강의 목표

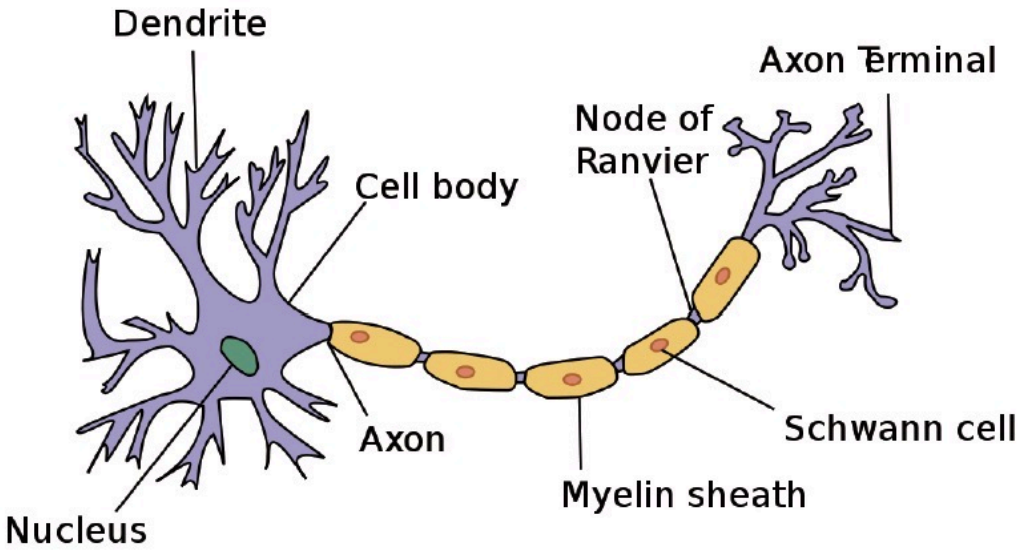
1. 인공신경망(ANN)의 구조와 개념을 이해합니다.
2. TensorFlow 2.0을 이용해서 ANN 구조를 구현해봅니다.

인공신경망(ANN)의 등장 배경

- 이제 실제 딥러닝 알고리즘에서 사용하는 **인공신경망**Artificial Neural Networks(ANN) 기법에 대해 알아봅시다.
- 인공신경망에 대한 연구는 우리가 일반적으로 예상하는 것보다 훨씬 긴 역사를 가지고 있습니다. 인공신경망 아이디어의 시초는 컴퓨터가 발명되던 시기인 1940년대까지 거슬러 올라갑니다. 인공신경망에 대한 개념은 McCulloch, Warren S, and Walter Pitts가 1943년에 발표한 “A logical calculus of the ideas immanent in nervous activity”라는 제목의 논문에서 최초로 제안되었습니다.
- 인공신경망은 이름에서 알 수 있듯이 **생물학적 신경망**Biological Neural Networks에서 아이디어를 얻었습니다.
- 인간의 뇌에 대한 연구가 발전하면서 인간의 뇌는 여러 개의 신경세포(뉴런Neuron)가 서로 연결되어 있고, 이들이 병렬적으로 연산을 진행하면서 정보를 처리한다는 사실이 발견되었습니다. 좀 더 정확히 사람의 뇌 속에는 약 10^{11}개의 뉴런이 있고, 1개의 뉴런은 약 10^{3}개의 다른 뉴런들과 연결되어 있다고 알려져 있습니다.
- 즉, 인간의 뇌 속에는 약 10^{14}개의 뉴런과 뉴런 사이의 연결이 존재하고, 각각의 뉴런들은 서로 전기 신호를 주고 받으면서 연산을 수행합니다.

인공신경망(ANN)의 등장 배경

- 아래 그림은 생물학적 뉴런의 구조를 보여줍니다. 수상돌기$_{\text{Dendrite}}$는 다른 뉴런으로 주어지는 전기 신호를 입력 받습니다. 축삭돌기$_{\text{Axon}}$는 수상돌기로부터 주어진 전기 신호를 축삭돌기 말단$_{\text{Axon Terminal}}$에서 다른 뉴런들(즉, 다른 뉴런의 수상돌기)로 전달합니다.
- 이때, 수상돌기로부터 입력받은 전기 신호는 역치를 넘지 못하지면 중간에 사라지기도 합니다. 인간의 뇌 속에 이런 수많은 뉴런이 전기 신호를 주고받는 연산을 수행함으로써 인간은 눈으로 물체를 인식하거나 귀로 어떤 사람의 목소리를 분간해내거나, 몸을 움직이는 등 다양한 고차원 행동을 수행할 수 있습니다. 따라서 **인간의 뇌는 엄청난 양의 병렬 처리 연산기**라고 할 수 있습니다.

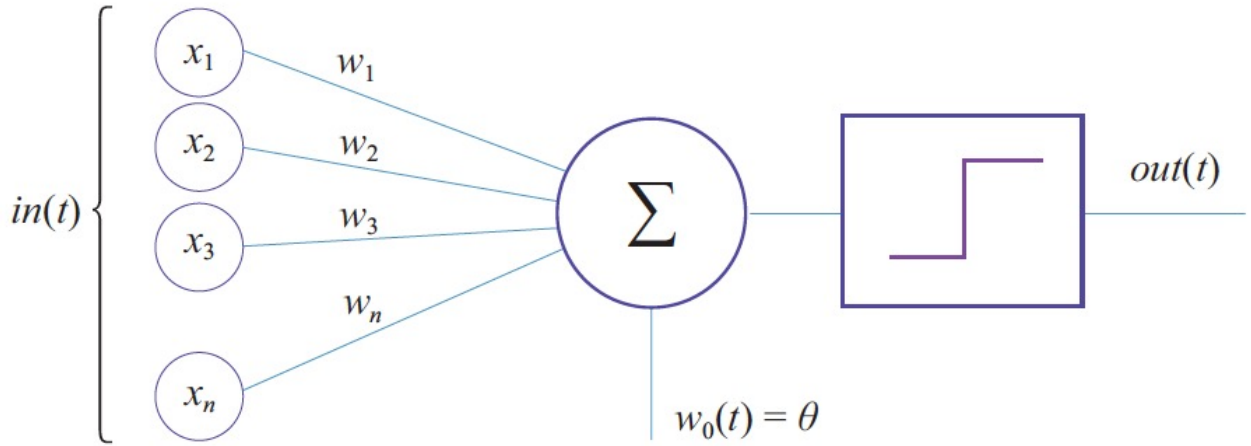


인공신경망(ANN)의 등장 배경

- 이에 반해 컴퓨터는 메모리에서 값을 불러와서 CPU에서 순차적으로 연산을 처리합니다. 즉, 컴퓨터는 **순차 처리 연산기**라고 할 수 있습니다.
- 컴퓨터는 인간의 뇌에 비해 하나의 연산을 처리하는 속도가 훨씬 빠르다는 장점이 있습니다. 따라서 단순한 덧셈 계산 등 기초 연산은 컴퓨터가 인간보다 월등히 뛰어난 능력으로 수행할 수 있습니다. 하지만 컴퓨터는 인간이 간단히 수행해내는 물체 인식이나 음성 인식 등 기초적인 인지활동을 잘해내지 못하는 문제점이 있습니다.
- 이에 기반해서 초기 인공신경망 연구자들은 “**컴퓨터도 인간의 뇌처럼 대량의 병렬 처리 연산을 수행하도록 만들면 컴퓨터도 인간이 쉽게 할 수 있는 인지행동을 할 수 있지 않을까?**”라는 아이디어로부터 초기 인공신경망 구조를 디자인하였습니다.

퍼셉트론Perceptron

- 1943년에 McCulloch, Warren과 Walter Pitts가 인공신경망의 개념을 제안하였지만 이는 개념적인 시도로써 구체적인 공학적 구현을 최초로 제안한 것은 1958년에 Frank Rosenblatt이 발표한 “The perceptron: A probabilistic model for information storage and organization in the brain.” 논문입니다.
- **퍼셉트론Perceptron**은 생물학적 뉴런을 공학적인 구조로 변형한 그림입니다. 아래 그림은 퍼셉트론의 구조를 보여줍니다.
- 퍼셉트론은 입력층Input Layer in(t)과 출력층Output Layer out(t)을 가지고 있습니다.
- 퍼셉트론은 입력층에서 인풋 데이터 x를 받고, 이를 가중치 W와 곱한 후, 이 값에 바이어스Bias b를 더합니다. 이 값을 활성 함수 σ의 입력값으로 대입해서 출력층은 최종적으로 0 또는 1의 값을 출력합니다.

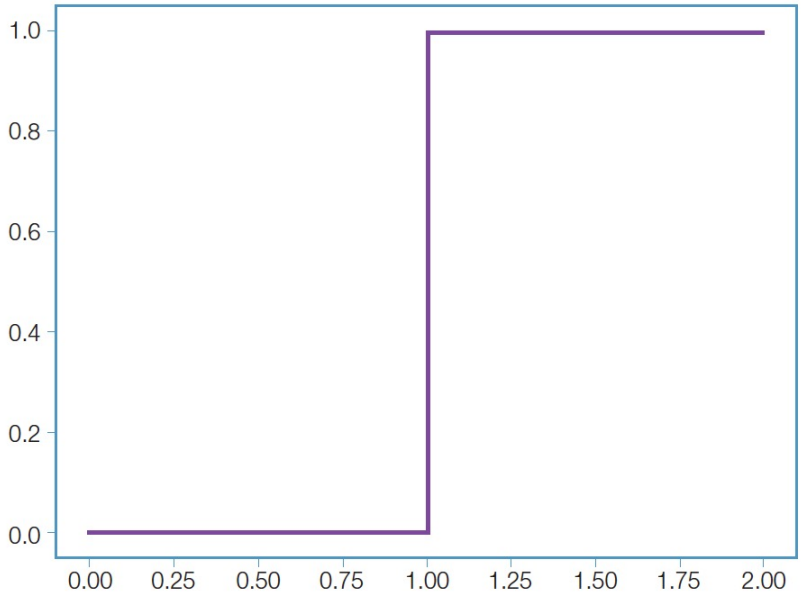


퍼셉트론Perceptron의 활성함수

- 퍼셉트론의 동작 과정을 수학적으로 표현하면 다음과 같습니다.

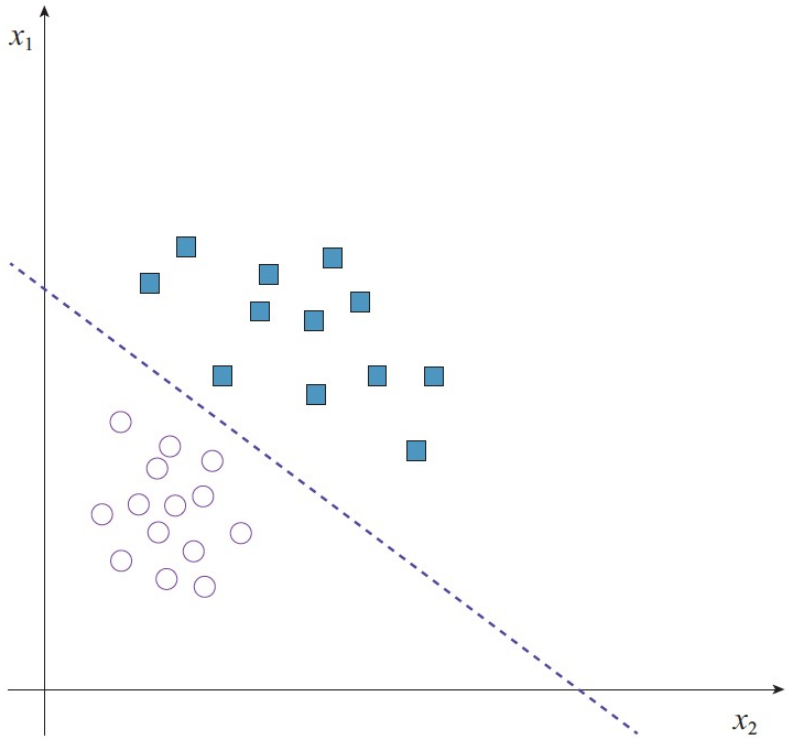
$$y = \sigma(Wx + b)$$

- 여기서 σ는 활성 함수를 나타냅니다. 퍼셉트론을 활성 함수로 아래 그림과 같은 **계단 함수Step Function**를 사용해서 값이 0보다 크면 1, 0보다 작으면 0을 출력합니다.



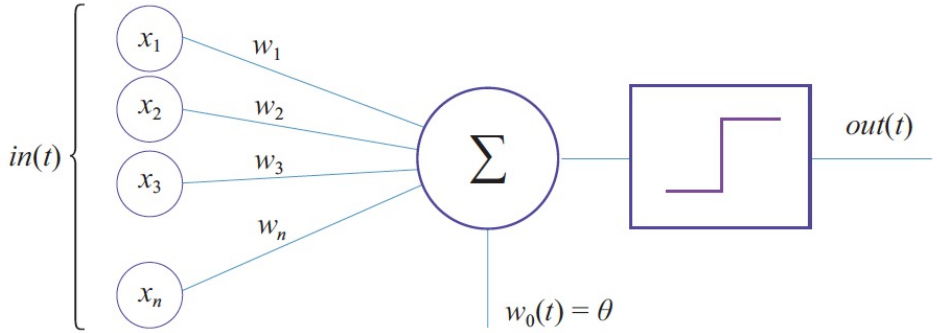
퍼셉트론Perceptron

- 즉 퍼셉트론은 입력값을 받으면 2개의 출력값 중 하나를 출력해내는 **선형 이진분류기Linear Binary Classifier**입니다.



Perceptron의 동작과정의 직관적 이해

- Perceptron의 가중치 : Input의 중요도를 나타낸다.
- 예를 들어, ”주말에 집에서 나가 데이트를 할것인가?”에 대한 의사결정 모델을 Perceptron을 이용해서 만든다고 생각해보자.
- 이때 의사결정 고려사항은 다음과 같이 세가지라고 가정하자
- 1. 날씨가 좋은가? (W1)
- 2. 남자친구 혹은 여자친구가 바쁜 일이 없는가? (W2)
- 3. 데이트 장소가 집에서 가까운가? (W3)



- 이렇게 세가지 고려사항을 Perceptron의 Input으로 넣을 수 있다. 이때 날씨를 가장 중요하게 고려하는 사람이라면 W1=6,W2=2,W3=2의 가중치를 줄 수 있다.
- 만약 데이트 장소가 집에서 가까운 것을 가장 중요하게 고려하는 사람이라면 W1=2,W2=2,W3=6의 가중치를 줄 수 있다.

퍼셉트론Perceptron 의 한계

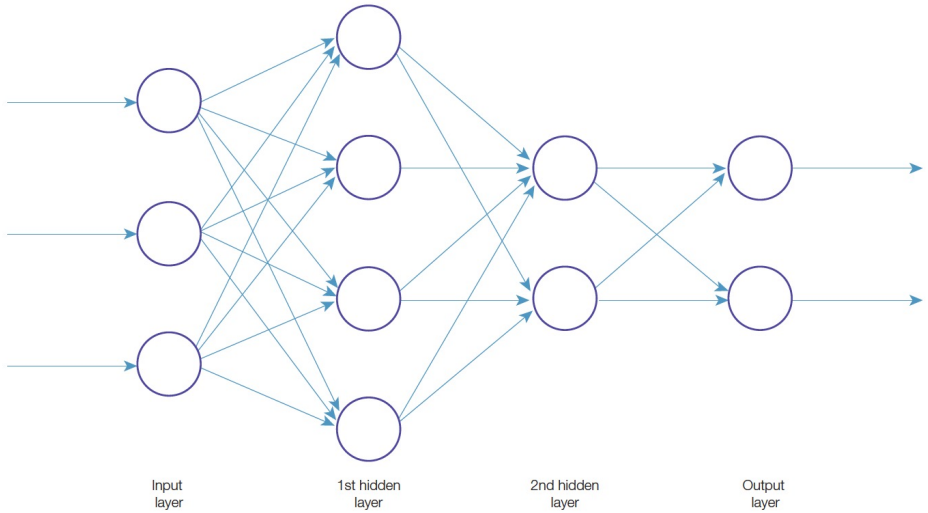
- 퍼셉트론은 지금의 딥러닝처럼 매스컴으로부터 당장이라도 실제 인간과 같은 인공지능을 만들 수 있을 것이라는 과도한 기대를 받았습니다.
- 하지만 1969년 Marvin Minsky와 Seymour Papert가 집필한 “Perceptrons: an introduction to computational geometry”라는 책에서 **퍼셉트론은 단순한 선형분류기에 불과하며, 따라서 간단한 XOR 문제(선형 분리가 불가능한 문제)도 해결할 수 없다는 사실**을 수학적으로 증명하면서 인기가 급속히 사그러들었습니다.
- 하지만, 퍼셉트론은 최초로 인공신경망 개념을 공학적인 구조로 구현했다는 점에서 큰 의미가 있는 모델입니다.

다층 퍼셉트론 MLP$_{\text{Multi-Layer Perceptron}}$ = 인공신경망(ANN)

- 퍼셉트론의 인기가 사그라들면서 인공지능 연구 트렌드는 인공신경망 대신에 논리적인 추론을 이용하는 방법으로 돌아섰습니다. 하지만 몇몇 연구자들은 인공신경망의 가능성을 믿고 연구를 지속하였고 퍼셉트론을 여러층 쌓아 올린 **다층 퍼셉트론**$_{\text{Multi-Layer Perceptron(MLP)}}$ 구조를 제안하였습니다.

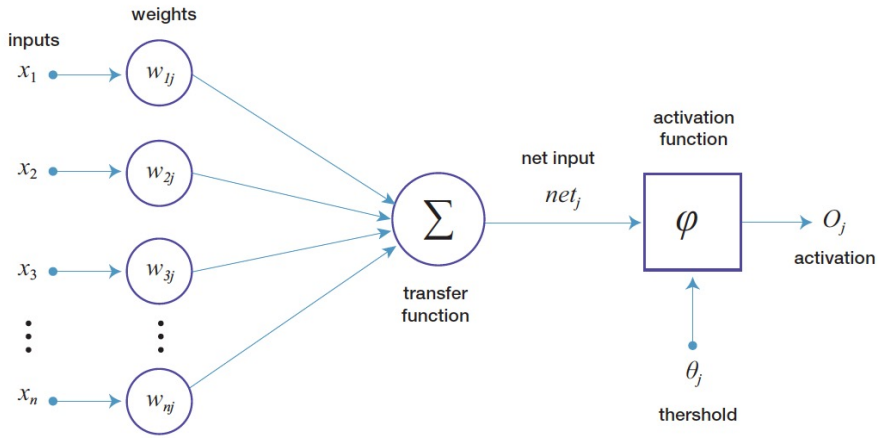
- 다층 퍼셉트론을 이용하면 **선형 분리가 불가능한 문제도 해결할 수 있다는 사실**이 밝혀지면서 다시 인공신경망 연구의 돌파구를 마련하게 됩니다.

- 우리가 **인공신경망(ANN)**이라는 용어를 사용할 때 일반적으로 이는 **다층 퍼셉트론**을 의미합니다. 아래 그림은 다층 퍼셉트론의 구조를 보여줍니다. 다층 퍼셉트론 입력층$_{\text{Input Layer}}$과 은닉층$_{\text{Hidden Layer}}$, 출력층$_{\text{Output Layer}}$으로 구성되어 있습니다. 은닉층은 데이터의 입출력 과정에서 직접적으로 보이진 않지만 숨겨진 특징을 학습하는 역할을 합니다.



다층 퍼셉트론 MLP$_{\text{Multi-Layer Perceptron}}$ = 인공신경망(ANN)

- 다층 퍼셉트론 1개의 층은 여러 개의 노드로 구성됩니다. 아래 그림은 다층 퍼셉트론 1개의 노드에서 일어나는 연산을 보여줍니다. 1개의 노드에서 수행하는 연산은 퍼셉트론 구조와 똑같다는 사실을 알 수 있습니다.



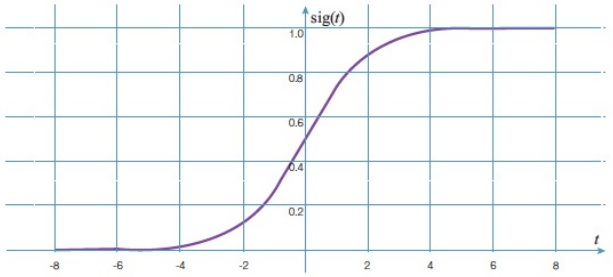
- 각각의 노드들에서 일어나는 연산을 수학적으로 표현하면 아래와 같습니다.

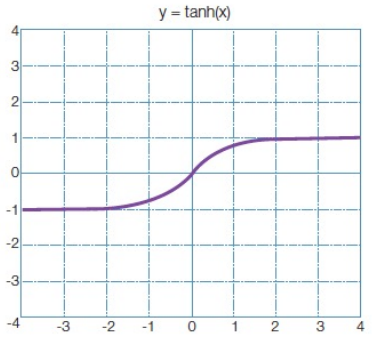
$$y=\sigma(Wx+b)$$

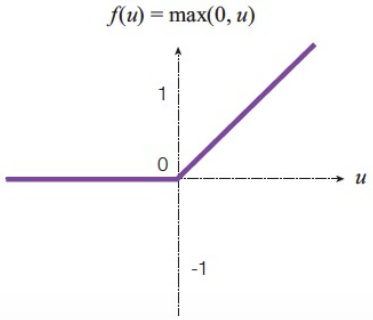
- 이때 σ는 활성 함수로써 퍼셉트론에서는 계단 함수를 사용하여 출력값을 0 또는 1로 이진분류 형태로 만드는 것이 목적이었지만, MLP 구조에서는 분류기가 **비선형적인**$_{\text{Non-Linear}}$ **특징**을 학습할 수 있도록 만드는 것이 목적입니다.

- 따라서 활성 함수로 계단 함수가 아닌 비선형 함수인 시그모이드$_{\text{Sigmoid}}$와 쌍곡 탄젠트$_{\text{Tangent Hyperbolic(Tanh)}}$ 혹은 ReLU를 사용합니다. 활성 함수의 출력 결과인 y를 활성값$_{\text{Activation}}$이라고 부릅니다. 다음장의 그림은 MLP의 대표적인 활성 함수들을 보여줍니다. 과거에는 sigmoid 함수를 많이 사용했지만, 최근에는 ReLU가 딥러닝 학습에 더 적합하다고 알려져서 ReLU를 많이 사용하는 추세입니다.

인공신경망(ANN)의 활성함수 – sigmoid, tanh, ReLU

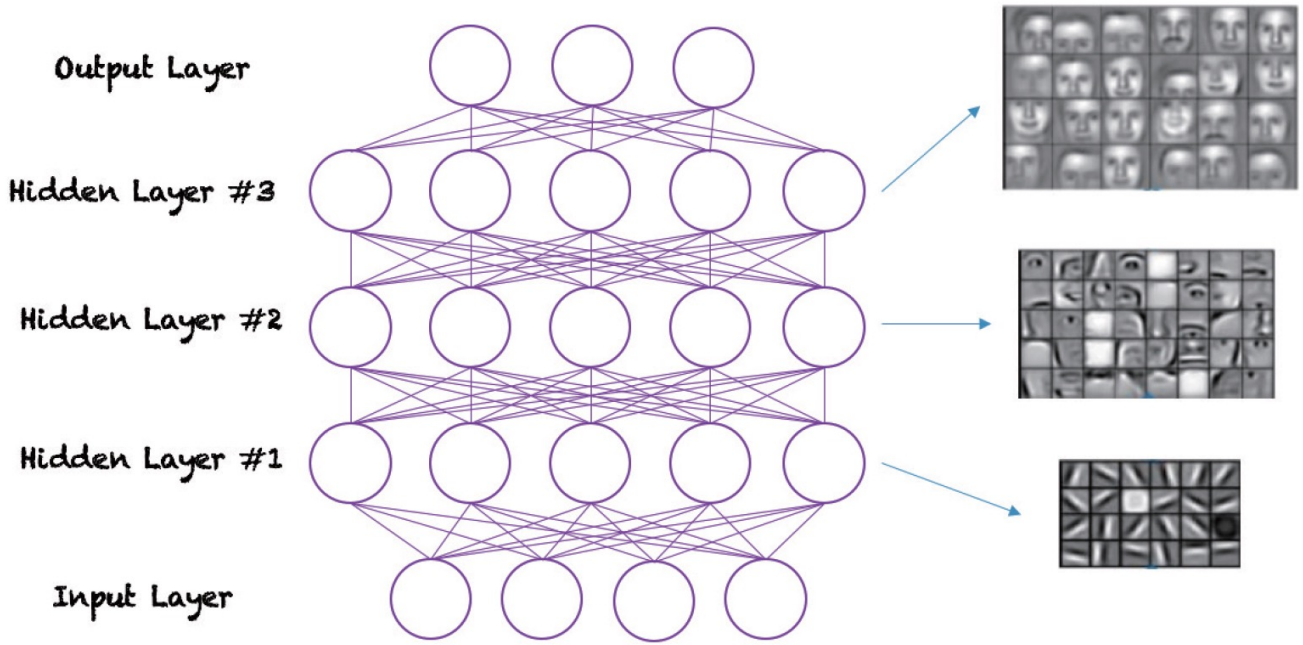






인공신경망(ANN) = 딥러닝(Deep Learning)

- 다층 퍼셉트론에서 은닉층Hidden Layer을 깊게 여러 번 쌓아 올린 형태를 **깊은 인공신경망Deep Neural Networks(DNN)**이라고 부르고 이 구조가 우리가 일반적으로 딥러닝이라고 부르는 기법입니다.
- 이제 우리는 딥러닝 기법이 무엇인지 알게 되었습니다.



TensorFlow 2.0을 이용한 MNIST 숫자분류를 위한 ANN 구현

- MNIST 숫자분류를 위한 ANN을 TensorFlow 2.0 코드로 구현해봅시다.
- https://github.com/solaris33/deep-learning-tensorflow-book-code/blob/master/Ch05-ANN/mnist_classification_using_ann_v2_keras.py

Chapter 4 - 머신러닝 기초 이론들

- 소프트맥스 회귀(Softmax Regression)를 이용한 MNIST 숫자분류기 (Code) (TF v2 Code) (TF v2 Keras Code)
- tf.nn.sparse_softmax_cross_entropy_with_logits API를 사용한 소프트맥스 회귀(Softmax Regression)를 이용한 MNIST 숫자분류기 (Code) (TF v2 Code)

Chapter 5 - 인공신경망(Artificial Neural Networks) - ANN

- ANN을 이용한 MNIST 숫자분류기 구현 (Code) (TF v2 Code) (TF v2 Keras Code)

Thank you!