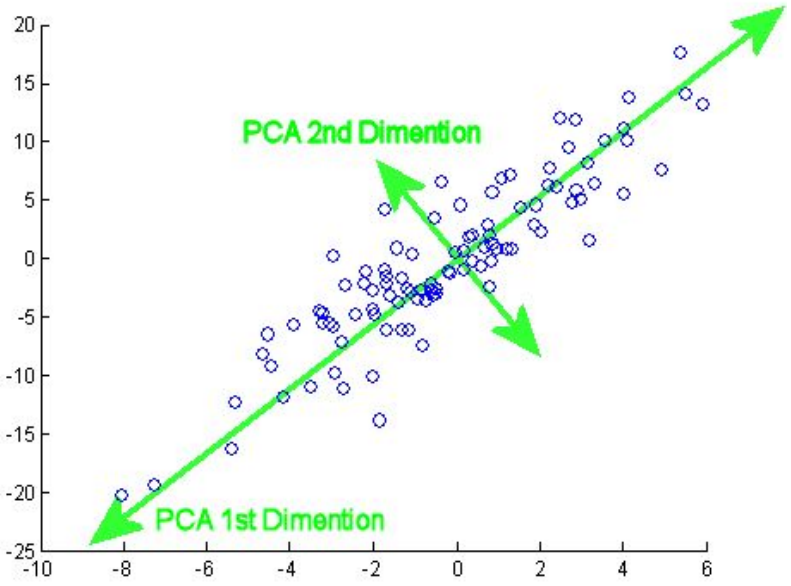
Gengbo Liu and Yutian Gui ICA and CNN report
ICA (Independent Component Analysis):

Description:

In the multi-dimensional data processing, in order to decompose data into components that simplify the analysis and reduce storage space, some dimension reduction methods are applied, include principal component analysis, PCA and independent component analysis, ICA.

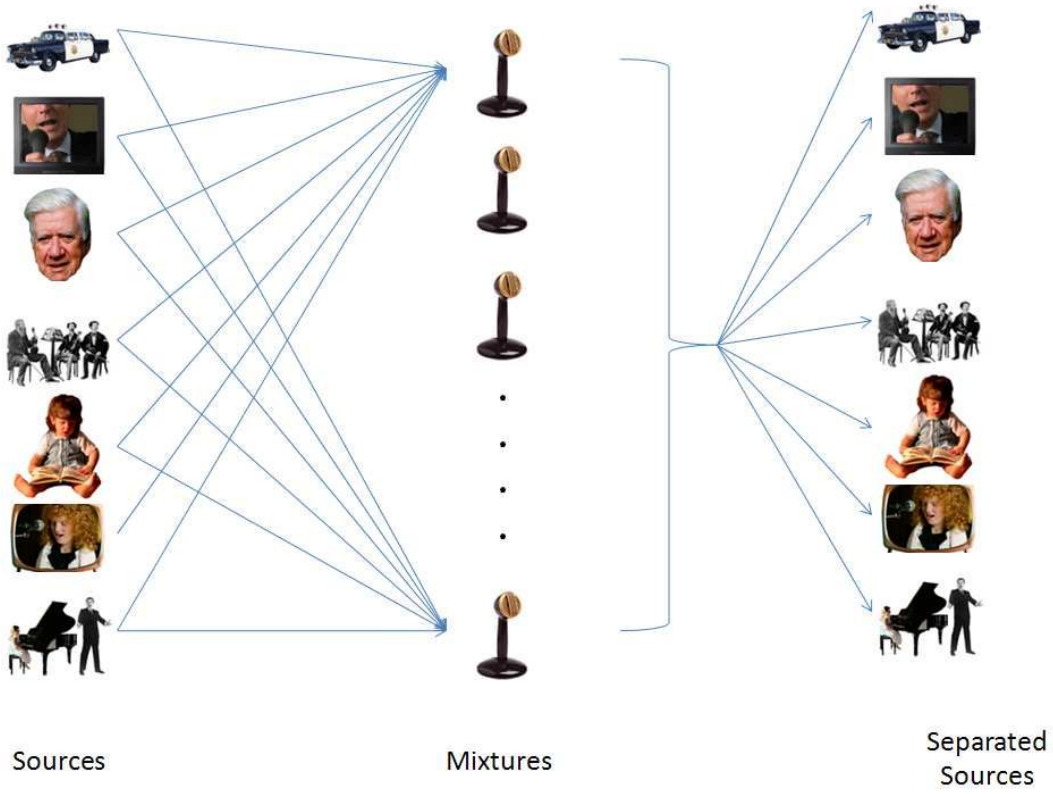
PCA

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possible correlated variables into a set of values of linearly uncorrelated variables called principal components. [1] PCA maximize the variance on the multi-dimensional statistic data and therefore find globel feature of the multi-dimensional data. [2] There are two way to apply PCA: eigenvalue decomposition, EVD and singular value decomposition, SVD [3] [4]



Comparing with PCA, ICA which is applied to solve blind source separation problem, BSS, do not need have orthogonal system. In signal processing, **independent component analysis** (**ICA**) is a computational method for separating a multivariate signal into additive subcomponents. This is done by assuming that the subcomponents are non-Gaussian signals and that they are statistically **independent** from each other. [5] One of the classic example to understand the problem ICA dealing with is cocktail party problem. [6] This problem is about separating different sound sources in a noisy,

unordered cocktail party. ICA is a perfect tool to solve this problem since it is used to find fundamental and mutually independent components.



The general model for ICA is that the sources are generated through a linear basis transformation, where additive noise can be present. Suppose we have N statistically independent signals, si (t), i = 1, ..., N. We assume that the sources themselves can not be directly observed and that each signal, si (t), is a realization of some fixed probability distribution at each time point t. Also, suppose we observe these signals using N sensors, then we obtain a set of N observation signals xi (t), i = 1, ..., N that are mixtures of the sources. A fundamental aspect of the mixing process is that the sensors must be spatially separated so that each sensor records a different mixture of the sources. There are three assumptions of ICA: 1 The sources being considered are statistically independent; 2 the independent components have non-Gaussian distribution; 3 the mixing matrix is invertible

According to central limit theorem the distribution of a sum of independent signals with arbitrary distributions ratios toward a Gaussian distribution under certain conditions. The sum of two independent signals usually has a distribution that is closer to Gaussian than distribution of the two original signals. Thus a gaussian signal can be considered as a linear combination of many independent signals. This furthermore elucidate that

separation of independent signals from their mixtures can be accomplished by making the linear signal transformation as non-Gaussian as possible. Non-Gaussianity is an important and essential principle in ICA estimation. To use non-Gaussianity in ICA estimation, there needs to be quantitative measure of non-Gaussianity of a signal. Before using any measures of non-Gaussianity, the signals should be normalised. Some of the commonly used measures are kurtosis and entropy measures. [7]

Applications:

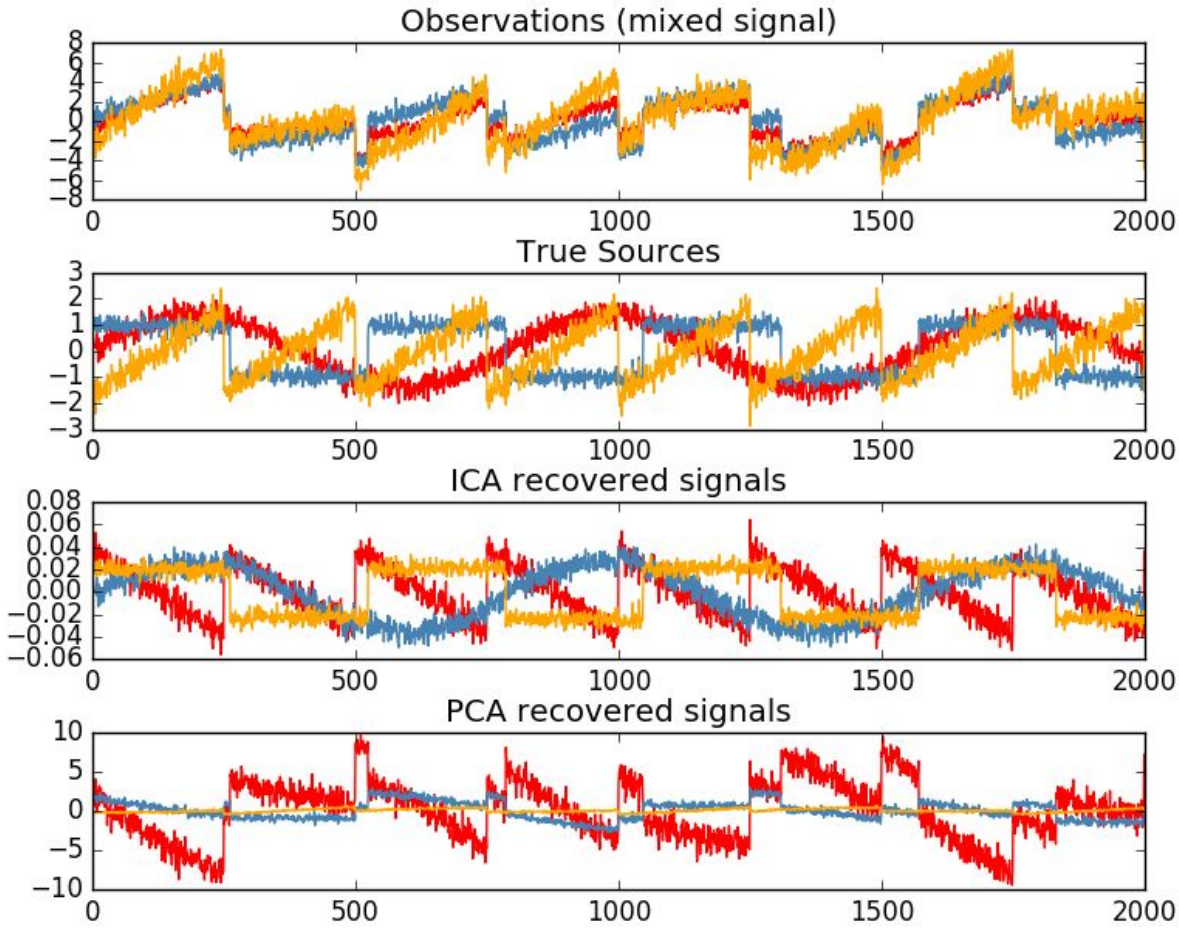
PCA and ICA are concepts in statistic and also widely used in machine learning, such as image and signal analysis. [8]

Input data of ICA:

Python :

```
# Generate sample data
np.random.seed(0)
N_samples = 2000     #x axis
Time = np.linspace(0, 8, n_samples) #y axis

s1 = np.sin(2 * time)   # Signal 1: sinusoidal signal
s2 = np.sign(np.sin(3 * time))           #Signal 2: square signal
s3 = signal.sawtooth(2 * np.pi * time)           #signal 3: saw tooth signal

S = np.c [s1, s2, s3]
S += 0.2 * np.random.normal(size = S.shape)          #add noise

S /= S.std(axis = 0)     #Standardize data
#Mix data
A = np.array([[1, 1, 1], [0.5, 2, 1.0], [1.5, 1.0, 2.0]])   #Mixing matrix
X = np.dot(S, A.T)      #Generate observations

#Compute ICA
Ica = FastICA(n_components=3)
S_ = ica.fit_transform(X)        #Reconstruct signals
A_ = ica.mixing_         #Get estimated mixing matrix

# We can ‘prove’ that the ICA model applies by reverting the unmixing.
Assert np.allclose(X, np.dot(S_, A_.T) + ica.mean_)

#For comparison, compute PCA
```

```
pca = PCA(n_components=3)
H = pca.fit.transform(X)          #Reconstruct signals based on orthogonal components

#Plot results

plt.figure()
models = [X, S, S_, H]
names = ['Observations (mixed signal)', 'True Sources', 'ICA recovered signals', 'PCA
recovered signals']
colors = ['red', 'steelblue', 'orange']
for ii, (model, name) in enumerate(zip(models, names), 1):
        plt.subplot(4, 1, ii)
        plt.title(name)
        for sig, color in zip(model.T, colors):
                plt.plot(sig, color = color)
plt.subplots_adjust(0.09, 0.04, 0.94, 0.94, 0.26, 0.46)
plt.show()
```



ICA code [9]

CNN (Convolutional Neural Network):

Description:

1. **Artificial Neural Network**

Artificial neural network(ANN), is the foundation of Convolutional Neural Network. The artificial neural network is a network inspired by biological neural networks [10], which are used to estimate or approximate functions that can depend on a large number of inputs that are generally unknown. [11]

Normally,Artificial neural networks are typically specified using three things: [12]

1) Architecture: specifies what variables are involved in the network and their topological relationships—for example the variables involved in a neural network might be the *weights* of the connections between the neurons, along with *activities* of the neurons.

2) Activity Rule: Most neural network models have short time-scale dynamics: local rules define how the *activities* of the neurons change in response to each other. Typically the activity rule depends on the *weights* (the parameters) in the network.

3) Learning Rule The learning rule specifies the way in which the neural network's weights change with time. This learning is usually viewed as taking place on a longer time scale than the time scale of the dynamics under the activity rule. Usually the learning rule will depend on the activities of the neurons. It may also depend on the values of the target values supplied by a teacher and on the current value of the weights.

Convolutional Neural Network(CNN) is a typical kind of feed-forward-artificial neural network in the area of computer science and electron. CNN is based on machine learning and it is widely used in a lot of application fields.

Simply, the main purpose of CNN is to reduce the complexity of data processing, especially in image processing and data mining. By adding some artificial mechanism, CNN algorithm can reduce the complexity of data set intelligently.

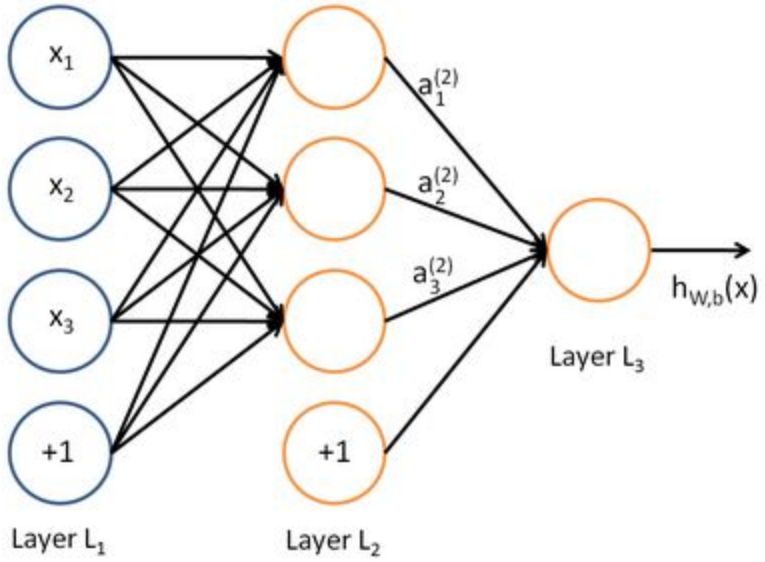


Figure Simple Architecture of an artificial neural network

Figure shows the general architecture of artificial neural network. Similar to biological neural network, the artificial neural network also consists of many neural nodes. There are three different types of neural nodes: input node, hidden node and output node. Each node are connected with one or more other nodes, and data can be processed by the different rules among these units. In figure, black lines means different rules among nodes in the neural network.

2. Convolutional Neural Network

In machine learning, a convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network in which the connectivity pattern between itsneurons is inspired by the organization of the animal visual cortex. [13]

On the basis of the multiple architecture of artificial neural network, if the quantity of input nodes and rules of hidden node is too many, then the calculation will be increased exponentially. For example, given a simple artificial neural network with 10000 inputs and a same number of hidden nodes, then the total number of data to be received in hidden level will be 10000 x 10000 = 10^8. As a result, if the complexity of input data and hidden node is high, the efficiency of whole system will be decreased obviously.

There are several ways to solve this problem. Reducing the amount of parameters(rules) between input nodes and hidden nodes is the most effective method

to improve the performance of neural network. By adding some algorithm which can reduce some relative parameters, the convolutional neural network can bring an remarkable improvement of performance to data processing and image processing. [14]

Convolutional neural networks (CNNs) consist of multiple layers of receptive fields. These layers can be seen as “filters”, and it works following different algorithms. Typically, there are two main steos to reduce and remove irrelative or less relative parameters in neural networks.

1) Local receptive fields: Similar to humans’ neural system, perceptrons in artificial neural network also work from local area to the whole situation. The connection of two nodes decreases with the increasing of distance between them. So, it is unnecessary to percept all the data set at same time. Local receptive fields provide a local connection mechanism to reduce the complexity of whole system. The whole data set is divided to several small groups of data, and data in each group are connected with each other independently. Then, the results of each small group are collected and computed on a higher level layer. Figure shows the working architecture of traditional neural network and convolutional neural network with local receptive fields mechanism.

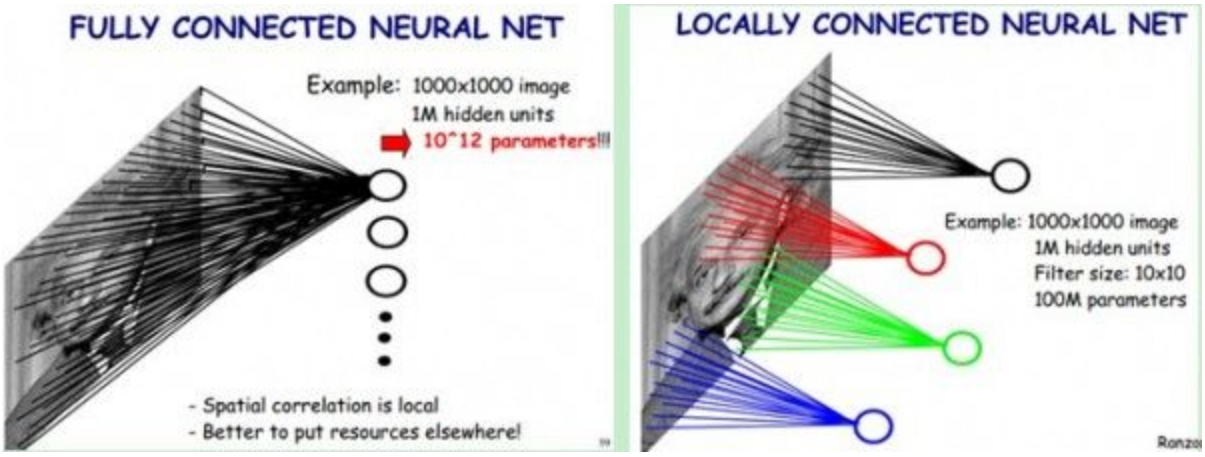


Figure Fully connected Neural Net vs. Locally Connected Neural Net

By local receptive fields, the calculation can be reduced easily. For example, in figure, data size is 1000 x 1000 and there are 1000000 hidden units in the data set. By using traditional neural network, there will be 10^12 parameters. By contrast, if each hidden node connect only 10 input nodes, then the number of parameters will be reduced to 10^8.

2) Weight sharing: After local receptive fields, the data set is divided to many small groups, and the size of each group is same. If some of these groups have same or similar feature, it means that these groups are replaceable to each other. Picking one of them as a convolution kernel, then the quantity of parameters will be reduced further. This process is called weight sharing.

   Generally, it is only a ideally situation that all the groups in data set are totally same. So, as an alternative solution, the system will generate several convolution kernels. This process sorts out same or similar groups and replace them with a convolution kernel.

Input data of CNN:
Using Mnist dataset by python [15][16]

```
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from urllib import urlretrieve
import cPickle as pickle
import os
import gzip
import numpy as np
import theano
import lasagne
from lasagne import layers
from lasagne.updates import nesterov_momentum
from nolearn.lasagne import NeuralNet
from nolearn.lasagne import visualize
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

def load_dataset():
    url = 'http://deeplearning.net/data/mnist/mnist.pkl.gz'
    filename = 'mnist.pkl.gz'
    if not os.path.exists(filename):
        print("Downloading MNIST dataset...")
        urlretrieve(url, filename)
    with gzip.open(filename, 'rb') as f:
        data = pickle.load(f)
```

```
    X_train, y_train = data[0]
    X_val, y_val = data[1]
    X_test, y_test = data[2]
    X_train = X_train.reshape((-1, 1, 28, 28))
    X_val = X_val.reshape((-1, 1, 28, 28))
    X_test = X_test.reshape((-1, 1, 28, 28))
    y_train = y_train.astype(np.uint8)
    y_val = y_val.astype(np.uint8)
    y_test = y_test.astype(np.uint8)
    return X_train, y_train, X_val, y_val, X_test, y_test
```

As you can see, we are downloading the MNIST pickled dataset and then unpacking it into the three different datasets: train, validation and test. After that we reshape the image contents to prepare them to input into the Lasagne input layer later and we also convert the numpy array types to uint8 due to the GPU/theano datatype restrictions. [17] [18]

[1] https://en.wikipedia.org/wiki/Principal_component_analysis
[2] https://www.youtube.com/watch?v=e4woe8GRjEI
[3] http://blog.csdn.net/zrjdds/article/details/50318065
[4] http://blog.csdn.net/xiaojidan2011/article/details/11595869
[5] https://en.wikipedia.org/wiki/Independent_component_analysis
[6] http://research.ics.aalto.fi/ica/cocktail/cocktail_en.cgi
[7] http://cdn.intechopen.com/pdfs-wm/39839.pdf
[8] http://www.mit.edu/~gari/teaching/6.555/LECTURE_NOTES/ch15_bss.pdf
https://www.youtube.com/watch?v=GfIQlql-i2k
[9]http://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_blind_source_separation.html
[10] Hentrich, Michael (2015)."Methodology and Coronary Artery Disease Cure"
[11] https://en.wikipedia.org/wiki/Artificial_neural_network
[12] MacKay, David, J.C. (2003). Information Theory, Inference, and Learning Algorithms . Cambridge University Press. ISBN 9780521642989.
[13] https://en.wikipedia.org/wiki/Convolutional_neural_network
[14] Ciresan, Dan; Meier, Ueli; Schmidhuber, Jürgen (June 2012). "Multi-column deep neural networks for image classification". 2012 IEEE Conference on Computer Vision and Pattern Recognition.
[15] http://www.csdn.net/article/1970-01-01/2825549
[16]http://blog.christianperone.com/2015/08/convolutional-neural-networks-and-feature-extraction-with-python/
[17] http://doc.okbase.net/u012162613/archive/126058.html
[18] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/