Understanding the history of pre-trained models

LeNet-5

LeNet-5

- Proposed in 1998

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

Abstract—

Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are

I. Introduction

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of learning techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

LeNet-5

- Proposed in 1998

- Handwritten and Machine-printed character recognition

LeNet-5

- Proposed in 1998
- Handwritten and Machine-printed character recognition
- Simple and straightforward architecture

Architecture: LeNet-5

Architecture: LeNet-5

- LeNet-**5**, since it has 5 layers with learnable parameters

Architecture: LeNet-5

- LeNet-**5**, since it has 5 layers with learnable parameters

- Architecture details:
  - 3 set of convolutional layers with average pooling

Architecture: LeNet-5

- LeNet-**5**, since it has 5 layers with learnable parameters

- Architecture details:
  - 3 set of convolutional layers with average pooling
  - 2 Fully connected layers

Architecture: LeNet-5

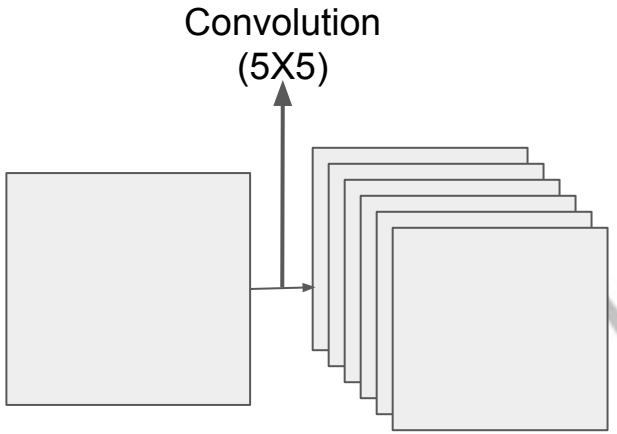
- LeNet-**5**, since it has 5 layers with learnable parameters

- Architecture details:
  - 3 set of convolutional layers with average pooling
  - 2 Fully connected layers
  - Softmax classifier

Architecture: LeNet-5

Input
32 X 32 X 1

Architecture: LeNet-5

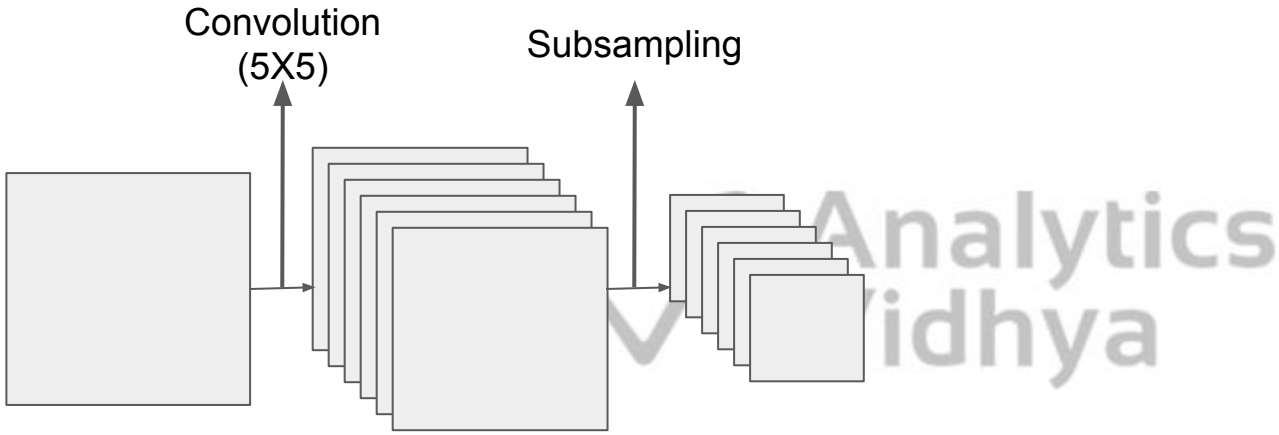


Output shape = ((32-5+1) X (32-5+1) X 6)
= (28 X 28 X 6)

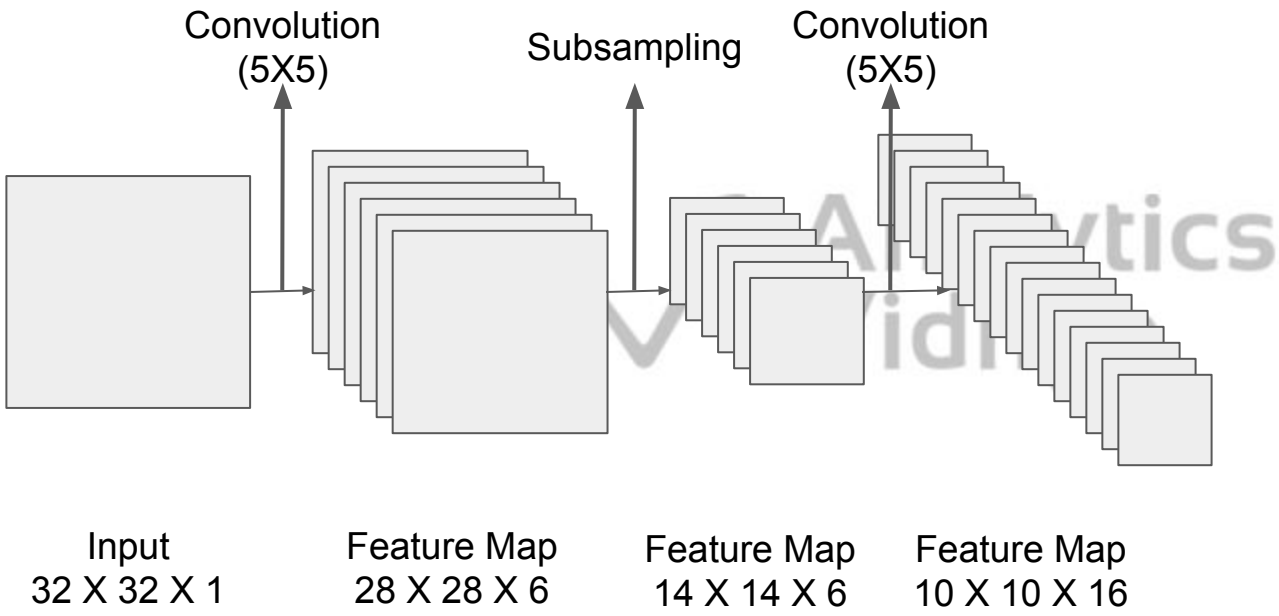
Input
32 X 32 X 1

Feature Map
28 X 28 X 6

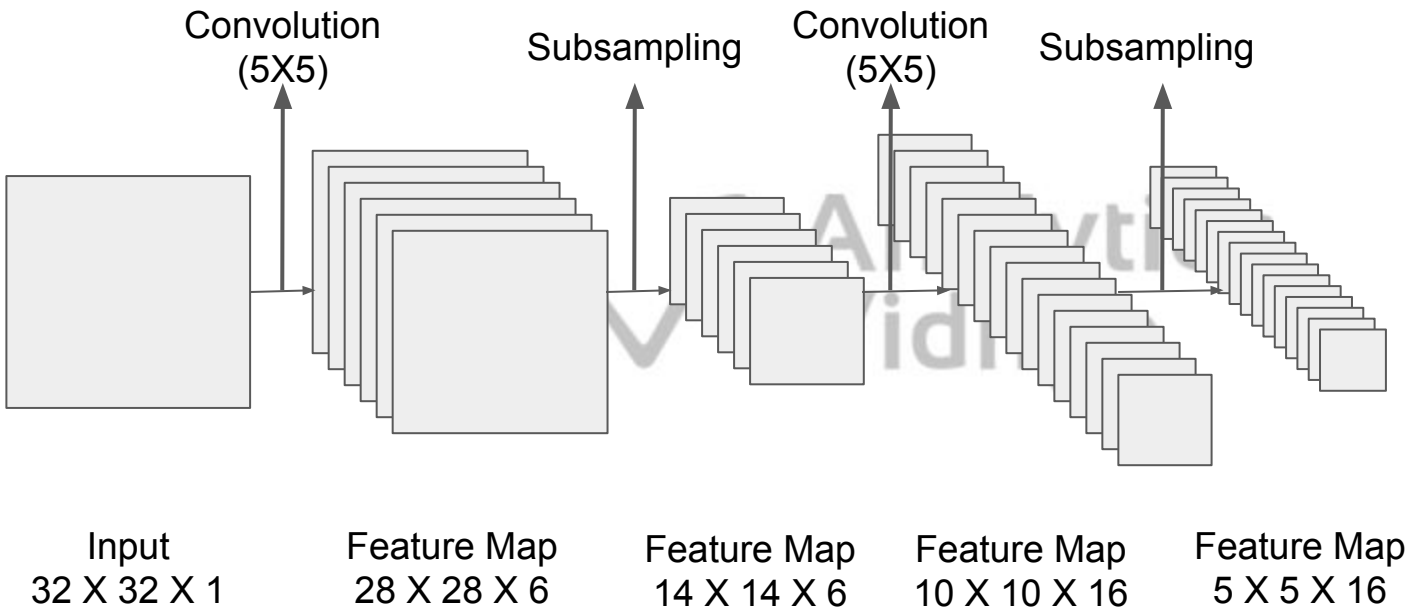
Architecture: LeNet-5



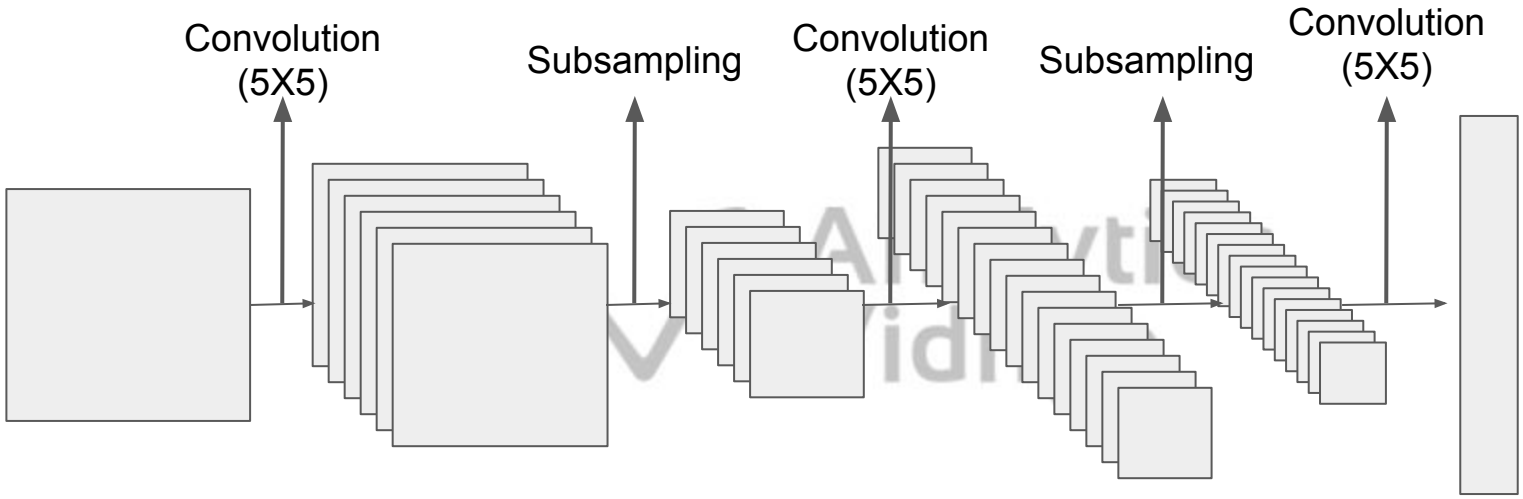
Architecture: LeNet-5



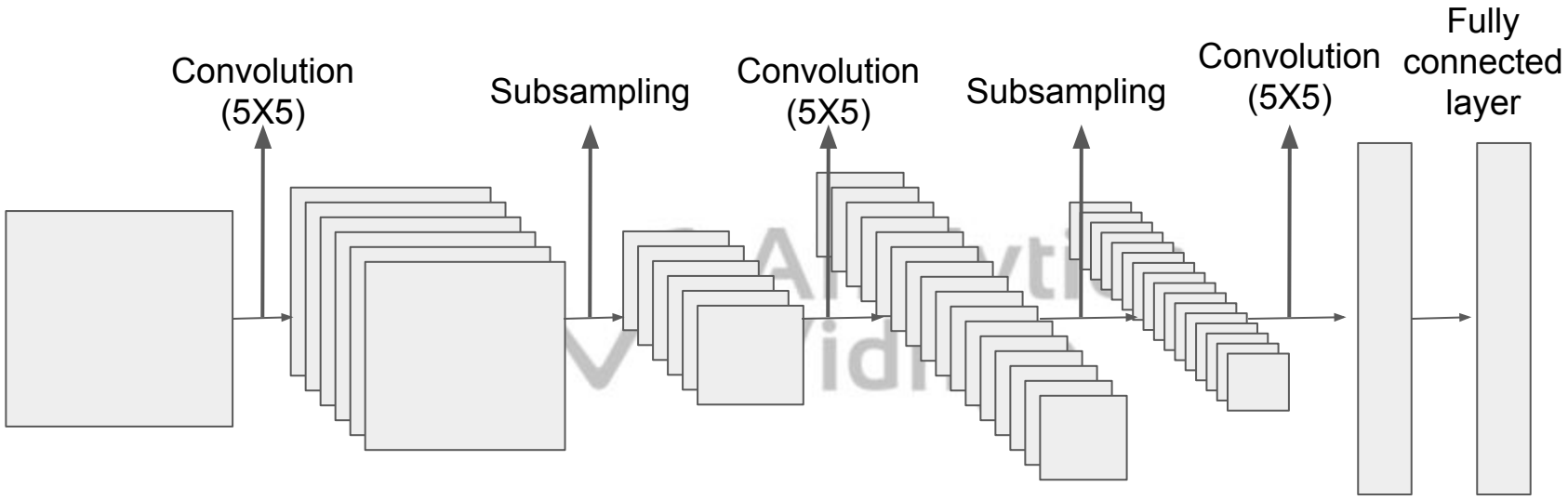
Architecture: LeNet-5



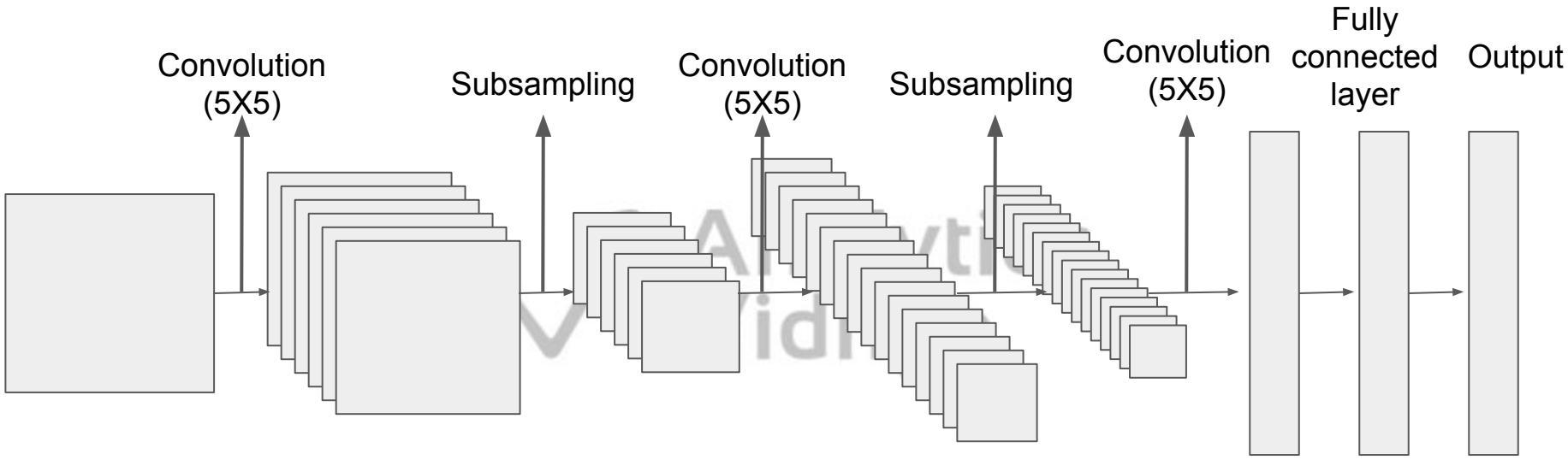
Architecture: LeNet-5



Architecture: LeNet-5



Architecture: LeNet-5



Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	
Conv 3	120	5 * 5	1	120	tanh

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	
Conv 3	120	5 * 5	1	120	tanh
Fully Connected 1	-	-	-	84	tanh

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	
Conv 3	120	5 * 5	1	120	tanh
Fully Connected 1	-	-	-	84	tanh
Fully Connected 2	-	-	-	10	Softmax

Architecture: LeNet-5

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	
Conv 3	120	5 * 5	1	120	tanh
Fully Connected 1	-	-	-	84	tanh
Fully Connected 2	-	-	-	10	Softmax

Number of Parameters = 60k

Summary: LeNet - 5

Analytics
Vidhya

Summary: LeNet - 5

- LeNet-**5**, since it has 5 layers with learnable parameters

Summary: LeNet - 5

- LeNet-**5**, since it has 5 layers with learnable parameters
- Takes grayscale image as input

Analytics Vidhya

Summary: LeNet - 5

- LeNet-**5**, since it has 5 layers with learnable parameters
- Takes grayscale image as input
- Architecture details:
  - 3 set of convolutional layers with average pooling
  - 2 Fully connected layers
  - Softmax classifier

Summary: LeNet - 5

- LeNet-**5**, since it has 5 layers with learnable parameters
- Takes grayscale image as input
- Architecture details:
  - 3 set of convolutional layers with average pooling
  - 2 Fully connected layers
  - Softmax classifier
- Total number of parameters: 60k

Thank You

Analytics
Vidhya