UNIVERSITY NAME

DOCTORAL THESIS

Thesis Title

Author:
John SMITH

Supervisor:
Dr. James SMITH

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

September 20, 2021

Declaration of Authorship

I, John SMITH, declare that this thesis titled, "Thesis Title" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY NAME

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

Thesis Title

by John SMITH

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

The acknowledgments and the people to thank go here, don’t forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

Physical Constants

Speed of Light	$c_0 = 2.997\,924\,58 \times 10^8\,\mathrm{m\,s^{-1}}$ (exact)

List of Symbols

a	distance	m
P	power	W (J s^{-1})
ω	angular frequency	rad

For/Dedicated to/To my…

Chapter 1

Machine Learning

Machine Learning is a subfield of Artificial Intelligence (AI) and is used in a wide range of applications, such as computer vision, speech recognition, drug discovery or medical image analysis. It is a study of computer algorithms that construct statistical models trained to perform some specific task. The models improve their performance automatically by learning from examples instead of relying on static program instructions. Importantly, learning in this context does not mean to memorize examples but to extract patterns or rules from the training data such that the model can make reasonable predictions for data points absent from the training examples too.

Broadly speaking, ML algorithms can be divided into two categories: Supervised and Unsupervised learning. The supervised learning approach uses labeled data, where each input is linked to a desired output. Typical examples of supervised learning are regression or classification tasks. The unsupervised approach deals with unlabeled data, where only the inputs are given, and aims to find structure in the data, like clustering the data points.

While ML is an umbrella term for many different algorithms, such as Kernel methods, decision trees or Artificial Neural Networks (ANNs), we will focus on the latter in this work.

1.1 Artificial Neural Networks

ANNs are inspired by nervous systems, such as our human brain. Their ability to organize themself and learn from experience has fascinated researchers over the past century and has motivated them to develop artificial models of its biological counterpart.

The history of ANNs began in 1943 with the creation of the first computational model of a neural network by Warren McCulloch and Walter Pitts based on algorithms called threshold logits. The next milestone was taken by F. Rosenblatt in 1958 with the invention of the perceptron: A two-layer neural network able to perform pattern recognition. The publication of the book 'perceptron' by Marvin Minsky and Seymour Papert in 1969 lead to the AI winter and research in the field stagnated as they stated that basic two-layer networks are incapable of solving the exclusive or circuit (non-linearly separable data). Larger networks were needed to solve the problem but could not be realized because of a lack of computational processing power and efficient algorithms at that time. Fortunately, the research field was not completely abandonned and ANNS were revived in the early 1980s due to the increased computational processing power and the introducing of the backpropagation algorithm in 1975 by Werbos that made it possible to train multi-layer networks. Nowadays, ANNs have won several state of the art ML contests and are used in many applications of our everyday life.

1.1.1 A Neuron

Nervous systems are built up from a large number of interconnected cells, called neurons. The human brain, as an example, consists of $\sim 10^{11}$ neurons. The main task of a neuron is to receive, process and transmit signals. To achieve this, a biological neuron is equiped with dendrites (receiver), a cell body (processor) and an axon (transmitter). Dendrites are thin fibers connected via synapses with the axons of thousands of other neurons. Signals captured by the dendrites are weighted by the synapses and can either increase or decrease the electrical cell potential. If a specific threshold potential is reached the axon fires a signal.

An artificial neuron works similarly to this: An input tensor x is weighted by a weight tensor w and the result is accumulated. Afterwards, a threshold value ψ is subtracted and a non-linear function, called activation function, $a()$ is applied to derive the output y.

$$y = a(\sum_i x_i w_i - \psi) \tag{1.1}$$

In this example x and w are vectors but higher rank tensors are common as well.

1.1.2 Deep Neural Network

A single neuron is a very simple processing unit, but a network of neurons becomes a powerful information processing system: Signals from the environment captured by sensory cells are encoded and processed by the network to create an appropriate response.

To explain the mechanism of information processing we can turn to visual object recognition: Our retina encodes visual stimuli into electrical signals that are transmitted to the visual cortex. Here, the incoming signal will cause a subset of neurons to respond, which can be described as a response vector. The response vector for a given object is not constant but varies under identity preserving transformations, such as shifts in position, rotations or changing illumination. Therefor, a given object has to be linked to a set of response vectors that span a manifold in the high dimensional space of all possible response vectors. At early stages of processing, the object identity manifolds for different objects might be highly tangled and introducing an accurate decision boundary for object recognition becomes impossible. This is where the special structure of the visual cortex plays an important role: Neurons are grouped into subsequent layers and the further the signals are processed the more flattened and seperated the manifolds become.

The same idea is applied in modern ANNs that are refered to as Deep Neural Networks (DNN) or Deep Learning (DL): Multiple layers are arranged subsequently and each layer learns to transform its input into a more abstract and composite representation. While the first layer might learn very basic features, like the positions of edges, subsequent layer can learn more high-level features composed of the preceding features.

Formally, the layer between the input layer and the output layer are referred to as hidden layer. In The simplest case information only flows forward in the network (feedforawrd neural network), from the input layer through the hidden layers to the output layer. In more complex network architectures the connections between the layer can form cycles (recurrent neural network), such that a layer can receive information from subsequent layer as well.

The mathematical foundation for DNNs is given by the universal function approximation theorem that was first proven by George Cybenko in 1989. In its classical formulation it states that any continuous function f on a compact set K can be approximated by a feedforward neural network F with just one hidden layer within arbitrary accuracy $|F(x) - f(x)| < \epsilon$, where $\epsilon > 0$ and $x \in K$. Importantly, the width of the hidden layer (number of neurons) needs to be unbound in this formulation of the theorem. A dual formulation states that the theorem holds true for bounded width but arbitrary depth (number of layer) as well. While both cases guarantee the existence of an appropriately tuned network capable to approximate any continuous function, practice has shown that DNNs perform better in many applications and suffer less from training issues, such as overfitting.

Modern ANNs are built up from various different layer architectures. In following, we want o give an overview over some of the most commonly used.

Dense Layer

A dense layer (also fully-connected layer) is the simplest and most common layer of an ANN. Each neuron in a dense layer receives signals from all the neurons of the preceding layer. We can express a dense layer as

$$\vec{y} = a(A\vec{x}) \tag{1.2}$$

where A is a matrix containing the weights and thresholds. The full-connectivity of each neuron makes it capable to detect global pattern in the data, but it also makes it impractical for large inputs as A grows with the size of $\vec{x}$

Convolutional Layer

Another idea originating from our understading of the visual cortex is local connectivity: Neurons are only locally connected to neurons in a restricted area of the previous layer known as the receptive field of the neuron. This architecture is perfectly suited to learn hierchacical pattern in the data as the receptive field of the neurons become bigger the higher they are placed in the hierachy of the network.

A convolutional layer consists of a bank of parameterized filters that are slized over the input. At each position the descrete convolution of the filter with the segment of the image it overlaps is computed and stored into a so called feature map. Convolutional layer can be applied to tensors of arbitrary rank. In the following we consider the two dimensional case where the input are images $X_j \in \mathbb{R}^{N_x \times N_y}$ with N_x the width, N_y the height of the image and $j \in \{0, .., N_c\}$ is the index for the N_c feature channels. The different feature channels provide different views on the data, such as the different colour channels for a RGB image. The bank of filters is denoted with $K_{i,j} \in \mathbb{R}^{m_x \times m_y}$, where m_x is the width, m_y is the height, connecting the jth feature channel of the input with the ith feature channel of the output. The ith feature map $Y_i \in \mathbb{R}^{n_x \times n_y}$ of the output can be computed as

$$Y_i = a(\Phi + \sum_{j}^{N_c} K_{i,j} * X_j) \tag{1.3}$$

where Φ is a bias matrix. The size of the output (n_x, n_y) can be derived as:

$$(n_x, n_y) = (N_x - m_x + 1, N_y - m_y + 1) \tag{1.4}$$

Furthermore, the size of the output can also dependent on zero-padding and slicing:

- zero-padding: the size of the input tensor can be artificially extended by adding zeros at the border or between input units. P denots the number of zeros concatenated at each side.
- strides: While the filter is sliced over the input the step size S, called stride, for the translation can be greater than one effectively reducing the output size.

In summary, the output size can be computed as

$$(n_x, n_y) = (\frac{N_x - m_x + 2P}{S} + 1, \frac{N_y - m_y + 2P}{S} + 1) \tag{1.5}$$

Note that depending on the choice of zero-padding and strides the size of the output can either decrease (downsample) or increase (upsample) compared to the size of the input. The latter one is often referred to as transposed (or fractionally-strided) convolution and is typically used in a decoder architecture as it introduces a concept to learn the upsampling transformation. While this concept correctly describes the concept of upsampling, it incolves many useless multiplcations with zero which are typically avoided by efficient implementations in real-world applications by efficient implementions.

Typically, a convolutional layer needs less parameters than a dense layer as the size of the filters are independent from the size of the input. Additionally, a convolutional layer is equivariant with respect to translations as the same filters are applied over the hole input.

1.2 Generative model

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:
`\hypersetup{urlcolor=red}`, or
`\hypersetup{citecolor=green}`, or
`\hypersetup{allcolor=blue}`.
If you want to completely hide the links, you can use:
`\hypersetup{allcolors=.}`, or even better:
`\hypersetup{hidelinks}`.
If you want to have obvious links in the PDF but not the printed text, use:
`\hypersetup{colorlinks=false}`.