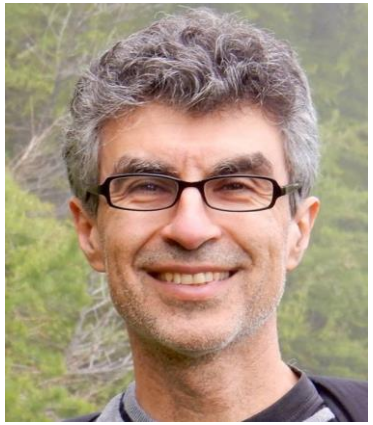
Yann LeCun



Yoshua Bengio

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 shown to outperform all other techniques.

Real-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modeling. A new learning paradigm, called Graph Transformer Networks (GTN), allows such multi-module systems to be trained globally using Gradient-Based methods so as to minimize an overall performance measure.

Two systems for on-line handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of Graph Transformer Networks.

A Graph Transformer Network for reading bank check is also described. It uses Convolutional Neural Network character recognizers combined with global training techniques to provides record accuracy on business and personal checks. It is deployed commercially and reads several million checks per day.

Keywords— **Neural Networks, OCR, Document Recognition, Machine Learning, Gradient-Based Learning, Convolutional Neural Networks, Graph Transformer Networks, Finite State Transducers.**

NOMENCLATURE

- GT Graph transformer.
- GTN Graph transformer network.
- HMM Hidden Markov model.
- HOS Heuristic oversegmentation.
- K-NN K-nearest neighbor.
- NN Neural network.
- OCR Optical character recognition.
- PCA Principal component analysis.
- RBF Radial basis function.
- RS-SVM Reduced-set support vector method.
- SDNN Space displacement neural network.
- SVM Support vector method.
- TDNN Time delay neural network.
- V-SVM Virtual support vector method.

The authors are with the Speech and Image Processing Services Research Laboratory, AT&T Labs-Research, 100 Schulz Drive Red Bank, NJ 07701. E-mail: {yann,leonb,yoshua,haffner}@research.att.com. Yoshua Bengio is also with the Département d'Informatique et de Recherche Opérationelle, Université de Montréal, C.P. 6128 Succ. Centre-Ville, 2920 Chemin de la Tour, Montréal, Québec, Canada H3C 3J7.

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.

The main message of this paper is that better pattern recognition systems can be built by relying more on automatic learning, and less on hand-designed heuristics. This is made possible by recent progress in machine learning and computer technology. Using character recognition as a case study, we show that hand-crafted feature extraction can be advantageously replaced by carefully designed learning machines that operate directly on pixel images. Using document understanding as a case study, we show that the traditional way of building recognition systems by manually integrating individually designed modules can be replaced by a unified and well-principled design paradigm, called *Graph Transformer Networks*, that allows training all the modules to optimize a global performance criterion.

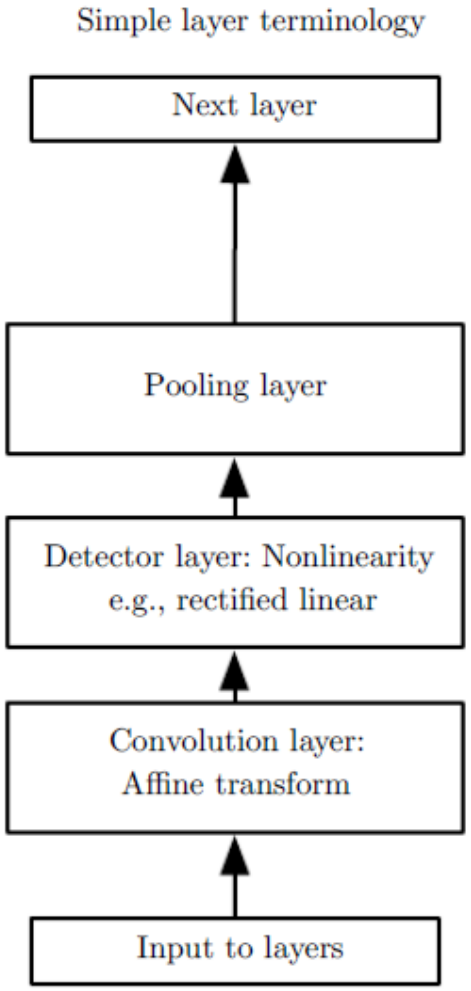
Since the early days of pattern recognition it has been known that the variability and richness of natural data, be it speech, glyphs, or other types of patterns, make it almost impossible to build an accurate recognition system entirely by hand. Consequently, most pattern recognition systems are built using a combination of automatic learning techniques and hand-crafted algorithms. The usual method of recognizing individual patterns consists in dividing the system into two main modules shown in figure 1. The first module, called the feature extractor, transforms the input patterns so that they can be represented by low-dimensional vectors or short strings of symbols that (a) can be easily matched or compared, and (b) are relatively invariant with respect to transformations and distortions of the input patterns that do not change their nature. The feature extractor contains most of the prior knowledge and is rather specific to the task. It is also the focus of most of the design effort, because it is often entirely hand-crafted. The classifier, on the other hand, is often general-purpose and trainable. One of the main problems with this approach is that the recognition accuracy is largely determined by the ability of the designer to come up with an appropriate set of features. This turns out to be a daunting task which, unfortunately, must be redone for each new problem. A large amount of the pattern recognition literature is devoted to describing and comparing the relative

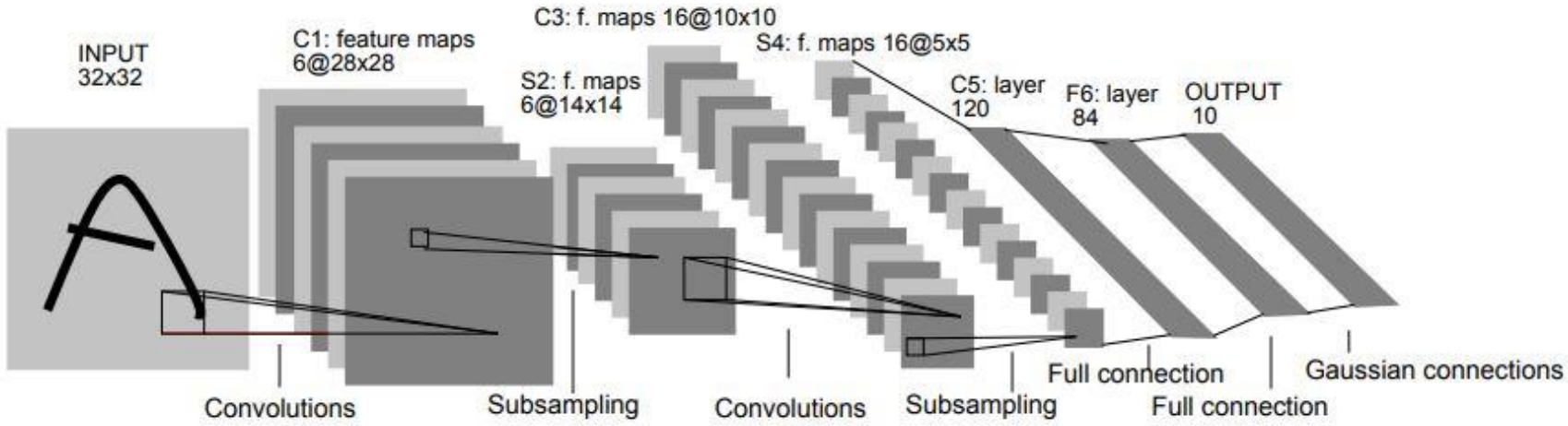
- The main message of this paper is that better pattern recognition systems can be built by relying more on **automatic learning**, and less on **hand-designed heuristics**.

$$E_{test} - E_{train} = k(h/P)^{\alpha}$$

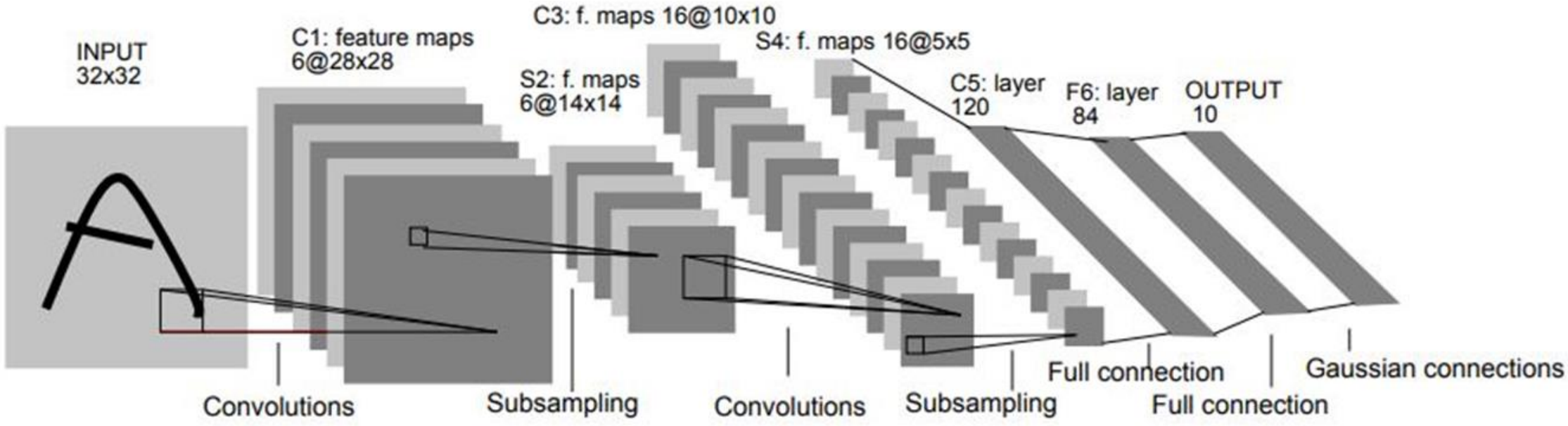
- P is the number of training samples, h is the measure of effective capacity (complexity of the machine), α is a number between 0.5 and 1, and k is a constant.
- Use backpropagation for training the proposed model.
- Experiments were conducted on MNIST dataset.

LeNet 5 - 1998

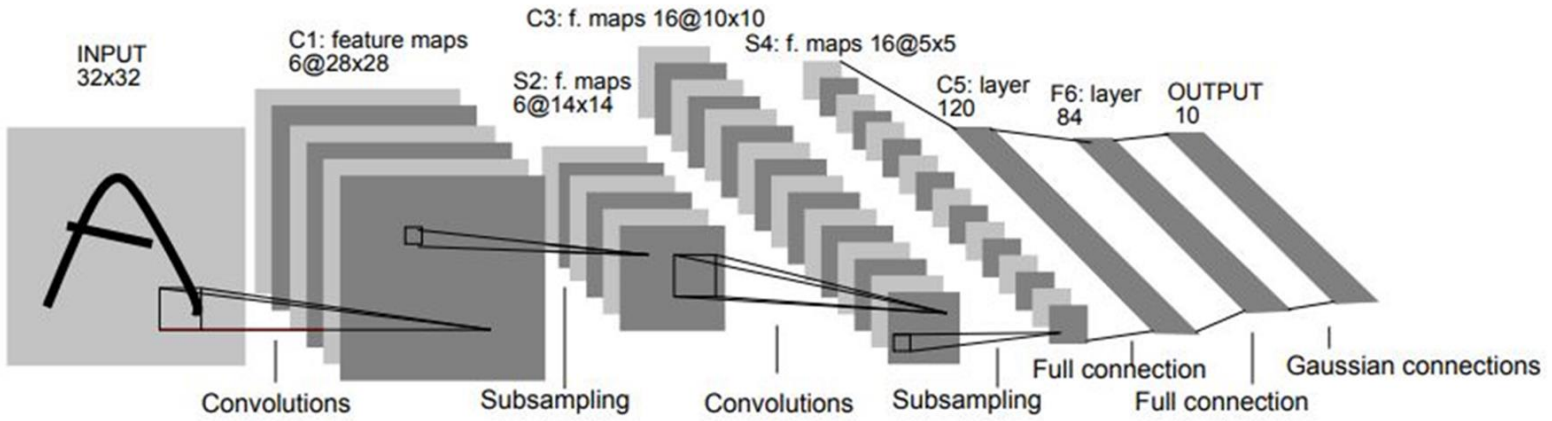




- Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.
- Convolution leverages three important ideas that can improve a machine learning system : **sparse interactions**, **parameter sharing**, **equivariant representations**.
- The spatial relationship between the pixels is not considered in ANNS.
- Convolution provides a means for working with inputs of variable size.

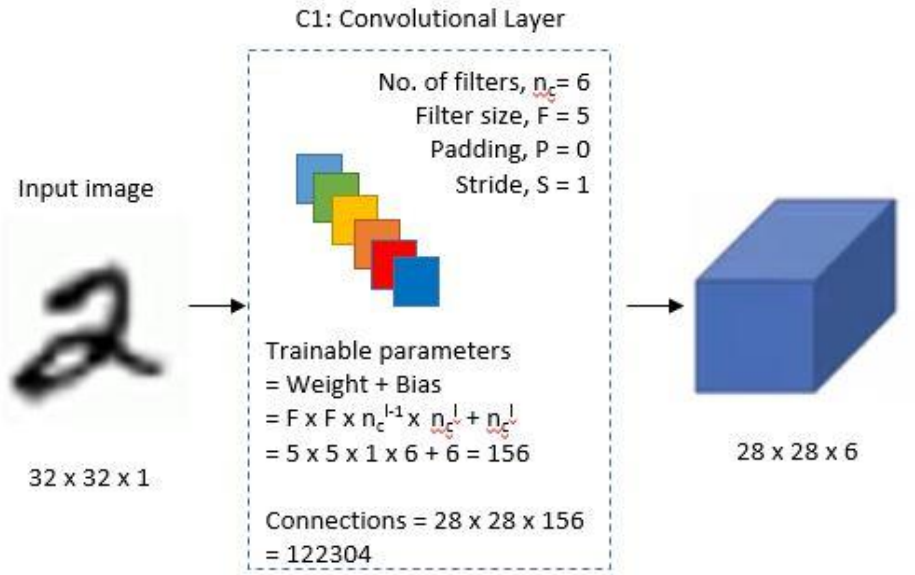


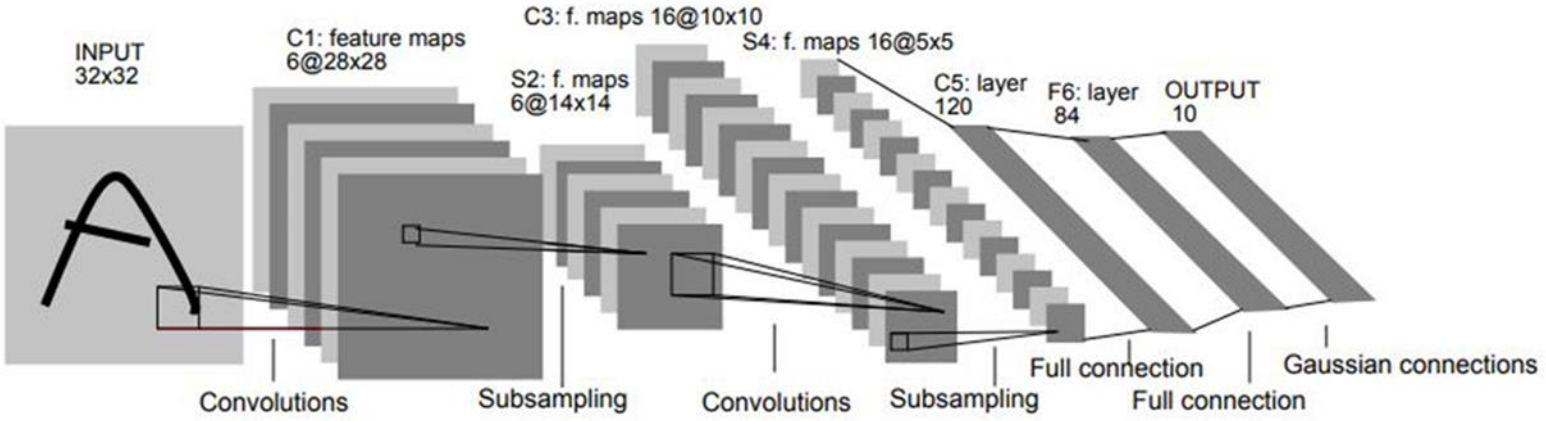
- LeNet-5 comprises 7 layers (excluding input layer), all of which contains trainable parameters.
- The input is a 32 x 32 pixel image. The values of the input pixels are normalized so that the background level (white) corresponds to a value of -0.1 and the foreground (black) corresponds to 1.175.
- Convolutional layers are labeled Cx, sub sampling layers are labeled Sx and fully connected layers are labeled Fx, where x is the layer index.



Layer C1

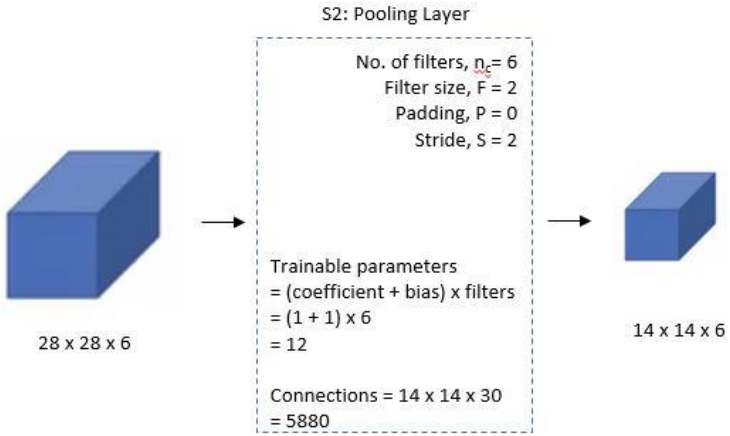
- Convolutional layer with 6 feature maps (6 filters)
- Convolutional filter size : 5 x 5
- Feature map size (image size after convolution) : 28 x 28
- Total trainable parameters : 156 (5 x 5 x 6 = 150w + 6b = 156)

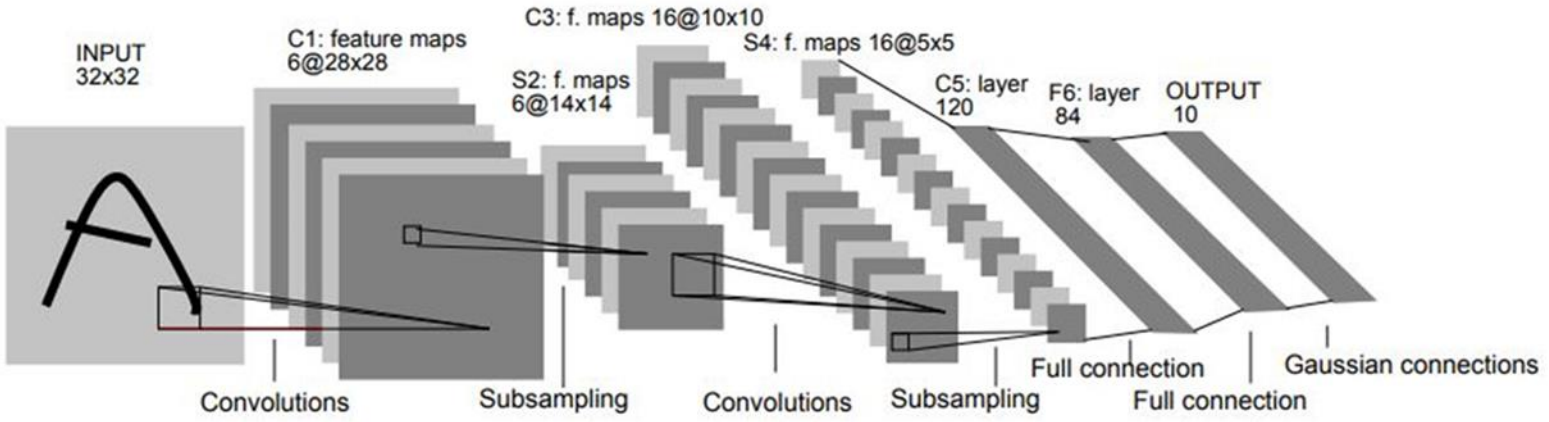




Layer S2

- Subsampling layer (pooling layer)
- Each unit in each feature map is connected to a 2 x 2 neighborhood in the corresponding feature map in C1.
- The four inputs to a unit in S2 are added, then multiplied by a trainable coefficient, and added to a trainable bias.
- The result is passed through a *tanh* function.
- The 2 x 2 receptive fields are non-overlapping, hence the output will be half the size of the input.
- Layer S2 has 12 trainable parameters (6w+6b)





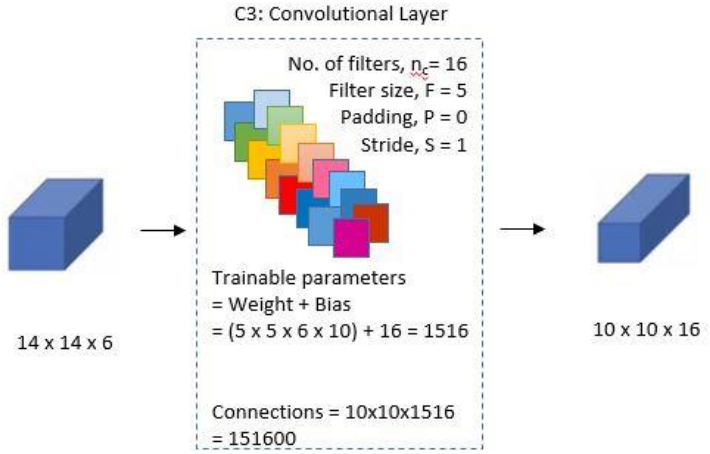
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

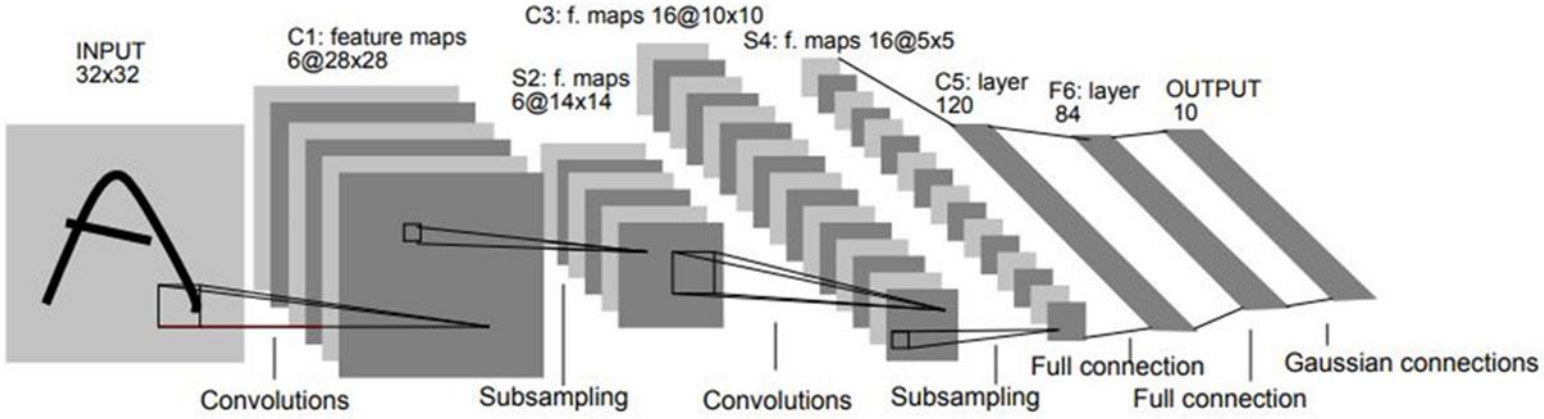
TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

Layer C3

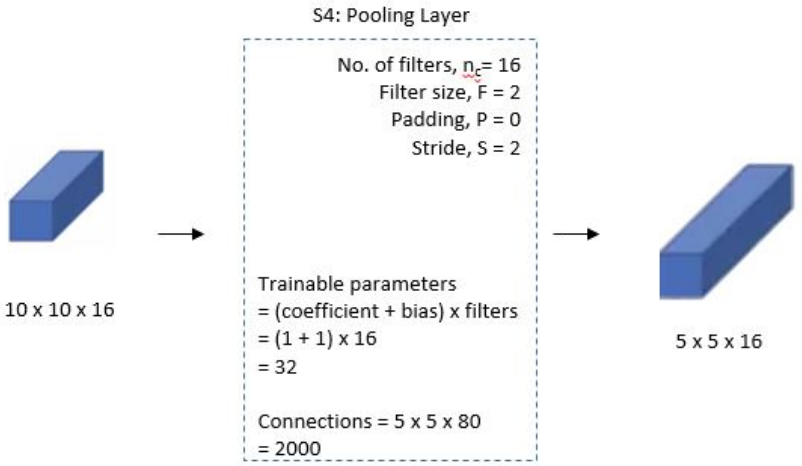
- Convolutional layer with 16 feature maps (16 filters)
- Convolutional filter size : 5 x 5 x Z
- The first six C3 feature maps take inputs from every contiguous subsets of three feature maps in S2 (filter size : 5 x 5 x 3)
- The next six take input from every contiguous subset of four (filter size : 5 x 5 x 4)
- The next three take input from some discontinuous subsets of four (filter size : 5 x 5 x 4)
- The last one takes input from all S2 feature maps (filter size : 5 x 5 x 6)
- Total trainable parameters : **1516** (450w + 600w + 300w + 150w + 16b)

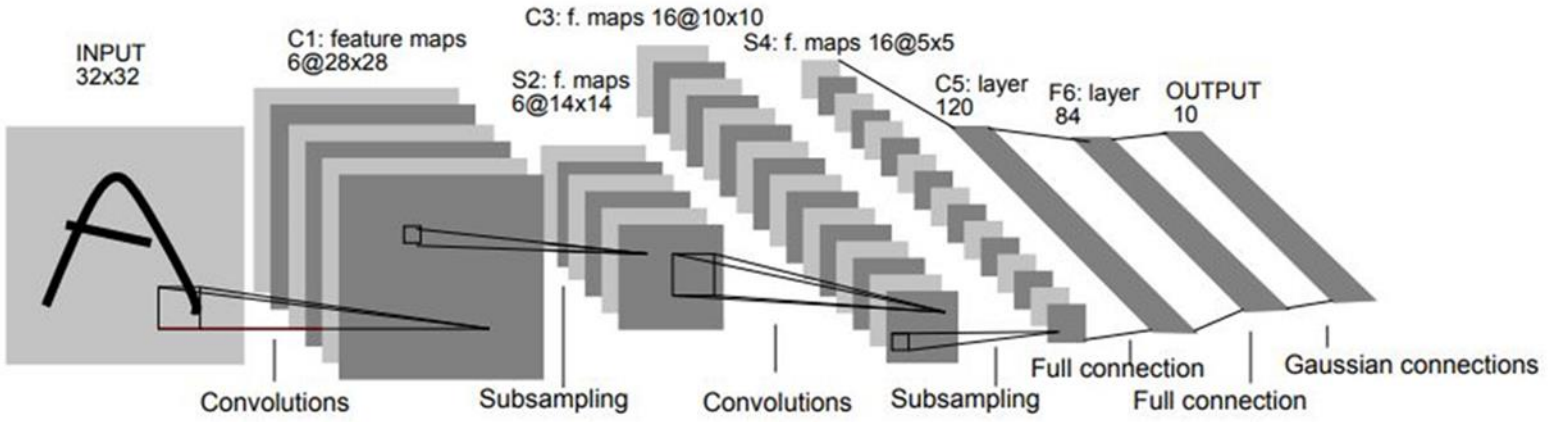


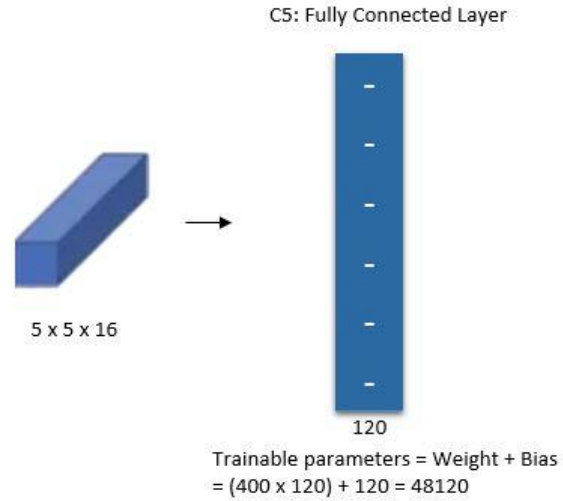


Layer S4

- Subsampling layer (pooling layer) with 16 feature maps of size 5 x 5
- Each unit in each feature map is connected to a 2 x 2 neighborhood in the corresponding feature map in C3, in a similar way as C1 and S2.
- Layer S4 has 32 trainable parameters (16w+16b)





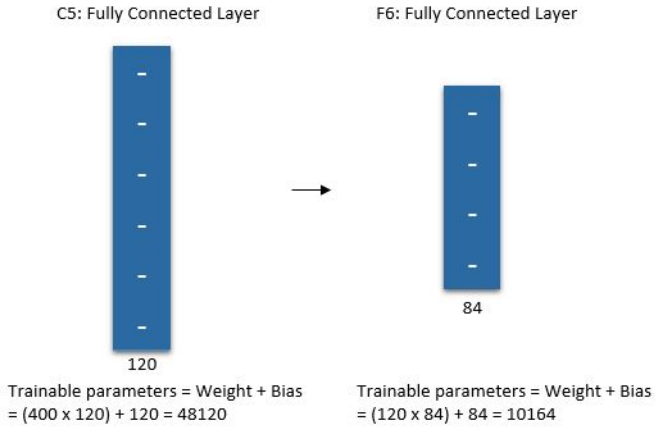


Layer C5

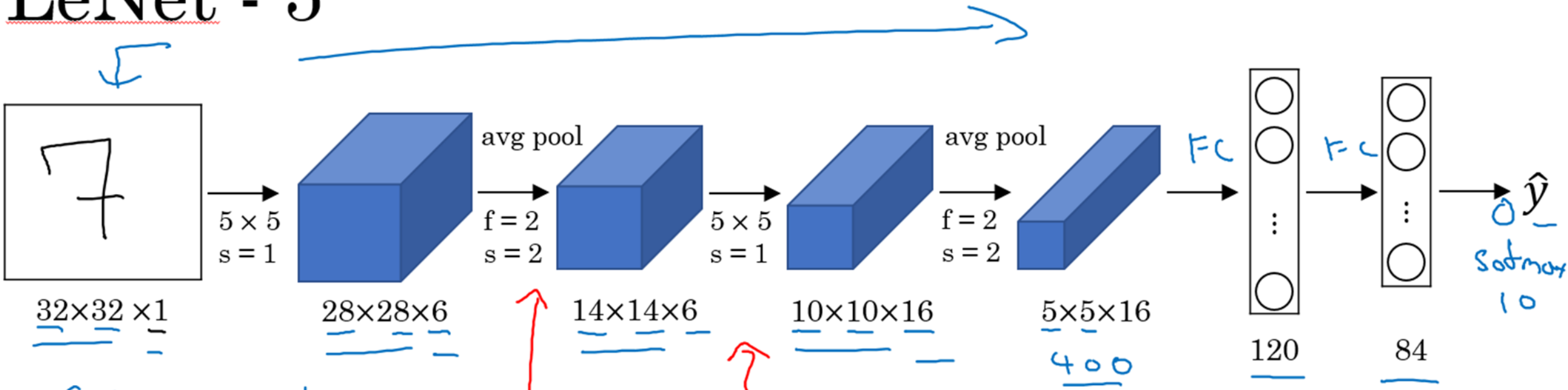
- Convolutional layer with 120 feature maps (120 filters)
- Convolutional filter size : 5 x 5 x 16
- Total trainable parameters : **48120** (48000w+120b)

Layer F6

- Contains 84 neurons and is fully connected to C5.
- Layer F6 has **10164** trainable parameters (10080 w + 84 b)



LeNet - 5



60k parameters.

$n_H, n_w \downarrow$ $n_c \uparrow$

Non-linearity after pooling

$n_H \times n_w \times n_c$ $f \times f \times n_c$

conv pool conv pool fc fc output

Advanced: sigmoid/tanh ReLU

II III

[LeCun et al., 1998. Gradient-based learning applied to document recognition]