

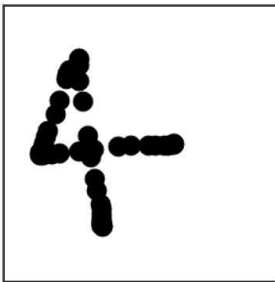
1번

[Fork me on GITHUD](#)

## GPU Deep Learning Demo

Please draw a digit into this canvas.

(It will probably only work in Firefox and Chrome. And it may not work on mobile. It should look like [this](#))



4

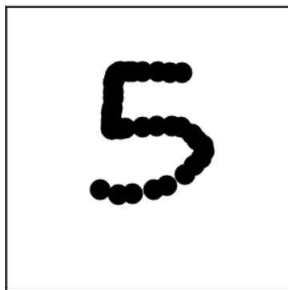
How does this work?

[Fork me on GITHUD](#)

## GPU Deep Learning Demo

Please draw a digit into this canvas.

(It will probably only work in Firefox and Chrome. And it may not work on mobile. It should look like [this](#))

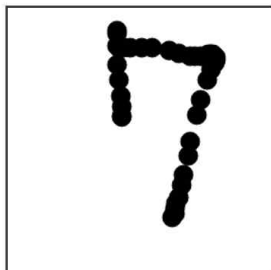


5

## GPU Deep Learning Demo

Please draw a digit into this canvas.

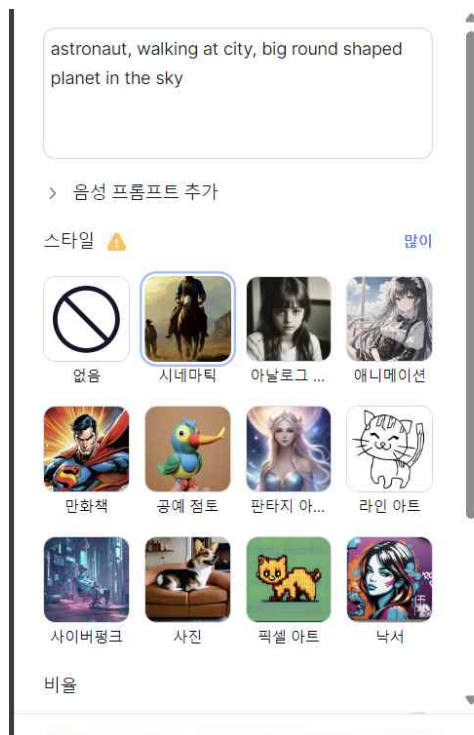
(It will probably only work in Firefox and Chrome. And it may not work on mobile. It should look like [this](#))



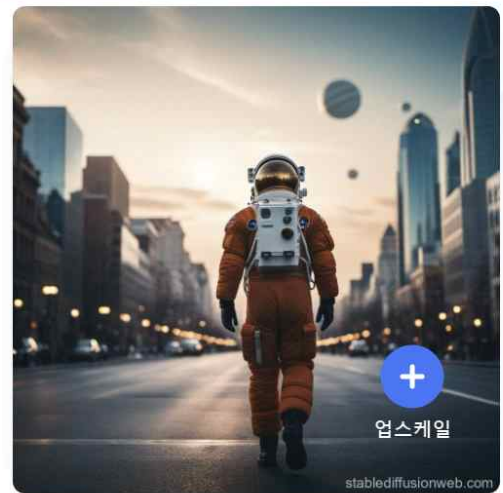
7

How does this work?

2번



⚡  
현재 무료 요금제를 사용하고 있습니다.  
우선 순위 세대, 추가 토큰 크레딧, 광고 없이, 그리고 더 많은 기능을 위해 업그레이드



3번- 머신러닝의 종류에 속하지 않는 것은?

>>④자연어 처리

4번- 지도학습, 비지도학습, 강화학습

5번- 머신러닝 활용 분야중 성질이 다른 하나? >>③금융, 투자, 비즈니스적 가격 판단

6번

(1,1)과 가장 가까운 거리에 위치한 3개의 점을 찾는다.

유클리드 거리( $d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$ )를 사용하면

(1,1)과 가장 가까운 거리에 위치한 3개의 점은 (1,2),(0,1),(1,0)이며 모두 1만큼 떨어진 거리에 위치한다.

세 점 중 ■ 클래스가 (1,2),(0,1)로 2개, ○클래스가 (1,0)으로 1개이므로  
(1,1)은 ■ 클래스에 속한다.

7번

-손실함수란?

: 예측값과 실제값(레이블)의 차이를 구하는 기준이다.

학습률 알고리즘이 잘못 예측하는 정도를 확인하기 위한 함수이며, 최적화를 위해 이를 최소화하는 것이 목적이다.

-손실함수의 종류

1) MSE(Mean Square Error): 예측값과 실제값 사이의 평균 제곱 오차를 합한 것.

2) RMSE: MSE에 루트를 씌운 값. MSE와 기본적으로 동일하지만 MSE가 오류의 제곱을 구하기 때문에 실제 오류 평균에 비해 왜곡되는 문제를 해결한다.

3) MAE: 예측값과 실제값 사이의 차에 절대값을 씌우고 그 값을 전부 더해 값의 개수로 나누어 평균한 값을 구한다.

8. 정밀도=TP/(TP+FP)=2/((2+7)=22.2%

재현율=TP/(TP+FN)=2/(2+1)=66.6%

9.

1)활성화 함수의 종류

-계단함수: 입력이 0보다 크면 1을 출력하고 그 외에는 0을 출력한다.

-시그모이드함수: 입력이 음수면 0에 수렴하는 값을 출력, 양수면 1에 수렴한다.

-TanH 함수: 입력값이 작은 신호일 시 -1에 가까운 숫자로 바꾸어 내보낸다.

-ReLU함수: 입력값이 0보다 작을 시 0으로 바꾸어 출력, 0보다 클 시 입력값을 출력한다.

2) 퍼셉트론에서 사용한 함수

: 입력이 0보다 크면 1을 출력, 그 외에는 0을 출력하는 계단함수를 사용하였다.

미분과 경사하강법을 사용할 수 없다는 단점이 있다.

10. 퍼셉트론이 분리하지 못하는 입력: ◎

(c는 하나의 직선으로 두 값을 완전히 분리할 수 없는 경우이다.)

11.  $x_1, x_2$ 의 입력이 0.0일 경우 출력은  $(1*0+1*0-1.5)<0$ 이므로 0이다.

바이어스를 0.5로 변경할 경우에도 입력값과 가중치가 동일하므로 출력은  $(1*0+1*0-0.5)<0$ 으로 역시 0이다.

12. 강화학습의 4가지 구성요소

1)에이전트: 환경에서 어떤 행동을 취할지 결정하는 학습주체

2)행동: 에이전트가 환경에서 선택할 수 있는 행동

3)환경: 에이전트 행동에 반응해 보상을 제공받는 공간

4) 보상: 환경이 에이전트 행동에 따라 제공하는 피드백. 보상의 최대화가 목표이다.

13.

(가): 최초의 머신러닝 프로그램인 체커. 경험을 통해 학습하며 알파고 등의 AI 바둑 소프트웨어에 영향을 주었다.

(나): 애플 소프트웨어 기기들에서 작동하는 인공지능 개인 비서 프로그램 시리.

(다): 마크 I 퍼셉트론 신경망 컴퓨터. 20x20의 화소를 가졌으며 연결선으로 연결강도를 조절해 학습이 가능하다.

(라): 역전파 알고리즘. 입력이 주어지면 순방향으로 계산해 출력을 계산하고, 실제 출력과 원하는 출력간 오차를 계산하여 오차를 다시 역방향으로 전파하면서 가중치를 변경한다.

(마): 컨볼루션 신경망. 시각세포의 작동 원리를 모방하여 영상에서의 물체가 어떤 물체일지의 가능성을 특징한다. 이미지를 특정한 영역별로 추출하여 학습하기에 부분의 특징을 찾아낼 수 있다.

(바): 심층 신경망. 여러개의 은닉층을 사용하며 다층 신경망에 학습을 통한 전처리과정을 추가하였다. 대규모 데이터를 사용한다.

개발 연도에 따라 정렬>> 가 다 라 마 바 나

14.

-GPU란: Graphic Processing Unit. 수많은 산술논리연산유닛이 탑재되어 CPU보다 빠른 기계학습이 가능하고, 대량 연산에 특화되어 있다. 게임, CG 등에도 활용할 수 있는 범용 칩이 사용되어 머신러닝 전용의 칩보다는 효율이 떨어진다.

-NPU란: Neural network Processing Unit. 대량 연산 동시 수행에 특화되었으며 머신러닝 전용으로 설계되어 범용성은 떨어지지만 GPU보다 효율적인 연산이 가능하다.

-TPU: 구글이 개발한 NPU의 일종이다. Tensor Processing Unit. 스스로 하드웨어를 준비하지 않아도 고효율의 머신러닝 작업이 가능하게 해준다. 딥러닝 작업에 특화되어 있으며 대규모 연산작업에서의 빠른 속도와 전력 효율을 보장한다.