



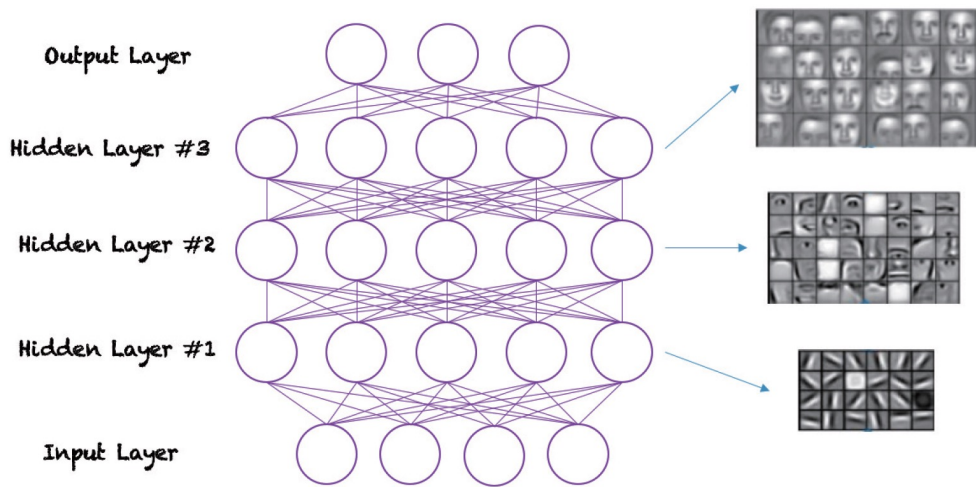
파이토치로 인공지능경망 구현하기

오늘의 학습내용

- 인공지능망(Artificial Neural Networks)
- 크로스 엔트로피 손실함수
- 소프트맥스 회귀(softmax regression)
- One-hot Encoding
- PyTorch로 인공지능망 구현하기

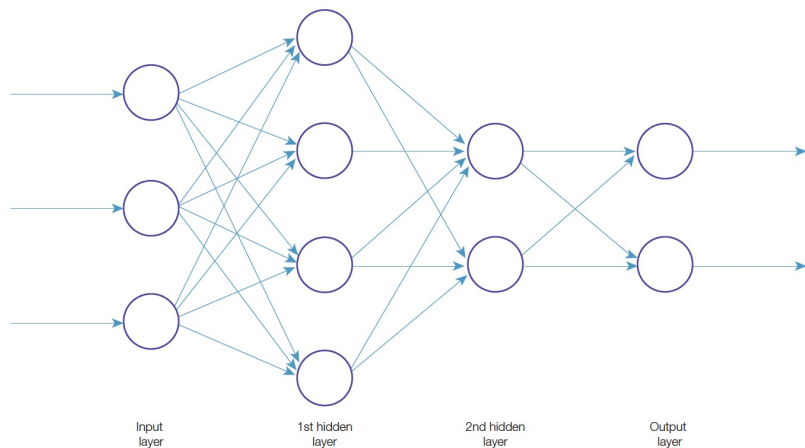
◆ 인공신경망(ANN) = 딥러닝(Deep Learning)

- ❖ 다층 퍼셉트론에서 은닉층 Hidden Layer 을 깊게 여러 번 쌓아 올린 형태를 **깊은 인공신경망** Deep Neural Networks(DNN) 이라고 부르고 이 구조가 우리가 일반적으로 딥러닝이라고 부르는 기법입니다.



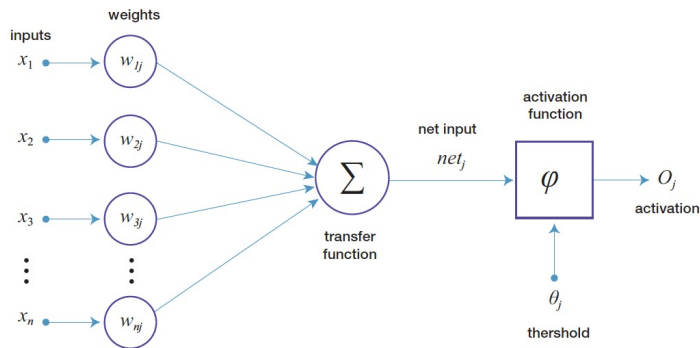
◆ 다층 퍼셉트론 MLP_{Multi-Layer Perceptron} = 인공신경망(ANN)

- ❖ 우리가 인공신경망(ANN)이라는 용어를 사용할 때 일반적으로 이는 다층 퍼셉트론_{Multi-Layer Perceptron}(MLP)을 의미합니다. 아래 그림은 다층 퍼셉트론의 구조를 보여줍니다.
- ❖ 다층 퍼셉트론 입력층_{Input Layer}과 은닉층_{Hidden Layer}, 출력층_{Output Layer}으로 구성되어 있습니다. 은닉층은 데이터의 입출력 과정에서 직접적으로 보이지 않지만 숨겨진 특징을 학습하는 역할을 합니다.



◆ 다층 퍼셉트론 MLP_{Multi-Layer Perceptron} = 인공지능망(ANN)

- ❖ 다층 퍼셉트론 1개의 층은 여러 개의 노드로 구성됩니다. 아래 그림은 다층 퍼셉트론 1개의 노드에서 일어나는 연산을 보여줍니다.

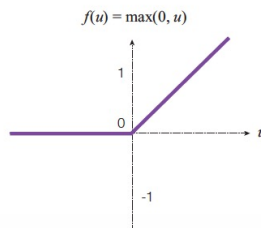
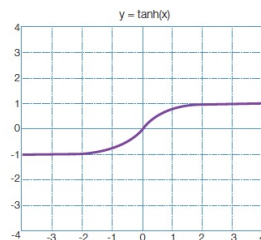


- ❖ 각각의 노드들에서 일어나는 연산을 수학적으로 표현하면 아래와 같습니다.

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

- ❖ 이때 σ 는 활성 함수로써 분류기가 **비선형적인 Non-Linear 특징**을 학습할 수 있도록 만드는 것이 목적입니다.
- ❖ 따라서 활성 함수로 비선형 함수인 시그모이드 Sigmoid와 쌍곡 탄젠트 Tangent Hyperbolic(Tanh) 혹은 ReLU를 사용합니다. 활성 함수의 출력 결과인 y 를 활성값 Activation이라고 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 활성화함수 – sigmoid, tanh, ReLU

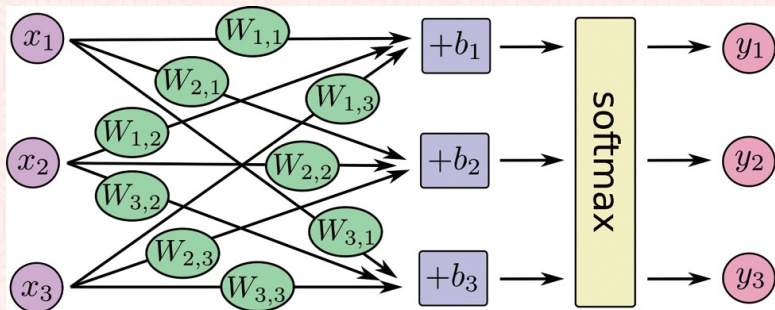


소프트맥스 회귀 Softmax Regression

- ❖ 이번 장에서는 **소프트맥스 회귀** Softmax Regression 기법에 대해 알아보시다. 소프트맥스 회귀는 n개의 레이블을 분류하기 위한 가장 기본적인 모델입니다. 모델의 아웃풋에 Softmax 함수를 적용해서 모델의 출력값이 각각의 레이블에 대한 확신의 정도를 출력하도록 만들어주는 기법입니다.
- ❖ 구체적으로 Softmax 함수는 Normalization 함수로써 출력값들의 합을 1로 만들어줍니다. Softmax 함수는 아래 수식으로 표현됩니다.

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

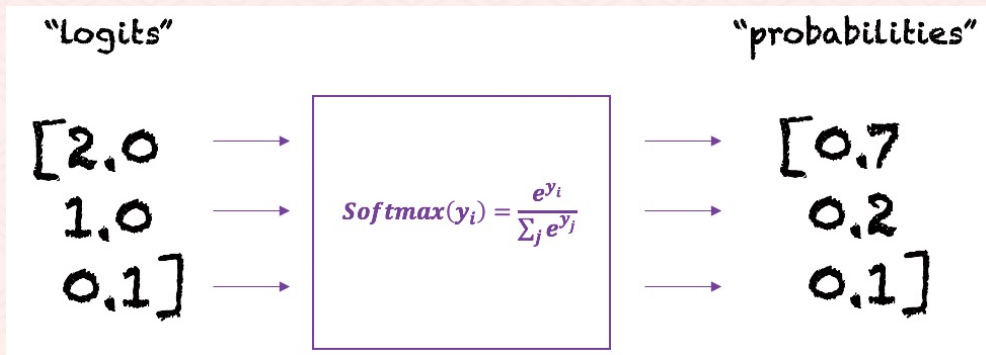
- ❖ Softmax 함수를 마지막에 씌우게 되면 모델의 출력값이 레이블에 대한 확률을 나타내게 됩니다(=출력값들의 합이 1이 되므로). 아래 그림은 소프트맥스 회귀의 구조를 보여줍니다.



$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax} \left(\begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

◆ 소프트맥스 회귀 Softmax Regression

- ❖ 아래 그림은 모델의 출력값 Logits에 Softmax 함수를 적용한 결과를 보여줍니다.
- ❖ 이 모델은 인풋에 대해서 각각의 레이블일 확률을 **0.7(70%)**, **0.2(20%)**, **0.1(10%)**로 확신하는 것으로 해석할 수 있습니다. 예를 들어, 어떤 이미지가 [개, 고양이, 말]인지 분류하는 분류기를 만든 경우, 이 모델은 지금 인풋으로 들어온 이미지가 70% 확률로 개, 20% 확률로 고양이, 10% 확률로 말이라고 판단한 것으로 해석할 수 있습니다.



◆ 크로스 엔트로피 Cross-Entropy 손실 함수

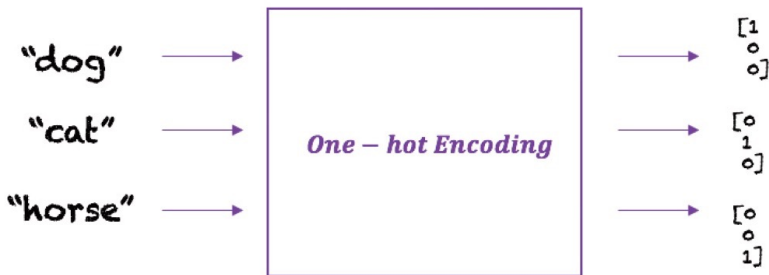
- ❖ 소프트맥스 회귀 Softmax Regression를 비롯한 분류 문제에는 크로스 엔트로피 Cross-Entropy 손실 함수를 많이 사용합니다. 크로스 엔트로피 손실 함수도 평균제곱오차(MSE)와 같이 모델의 예측값이 참값과 비슷하면 작은 값, 참값과 다르면 큰 값을 갖는 형태의 함수로 아래와 같은 수식으로 나타낼수 있습니다.

$$H_{y'}(y) = - \sum_i y'_i \log(y_i)$$

- ❖ 위 수식에서 y' 는 참값, y 는 모델의 예측값을 나타냅니다. 일반적으로 분류 문제에 대해서는 MSE보다 크로스 엔트로피 함수를 사용하는 것이 학습이 더 잘되는 것으로 알려져 있습니다.
- ❖ 따라서 대부분의 텐서플로 코드들에서 크로스 엔트로피 손실 함수를 사용합니다. 본 강의에서도 대부분의 코드 구현에서 크로스 엔트로피 함수를 손실 함수로 사용합니다.

One-hot Encoding

- ❖ **One-hot Encoding**은 범주형 값 Categorical Value을 이진화된 값 Binary Value으로 바꿔서 표현하는 것을 의미합니다. 범주형 값은 예를 들어 "개", "고양이", "말"이라는 3개의 범주형 데이터가 있을 때 이를 ["개"]=1, "고양이"]=2, "말"]=3이라고 단순히 Integer Encoding으로 변환하여 표현하는 것입니다.
- ❖ 이에 반해 One-hot Encoding을 사용하면 범주형 데이터를 "개"=[1 0 0], "고양이"=[0 1 0], "말"=[0 0 1] 형태로 해당 레이블을 나타내는 인덱스만 1의 값을 가지고 나머지 부분은 0의 값을 가진 Binary Value로 표현합니다.



- ❖ 단순한 Integer Encoding의 문제점은 머신러닝 알고리즘이 정수 값으로부터 잘못된 경향성을 학습하게 될 수도 있다는 점입니다. 예를 들어, 위의 예시의 경우 Integer Encoding을 사용할 경우 머신러닝 알고리즘이 ["개"(=1)와 "말"(=3)의 평균($1+3/2=2$)은 "고양이"(=2)이다.] 라는 지식을 학습할 수도 있는데, 이는 명백히 잘못된 학습입니다. 따라서 머신러닝 알고리즘을 구현할 때 타겟 데이터를 One-hot Encoding 형태로 표현하는 것이 일반적입니다.

◆ MNIST 데이터셋

- ❖ 머신러닝 모델을 학습시키기 위해서는 데이터가 필요합니다. 하지만 데이터를 학습에 적합한 형태로 정제하는 것은 많은 시간과 노력이 필요한 일입니다. 따라서 많은 연구자가 학습하기 쉬운 형태로 데이터들을 미리 정제해서 웹상에 공개해놓았습니다.
- ❖ 그중에서도 **MNIST 데이터셋**은 머신러닝을 공부하는 사람들이 가장 먼저 접하게 되는 데이터셋으로 머신러닝 분야의 "Hello World"라고 볼 수 있습니다.
- ❖ 구체적으로 MNIST 데이터는 60,000장의 트레이닝 데이터와 10,000장의 테스트 데이터로 이루어진 데이터셋으로 0~9사이의 28×28 크기의 필기체 이미지로 구성되어있습니다. 아래 그림은 MNIST 데이터의 예제를 보여줍니다.



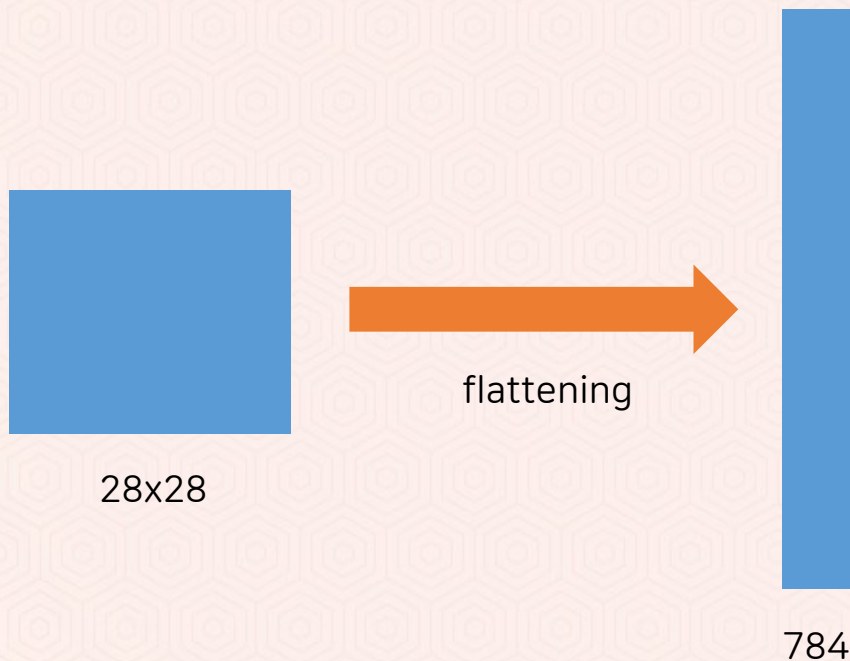
◆ Fashion-MNIST 데이터셋

- ❖ **Fashion-MNIST** 데이터셋은 MNIST 데이터셋과 유사하게 28x28 grayscale 이미지와 10개의 레이블로 구성된 데이터셋입니다.
- ❖ 다음과 같이 10개의 패션 아이템에 대한 레이블로 구성되어 있습니다.
- ❖ "T-shirt/top", "Trouser", "Pullover", "Dress", "Coat", "Sandal", "Shirt", "Sneaker", "Bag", "Ankle boot"



◆ Flattening

❖ 이미지 형태의 matrix를 긴 vector 형태로 변경하는 작업을 flattening이라고 합니다.



▶ 파이토치(PyTorch)를 이용해서 인공신경망 구현하기

- ❖ 파이토치(PyTorch)를 이용해서 인공신경망을 구현해봅시다.
- ❖ 3강_PyTorch로_인공신경망_구현하기.ipynb
- ❖ https://colab.research.google.com/drive/1Zm_1pr4eF3n-a1KLH-gUvINjyuabDXam?usp=sharing