EEP 596: AI and Health Care || Lecture 7

Dr. Karthik Mohan

Univ. of Washington, Seattle

Apr 18, 2022

Logistics

- Next assignment on Arrhythmia Detection (Assignment 3)

Logistics

- Next assignment on Arrhythmia Detection
- How were Assignments 1 and 2 and Kaggle format?

(Percentile class)

Logistics

- Next assignment on Arrhythmia Detection
- How were Assignments 1 and 2 and Kaggle format?
- Anything else?

Last Lecture

- **Wearables and Data Access**

Last Lecture

- **Wearables and Data Access**
- **Sleep and Relaxation self case-study**

Last Lecture

- **Wearables and Data Access**
- **Sleep and Relaxation self case-study**
- **Introduction to Deep Learning**

Application

Today

- **Deep Learning Recap and Focus**

Today

- **Deep Learning Recap and Focus**
- **Deep Learning Methods for Anomaly Detection**

Today

- **Deep Learning Recap and Focus**
- **Deep Learning Methods for Anomaly Detection**
- **Deep Learning for other health care problems**

Anomaly Detection in IoT context

Properties of a good Anomaly Detector for IoT data streams

1. **Speed:** Ability to handle data coming in rapidly. In the case of heart rate monitoring and Arrhythmia Detection, data coming in every 5 seconds (e.g. Cardiogram App for Apple Watch).

Anomaly Detection in IoT context

Properties of a good Anomaly Detector for IoT data streams

1. **Speed:** Ability to handle data coming in rapidly. In the case of heart rate monitoring and Arrhythmia Detection, data coming in every 5 seconds (e.g. Cardiogram App for Apple Watch).
2. **Memory:** Ability to handle massive amounts of data with limited memory. One data point a second is 86400 data points a day. With multiple sources of data, this can go to a million data points a day - However, anomaly detectors may only be able to use a small window size around the current timestamp to analyze and detect anomalies.

Anomaly Detection in IoT context

Properties of a good Anomaly Detector for IoT data streams

1. **Speed:** Ability to handle data coming in rapidly. In the case of heart rate monitoring and Arrhythmia Detection, data coming in every 5 seconds (e.g. Cardiogram App for Apple Watch).
2. **Memory:** Ability to handle massive amounts of data with limited memory. One data point a second is 86400 data points a day. With multiple sources of data, this can go to a million data points a day - However, anomaly detectors may only be able to use a small window size around the current timestamp to analyze and detect anomalies.
3. **High dimensionality:** Heart rate is single dimension. Combine this with other sensors, and many more dimensions emerge and make the data stream more complex. Health care applications might be good on this.

Streaming Videos Analytics → Super challenging!

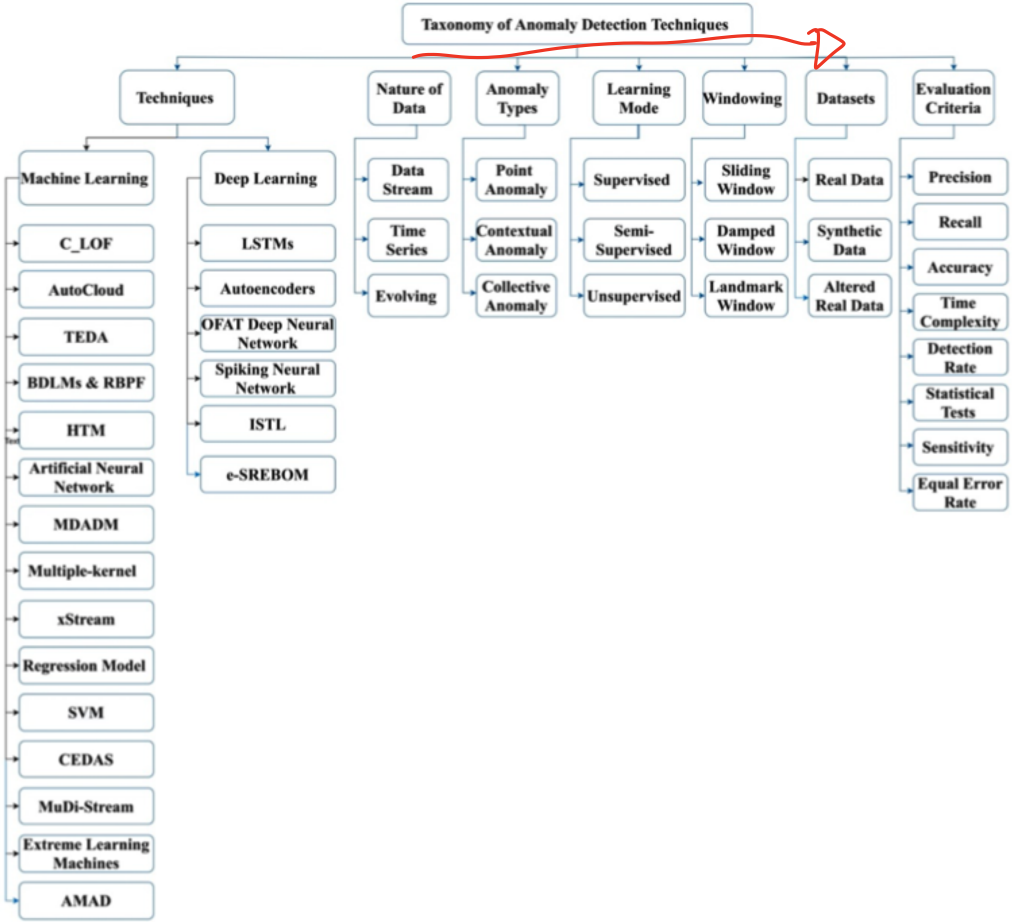
→ Medical Imaging (Static) or IoT → Time-Series but not high-dim!

Anomaly Detection in IoT context

Properties of a good Anomaly Detector for IoT data streams

1. **Speed:** Ability to handle data coming in rapidly. In the case of heart rate monitoring and Arrhythmia Detection, data coming in every 5 seconds (e.g. Cardiogram App for Apple Watch).
2. **Memory:** Ability to handle massive amounts of data with limited memory. One data point a second is 86400 data points a day. With multiple sources of data, this can go to a million data points a day - However, anomaly detectors may only be able to use a small window size around the current timestamp to analyze and detect anomalies.
3. **High dimensionality:** Heart rate is single dimension. Combine this with other sensors, and many more dimensions emerge and make the data stream more complex. Health care applications might be good on this.
4. **Data Drift:** Ability to handle changing data streams, changing baselines in HR or O2, understanding contexts.

Taxanomy of Anomaly Detection Landscape



Depth

Anomaly Detection Methods

Table 1. Summary of machine learning techniques for data stream anomaly detection.

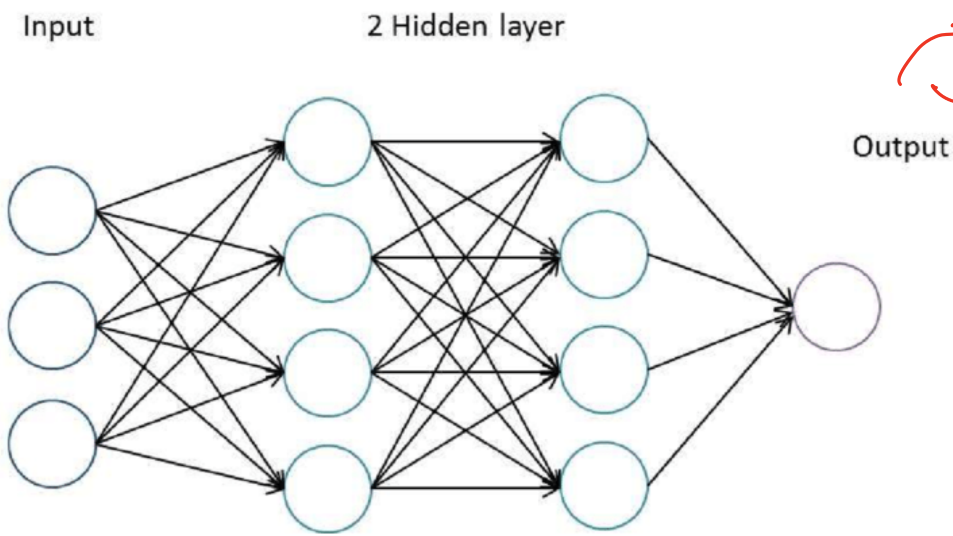
Techniques	Nature of the Data	Types of Anomaly	Anomaly Detection Types	Windowing	Dataset	Evaluation Criteria
C_LOF [14]	Data Stream (evolving)	Point anomaly	Unsupervised learning using density	Sliding window	synthetic and real-life datasets.	Precision, Recall, and Accuracy
AutoCloud [24]	Data Stream (evolving)	Point anomaly	Unsupervised learning using clustering	Sliding window	Artificial and real dataset	N/A
TEDA Clustering [25]	Data Stream (evolving)	Point anomaly	Unsupervised learning using clustering	Sliding window	Own synthetic data sets	Accuracy, Time complexity
Combination of (BDLMs) & (RBPF) [26]	Data Stream (evolving)	Point anomaly	Unsupervised learning using density	Sliding window	Artificial dataset	Accuracy, the Detection rate
HTM [27]	Data Stream	Point anomaly	Unsupervised learning based on HTM	N/A	Dataset of space imager data stream	Accuracy
Artificial Neural Network [28]	Continuous and image data	Point anomaly	Unsupervised learning on patterns of WSN nodes	Sliding window	The experimental tests that have been conducted and cover more than 27	Accuracy
MDADM [29]	Continuous data	Point anomaly	Supervised learning	N/A	Own dataset	Accuracy

DL Methods

Techniques	Nature of the Data	Types of Anomaly	Anomaly Detection Types	Windowing	Dataset	Evaluation Criteria
LSTMs [40]	Time-Series	Point anomaly	Supervised learning using deep learning	Sliding window	Yahoo Webscope	Confusion matrix.
Autoencoder [41]	Data Stream (evolving)	Point anomaly	Unsupervised learning based on Ensembles neural networks	Sliding window	HTTP, SMTP, SMTP+HTTP, COVERTYPE, SHUTTLE, Weather	AUC
(OFAT) Deep neural network [42]	Time series	Point anomaly	Supervised learning	Window-based	Web traffic dataset, Avocado dataset, Temperature dataset	Statistical tests (average Rank), Mean Average Score (MAS)
Evolving spiking neural network [43]	Data Stream (evolving)	Point anomaly	Unsupervised learning	Sliding window	3 Benchmark dataset	Accuracy
ISTL [44]	Data Stream (evolving)	Point anomaly	Unsupervised learning based on deep learning	Sliding Window	UCSD Pedestrian datasets, Ped 1 and Ped 2) and CUHK Avenue dataset	Accuracy (ACU), Equal Error Rate (EER),
(e-SREBOM) [43]	Data Stream (evolving)	Point anomaly	Unsupervised learning using Spiking Neural Networks (eSNN)	Window-based	Water_tower_dataset, gas_dataset, electric_dataset	Accuracy, Speed, Time to learn

Feed Forward Neural Based Anomaly Detection

sliding window



Comparison of Methods on different dimensions

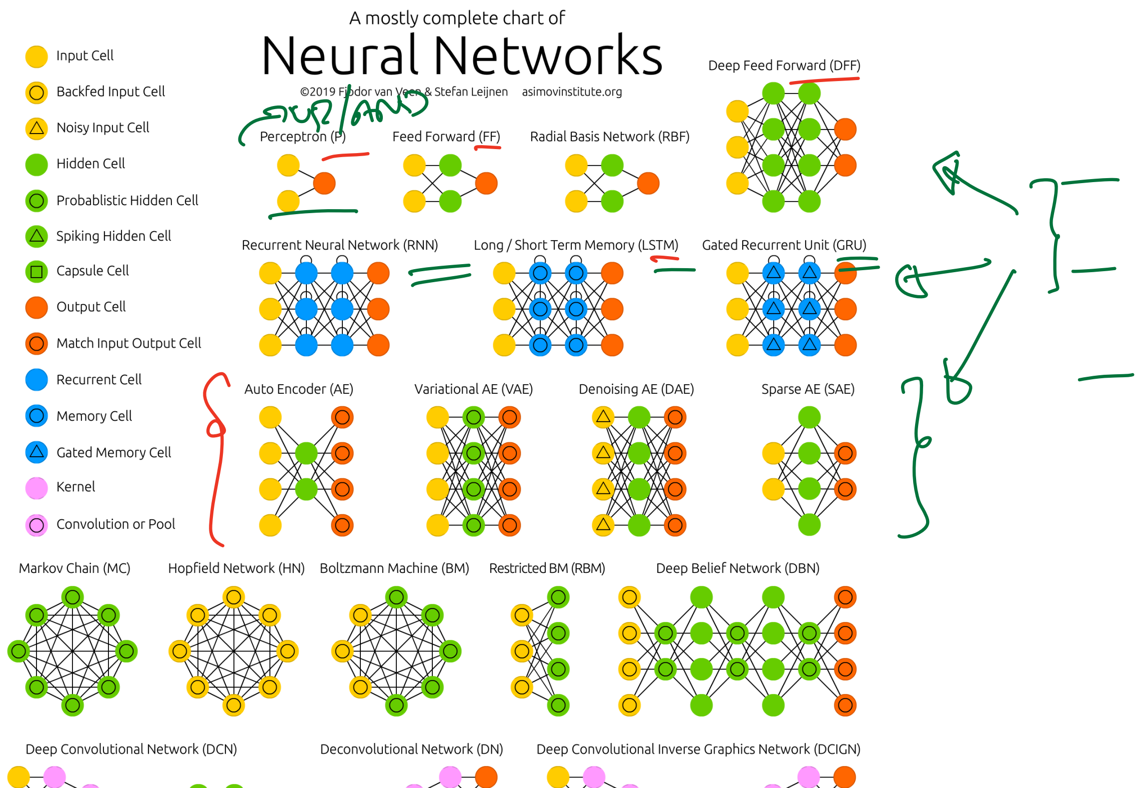
Techniques/Methods	Projection	Handling Noisy Data	Limited Time	Limited Memory	Handling Evolving Data	Handling High Dimensional Data	Evolving Features	Scalability
Regression Model [32]			√					
Super Vector Machine [33]		√		√				√
HTM [34]	√	√		√	√	√		
CEDAS [36]		√		√				
HTM [35]		√	√	√	√			
MuDi-Stream [37]				√		√		
Extreme Learning Machines [38]	√	√		√	√	√		√
AMAD [39]	√	√	√	√	√	√		√
LSTMs [40]		√		√				
Autoencoder [41]		√			√			√
(OFAT) Deep neural network [42]		√			√	√	√	√
Evolving spiking neural network [43]		√			√			
ISTL [44]			√	√	√			
(e-SREBOM) [43]		√	√	√	√			√

Deep Learning Recap and Architectures for Anomaly Detection

More DL Architectures

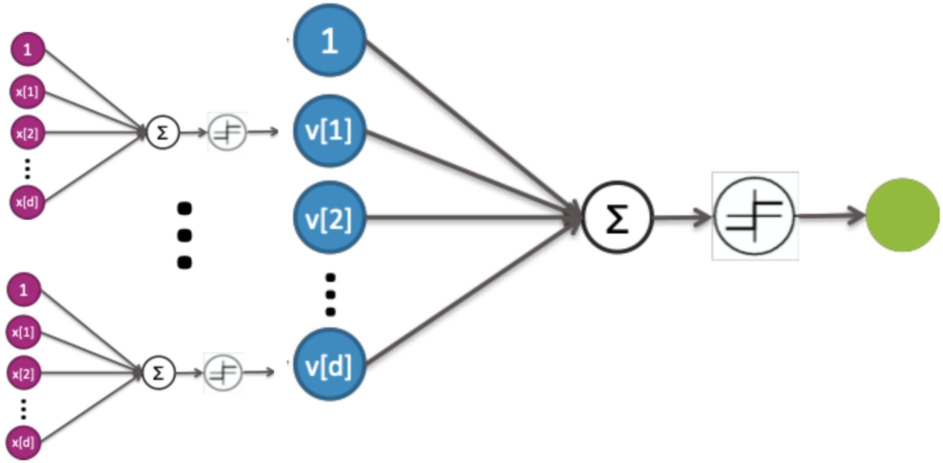
Neural Networks Zoo

Zoo Reference



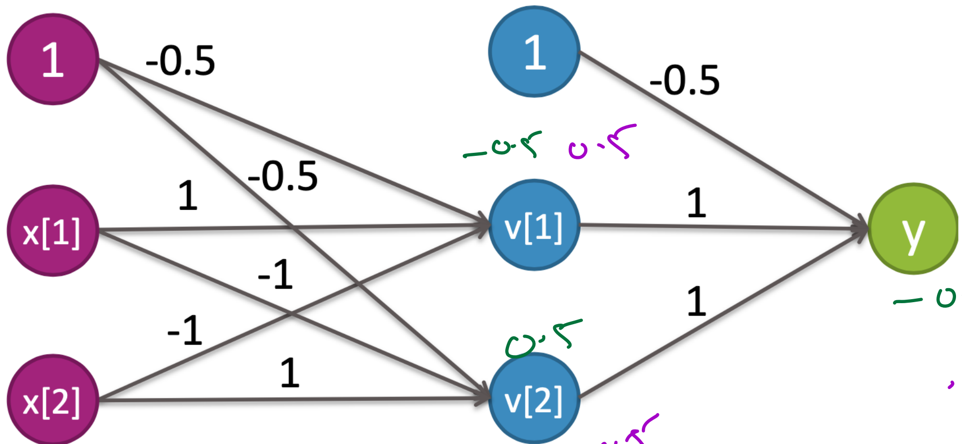
Multi-Layer Perceptron (MLP)

Feed forward NW
– Can learn almost any function!!



Multi-Layer Perceptron (MLP)

→ can learn an XOR!

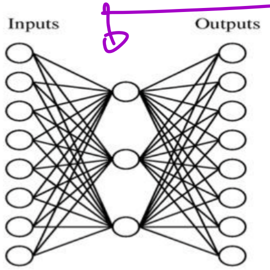


} XOR

2 Layer Neural Network

Two layer neural network (alt. one hidden-layer neural network)

1 hidden Layer



Single

$$out(x) = g\left(w_0 + \sum_j w_j x[j]\right)$$

1-hidden layer

$$out(x) = g\left(w_0 + \sum_k w_k g\left(w_0^{(k)} + \sum_j w_j^{(k)} x[j]\right)\right)$$

Perceptron to Logistic Regression

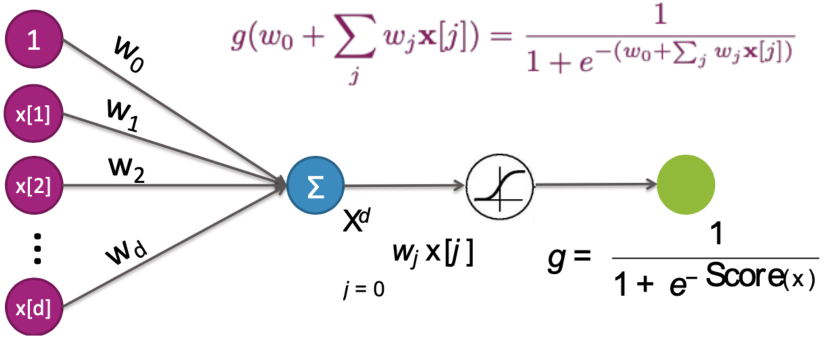
1 layer NN

LR

Non-Linearity

Feedforward NN

$$g(w_0 + \sum_j w_j \mathbf{x}[j]) = \frac{1}{1 + e^{-(w_0 + \sum_j w_j \mathbf{x}[j])}}$$



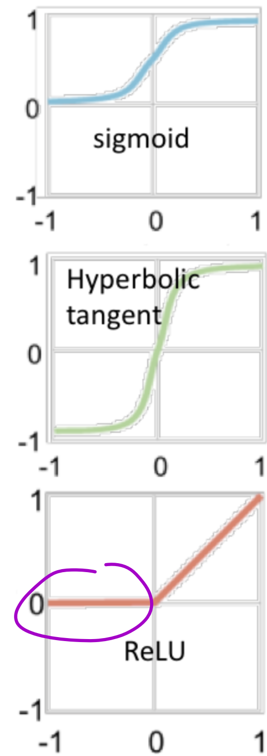
Choices for Non-Linear Activation Function

•Sigmoid
-Historically popular, but (mostly) fallen out of favor
•Neuron's activation saturates
(weights get very large -> gradients get small)
•Not zero-centered -> other issues in the gradient steps
-When put on the output layer, called "softmax" because interpreted as class probability (soft assignment)

•Hyperbolic tangent $g(x) = \tanh(x)$
-Saturates like sigmoid unit, but zero-centered

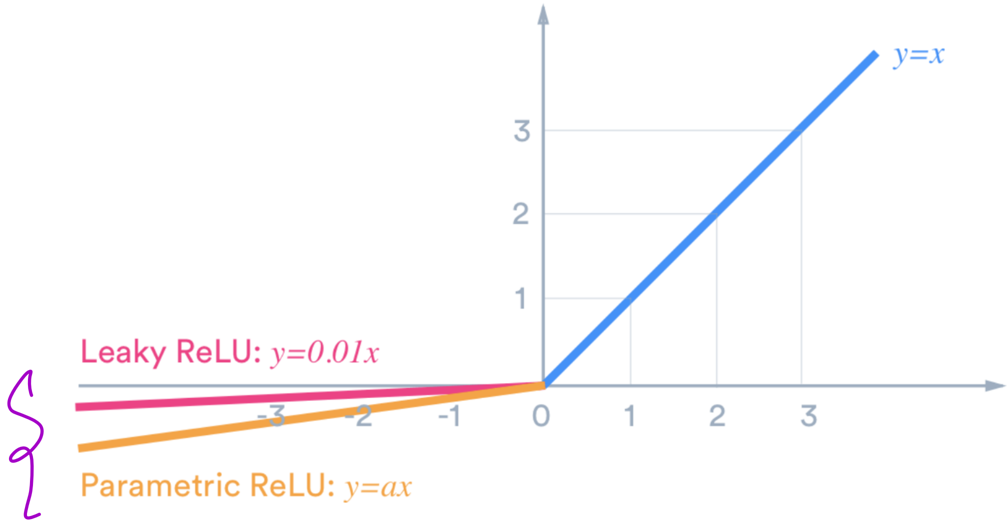
•Rectified linear unit (ReLU) $g(x) = x^{+} = \max(0,x)$
-Most popular choice these days
-Fragile during training and neurons can "die off"… be careful about learning rates
-"Noisy" or "leaky" variants

•Softplus $g(x) = \log(1+\exp(x))$
-Smooth approximation to rectifier activation



LR

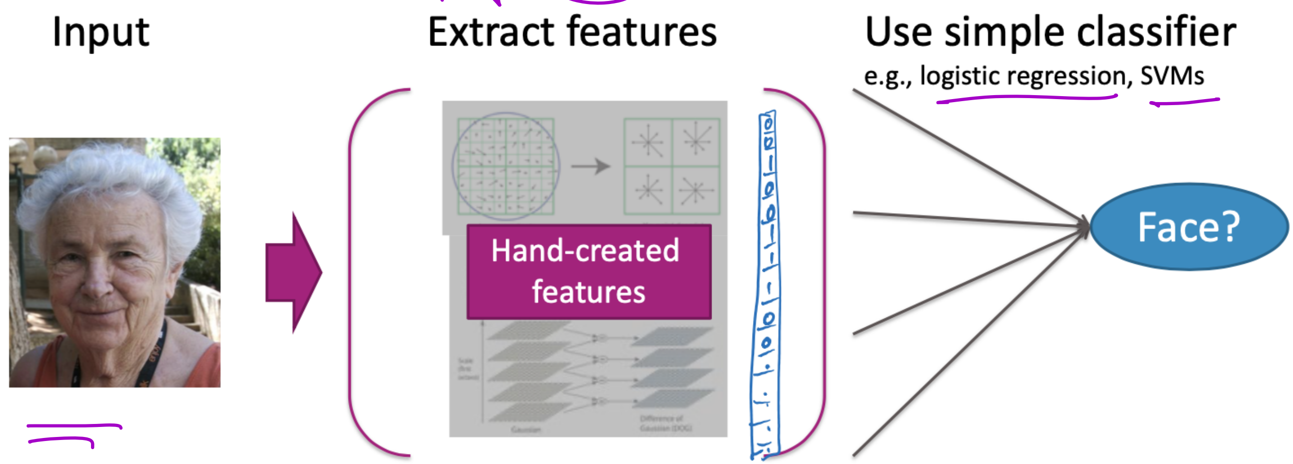
RELU vs Leaky RELU



Computer vision before deep learning

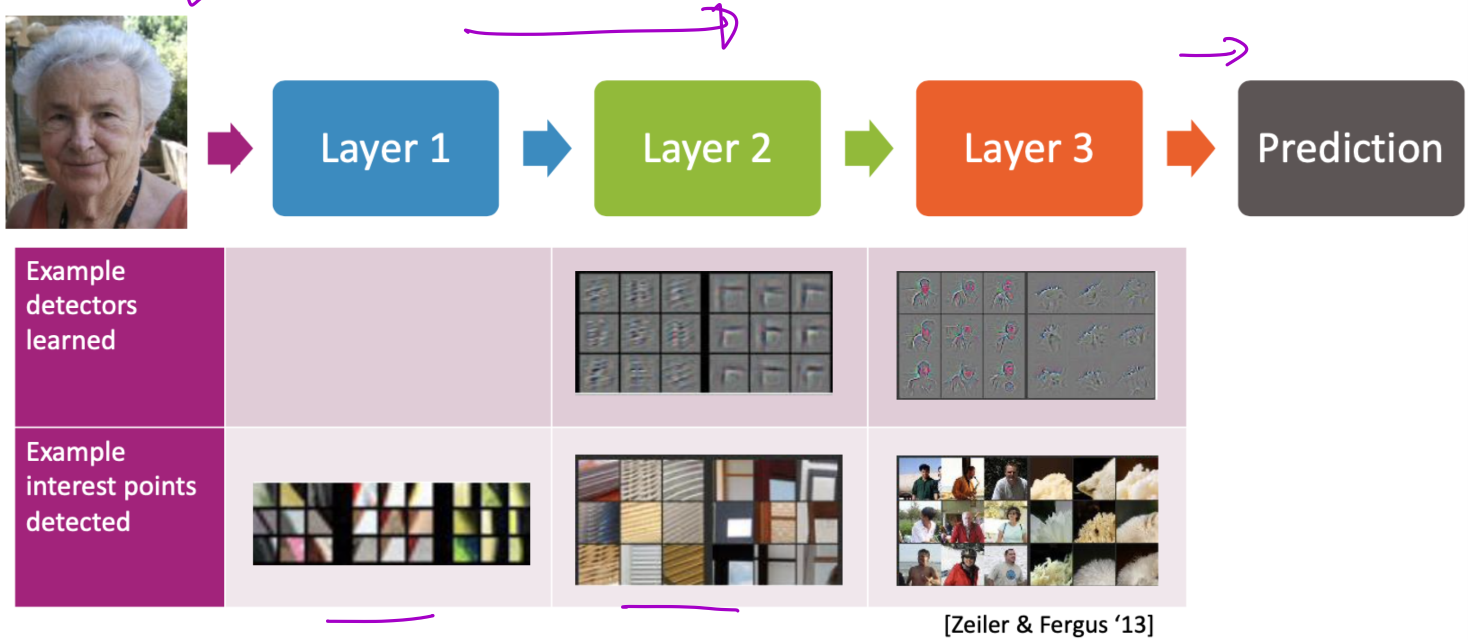
Lot of time spent here!

SIFT/...

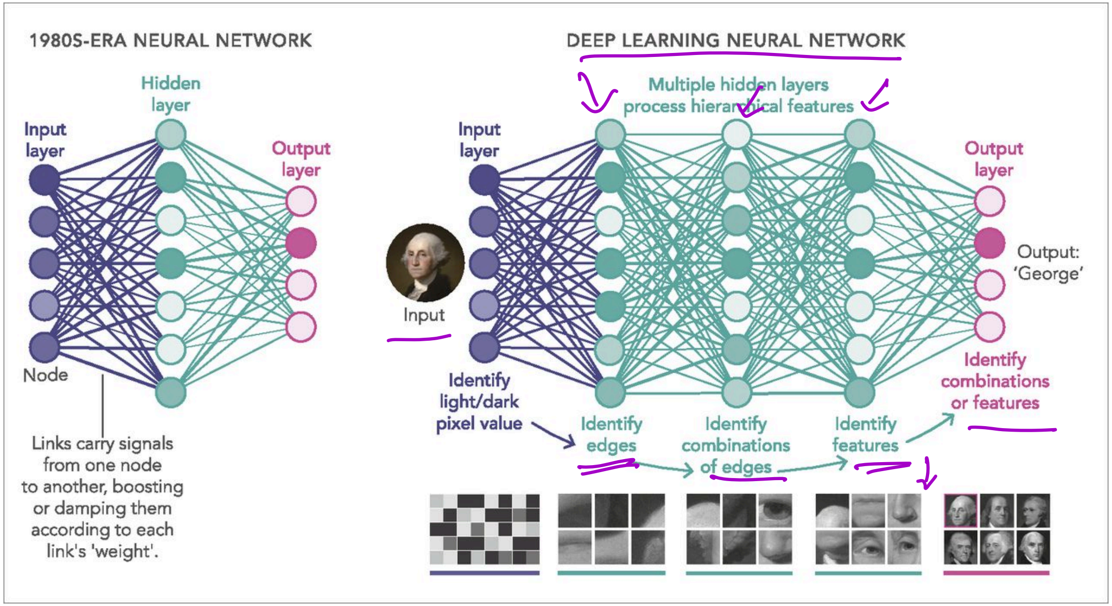


Computer vision after deep learning

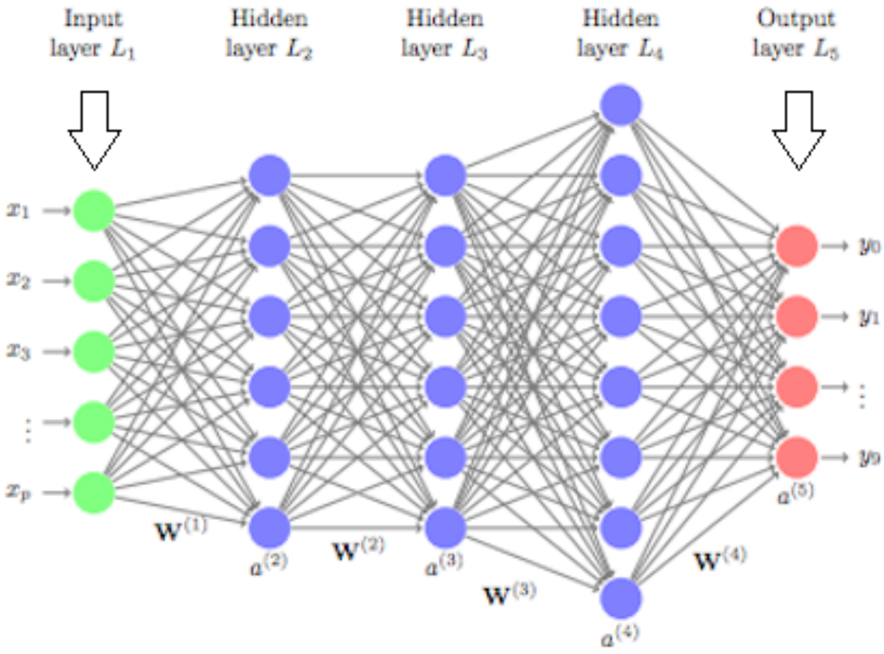
Feature Engineering



Feed-forward Deep Learning Architecture Example



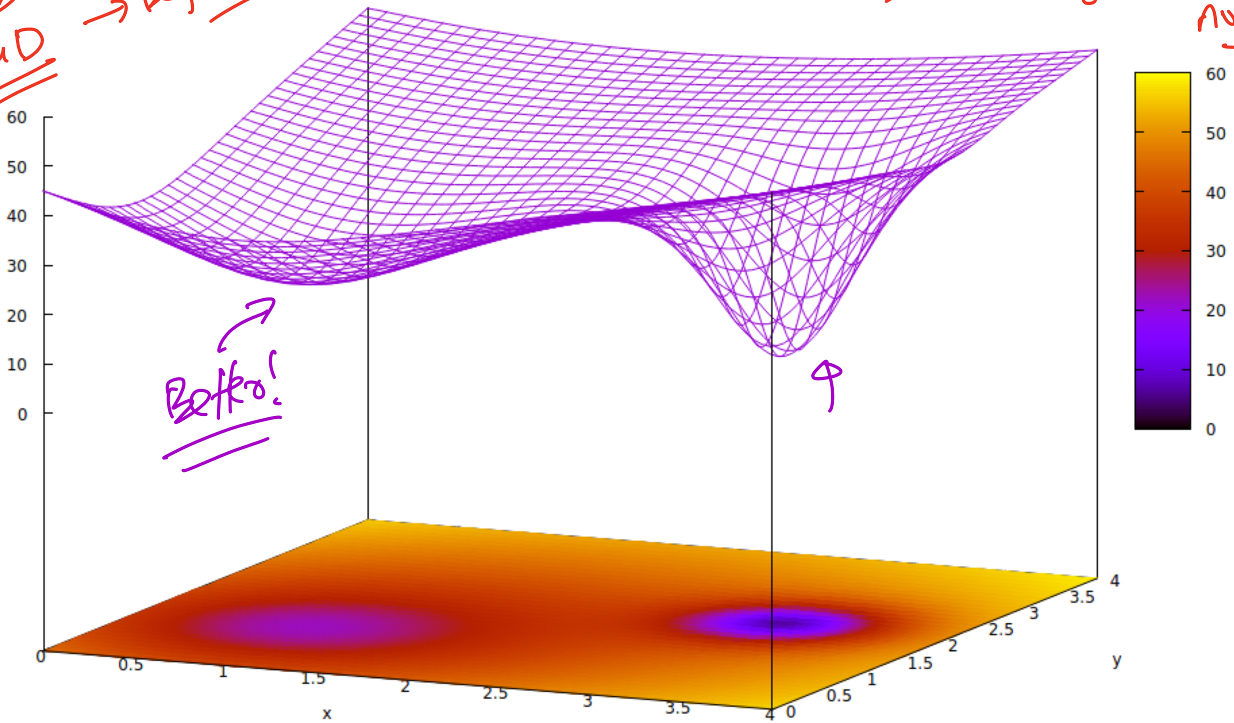
Feed-forward Deep Learning Architecture Example



Good vs Bad Local minima

mini-batch SGD → Default optimizer

Good Local minima → Generalize better & not overfit!

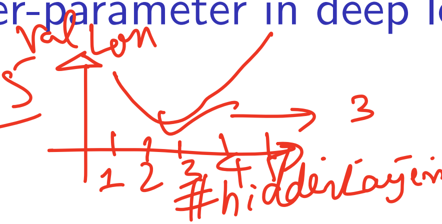


Hyper-parameters in Deep Learning

pollev.com/karthikmohan088

ICE #1: Which of the following is not a hyper-parameter in deep learning?

1. Learning rate
2. Number of Hidden Layers → 2, 4, 5
3. Number of neurons per hidden layer
4. None of the above



Hyper-parameters in Deep Learning

Hyper-parameters

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer

Hyper-parameters in Deep Learning

Hyper-parameters

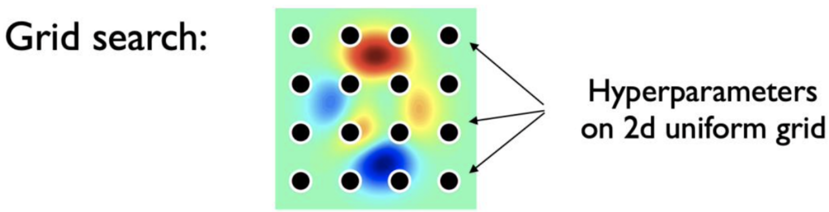
1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. Type of non-linear activation function used

Hyper-parameters in Deep Learning

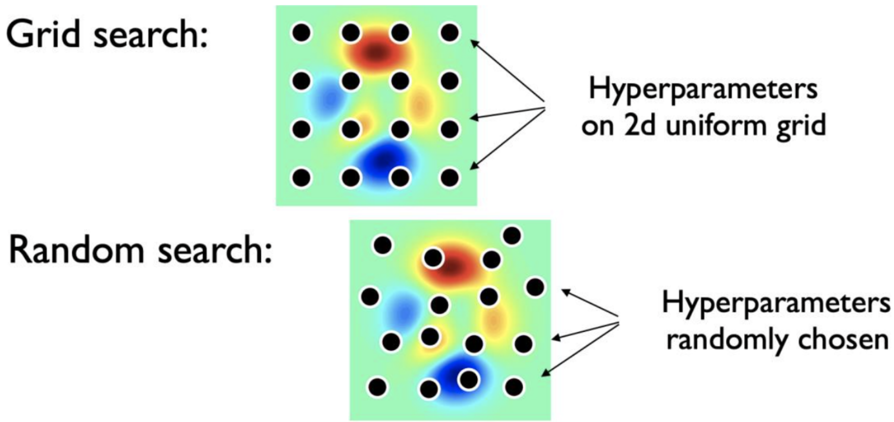
Hyper-parameters

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. Type of non-linear activation function used
5. Anything else?

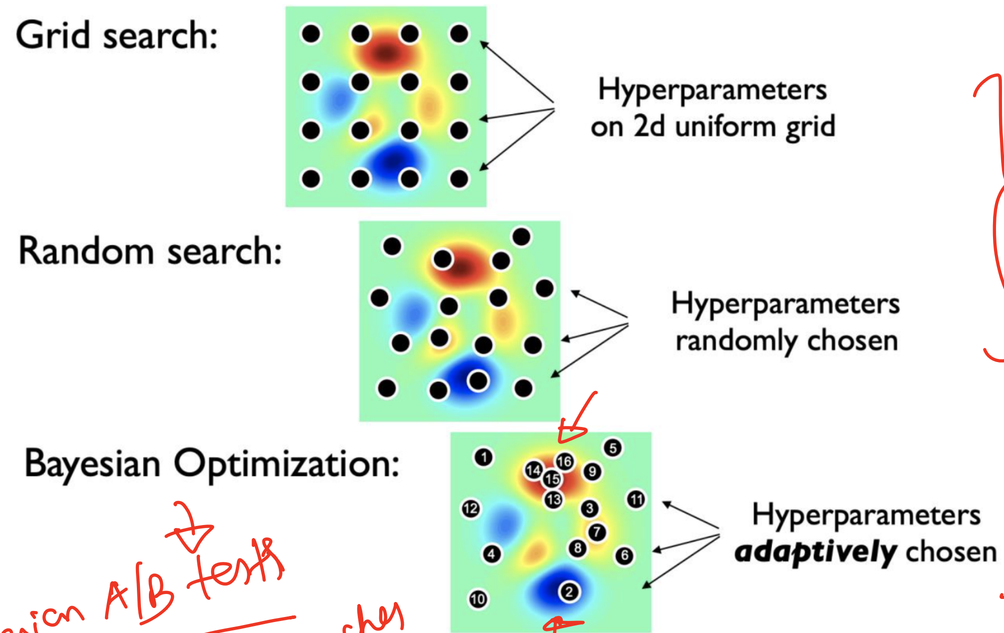
Hyper-parameter tuning methods



Hyper-parameter tuning methods



Hyper-parameter tuning methods



Not data driven

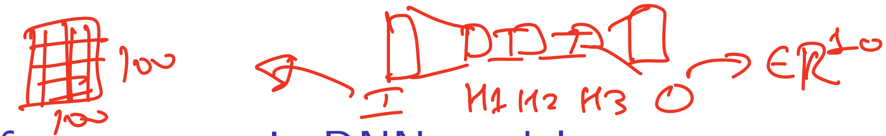
Random + ~~Un~~ Data driven

Bayesian A/B tests

Product Launches

Drug Trials

ICE #2



Compute the number of parameters in DNN model

Consider a DNN model with 3 hidden layers where each hidden layer has 1000 neurons. Let the input layer be raw pixels from a 100x100 image and the output layer has 10 dimensions, let's say for a 10 class image classification example. How many total parameters exist in the DNN model?

(Fully connected - FC)

1. 10 million parameters
2. 11 million parameters
3. 12 million parameters
4. 13 million parameters

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?

Over-fitting in DNNs

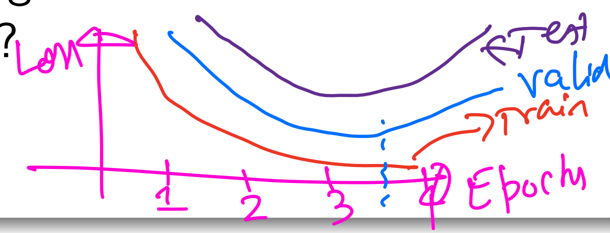
How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts! → Ensembling

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!
5. Early stopping is also a great strategy! Stop training the DL model when the validation error starts increasing. How's this different from regular validation we were doing earlier??



Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!
5. Early stopping is also a great strategy! Stop training the DL model when the validation error starts increasing. How's this different from regular validation we were doing earlier??
6. Book by Yoshua Bengio has tons of details and great reference for Deep Learning!

ICE #3

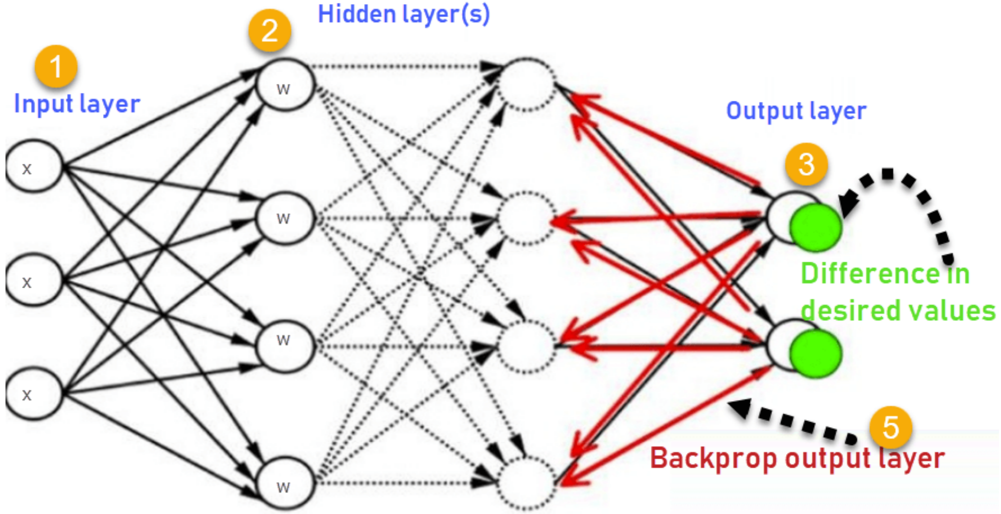
ML Models

Which of the following ML models can possibly learn the XOR function with enough training data:

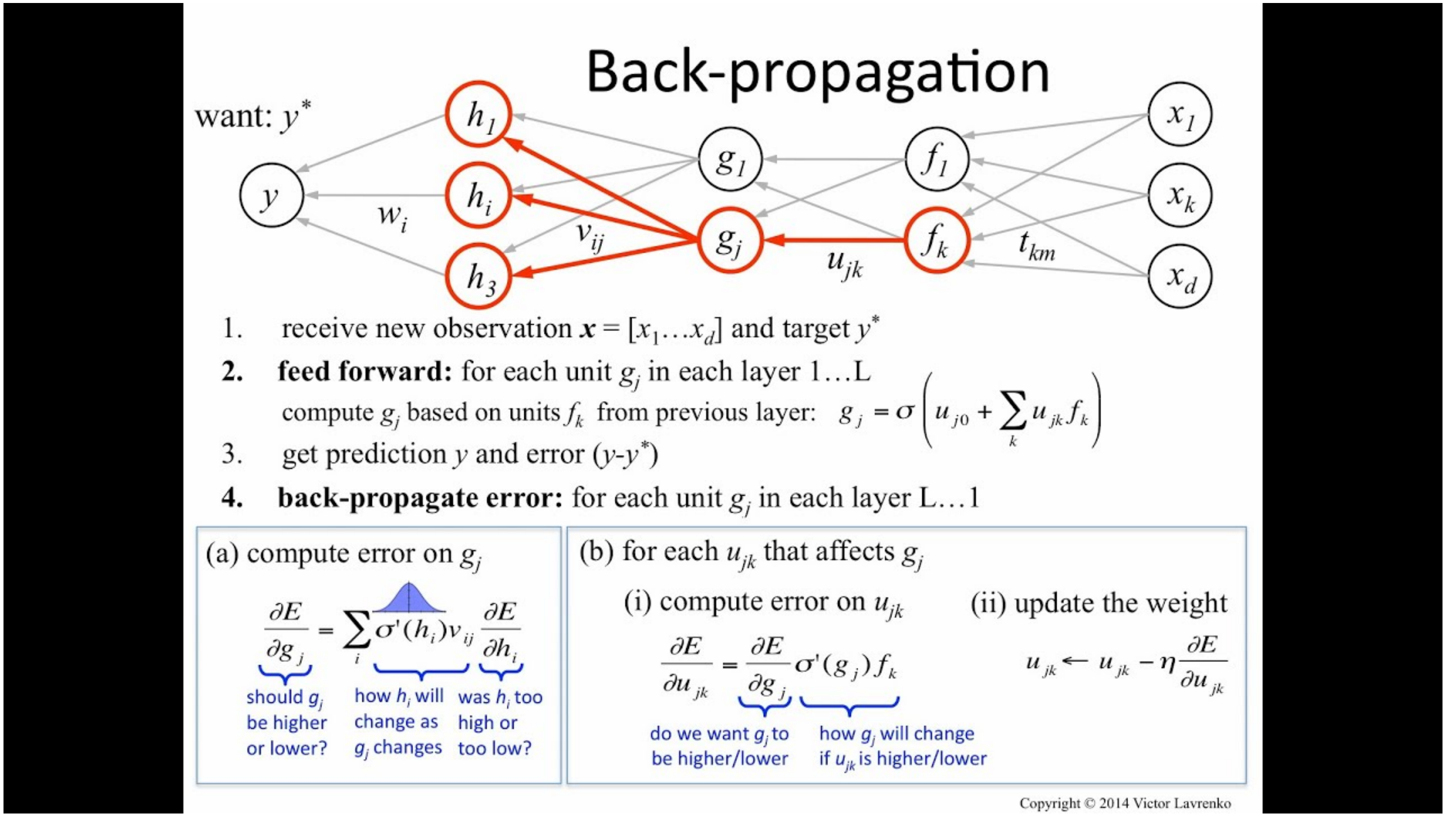
1. Logistic Regression
2. Decision Trees
3. SVM
4. Multi-Layer Perceptron

Forward Propagation vs Back-propagation

$$x^{100} \xrightarrow{\text{Forward}} \hat{y}^{100} - y^{100} = e^{100}$$



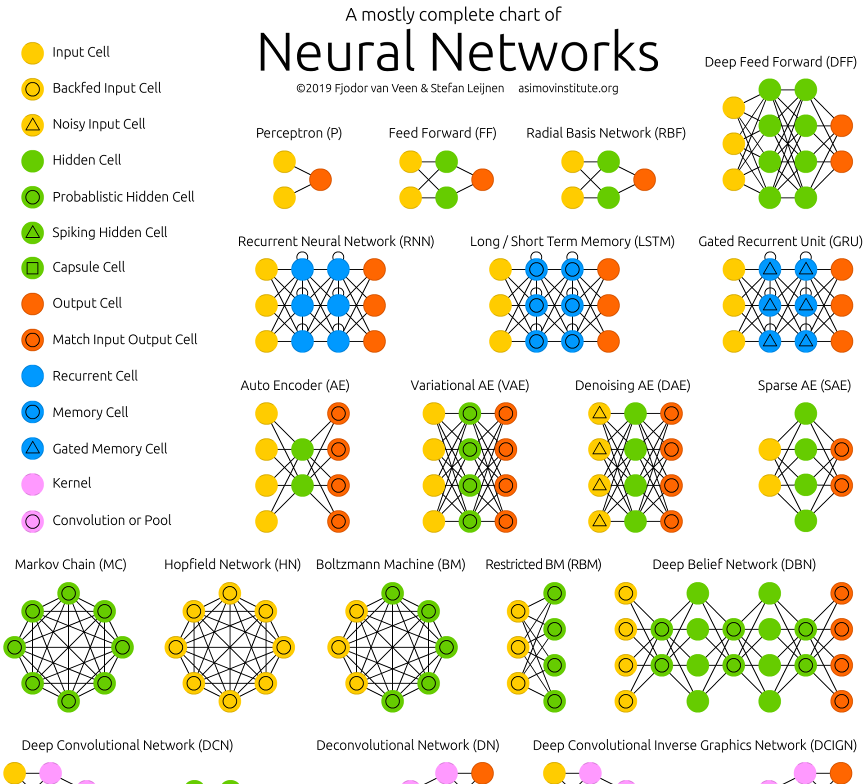
Back Propagation explained



More DL Architectures

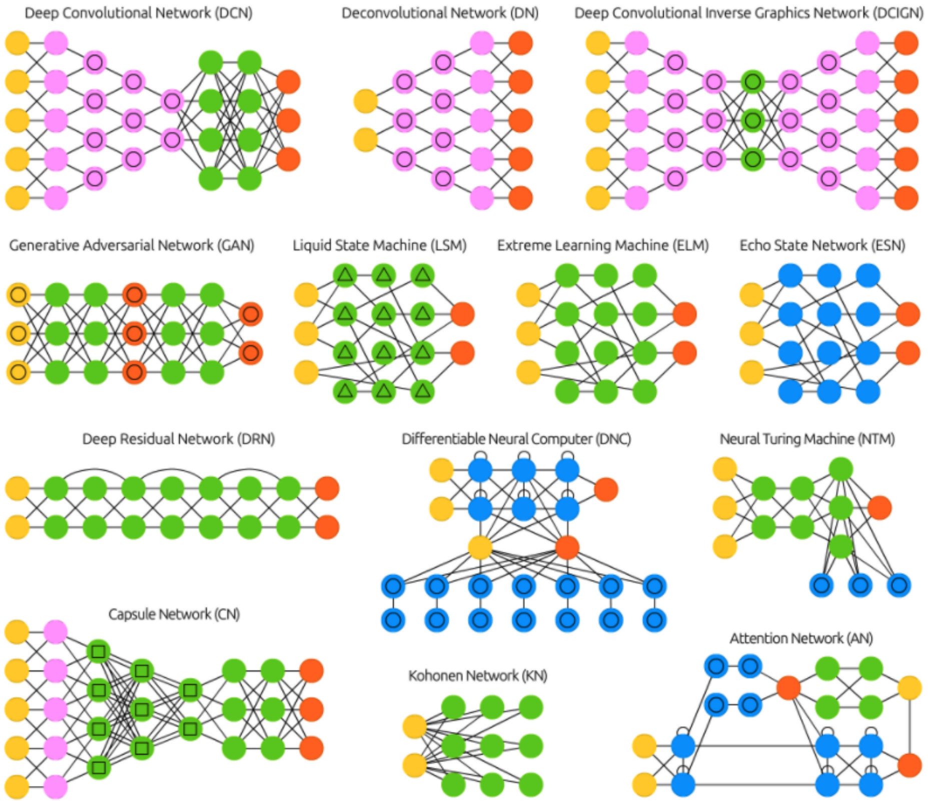
Neural Networks Zoo

Zoo Reference



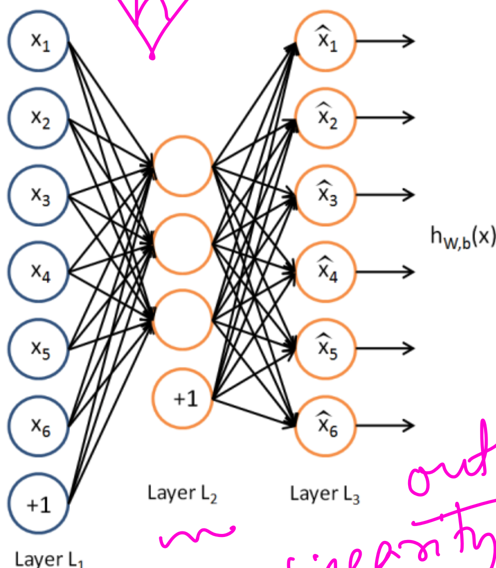
More DL Architectures

Neural Networks Zoo



Auto Encoders

$X \xrightarrow{h} X$



output

non-linearity

Input

ICE #5

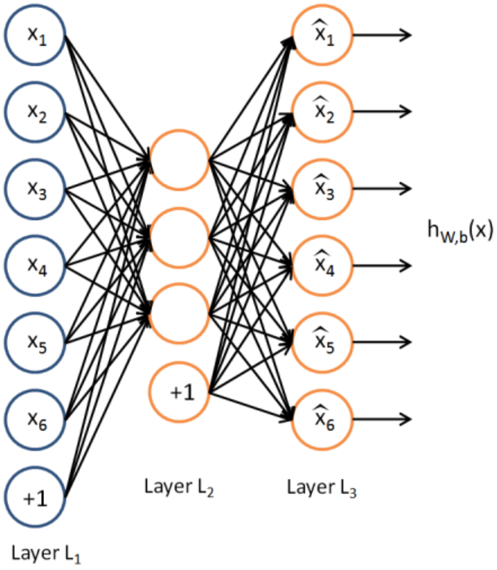
→ Dim. Reduction technique | SVD!

PCA vs Auto Encoder

Which of the following statements are true ?

1. Both PCA and Auto Encoders serve the purpose of dimensionality reduction
2. They are both linear models but one uses a neural nets architecture and the other is based on projections
3. PCA is robust to outliers while Auto Encoders are not
4. Auto Encoders are as better than Glove Embeddings to find low-dim embeddings for words

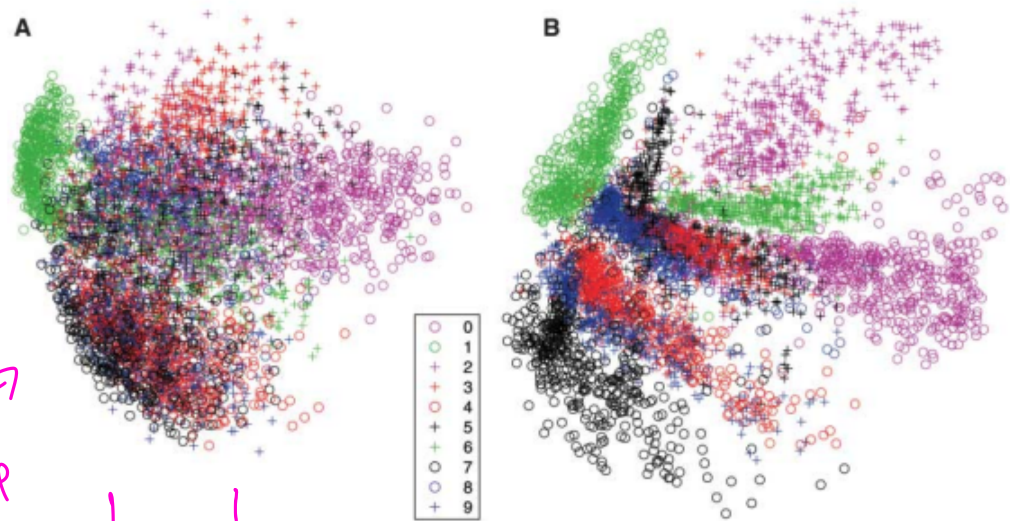
PCA vs Auto-Encoders



9 → 9 9 9 9

Reading Reference for AE Dimensionality Reduction

Fig. 3. (**A**) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (**B**) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (*8*).



No clear clustering

PCA

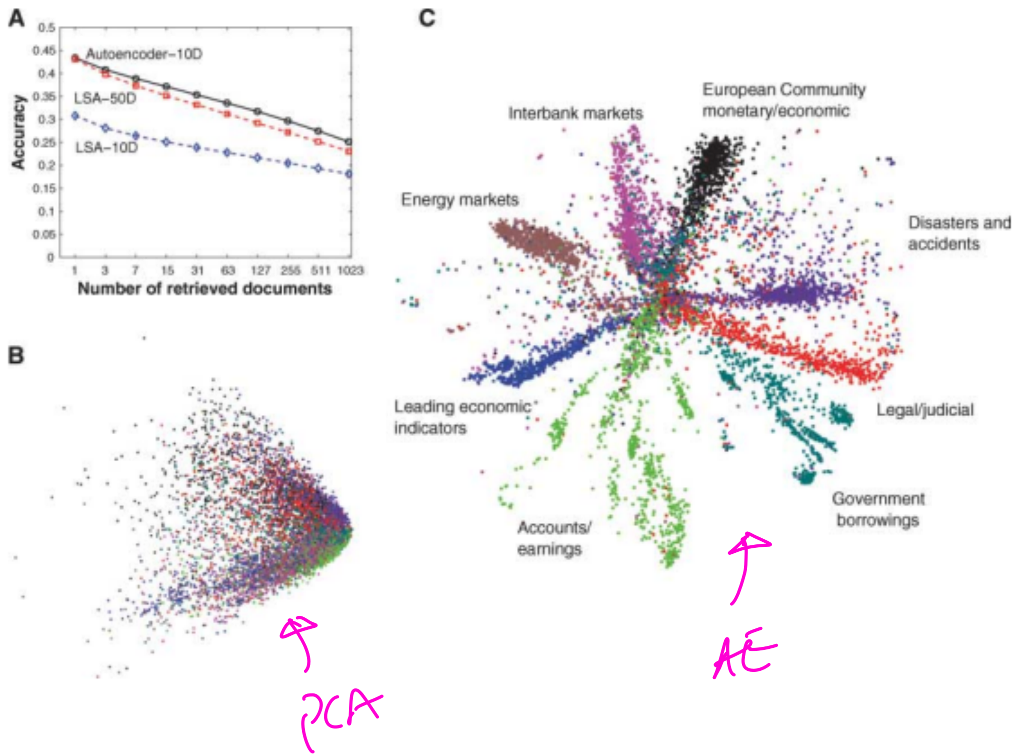
Digits

AE → Non-Linearity

AutoEncoders and Dimensionality Reduction

Reading Reference for AE Dimensionality Reduction

Fig. 4. **(A)** The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. **(B)** The codes produced by two-dimensional LSA. **(C)** The codes produced by a 2000-500-250-125-2 autoencoder.



AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization

unsupervised Learning

AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization
2. Use Neural Networks architecture and hence can encode non-linearity in the embeddings

AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization
2. Use Neural Networks architecture and hence can encode non-linearity in the embeddings
3. Anything else?

AutoEncders Summary

1. Auto-Encoders are a method for dimensionality reduction and can do better than PCA for visualization
2. Use Neural Networks architecture and hence can encode non-linearity in the embeddings
3. Anything else?
4. Auto Encoders can learn convolutional layers instead of dense layers - Better for images! More flexibility!!

Removing obstacles in images

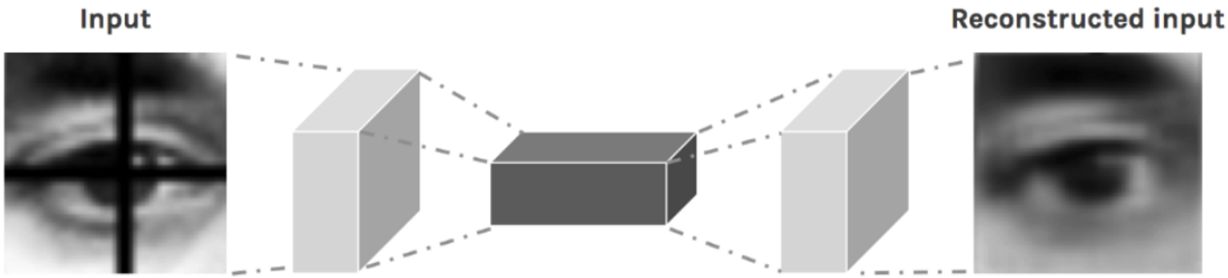
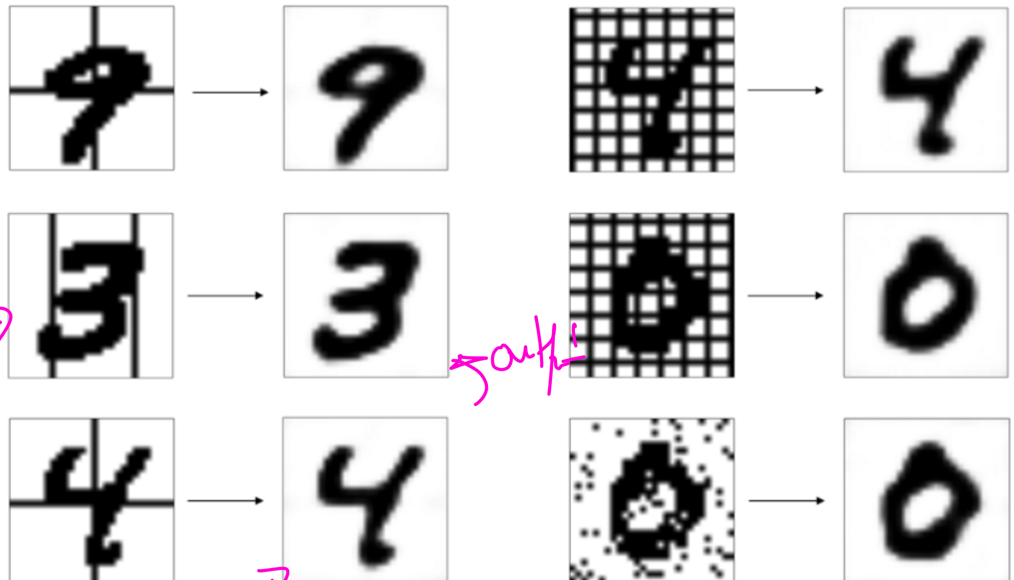


Figure 12: Reconstructed image from missing image [14]

Removing obstacles in images

Input

Output

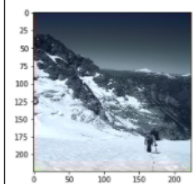
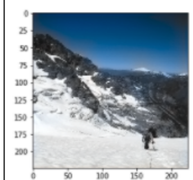
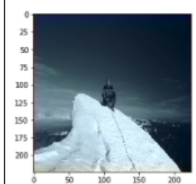
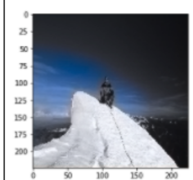
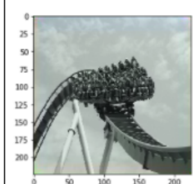
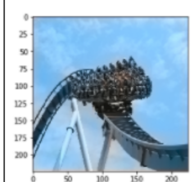


Prediction

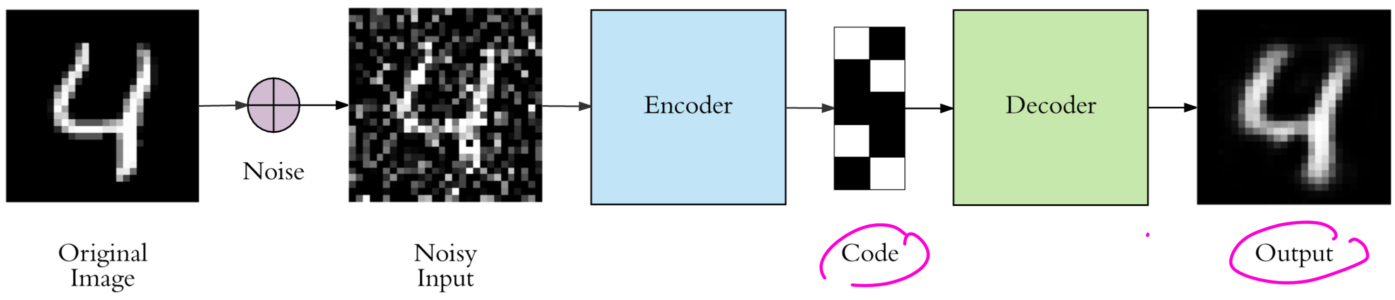
Figure 13: Source [15]

Gaussian noise

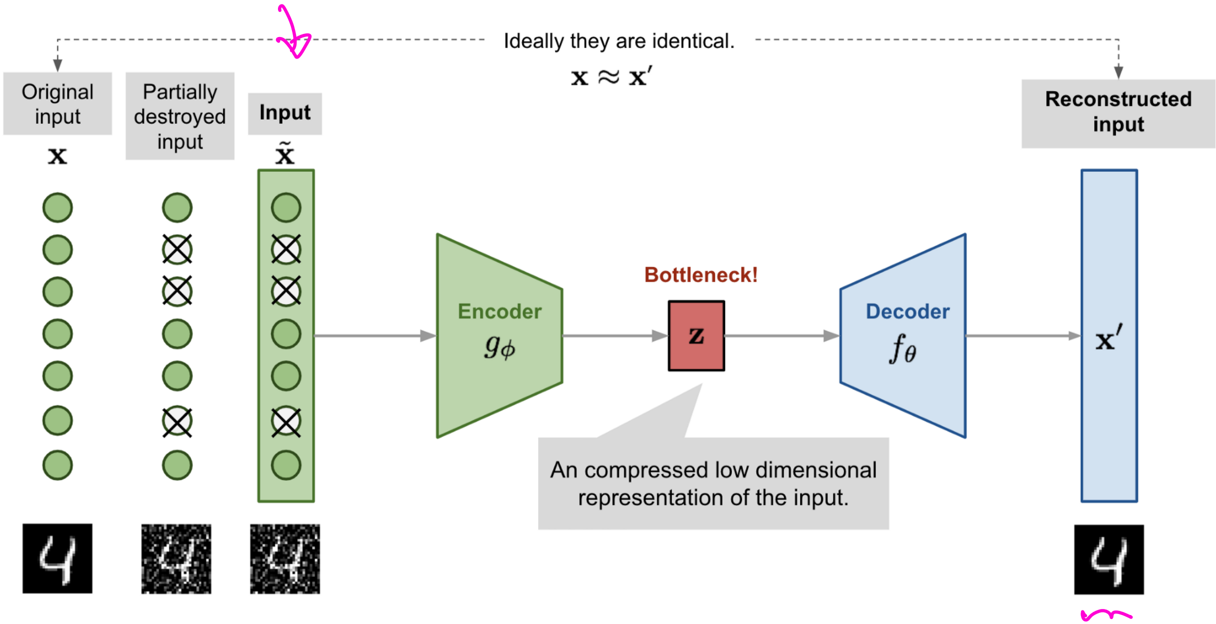
Coloring Images

Gray Image	Vanilla Autoencoder	Merge Model (YCbCr)	Merge Model (LAB)	Original
				
				
				

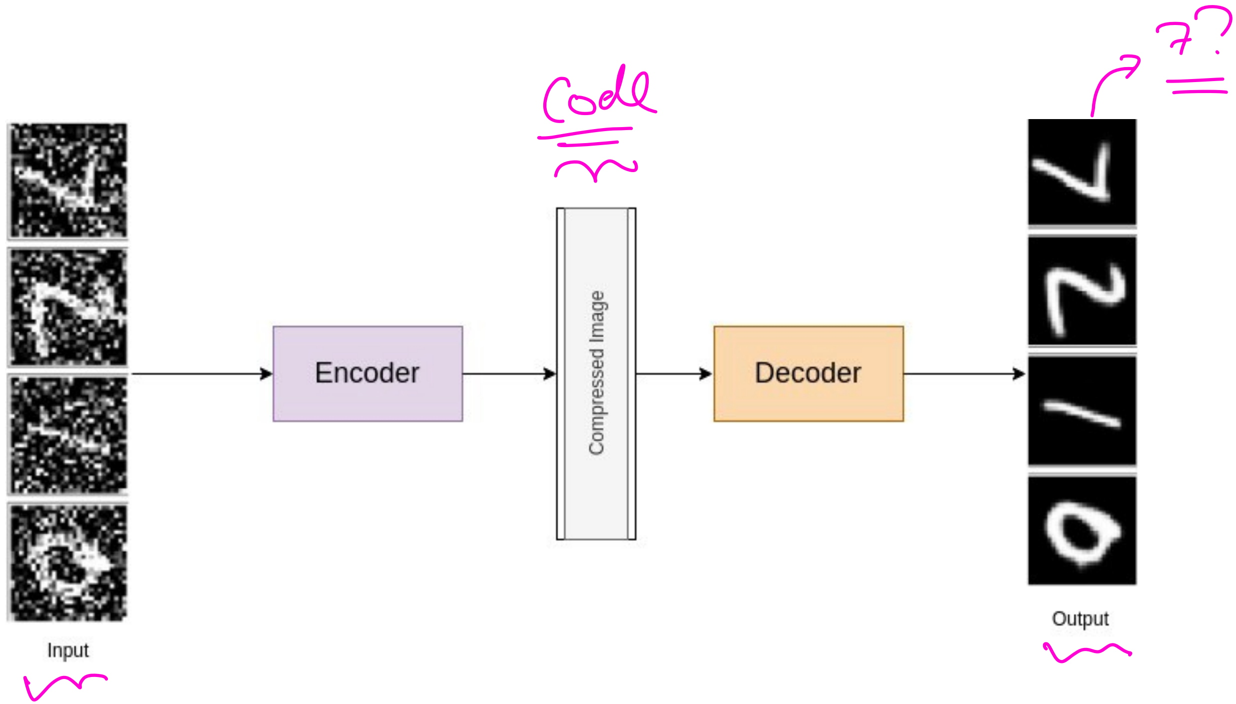
De-noising Auto Encoders



De-noising Auto Encoders



De-noising Auto Encoders



De-noising Auto Encoders

Details

- Just like an Auto Encoder

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to "de-noise" data, esp. useful for images!

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to "de-noise" data, esp. useful for images!
- Esp. useful for a category of objects or images (e.g. digit recognition or face recognition, etc)

ICE #6

Unsupervised Learning

Which of these is NOT an example of unsupervised learning?

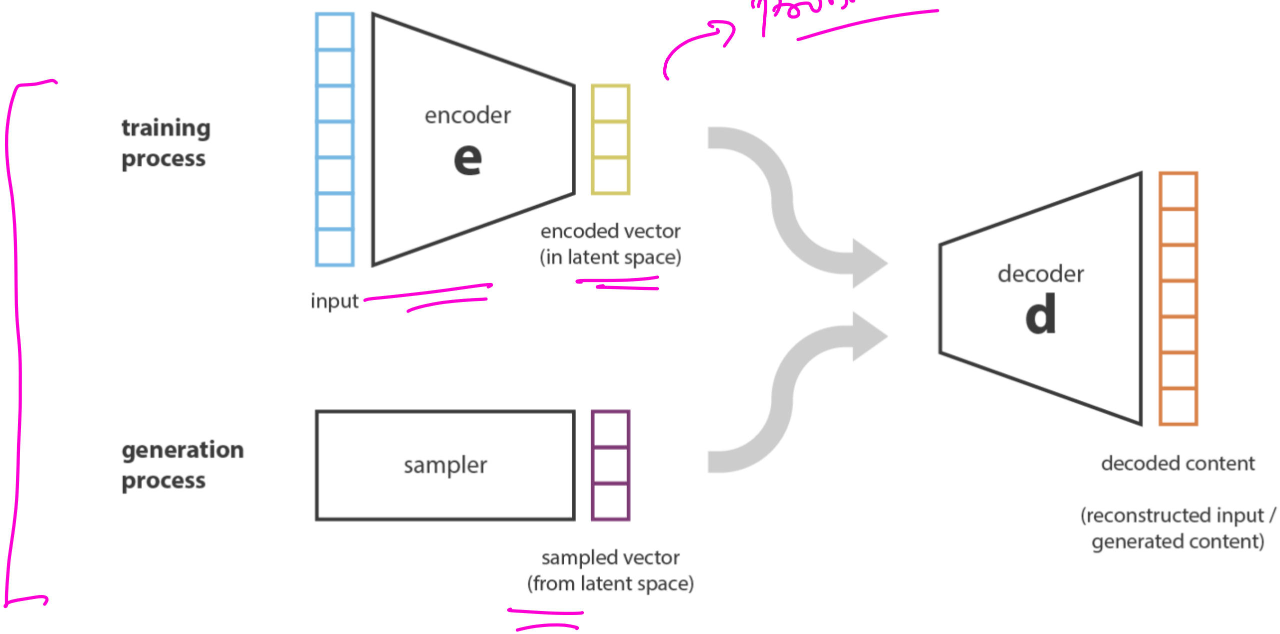
1. Perceptron
2. Auto Encoder
3. De-noising Auto Encoder
4. K-means++
5. None of the above
6. All of the above

Variational Auto Encoders

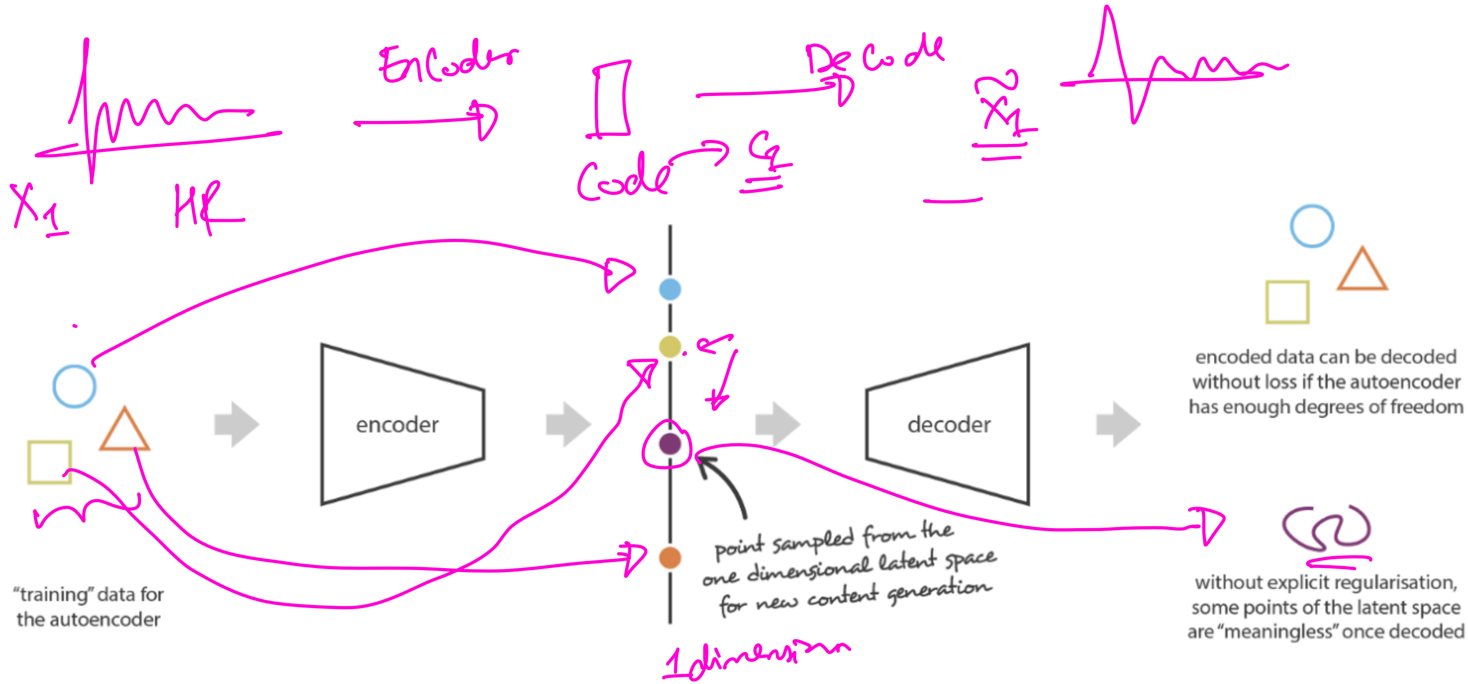
→ Anomaly

HR

→ Prob. distn

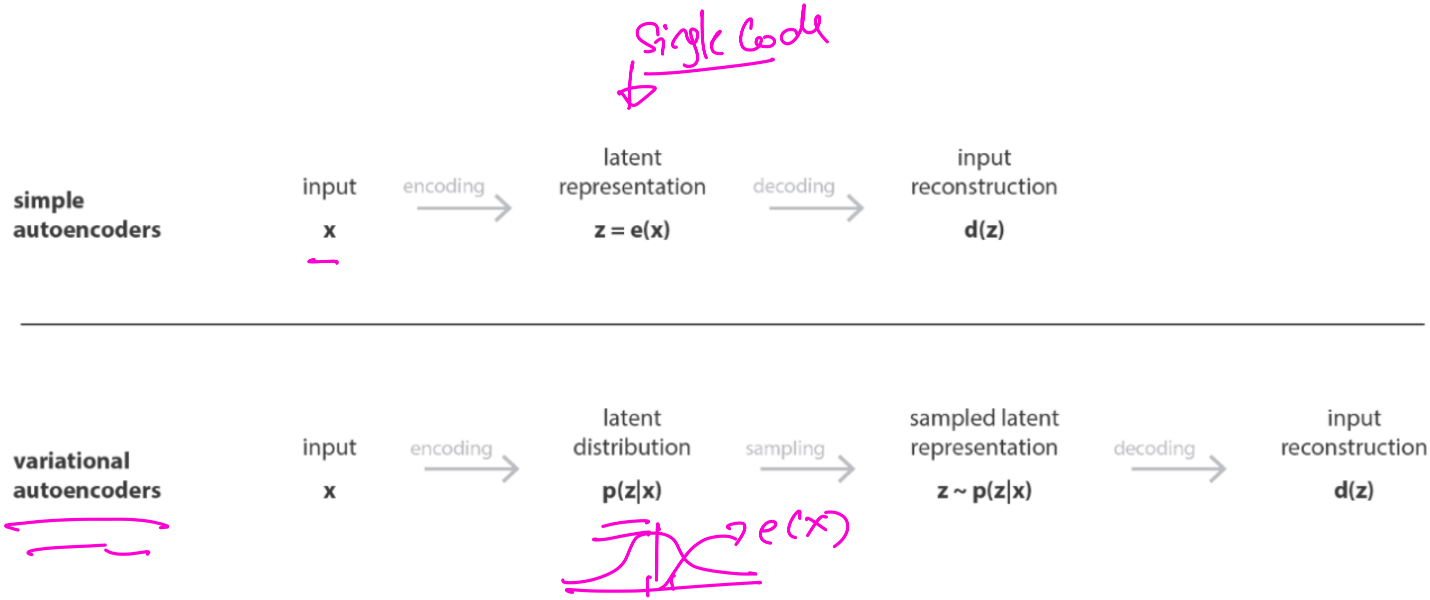


Variational Auto Encoders



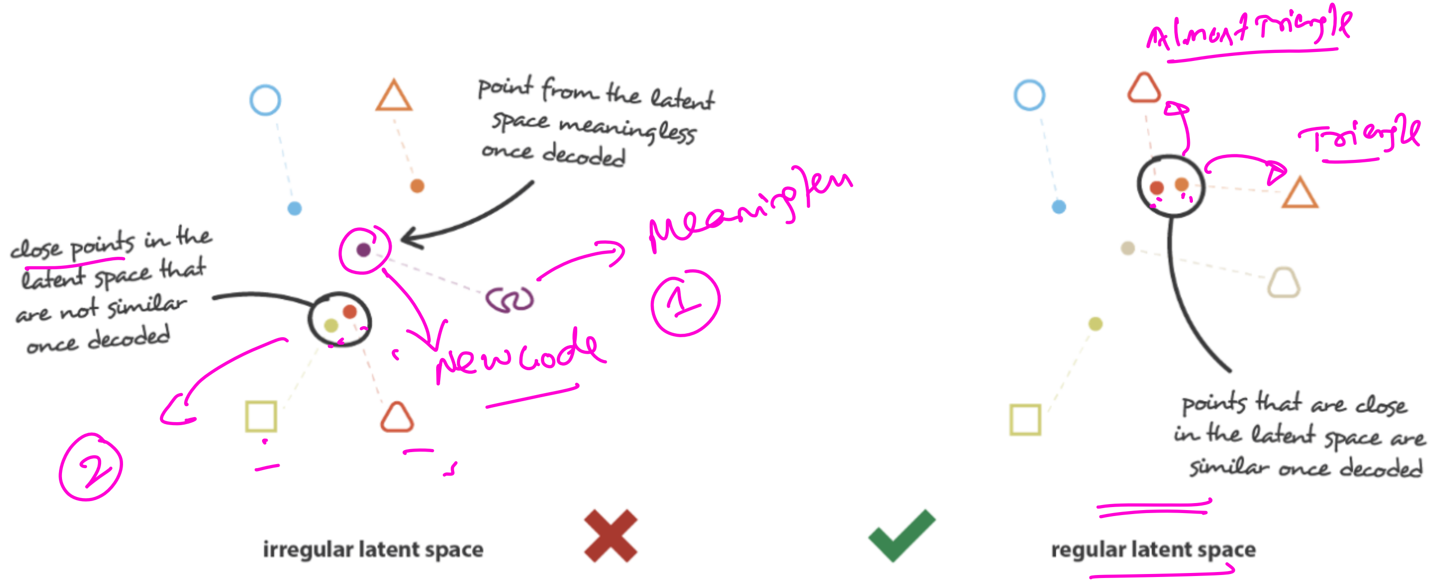
Irregular latent space prevent us from using autoencoder for new content generation.

Variational Auto Encoders



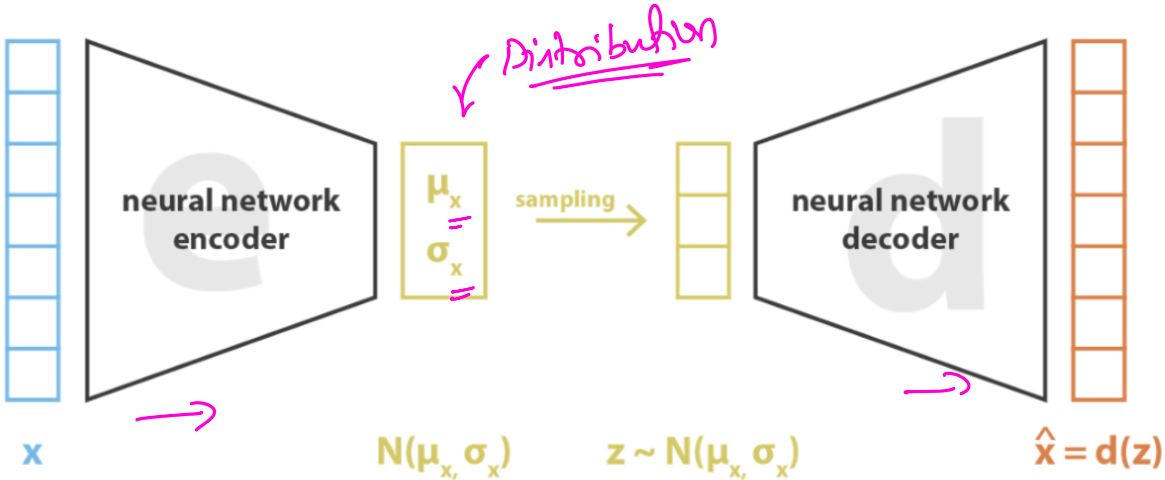
Difference between autoencoder (deterministic) and variational autoencoder (probabilistic).

Variational Auto Encoders



Difference between a "regular" and an "irregular" latent space.

Variational Auto Encoders



$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,] = \| x - d(z) \|^2 + \text{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,]$$

Breakouts Time #1

Usefulness of AEs

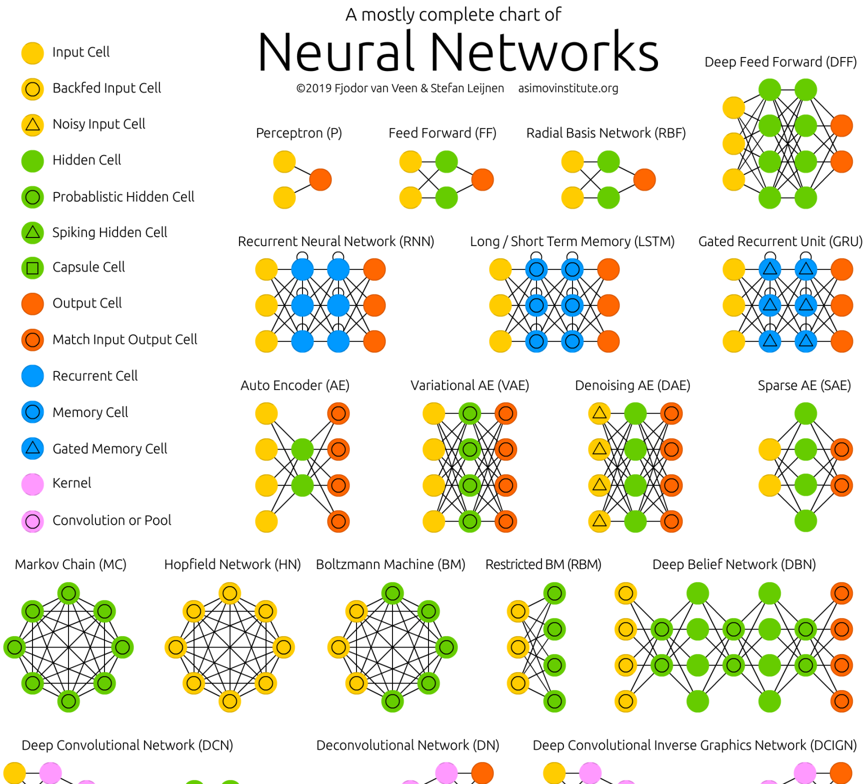
Discuss in your groups how any of the Auto Encoders can be helpful for anomaly detection in health care metrics (e.g heart rate) - E.g. Arrhythmia detection. or Medical Imaging

MRI Cat scan

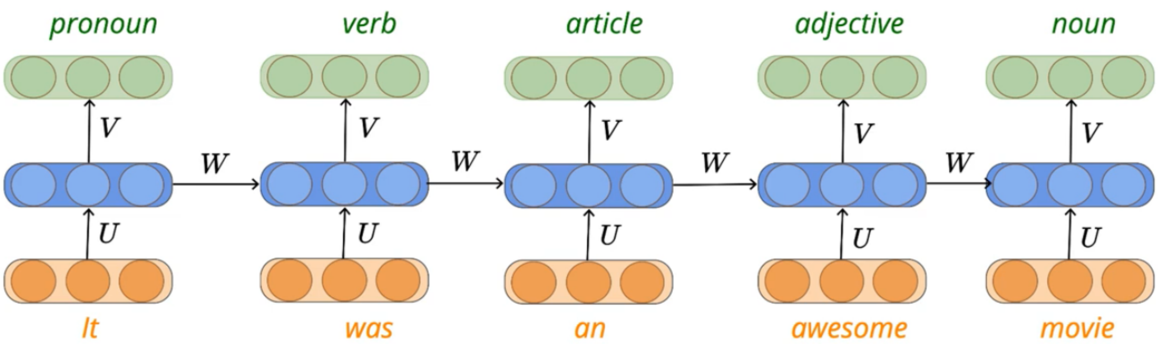
More DL Architectures

Neural Networks Zoo

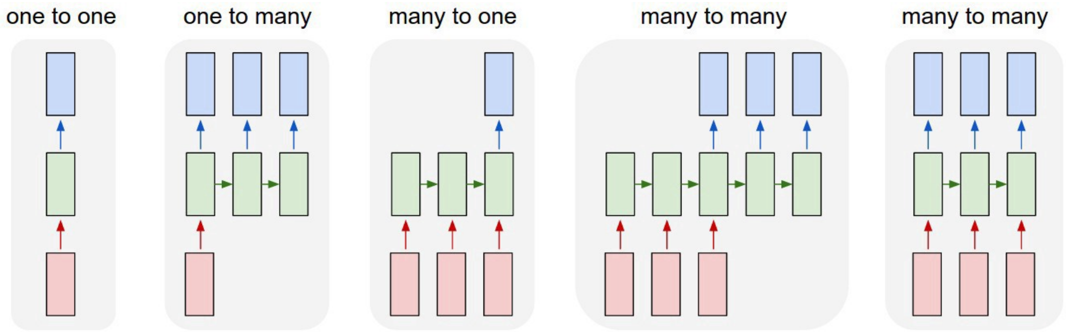
Zoo Reference



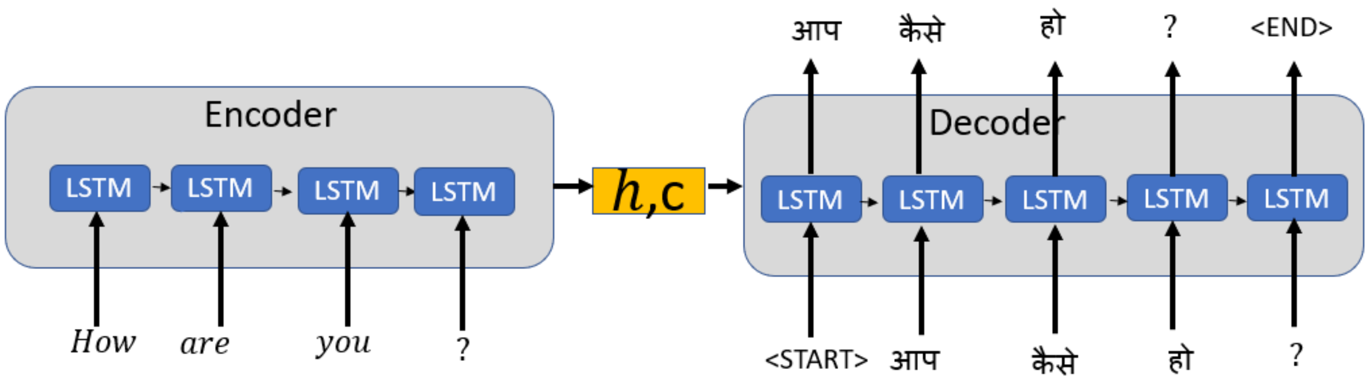
Sequence to Sequence Model (LSTM) Applications



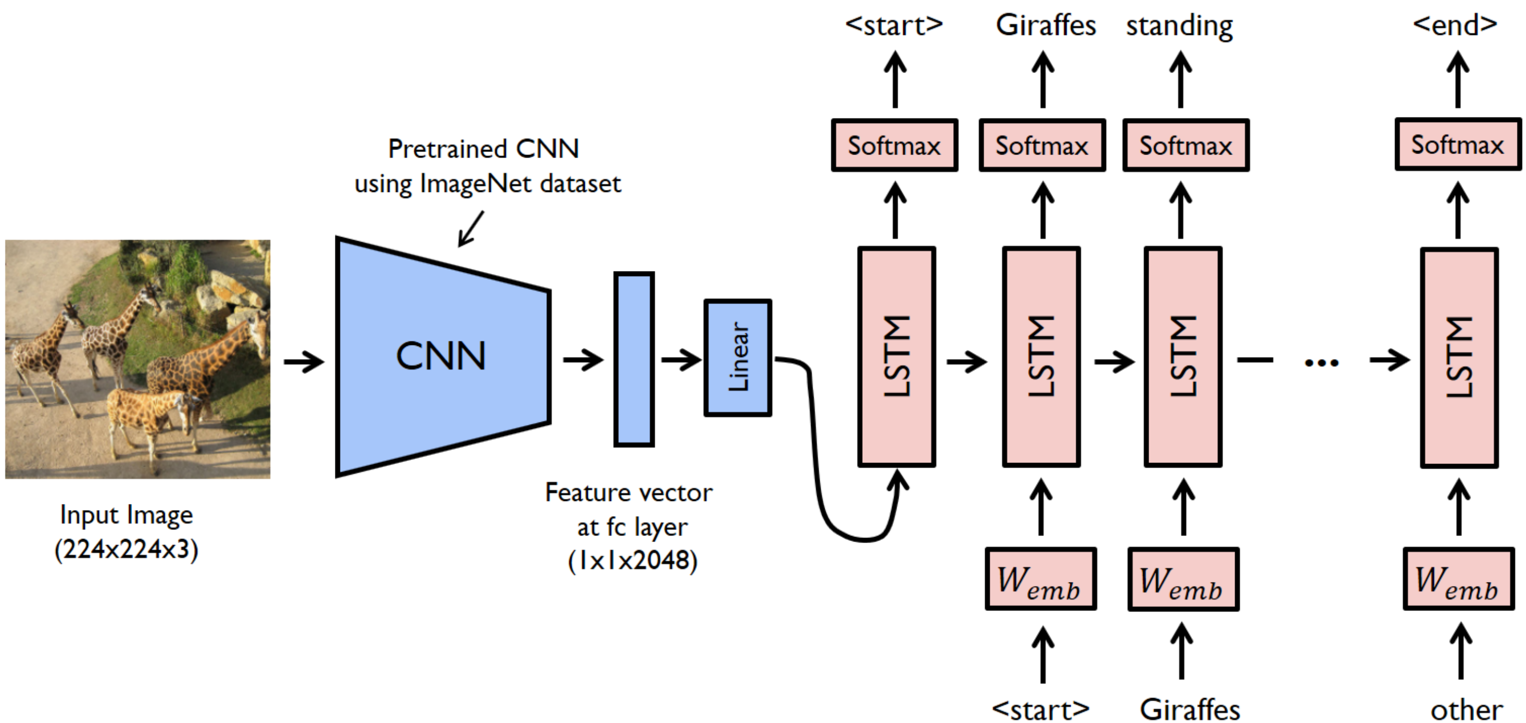
Sequence to Sequence Model (LSTM) Applications



Sequence to Sequence Model (LSTM) Applications



Sequence to Sequence Model (LSTM) Applications



Breakouts Time #2

Usefulness of LSTMs

Brainstorms the problems in health care that could benefit from the use of an LSTM model.

Time-series Anomaly Detector for KPIs

Framework for Anomaly Detection using Deep Learning

- Applies wherever there is a KPI (Key Performance Indicator) as a time-series.. esp with cloud applications. This holds true for Health care data on the cloud - HR, pulse, gluocse, etc.

Reference: KPI-TSAD: A Time-Series Anomaly Detector for KPI Monitoring in Cloud Applications

Time-series Anomaly Detector for KPIs

Framework for Anomaly Detection using Deep Learning

- Applies wherever there is a KPI (Key Performance Indicator) as a time-series.. esp with cloud applications. This holds true for Health care data on the cloud - HR, pulse, gluocse, etc.
- Uses **VAE** for over-sampling - I.e. **Data Augmentation** on the minority class (E.g. anomalies)

Reference: KPI-TSAD: A Time-Series Anomaly Detector for KPI Monitoring in Cloud Applications

Time-series Anomaly Detector for KPIs

Framework for Anomaly Detection using Deep Learning

- Applies wherever there is a KPI (Key Performance Indicator) as a time-series.. esp with cloud applications. This holds true for Health care data on the cloud - HR, pulse, gluocse, etc.
- Uses **VAE** for over-sampling - I.e. **Data Augmentation** on the minority class (E.g. anomalies)
- Can try this for Assignment 3 on Arrhythmia Detection to over-sample the anomalies using VAE

Reference: KPI-TSAD: A Time-Series Anomaly Detector for KPI Monitoring in Cloud Applications

Time-series Anomaly Detector for KPIs

Framework for Anomaly Detection using Deep Learning

- Applies wherever there is a KPI (Key Performance Indicator) as a time-series.. esp with cloud applications. This holds true for Health care data on the cloud - HR, pulse, gluocse, etc.
- Uses **VAE** for over-sampling - I.e. **Data Augmentation** on the minority class (E.g. anomalies)
- Can try this for Assignment 3 on Arrhythmia Detection to over-sample the anomalies using VAE
- Uses CNN + LSTM architecture to capture spatio-temporal and sequential features.

Reference: KPI-TSAD: A Time-Series Anomaly Detector for KPI Monitoring in Cloud Applications

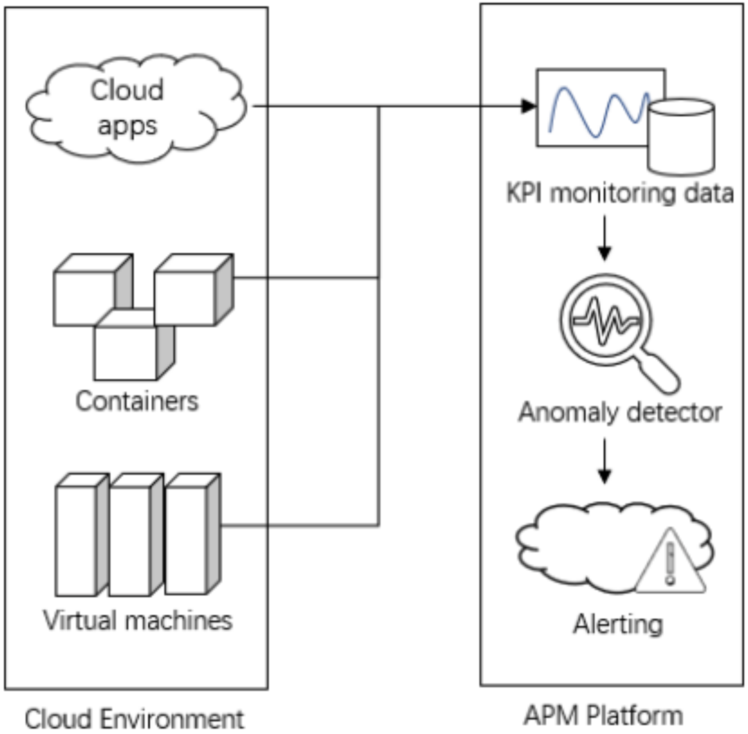
Time-series Anomaly Detector for KPIs

Framework for Anomaly Detection using Deep Learning

- Applies wherever there is a KPI (Key Performance Indicator) as a time-series.. esp with cloud applications. This holds true for Health care data on the cloud - HR, pulse, gluocse, etc.
- Uses **VAE** for over-sampling - I.e. **Data Augmentation** on the minority class (E.g. anomalies)
- Can try this for Assignment 3 on Arrhythmia Detection to over-sample the anomalies using VAE
- Uses CNN + LSTM architecture to capture spatio-temporal and sequential features.
- Beats baselines by a good margin.

Reference: KPI-TSAD: A Time-Series Anomaly Detector for KPI Monitoring in Cloud Applications

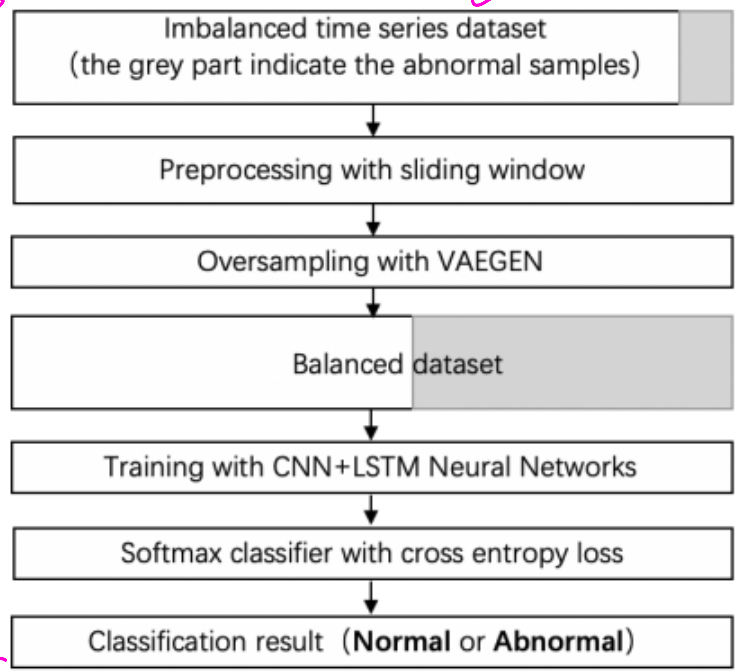
Time-series Anomaly Detector for KPIs



Reference: KPI-TSAD: A Time-Series Anomaly Detector for KPI Monitoring in Cloud Applications

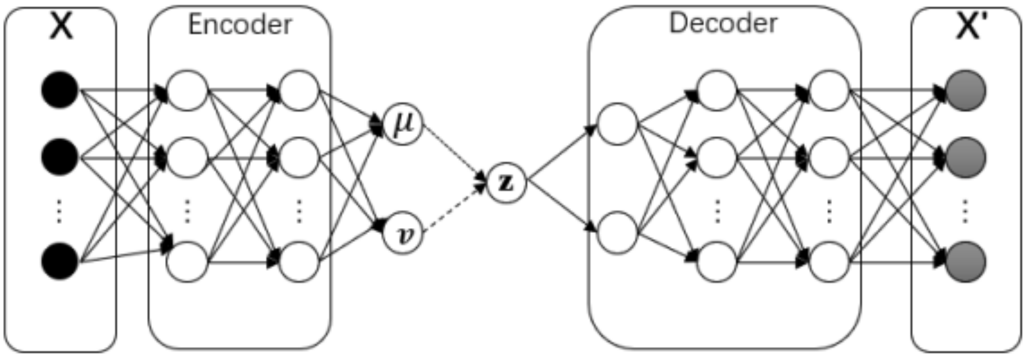
Anomaly Detection Recipe using DL for KPIs (Time-series Anomaly Detector for KPIs)

anomalies



VAE

VAE architecture (Time-series Anomaly Detector for KPIs)



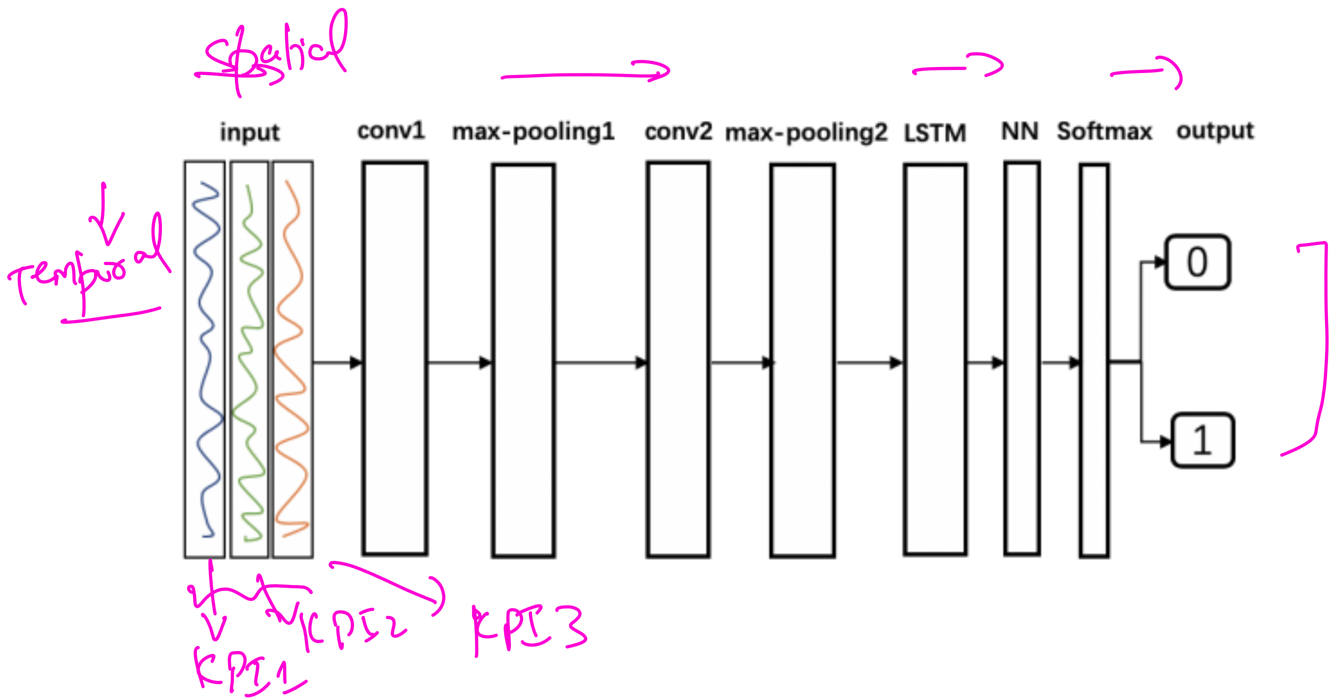
Over-sampling/Data Augmentation! (Time-series Anomaly Detector for KPIs)

Algorithm 3 VAEGEN: VAE based Oversampling algorithm.

Input: KPI values X, KPI labels Y, Oversampling rate R, The decoder D from VAE model
Output: KPI values X' after oversamling, KPI labels Y' after oversampling

$data\_size \leftarrow$ length of X
$augment\_size \leftarrow data\_size$ multiplied by R
$train\_X, valid\_X \leftarrow$ split X, Y as training set and validation set
$train\_X', valid\_X' \leftarrow$ filter the abnormal samples from $train\_X, valid\_X$ respectively
$D \leftarrow$ create a decoder by fitting M with $train\_X'$ and $valid\_X'$
$samples \leftarrow$ generate $augment\_size$ samples from a standard normal distribution
$augment\_data \leftarrow$ generate oversampling data by passing the $samples$ into the decoder D
$X' \leftarrow$ concatenate the $augment\_data$ and the original input X
$Y' \leftarrow$ concatenate the $augment\_size$ numbers of "1" with original input labels Y
return X', Y'

Overall Architecture for Anomaly Detection (Time-series Anomaly Detector for KPIs)



Summary

1. Recap on DL and DL architectures
2. Usefulness in Healthcare
3. Frameworks for Anomaly Detection
4. Data augmentation through VAE esp for the minority class

References

1. A review of ML and DL techniques for Anomaly Detection in IoT Data
2. KPI-TSAD: A Time-Series Anomaly Detector for KPI Monitoring in Cloud Applications