1st International Conference on Artificial Intelligence and Robotics (ICAIR 2021)

DEEP LEARNING NEURAL NETWORKS

Architectures and Applications

Dr. E. P. Fasina

Machine Intelligence Research Group (MIRG)

Department of Computer Sciences

University of Lagos

Neural Networks

Neural Networks have been studied for **78 years** (McCulloch & Pitts, 1943)

Multilayered Artificial Neural Networks can learn complex, complex non-linear functional mappings, given sufficient resources and training data – performance scales with training data

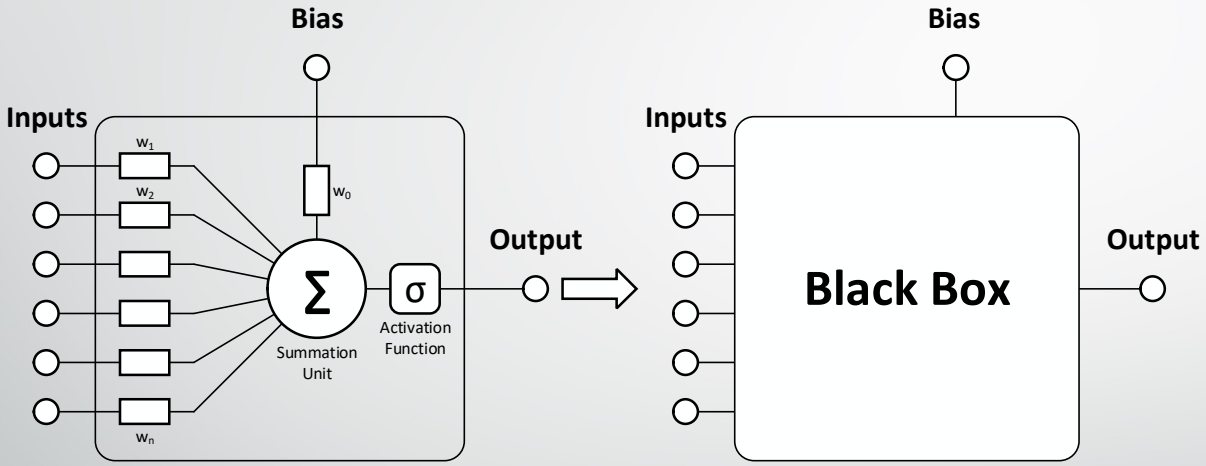
Are treated as BLACK BOX systems that can deliver excellent performance (surpassing Humans in several instances) in applications that require insights from **large**, **unstructured**, **high-dimensional data**

RECOMMENDED PAPER

Saptarshi Sengupta, Sanchita Basak, Pallabi Saikia, Sayak Paul, Vasilios Tsalavoutis, Frederick Atiah, Vadllamani Ravi, Alan Peters (2020) A Review of Deep Learning with Special Emphasis on Architectures, Applications, and Recent Trends, *Knowledge-Based Systems,* vol. 194, 105596

ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2020.105596

Perceptron, the Basic Unit of Artificial Neural Networks

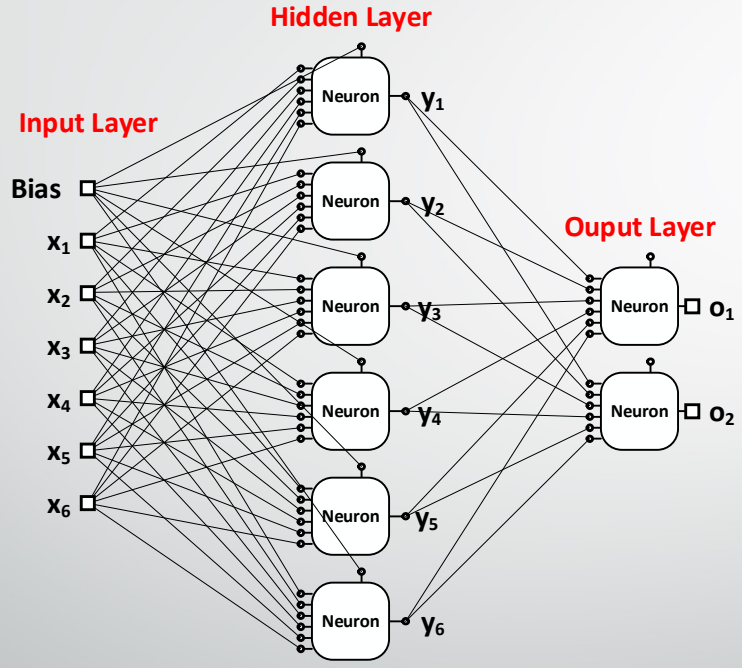


The Perceptron Learning Model

Artificial Neuron consists of:

1) Inputs + Bias
2) Weights
3) Summation Unit
4) Activation Function
5) Output

The Feedforward Artificial Neural Network Architecture



$$\sigma\left(\begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Learning – adjusting weights for Classification and Regression

1) Supervised
2) Semi-Supervised
3) Unsupervised
4) Reinforcement

Ways to Learning

- **Supervised Learning**

  Learning by examples or labelled data set (Input – Output pairs)

- **Semi-Supervised Learning**

  Learning with some labelled data in data set

- **Unsupervised Learning**

  Learning without labelled data. Algorithms must first discover any naturally occurring patterns in the training data set then learn

- **Reinforcement Learning**

  Intelligent agents learn to take actions in an environment in order to maximize cumulative reward

- **Weakly Supervised Learning**

  Noisy, limited, or imprecise sources are used to provide supervision signal for labeling large amounts of training data in a supervised learning setting

- **Active Learning**

  Learn from data collected from users through queries. Queries are based on unlabeled data

Key Contributions from ANN to Deep NN

- **McCulloch & Pits (1943)** – ANN model with adjustable weights
- **Rosenblatt (1957)** – Perceptron learning algorithm
- **Werbos (1974)** – Backpropagation
- **Hopfield (1982)** – Hopfield Networks
- **Rumelhart et al. (1986)** – Multilayer backpropagation
- **Vapnik, Cortes (1995)** – Support Vector Machines
- **Hochreiter & Schmidhuber (1997)** – Long Short Term Memory Networks
- **LeCunn et al. (1998)** – Convolutional Neural Networks
- **Hinton & Rusian (2006)** – Hierarchical Feature Learning in Deep Neural Networks

What is the "Deep" in Deep Learning?

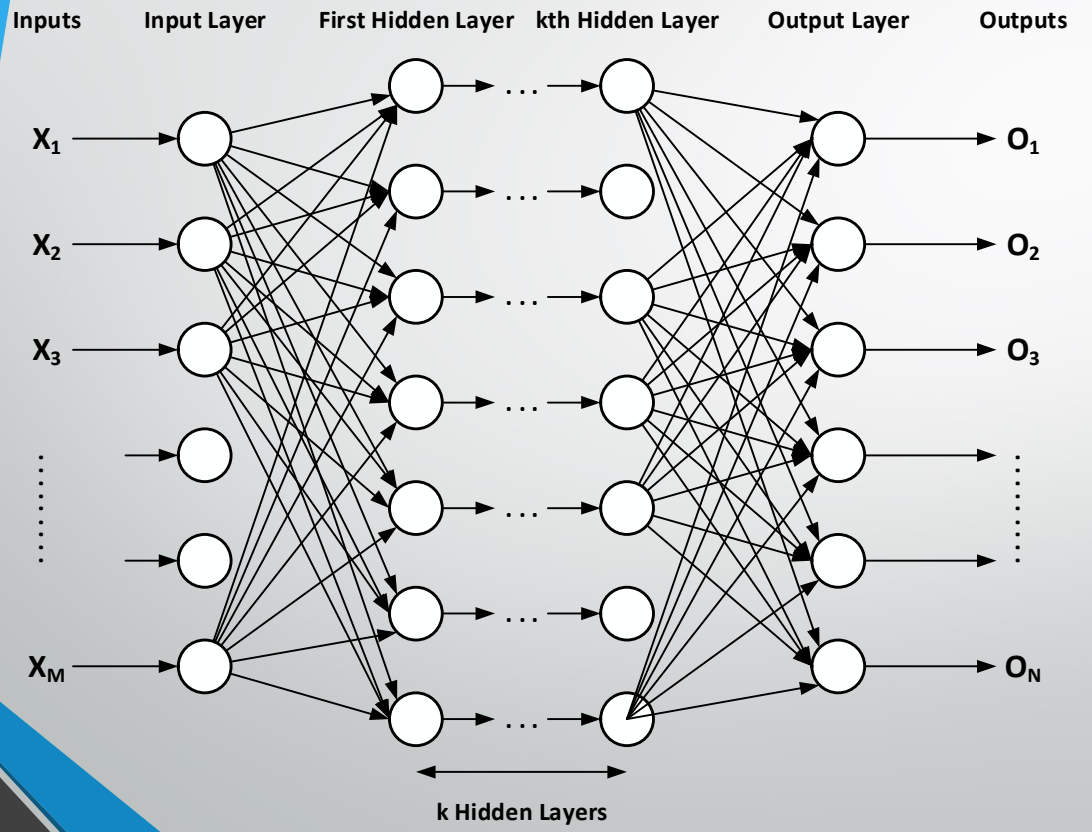
- Describes artificial neural networks with many HIDDEN layers with a large number of ADJUSTABLE weights
- The large number of weights allow **Deep Learning Networks** to learn detailed representation of data and generalized more accurately when classifying or fitting data.
- **The large number of weight also allows it to scale better with large training data sets.**

Why the Rapid Development and Rise of Deep Learning?

- Availability of large training data set
- **Advances in High Performance Computing** – Multicore, Multiprocessors, Parallel Computers
- **Open Platforms** – PyTorch, Tensorflow, Caffe, Chainer, Keras, BigDL etc.
- **Improved Regularization Techniques** – Batch Normalization, Dropout, Data Augumentation, Early Stopping etc.
- **Robust Optimization** – Adaptive Learning Rates (AdaGrad, RMSProp, Adam, Adaboost), Stochastic Gradient Descent (with Standard Momentum, Nesterov Momentum), Particle Swarm Optimization, Differential Evolution, Genetic Algorithms etc.

Deep Feedforward Networks - 1

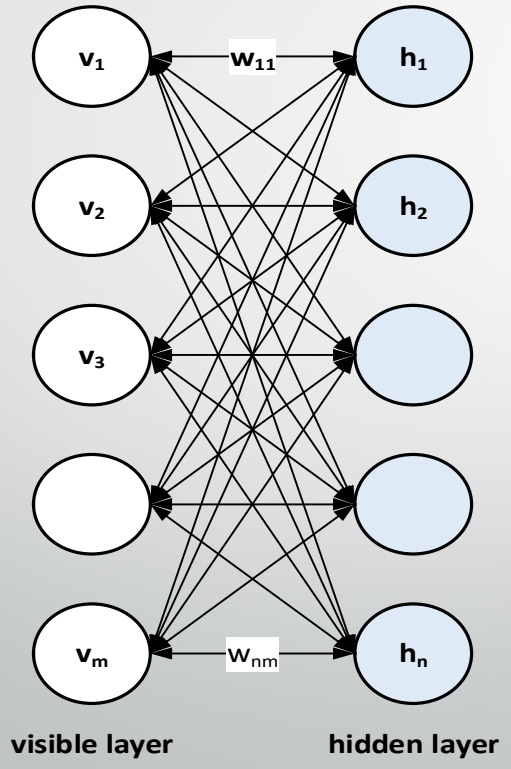
Y. Bengio, et al., Learning Deep Architectures for AI, Trends Mach. Learn. 2 (1) (2009) 1–127.



1) Can model complex nonlinear relations more efficiently than shallow architecture using supervised or unsupervised training algorithms
2) Hierarchical learning possible because of multiple levels of nonlinearity
3) Usually trained with Backpropagation using gradient descent in backward pass
4) May surfer from overfitting, trapped in local minima, vanishing gradient etc.
5) Performance enhancing techniques – L1 and L2 regularization, dropout, batch normalization, weight initialization techniques, good activation functions etc.

Restricted Boltzmann Machines - 1

P. Smolensky, Information Processing in Dynamical Systems: Foundations of Harmony Theory, Tech. Report, Colorado University at Boulder Department. of Computer Science, 1986.



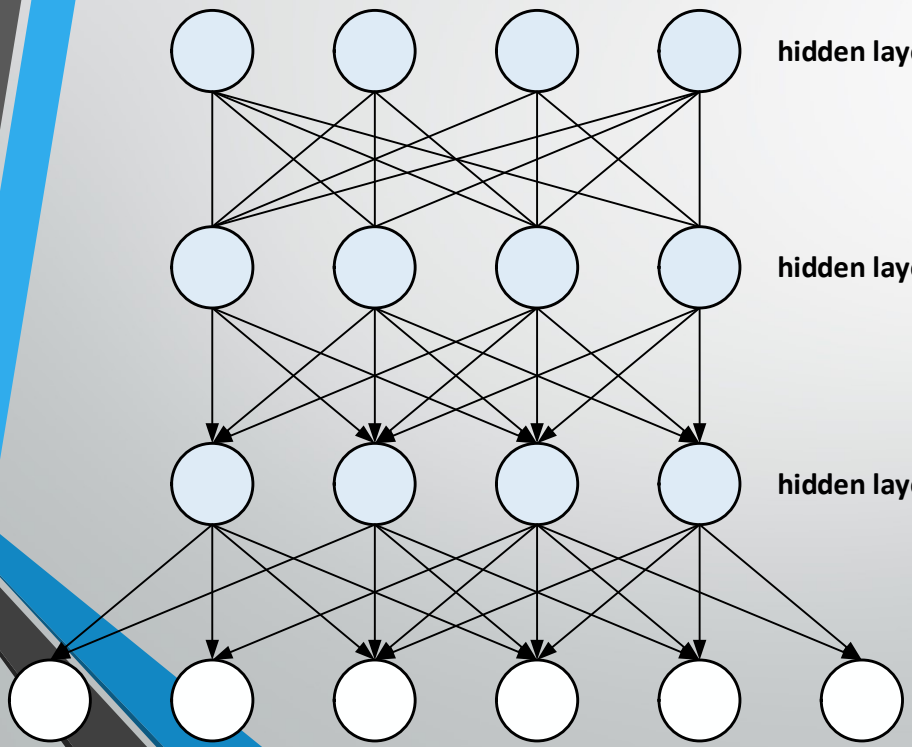
1) May be taken as a Stochastic Neural Network
2) Bipartite graph of visible and hidden layers of artificial neurons
3) Can learn the input probability distribution in a supervised and unsupervised manner
4) Variation of the Boltzmann machine with the restriction in the intra-layer connection between units
5) Basic RBM used Bernoulli units
6) The (gradient-based) contrastive divergence algorithm (Hinton, 2002) is the most popular training algorithm

Restricted Boltzmann Machines - 2

- Some variations of RBM are **Deep RBMs**, **Temporal RBMs** & **Conditional RBMs** [used for modeling **multivariate time series data** and generate **motion captures**] **Gated RBMs** [used to learn transformation between two images] **Convolution RBMs** [used to understand time structure of time series] Mean-Covariance RBMs [for representing covariance structure of data] **Recurrent Temporary RBMs** and **Factored Conditional RBMs**
- Applied to solve diverse tasks as **Representation Learning, Dimensionality Reduction, Classification, Regression, Prediction, Building Block of Deep Belief Network, Topic Modeling, Collaborative Filtering etc.**

Deep Belief Networks - 1

G.E. Hinton, S. Osindero, and Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Computation. 18 (7) (2006) 1527–1554.



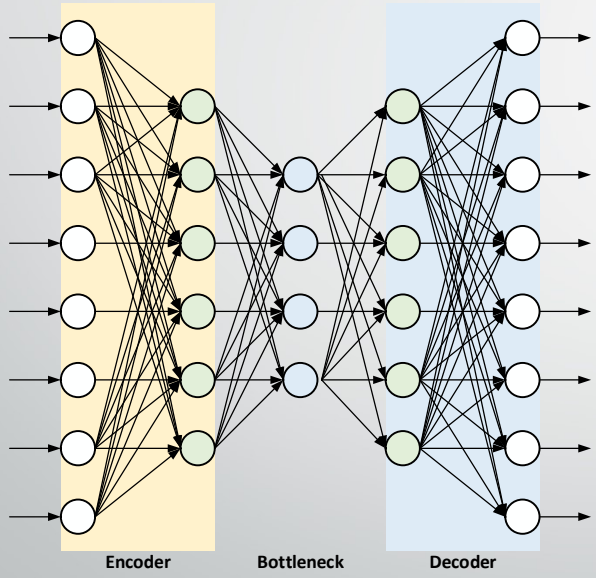
1) Generative graphical model of multiple layers of latent variables. Latent variables are typically binary and represent hidden features present in input observations
2) Stacked RBM
3) Hinton (2006) proposed a fast efficient way of training DBNs
4) Log-probability of training data can be improved by adding layers to the network which increases the true representational power of the network
5) Applied as discriminative models by appending a discrimination layer at the end and fine-tuning the model using target labels

Deep Belief Networks - 2

- **Variants** of Deep Belief Networks are **Convolutional DBN** and **Discriminative DBN**
- Has been used as **initial models** in **Dimensionality Reduction**, **Feature Extraction**, **Feature Learning**, **Feature Generation**, **Classification** [Phone recognition, Handwritten digit recognition, Computer vision, Speech recognition] **Pretraining** [Deep Neural Networks, Deep Convolutional Neural Networks] **Prediction** [Time series prediction, Traffic prediction]

Autoencoders - 1

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning Internal Representations by Error Propagation. Parallel Distributed Processing. Vol 1: Foundations. MIT Press, Cambridge, MA, 1986.



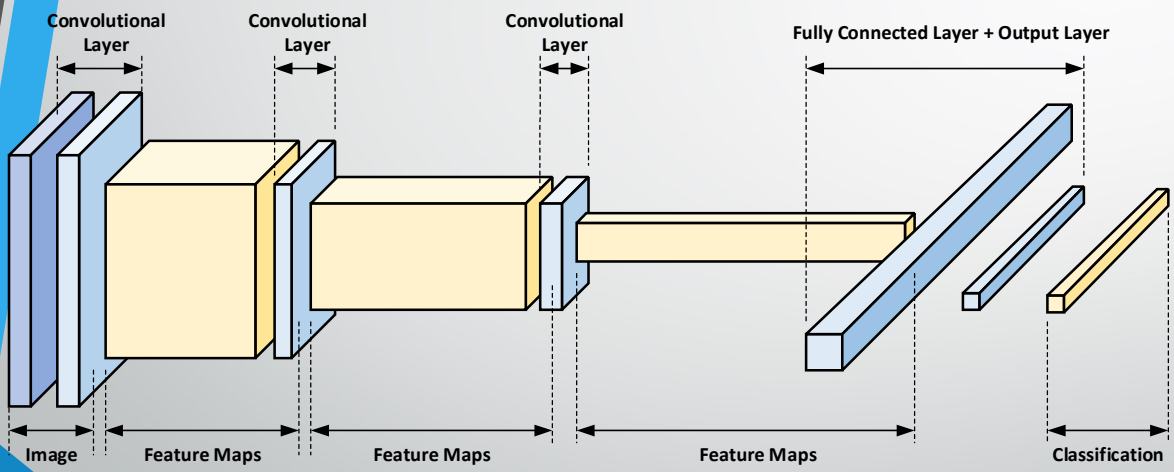
1) Tries to reconstruct its input at the output
2) It is trained using the backpropagation algorithm
3) The encoder generates a representation of the input as the output of the hidden layer with the smallest number of artificial neurons
4) The decoder reconstructs the input at the output from the internal representation.
5) The unsupervised pretraining of using autoencoders have been successfully applied to solve many problems
6) The have become popular as generative models in recent years

Autoencoders - 2

- **Variants** of autoencoders are Undercomplete AE, Sparse AE, Contractive AE, Denoising AE, Variational AE, Transforming AE, Concrete AE…
- Autoencoders have been applied in many fields such as **Dimensionality Reduction**, **Principal Component Analysis**, **Information Retrieval**, **Anomaly Detection, Image Processing, Drug Discovery, Popularity Prediction, Multimodal Learning in Speech and Video Images, etc.**

Convolutional Neural Networks - 1

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.



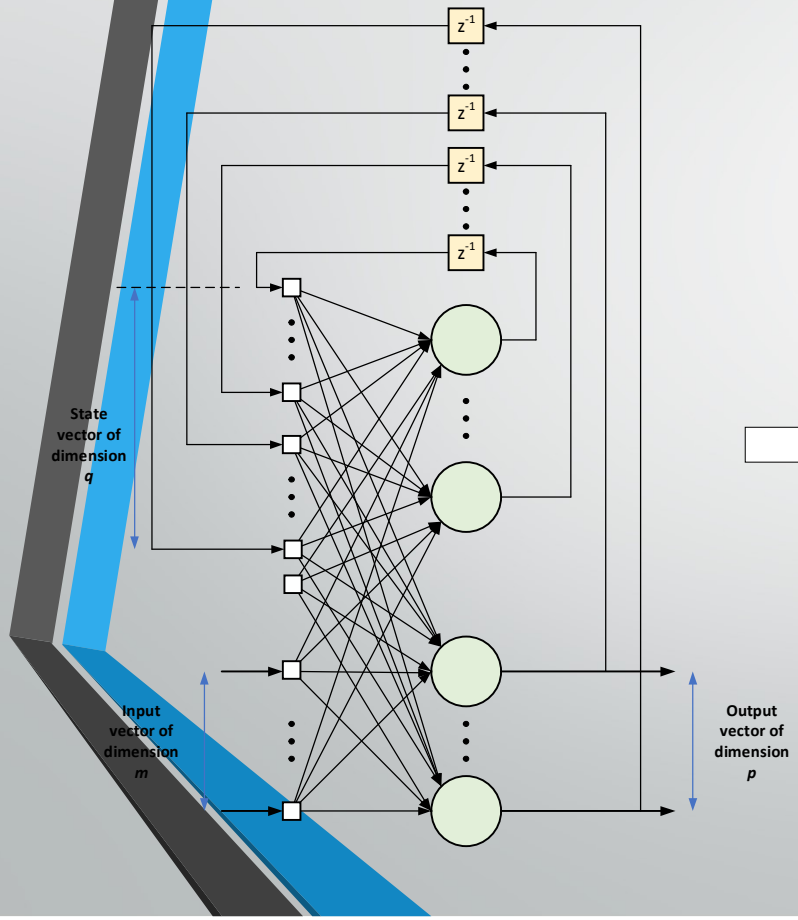
1) Inspired by the human visual system
2) Consists of multiple layers of convolutional and pooling layers of artificial neurons and a fully connected layer
3) Extracts image descriptors (edges, contours, strokes, textures, gradients, orientation and color) using latent spatial information

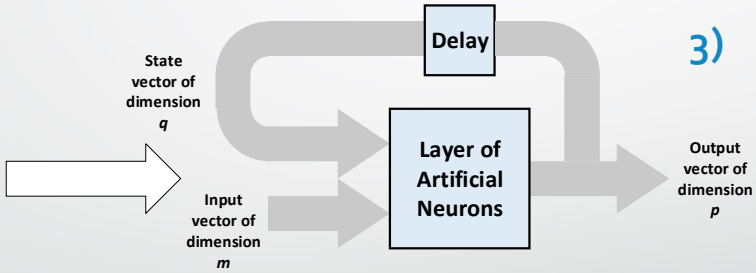
Convolutional Neural Networks - 2

- **Popular architectures** of **naïve CNN** include **LeNet**, **AlexNet**, **VGGNet**, **GoogleNet**, **ResNet**, **ZFNet** and so on.
- **Variants** include **Region-Based CNN**, **Fast R-CNN**, **Faster R-CNN**, **YOLO**, **Single-Shot Multi-box Detector** (SSMD) etc.
- **Applications** include **Image Detection**, **Image Segmentation**, **Image Classification**, and **Image Super-Resolution Construction**.

Recurrent Neural Networks - 1

Fully Connected RNN



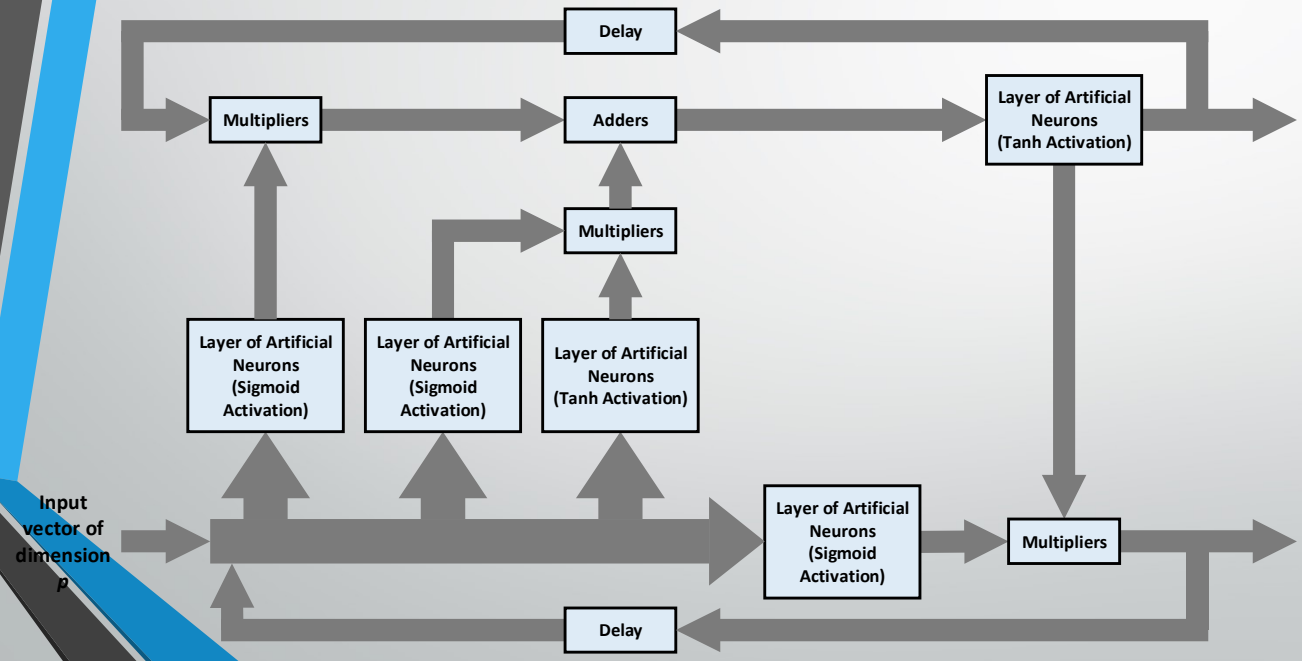


1) Can model time dependencies
2) Better than Hidden Markov Models (HMMs) at learning long term dependencies
3) During the propagation of errors across multiple timesteps the problem of vanishing and exploding gradients (instabilities due to feedbacks) occur
4) Instabilities can handled by another variant of the RNN, the Long Short Term Memory Neural Network

Recurrent Neural Networks – 2

Long Short-Term Memory (LSTM) RNN

S. Hochreiter, J. Schmidhuber, Long Short-term Memory, Neural Computation, MIT Press, 9 (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.



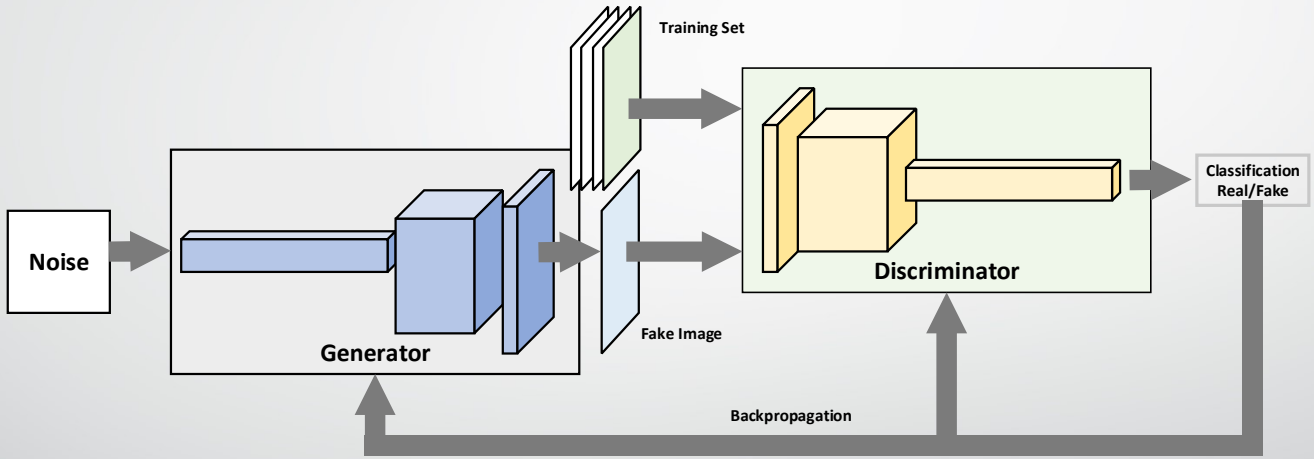
1) Solves the problem of vanishing and exploding gradients in RNNs
2) Amount of information to be retained from previous time steps is controlled by a sigmoid layer known as forget gate
3) The sigmoid activated input gate decides on the new information to be stored in the cell
4) Hyperbolic tangent layer produce new candidate values which is updated taking forget gate coefficient weighted old state candidate value

Recurrent Neural Networks – 3

- Variants of LSTM are **LSTM without a Forget Gate**, **LSTM with a Forget Gate**, **LSTM with Peephole**, **Gated Recurrent Unit (GRU)**, **Bidirectional-LSTM**, **Minimal Gated Unit (MGU)**, **Deep Recurrent Attentive Writer**, **Grid Long Short-Term Memory** **etc.**
- Applications of RNNs include **Prediction**, **Language Modelling and Generating Text**, **Machine Translation**, **Speech Recognition**, **Generating Image Descriptions**, **Video Tagging**, **Text Summarization**, **Call Center Analysis**, **Face detection**, **OCR Applications as Image Recognition**, **Music Composition etc.**

Generative Adversarial Networks – 1

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, Y. Bengio, Generative adversarial nets, in: NIPS, 2014



1) Simultaneous training of generative and discriminative models
2) Generative model tries to capture the data distribution while the discriminative model learns to estimate the probability of a sample coming from data or the generative model
3) If both models are multilayer perceptron only backpropagation and dropout is required to train them
4) The goal is to train the generative model so that the discriminative model makes a mistake
5) Training stops the discriminator has 50/50 chance of making a mistake – training is a minimax two player game

Generative Adversarial Networks – 2

- **Variants** are Conditional GAN, Deep Convolution GAN, Laplacian GAN, Information Maximizing GAN, Energy-Based GAN, Wasserstein GAN, Boundary Equilibrium GAN, Progressive-Growing GAN, Big GAN, Style-Based, etc.
- Some impressive **applications** are **Generate Examples for Image Datasets, Generate Photographs of Human Faces, Generate Realistic Photographs, Generate Cartoon Characters, Image-to-Image Translation, Text-to-Image Translation, Semantic-Image-to-Photo Translation, Face Frontal View Generation, Generate New Human Poses, Photos to Emojis, Photograph Editing, Face Aging, Photo Blending, Super Resolution, Photo Inpainting, Clothing Translation, Video Prediction, 3D Object Generation**

Swarm Intelligence in Deep Learning

- Evolutionary Computation, Particle Swarm Optimization, Quantum Dirac-Delta Swarm
- Algorithms used mainly to **tune network parameters** to **optimize the design** of complex neural network architectures and learning process
- Optimize **topology parameters** (number of layers and number of neurons) neuron parameters (synaptic weights and transfer function) and hyperparameters (learning and data augmentation parameters) of each layer

Broad Areas of Application

- **FRAUD DETECTION** in Financial Services – Banking, Insurance
- Financial Time Series **FORECASTING**
- Prognostics and Health **MONITORING**
- (Medical) IMAGE PROCESSING
- POWER SYSTEMS e.g., Load Forecasting, Energy Consumption Estimation
- RECOMMENDER SYSTEMS
- MANUFACTURING - Design of Products and Processes
- COMPUTER VISION
- ADVERSARIAL LEARNING
- AUTONOMOUS VEHICLES
- NATURAL LANGUAGE PROCESSING
- BIG DATA ANALYTICS

Deepkurtex Deep Learning Software Library - 1

- Open Platforms – **PyTorch, Tensorflow, Theano, Microsoft Cognitive Toolkit, Caffe, Chainer, Keras, BigDL**
- **Why Deepkurtex?**
  - **To gain insights into the architecture and inner workings of Deep Learning NNs**
  - **Suitable for learning and design of projects**
- **What have we achieved so far?**
  - Object-oriented – we use .NET & C#
  - Three models – **Graph** (Object-Oriented), **Matrix** and **Tensor**
  - Optimize Matrix models with BLAS (Basic Linear Algebra Subroutines) to get Tensor models
  - We have implemented – Deep FFNN, Convolutional Neural Networks, RNNs – all popular versions of LSTM, GRU, MGU
  - Pytorch is about 20 time faster than our Optimized RNN Models and about 16 times smaller in terms of memory requirements
  - All experiments were conducted on Azure

Deepkurtex Deep Learning Software Library - 2

- **What Next?**

  Build **Autoencoders**, **Generative Adversarial Networks** and **Deep Belief Networks** (in the listed order)

  Improve design of models

  Improve optimization

  Adopt Automatic Differentiation for training

  Implement new training strategies

  Use CUDA

  Make it Open Source?