K-means Cluster

- Simple But Powerful!

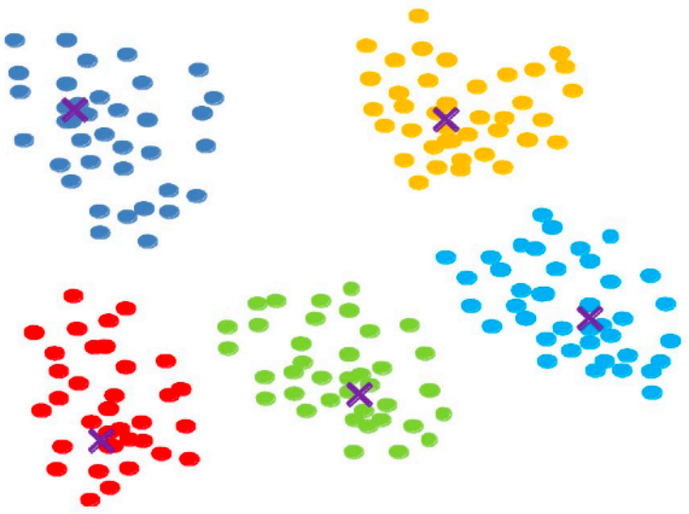
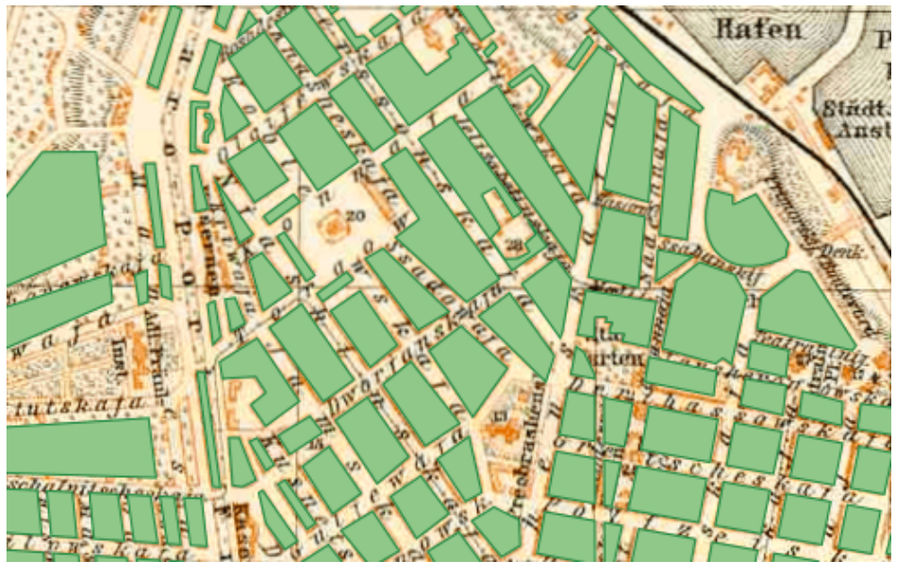


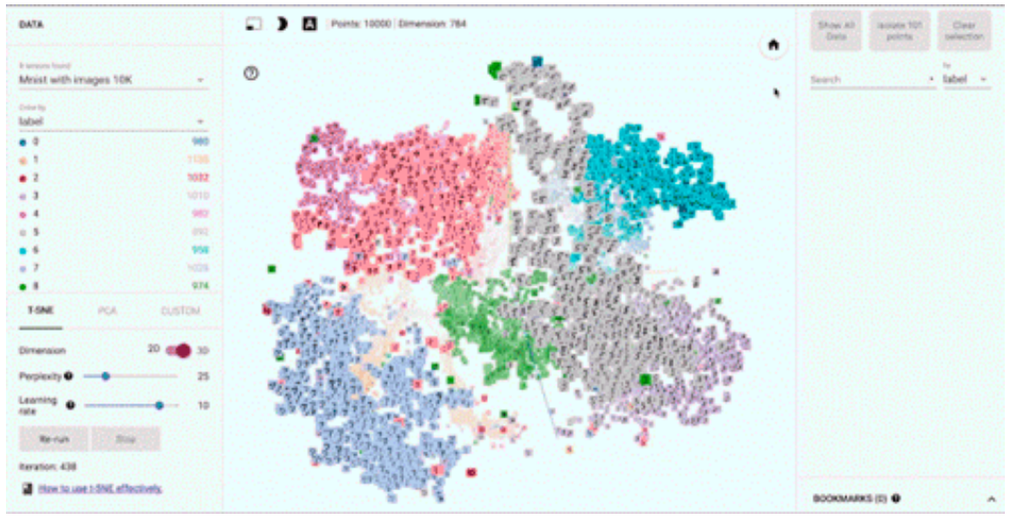
Fig. 13. Exemplary K-Means result

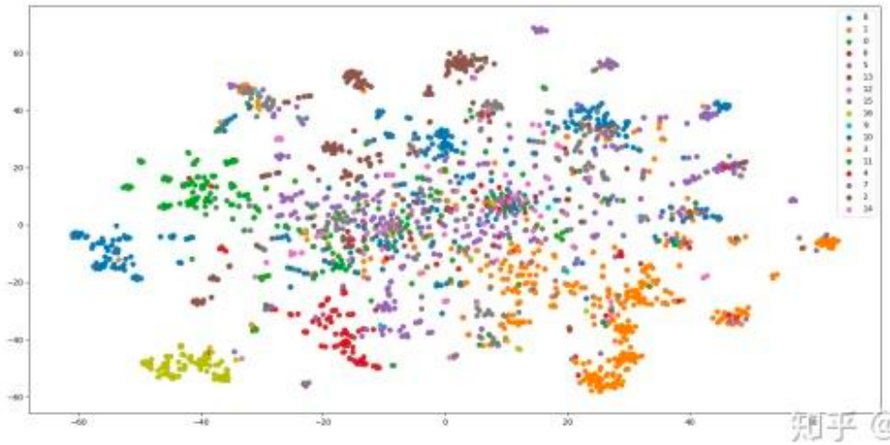


- An example of text cluster, from new corpus.

Embedding Cluster

- Embedding algorithm we will teach in course 11.





Semi Supervised & Activate Learning

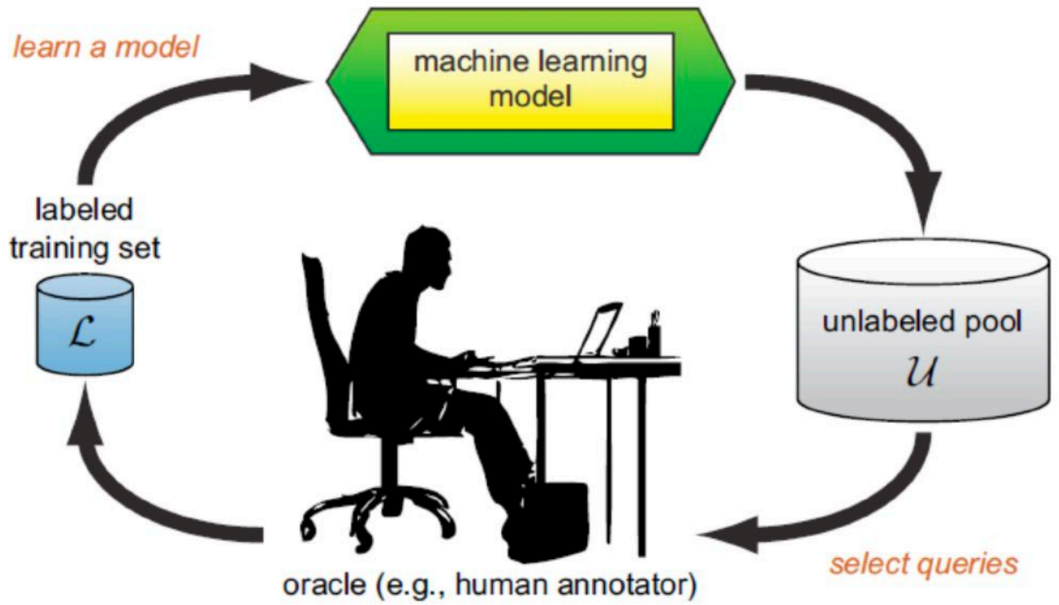
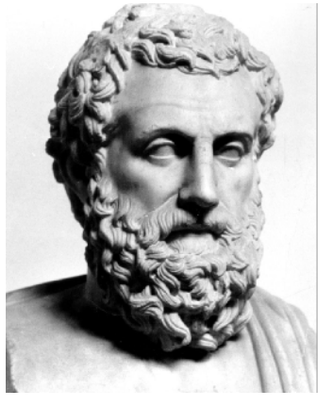


Figure 1: The pool-based active learning cycle.

Artificial Intelligence For NLP Lesson- 09

人工智能与自然语言处理课程组

2019.August. 31



Outline

Review

- What does Machine Learning Focus?
- Machine Learning steps.

Neural Networks

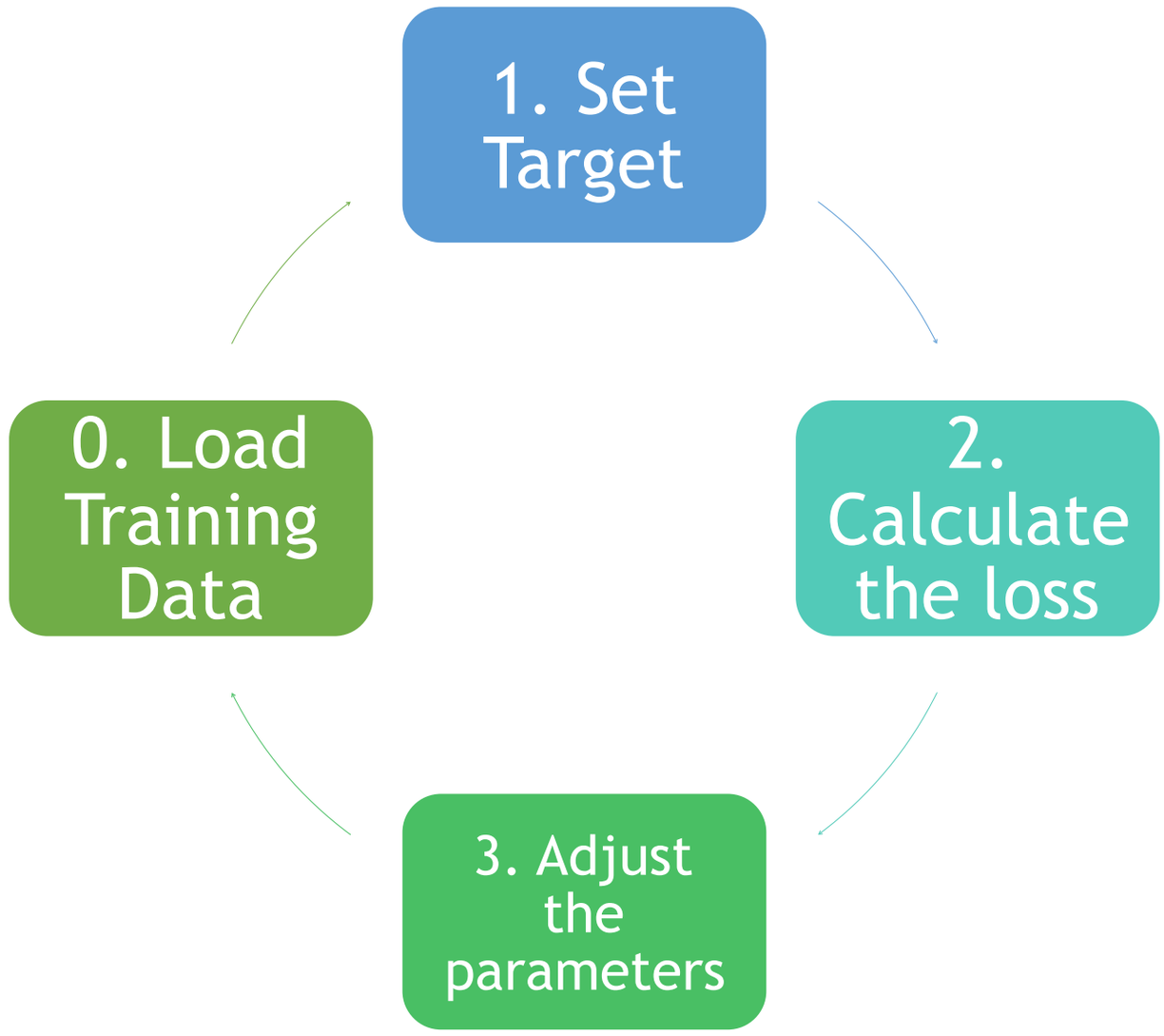
Loss, Backpropagation, Cross Entropy, (Stochastic) Gradient Descent

Optimizer, Activation Unit

How to implement a Neural Network

PyTorch, Tenorflow, Keras

What Machine Learning Concern ?



- Loss -> 0
- Predicate -> Ok
- Running Time -> Short
- Training Corpus -> Small
- Other (active learning, etc)

Our Target?

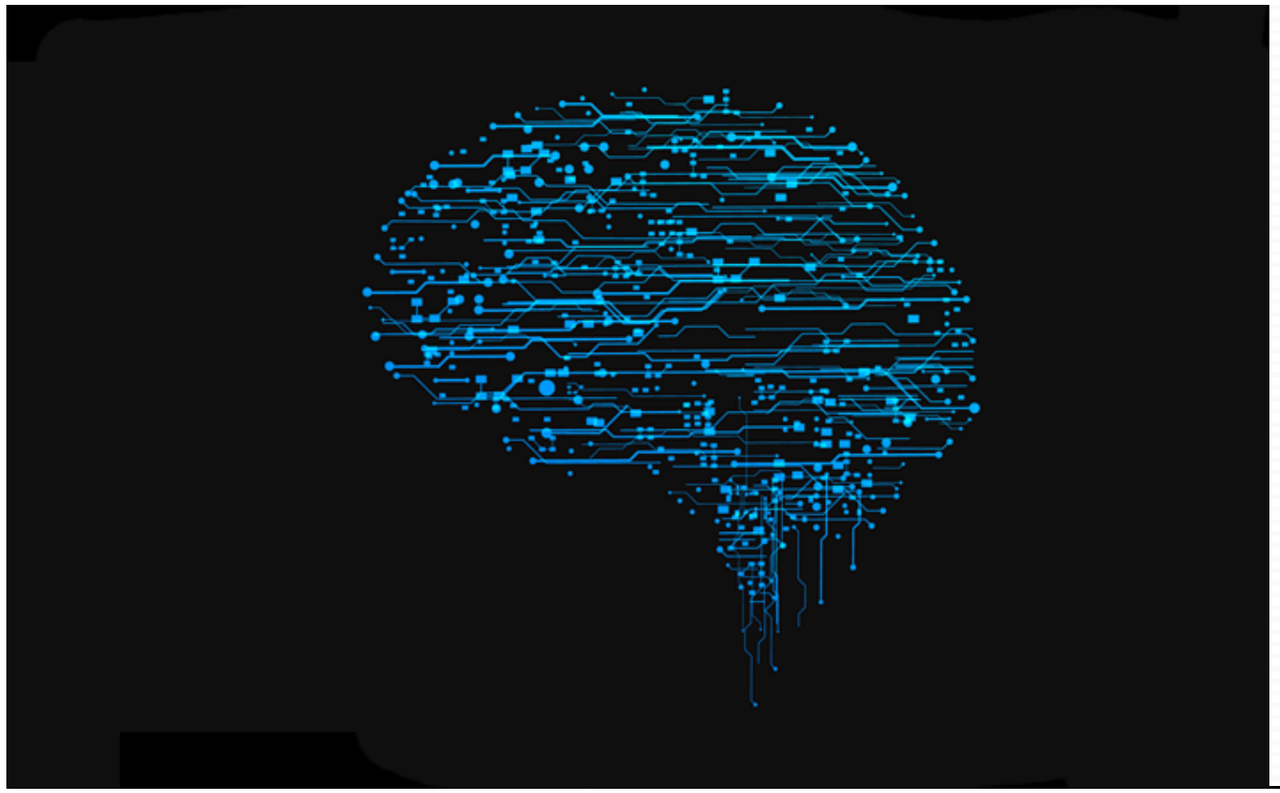
Classification

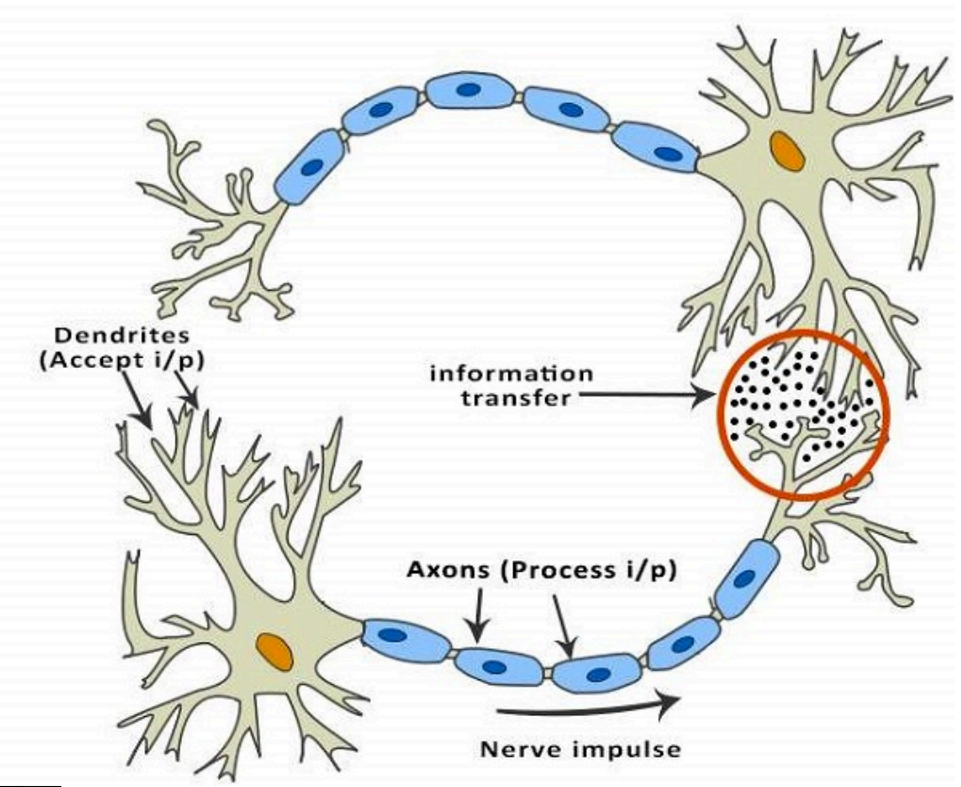
Regression

Reinforcement Learning

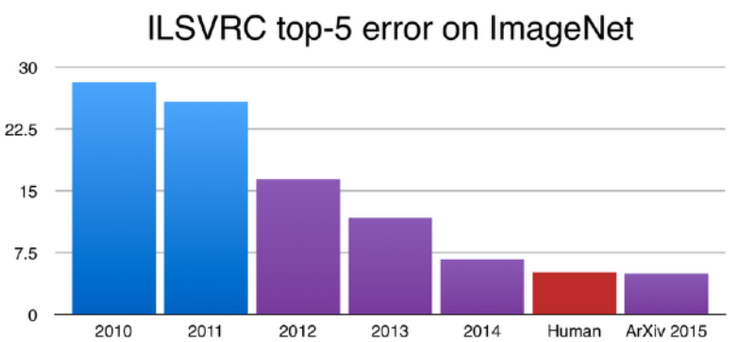
Cluster

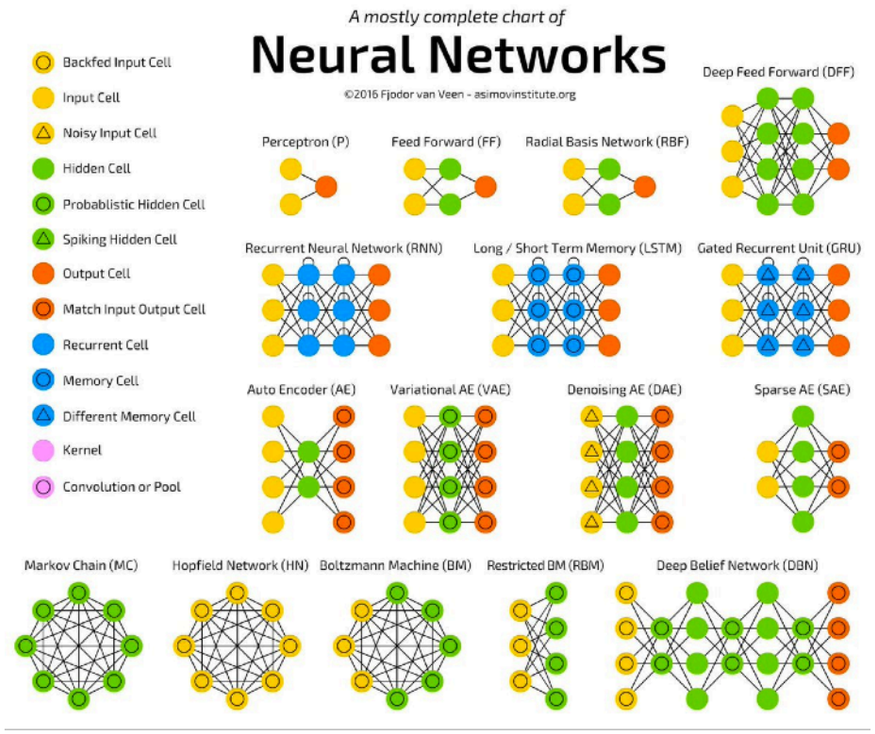
Generative Model





https://github.com/Artificial-Intelligence-for-NLP/References/blob/master/AI%20%26%20Machine%20Learning/backprop_hinton.pdf





The Rise of Neural Networks

Neural Networks

X: <x1, x2, … , xN>

Process X, get some Value M1

Proess M1, get some Value M2 ..

…..

Numeric : 0.13, 0.15 OR [0, 0, 1], [0, 1, 0]

$$f(x) = ax + b$$

$$f(x) = sigmoid(ax + b)$$

$$f(x) = ax^2 + bx + c$$

$$f(x) = sin(x) + cos(x)$$

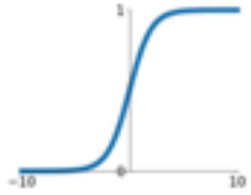
- Assuming:
  - Y = f(X)

Activation

$$f(x) = NoLinear_1(a_1 * NoLinear_0(a_0 x + b_0) + b_1)$$

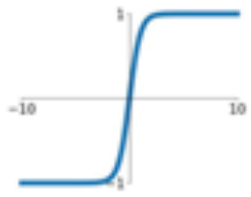
Sigmoid

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



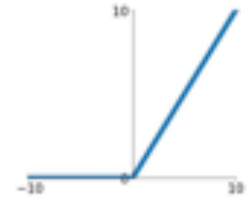
tanh

$\tanh(x)$



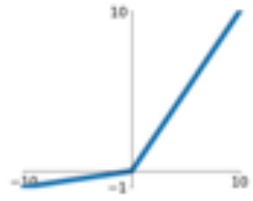
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

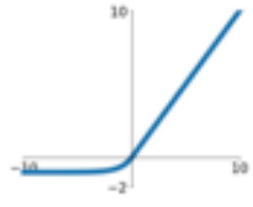


Maxout

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

ELU

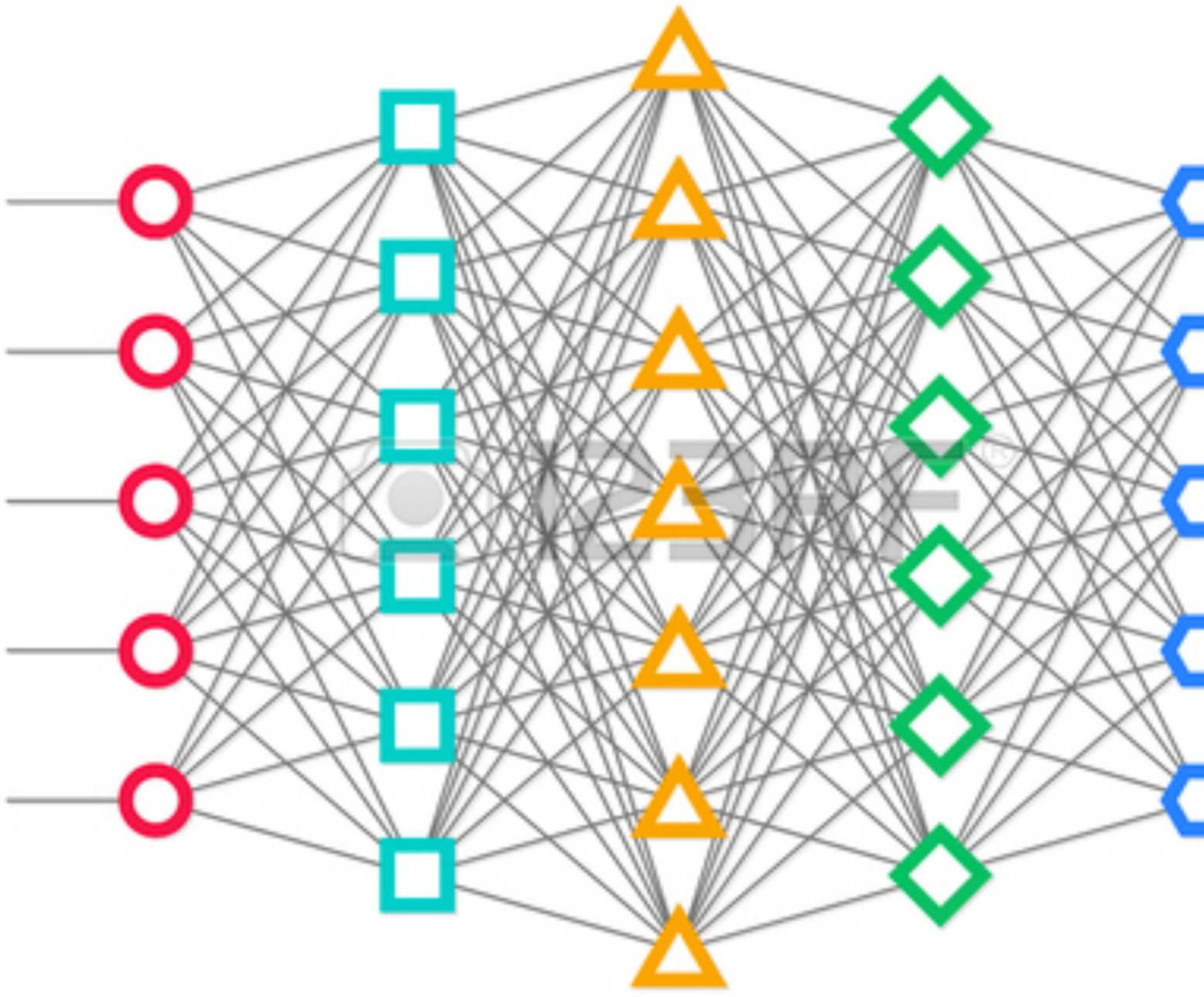
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



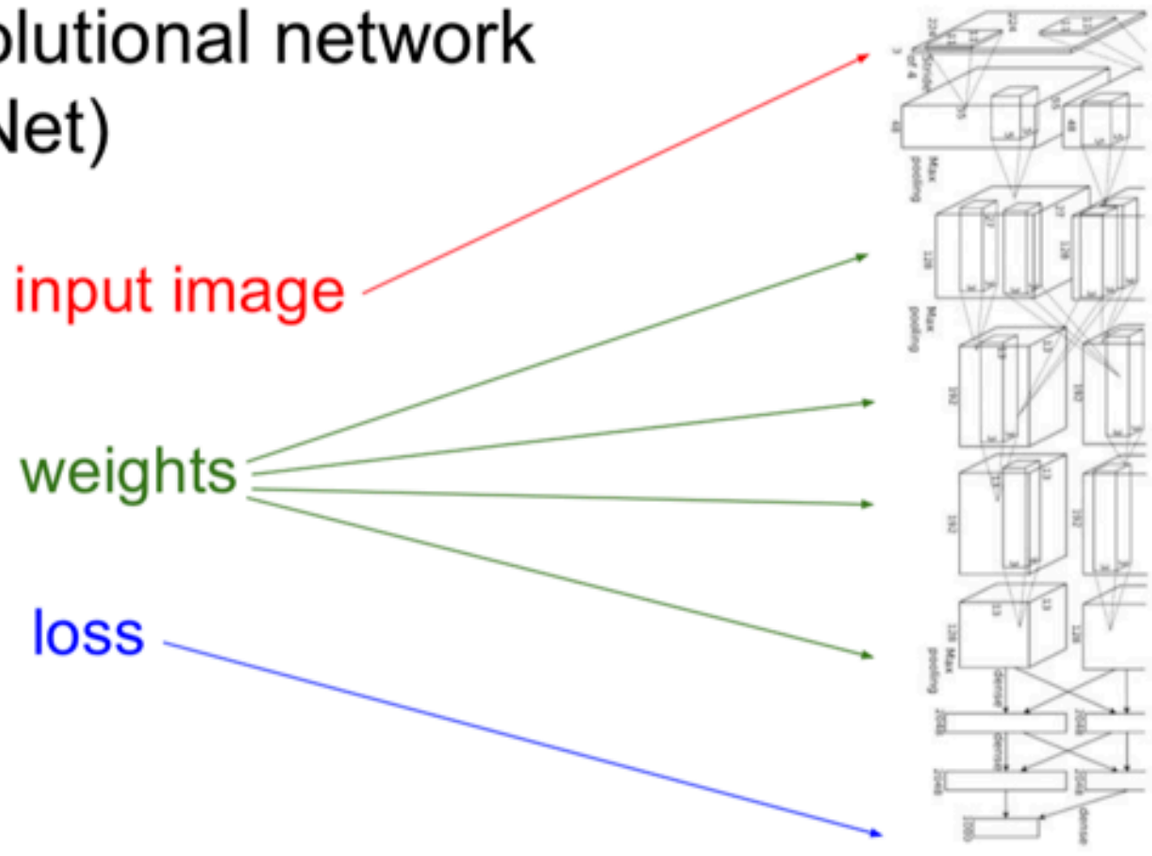
The Power of NonLinear

Or Complexity Relation Mapp

- X = [x0, x1, x2, … Xn]



AlexNet

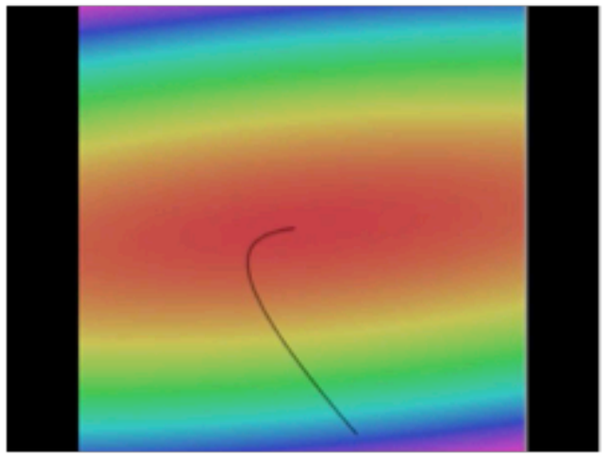


Loss Function

$$Loss(f(\mathbf{x}); \mathbf{y}) = \sum_{i=1}^{M} (y_i - f(x_i))^2$$

$$f(x) = NoLinear_1(a_1 * NoLinear_0(a_0 x + b_0) + b_1)$$

- How to get the optimal weights?



$\nabla\varphi$或$\operatorname{grad}\varphi$

其中∇（nabla）表示向量微分算子。

$\nabla\varphi$在三维直角坐标中表示为

$$\nabla\varphi = \left(\frac{\partial\varphi}{\partial x}, \frac{\partial\varphi}{\partial y}, \frac{\partial\varphi}{\partial z}\right)$$

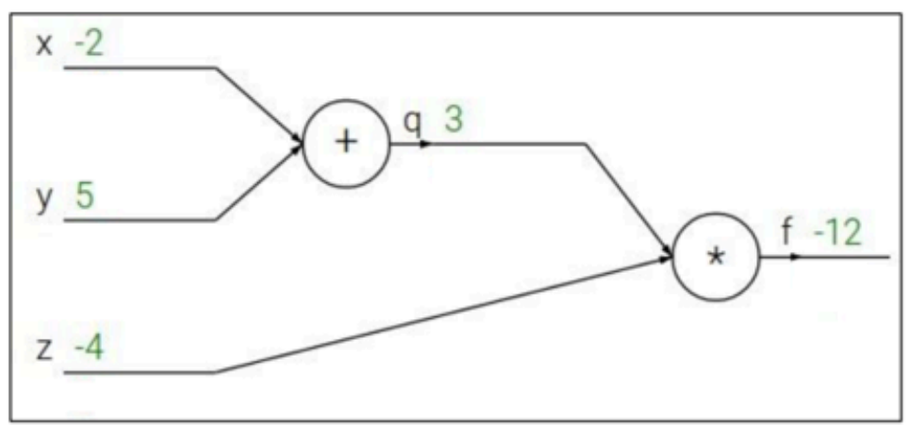
```
while True:
    weight_grad = evaluate_gradient(loss_fun, data, weights)
    weights += - step_size * weights_grad
```

Gradient Descent
More About: https://cs231n.github.io/neural-networks-3/

Backpropagation: a simple example

$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4



$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Backpropagation

$$Loss(f(\mathbf{x}); \mathbf{y}) = \sum_{i=1}^{M} (y_i - f(x_i))^2$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

```
y_hat = [1.3, -1.3, 0.2, 2.5]
y_hat = np.exp(np.array(y_hat)) / sum(np.exp(np.array(y_hat)))
y_hat
```

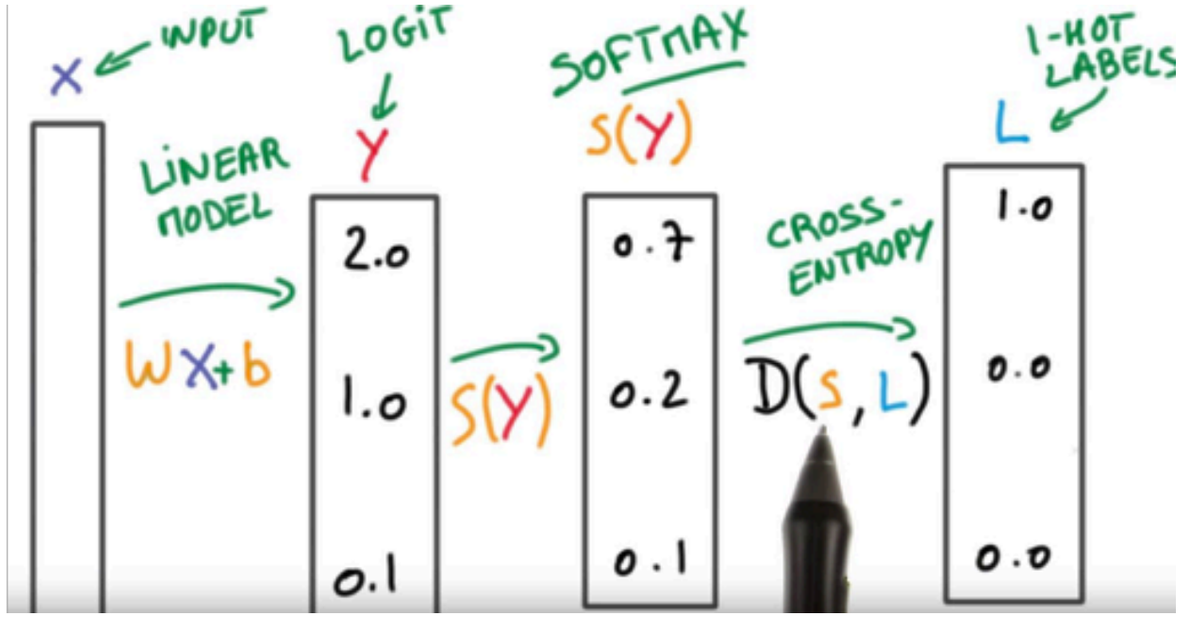
```
array([ 0.21153896,  0.01571176,  0.0704152 ,  0.70233408])
```

What loss function we use? For Regression and Classification

- 1. Get the probability of each class
- 2. The predicated *right* label close to 1.0

$$\mathrm{H}(X) = \sum_{i=1}^{n} \mathrm{P}(x_i)\,\mathrm{I}(x_i) = -\sum_{i=1}^{n} \mathrm{P}(x_i)\log_b \mathrm{P}(x_i),$$

$$-\sum_{c \in C} y_c * log(\hat{y}_c)$$



- Image source: Google Tensorflow Course

After we get loss

- Using backpropagation to get the better weights
- W += -1 * learning_rate * W

$$Loss = Loss + \lambda \sum_{p \in Parameters} W_p^2$$

Regularization: To make weights smoother

Neural Network Vs Other Machine Learning Models

- Powerful
- Generalization
- Versatile
- Data Feeding

Assignment

1. Complete the tensorflow tutorial : https://github.com/Computing-Intelligence/jupyters_and_slides/tree/master/2019-summer/assignments

2. Using K-means to make a news text cluster.

Next: MiniFlow, Tensorflow, Keras, pytorch

- https://www.tensorflow.org/tutorials/keras/
- https://keras.io/getting-started/sequential-model-guide/
- https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/udacity