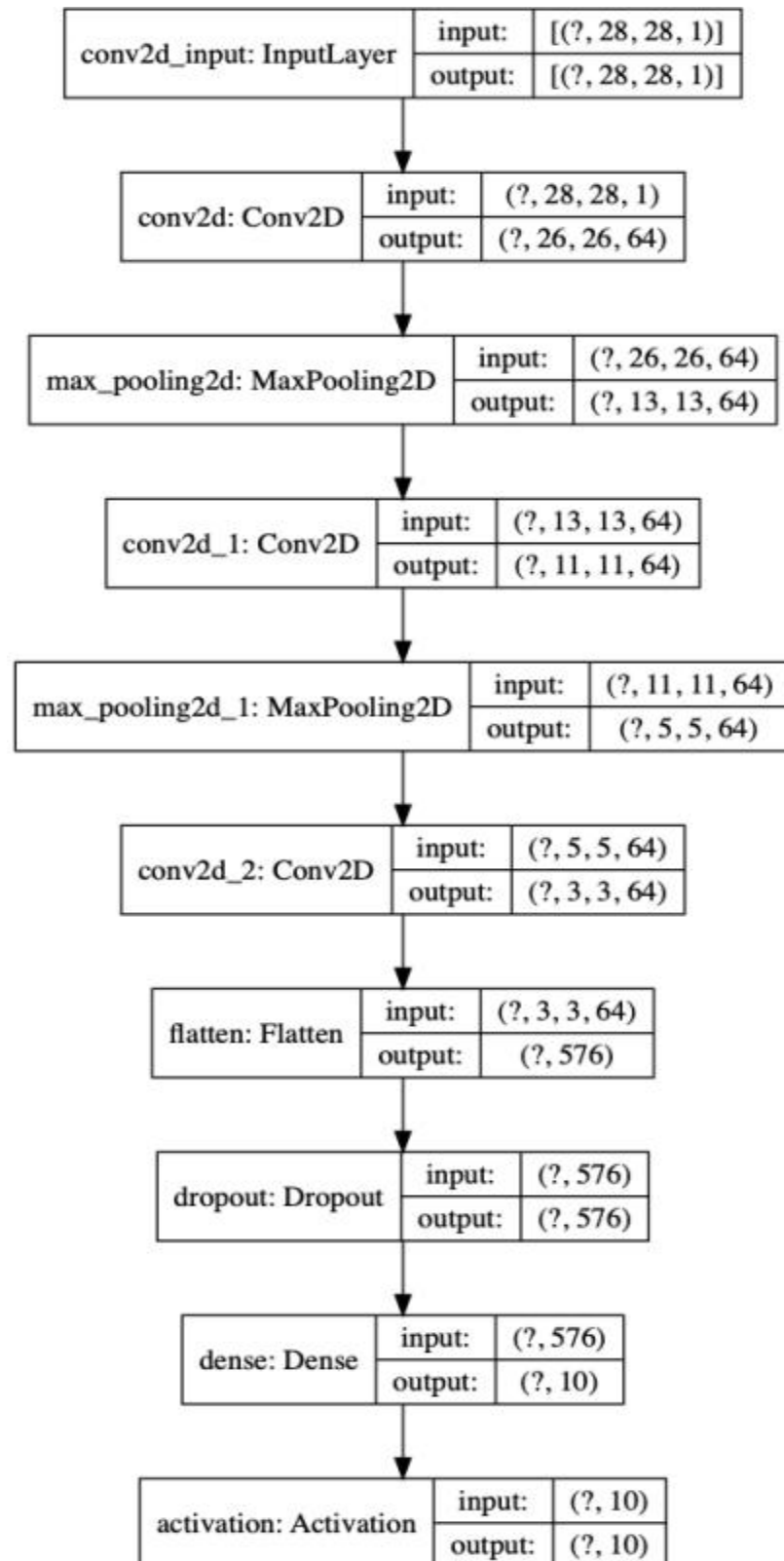
MaxPooling2D 연산

간단히 하기 위해, 입력 특징 지도(feature map)는  $4 \times 4$ 이고 결과는  $2 \times 2$  특징 지도이다.

■ Pooling 연산의 종류: Max Pooling, Average Pooling 등

Layers	Optimizer	Regularizer	Train Accuracy (%)	Test Accuracy (%)
64-64-64	SGD	Dropout(0.2)	97.76	98.50
64-64-64	RMSprop	Dropout(0.2)	99.11	99.00
64-64-64	Adam	Dropout(0.2)	99.75	99.40
64-64-64	Adam	Dropout(0.4)	99.64	99.30

여러 가지 CNN 네트워크 설정 및 CNN MNIST 숫자 분류기의 성능 측정

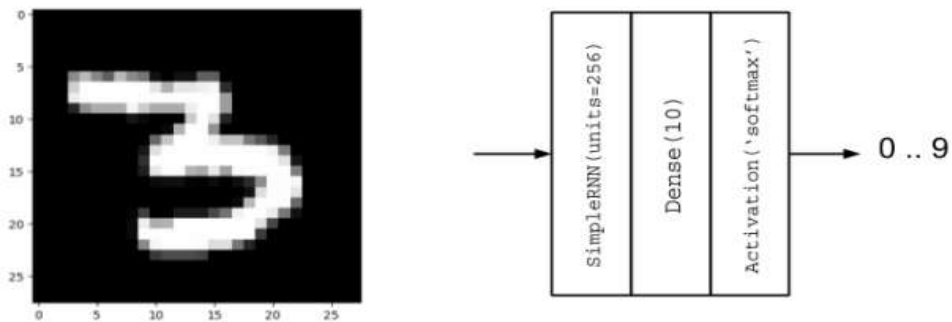


CNN MNIST 숫자 분류기의 도식화 설명

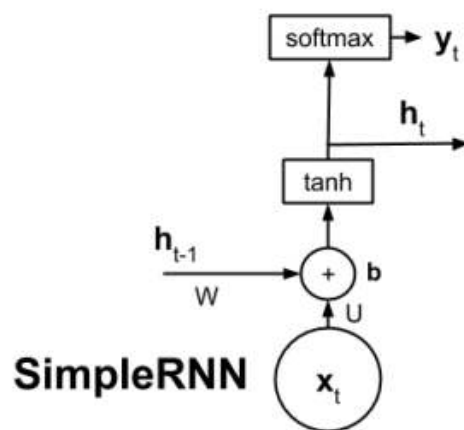


### 3. RNN (Recurrent Neural Network, 순환 신경망)

■ MNIST 데이터 샘플이 기본적으로 순차적이진 않지만, 모든 이미지를 픽셀의 행 또는 열의 순서로 볼 수 있다고 생각하는 것은 어렵지 않다. 게다가, RNN 기반 모델은 각 MNIST 이미지를 28개의 타임 스텝(timestep)을 갖는 28개의 요소로 된 입력 벡터들의 순서로 처리할 수 있다.



MNIST 숫자 분류기를 위한 RNN 모델



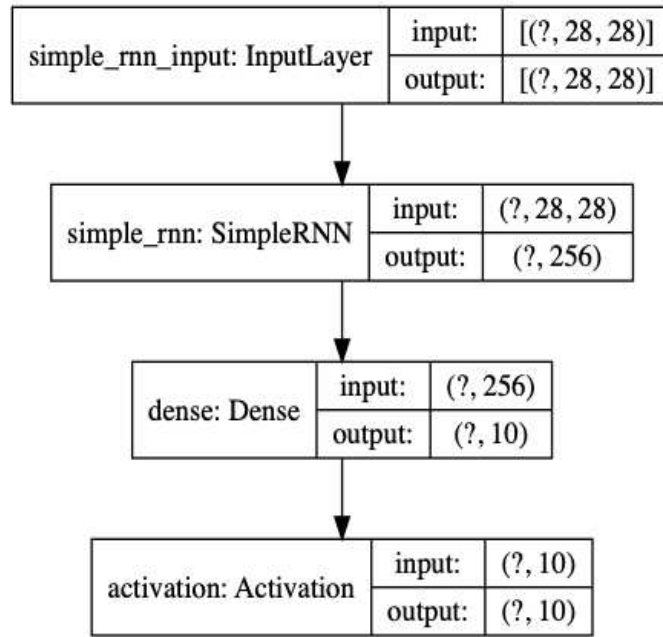
SimpleRNN의 출력

$$h_t = \tanh(b + Wh_{t-1} + Ux_t)$$

여기서,  $b$ : bias,  $W$ : 반복커널(이전 출력의 가중치),  $U$ : 커널(현재 입력의 가중치)

Layers	Optimizer	Regularizer	Train Accuracy (%)	Test Accuracy (%)
256	SGD	Dropout(0.2)	97.26	98.00
256	RMSprop	Dropout(0.2)	96.72	97.60
256	Adam	Dropout(0.2)	96.79	97.40
512	SGD	Dropout(0.2)	97.88	98.30

여러 가지 SimpleRNN 네트워크 설정 및 성능 측정



SimpleRNN MNIST 숫자 분류기의 도식화 설명