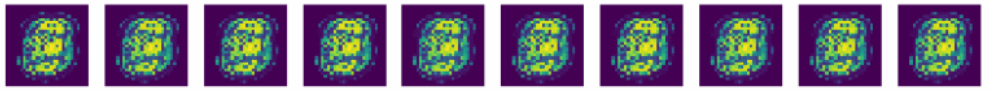
한림대학교 Summer Deep Learning Workshop

Neural Network

July 11, 2017

Seung-Chan Kim, Ph. D



- 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)

https://github.com/dalek7/DLWorkshop17Summer

복습

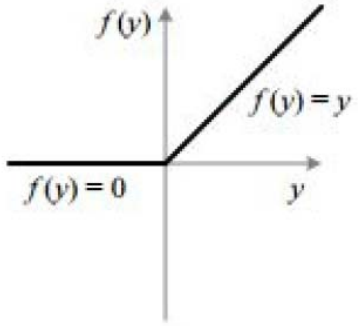
- Optimization
- Regression
- Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU)

$a_1 = \text{relu}(a_1)$

$b_1 = \text{relu}(b_1)$

$c_1 = \text{relu}(c_1)$

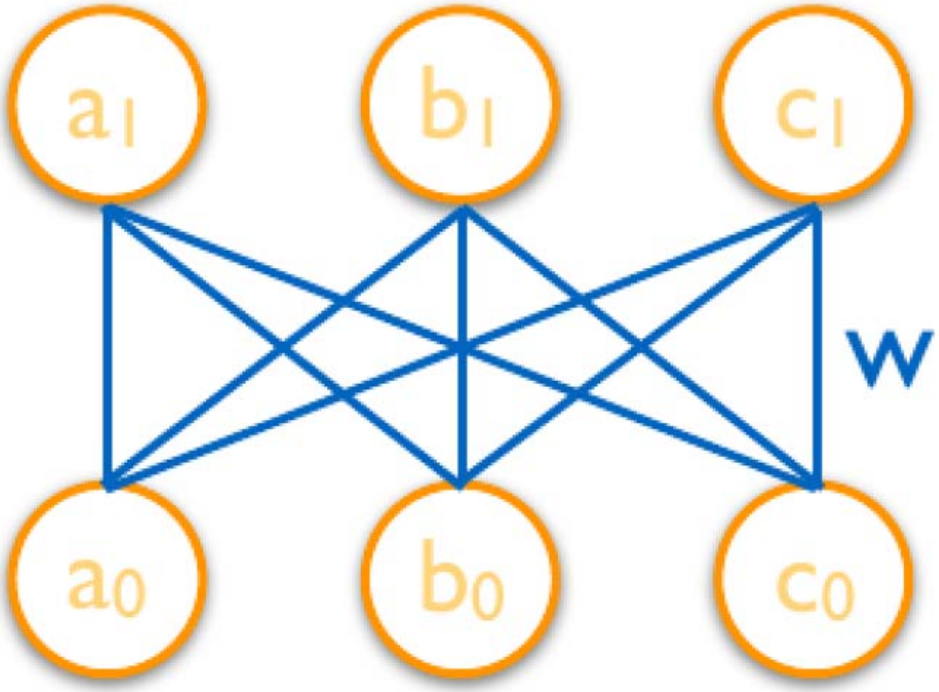


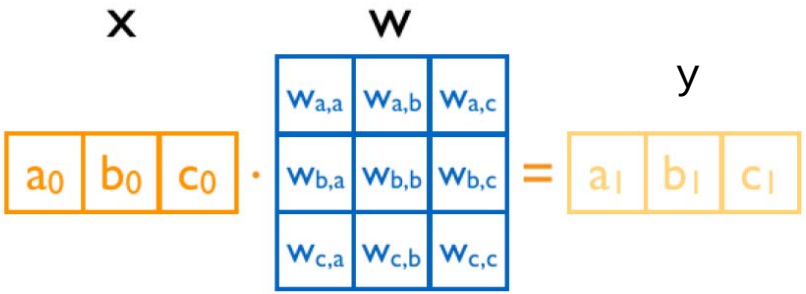

```
out = tf.nn.relu(y)
```

```
(-5, '-->',     )
(-4, '-->',     )
(-3, '-->',     )
(-2, '-->',     )
(-1, '-->',   ?  )
( 0, '-->',     )
( 1, '-->',     )
( 2, '-->',     )
( 3, '-->',     )
( 4, '-->',     )
```

00-5-relutest.py 를 열어주세요

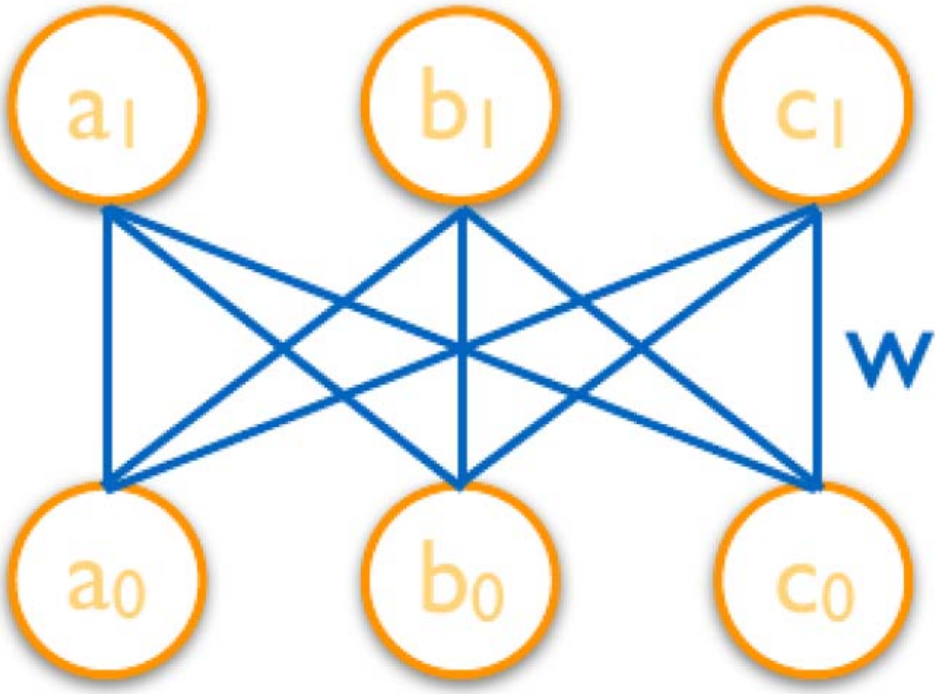
A simple Rectified Linear Unit (ReLU) network

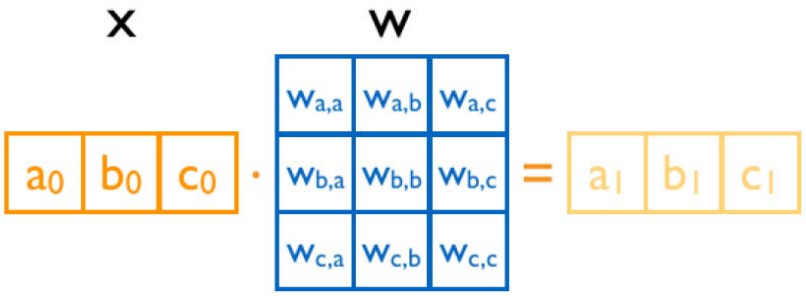




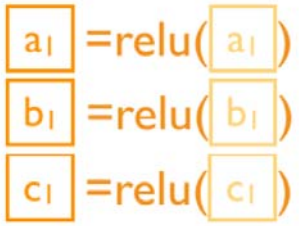

```
y = tf.matmul(x,w)
```

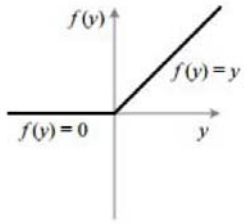
A simple Rectified Linear Unit (ReLU) network





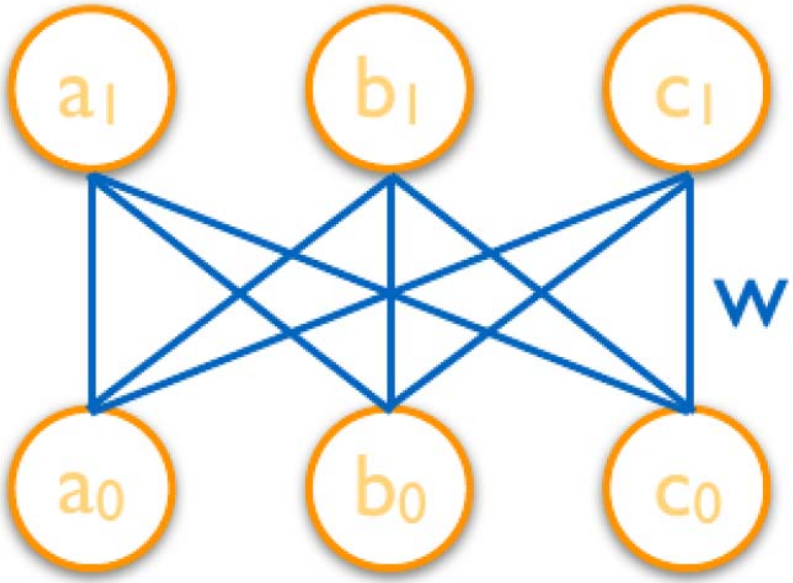

```
y = tf.matmul(x,w)
```






```
out = tf.nn.relu(y)
```

A simple Rectified Linear Unit (ReLU) network




```
import tensorflow as tf
sess = tf.Session()
x = tf.placeholder("float", [1, 3])
w = tf.Variable(tf.random_normal([3, 3]), name='w')
y = tf.matmul(x, w)
relu_out = tf.nn.relu(y)
```

01-3-simplenetwork.py 를 열어주세요

Why Neural Networks?

- Consider humans
  - Neuron switching time ~ .001 second
  - Number of neurons ~ 10^{10}
  - Connections per neuron ~ $10^{4\sim5}$
  - Scene recognition time ~ .1 second
  - 100 inference steps doesn't seem like enough
    $\Rightarrow$ massively parallel computation
- Properties of artificial neural nets (ANN)
  - Many neuron-like threshold switching
  - Many weighted interconnections among units
  - Highly parallel, distributed process
  - Emphasis on tuning weights automatically
- Other names: connectionism, parallel distributed processing, neural computation

When to Consider Neural Networks?

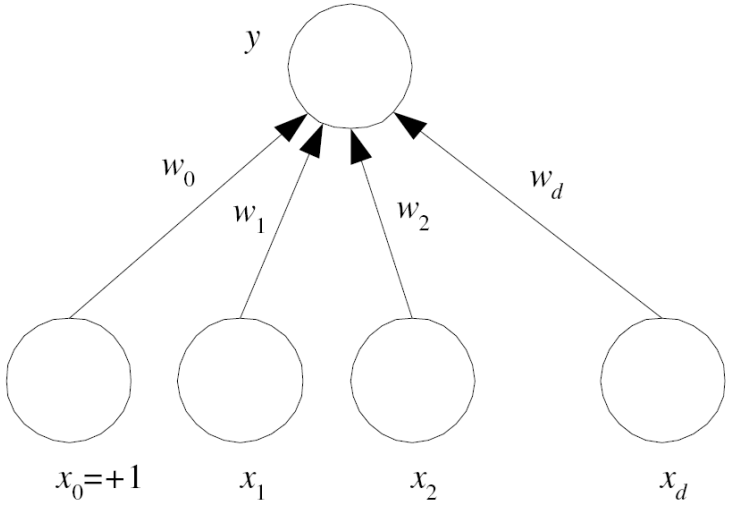
- Input is high-dimensional discrete or real-valued (e.g., raw sensor input)
- Output is discrete or real-valued
- Output is a vector of values
- Possibly noisy data
- Form of target function is unknown
- Human readability of result is unimportant

- Examples
  - Speech phoneme recognition
  - Image classification
  - Financial prediction

Single-Layer Perceptron

- Classification
  - Sigmoid activation function

$$y = \text{sigmoid}(in) = \frac{1}{1 + \exp[-\mathbf{w}^T \mathbf{x}]}$$

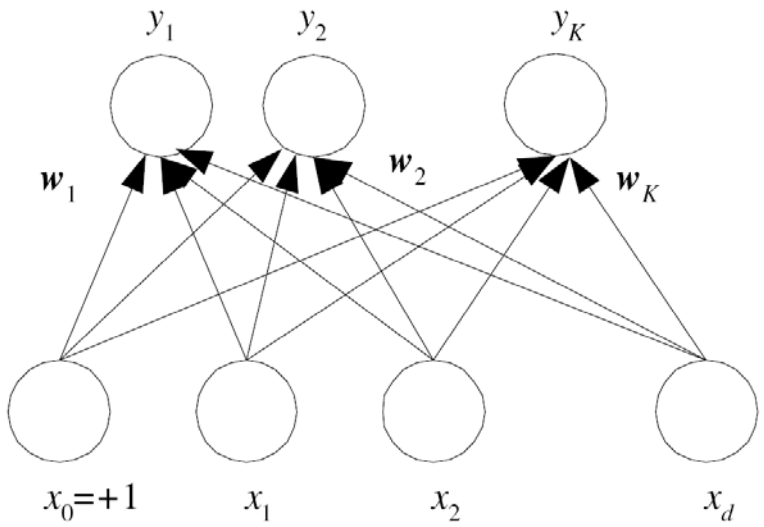


Single-Layer Perceptron with K Outputs

- Classification
  - Sigmoid activation function

$$y_i = \text{sigmoid}(in_i) = \frac{1}{1 + \exp[-\mathbf{w}_i^T \mathbf{x}]}$$

$$\text{choose } C_i \text{ if } y_i = \max_k y_k$$

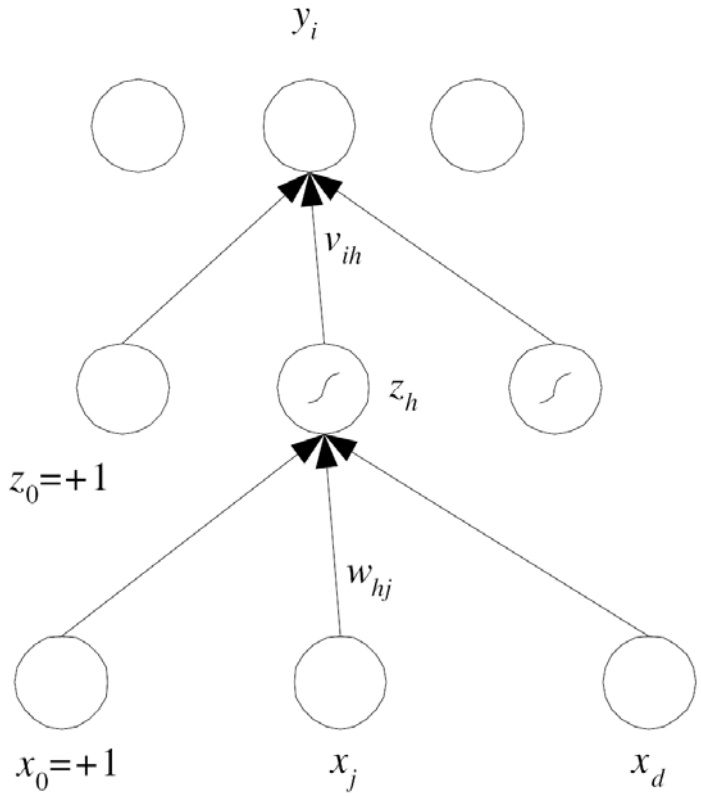


Multi-layer Perceptrons

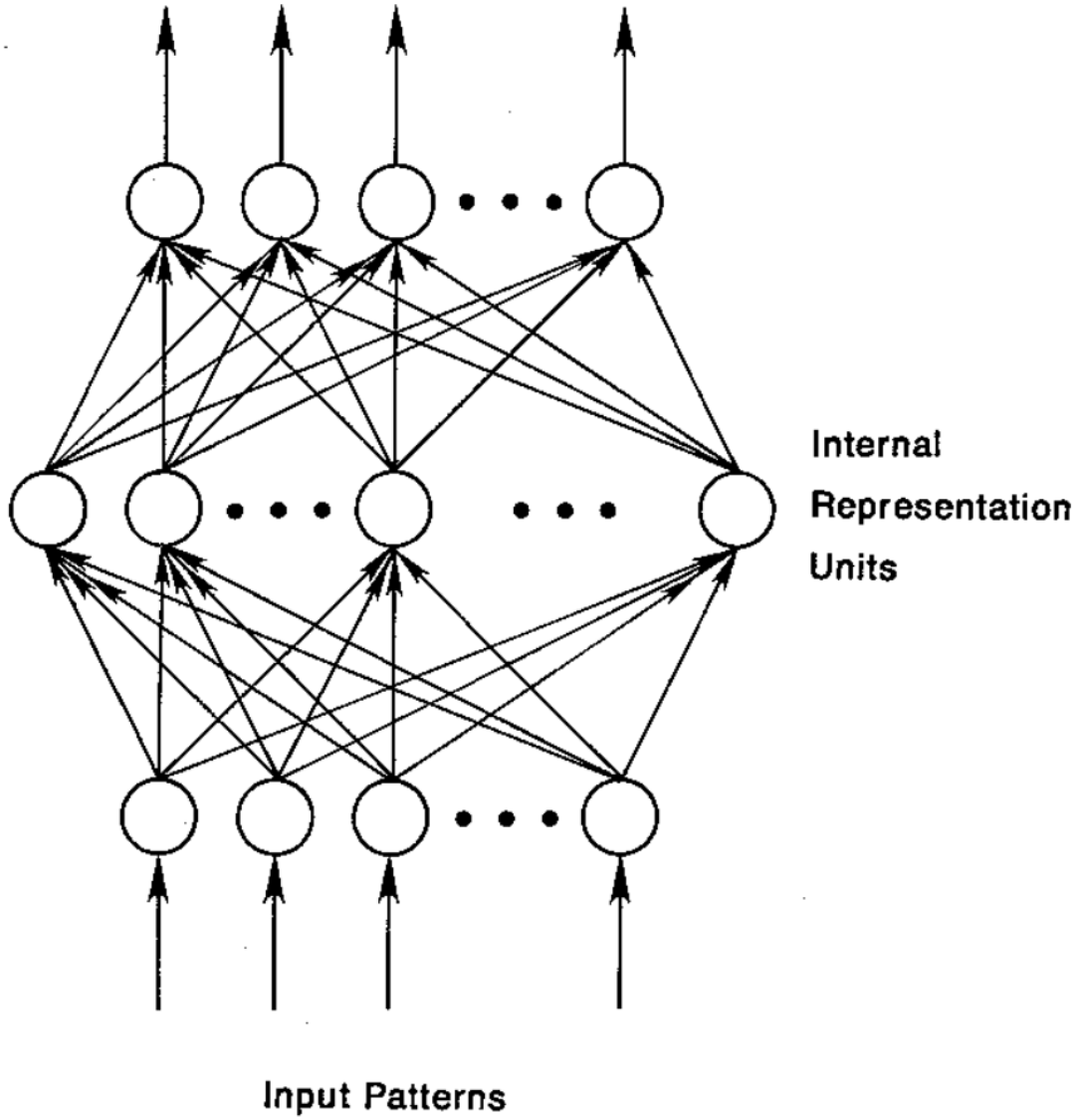
- [Rumelhart et al., 1986]

$$y_i = \mathbf{w}_i^T \mathbf{z} = \sum_{h \in H} w_{hi} z_h + w_{0i}$$

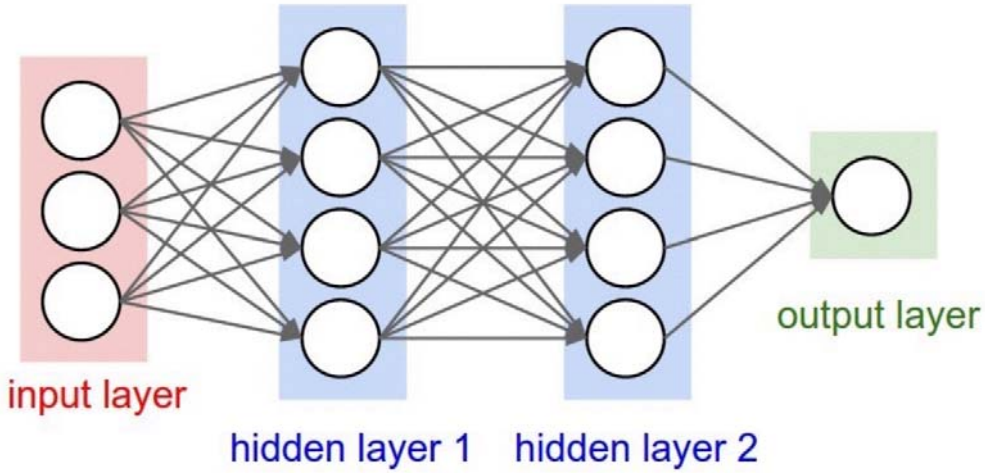
$$z_h = a_h(\mathbf{w}_h^T \mathbf{x}) = \frac{1}{1 + \exp[-(\sum_{j=1}^{d} w_{jh} x_j + w_{0h})]}$$



D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, 10.1038/323533a0 vol. 323, no. 6088, pp. 533-536, 10/09/print 1986.


Output Patterns
Internal
Representation
Units
Input Patterns

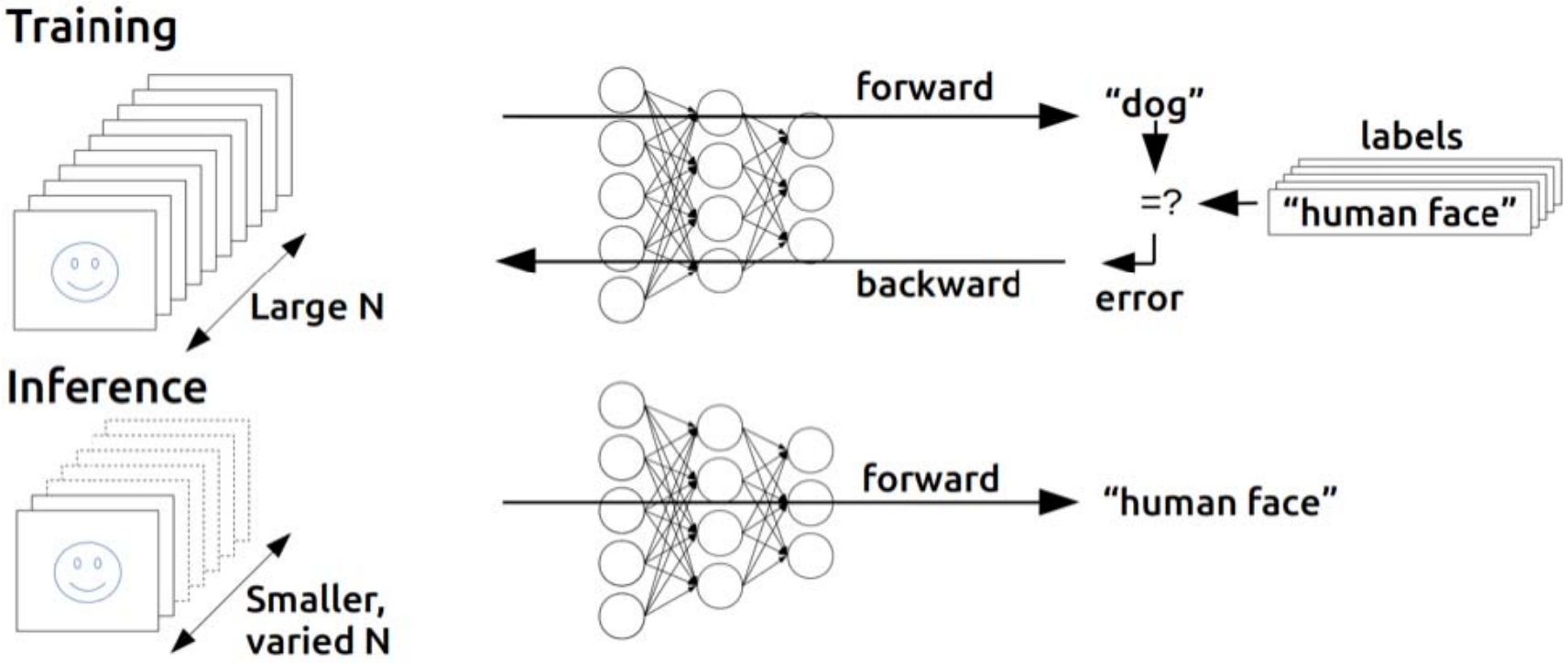
No one on earth had found a viable way to train !



*Marvin Minsky, 1969

http://cs231n.github.io/convolutional-networks/

Backpropagation
(1974, 1982 by Paul Werbos, 1986 by Hinton)



https://devblogs.nvidia.com/parallelforall/inference-next-step-gpu-accelerated-deep-learning/

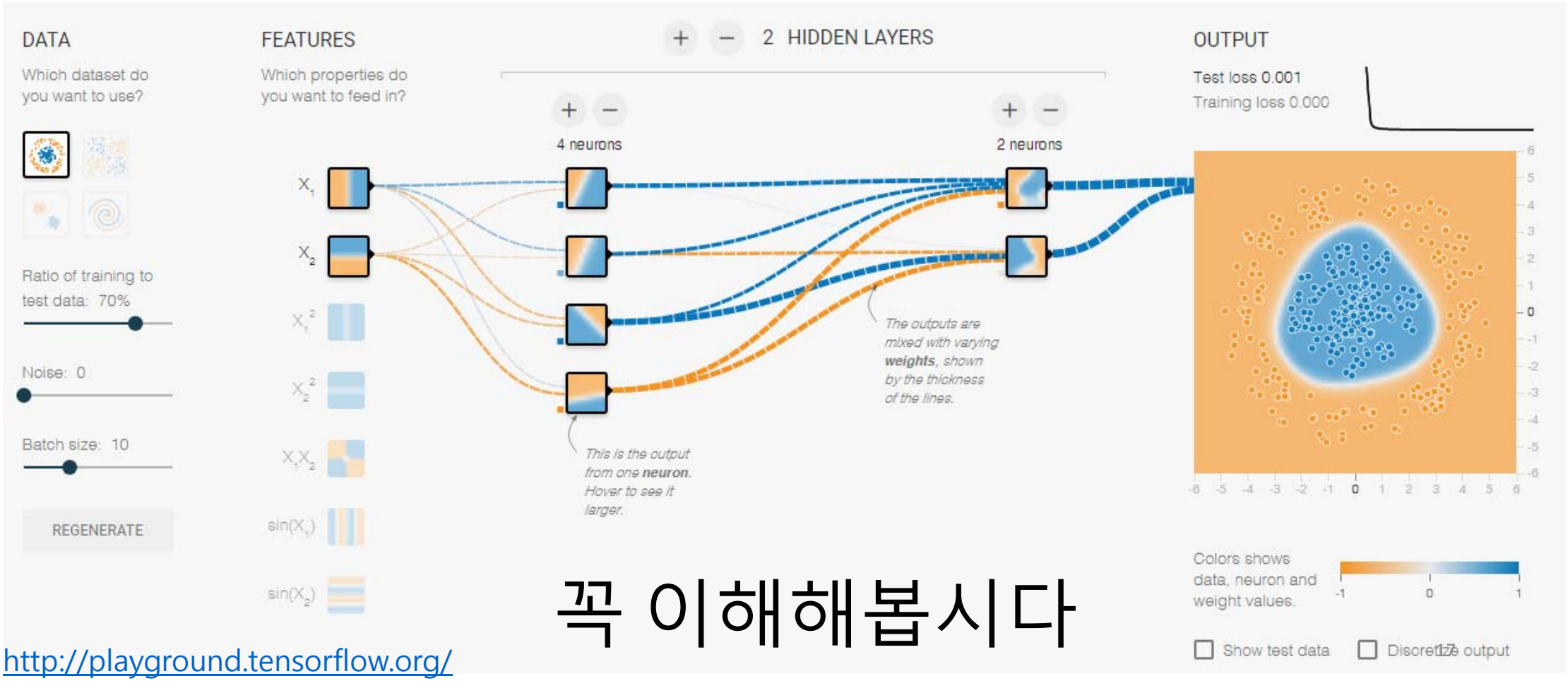
$$\text{cost} = \frac{1}{m}\sum_{i=1}^{m}(wx^{(i)} - y^{(i)})^2$$

- W는 어떻게 업데이트 하는가?
  - learning rate α에 cost function을 w에 대하여 미분한 값, 즉 기울기를 곱한 값을 빼는 것이다.

$$\begin{aligned} w := w - \alpha\frac{\partial}{\partial w}cost(w) &= w - \alpha\frac{\partial}{\partial w}\frac{1}{2m}\sum_{i=1}^{m}\left(wx^{(i)} - y^{(i)}\right)^2 \\ &= w - \alpha\frac{1}{2m}\sum_{i=1}^{m}2\left(wx^{(i)} - y^{(i)}\right)x^{(i)} \\ &= w - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(wx^{(i)} - y^{(i)}\right)x^{(i)} \end{aligned}$$

자세한 계산 방법은 2학기 "인공지능 개론" 수업에서 다룸

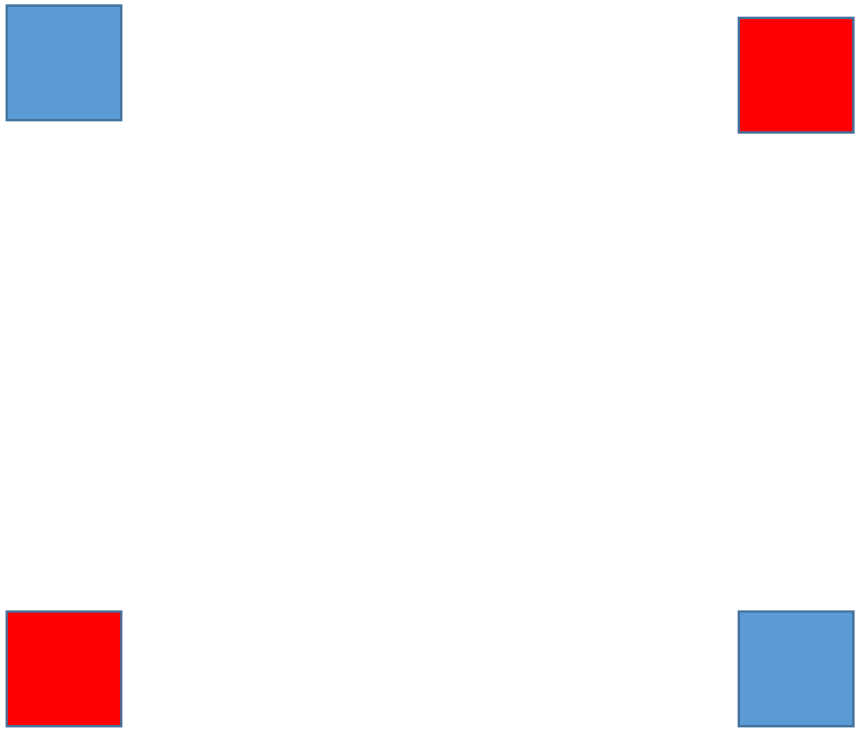
Neural Network : playground.tensorflow.org



꼭 이해해봅시다

http://playground.tensorflow.org/

Recap - XOR 문제 !!

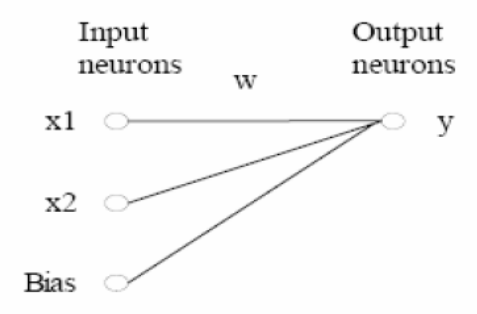


Input Patterns		Output Patterns
00	→	0
01	→	1
10	→	1
11	→	0

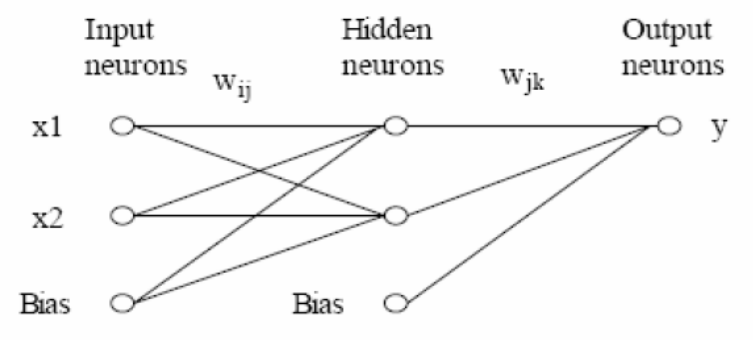
XOR classification using single and two layer neural networks

XOR classification using single and two layer neural networks

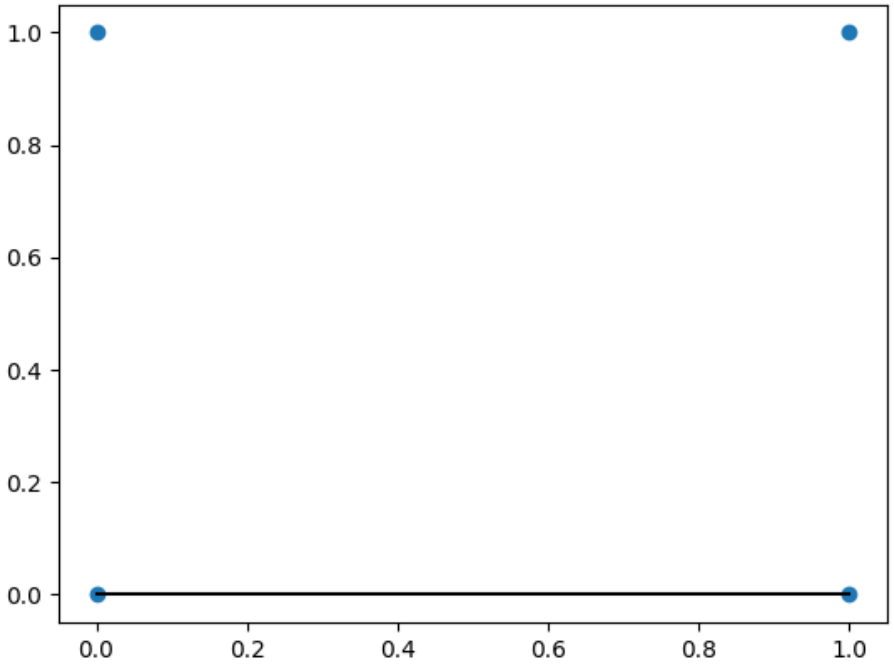
- **Single layer networks (ADALINE)**



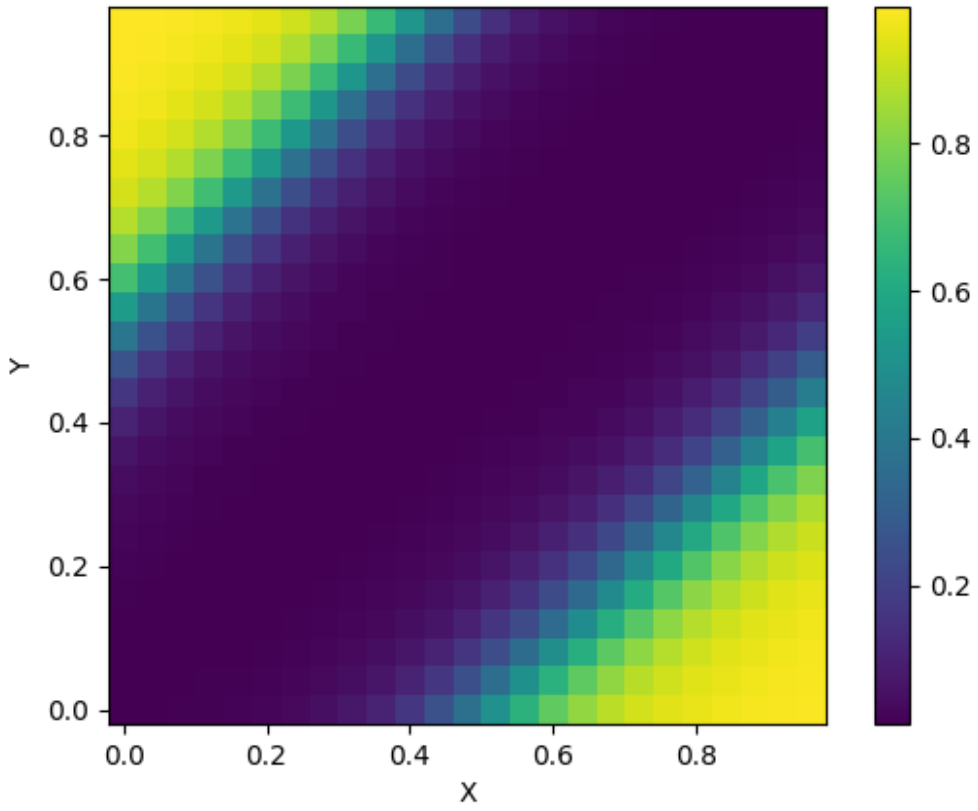
- **An example of two layer neural networks: 2-2-1 structure**



Input Patterns		Output Patterns
00	→	0
01	→	1
10	→	1
11	→	0



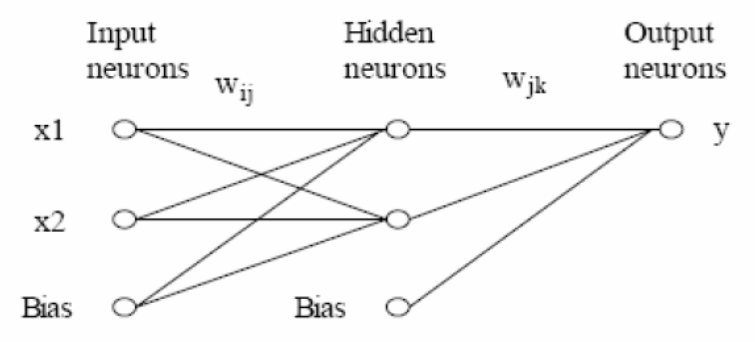
03-1-xor-simple.py 를 열어주세요




```
0 [[ 0.01338216]] [[ 0.]]
1 [[ 0.98166394]] [[ 1.]]
2 [[ 0.98809403]] [[ 1.]]
3 [[ 0.01135799]] [[ 0.]]
```

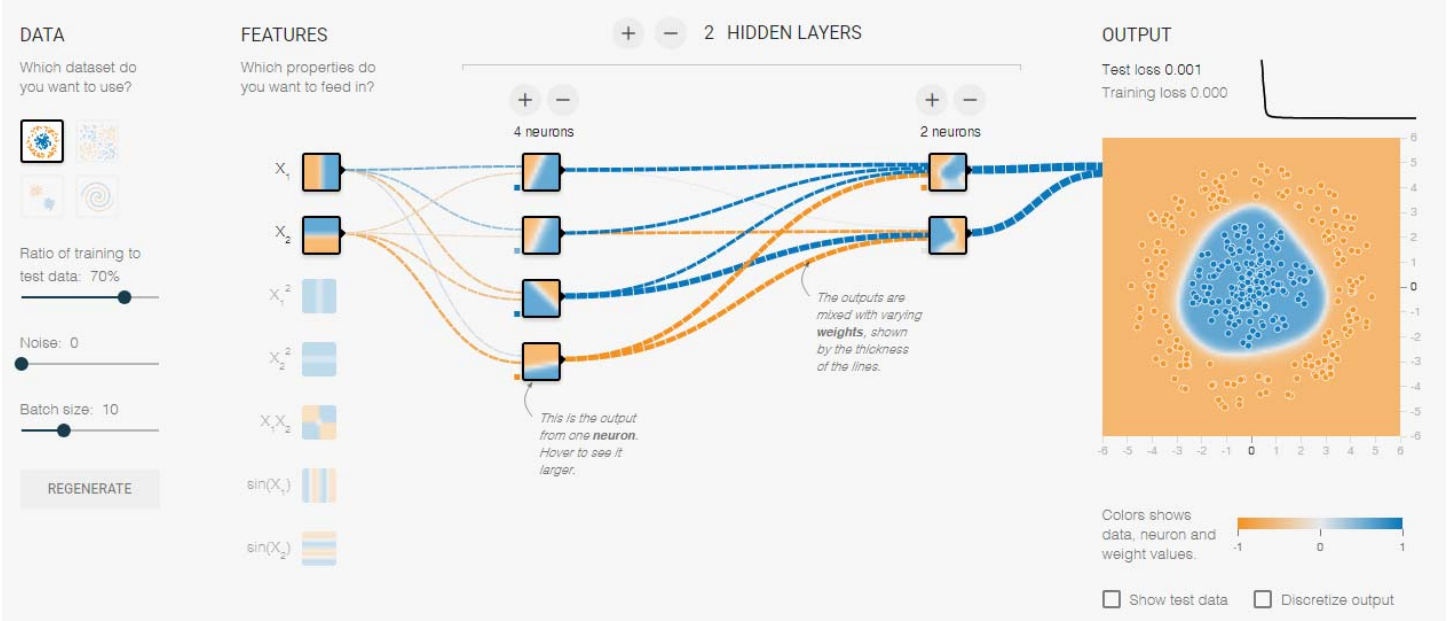
03-2-xor-layers.py 를 열어주세요

- **An example of two layer neural networks: 2-2-1 structure**



HW

- Neural Network (playground.tensorflow.org) 사이트에서 문제 하나를 고른후 layer 개수, hiden state 개수를 자유롭게 선택하여, 문제 해결



Acknowledgement

모두를 위한 머신러닝/딥러닝 강의

Seung-han

Seung-Chan

Jeung-Chan

감사합니다.