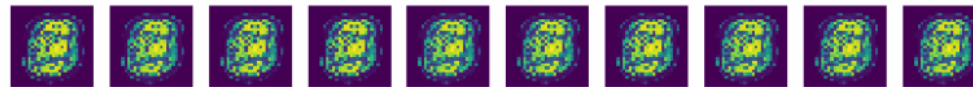




# 딥러닝 개론

July 4, 2017

Seung-Chan Kim, Ph. D



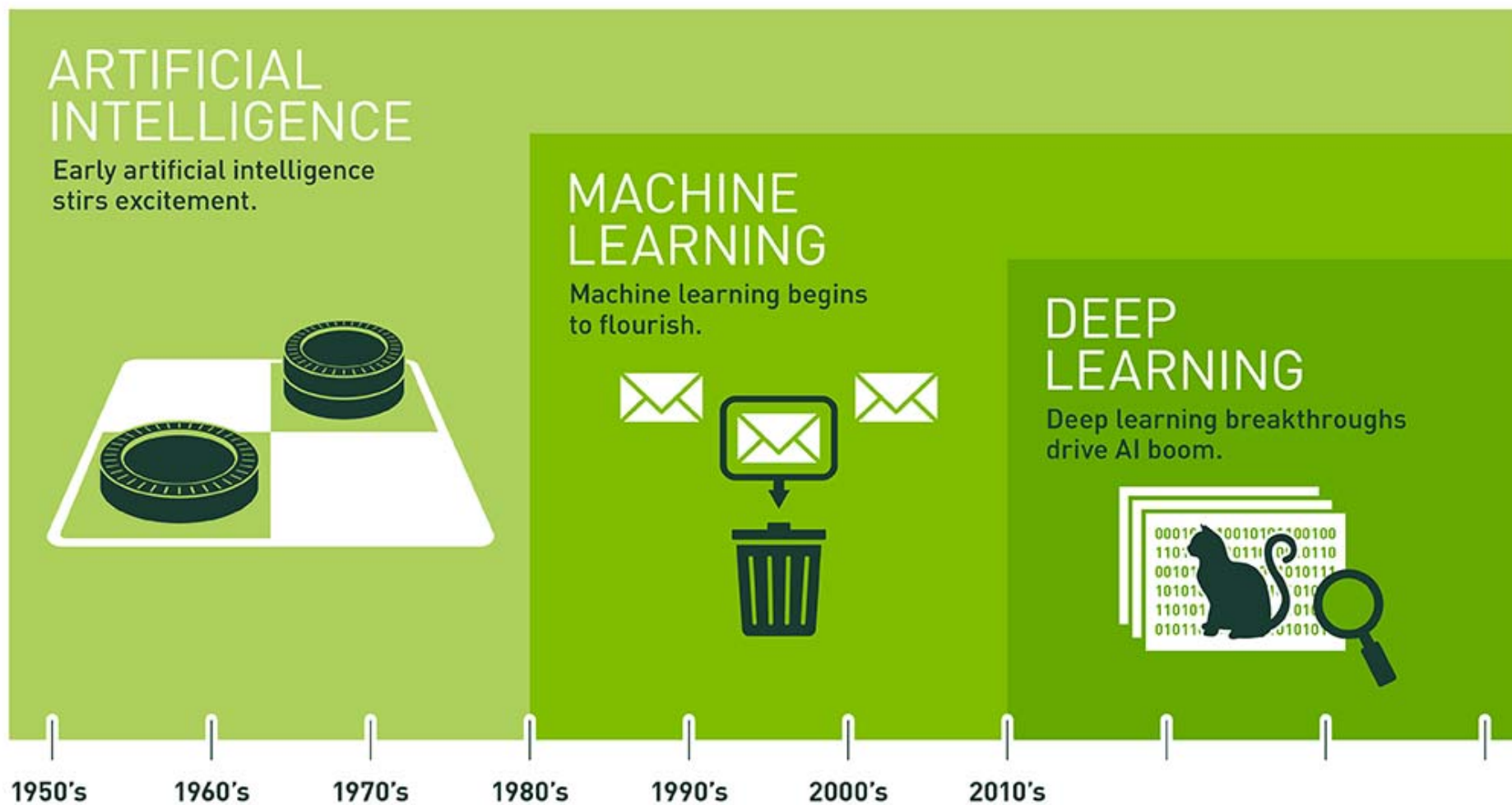


All my great stuff, I learned outside of school

<http://www.cnbc.com/2016/04/21/steve-wozniak-school-is-not-enough-go-beyond-it.html>



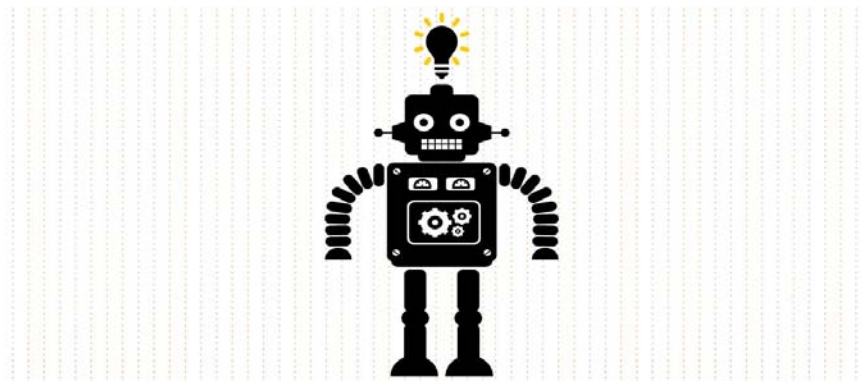
- 1. 머신러닝 개론 및 주요 개념의 이해. Tensorflow 시스템 설치 및 환경설정 (7/4 화)
- 2. Tensorflow 에 익숙해지기 실습 및 Regression의 이해 (7/6 목)
- 3. Neural Network 이해 및 tensorflow 를 이용한 구현 (7/11 화)
- 4. 이미지 분류 이해 및 Tensorflow를 이용한 구현 (7/13 목)



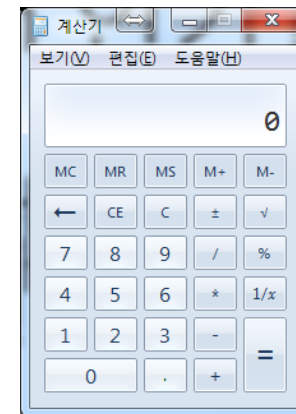


# 왜 Learn 인가?

- 예시 데이터 (**example data**) 또는 과거의 경험 데이터 (**past experience**) 등을 이용하여 컴퓨터가 특정 performance 조건을 최적화 할 수 있도록 프로그래밍 하는 것.



- 월급 등을 계산하기 위해 learn을 할 필요는 없다!





# Machine Learning (ML) 의 분류 (또는 활용)

- Association Analysis



$$P(\text{chips} \mid \text{beer}) = 0.7$$

- Supervised Learning

- Classification
- Regression/Prediction



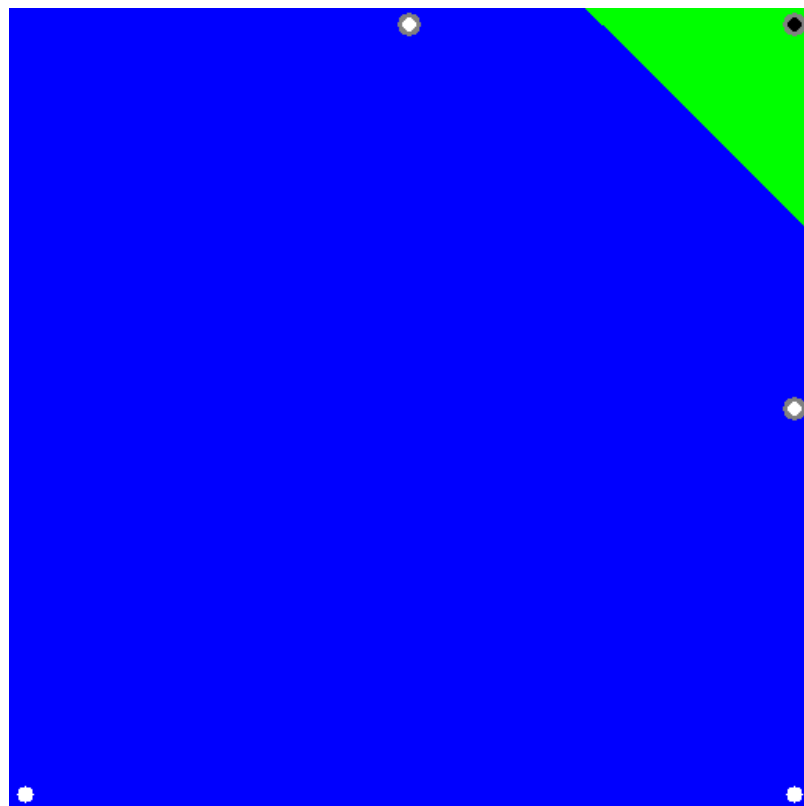
- Unsupervised Learning



- Reinforcement Learning



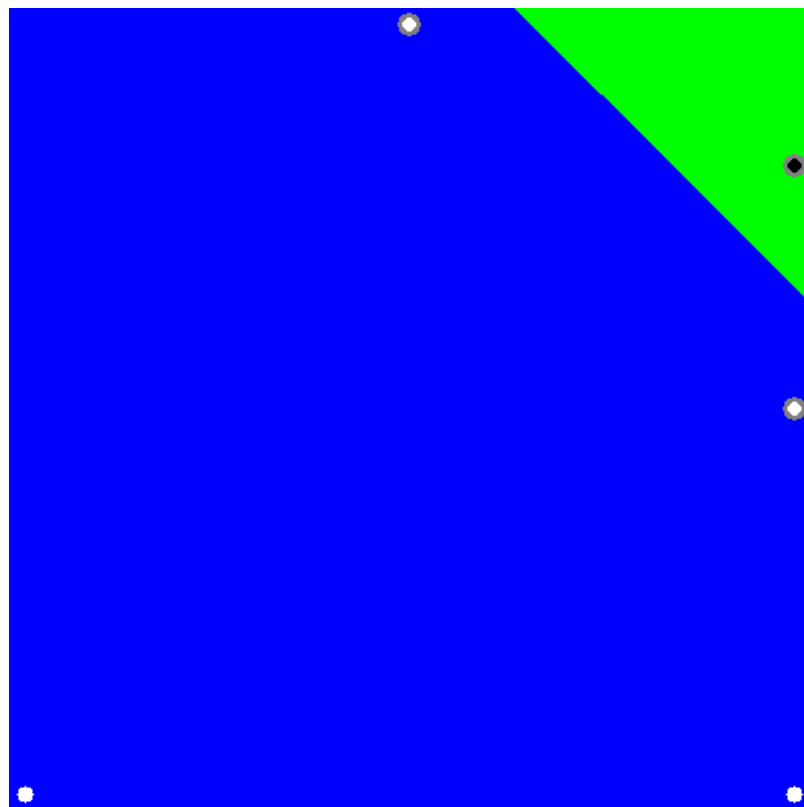
0



1



0

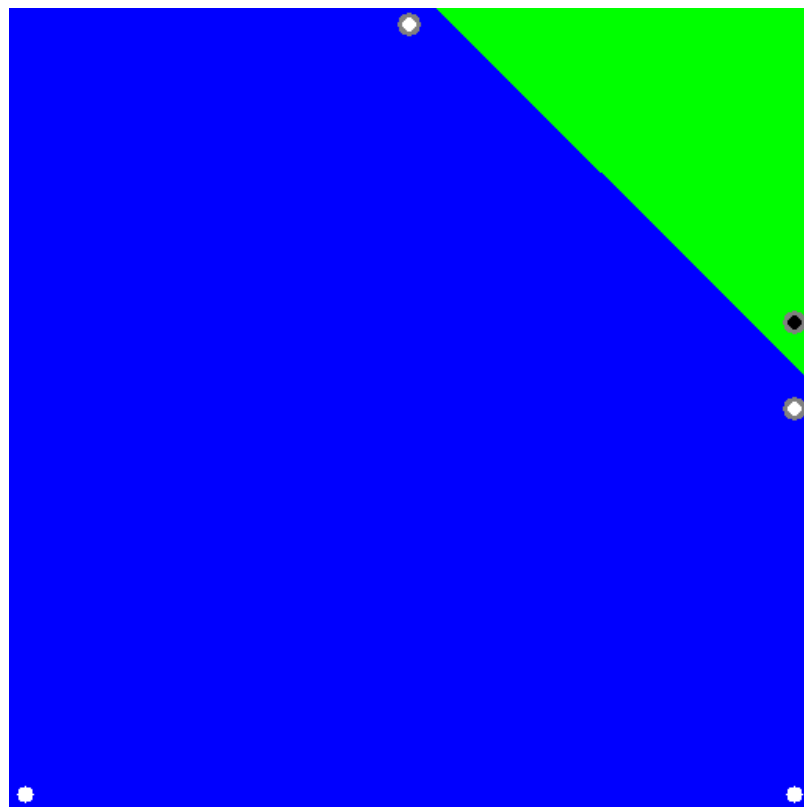


1





0



1







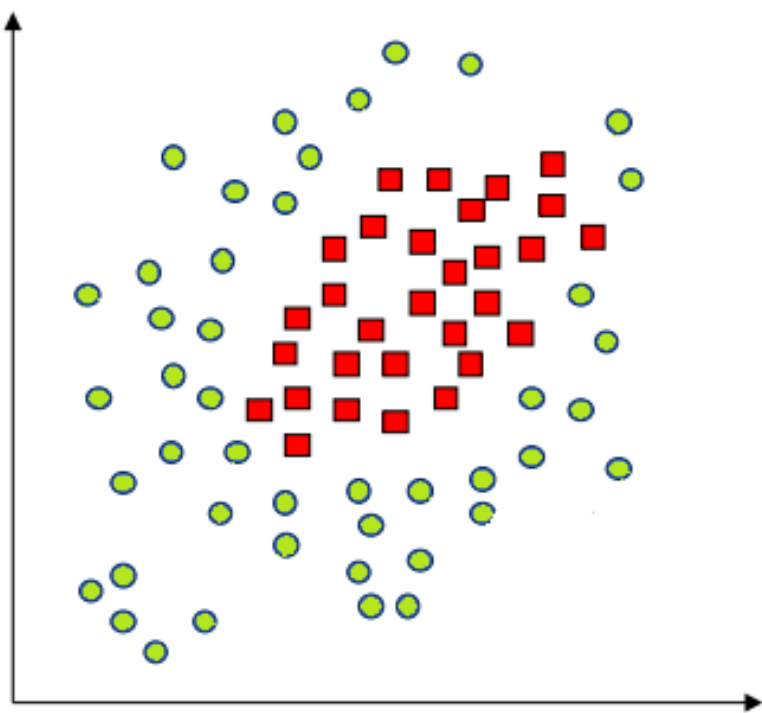


Quiz ?!



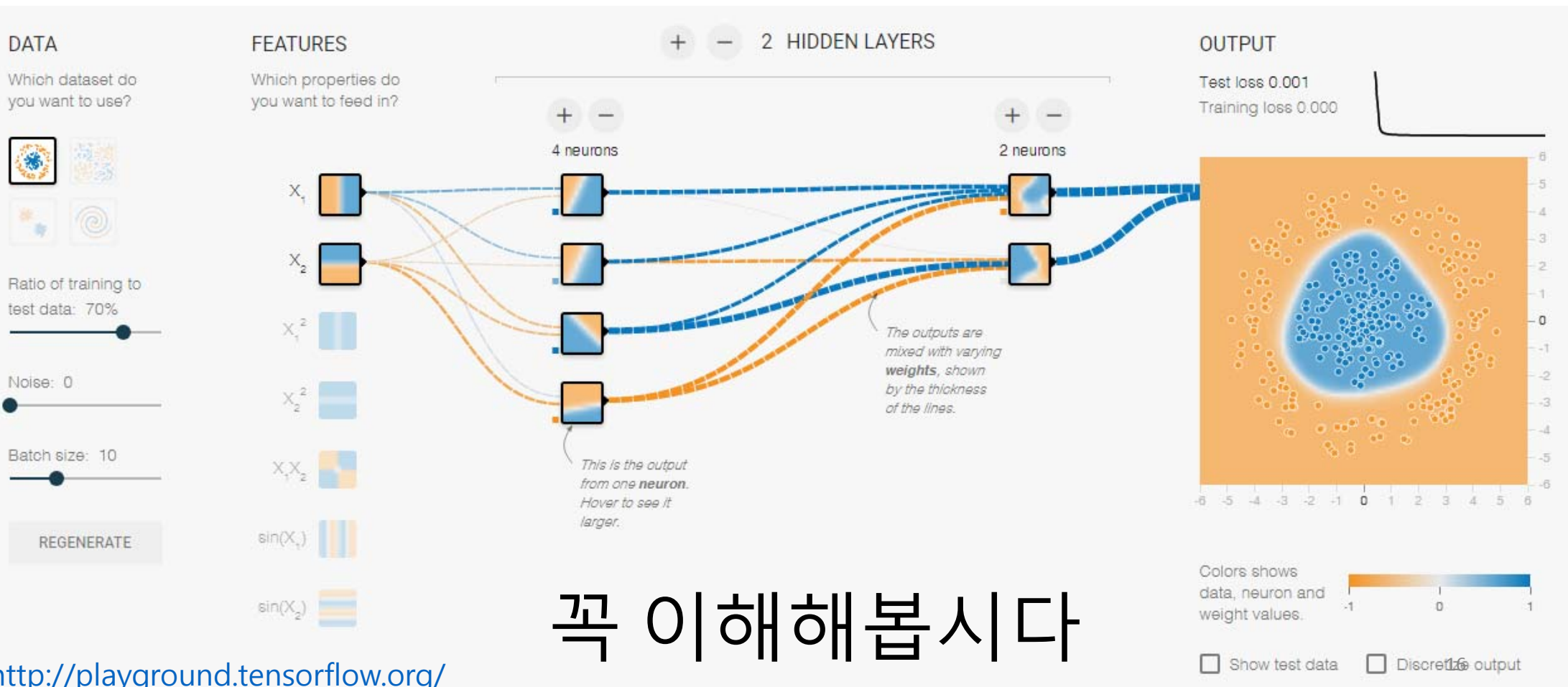


# Quiz !





# Neural Network : playground.tensorflow.org

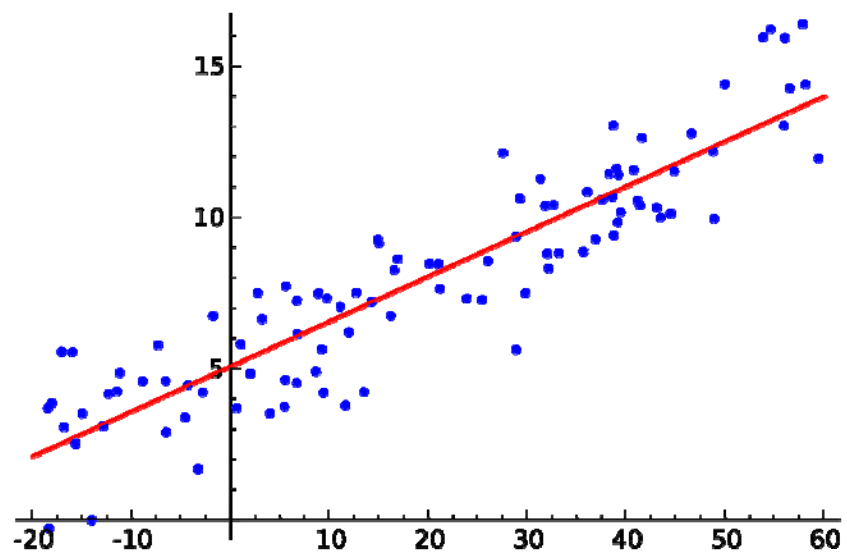


꼭 이해해봅시다

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



# Regression (회귀)

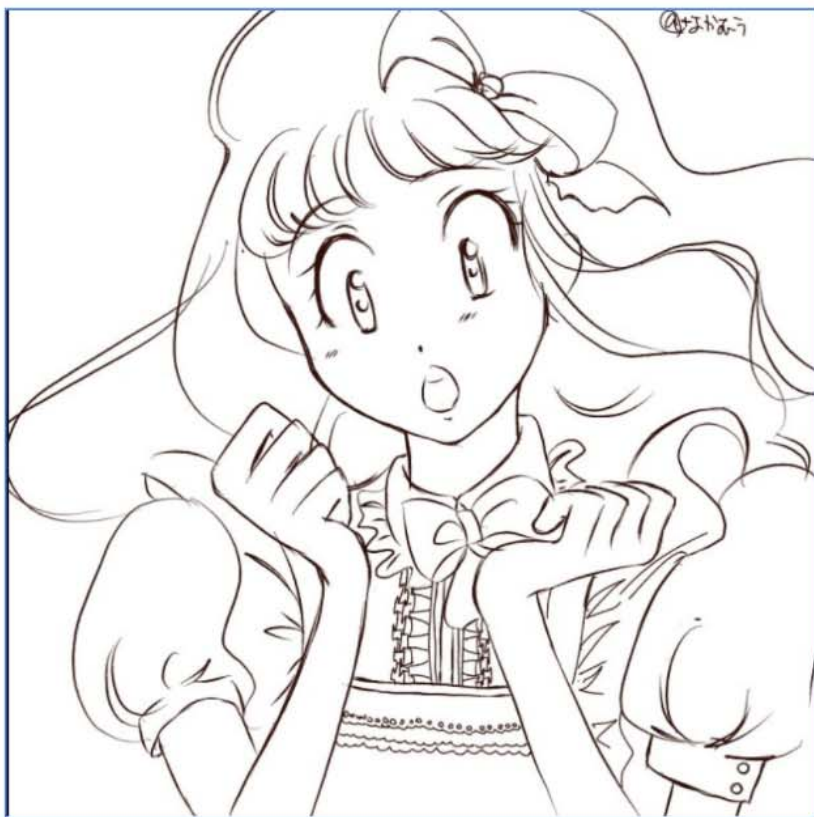




# Applications



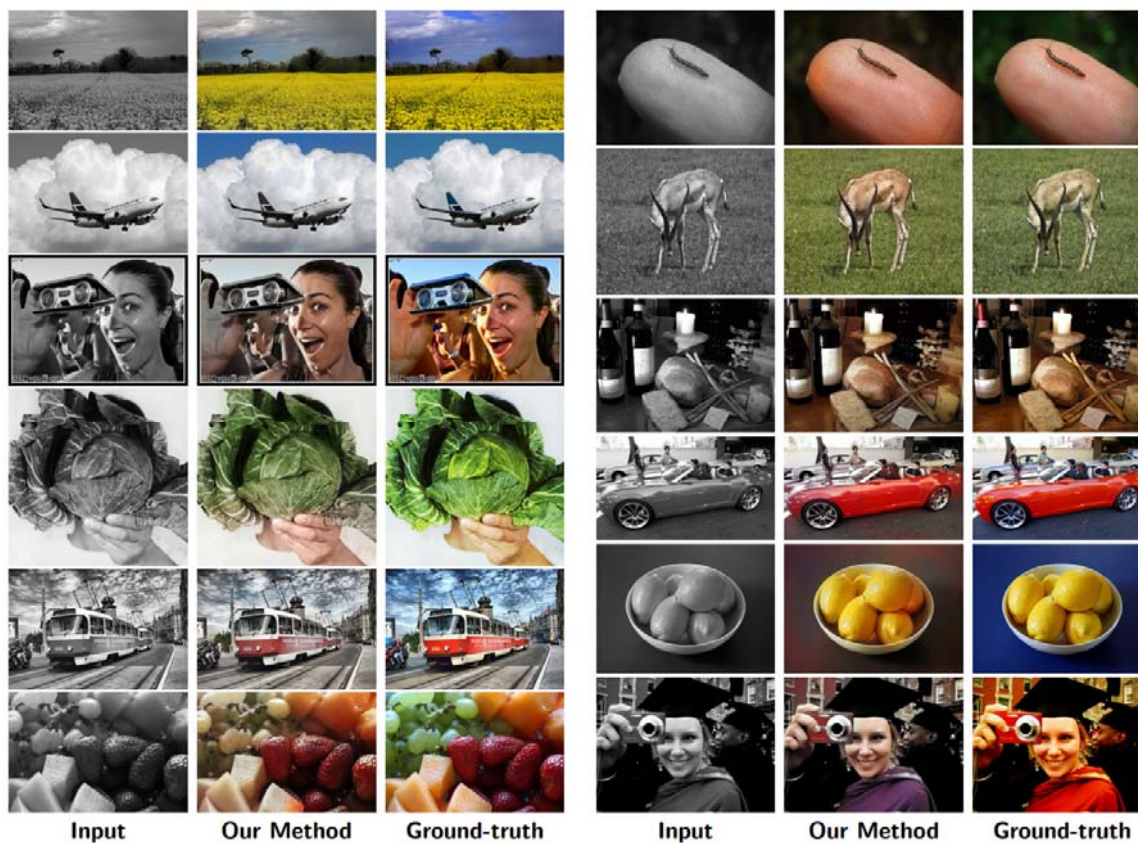
# 그림 자동 색칠하기



#중학생을위한딤러닝



# Colorization : 흑백 → Color



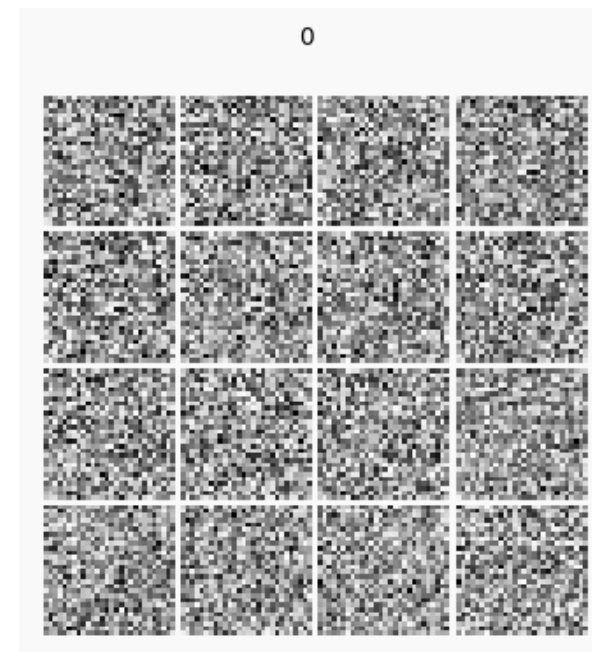
Larsson, Gustav, Michael Maire, and Gregory Shakhnarovich. "Learning representations for automatic colorization." *European Conference on Computer Vision*. Springer International Publishing, 2016.



# 그림 자동 완성



<https://www.autodraw.com/>  
[#중학생을위한딥러닝](#)

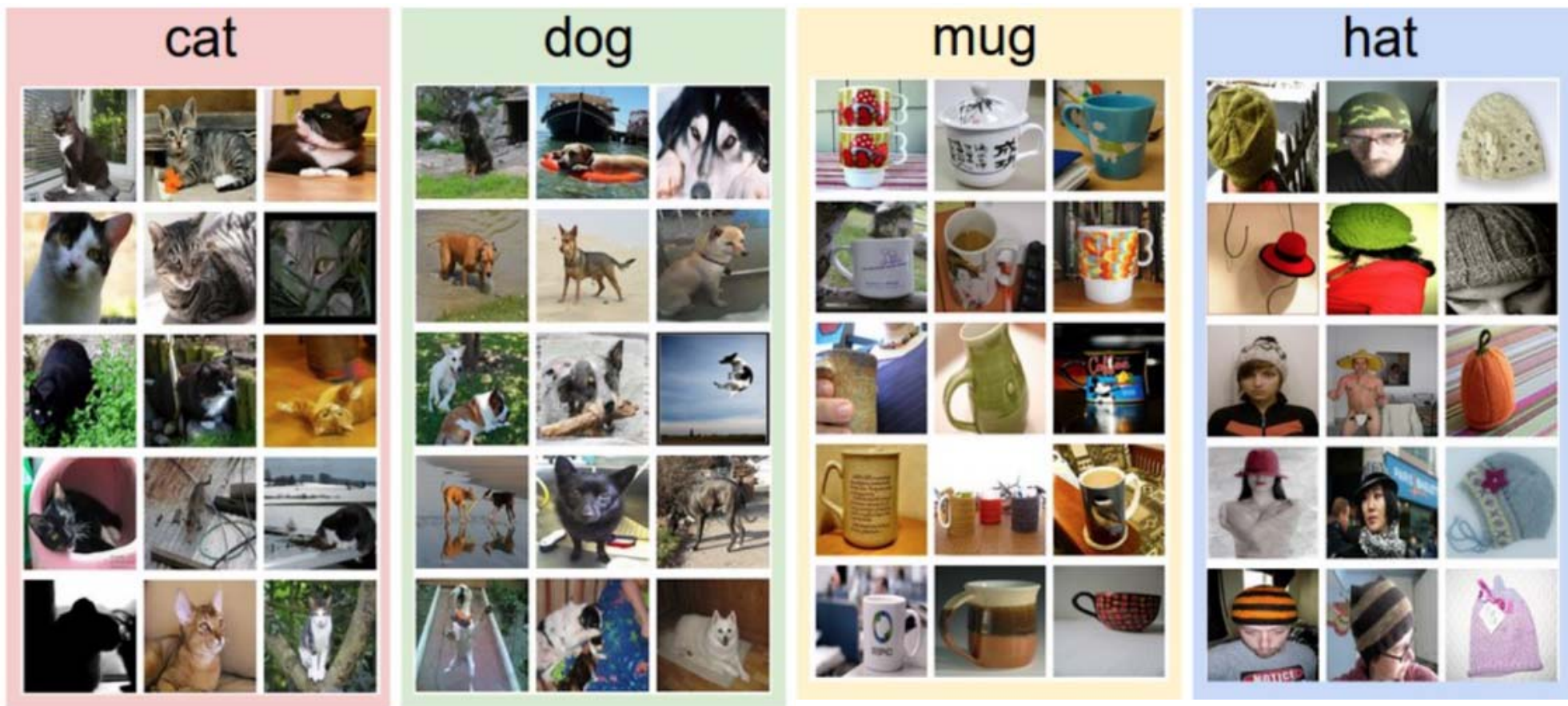


Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.





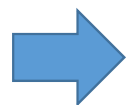
# 그림 의미 인식하기



<http://cs231n.github.io/classification/> #중국어의어휘인식



# 강아지 종류 맞추기 !



('Image', 1)

('Great Pyrenees', 0.26985887)

('kuvasz', 0.1741506)

('borzoi, Russian wolfhound', 0.091033742)

('clumber, clumber spaniel', 0.054091215)

('Maltese dog, Maltese terrier, Maltese', 0.042495646)

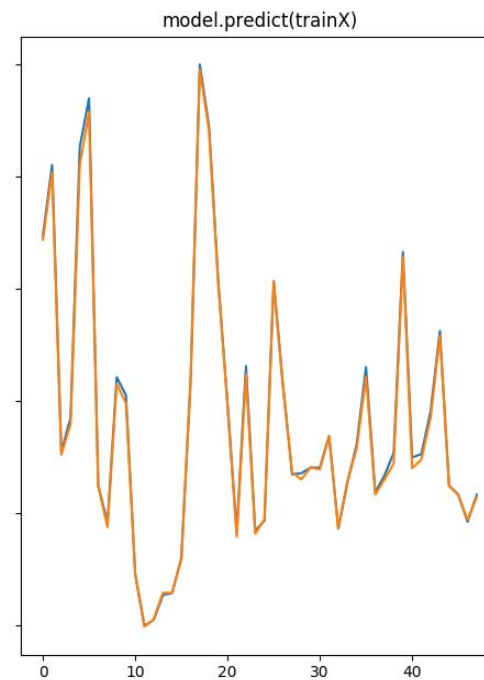
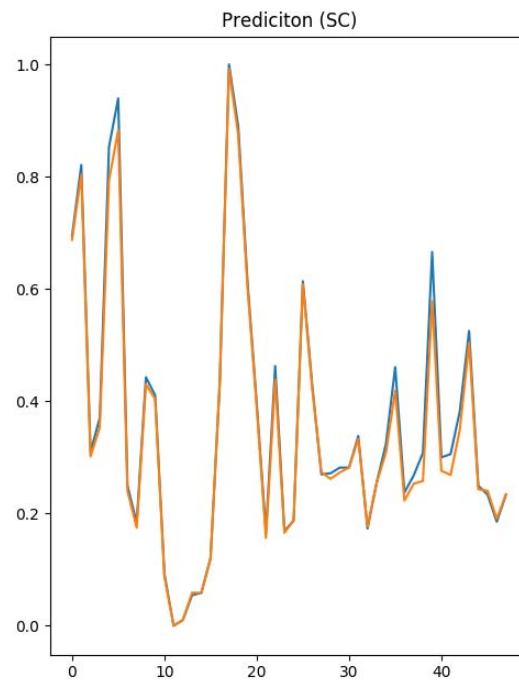


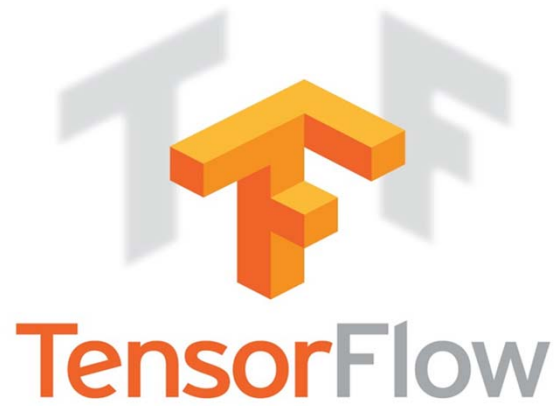
# Character Sequence RNN

```
(0, 'loss:', 2.9521115, 'Prediction:', 'eeeeee      a  aa a ')\n(1, 'loss:', 2.6848218, 'Prediction:', 'eeeeeeeeeee    aaabbbaaaaaaa')\n(2, 'loss:', 2.5180614, 'Prediction:', 'eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee')\n(3, 'loss:', 3.3403704, 'Prediction:', 'eeeeeeeeeeeeeee    ')\n(4, 'loss:', 2.7485611, 'Prediction:', 'ttteeeeeeeee    ')\n(5, 'loss:', 2.6026878, 'Prediction:', 'tttttttllll      a  a a ')\n(6, 'loss:', 2.5338719, 'Prediction:', 'tttttttaa aa      aa aaaa ')\n(7, 'loss:', 2.4470143, 'Prediction:', 'ttttttt aa        a ')\n(8, 'loss:', 2.3371212, 'Prediction:', 'ttttttt      a  a ')\n(9, 'loss:', 2.2038879, 'Prediction:', 'ttttttt tt  fy ya a ')\n(10, 'loss:', 2.0571938, 'Prediction:', 'ttttttt  yf y ya t ba !')\n(11, 'loss:', 1.8869145, 'Prediction:', 'ttttttt e  yf y ya t ba e !')\n(12, 'loss:', 1.6981899, 'Prediction:', 'Gtththtt ea  yf you yant bage !')\n(13, 'loss:', 1.4863166, 'Prediction:', 'Gtththtr ear yf you yant bage !')\n(14, 'loss:', 1.288029, 'Prediction:', 'Get t er  ear y yf you yant bage s!')\n(15, 'loss:', 1.0916867, 'Prediction:', 'Get t err early yf you want bagels!')\n(16, 'loss:', 0.91426879, 'Prediction:', 'Get e errrearly yf you want bagely!')\n(17, 'loss:', 0.76181585, 'Prediction:', 'Get e err early yf you want bagely!')\n(18, 'loss:', 0.62807894, 'Prediction:', 'Get e ere early if you want bagels!')\n...\n(48, 'loss:', 0.0033647721, 'Prediction:', 'Get there early if you want bagels!')\n(49, 'loss:', 0.0031009316, 'Prediction:', 'Get there early if you want bagels!')
```



# Time series data 학습 !





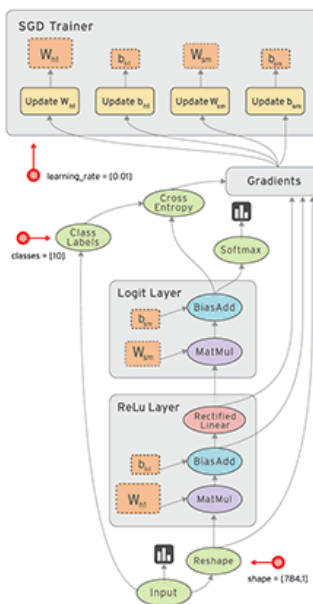
# Tensorflow

설치 및 익숙해지기



# Tensorflow

- 기계 학습과 딥러닝을 위해 구글에서 만든 오픈소스 라이브러리
- 데이터 플로우 그래프 (Data flow graph) 방식을 사용





## Deep learning libraries: Accumulated GitHub metrics

| Aggregate popularity (30•contrib + 10•issues + 5•forks)•1e-3 |        |                               |
|--|--------|-------------------------------|
| #1:  | 172.29 | tensorflow/tensorflow         |
| #2:  | 89.78  | BVLC/caffe                    |
| #3:  | 69.70  | fchollet/keras                |
| #4:  | 53.09  | dmlc/mxnet                    |
| #5:  | 38.23  | Theano/Theano                 |
| #6:  | 29.86  | deeplearning4j/deeplearning4j |
| #7:  | 27.99  | Microsoft/CNTK                |
| #8:  | 17.36  | torch/torch7                  |
| #9:  | 14.43  | baidu/paddle                  |
| #10:   | 13.10  | pfnet/chainer                 |
| #11:   | 12.37  | NVIDIA/DIGITS                 |
| #12:   | 10.42  | tflearn/tflearn               |
| #13:   | 9.20   | pytorch/pytorch               |

<https://twitter.com/fchollet/status/830499993450450944/>



# 설치하기

- MAC / Linux

<https://github.com/dalek7/DLWorkshop17Summer/blob/master/installation-mac.md>

- Windows

<https://github.com/dalek7/DLWorkshop17Summer/blob/master/installation-windows.md>



# 설치가 완료되면

- <https://github.com/dalek7/DLWorkshop17Summer>



# 모든 것은 Tensor로 구성되어 있음 !

```
In [2]: t = tf.constant(3) # a rank 0 tensor; this is a scalar with shape []  
print(sess.run(t))  
3
```

```
In [3]: t = tf.constant([1., 2., 3.]) # a rank 1 tensor; this is a vector with shape [3]  
print(sess.run(t))  
[ 1.  2.  3.]
```

```
In [4]: t = tf.constant([[1., 2., 3.], [4., 5., 6.]]) # a rank 2 tensor; a matrix with shape [2,3]  
print(sess.run(t))  
[[ 1.  2.  3.]  
 [ 4.  5.  6.]]
```

```
In [5]: t = tf.constant([[[1., 2., 3.]], [[7., 8., 9.]]) # a rank 3 tensor with shape [2,1,3]  
print(sess.run(t))  
[[[ 1.  2.  3.]]  
 [[ 7.  8.  9.]]]
```



# Tensor Ranks, Shapes, and Types

| Rank | Math entity                      | Python example  |
|------|----------------------------------|---|
| 0    | Scalar (magnitude only)          | <code>s = 483</code>  |
| 1    | Vector (magnitude and direction) | <code>v = [1.1, 2.2, 3.3]</code>  |
| 2    | Matrix (table of numbers)        | <code>m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]</code>                        |
| 3    | 3-Tensor (cube of numbers)       | <code>t = [[[2], [4], [6]], [[8], [10], [12]], [[14], [16], [18]]]</code> |
| n    | n-Tensor (you get the idea)      | <code>....</code>   |

[https://www.tensorflow.org/programmers\\_guide/dims\\_types](https://www.tensorflow.org/programmers_guide/dims_types)





# 다음시간에는...

한 개의 종속 변수(dependent variable)와 설명 변수들(explanatory variables)과의 관계를 모델링.

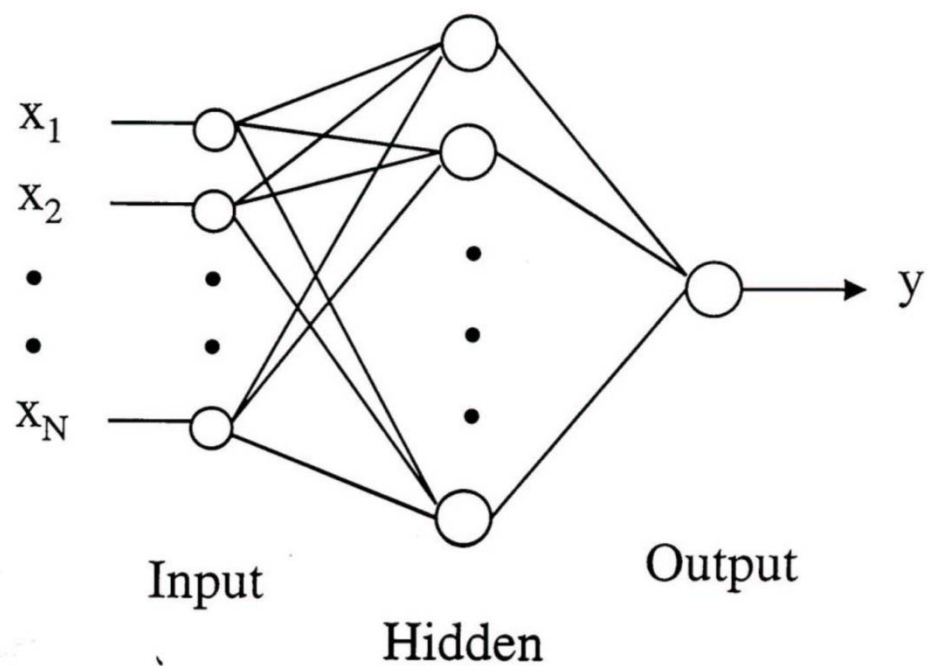
독립 변수

종속 변수

| x (hours) | y (score) |
|-----------|-----------|
| 10        | 90        |
| 9         | 80        |
| 3         | 50        |
| 2         | 30        |



다음시간에는...

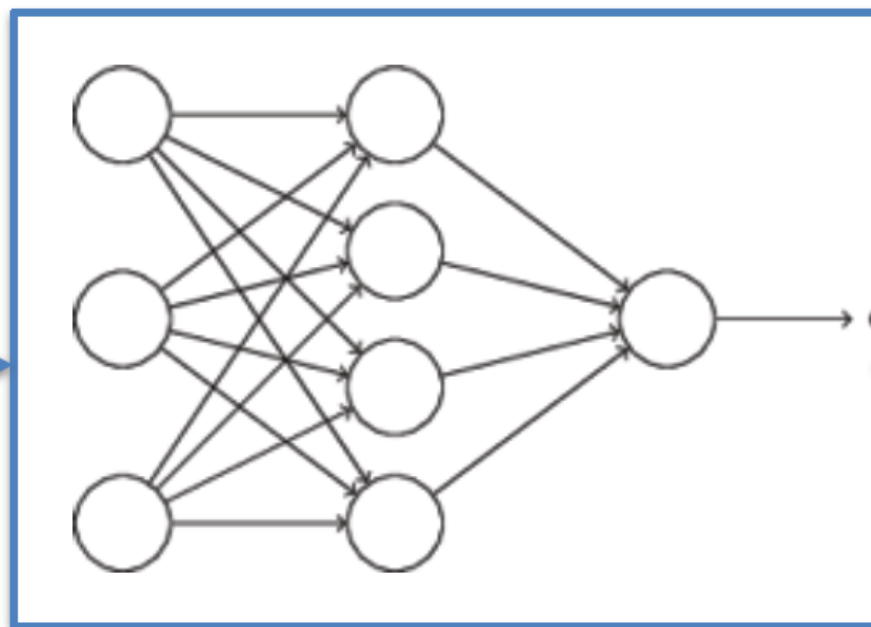




# Image



[201, 185, 125, ..., 233]

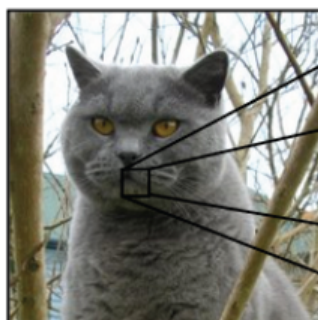


Classes:  
82% cat  
15% dog  
2% hat  
1% mug



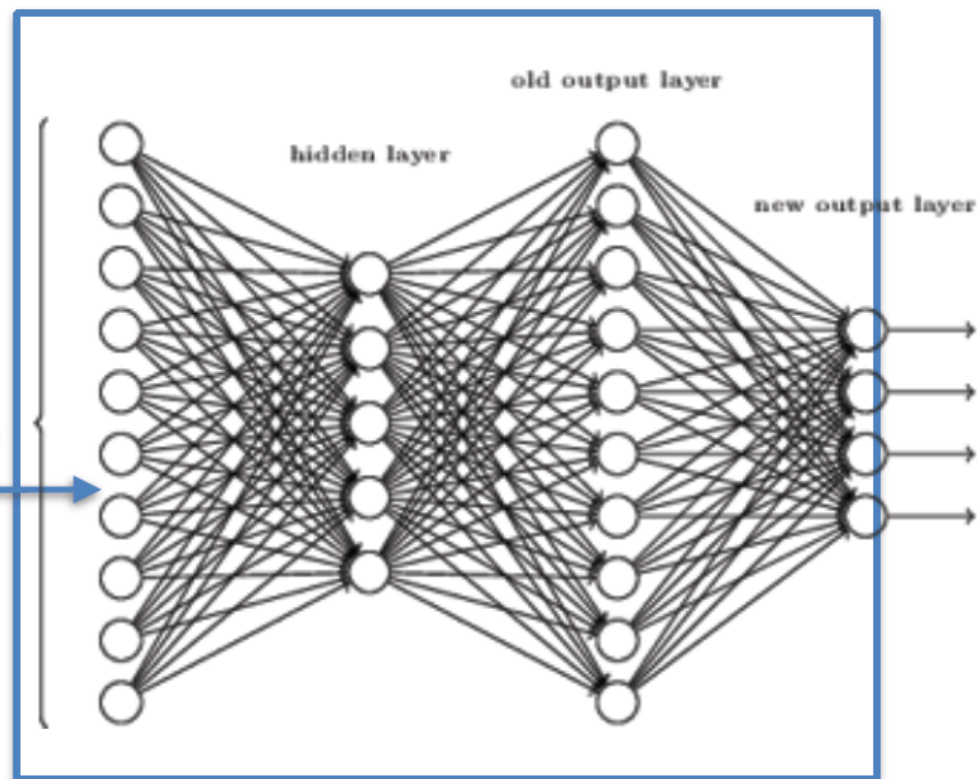
# DEEP NEURAL NETWORKS

Image



[201, 185, 125, ..., 233]

input layer  
(784 neurons)



Classes:  
82% cat  
15% dog  
2% hat  
1% mug



# 문과 계열 학생을 위한 Guide

- 행렬

- [http://cs.sungshin.ac.kr/~dkim/Lectures/dm\\_2013\\_ch05.pdf](http://cs.sungshin.ac.kr/~dkim/Lectures/dm_2013_ch05.pdf)
- <http://www.sfu.ca/~wainwrig/Econ331/331-notes-matrix.pdf>



# All my great stuff, I learned outside of school

- 모두를 위한 머신러닝/딥러닝 강의  
<https://github.com/hunkim/DeepLearningZeroToAll>
- Reddit : machine learning  
<https://www.reddit.com/r/MachineLearning/>
- 머신러닝 속 수학  
<https://mingrammer.com/translation-the-mathematics-of-machine-learning>
- 수학을 포기한 직업 프로그래머가 머신러닝 학습을 시작하기위한 학습법 소개  
<http://qiita.com/daxanya1/items/218f2e3b922142550ef9>

Seung-han

Seung-Chan

Jeung-Chan

감사합니다.



정답 :

