

# 인공지능 기초

**인공지능\_ Day01**

**김새봄**

인공지능	01
머신러닝과 딥러닝	02
딥러닝	03
퍼셉트론	04
다층 퍼셉트론	05
실습	06

  
TensorFlow

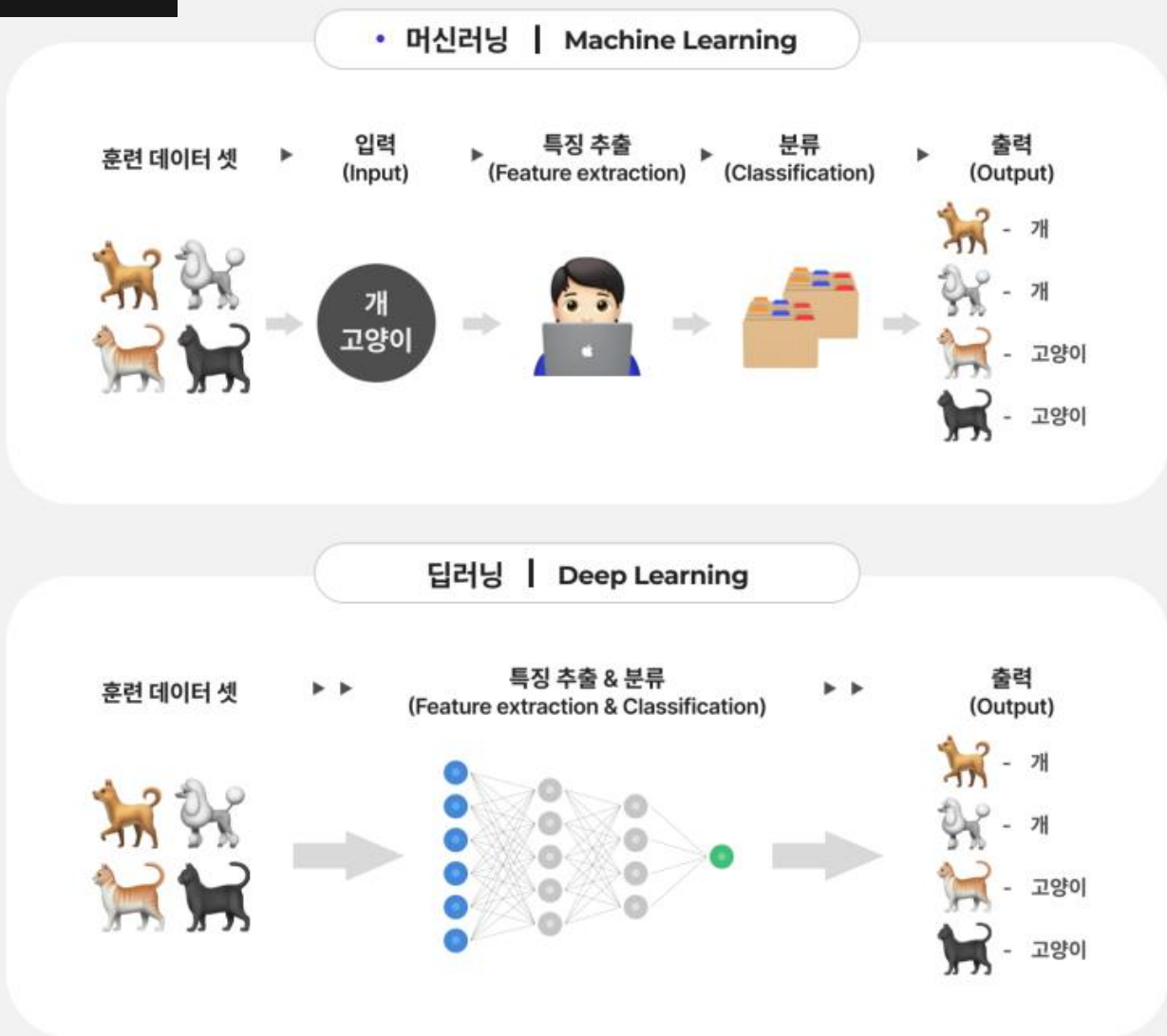
PYTORCH



<https://www.bnviit.com/>



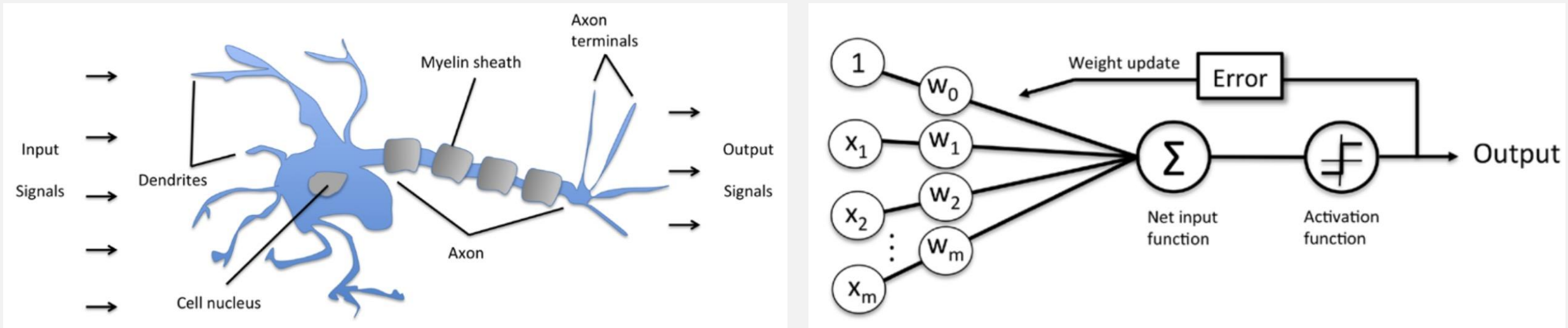
Figure 1. 인공지능과 머신러닝, 딥러닝



<https://www.codestates.com/>

Figure 2. 머신러닝과 딥러닝

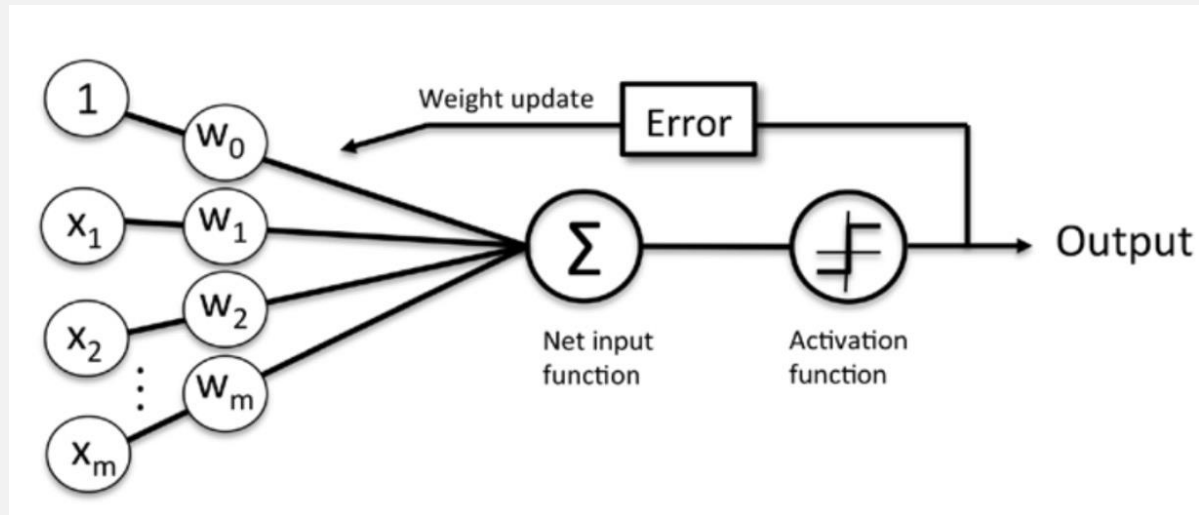
- 일련의 단계들 또는 레이어들의 연산을 사용하는 머신러닝 알고리즘
- 레이어를 세로로 쌓았을 때 모양 때문에 깊이(deep)라고 부른다



<http://www.nextobe.com/2020/04/15/%ED%8D%BC%EC%85%89%ED%8A%B8%EB%A1%A0/>

**Figure 3. 인간의 신경망과 인공신경망**

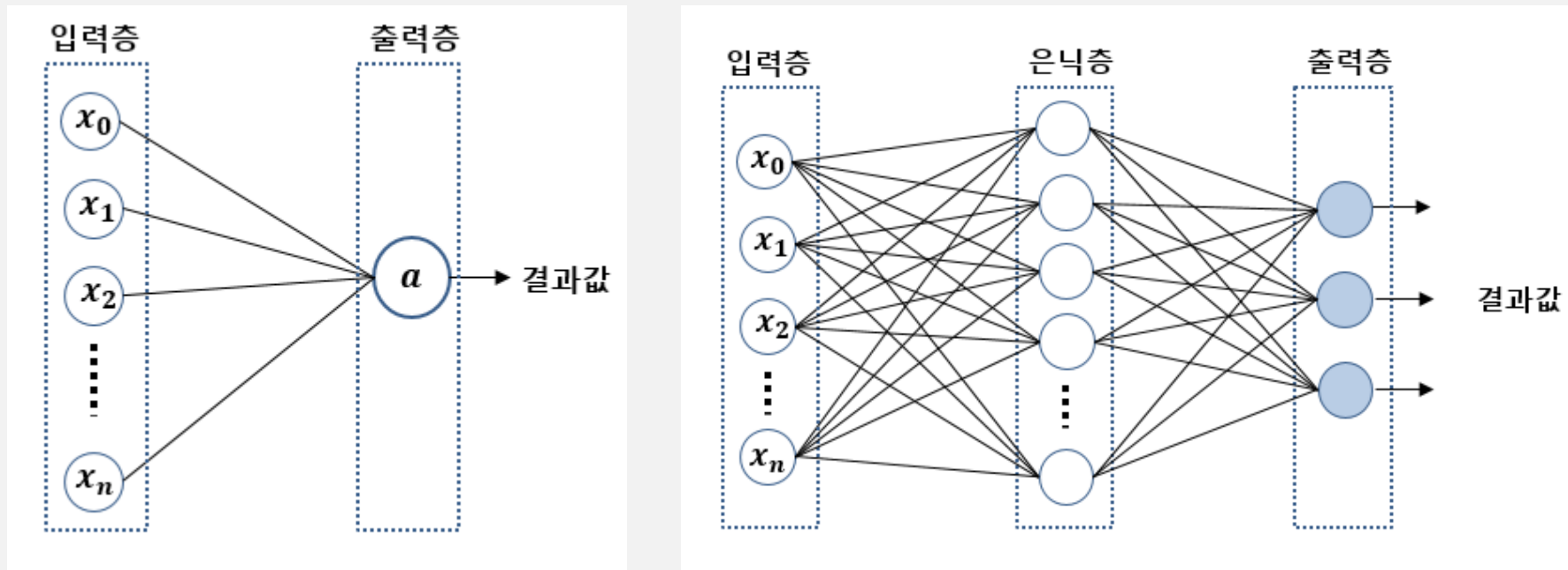
- 인공 신경망의 한 종류
- 뉴런의 모델을 모방하여 입력층, 출력층으로 구성한 모델
- 구성요소 : 입력값, 가중치, 활성화함수, 출력값(예측값)



<http://www.nextobe.com/2020/04/15/%ED%8D%BC%EC%85%89%ED%8A%B8%EB%A1%A0/>

Figure 4. 퍼셉트론

- 딥러닝은 다층 퍼셉트론의 구조에서 나온 것
- 명칭 : 노드(node, 뉴런)는 에지(edge)라고 하는 선으로 연결됨
- 입력층(input layer), 은닉층(hidden layer), 출력층(output layer)으로 구성



<http://www.nextobe.com/2020/04/15/%ED%8D%BC%EC%85%89%ED%8A%B8%EB%A1%A0/>

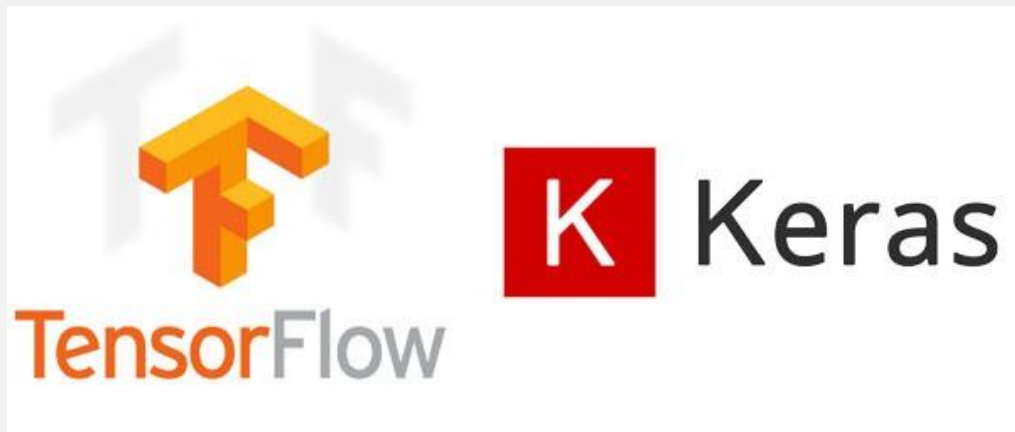
Figure 5. 퍼셉트론과 다층 퍼셉트론

실습





- 데이터 과학 분야를 위한 표준 프로그래밍 언어
- 데이터 적재, 시각화, 통계, 자연어 처리, 이미지 처리 등에 필요한 라이브러리들을 가지고 있음
- 반복 작업을 빠르게 처리하고 손쉽게 조작가능



- Keras와 TensorFlow는 모두 딥러닝을 위한 파이썬 라이브러리
- Keras는 TensorFlow를 백엔드(backend)로 사용
- TensorFlow는 구글에서 개발한 딥러닝 엔진

# scikit-learn

- 오픈 소스인 scikit-learn은 파이썬 머신러닝 라이브러리
- 사용자 가이드 [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html)
- API 문서 <https://scikit-learn.org/stable/modules/classes.html>
- 설치

```
pip install scikit-learn
```

```
conda install scikit-learn
```



- Numerical Python의 약어로, 파이썬에서 과학 계산을 위한 핵심 라이브러리
- 다차원 배열과 행렬 연산을 위한 라이브러리
- 수치 계산을 위한 다양한 함수와 메소드를 제공
- 다차원 배열(ndarray)의 사용

파이썬의 리스트와 유사하지만, ndarray는 벡터, 행렬 등 다차원 데이터의 처리 및

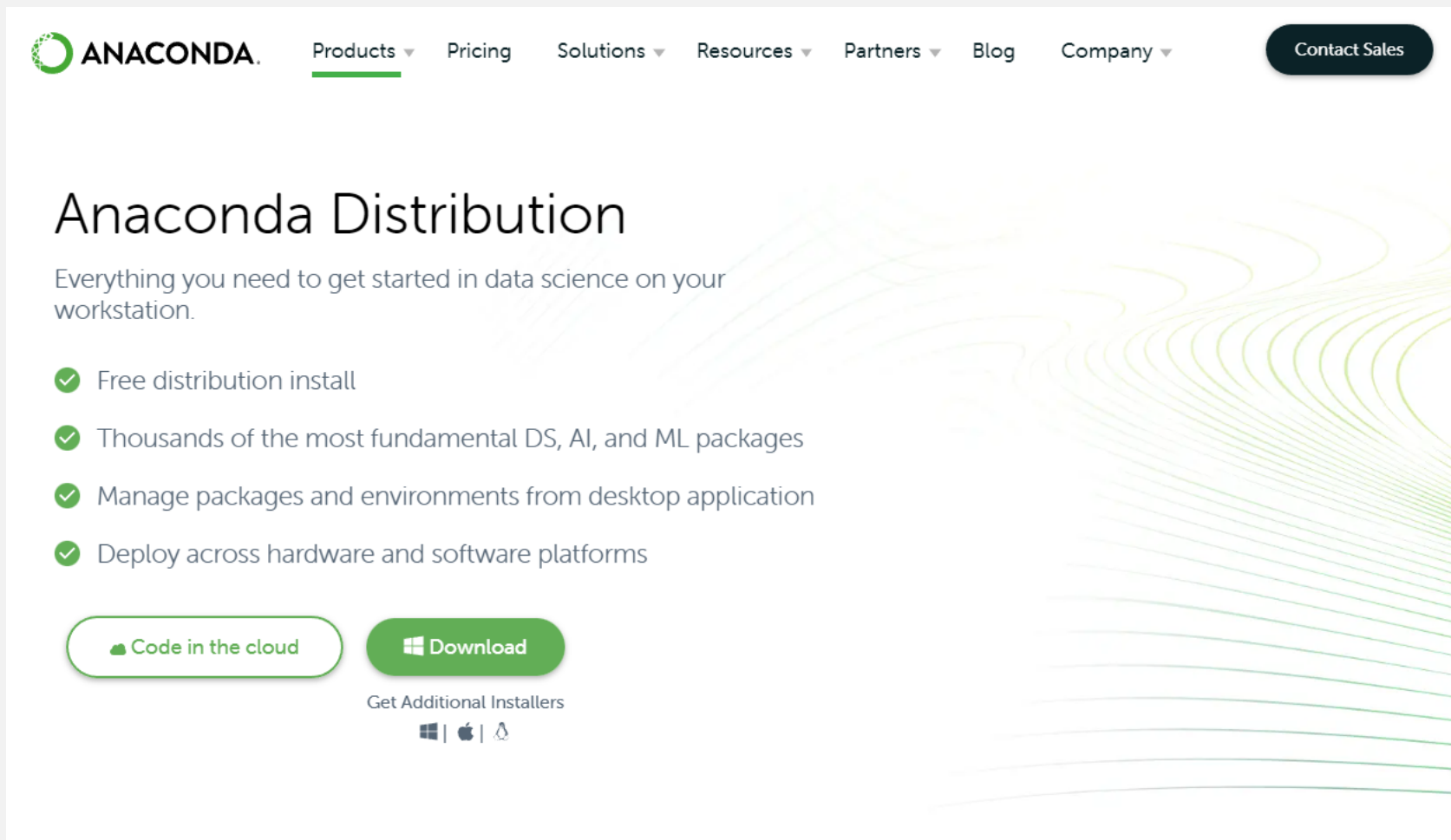
다양한 함수를 사용하여 데이터를 빠르고 효율적으로 처리가능

- 데이터 처리와 분석을 위한 파이썬 라이브러리
- DataFrame이라는 테이블 형태의 데이터 구조를 기반으로 만들어 짐
- 엑셀파일, csv 파일 등 다양한 파일과 데이터베이스의 데이터를 읽어들이 수 있음

```
import pandas as pd
```

```
df = pd.DataFrame({'name': ['Alice', 'Bob', 'Charlie'], 'age': [25, 30, 35], 'gender': ['F', 'M', 'M']})  
print(df)
```

	name	age	gender
0	Alice	25	F
1	Bob	30	M
2	Charlie	35	M

The image is a screenshot of the Anaconda Distribution website. At the top, there is a navigation bar with the Anaconda logo on the left, followed by links for Products, Pricing, Solutions, Resources, Partners, Blog, and Company. A 'Contact Sales' button is located on the right side of the navigation bar. Below the navigation bar, the main heading 'Anaconda Distribution' is displayed. Underneath the heading, a subtext reads 'Everything you need to get started in data science on your workstation.' A list of four features is presented, each preceded by a green checkmark: 'Free distribution install', 'Thousands of the most fundamental DS, AI, and ML packages', 'Manage packages and environments from desktop application', and 'Deploy across hardware and software platforms'. Below the list, there are two buttons: 'Code in the cloud' and 'Download'. Under the 'Download' button, the text 'Get Additional Installers' is shown, followed by icons for Windows, macOS, and Linux. The background of the page features a decorative pattern of green and yellow wavy lines.

가상환경 구성 후 독립적인  
패키지 버전 관리가 가능

```
#1. 데이터
import numpy as np
x = np.array([1, 2, 3])
y = np.array([1, 2, 3])

#2. 모델구성
# from tensorflow.python.keras.models import Sequential
# from tensorflow.python.keras.layers import Dense
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()
model.add(Dense(4, input_dim=1))
model.add(Dense(10))
model.add(Dense(5))
model.add(Dense(3))
model.add(Dense(2))
model.add(Dense(1))
```

```
#3. 컴파일, 훈련
model.compile(loss='mse', optimizer='adam')
model.fit(x, y, epochs=1000)

#4. 평가, 예측
loss = model.evaluate(x, y)
print('loss : ', loss)

result = model.predict([4])
print('4의 예측값 : ', result)
```

딥러닝 기본 코드 실습

- 실제값과 예측값의 차이(오차)를 비교하는 지표
- 값이 낮을수록 학습이 잘 된 것이며, 정답과 알고리즘 출력을 비교할 때 사용
- 최적의 매개변수(가중치와 편향)을 탐색할 때  
손실 함수를 가능한 작게 하는 값을 찾음
- MSE(Mean Squared Error) : 평균 제곱 오류  
출력결과와 데이터 차이 제곱의 평균으로 정답과 오답의 모든 확률 고려
- CEE(Cross Entropy Error) : 교차 엔트로피 오차  
실제 정답의 확률만을 고려한 손실 함수(훈련데이터의 정답 혹은 확률이 0 또는 1)



- 배치 사이즈(batch size)란, 한 번의 업데이트에 사용되는 샘플의 개수  
배치 사이즈가 32인 경우, 32개의 샘플을 모아서 한 번의 업데이트를 수행함
- 작은 배치 사이즈는 모델이 더 자주 업데이트되므로  
노이즈가 많은 경향이 있으며, 더 많은 학습 시간이 필요
- 큰 배치 사이즈는 학습 데이터의 분포와 유사해지기 때문에  
학습 시간이 빠르고 노이즈가 적어질 수 있으나 과적합이 일어날 수 있음
- 딥러닝에서 배치 사이즈는 학습시에 지정되며  
일반적으로 32, 64, 128 등의 2의 거듭제곱 수를 사용함
- 배치 사이즈는 모델의 크기와 데이터셋의 크기에 따라 적절하게 조정되어야 함

# Day01. 인공지능 Study

1. 인공지능 개념 정리 - 머신러닝, 딥러닝
2. 퍼셉트론 (Perceptron)
3. 다층 퍼셉트론 (Multi-Layer Perceptron: MLP)
4. 옵티마이저 (Optimizer)
5. 학습률 (learning rate)
6. 경사하강법 (Gradient Descent)
7. 손실함수 (Loss Function)
8. 활성화 함수 (Activation Function) - Sigmoid, ReLU, Softmax

수고하셨습니다.