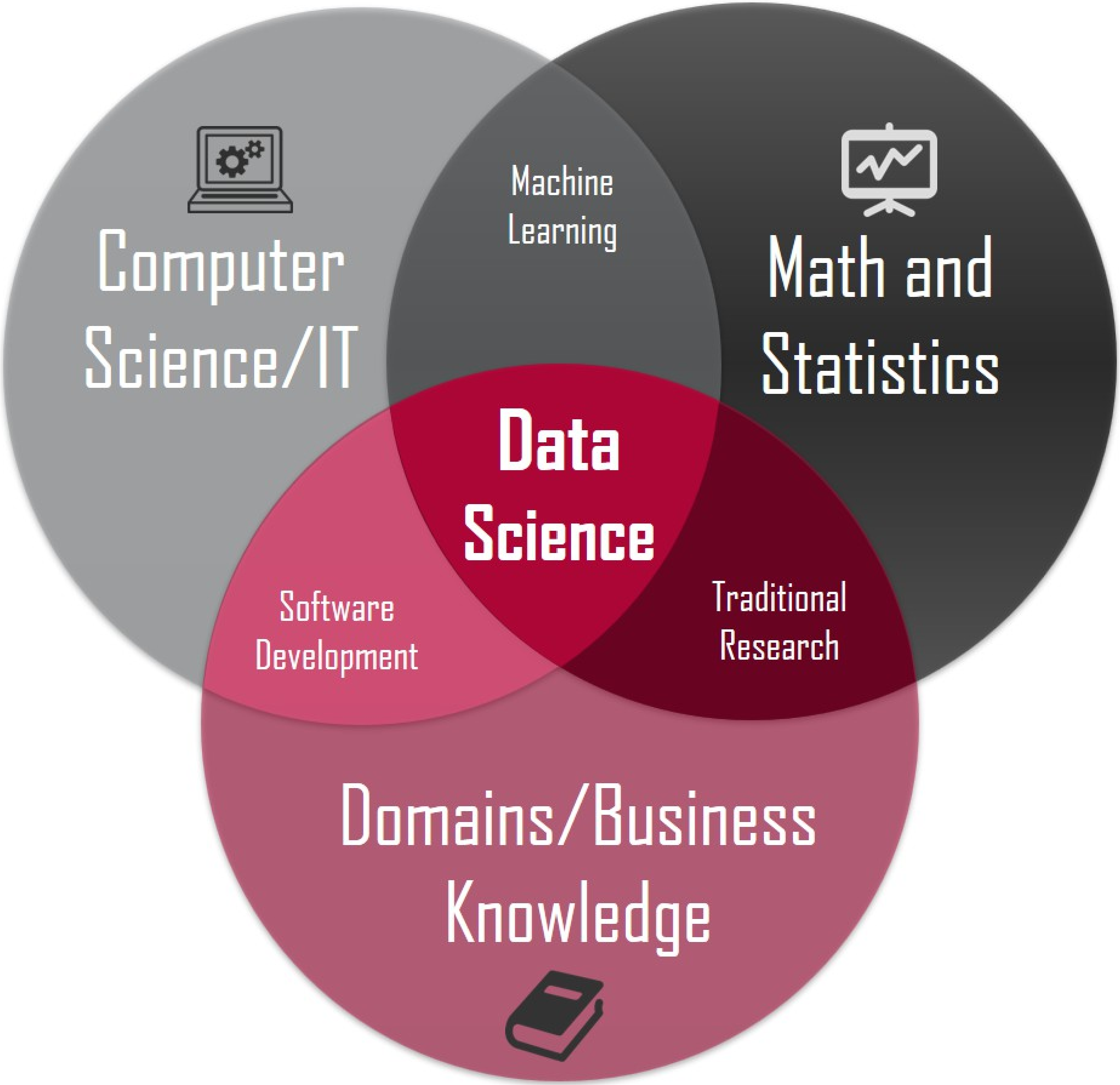
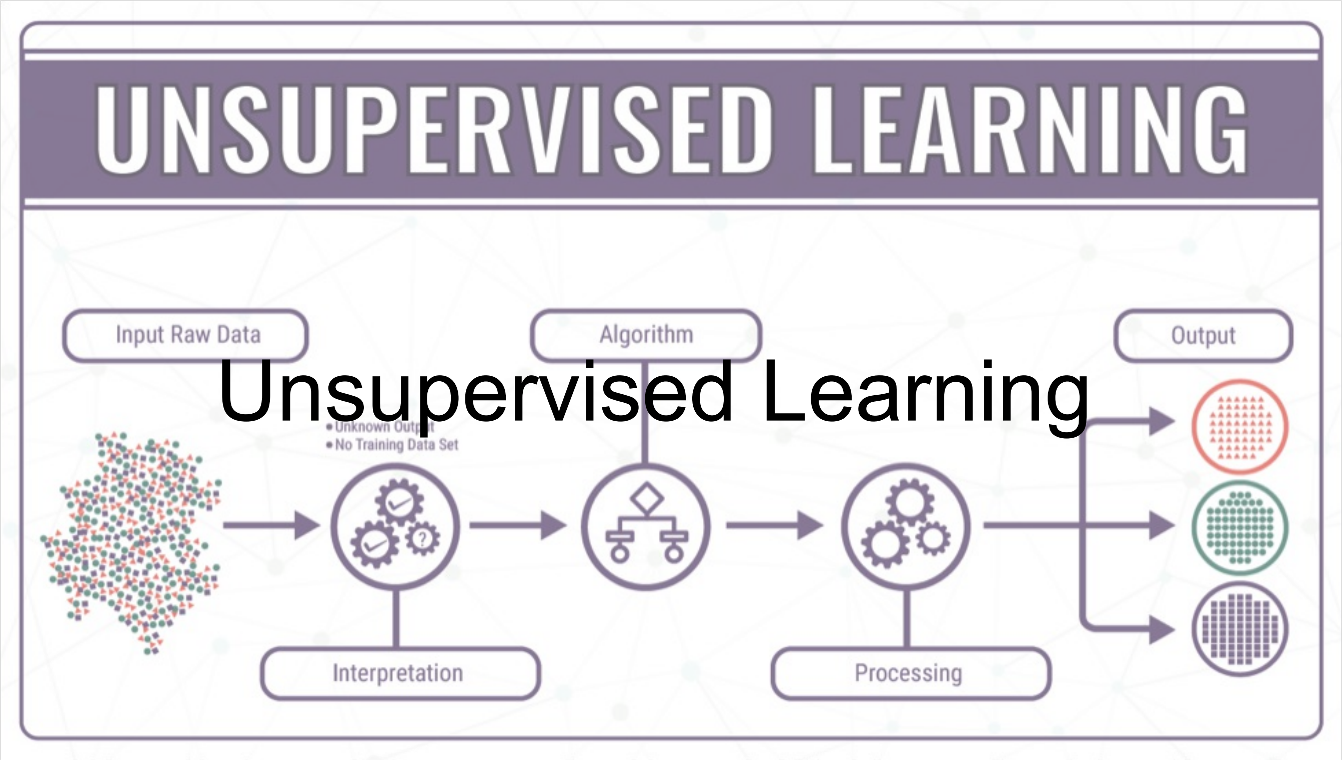
Summary/Recap Lecture 1

- Machine Learning is tightly coupled to an optimization problem
- Checking variance in model performance is extremely important
- Large model require lots of data
- Assembling models yields better performance


Computer Science/IT
Machine Learning
Math and Statistics
Data Science
Software Development
Traditional Research
Domains/Business Knowledge

Lecture - Part 2/3

Unsupervised Learning



Unsupervised Learning : outline

- We are given a dataset, and **no ground truth/label/target**
- No objective function either
- Aim is at **finding structure, similarities, trend,** …

- Most common applications
  - Dimensionality reduction
  - Clustering
  - Anomaly detection
  - Generative model

Dimensionality Reduction

Dimensionality Reduction

- In the context of machine learning, the dimensionality of the input can be a showstopper computationally
- The input data dimensionality (number of pixels in the image) is usually much bigger than the dimension of the manifold where information lives

- Find a lower dimension representation of the data
  - PCA
  - Auto-encoder

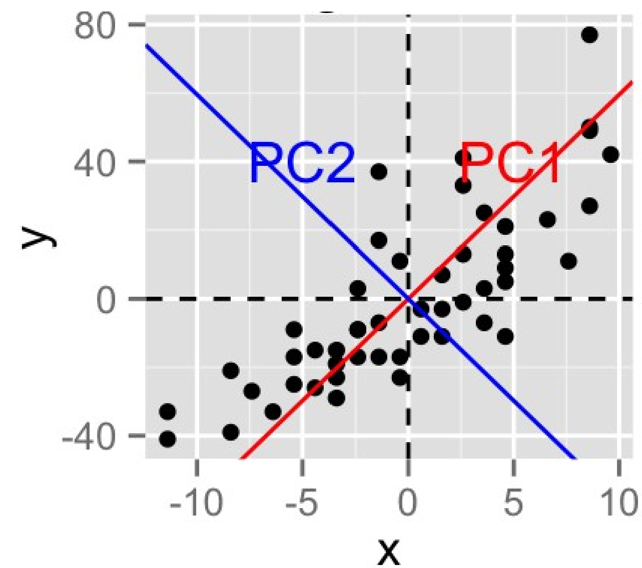
Principal Component Analysis

- Generally useful in data pre-processing
- The method aims at finding a new basis of the data in which components have maximal and decreasing variance
- The eigendecomposition of the covariance matrix provides this new basis. Composed of the first eigenvectors in decreasing order of the eigenvalues.
- Numerically straight forward since S is positive definite

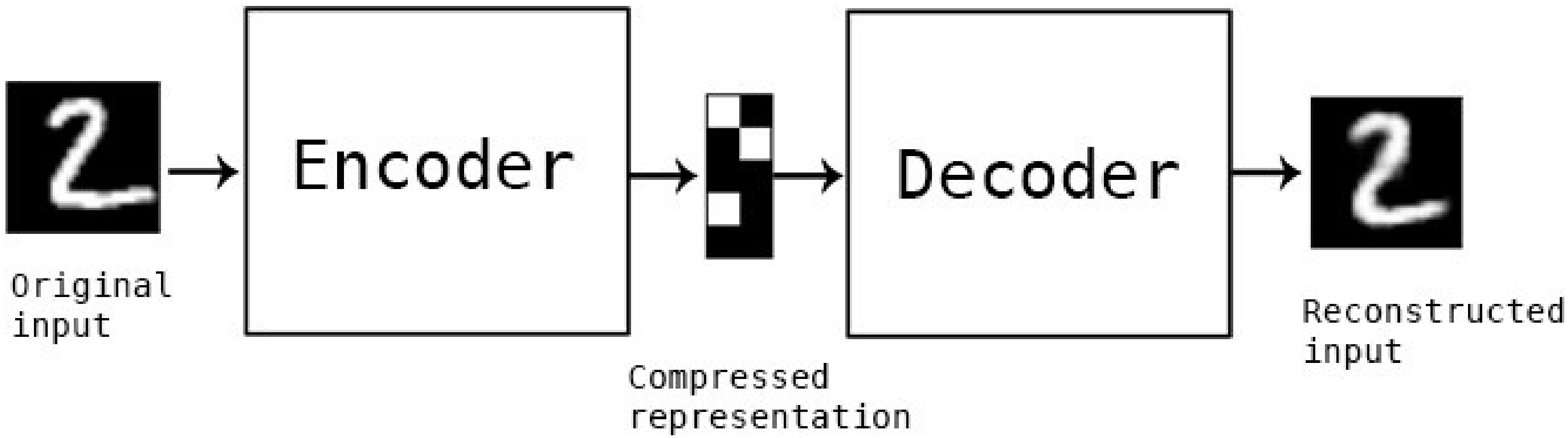
$$S=\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2$$

$$u=Ux \text{ with } S=U\begin{bmatrix}\lambda_1 & & 0\\ & \ddots & \\ 0 & & \lambda_n\end{bmatrix}U^{-1}$$

$$\text{and } \lambda_i \geqslant \lambda_j \text{ for } i \leqslant j$$



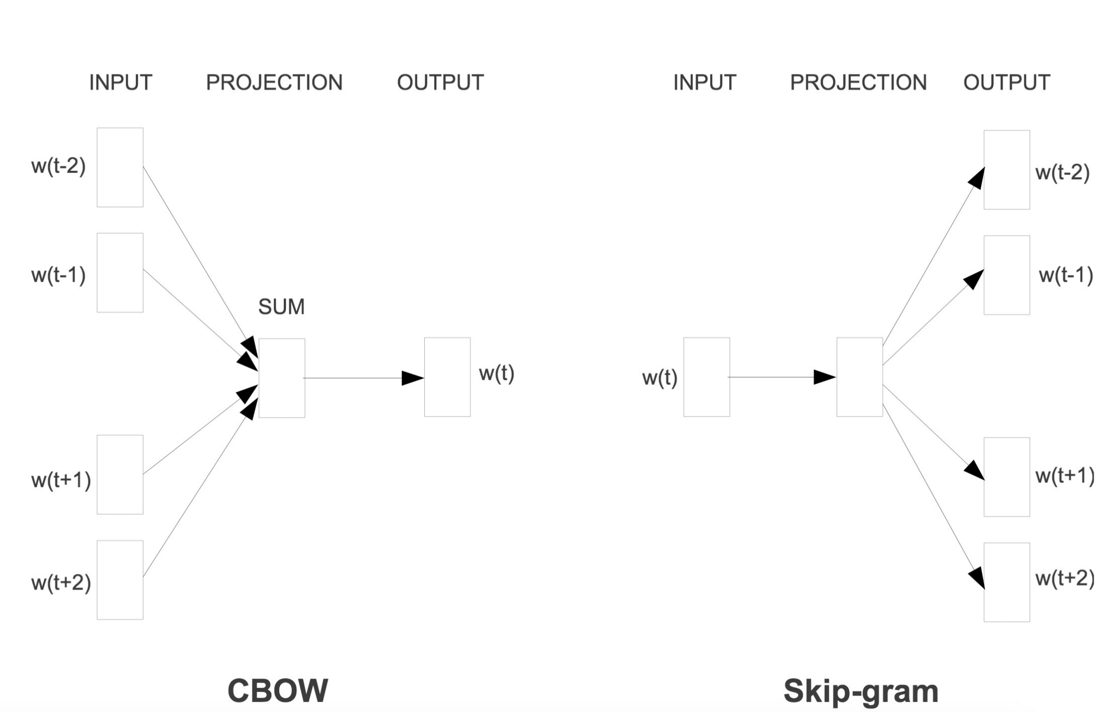
Auto-Encoder



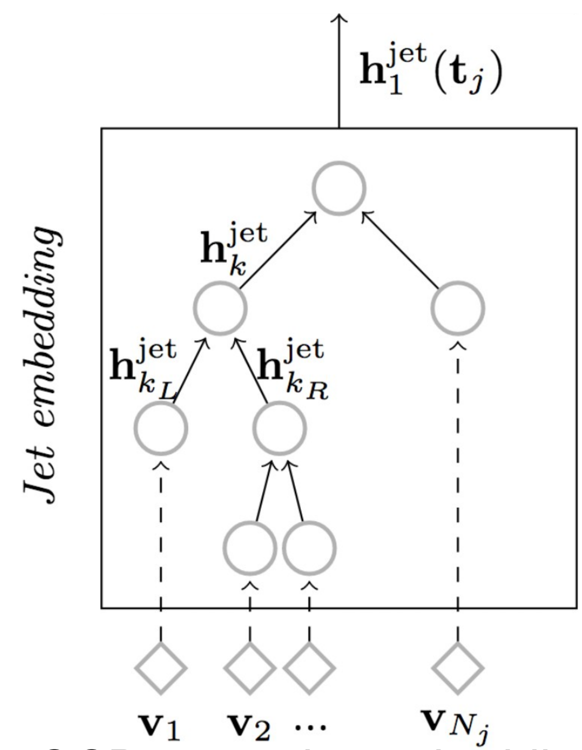
- Learning the identify function through an encoder and a decoder
- Multiple possible usage
  - Clusterization, using distance in the compressed representation instead of original input space
  - Anomaly detection using find outliers in the internal representation
  - De-noising model, by training on noisy input and de-noised output

Embedding

- Embedding is a mapping from input space to real valued tensor
- Learned as part of the model, or as a standalone task

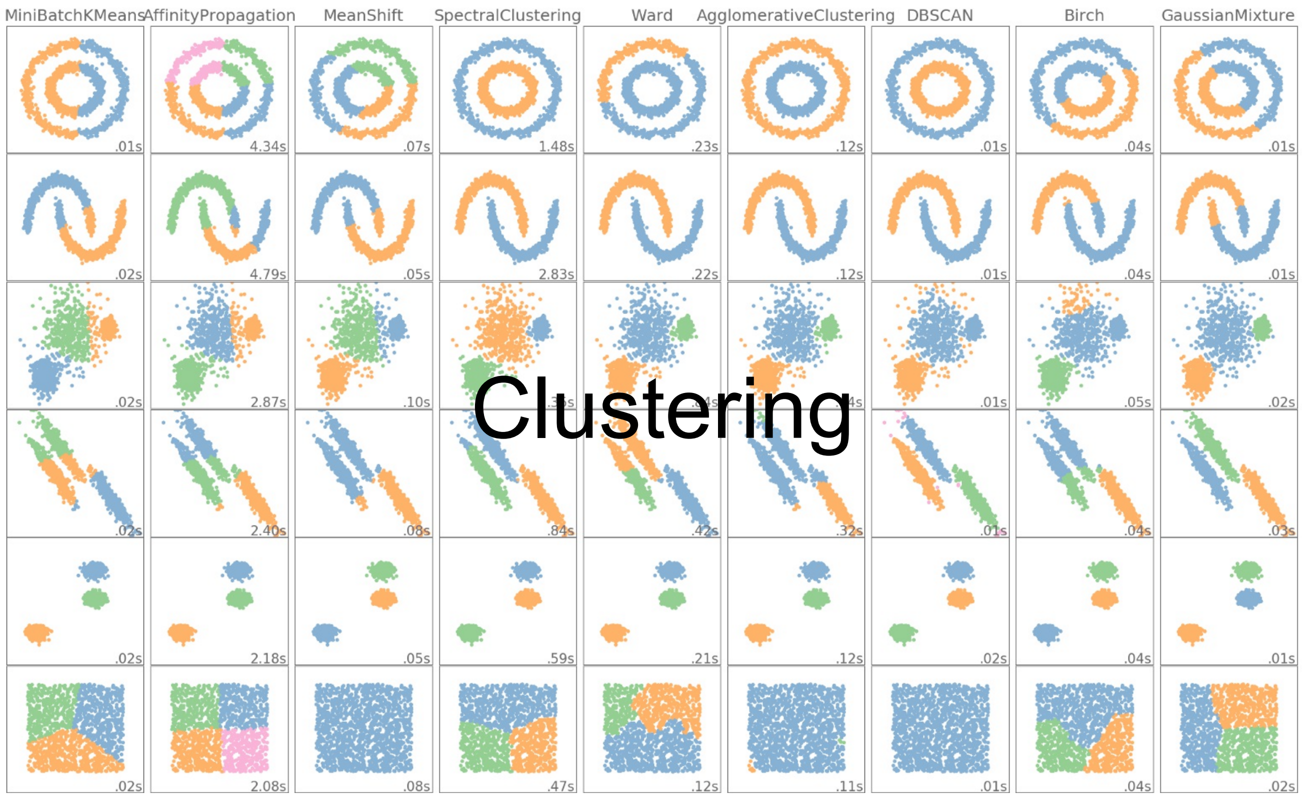


word2vec 1301.3781



QCD-aware jet embedding
1702.00748

Clustering


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture

Clustering

- Problem statement : are there different populations of samples in the dataset, where a population is a subset of samples similar among themselves, with a given metric of similarity
- A natural choice of similarity on real valued tensors is the euclidian distance, but any metric can be specified
- Clustering can be done on the raw data, embedding data, or compressed representation
- Several popular methods
  - K-means
  - DBScan
  - Self-organizing map (SOM)

K-Means

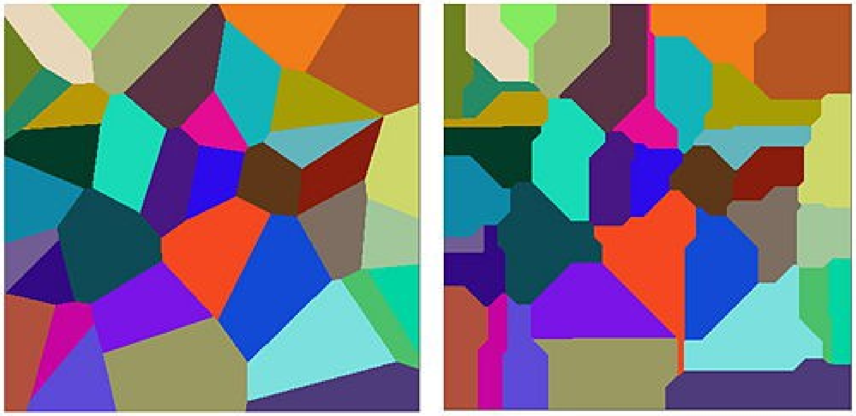
- Assume a number of clusters K originally positioned at random in feature space $\mu_k \quad for \quad C_k$
- Assign each data point to the nearest cluster, according to the chosen metric D

$$k_i = \underset{k}{arg\,min}\, D(x_i, \mu_k)$$

- Update the cluster position in feature space, once all samples are assigned

$$\mu_k = \frac{1}{n_k} \sum_{i \in C_k} x_i$$

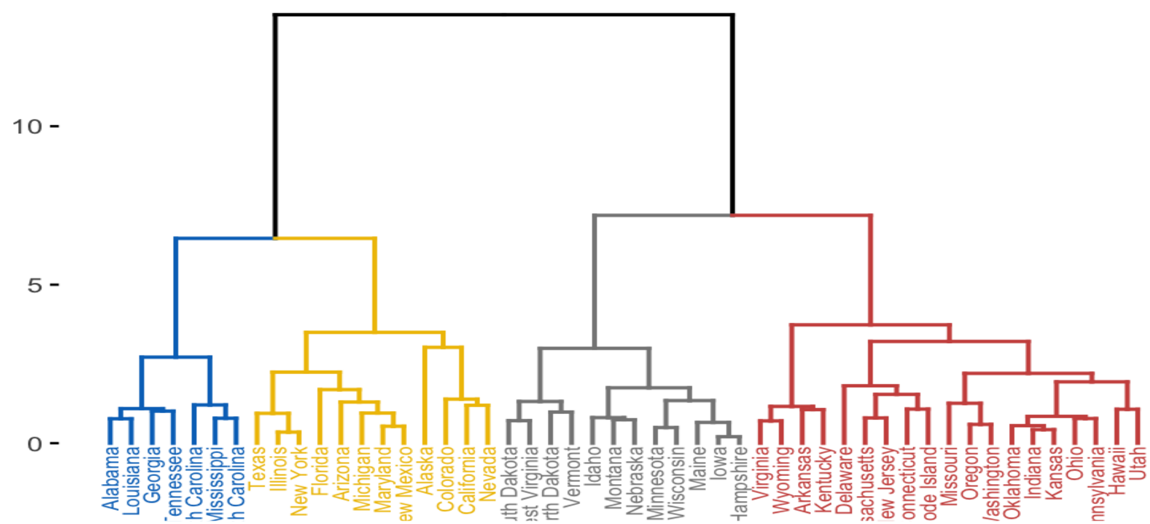
- ✓ Repeat until cluster positions are not changing significantly
- ✓ Predefined value of K can be obtained with optimization



Euclidean Manhattan

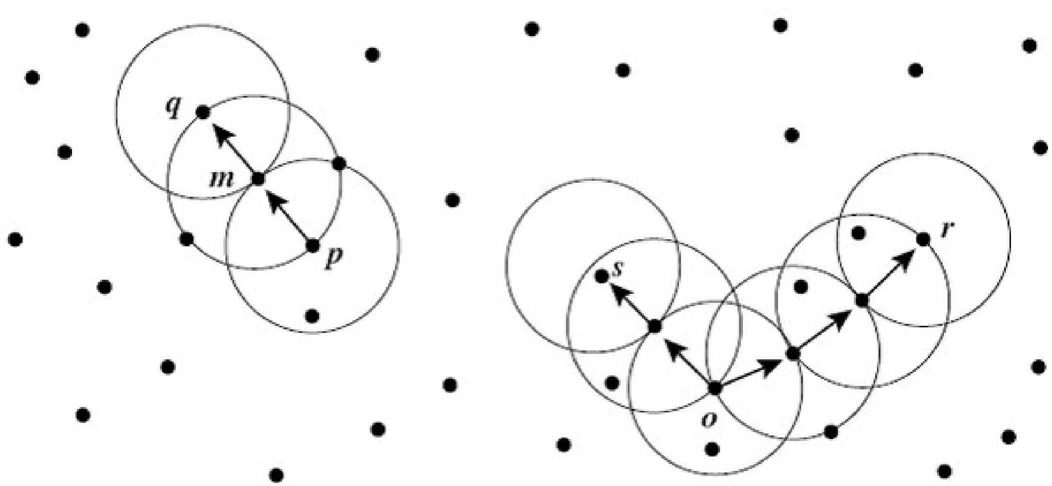
Hierarchical Agglomerative Cluster

- HCA is an algorithm that provides a dendogram over the dataset provided a distance
- In each branch of the tree samples are closer to each other then samples in the other branch
- Each horizontal cut of the tree provides a clustered view of the sample
- Computationally intensive if no adjacency provided



DBSCAN

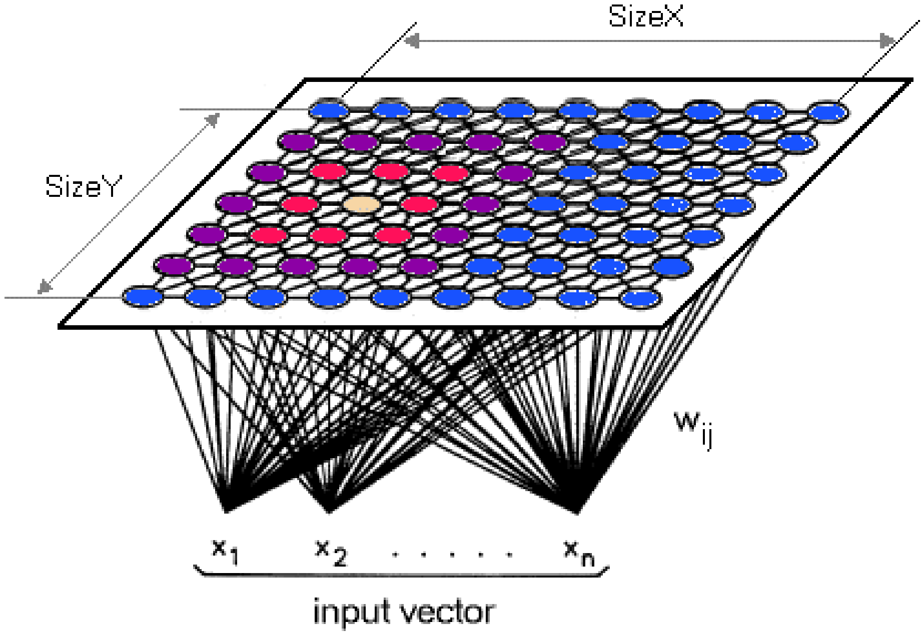
- Density-based spatial clustering of application with noise (DBSCAN) is not making assumption on the number of clusters, and can define outliers.
- For each data point find all points within a certain distance (proximity parameter) and grow clusters by vicinity if sufficient neighbors (population parameter)
- Performance depends on the choice of parameters. Should be optimized according to an extra figure of merit



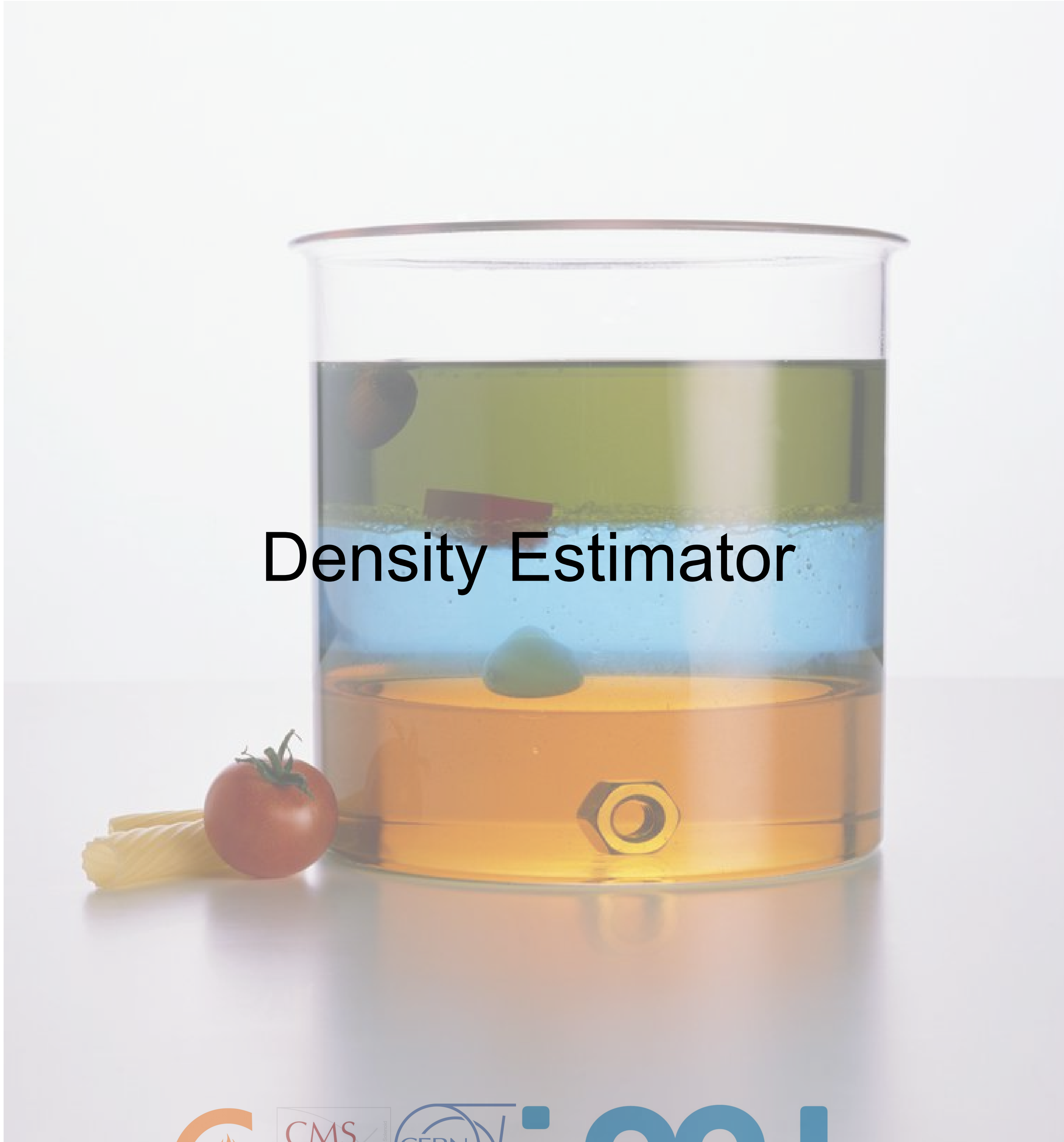
Self Organizing Map

- Kohonen network, or SOM use internal representation (usually 2D for visualization purpose, but not limited to) to encode the content of the training dataset
- Internal representation are pulled towards data input, within the neighborhood of the node most similar to the presented sample

  - Useful in visualization
  - Conserve topologies from data
  - Used for clustering
  - Can find similarities between samples in the data



Density Estimator



Density Estimator

- Further than clustering, one may want to learn the structure of the full dataset and have an estimate of the **similarity of unseen samples** to the original dataset
- In the limit of infinite number of clusters, the fine grained description of the full population is learned
- An accurate model of P(X) provides de-facto a generative model if sampled properly
- Present here advanced methods, with specific applications in mind
  - Mixture of Gaussians
  - NADE
  - Generative models
  - Variational Auto-Encoder

Mixture of Gaussians

- Mixture of Gaussian provides a smooth modeling
- Tractable recursive training with expectation maximization (EM)
- Model can be used to generate new sample
- Initialization comes naturally from z-means

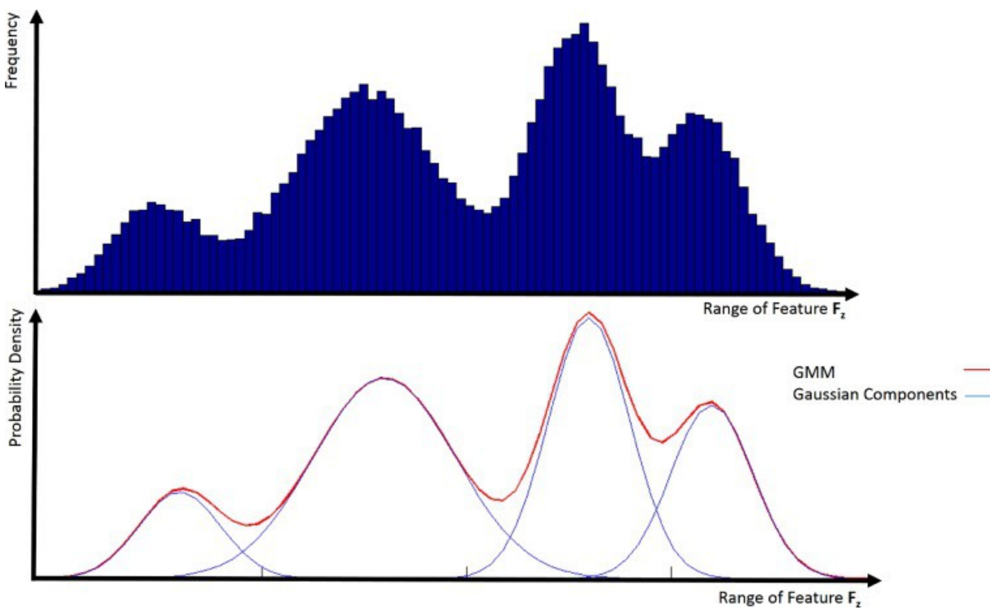
$$P(x|\theta)=\sum_k \alpha_k p_k(x|\theta_k)$$

$$p_k(x|\theta_k)=\frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$$

$$\omega_{ik}=\frac{p_k(x_i|\theta_k)\alpha_k}{\sum_m p_m(x_i)\alpha_m} \quad , \quad N_k=\sum_i \omega_{ik}$$

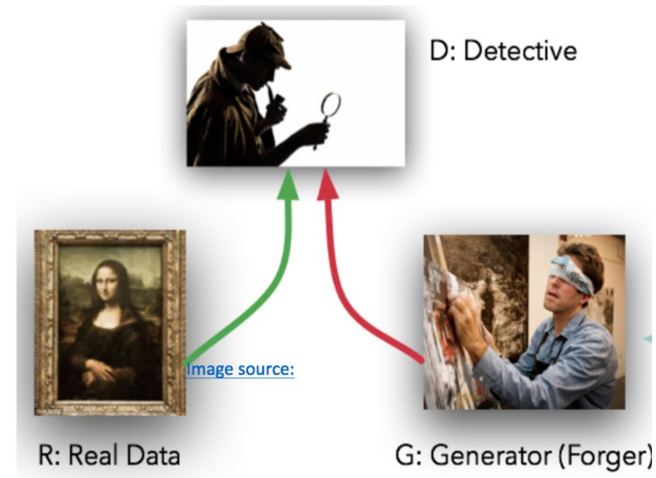
$$\mu_k \leftarrow \mu_k^{new}=\frac{1}{N_k}\sum_i \omega_{ik} x_i$$

$$\Sigma_k \leftarrow \Sigma_k^{new}=\frac{1}{N_k}\sum_i \omega_{ik}(x_i-\mu_k^{new})(x_i-\mu_k^{new})^T$$



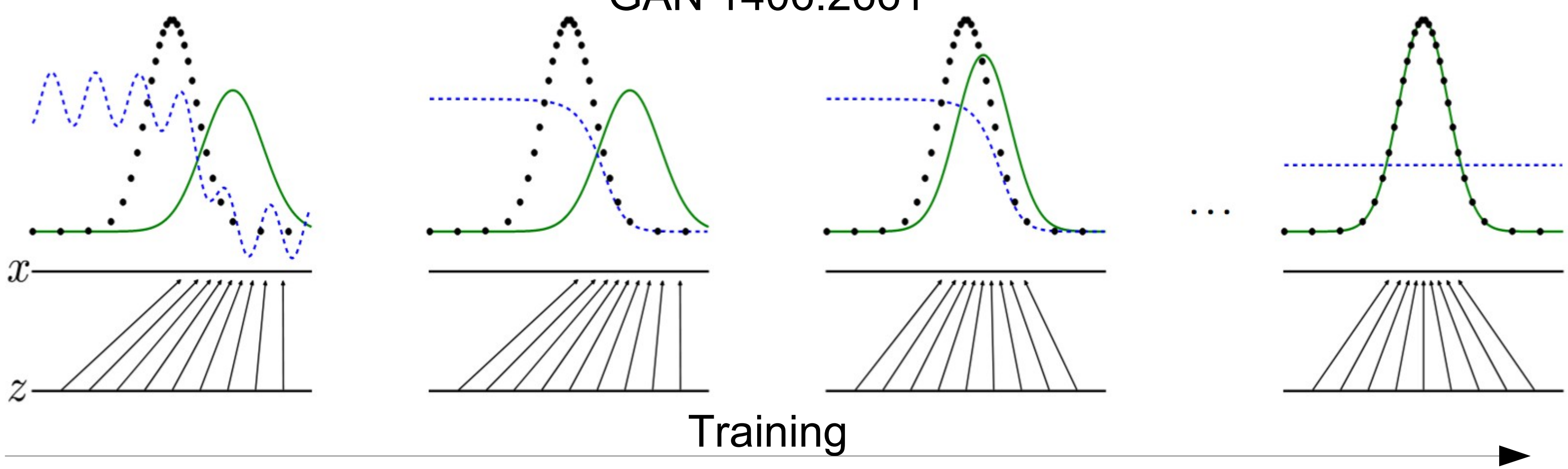
Generative Adversarial Network

- GAN are composed of two elements competing with each others
  - A **generator** : the role of which is to produce samples as if they were drawn from the original dataset
  - A **discriminator** : the role of which is to distinguish between a real sample and a generated sample



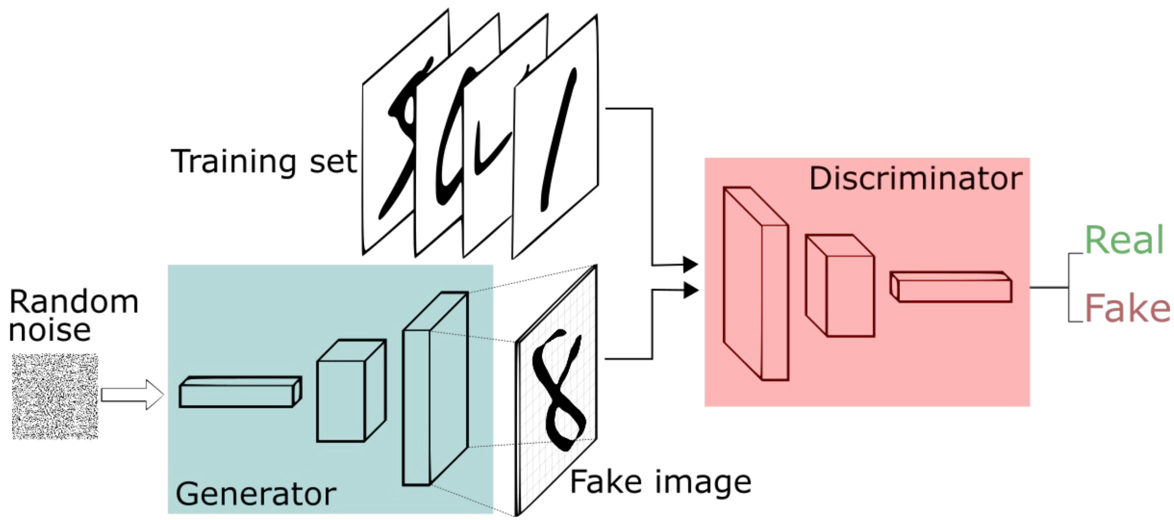
GAN 1406.2661

Data
Generator
disctriminator



DCGAN

- Generator produces samples in feature space
- Discriminator classifies real vs fake



$$\min_D \max_G Loss_{data/fake}$$

$$Loss = -\frac{1}{m}\sum_i \left[\log D(x_i) + \log\left(1 - D(G(z_i))\right)\right]$$

$$D \leftarrow D - \eta \nabla_D Loss$$

$$G \leftarrow G + \eta \nabla_G Loss$$

W-GAN

- 1st Wasserstein distance a.k.a Earth mover's distance (EMD) is a measure of similarity of probability distribution function

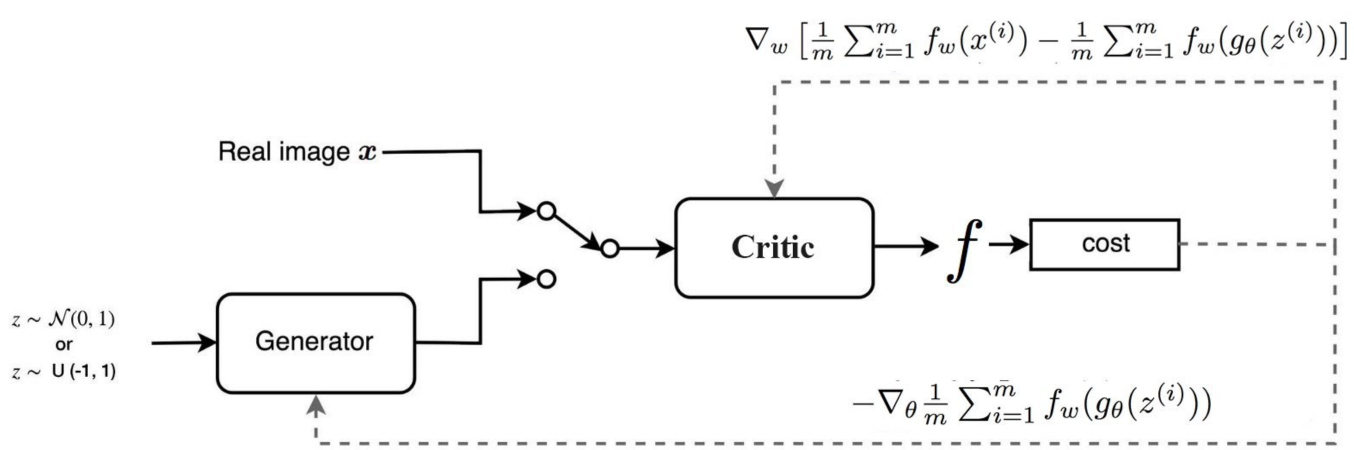
$$EMD(p,q) \equiv W(p,q) = \inf_{\gamma} E_{(x,y)\sim\gamma}\left[\|x-y\|\right]$$

$$s.t.\ \gamma_X = p\,;\ \gamma_Y = q$$

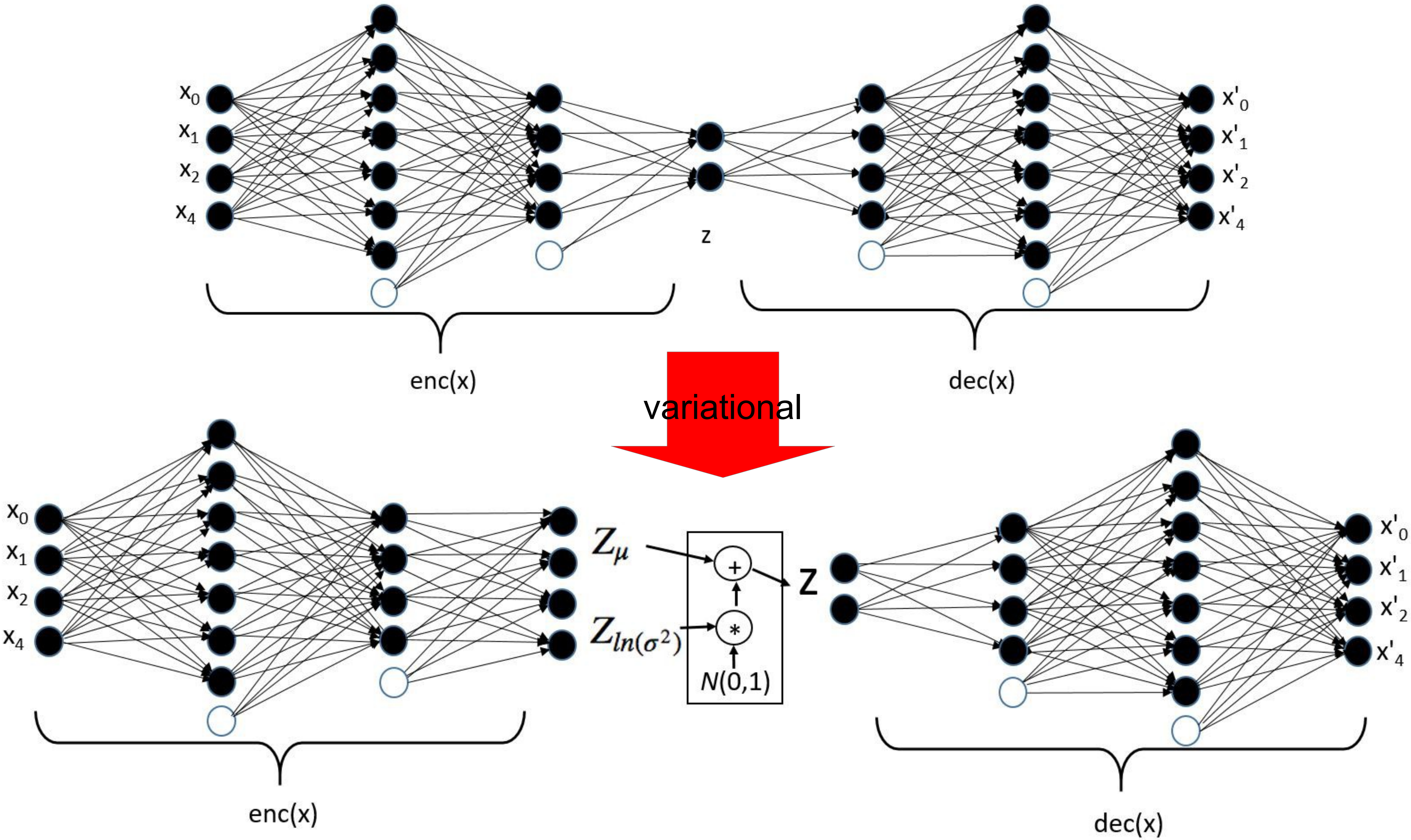
$$\text{in}\quad \gamma(x,y) = \gamma_{X|Y}(x|y)\,\gamma_Y(y) = \gamma_{Y|X}(y|x)\,\gamma_X(x),$$

- EMD is intractable for optimization
- EMD can be approximated using the critics, and

$$\nabla_\theta W(data, fake) = -E_{z\sim p(z)}\left[\nabla_\theta C(g_\theta(z))\right]$$



Variational Auto-Encoder

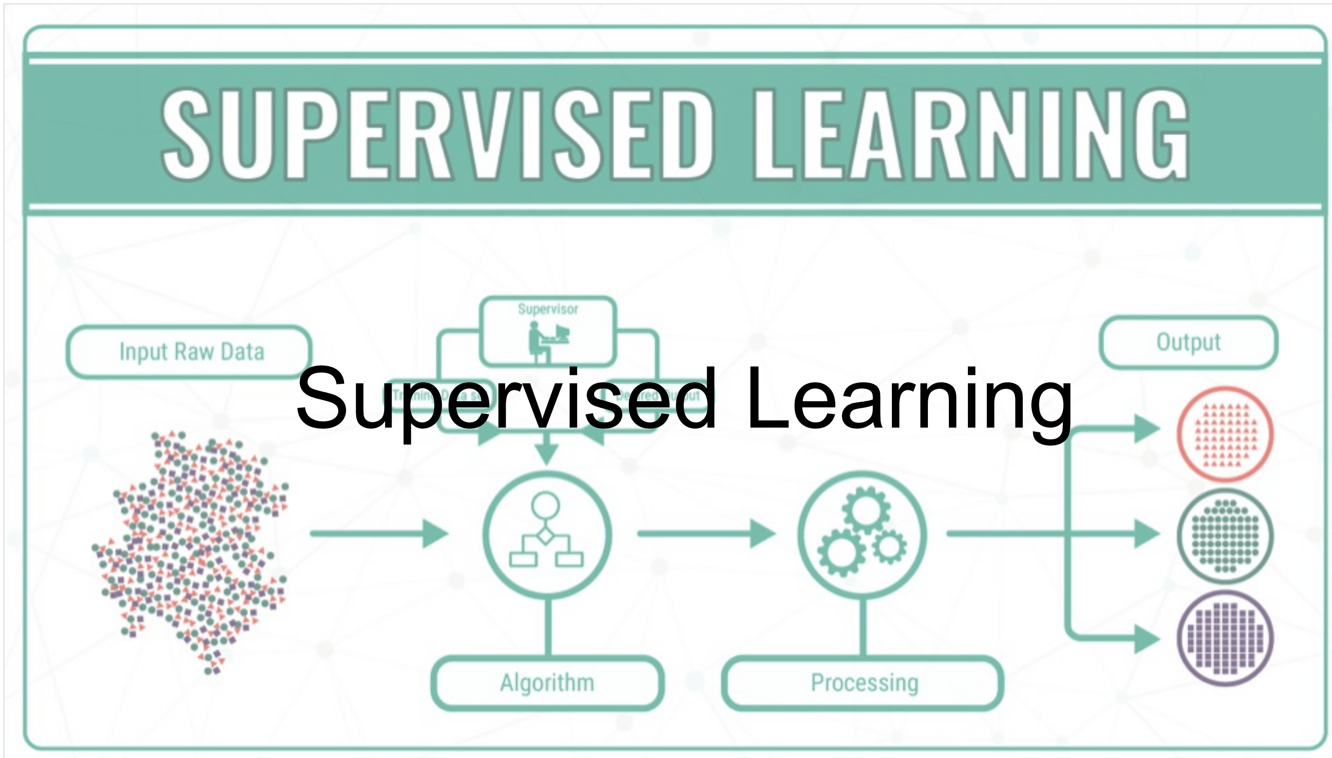


Gradient Descent on VAE

- The output of the model in the latent space and reconstructed space are **mixtures of Gaussians distributions** q (enc) and p (dec) respectively
- The loss function, per input sample, provides
  - Measure of reconstruction self consistency
  - Measure of compatibility between latent distribution and unit gaussian
- Gradient of the KL divergence between two multivariate Gaussians is **analytical**
- Gradient of expectation term simplified using the **reparametrization trick**, and evaluated using a single sample

$$loss(x_i) = -D_{KL}\left[q(z|x_i) \| Gauss(0,1)\right] + E_{q(z|x_i)}\left[\log p(x_i|z)\right]$$

Supervised Learning



Supervised Learning : Outline

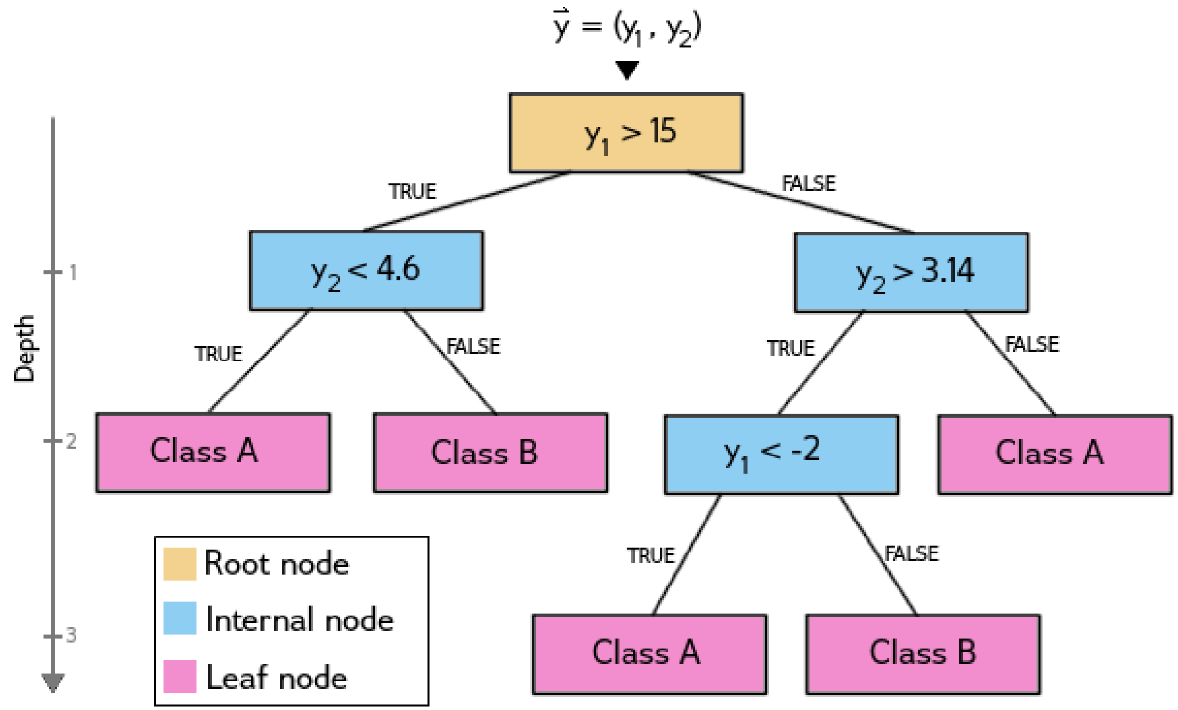
- Provided a dataset with input quantities, and quantities(target) one wishes to be able to predict

$$\{x_i, y_i\} \rightarrow y \equiv h(x)$$

- y is within a **finite set : classification**
- y is a **continuous : regression**
  - N.B. A regression and be binned and casted in a classification problem
- Looking at the most commonly used algorithms
  - Decision Tree
  - Gaussian processes
  - Support Vector Machine
  - Artificial Neural Network
    - Embeddings
    - Convolutional Layers
    - Recurrent Neural Network
    - Dense Connections

Decision Tree

- Decision trees is the most known tool in supervised learning.
- It has the advantage of being easily interpretable
- Can be used for classification or regression



Growing Decision Tree

- Decision trees are grown recursively using an impurity metric (entropy, gini, error, ...)
- At each iteration dataset's splits are enumerated and the one with the largest the impurity gain is selected
- Stopping mechanism based on tree depth, population in leaves, number of leaves, ...
- Branches and leaves subject to pruning for further improvement

$$s \equiv x_{j_s} \geqslant c_s \Rightarrow t_0 \rightarrow (t_L, t_R)$$

$$I(t) = \Phi\left(\left\{p_k \equiv \frac{N_{k,t}}{N_t}\right\}_{class\,k}\right)$$

$$\Delta I(s) = \frac{N_{t_0}}{N} I(t_0) - \frac{N_{t_L}}{N} I(t_L) - \frac{N_{t_R}}{N} I(t_R)$$

$$s_t = arg\,\max_s \Delta I(s)$$

Decision Tree Impurity

- Algorithm is generic and only relies on the impurity function
- Various possible choices for categorization and regression

Categorization

$$\Phi_{gini} = 1 - p^2 - q^2$$

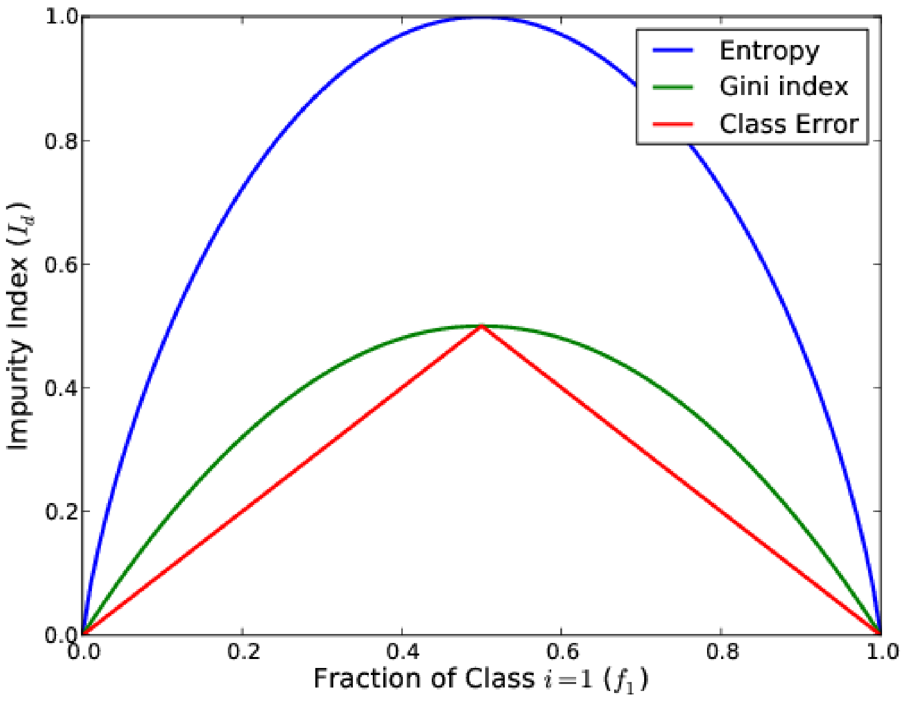
$$\Phi_{xe} = -\left(p\log(p) + q\log(q)\right)$$

$$\Phi_E = 1 - max(p,q)$$

Regression

$$\Phi_{SE}(t) \equiv \underset{t}{var}(y_i)^2 = \frac{1}{N_t}\sum_{i\in t}\left(y_i - c_l\right)^2$$

$$with\ c_l = \sum_{i\in t} y_i$$

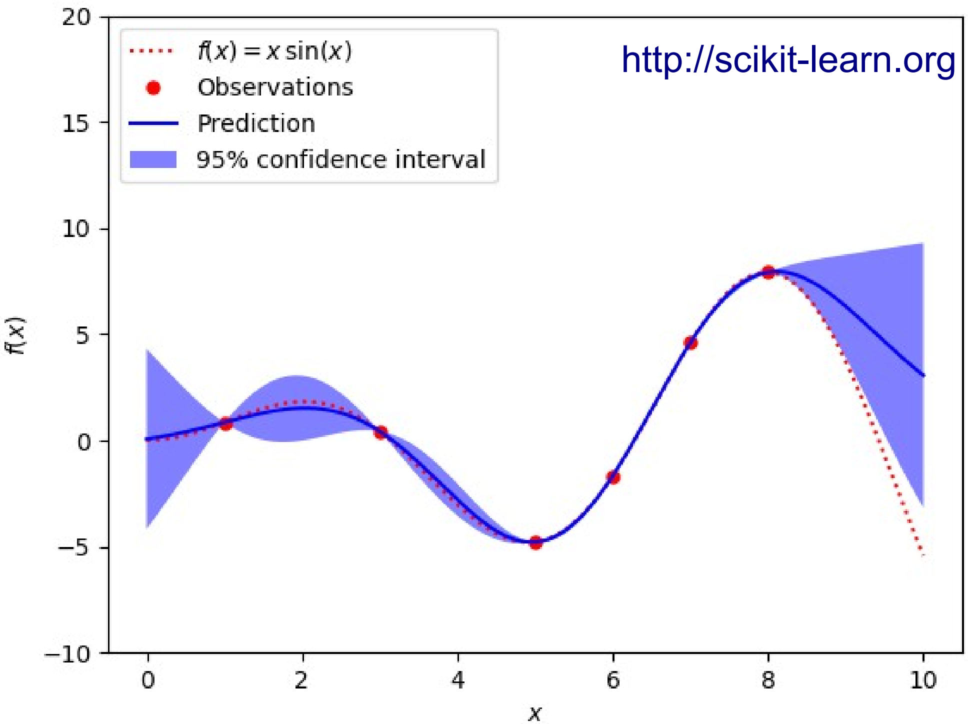


Gaussian Processes

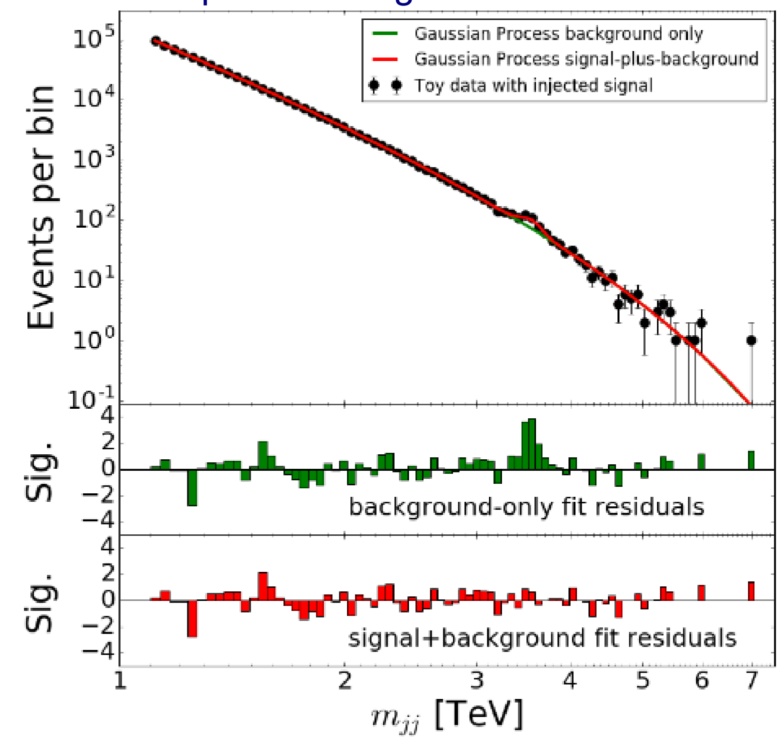
A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- In 2-d, it can be visualized as a family of functions *y,σ = f(x)* where one defines $m(x)=E[f(x)]$ and $k(x,x')=V(f(x),f(x'))$
  - Setting the expectation and how values correlate as a function of the inputs
- For a given dataset and a set of m and k, the interesting GP is the one that passes through the data, providing a prediction/extrapolation to unseen data, respecting the same observed correlation in the data
- In practice, m=0 and the choice of k is crucial
  - Squared exponential (SE) is common usage $k_{SE}(x,x')=\exp(-\frac{1}{2}(x-x')^2)$
- With k depending on a set of parameters θ, one can regress on θ to obtain the GP most consistent with data

GP in practice



https://arxiv.org/abs/1709.05681



- Modeling smooth distribution
- Computation complexity grows cubic with dataset size (can scale better with sparse approximation)
- Use in model hyper-optimization scheme to model the loss function landscape
- Categorization ...

Support Vector Machine

- SVM aim at finding an hyper-plane that separates maximally two populations
- Economical method, yet robust and performant
- Solving the primal problem

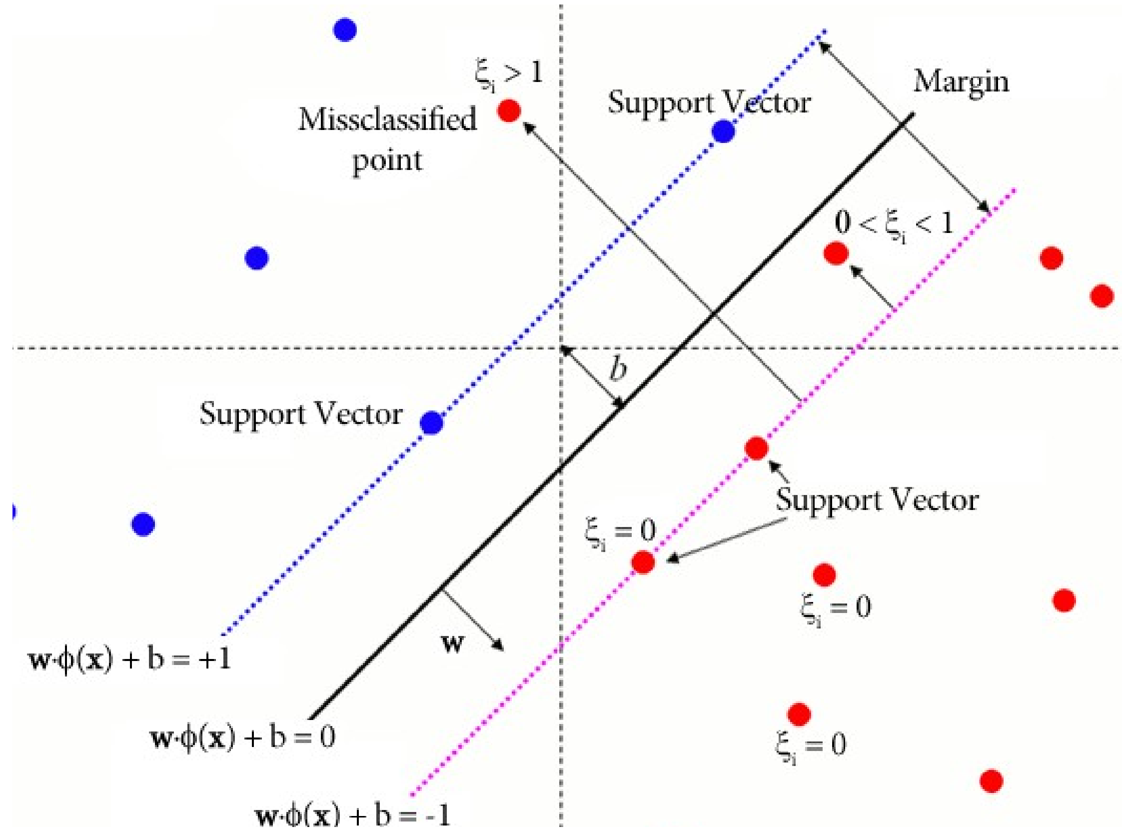
$$\min_{\xi,\omega,b} \|\omega\|^2 + C\sum_i \xi_i$$

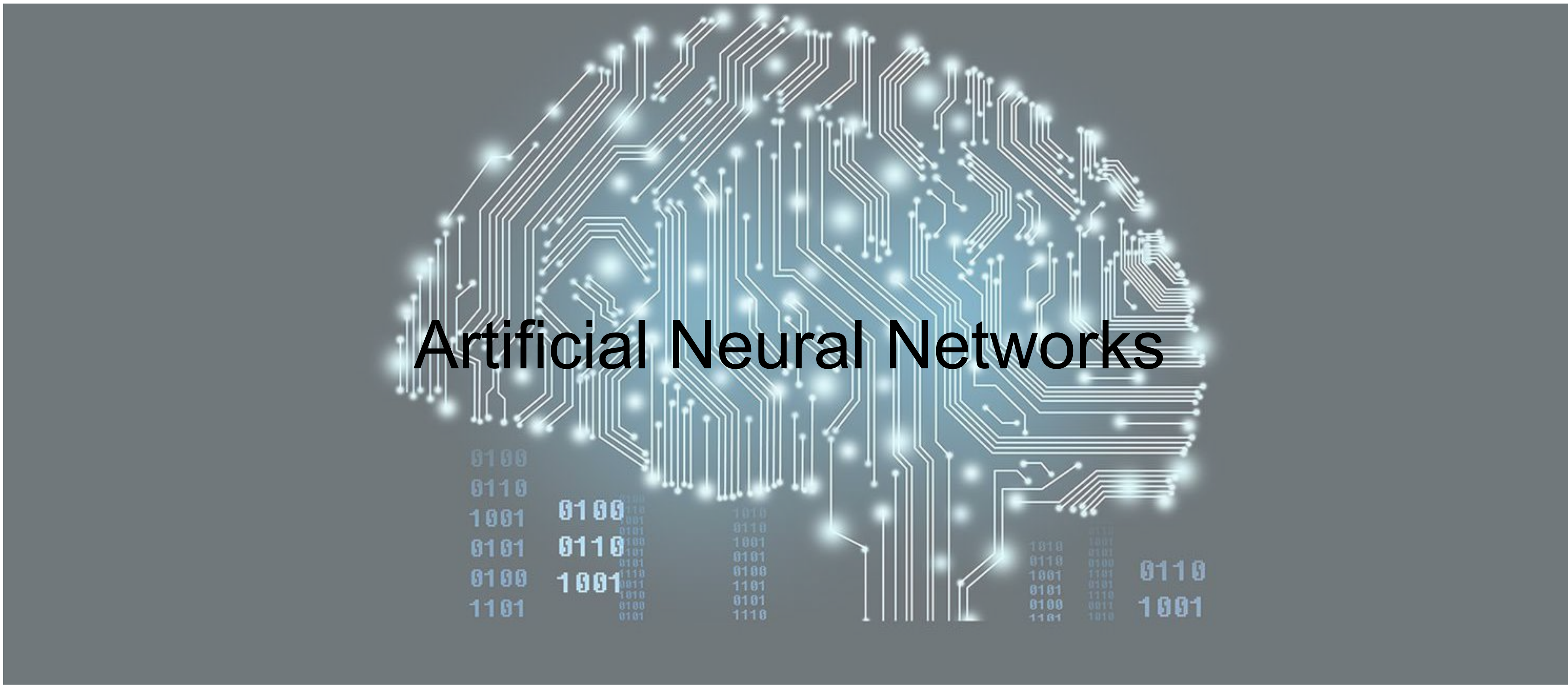
$$s.t.\quad \xi_i \geqslant 0 \quad \text{and} \quad y_i(\omega^T x_i + b) \geqslant 1 - \xi_i$$

- Equivalent to solving in dual space

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{jk} \alpha_j \alpha_k y_j y_k (x_j^T x_k)$$

$$s.t.\quad 0 \leqslant \alpha_i \leqslant C \quad \text{and} \quad \sum_i \alpha_i y_i = 0$$



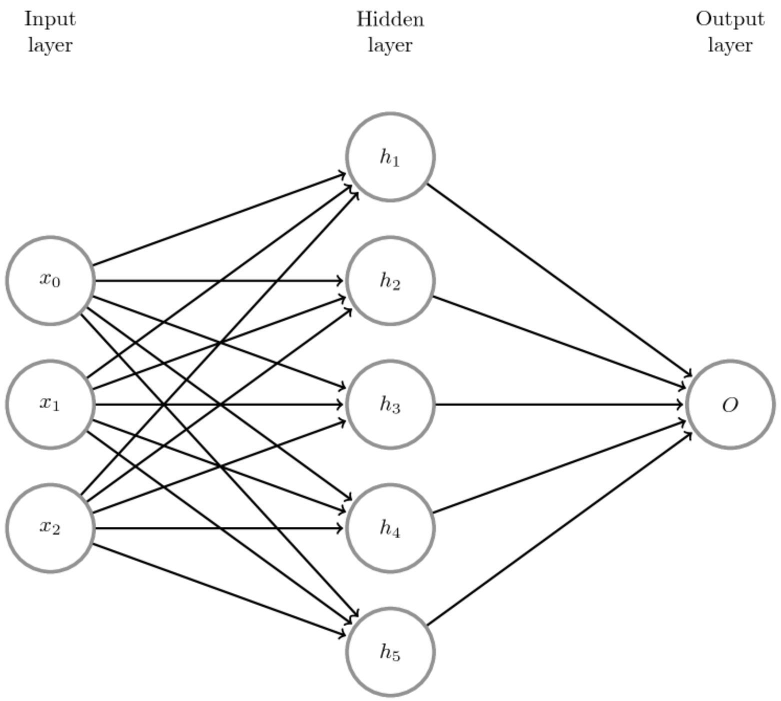


Artificial Neural Networks

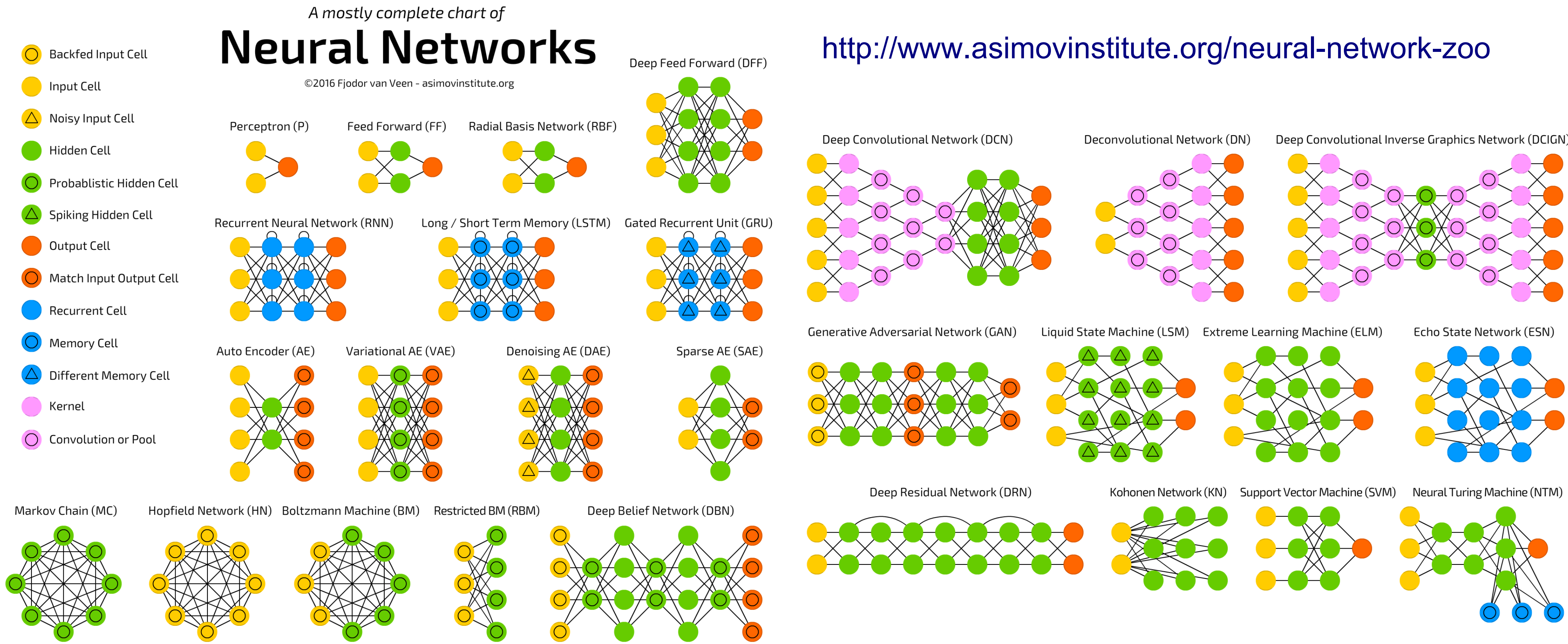
Artificial Neural Network

- **Biology inspired** analytical model, but **not bio-mimetic**
- Booming in recent decade thanks to large dataset, increased computational power and theoretical novelties
- Origin tied to logistic regression with change of data representation
- Part of any "deep learning" model nowadays
- Usually large number of parameters trained with stochastic gradient descent

$$h=\phi(Ux+v)$$
$$o(x)=\omega^T h+b$$
$$p_i\equiv p(y=1|x)\equiv\sigma(o(x))=\frac{1}{1+e^{-o(x)}}$$
$$loss_{XE}=-\sum_i y_i \ln(p_i)+(1-y_i)\ln(1-p_i)$$



Neural Net Architectures



http://www.asimovinstitute.org/neural-network-zoo

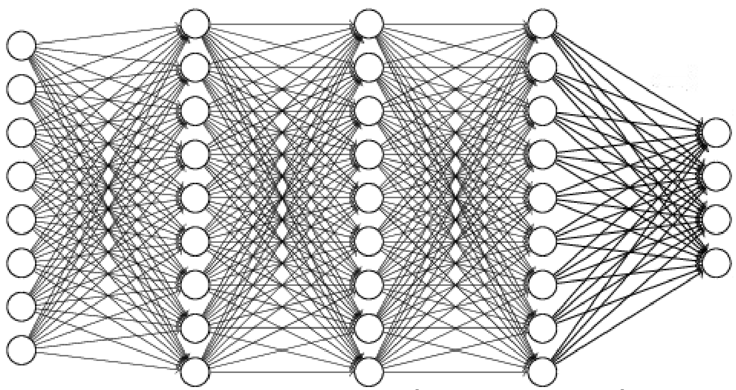
- Does not cover it all : capsule, densenet, attention, graph network, ...

Curse of Dimensionality

- Fully connected layers require a large number of parameters

$$N^l_{par} = N^l_{input} \times N^l_{node} + N^l_{node}$$

- Lots of a capacity in this kind of models
- Convergence of models with millions of parameters can be hard numerically
- Hashing and pruning studies showed lots of redundancies : not all weights are necessary
- Weight sharing helps reducing dimensionality



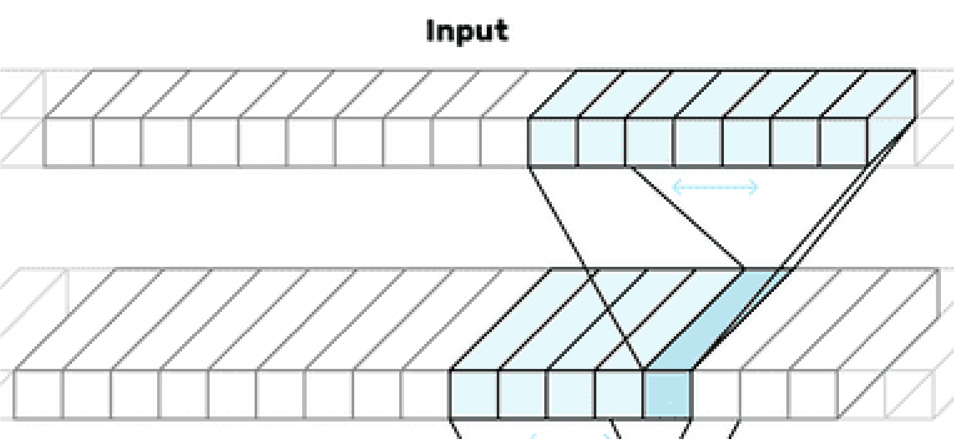
301 parameters (8-9-9-9-4)
~5B for 200x200 pixels image

Convolutional Layer

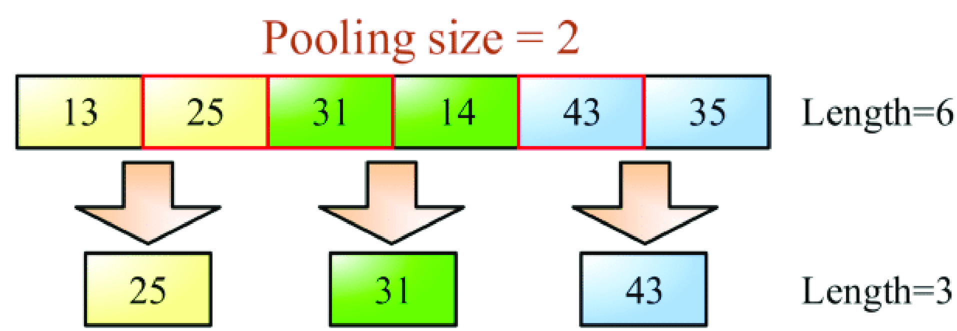
- Fully connected layers require a large number of parameters
- Weights sharing applied as stencil code

$$N^l_{par} = (N^l_{input} / S^l_{kernel}) \times (S^l_{kernel} \times N^l_{filter} + N^l_{filter})$$

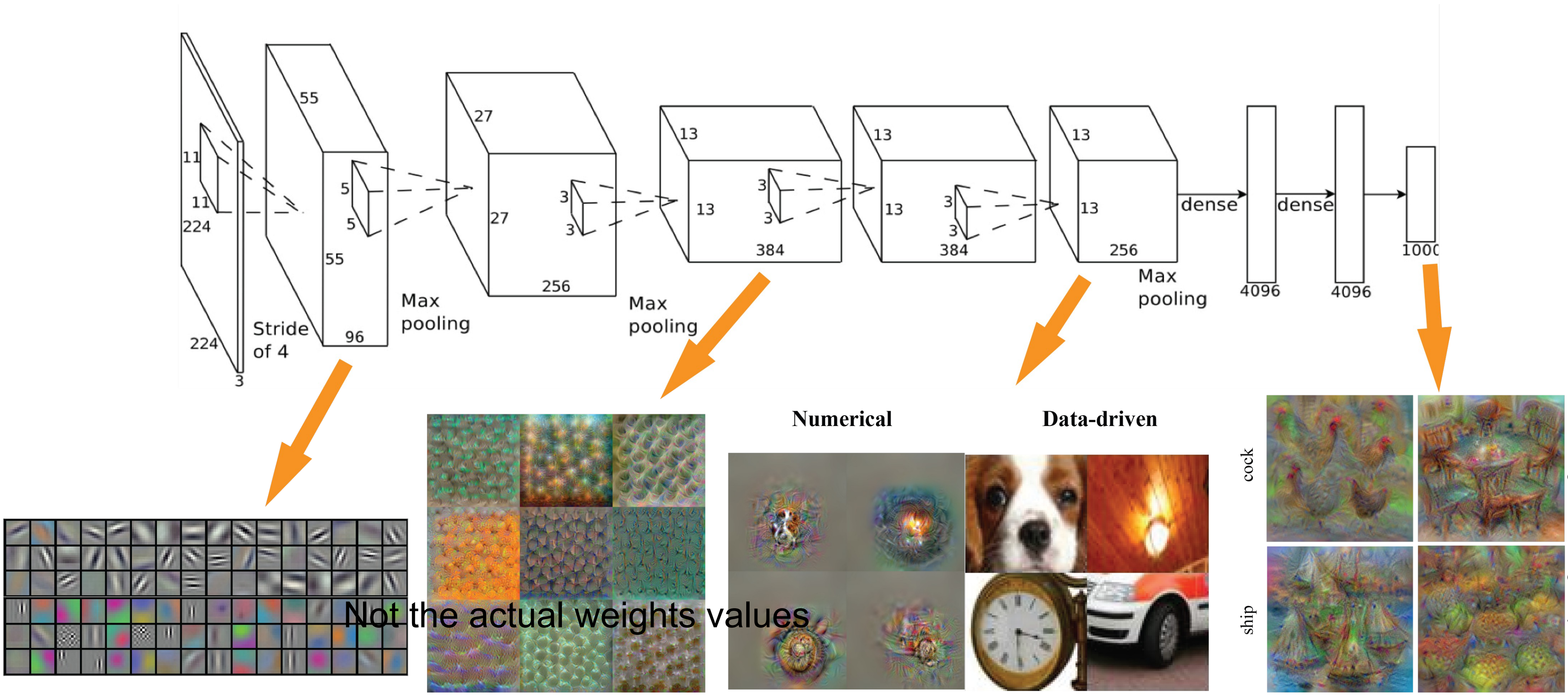
- Number of parameters are dramatically reduced
- Available in 2D and 3D
- Various ways of handling multiple channels (r,g,b)
- Often associated with maxpooling for dimensionality reduction



1D convolution, kernel size = 7, 1 filter



Stacked Convolution

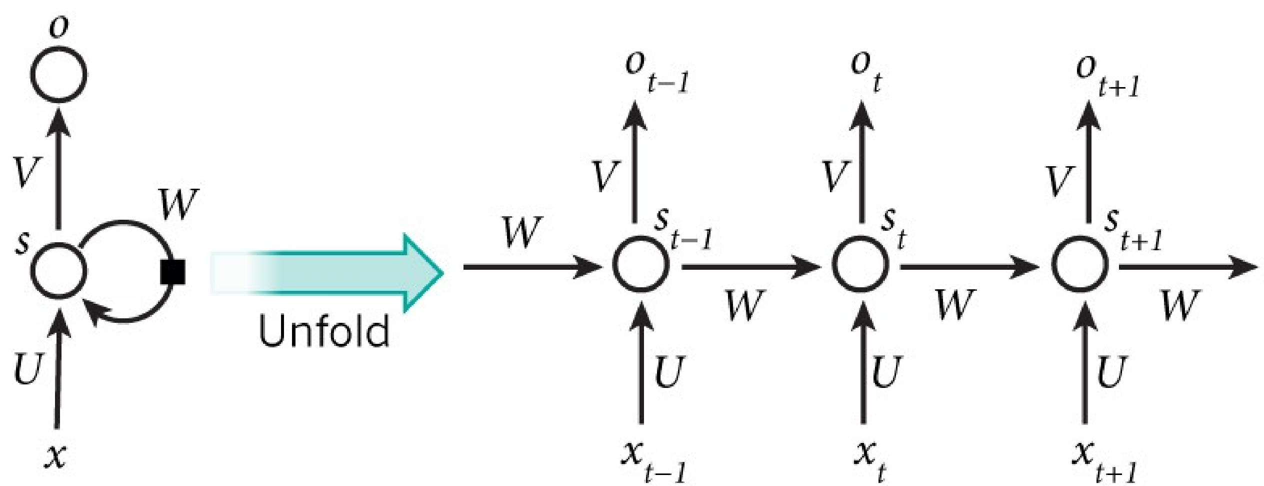


Conv 1: Edge+Blob
Conv 3: Texture
Conv 5: Object Parts
Fc8: Object Classes

- Early convolution layer capture **local information**
- Late convolution layer capture **global information**

Recurrent Neural Network

- Sequential (text) of temporal (voice) data contains information in their structure
- Model that can naturally accommodate for variable sized input
- Characterized by an hidden state carried over steps
- Concern over natural ordering

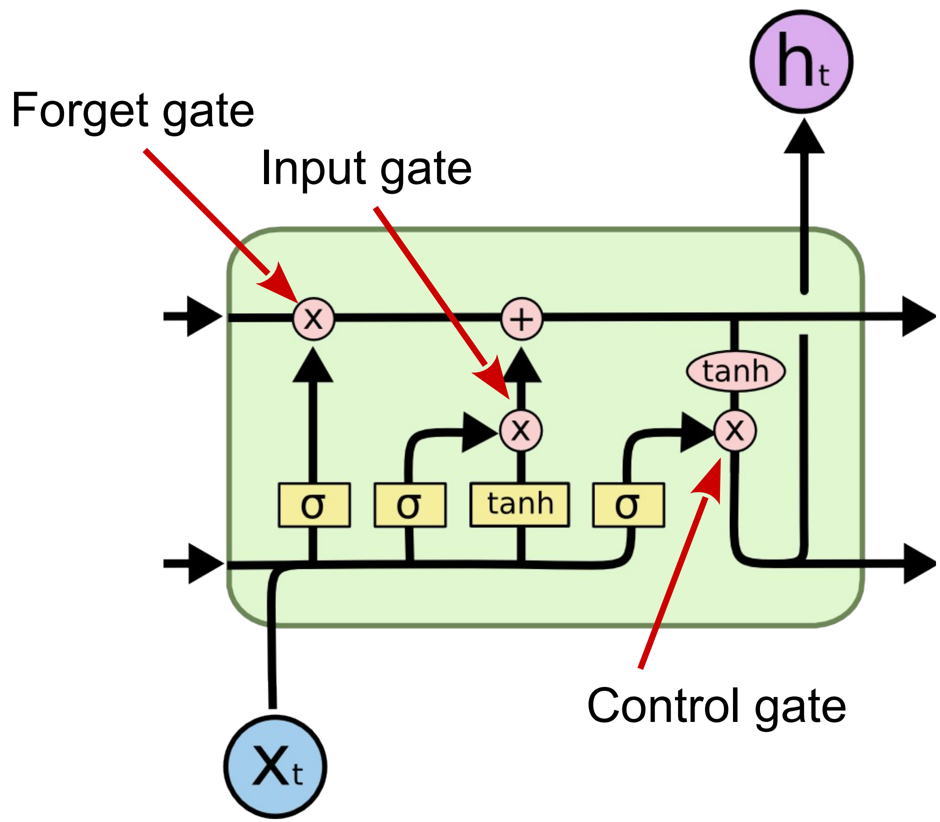


$$s_t = \tanh(U x_t + W s_{t-1} + b_r)$$
$$o_t = \sigma(V s_t + b_o)$$

Long Short Term Memory Cell

- LSTM revolutionized text processing in the late 90s
- Carries around a cell state (C_t) and hidden state (h_t)
- Computationally expensive

$$c_t = [h_{t-1}, x_t]$$
$$f_t = \sigma(W_f c_t + b_f)$$
$$i_t = \sigma(W_i c_t + b_i)$$
$$\tilde{C} = \tanh(W_C c_t + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o c_t + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

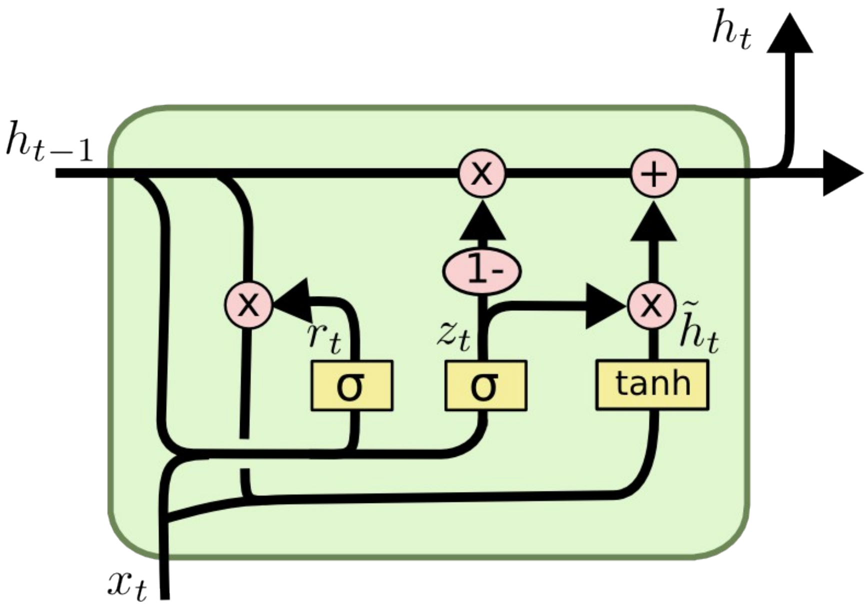


http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit

- GRU simplifies the computation from LSTM
- Only hidden state

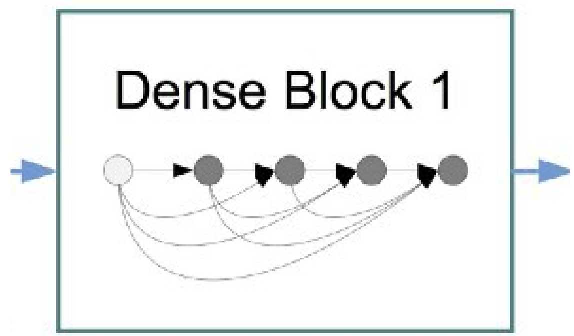
$$c_t = [h_{t-1}, x_t]$$
$$z_t = \sigma(W_z c_t + b_z)$$
$$r_t = \sigma(W_r c_t + b_r)$$
$$\tilde{h}_t = \tanh(W_h [r_t * h_{t-1}, x_t] + b_h)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$



Skip Connections

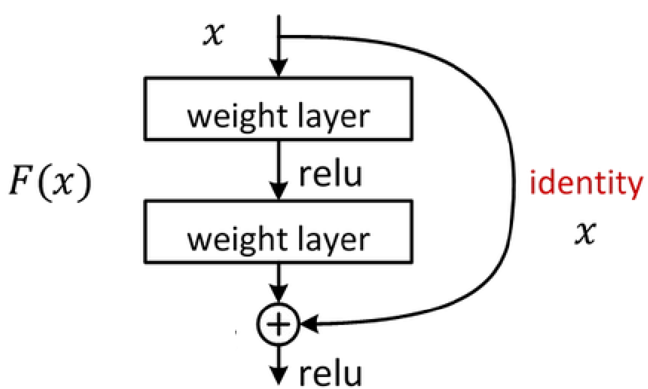
- Stacked convolution layers distill information at consecutive scales
- Several ways of conserving the information from previous layers

densenet 1608.06993



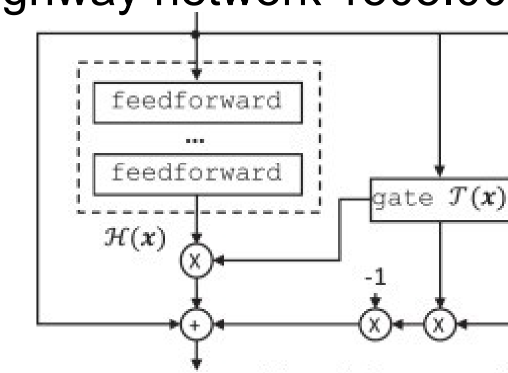
$$f_l(i_l) \rightarrow f_l\left(\bigcup_{k \leqslant l} (i_k)\right)$$

resnet 1512.03385



$$f_l(i_l) \rightarrow f_l(i_l) + i_l$$

Highway network 1505.00387

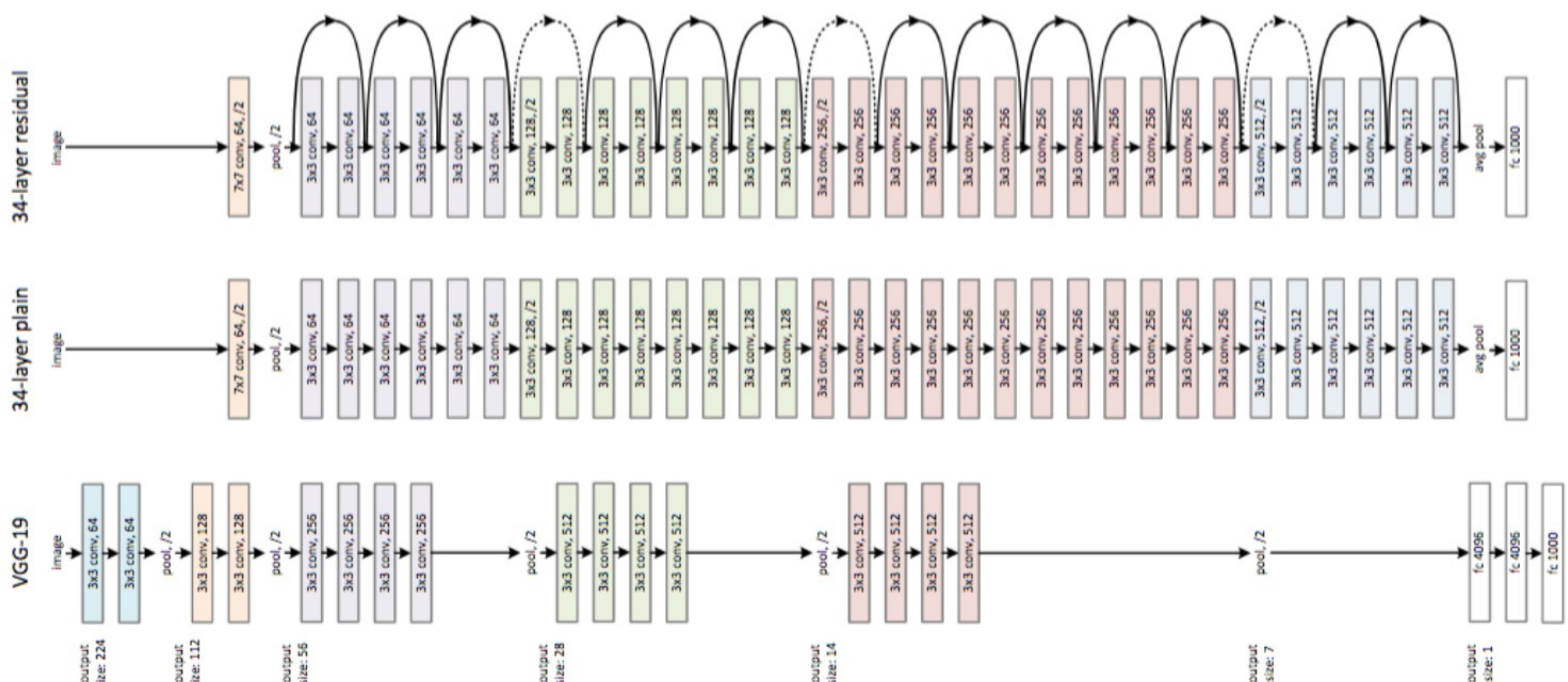


$$f_l(i_l) \rightarrow f_l(i_l) \cdot T(i_l) + i_l \cdot (1 - T(i_l))$$

Residual Connection

- Stacked convolution layers distill information at consecutive scales
- Residual connection carries the input other to the output, dimensionality allowing

$$f_l(i_l) \rightarrow f_l(i_l) + i_l$$

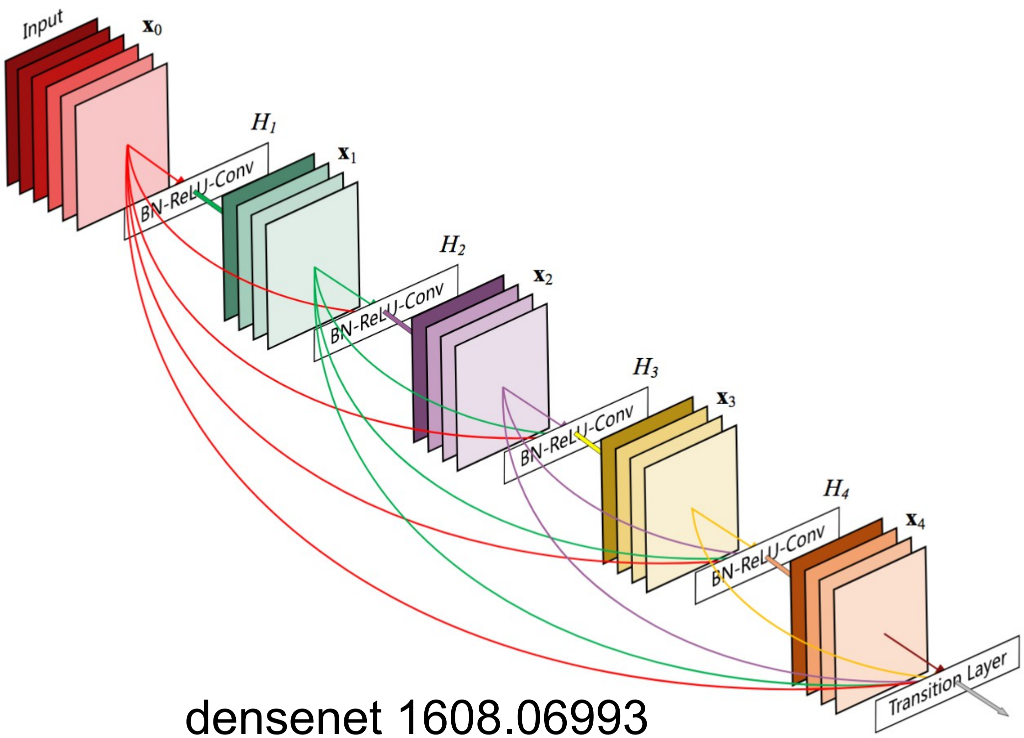


resnet 1512.03385

Dense Connection

- Stacked convolution layers distill information at consecutive scales
- dense-net provides the concatenation of all previous layer input to the next layer

$$f_l(i_l) \rightarrow f_l\left(\bigcup_{k \leqslant l}(i_k)\right)$$

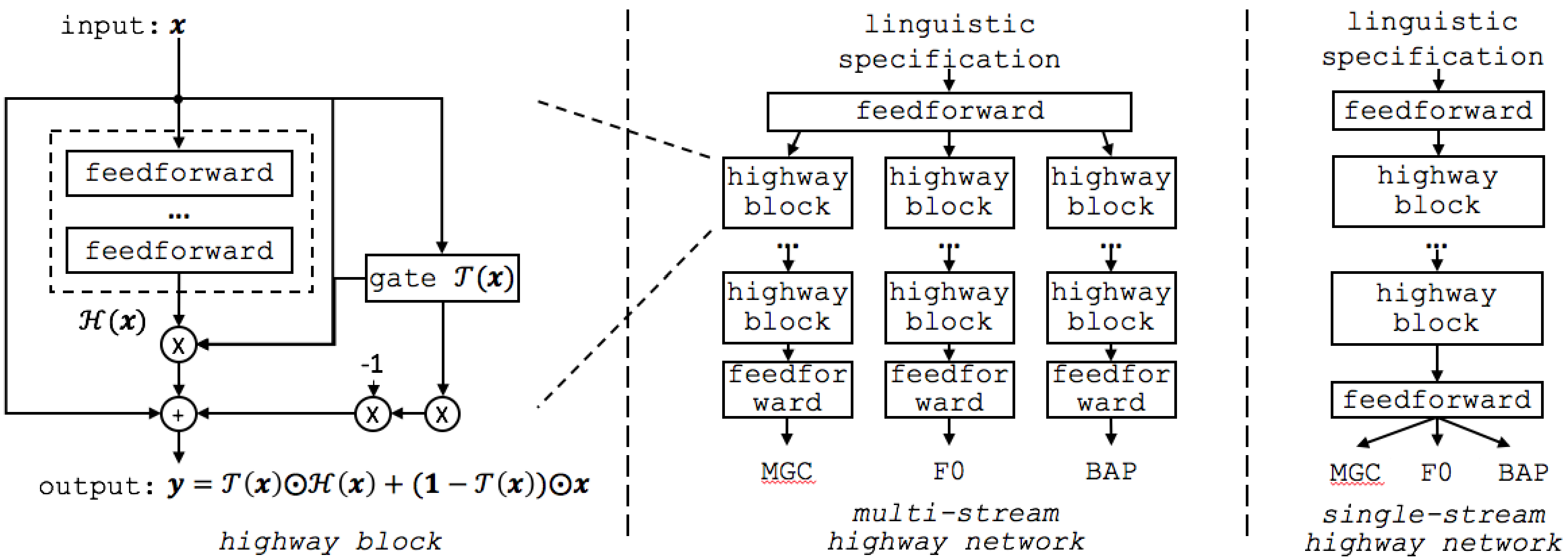


densenet 1608.06993

Highway Connection

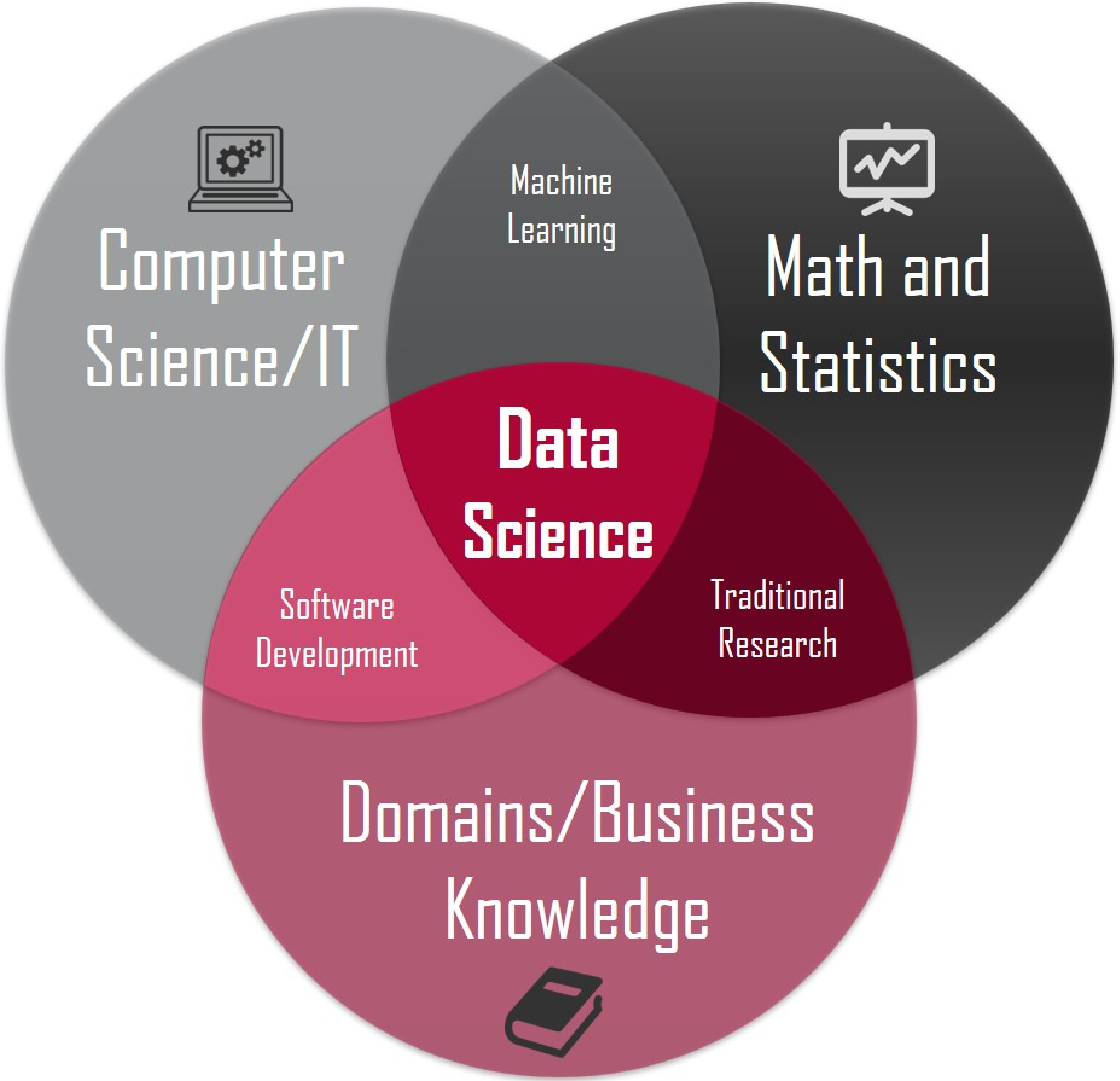
- Stacked convolution layers distill information at consecutive scales
- Highway network controls how much information from previous layer needs to move forward as input to the next

Highway network 1505.00387



Summary/Recap Lecture 2

- Unsupervised learning for clustering, dimensionality reduction, density estimation and generative models
- Supervised learning for regression and classification
- Artificial neural network are in rapid evolution. Methods providing lots of flexibility and at the forefront of performance on many complex tasks


Computer Science/IT
Machine Learning
Math and Statistics
Data Science
Software Development
Traditional Research
Domains/Business Knowledge