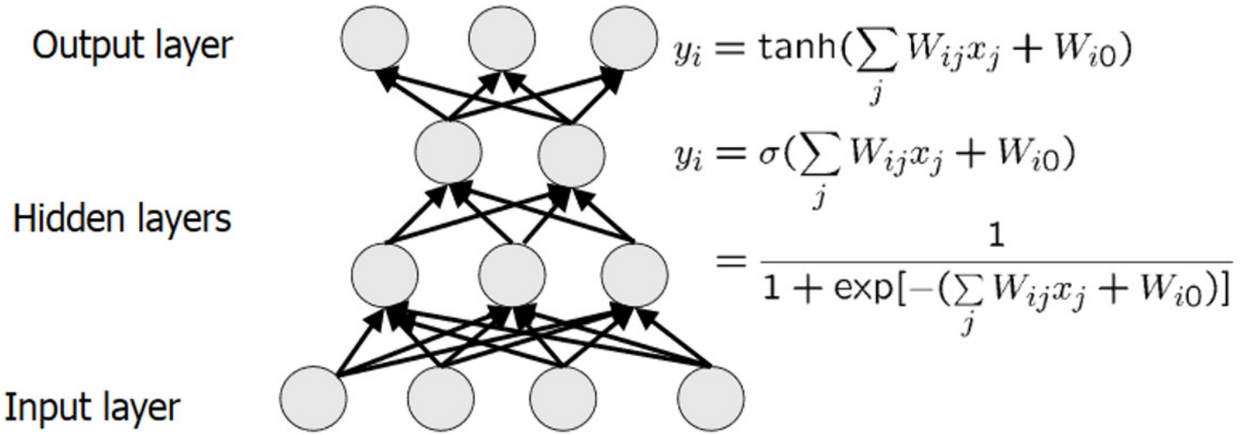
DLI Accelerated Data Science Teaching Kit

Module 16.4 - Advanced Neural Networks

Breakthrough of Machine Learning/Artificial Intelligence

Deep Learning

- Advanced Neural Networks
- Machine learning algorithms based on learning multiple levels of representation/abstraction
- Deep architectures consists of multiple levels of non-linear operations.

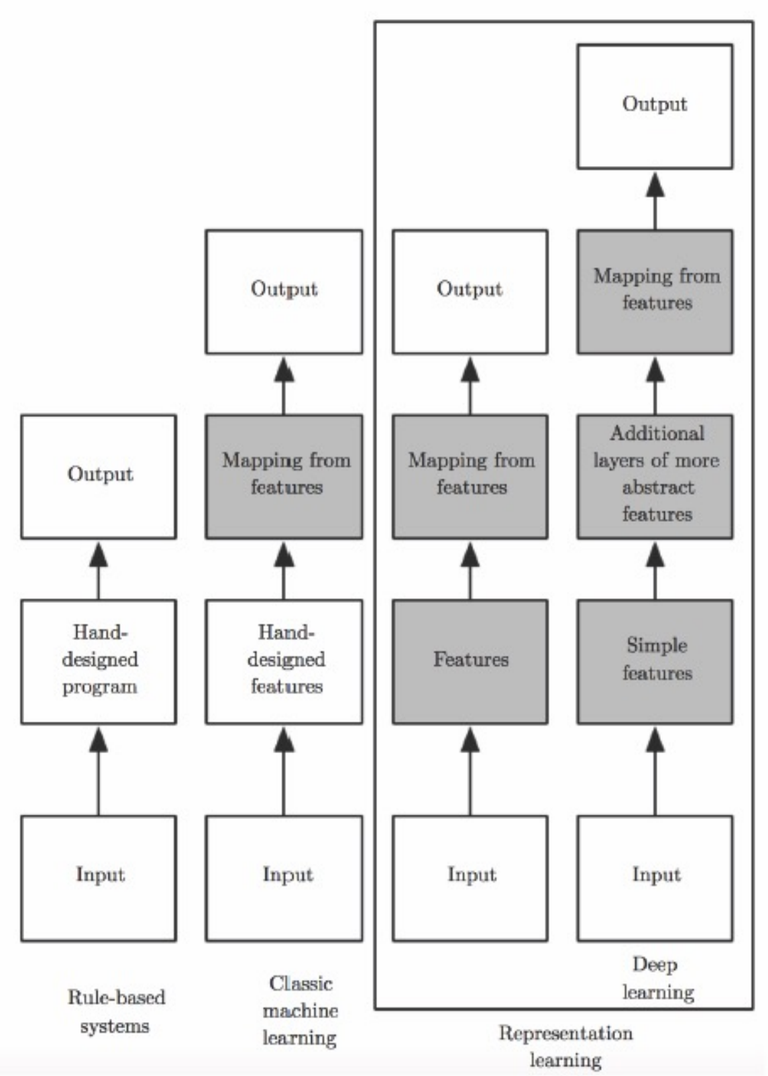


- Amazing improvements in error rate
  - Object detection
  - Object recognition
  - Speech recognition
  - Machine translation
  - Natural language understanding

Comparison Between Traditional ML and Deep Learning

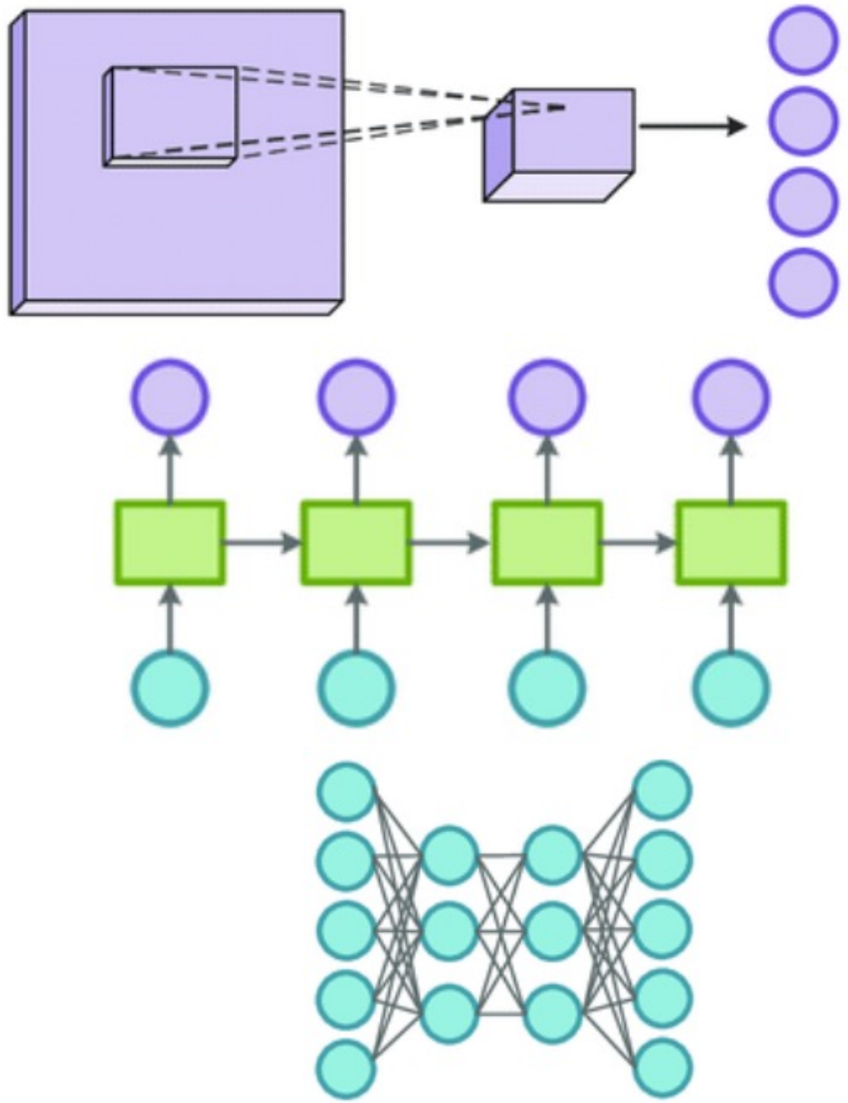
- Traditional ML algorithms depend heavily on representation of given data
  - Feature Extraction and Selection
  - Feature Relations: Linear vs Non-linear

- Learn not only the mapping from representation to output but the representation itself
  - Learned representations often provide much better results than hand-coded representations



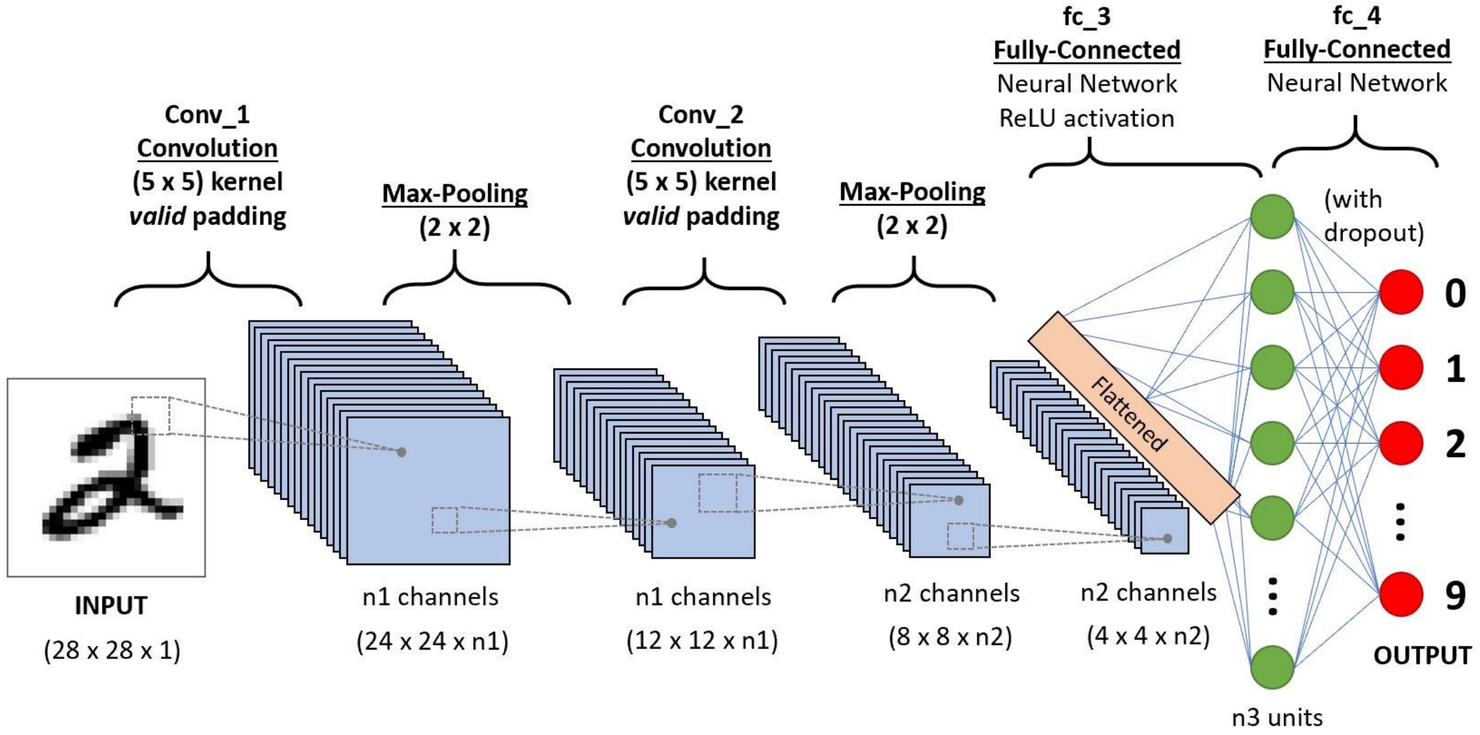
Deep Learning Models

- Convolutional Neural Networks
- Recurrent Neural Networks
- Autoencoder



Convolutional Neural Networks

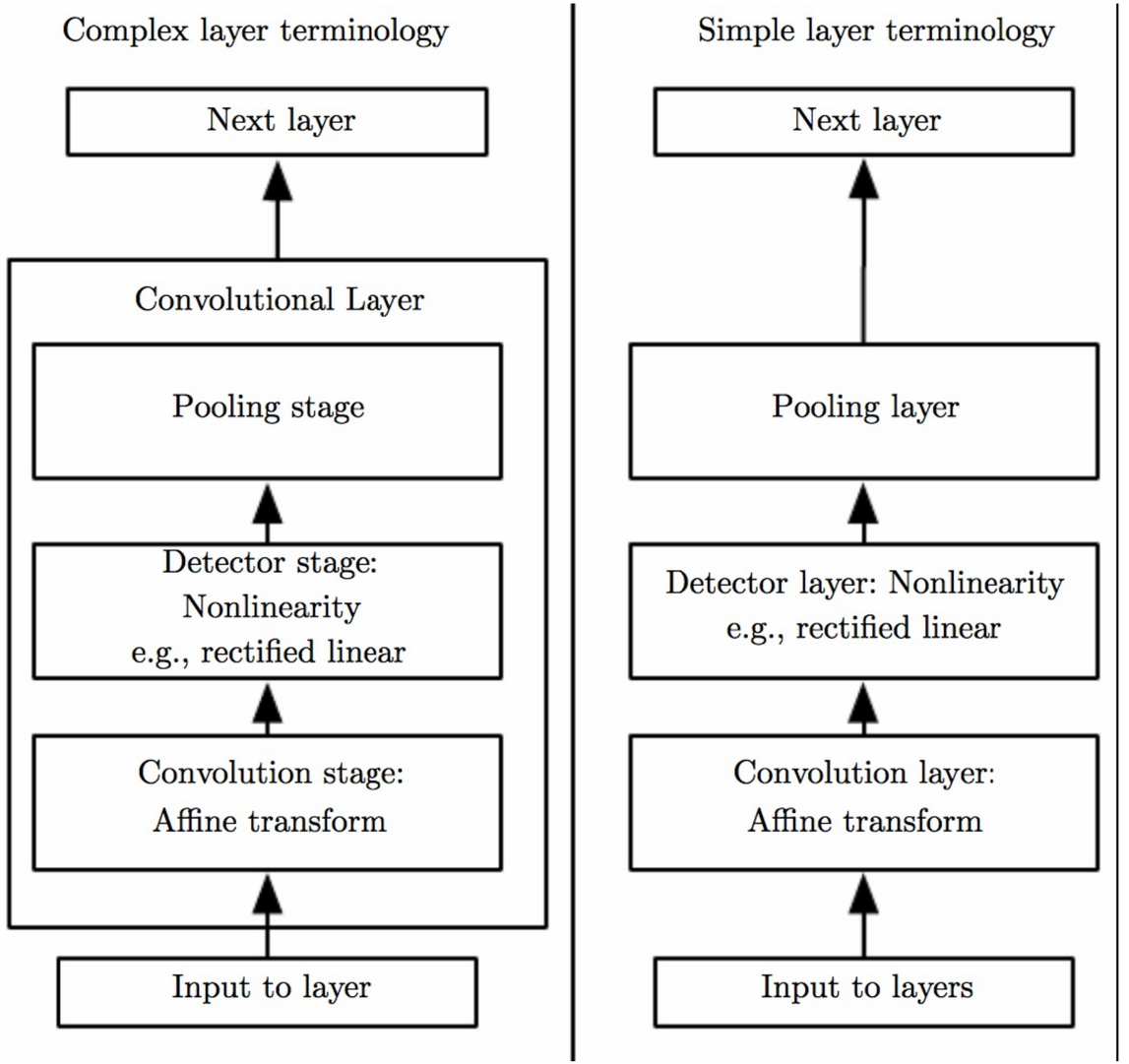
- Motivation
  - Reducing model complexity by removing redundancy connections between neurons
  - Improving effectiveness and efficiencies of feature extraction by sharing parameters
- General Architecture
  - Digital Recognition
- Application
  - Time series data: 1-D grid taking samples at discrete intervals
  - Image Data: 2-D grid of pixels



Source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Components of a typical CNN layer

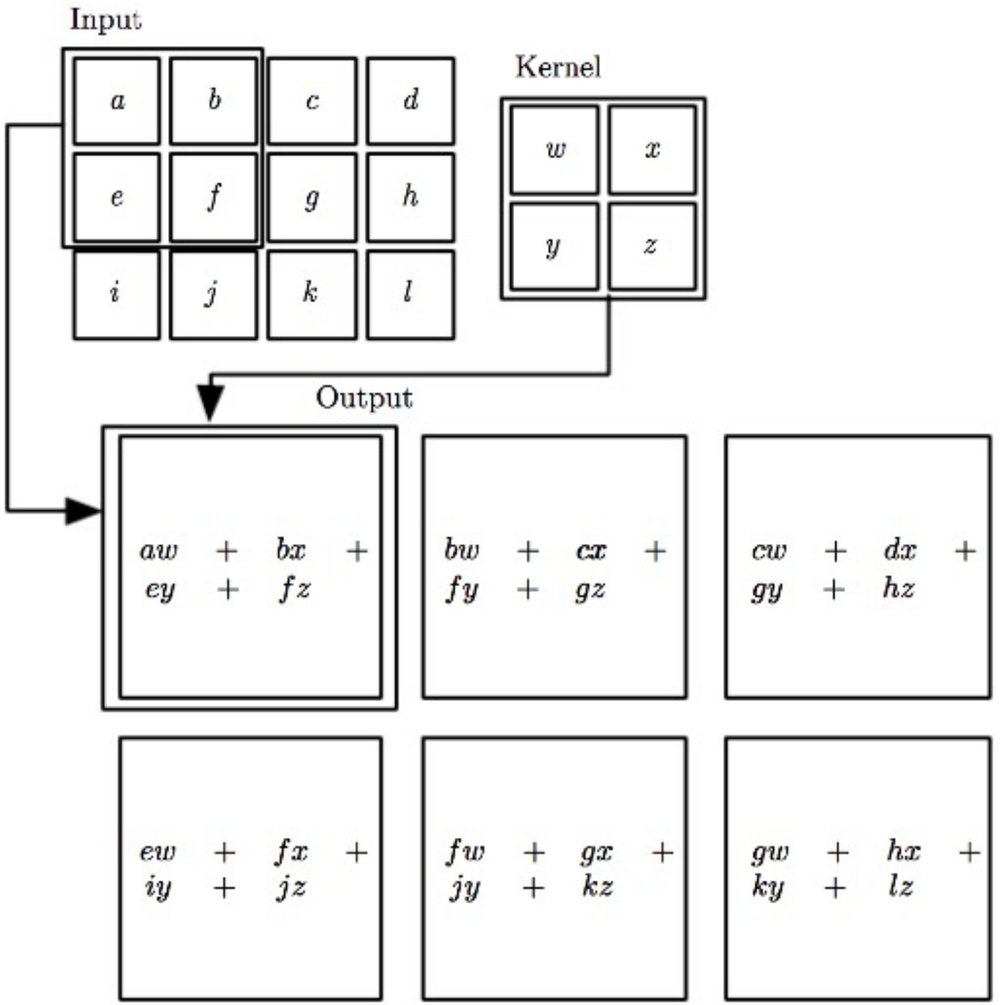
- Two terminologies for describing layers
  - Left: convolutional net is viewed as a small no. of relatively complex layers, each layer having many stages
  - Right: CNN is viewed as a larger no. of simple layers



Convolution Operation for CNN

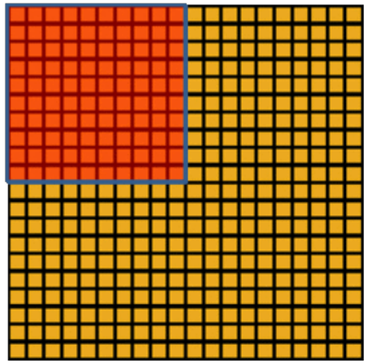
- In convolution network terminology the first function x is referred to as the input, the second function w is referred to as the kernel
- The output s is referred to as the feature map

$$f = x \times w = \sum_{i=1}^{m} (x_i \times w_i)$$

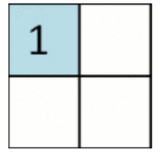


Pooling Operation for CNN

- Typical layer of a CNN consists of three stages
  - Stage 1: perform several convolutions in parallel to produce a set of linear sum
  - Stage 2 (Detector): each linear sum is run through a nonlinear activation function such as ReLU
  - Stage 3 (Pooling): Use a pooling function to modify output of the layer further
- Pooling significantly reduce the number of features.



Convolved feature

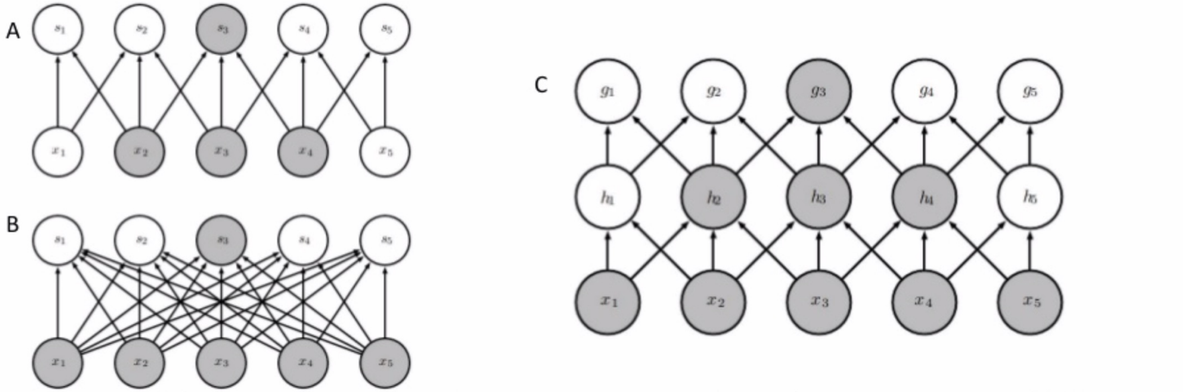


Pooled feature

Benefits

- Convolution leverages three important ideas to improve ML systems:
  - Sparse connections
    - A refers to local connections with convolutional operation. One output neuron connected to three inputs (gets three inputs).
    - In C subfigure, with depth, the receptive field becomes larger. It means g3 will directly obtain inputs not only from middle layer, but also from the bottom layer indirectly.
    - B refers fully connections in the standard neural networks.

Comparison with MLP
→ Local Connections

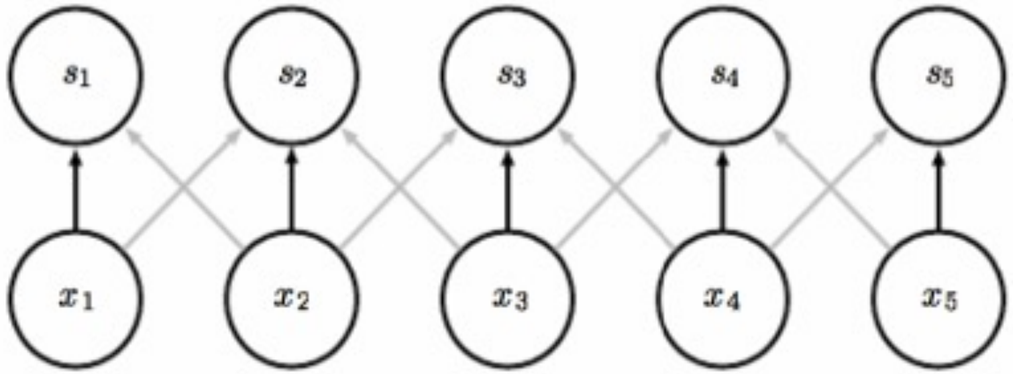


A, with convolution kernel size = 3, the activated neurons are only affected by local neurons , unlike in B, where there are full connections; however, with depth, the receptive field can expand, and get global connections to neurons in lower layer.

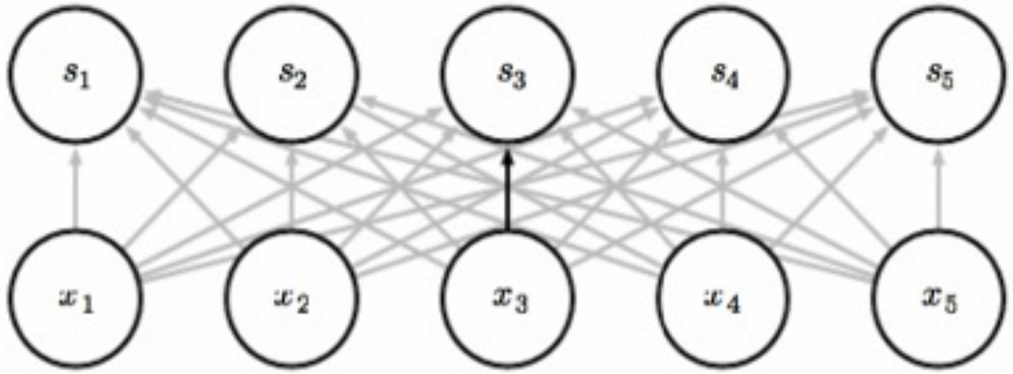
Benefits

- Convolution leverages three important ideas to improve ML systems:
  - Parameter Sharing
    - Using the same parameter for more than one function in a model
    - Black arrows: connections that use a particular parameter in two different models
    - Top: use of the central element
    - Bottom: no parameter sharing

Top



Bottom



Convolution and Pooling can Cause Under-fitting

- If a task relies on preserving precise spatial information, then using pooling on all features can increase the training error.
- Some Solution
  - Pooling on some channels but not on other channels, in order to get both highly invariant features and features that will not under-fit.
  - When a task involves incorporating information from very distant locations in the input, then the prior imposed by convolution may be inappropriate.

Recurrent Neural Networks

- Recurrent neural networks or RNNs are a family of neural networks for processing sequential data.
  - Natural Language Processing
- Most recurrent networks can also process sequences of variable length.
- Sharing parameters across different parts of a model.

This morning I took a dog for a walk

Given these two words

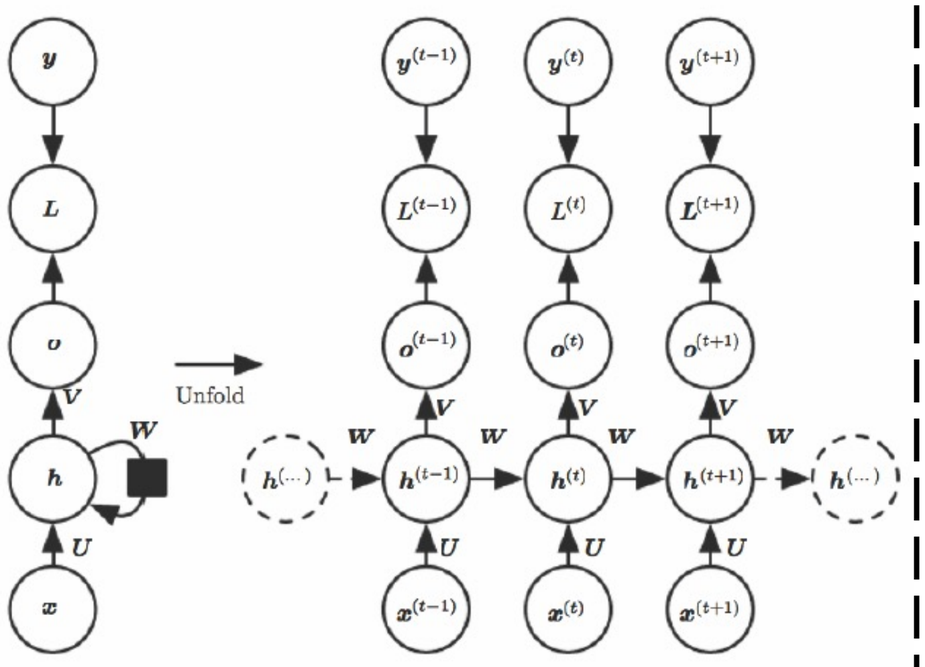
Predict next word

This morning I took a dog for a walk

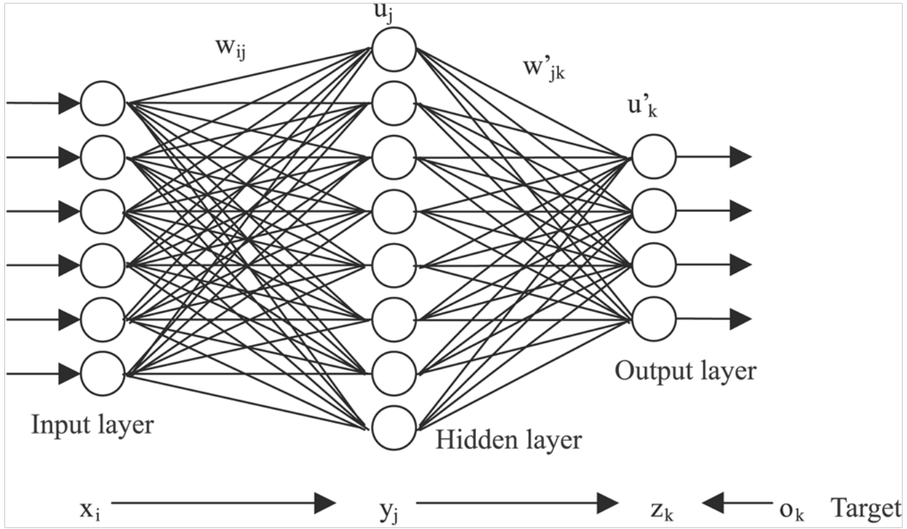
Given these words

Predict next word

Recurrent Neural Networks VS Neural Networks



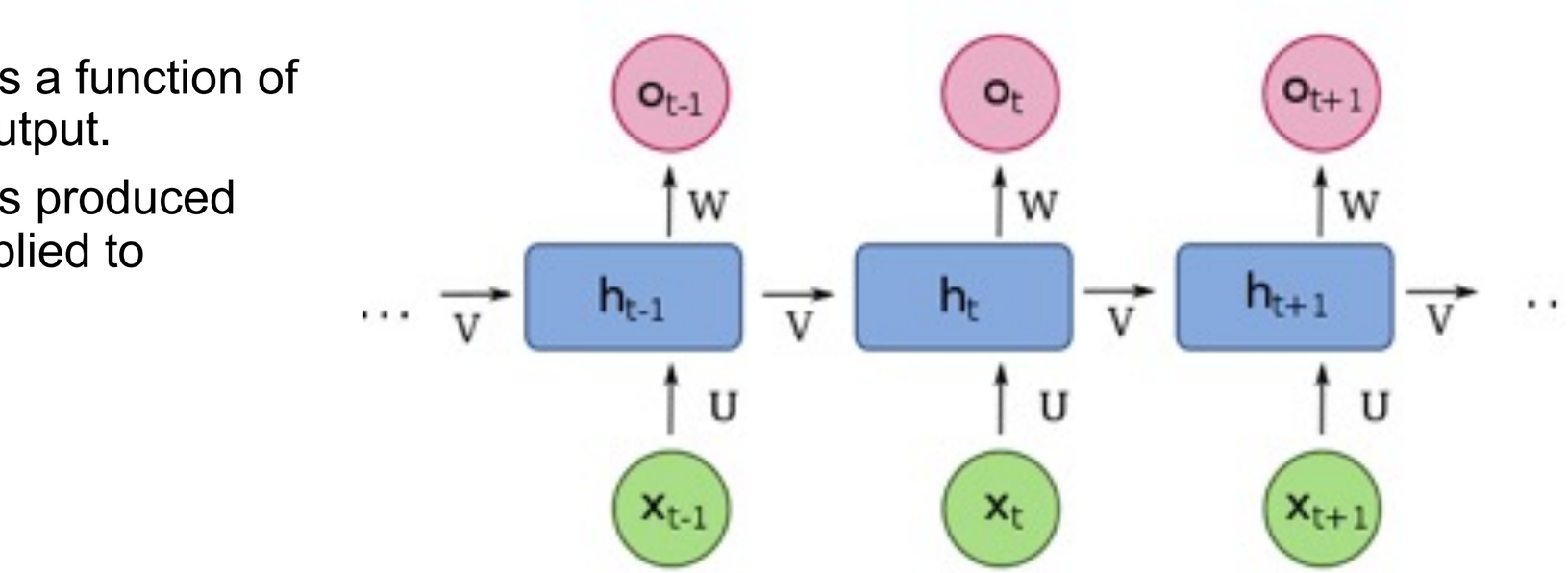
RNNs



NNs

Parameter Sharing

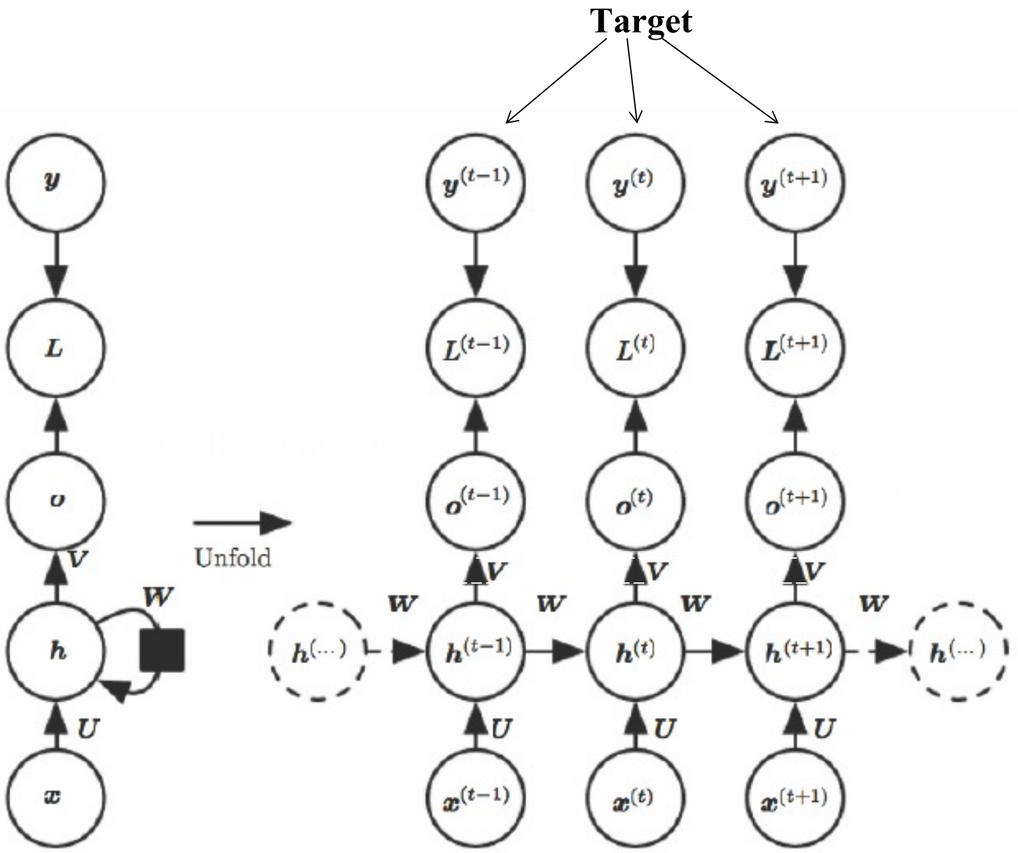
- Each member of the output is a function of the previous members of the output.
- Each member of the output is produced using the same update rule applied to the previous outputs.



Forward Propagation

- Begins with initial specification of h(0)
- Then for each time step from t = 1 to t = τ
we apply the following update equations

$$\boldsymbol{a}^{(t)} = \boldsymbol{b} + W\boldsymbol{h}^{(t-1)} + U\boldsymbol{x}^{(t)}$$
$$\boldsymbol{h}^{(t)} = \tanh(\boldsymbol{a}^{(t)})$$
$$\boldsymbol{o}^{(t)} = \boldsymbol{c} + V\boldsymbol{h}^{(t)}$$
$$\hat{y}^{(t)} = \mathrm{softmax}(\boldsymbol{o}^{(t)})$$



Loss function for a given sequence

- Total loss for a given sequence of x values with a sequence of y values is the sum of the losses over the time steps
  - If L(t) is the negative log-likelihood of y(t) given x(1),..x(t) then

$$L\left(\left\{\boldsymbol{x}^{(1)},..\boldsymbol{x}^{(\tau)}\right\},\left\{\boldsymbol{y}^{(1)},..\boldsymbol{y}^{(\tau)}\right\}\right)=\sum_{t}L^{(t)}$$

$$=-\sum_{t}\log p_{model}\left(y^{(t)}\,|\left\{x^{(1)},..x^{(t)}\right\}\right)$$

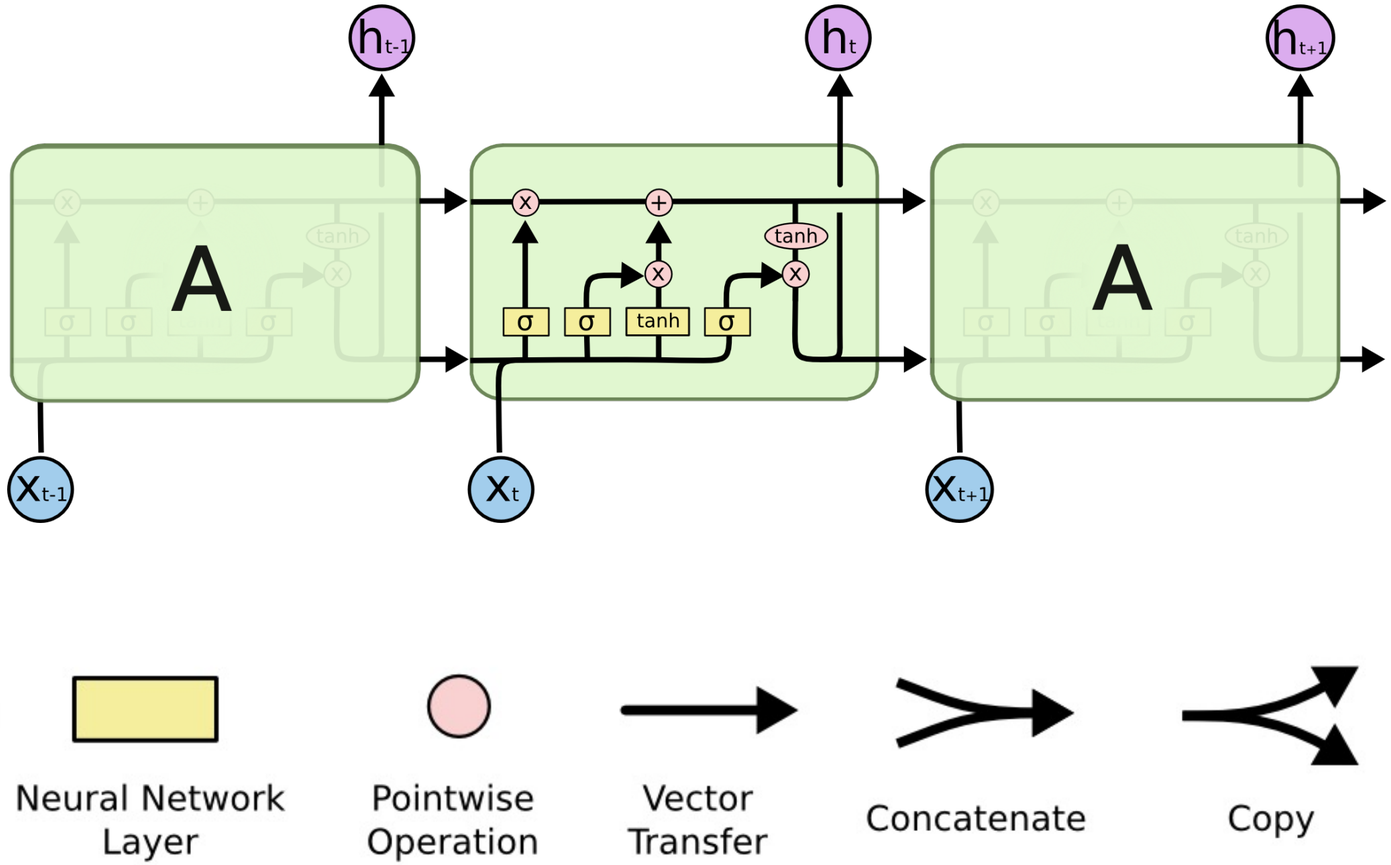
  - where p_{model} is given by reading the entry for y(t)

Long Short Term Memory (LSTM) RNNs

- Each line carries an entire vector, from the output of one node to the inputs of others

- The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers.

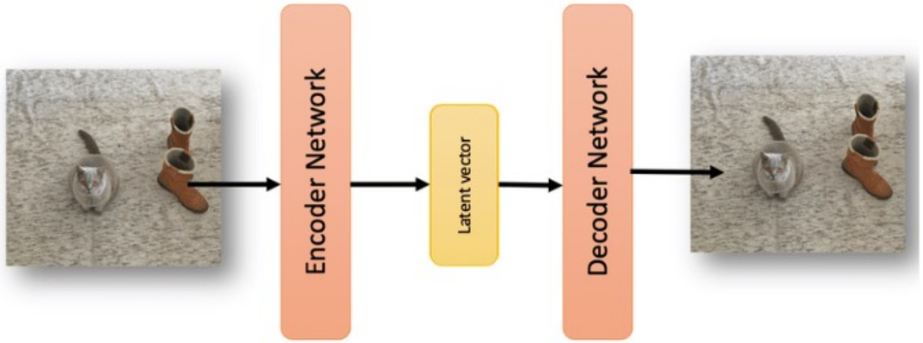
- Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

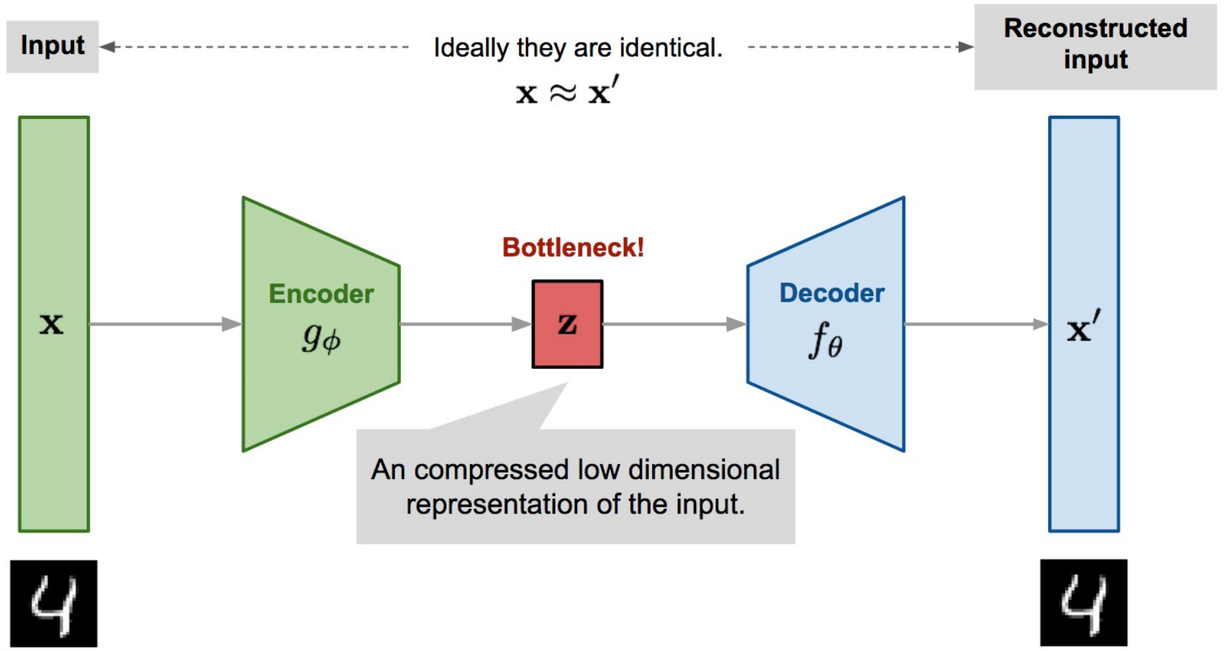


Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Autoencoder

- An autoencoder is a feed-forward neural net whose job it is to take an input and predict the reconstruction.
- To make this non-trivial, we need to add a bottleneck layer whose dimension is much smaller than the input.





Source 1: https://spraphul.github.io/blog/VAE

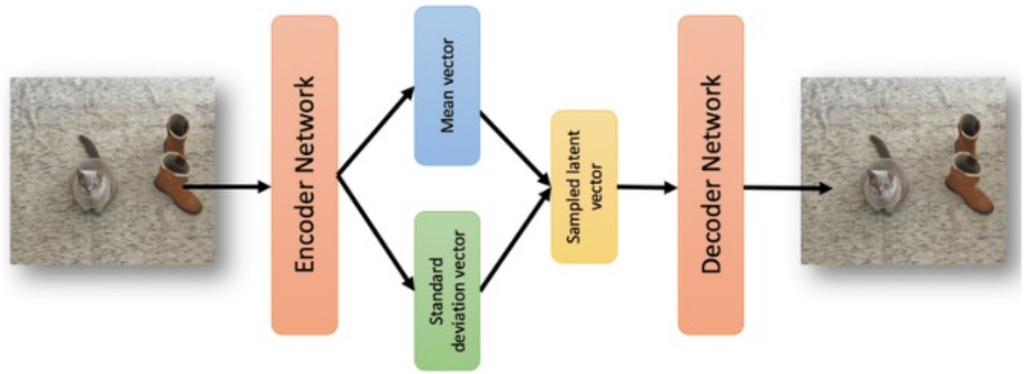
Source 2: https://www.researchgate.net/figure/In-a-a-Variational-AutoEncoder-VAE-scheme-with-the-mean-and-standard-deviation_fig1_339447623

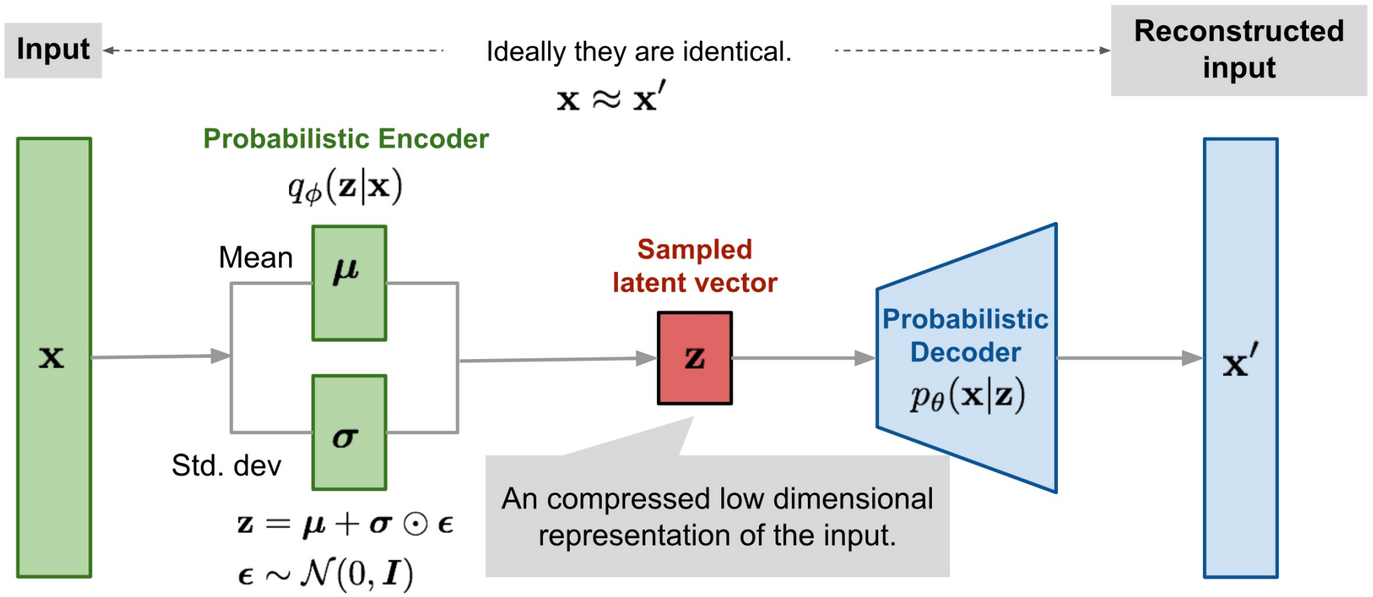
Autoencoder

- Motivation
  - Mapping high-dimensional data to two dimensions for visualization like PCA
  - Data Compression (i.e., reducing the file size)
  - Learning abstract features in an unsupervised way so you can apply them to a supervised task.
  - Unlabeled data can be much more plentiful than labeled data for certain tasks.
- Bottleneck hidden layer forces network to learn a compressed latent representation.
- Reconstruction loss forces the latent representation to capture as much information about the data as possible.
- Autoencoding = Automatically encoding data
- Main limitation
  - Autoencoder cannot generate data without input images since it required latent vectors z to conduct data.

Variational Autoencoder

- Variational autoencoder is a probabilistic twist on autoencoder.
- Sample from the mean and standard dev. to compute latent samples (vectors)





Source 1: https://spraphul.github.io/blog/VAE

Source 2: https://www.researchgate.net/figure/In-a-a-Variational-AutoEncoder-VAE-scheme-with-the-mean-and-standard-deviation_fig1_339447623

DEEP
LEARNING
INSTITUTE

PRAIRIE VIEW
A&M UNIVERSITY

DLI Accelerated Data Science Teaching Kit

Thank You