Why Representation Learning ?

- ML Methods performance heavily depends on Data Representation.
- Representation Learning: Learning Representations of the data that make it easier to extract useful information when building models.
- Fields in which Representation Learning has had success: Speech Recognition and Signal Processing, Object Recognition, Natural Language Processing, Multi-Task and Transfer Learning, Domain Adaptation.

What makes a Representation GOOD ?

Priors useful for a learning machine to solve AI tasks :

- Smoothness: Property of function which says if x==y, then f(x) == f(y), struggles to get around curse of Dimensionality.
- Multiple Explanatory Factors: Disentangle underlying factors of variation in Data. Based on idea of Distributed Representations.
- Hierarchical Organization of explanatory factors: Abstract concepts are found higher defined in terms of less abstract ones. Exploited with Deep Representations.
- Semi - Supervised Learning & Shared factors across tasks: : Sharing of Statistical Strength between the unsupervised and supervised learning tasks.
- Manifolds: Probability mass concentrates near regions having a smaller dimensionality, exploited in Representation Learning Algorithms.
- Natural Clustering: Local variations on Manifold tend to preserve the value of a category
- Temporal and Spatial Coherence: Consecutive / Spatially near by observations tend to have same value of categorical concepts.
- Sparisity: most of the extracted features are insensitive to small variations.
- Simplicity of Factor Dependencies: In good high level representations, factors are related to each other through simple & linear dependencies.

These priors help in discovering and disentangling some of the underlying factors of variation that the data may reveal.

- Smoothness and the Curse of Dimensionality: Smooth functions cannot capture enough of the complexity of interest unless provided with the appropriate feature space, thus comes the importance of Representation Learning.
- Distributed Representations: Good representations are expressive, a reasonably - sized learned representations can capture a huge number of

possible input configurations. Sparse or Distributed Representations can represent exponential number of inputs as function of the number of parameters, this is in comparison to traditional ones, where it is linear as function of number of parameters.

- Depth and Abstraction: Deep Architectures promotes re-use of features and leads to more abstract features at higher layers of representation. Reuse explains power of distributed representations. Ways to reuse grows exponential with respect to depth.Abstract concepts are generally invariant to most local changes of input and also have greater predictive power.
- Disentangling Factors of Variation: A good representation is one that dientagles the underlying factors of variation. Different explanatory factors of data tend to change independently of each other in the input distribution and only a few at a time tend to change. Disentangle as many factors as possible, discarding as little information about the data as is practical.
- Building Deep Representations:
    - Deep Features can be got from either of these methods:
        - Greedy layerwise unsupervised pre-training
        - Greedy layerwise supervised pre-training
    - Features got from these methods can be used in Machine Learning Predictors such as SVM or even in Deep Belief Networks.
- Single Layer Leaning Modules : Deep Learning model can describe a
- probabilistic graphical models while the other in neural networks.
- Major Developments in single-layer training modules used in Representation Learning from a deep learning perspective are:
    - The probabilistic models (both directed and undirected)
    - Reconstruction - based algorithms (neural networks)
    - Geometrically motivated manifold-learning approach
- Principal Component Analysis : An unsupervised single-layer representation learning algorithm spanning all three views.
- Probabilistic Models: Recover a parsimonious set of latent random variables that describe a distribution over the observed data. Feature values are got while inferring the posterior probability.
    - Directed Graphical Models [Based on Explaining Away property, i.e, A Priori Independent causes of an event can become non-independent given the observation of the event]
        - Sparse Coding

        - Spike and Slab Sparse Coding
    - Undirected graphical Models [Markov random Fields]
        - Restricted Boltzmann Machines
            - Gaussian RBMs
            - mcRBM
            - mPOT model
            - ssRBM
        - RBM parameter estimation can be done by:
            - Contrastive Divergence Stochastic Maximum Likelihood
            - Pseudo-likelihood
            - Ratio-matching
- Directly Learning a parametric map from input to representation [NN]:
    - Auto-Encoders
    - Regularized Auto-Encoders:
        - Sparse Auto-Encoders
        - Denoising Auto-Encoders
        - Contractive Auto-Encoders
        - Predictive Sparse Decomposition
- Representation Learning as Manifold Learning:
    - Learning a parametric mapping based on a neighborhood graph
    - Learning to represent non-linear manifolds
    - Leveraging the modeled tangent spaces
- Connections between probabilistic and direct encoding models: They have more characteristics in common than separating them. Computational Graph of an autoencoder is similar to that of a deep Boltzmann machine. In general, computing representation in the neural network model corresponds exactly to the computation graph that corresponds to inference in the probabilistic model.
- Instead of single-layer training we can also do global training of deep models. Joint Training of Deep Boltzmann Machines involves Mean-field approximate inference.
- In-variance can be built by using Convolutions and pooling and generating transformed examples using other methods.