

Lecture 1

Simple Machine Learning & Neural Network

A-Task/Yeolkwo Choi

2023. 7. 5

Coarse Introduction

Basic	Intermediate
<p>신경망 관련 기본 내용</p> <ul style="list-style-type: none"> - 신경망의 구성 / 기본 용어 - 신경망의 작동 원리 - 활성화 함수 (activation function) - 최적화 함수 (optimizers) <p>기본 딥러닝 알고리즘</p> <ul style="list-style-type: none"> - Feedforward Neural Network (FNN) - Convolution Neural Network (CNN) - Recurrent Neural Network (RNN) 	<p>생성모델</p> <ul style="list-style-type: none"> - Autoencoder (AE) - Variational Autoencoder (VAE) - Generative Adversarial Network (GAN) <p>CNN 심화</p> <ul style="list-style-type: none"> - 주요 CNN 아키텍처 - 사전학습 모형 사용하기 (Transfer Learning) <p>RNN 심화</p> <ul style="list-style-type: none"> - LSTM, GRU, seq2seq <p>Attention & Transformer</p>

Prerequisites

- Linear Algebra
- Basic python

Contents

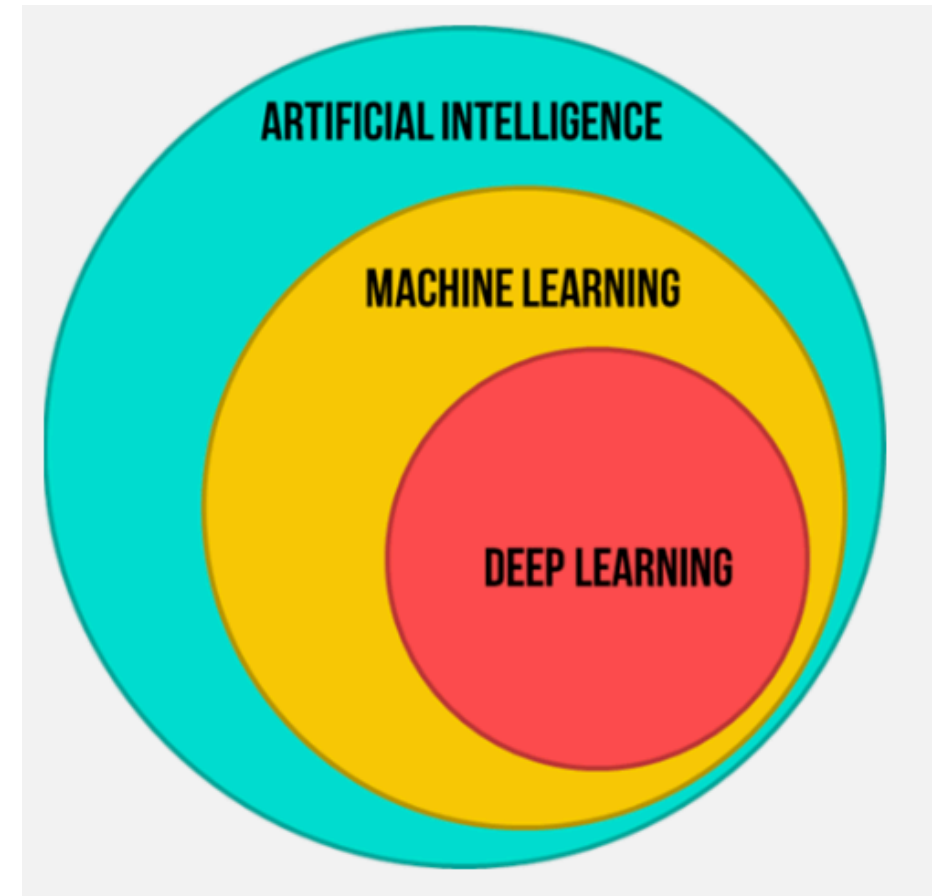
딥러닝의 기본 내용

- 신경망의 구성 / 기본 용어
- 기본 신경망 (ANN or FNN, a.k.a., MPL)의 작동원리
- 활성화 함수 (Activation function)
- 신경망에서의 경사하강법
 - Backward propagation of errors
 - Vanishing Gradient
 - Various Optimizers
 - Weight Initialization
- 신경망에서의 과적합문제

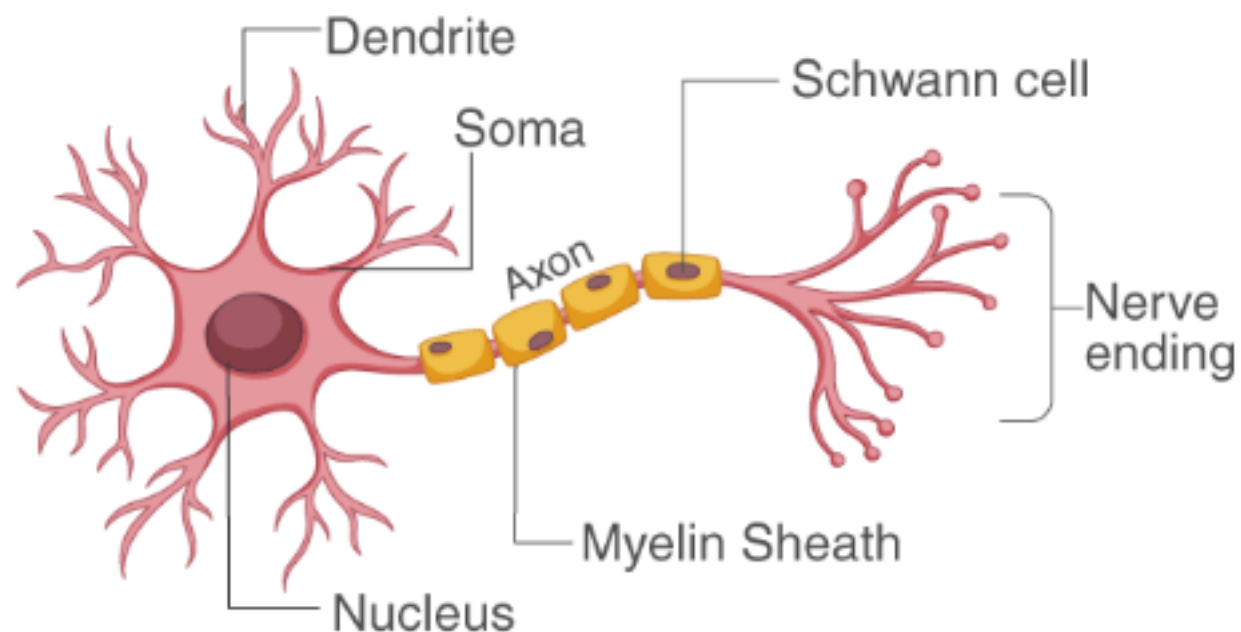
Regularization, Dropout, Early Stopping, Batch Normalization

파이썬 코딩

- Pytorch tutorial
- 선형회귀문제



Deep Learning

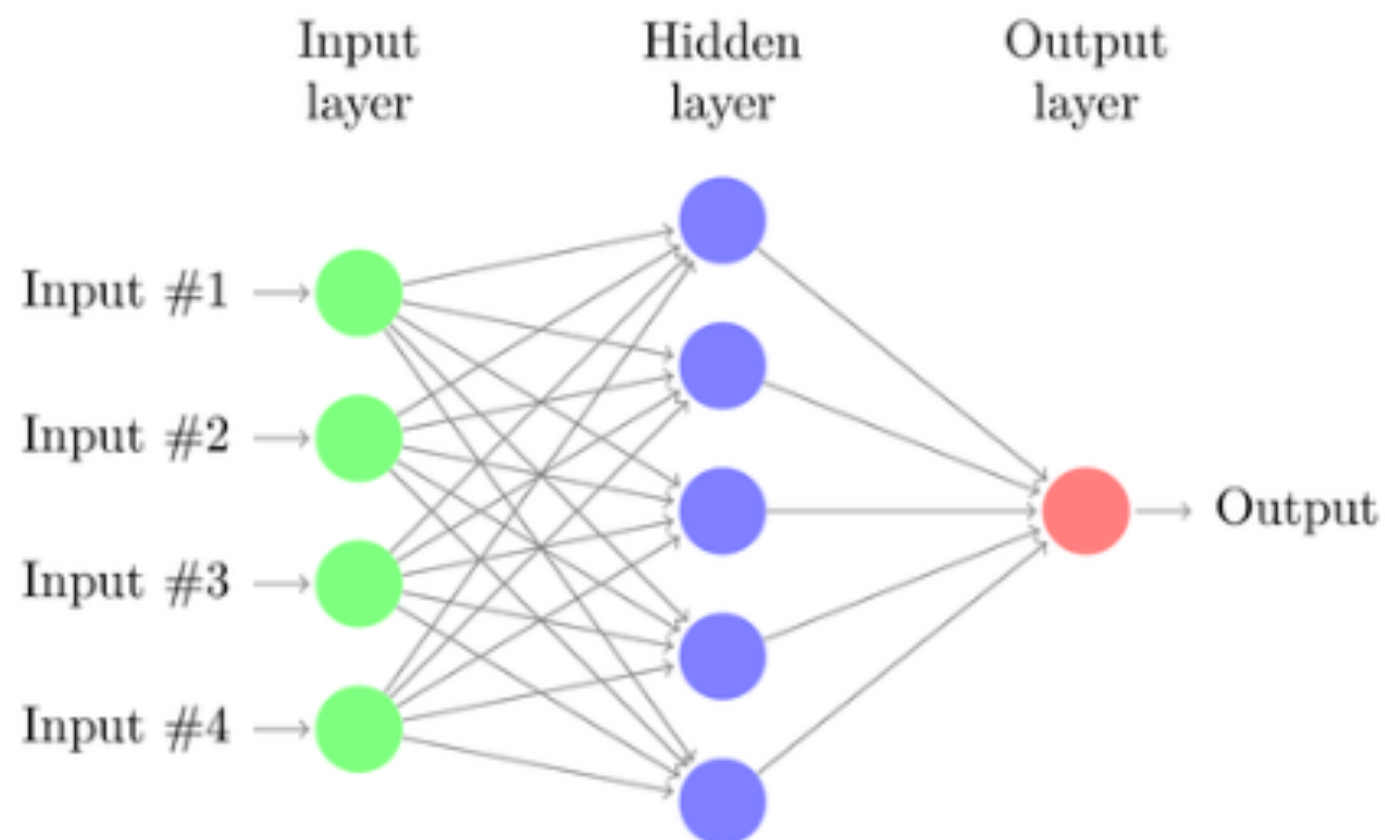


3 basic layers

Input layer : put feature information and transfer it to the hidden layer

Hidden layer : extract important feature

Output layer : prediction

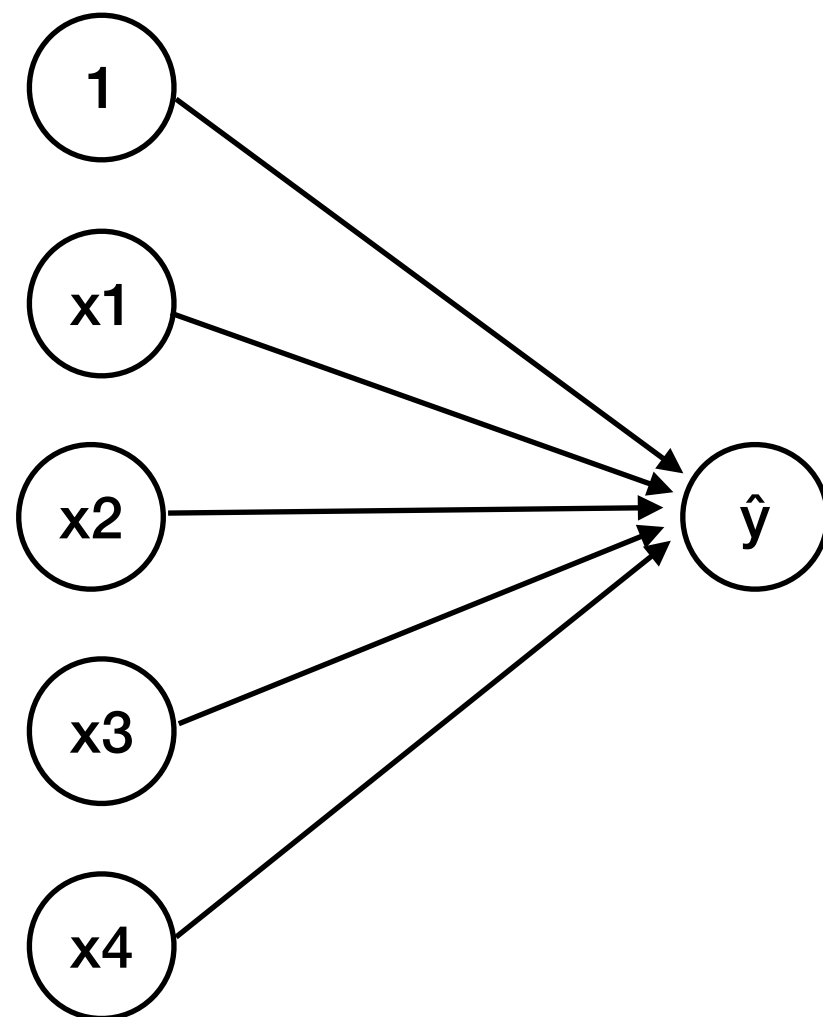


of hidden layer ≥ 1

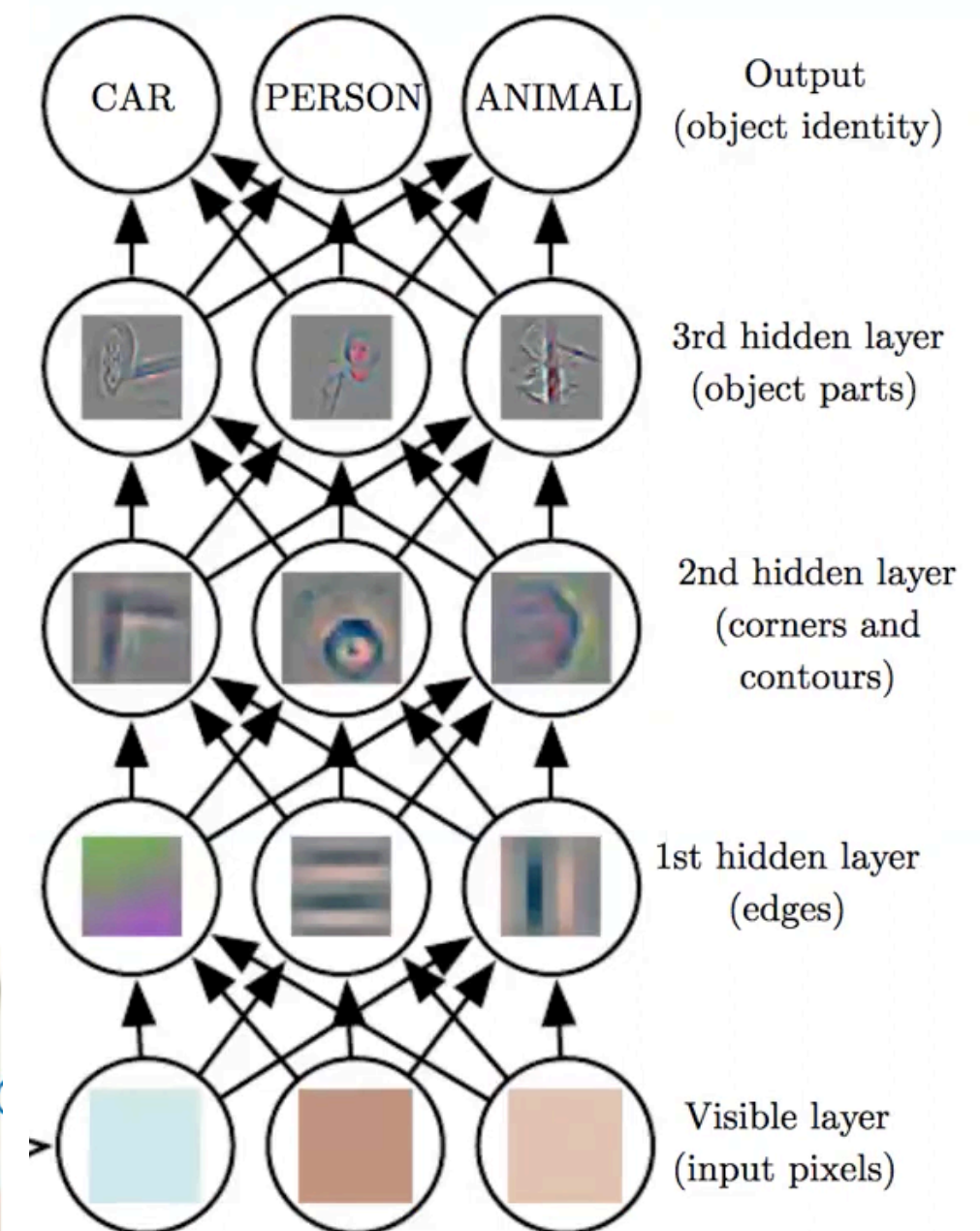
Machine Learning vs Deep Learning

Multivariable Linear regression

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$



Convolution Neural Network



Example of DNN

simple CNN: <https://transcranial.github.io/keras-js/#/mnist-cnn>

Neural Network: <http://playground.tensorflow.org/>

The image shows the Keras.js web application interface for a Basic Convnet for MNIST. On the left is a sidebar with the Keras.js logo and a list of demos. The main area has a green header with a menu icon, the title 'Basic Convnet for MNIST', and an info icon. Below the header is a large drawing area with the text 'Draw any digit (0-9) here' and a green border. To the right of the drawing area is a 'use GPU' toggle switch (which is turned on) and a 'CLEAR' button with a red 'X' icon. Below the drawing area is a 'Conv2D' section with the text '32 3x3 filters, padding valid, 1x1 strides'. At the bottom right, there are four buttons labeled '0', '1', '2', and '3'.

Keras.js

DEMOS

- Basic Convnet MNIST
- Convolutional VAE MNIST
- AC-GAN MNIST
- ResNet-50 ImageNet
- Inception v3 ImageNet
- DenseNet-121 ImageNet
- SqueezeNet v1.1 ImageNet
- Bidirectional LSTM IMDB
- Image Super-Resolution

LINKS

- [GitHub repo](#)
- [MD.ai](#)

Basic Convnet for MNIST

Draw any digit (0-9) here

☒ use GPU

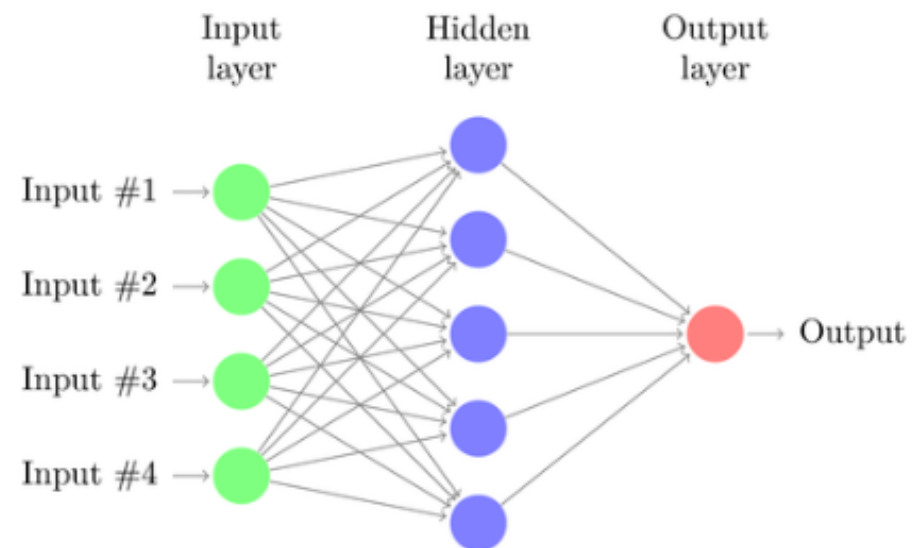
0 1 2 3

Conv2D
32 3x3 filters, padding valid, 1x1 strides

CLEAR

Regression vs Classification

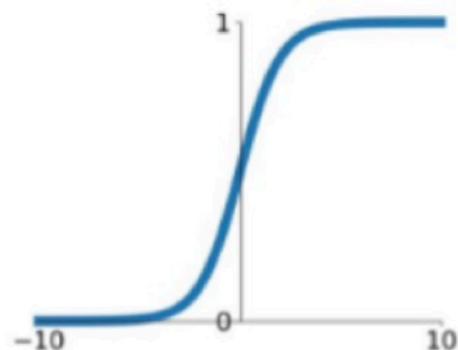
How many the number of output nodes?



Activation Function

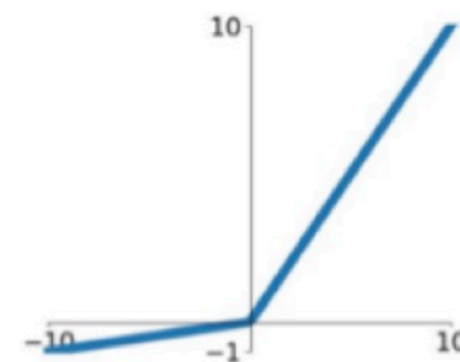
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



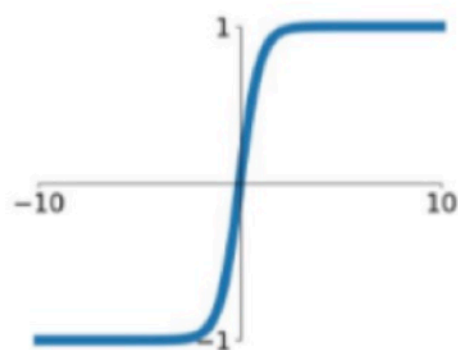
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

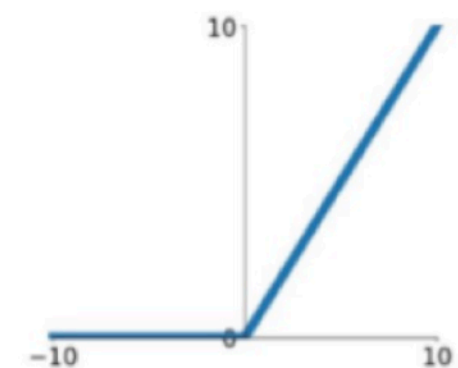


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

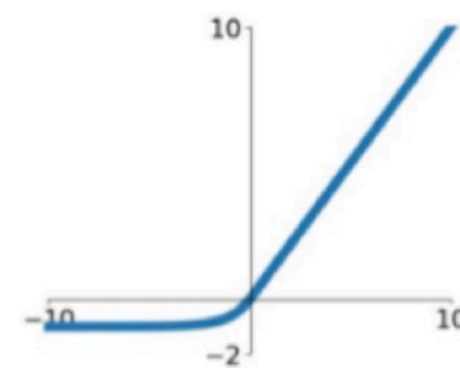
ReLU

$$\max(0, x)$$

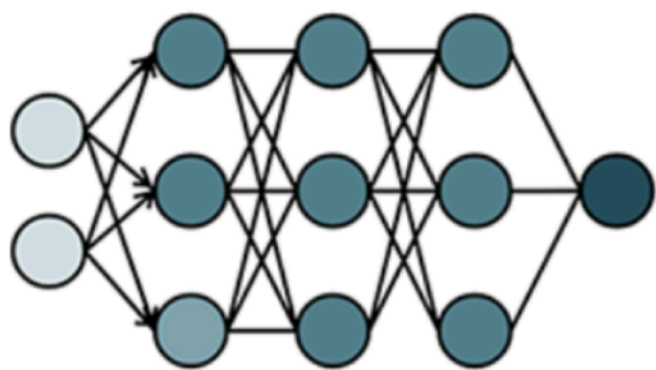


ELU

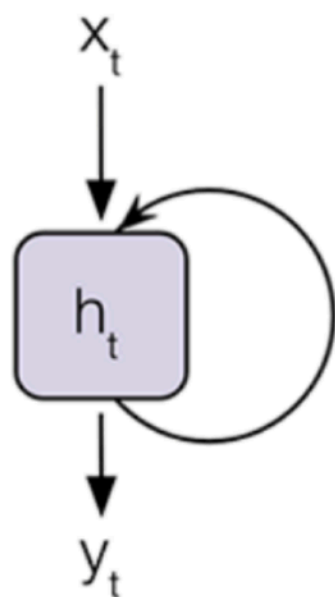
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



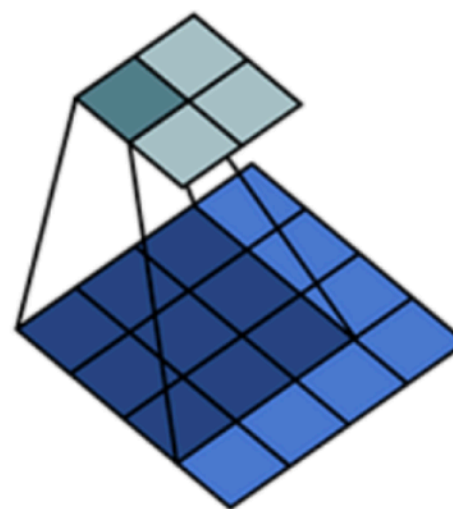
Type of Neural Network



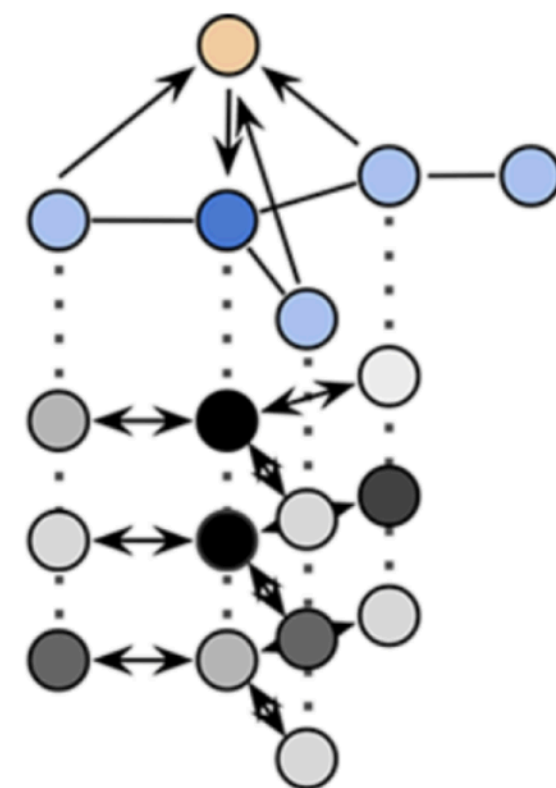
A. feed-forward
neural network
(NN)



B. recurrent
neural network
(RNN)



C. convolutional
neural network
(CNN)



D. graph
neural network
(GNN)