Medical Image Processing

Deep Learning for Medical Image Processing

Professor Dr. Rafiqul Islam

Outline

- Introduction to Machine Learning
- The example application
- Machine Learning Models

Deep Learning

1. **what exactly is deep learning** ?

And,

2. **why is it generally better** than other methods on image, speech and certain other types of data?

The short answers

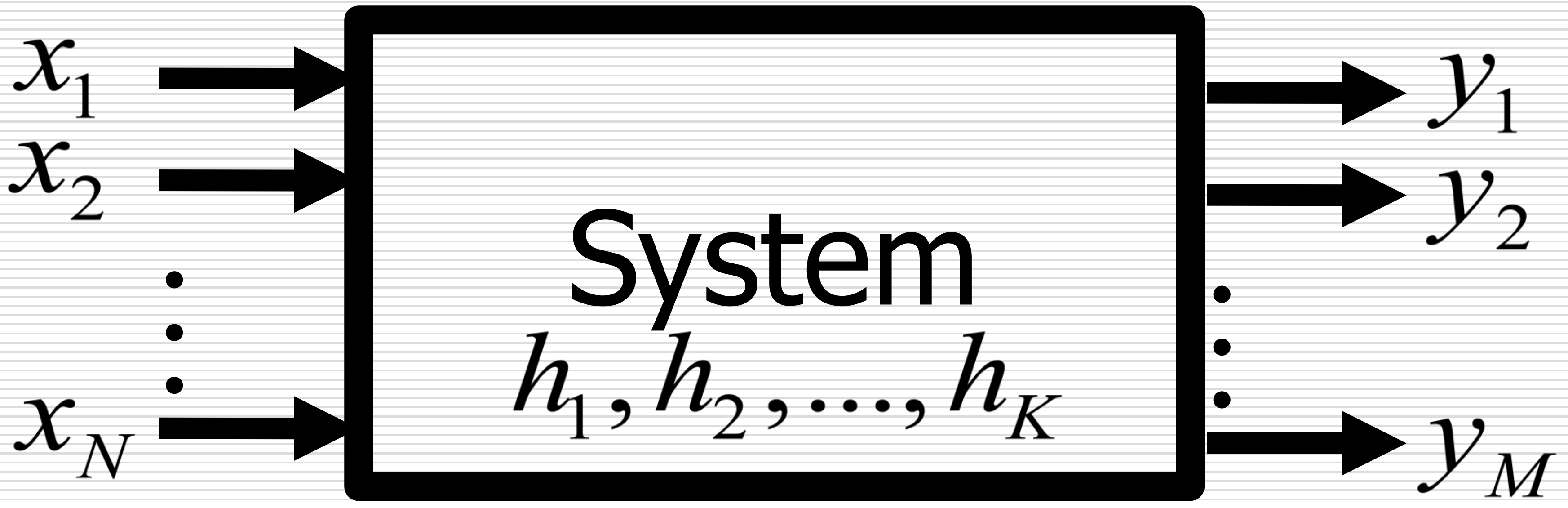
1. **'Deep Learning' means** using a **neural network** with **several layers of nodes** between input and output.

2. The series of layers between input & output do feature identification and processing in a series of stages, just as our brains seem to.

What is Machine Learning

- Machine Learning (ML) is constructing computer programs that develop solutions and improve with experience
- Solves problems which can not be solved by enumerative methods or calculus-based techniques
- Intuition is to model human way of solving some problems which require experience
- When the relationships between all system variables is completely understood ML is not needed

A Generic System



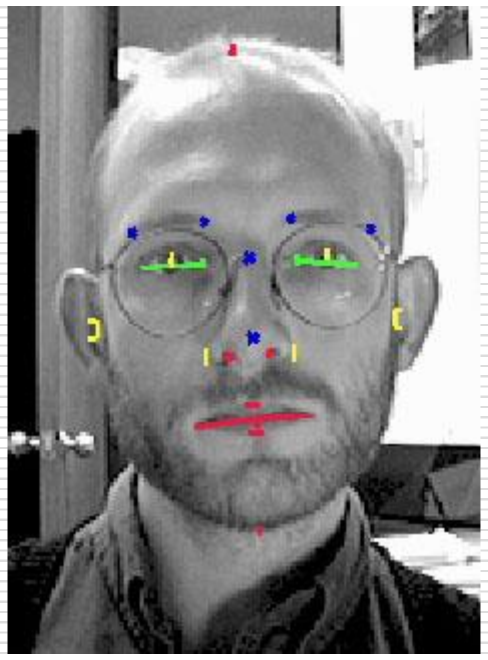
Input Variables: $\mathbf{x} = \left(x_1, x_2, \ldots, x_N\right)$

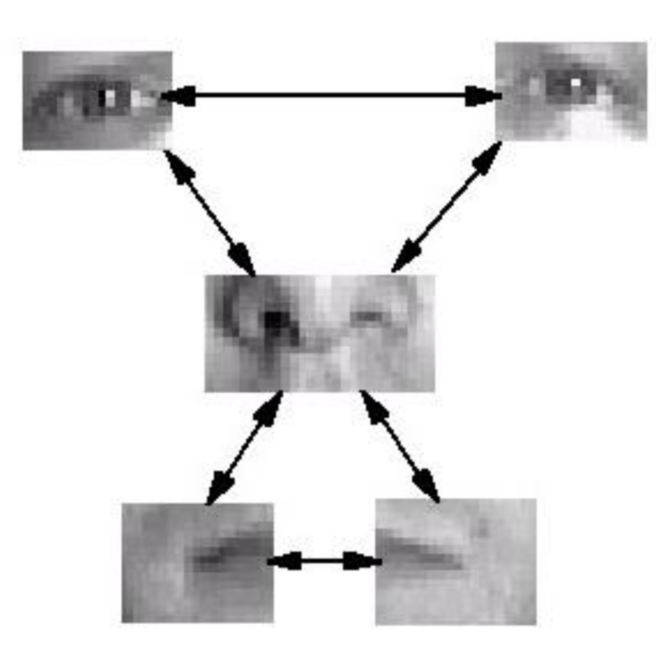
Hidden Variables: $\mathbf{h} = \left(h_1, h_2, \ldots, h_K\right)$

Output Variables: $\mathbf{y} = \left(y_1, y_2, \ldots, y_K\right)$

Learning Task

- Face recognition problem: Whose face is this in the picture?
- Hard to model describing face and its components
- Humans recognize with experience: The more we see the faster we perceive.



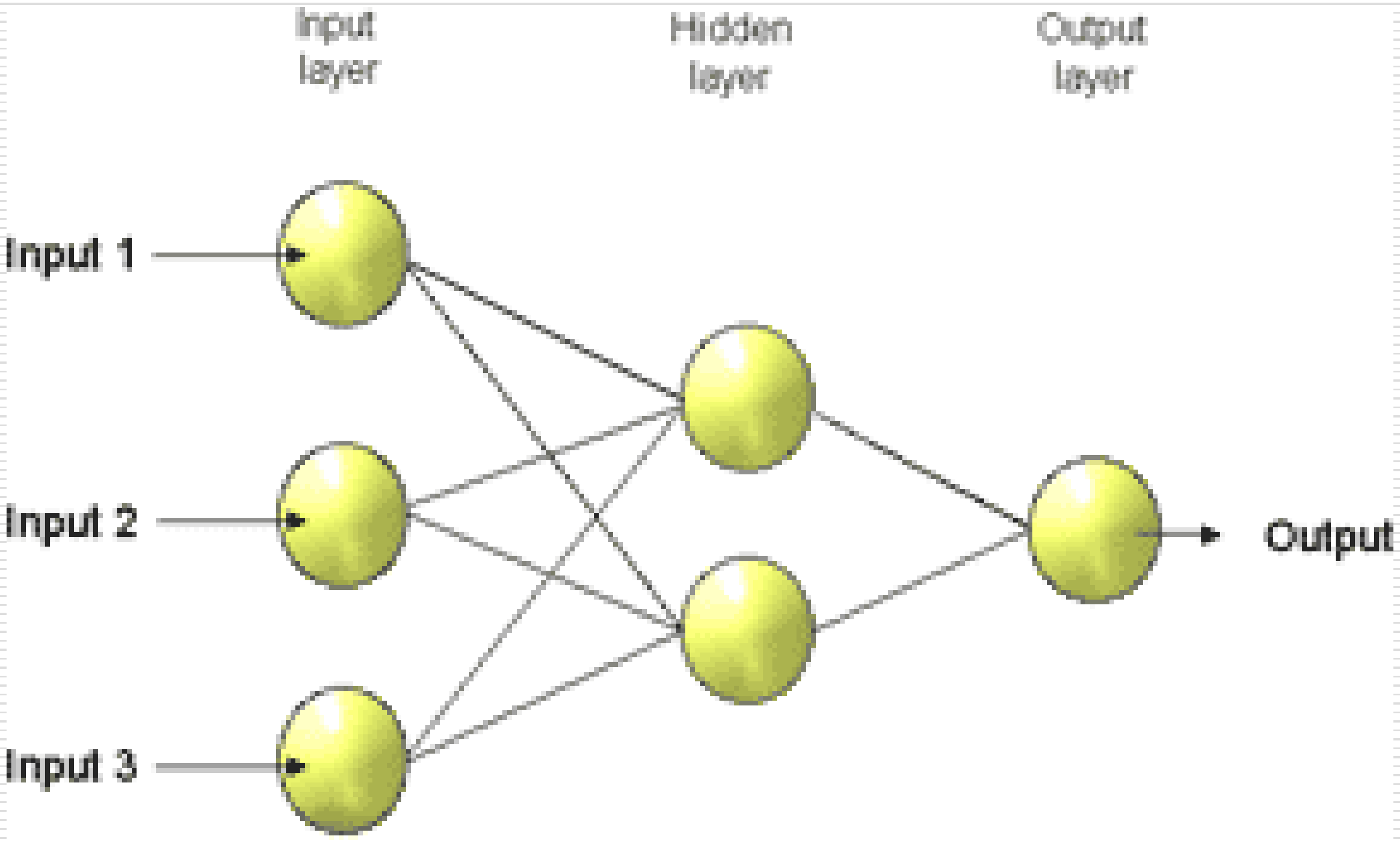


Deep Learning Applications

- Computer Vision, AI
- Image classification/ Reconstruction/ Super-resolution
- Video object segmentation and tracking
- Activity modeling/detection/recognition
- Video understanding
- Signal Processing
- Non-linear signal processing
- Learned image restoration, super-resolution
- Learned image/video compression

Deep Learning Process

- Made up of interconnected processing elements which respond in parallel to a set of input signals given to each



Deep Learning Process

	Use a Pretrained Network for Transfer Learning	**Create a New Deep Network**
Training Data	Hundreds to thousands of labeled images (small)	Thousands to millions of labeled images
Computation	Moderate computation (GPU optional)	Compute intensive (requires GPU for speed)
Training Time	Seconds to minutes	Days to weeks for real problems
Model Accuracy	Good, depends on the pretrained model	High, but can overfit to small data sets

Deep Learning Layers

Layers

- Input Layers
- Hidden Layers
- Output Layers

Input Layers

- **imageInputLayer([28 28 1])** -- An imageInputLayer is where you specify the image size, which, in this case, is 28-by-28-by-1. These numbers correspond to the height, width, and the channel size.

Hidden Layers

- **convolution2dLayer --** Convolutional Layer In the convolutional layer, the first argument is filterSize, which is the height and width of the filters the training function uses while scanning along the images
- **batchNormalizationLayer --** Batch normalization layers normalize the activations and gradients propagating through a network, making network training an easier optimization problem. Use batch normalization layers between convolutional layers and nonlinearities, such as ReLU layers, to speed up network training and reduce the sensitivity to network initialization
- **reluLayer --** ReLU Layer is a nonlinear activation function
- **maxPooling2dLayer --** Max Pooling Layer Convolutional layers (with activation functions) are sometimes followed by a down-sampling operation that reduces the spatial size of the feature map and removes redundant spatial information.

Output Layers

- **fullyConnectedLayer --** A fully connected layer is a layer in which the neurons connect to all the neurons in the preceding layer. This layer combines all the features learned by the previous layers across the image to identify the larger patterns. The last fully connected layer combines the features to classify the images.
- **softmaxLayer --** The softmax activation function normalizes the output of the fully connected layer. The output of the softmax layer consists of positive numbers that sum to one, which can then be used as classification probabilities by the classification layer
- **ClassificationLayer --** This layer uses the probabilities returned by the softmax activation function for each input to assign the input to one of the mutually exclusive classes and compute the loss
- **regressionLayer** -- A regression layer computes the mean-squared-error loss for regression problems.

Reconstruction Model

- networkDepth = 20;
- **firstLayer** = imageInputLayer([41 41 1],’Name','InputLayer');
- convLayer = convolution2dLayer(3,64,'Padding',1, 'Name','Conv1');
- relLayer = reluLayer(‘Name','ReLU1');
- **middleLayers** = [convLayer relLayer]
- *for layerNumber = 2:networkDepth-1*

    convLayer = convolution2dLayer(3,64,'Padding',[1 1], 'Name',['Conv’ num2str(layerNumber)]);

    relLayer = reluLayer('Name',['ReLU' num2str(layerNumber)]);

    middleLayers = [middleLayers convLayer relLayer];
- *End*
- convLayer = convolution2dLayer(3,1,'Padding',[1 1], 'NumChannels',64,'Name',['Conv' num2str(networkDepth)]);
- **finalLayers** = [convLayer regressionLayer(‘Name','FinalRegressionLayer')];
- layers = [firstLayer middleLayers finalLayers];

Classification Model

```
• layers = [
•    imageInputLayer([28 28 1])
•    convolution2dLayer(3,8,'Padding','same')
•    batchNormalizationLayer
•    reluLayer
•    maxPooling2dLayer(2,'Stride',2)
•    convolution2dLayer(3,16,'Padding','same')
•    batchNormalizationLayer
•    reluLayer
•    maxPooling2dLayer(2,'Stride',2)
•    convolution2dLayer(3,32,'Padding','same')
•    batchNormalizationLayer
•    reluLayer
•    fullyConnectedLayer(10)
•    softmaxLayer
•    classificationLayer];
```