Deep Learning

for dummies®
A Wiley Brand

Join the deep learning revolution

Validate and measure performance

Explore real-world applications

Eli David, PhD

Deep Instinct Special Edition

Deep Learning

Deep Learning

Deep Instinct Special Edition

by Eli David, PhD

for dummies®
A Wiley Brand

Deep Learning For Dummies®, Deep Instinct Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119-48358-8 (pbk); ISBN 978-1-119-48355-7 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

We're proud of this book and of the people who worked on it. Some of the people who helped bring this book to market include the following:

Project Editor: Martin V. Minner
Editorial Manager: Rev Mengle
Executive Editor: Katie Mohr
Business Development Representative: Karen Hattan
Production Editor: Tamilmani Varadharaj

Table of Contents

Introduction

Can you tell the difference between a dog and a dogwood tree? Yes, you've been making that distinction since before you can even remember. How about a dog from a cat? Still simple for you. Are you looking at a West Highland White Terrier or a Bichon Frise? That's a bit more advanced, but plenty of dog lovers know the difference. But, is it possible to teach a computer to tell the difference? Can you train a computer to look at a picture of a Bichon running across a hardwood floor with a red chew toy in its mouth — and then ask that computer to write an accurate photo caption? In French?

Believe it or not, yes. It takes some serious artificial intelligence (AI), but yes, a computer can do that. It's all possible through an exciting subset of AI known as *deep neural networks,* or as many of us like to call it, *deep learning.* Deep learning is a great name for this dazzling variety of AI, because it implies the computer is doing pretty much what our brains do. The computer is learning, not just memorizing, not just following rote instructions, but looking at something it doesn't know and truly learning all about it. On its own.

Sundar Pichai, CEO of Google, recently referred to the deep learning revolution as "one of the most important things humanity is working on. It is more profound than electricity or fire."

Not a lot of people are fully immersed yet in making deep learning happen, but most of us will live to see the day when deep learning fuels all of the most amazing products and services that shape our lives. Will you help blaze this path?

About This Book

Deep Learning For Dummies, Deep Instinct Special Edition, is your handbook for exploring this remarkable new world of AI. This book provides background on what deep learning is, how it developed from earlier methods of AI and machine learning, and why it's so much more powerful.

To be sure, deep learning isn't easy. It isn't for everything or for everyone, at least not yet. It requires a very certain kind of

expertise, the right equipment, and a large collection of the right kind of data. But this introduction will help you determine whether you face a situation that will benefit from deep learning, and if so, how to proceed. You'll also have the insights you need to decipher the AI solutions that others are inviting you to try, and figure out just how revolutionary they really are.

Foolish Assumptions

I've made some basic assumptions about you, the reader, as I've gathered information to include in this book:

- You're a high-level business executive, perhaps a chief information officer, chief technology officer, or chief information security officer.
- You have a wealth of technology background and experience, but need insights about deep neural networks.
- You've had offers to apply AI to your organization's needs, and you want to discern whether or not those solutions really involve deep learning.

Icons Used in This Book

The point of this book is to put lots of info at your fingertips and make it easy to navigate. Some icons will help.

REMEMBER

If you have limited time and can't read every word right now, at least pay close attention to the paragraph next to this icon.

TIP

I want to help you achieve what you want, and this paragraph offers some useful advice.

TECHNICAL STUFF

AI is a seriously technical topic. I've spared you a lot of the detail, but this paragraph goes down the techie path.

WARNING

Deep learning is amazing, and also tricky. Check this insight to help steer clear of trouble.

Where to Go from Here

This part is not computer science — just turn the page! That said, you don't have to turn to the *next* page. Drop in on whatever chapter interests you most, because this book is organized in a way that lets you pick and choose just the information you need. Wherever you turn, enjoy your reading!

IN THIS CHAPTER

- Understanding artificial intelligence
- Moving ahead with machine learning
- Diving into deep learning
- Gauging the success of deep learning

Chapter 1
The Deep Learning Revolution

During the past few years, deep learning has revolutionized nearly every field it has been applied to, resulting in the greatest leap in performance in the history of computer science. The application of deep learning has made those small, gradual annual improvements a thing of the past — these days, it isn't uncommon to witness improvements of 20 to 30 percent, in months and not years.

There's no keeping that kind of success under wraps, which means the media have been filled with references to "artificial intelligence," "machine learning," and "deep learning." These terms are used not only very widely, but most of the time inaccurately and confusingly. With that in mind, this chapter aims to clarify and demystify the distinctions among these technical terms.

The Story Begins with Artificial Intelligence


REMEMBER

The term *artificial intelligence* (AI) was coined by the pioneering computer scientist John McCarthy in the 1950s. It's an umbrella term covering all the methods and disciplines that result in any form of intelligence exhibited by machines.

This includes the 1980s expert systems (which were basically datasets of hard-coded knowledge) all the way up to most advanced forms of AI now in use. Nearly all software that's currently being used in just about all industries employs at least some form of AI, even if it's limited to some basic manually coded procedures.

What Is Machine Learning?

The leading subfield of AI today is known as *machine learning.* Through machine learning, computers can learn without being explicitly programmed. AI methods based on machine learning have dominated AI in the 2000s, and they have outperformed AI that is not based on machine learning.


REMEMBER

Machine learning is successful, to be sure, but that doesn't mean it is without limitations. One of its major limitations is its reliance on *feature extraction.* In this process, human experts dictate what the important features or properties of each problem are.

Consider the challenge of face recognition. You can't just feed the raw pixels of an image into a machine learning module. Instead, those pixels must first be converted into specific features that the machine learning module will be on the lookout for, such as distance between pupils, proportions of the face, texture, and color.

Certainly, face recognition through this approach is pretty impressive. But the fact is, by focusing on those very specific aspects defined by human experts through feature extraction, the approach is ignoring most of the raw data. As useful as the selected features may be, this method for face recognition misses the rich, complex patterns in the data.

For more detailed information on traditional machine learning, check Chapter 2.

Advancing into Deep Learning

Deep learning, also known as *deep neural networks*, is a subfield of machine learning, which is a subset of AI, as shown in Figure 1-1. Deep learning takes inspiration from how the human brain works.

What's the difference between deep learning and traditional machine learning? Perhaps the biggest distinction is that deep learning is the first — and currently the only — learning method that is capable of training directly on the raw data.

No need for feature extraction with deep learning. In the example of facial recognition, deep learning would be able to dive in and examine the raw pixels of an image, without explicitly being told to pay attention to facial proportions or distance between pupils or other specifics called out by human experts.

What's more, deep learning scales well to hundreds of millions of training samples. As the training dataset gets larger and larger, deep learning continuously improves.

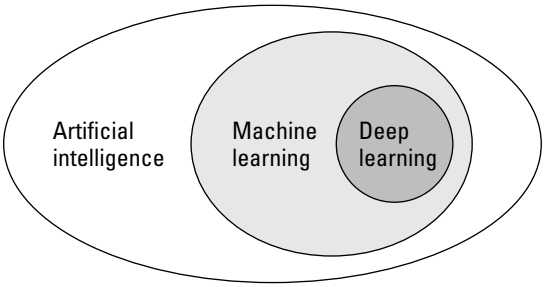


FIGURE 1-1: Deep learning, a subset of a subset of AI.

Deep Learning's Mind-Boggling Success

During the past few years, deep learning has achieved 20 to 30 percent improvement in most benchmarks of computer vision, speech recognition, and text understanding. It's the greatest leap in performance in the history of AI and computer science. No wonder lots of people are talking about deep learning.

Two major drivers have contributed to the sudden mind-boggling success of deep learning. The first was the improvement in algorithms.

Until a few years ago, it was only possible to train shallow neural networks. Because of the limits of algorithms, it was not possible for deeper and more complex networks to converge. Today's improved methods allow for the successful training of very deep neural networks. Current achievements have reached many tens of layers and billions of synapses, which are the connectors between the neurons.

The second, even more important, factor is the use of *graphics processing units* (GPUs). These days, nearly all deep learning training is conducted on GPUs made by the technology company Nvidia.

Technology developed by Nvidia has yielded speeds up to 100 times greater than other kinds of hardware, such as *central processing units* (CPUs). Consider the comparison: The training of a deep-learning brain that might have taken three months on CPUs could happen in a single day using GPUs.

Despite the success of deep learning in many tasks, the barrier to entry into deep learning remains high, mainly because of the scarcity of deep-learning researchers and scientists, who are critical for its successful application.

IN THIS CHAPTER

- Defining the training data
- Extracting features
- Understanding the drawbacks of feature extraction

Chapter 2
Traditional Machine Learning

Like a child moving up through the various levels of school, artificial intelligence is constantly growing more sophisticated. Earlier incarnations of AI relied on routines that programmers specified manually, along with *heuristics*, which are essentially shortcuts for facilitating fast but accurate decision-making.

This chapter looks at the higher level of AI known as *machine learning*, through which the computer learns by itself without being explicitly programmed. The next few pages explore how machine learning happens through the use of data known as *training samples*, and how machine learning requires both training data and a mechanism for feature extraction.

Assembling the Training Data

REMEMBER

In order to train a machine learning model, you first need data samples. These are essential ingredients without which machine learning can't happen.

Imagine trying to train a "dog detector." The aim is to create a machine learning model that can look at an image and determine

whether or not the image contains a dog. Your training data, which must be prepared in advance, will be a large set of images that fall into two basic categories: images that contain dogs and images that don't.

For each image in this dataset, you'll add a *label* indicating whether the image can be classified "dog" or "not dog." Note that this simple example has those two classes, but it's possible to do this kind of thing with a whole lot more classes, even thousands of different labels representing different object classes.

REMEMBER

The labels in this dataset will guide the training. Think of the labels as supervising the training, because this kind of training that uses a fully labeled dataset is known as *supervised training.* So is there such a thing as unsupervised training? Yes, it is possible to train a machine learning module using training data that doesn't have any associated labels. Most real-world scenarios, though, use supervised training, because if labeled data is available, it usually gets better results. Unsupervised learning has huge untapped potential as well because most of the data in the world is unlabeled.

Understanding the Importance of Feature Extraction

So, you've got your set of labeled images, some with dogs and some without. You were able to create those labels because your brain can look at the images and recognize dogs. As you look at the pictures, you probably aren't even noticing that each image is made up of thousands and thousands of raw pixels, because your brain has learned to pull those pixels into a recognizable image.

REMEMBER

In traditional machine learning, however, you can't simply feed this raw data into the machine learning module. The machine, it turns out, is aware of those pixels, because that's all it sees initially: a whole bunch of pixels. For those pixels to be useful in training the model, you first need to conduct what's known as a *feature extraction* phase (or *feature engineering*).

This process extracts a set of predefined properties or features from the raw data. In the dog detector example, each input sample is represented as a vector (list) of values. Each list corresponds to a single feature.

For example, important dog features could be length of the animal, height, length of its snout, length of its ears, color, texture, and that kind of thing. Sounds simple enough, but it isn't. Who's to say what the most important features are for recognizing dogs?

REMEMBER

To perform feature extraction, you first need a *domain expert* to specify what the important features are. As the name suggests, the domain expert is somebody who has expertise in the specific problem domain. For problems involving images, you need an image processing expert who can analyze the problem domain and the samples, and then determine the features to extract.

In a typical case, you start with raw data and extract a few tens of features, perhaps hundreds, or maybe even thousands. Each input sample is converted to a vector of numbers, each corresponding to a feature. This vector is then fed into the machine learning module. Check Figure 2-1 to see how a cute dog is transformed by feature extraction into a vector of numbers.

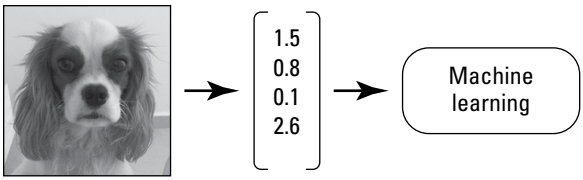


FIGURE 2-1: Feature extraction for the dog detector.

Other real-world examples work in a similar fashion, even if they don't involve cute dog pictures. Whenever using traditional machine learning, you must first specify features. If you want to train a model for detecting malicious computer files rather than dogs, your feature extraction would be based on properties of files. These might include important API or function calls, registry keys used, and that type of feature.

Recognizing the Drawbacks of Feature Extraction

Feature extraction is a mandatory phase in traditional machine learning, but it has its drawbacks. The use of feature extraction introduces two severe limitations into the equation.

First, when you convert the richness of raw data into a small vector of features (as good as they may be), you're inevitably throwing away most of the data. Consider that an image that is 512 x 512 pixels contains hundreds of thousands of pixels, which are the raw data values. Convert this image into a list of tens or hundreds of features, and you've grasped only a small portion of the information contained in the raw data.

Second, and even more important, even the best human experts in this field can only grasp the linear, simple properties when they're selecting features. Most of the patterns and correlations in raw data are too complex for any human to specify, no matter how talented.

More considerations of cats and dogs will help illustrate this point. You are quite capable of detecting cats and dogs in images, as is just about every human out there. If you're handed an image containing a cat or a dog, within a few milliseconds you'll correctly detect if there is a cat or a dog in the image, with near 100 percent accuracy. So far, so good.

Now, try to articulate the differences between a cat and a dog. How easy is it to put into words which features you use for detecting cats and dogs? The more you think about this, the more you realize it's an extremely difficult task. Sure, dogs usually are bigger, typically have longer snouts, and that kind of difference. And yet, there are many dogs that are smaller than cats and have shorter snouts, but you still can easily recognize them as dogs. Sure, there are some pretty distinct features that separate dogs and cats, such as the shape of their pupils. But in reality, you would be able to tell a cat from a dog even if its eyes were shut!

REMEMBER

The point is, even for the easiest task for which darn near everyone is an expert, it's pretty tough to do feature engineering. Venture down the path into even more complex tasks such as computer vision, speech recognition, language understanding, or cybersecurity, and feature engineering becomes all the more difficult to perform. Even after years of meticulous work determining engineered features, it's still only possible to extract the superficial linear features. You're still missing out on most of the complex patterns.

As limited as feature extraction has been, until recently it has been the best possible approach because no alternative was available. That is, until deep learning came along. As outlined in subsequent chapters, deep learning is the first — and currently only — family of machine learning methods that is capable of entirely skipping feature extraction in order to operate directly on raw data.

IN THIS CHAPTER

- Setting up your datasets
- Measuring how well your model performs
- Understanding measures beyond accuracy

Chapter 3
Validating and Measuring Performance

You can measure the performance of machine learning and deep learning models using various terms, such as accuracy, detection rate, and false positive rate. But to really understand what these metrics are telling you, it's vital to fully understand the context.

This chapter explains why context is so critical for judging whether a statement such as "We have a 100 percent detection rate" has any real meaning. It also spotlights the need to conduct rigorous tests to ensure that results aren't contaminated by various biases. The chapter goes into detail on how machine learning models should be tested and compared.

Training, Validation, and Test Sets

REMEMBER

With any machine learning method, including deep learning, you first use a set of training data, known as the *training set*, to train the machine learning model. Once that's done, you use a separate *test set* — including samples that weren't included in the training — to measure the performance of the model. It's important to ensure that the test set is *representative*, meaning that it truly reflects the distribution available in the real world.

Training and testing

WARNING

There must be absolute, stringent separation between the training and test sets. Some sort of contamination between the test and train sets, or datasets that aren't representative, cause probably nine out of ten mistakes made by researchers new to machine learning. Even experienced researchers who are getting a bit too loose with the rules can be tripped up by these issues. Contamination can be incredibly subtle and mess up the whole process.

Imagine trying to train a machine learning model that detects tanks hiding behind trees. To do so, you create a large dataset of images containing tanks behind trees, and a large dataset of images showing only trees. Then you randomly split the data into train and test sets. You train a machine learning module on the training data, then test it on the test data, and you're delighted to see that it obtains 100 percent accuracy on the test data.

On the face of it, this sure seems like a rigorous test. You decide to put this module through further tests on newly obtained data (again, images of trees with and without tanks hiding behind them). To your surprise, the accuracy is only 50 percent this time, and that happens to be the same accuracy level as pure chance. What has gone wrong?

The usual suspect in this situation is data contamination. For example, if the images of tanks behind trees were taken on a cloudy day, and the images of trees without tanks were taken on a sunny day, the machine learning module may simply learn to detect clouds instead of tanks! Bear in mind that for each image there's an associated label, which in this case is "0=no tank," while "1=tank." But a black box machine learning model simply tries to improve its accuracy by better separating between different classes. So, the model might inadvertently learn that 0=cloud and 1=no cloud, and not pay any attention to tanks at all.

Here's another example. Assume you'd like to train a model that detects malicious files. Now say that you gather a dataset of malicious files from various sources. For benign files, you use files created by Microsoft that come directly with Windows and are definitely not malicious. Again, you separate the train and test sets, train and then test a model, and score 100 percent accuracy. Your model has correctly classified all malicious and benign files — great work!

Not so fast. Here again, the results are biased. By using files created by Microsoft only as benign samples, your model might not really be classifying whether a file is malicious or not. More likely, it just learned to classify whether a file is created by Microsoft or not, by looking to see whether it contains a "Microsoft" string. The problem is that the dataset is not representative. To remedy this bias, your benign dataset should contain many different files created by many different developers, not just Microsoft.


REMEMBER

That's two examples, each illustrating how subtle contamination of the data can completely skew the results. You can start to see why it's extremely important to be sure the test data is completely separated from train data, and that the data is representative of the type and distribution of data that you'd encounter in the real world.

Setting aside a validation set

It's critical that insights gained from test data aren't incorporated back into training of the models. If they were, it would sort of be like teaching to the test, which is not an effective way to learn in the real world. In many cases, though, you'll want to train various models, measure their performance on new data, then use the insights gained for the next round of training and improvements.


REMEMBER

You can't use a test set for this purpose, so it's common to set aside another portion of data as a *validation set*. A validation set is similar to a test set in the sense that it's not used in training the model. The difference is that the insights obtained from the validation model can be used for further training and improvement. That way, the test set remains the ultimate test, striving to replicate as closely as possible the conditions that will be encountered by the model once it hits the real world.

Just how stringent should the measures be that you take to ensure the reliability of the test set? Sometimes they can be quite stringent. For example, in the cybersecurity domain when training a model for detecting malicious files, you'll need to add considerations that separate train and test datasets temporally, so that they're from different time periods. For example, use all data until mid-2017 for training, and use files that first appeared after mid-2017 in the test set, because this is how the model will be used in the real world, where new malicious file types appear every day.

Understanding the Intricacies of Measuring Performance

So, you've created train and test datasets that are comprised of representative samples, and you've taken rigorous measures to ensure that there are no biases or any contaminations between the datasets. Now you can feel confident that you can objectively measure the performance of the trained model on the test set.

Accuracy is important, but isn't everything

REMEMBER

The most important measure of a model's performance is *accuracy*. It's a pretty simple measure, defined as "the total number of correct classifications divided by the total number of samples." For example, if the test set contains 1,000 images with and without dogs in them, and the model correctly classifies 900 of them, then its accuracy would be 900 divided by 1,000, which equals 0.9 or 90 percent.

While accuracy is clearly a very useful measure, in some cases it can be misleading. For instance, assume you've trained a fraud detection module that processes financial transactions and detects which ones are fraudulent. Given a test set containing 100,000 legitimate transactions and 100 fraudulent ones, it obtains 99.9 percent accuracy, or a single misclassification for every 1,000 transactions. Sounds pretty good, right?

In reality, this result is completely meaningless. Imagine you used an unintelligent system that simply classifies everything as legitimate. On this dataset, it would stumble across the correct answer nearly all the time, because 100,000 of the transactions were, in fact, legitimate. It would only be wrong on the 100 fraudulent transactions. Do the math: 100,000 correct results divided by the 100,100 total samples and you get 0.999, which happens to be 99.9 percent! Your unintelligent system just scored 99.9 percent accuracy without even trying.

How can this be? This particular example reveals that the accuracy measure isn't a good indication of performance when you have an *unbalanced dataset*. That's a dataset in which you have a lot more samples of one class compared to another.

REMEMBER

The thing is, many real-world problems involve extremely unbalanced datasets. One hopes that if you're measuring for fraud, your data will contain mostly legitimate transactions. Your business would be in a world of hurt otherwise. The same can be said for malware detection. Your dataset is likely to have a whole lot more benign files than malicious ones, which in and of itself is a good thing, but it makes an accuracy measure potentially misleading.

To get a better picture of the performance of a model in these cases, you need some additional metrics.

True and false, positives and negatives

REMEMBER

It's best to proceed by introducing a few terms that are used within all measurements of performance. In classification tasks, samples that have the target label are known as *positive*, while *negative* refers to those that don't. For example, say the task is to detect whether each file is either malware or a legitimate file. All the samples which are malware are positive samples, while the rest are negative samples.

Note that the terms positive and negative refer to the test data, rather than the decisions of the machine learning module. Meanwhile, the term *true* refers to a correct classification by the machine learning module, while *false* refers to an incorrect classification.

With that in mind, you can examine all decisions made by a classifier group under one of the following four terms:

- **True positive (TP):** This sample contains a target class, and the classifier has detected it correctly. For example, the sample is malicious, and the classifier correctly detected it as malicious.
- **False positive (FP):** The sample does not contain a target class, but the classifier incorrectly detected it as being in the target class. In the example of the malware detector, the sample is a legitimate file, not malicious, but the classifier falsely declared that it is malware.
- **True negative (TN):** The sample does not contain a target class, and the classifier detected that fact correctly. For example, the sample is a legitimate file rather than a malicious one, and the classifier got the answer right.

- **False negative (FN):** The sample contains a target class, but the classifier incorrectly detected it as otherwise. To carry on with the example, the sample is a malicious file, but the classifier falsely detected it as a legitimate file. It missed detecting the malware.

These four measurements are often presented together in what's called a *confusion* matrix. Now that the definitions are established, check out how these terms can be used within additional measurements for accurately evaluating a classifier.

Using performance metrics

A good classifier has a high detection rate along with a low false positive rate. These two simple measures could formally be defined as follows.


REMEMBER

The *detection rate* is often referred to as *recall*, and is sometimes known as *sensitivity* or the *true positive rate* (TPR). Basically, this is a measure of the number of correct positive detections divided by the total number of positive samples. Check Figure 3-1 for the equation.

$$Recall = \frac{TP}{All\ positive\ samples} = \frac{TP}{TP + FN}$$

FIGURE 3-1: The detection rate.

For example, imagine that there are 1,000 malicious files (positive samples). The classifier correctly detects 900 of them (TP = 900). The recall or detection rate would be 900 divided by 1,000, which is 0.9 or 90 percent. Not surprisingly, you want the recall for the classifier to be as high as possible.


REMEMBER

The *false positive rate* (FPR), also known as the *false positive ratio*, is a measure of the number of erroneous positive detections divided by the total number of negative samples. It's spelled out in Figure 3-2.

$$False\ Positive\ Rate = \frac{FP}{All\ negative\ samples} = \frac{FP}{FP + TN}$$

FIGURE 3-2: The false positive rate.

For example, imagine there are 1,000 legitimate files, which are negative samples. Ten of them are erroneously classified as malicious (FP = 10). This means the FPR would be 10 divided by 1,000, which equals 0.01 or 1 percent. As you can imagine, you want the FPR to be as low as possible.

Note that these metrics are useful only when presented together. They don't tell you anything meaningful on their own. Consider what happens if you have a classifier that simply detects everything as positive, such as an antivirus that pronounces every file to be malicious. You just achieved 100 percent recall. Or, imagine that your classifier calls everything negative (such as an antivirus that never finds a virus). You certainly won't ever get a false positive, so your false positive rate is 0 percent. But that's clearly not a meaningful stat on its own, either.

REMEMBER

Another useful metric is *precision*. This is a measure of the number of correct positive detections divided by all positive detections (correct and incorrect). Check Figure 3-3.

$$Precision = \frac{TP}{TP + FP}$$

FIGURE 3-3: Measuring precision.

Precision doesn't care about the detection rate or the false positive rate. Instead, it asks the question, "Out of all the samples that are pronounced positive, what portion really is positive?" Naturally, the higher the precision, the better the classifier model.

TIP

Many additional performance metrics are available to researchers, and some of them use a combination of the metrics outlined in this chapter. For example, in the context of medical research the two measurements often used are sensitivity (as explained earlier in this section, it is the same as recall), and *specifity,* which is the *true negative rate* (a measure of the number of correct negative detections divided by all negative detections). That said, the basic metrics presented here provide a very good estimate of the real-world usability of a classifier model.

So, next time you hear someone mention the detection rate, be sure to ask about the false positive rate, and vice versa. The graveyards of machine learning are full of classifier models that have tallied impressive detection rates, but also had a false positive rate that was too high to make them deployable within a real-world solution.

IN THIS CHAPTER

- Introducing the original neural network
- Going artificial
- Learning to train an artificial brain
- Understanding different neural networks
- Taking neural networks to the deepest level

Chapter 4
Understanding How Neural Networks Evolved into Deep Learning

It takes a bit of a history lesson to understand just what deep learning is and how it emerged to become such a driving force in the world of machine learning. Today's remarkable deep learning was yesterday's stepchild that only a few researchers really believed in.

This chapter covers the development of neural networks, beginning with a brief explanation of the biological brain from which they take inspiration. It spotlights what has changed from the basic neural networks of the 1970s to the deep neural networks of the 2010s — what people now call deep learning. Following this journey provides a deeper understanding of what's known as deep learning.

The Biological Brain Was the First Real Neural Network

REMEMBER

The human brain consists of tens of billions of small processing units known as *neurons*. These neurons are connected to each other via *synapses*. Get a picture of these brain components in Figure 4-1.

You've probably read that the human brain has different regions — such as the visual cortex and auditory cortex — that each perform a certain task. These differences mainly arise from the input each region receives. For example, when the optic nerve transfers signals (the input) from our eyes to a certain region in the brain (the processing area), the neurons in that area *learn* to process these signals, and form the visual cortex.

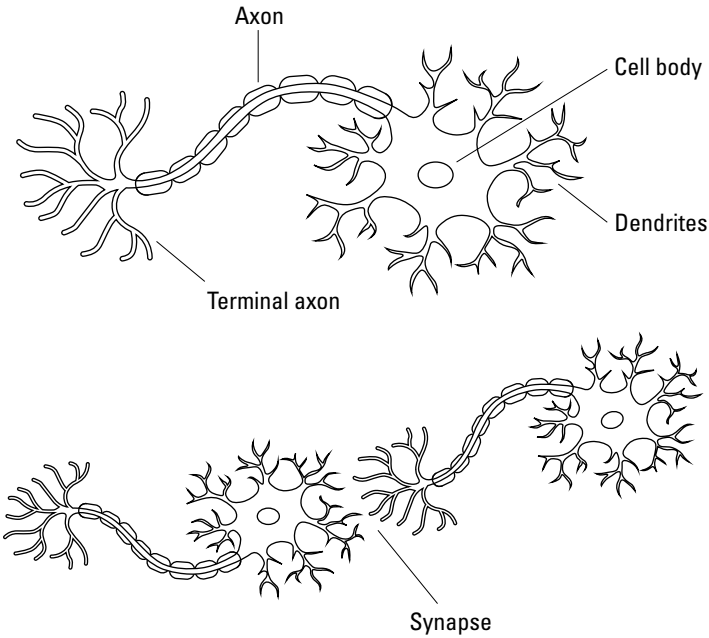


FIGURE 4-1: Making connections in the brain.

We can refer to the neurons as general processing units, which are agnostic of the data they process. The learning process itself takes place when the connection strength between neurons is formed,

removed, strengthened, or weakened. In other words, everything humans learn, everything we remember, everything we do, is the result of synaptic activity in the brain.

You might consider the *cerebral cortex* to be the most "interesting" part of our brain, because it's associated with our high-level cognitive capabilities. Mammals are the only animals that have a *cerebral cortex*. Figure 4-2 shows the main parts of the human brain.

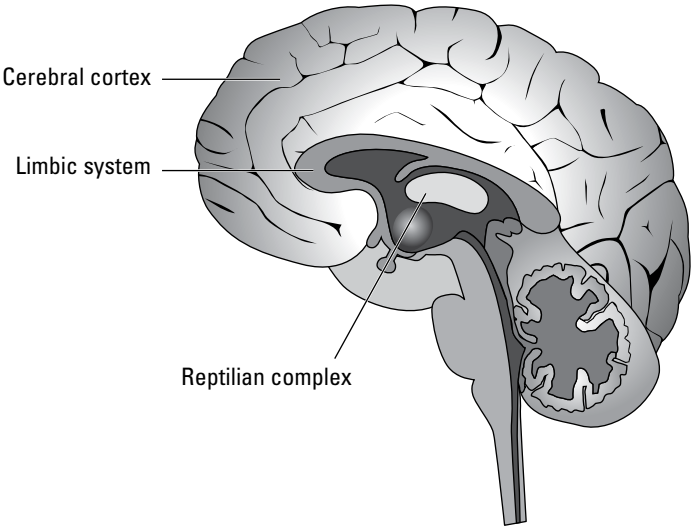


FIGURE 4-2: A few parts of the brain.

Why is it that humans are smarter than all other animals? Brazilian neuroscientist Suzana Herculano-Houzel invented a novel method for accurately counting the number of neurons in the brain. Her research suggests that intelligence is correlated with the number of neurons in the cerebral cortex. The higher this number, the higher the intelligence. An elephant has a brain with a much larger mass, but the human brain's cerebral cortex has a far greater absolute number of neurons.

Artificial Neural Networks

You can trace the origins of artificial neural networks back to 1943. That's when researchers Warren McCulloch and Walter Pitts proposed a very simple model for what they called an *artificial neuron*. The idea was that this artificial neuron would

receive signals from other neurons in the form of input numbers. It would then fire signals or output numbers to other neurons. Fifteen years later, psychologist Frank Rosenblatt created the *perceptron*, a simple neural network with just two layers: the input layer and the output layer.

In Figure 4-3, you see three input neurons and two output neurons. Each connection between two neurons contains a numeric weight value. The input flowing to each output neuron is the sum of all inputs multiplied by the weight connecting them to the output neuron.

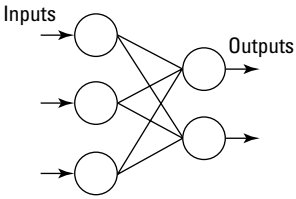


FIGURE 4-3: Connecting neurons in a perceptron neural network.

TECHNICAL STUFF

For example, assume two input neurons X and Y are connected to an output neuron Z. The weight between X and Z is 0.5, and the weight between Y and Z is 1.4. The input value for X is 1, and for Y it's 2. Multiply those values by their respective weights, and you end up with a total input value of (1 x 0.5) + (2 x 1.4) = 3.3.

These simple perceptron networks were very limited with respect to what they could learn. There was early excitement about the field, but progress pretty much came to a halt in the late 1960s.

REMEMBER

Researchers found they could substantially expand the capabilities of neural networks by adding *hidden layers*, which were layers between input and output layers. What they ended up with were known as *multilayered neural networks* or *multilayered perceptrons* (MLP), as shown in Figure 4-4. But they could not be trained using the conventional training mechanisms.

REMEMBER

Flash forward to the early 1980s, when scientists Paul Werbos, David Rumelhart, and their colleagues invented a new method called *backpropagation* that was capable of training multilayered neural networks. Backpropagation remains to this day the backbone algorithm for training neural networks. It's used for nearly all state-of-the-art deep neural networks today.

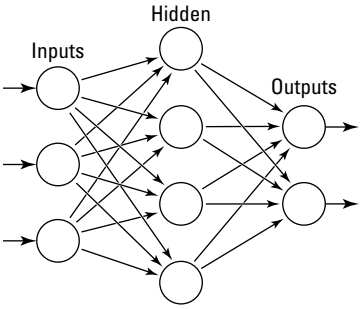


FIGURE 4-4: Multilayered perceptron.

Training a Neural Network with Backpropagation

The underlying mathematics of backpropagation can be a bit complex, but the general principle is quite intuitive. To gain an understanding, imagine that you'd like to train a "cat detector." You've assembled a training dataset with 10,000 images containing cats, plus 10,000 images that do not contain cats. This is the kind of binary classification problem outlined in Chapter 2. Now, assume all these images are of size 30 x 30 pixels, and they are in grayscale (that means a single value describes each pixel, rather than three values for red, green, and blue).

A neural network for this classification task would contain an input layer, a few hidden layers, and an output layer. In this example, the input layer should contain 900 neurons, which is 30 times 30, determined by the pixel size. The output layer should contain two neurons (one neuron represents the "no cat" class, and the other neuron represents a "cat" class).

Note that in this example, even a single output neuron would have sufficed — it would fire 0 for "no cat" and 1 for "cat." For practical convenience, though, the number of neurons at the output layer is typically equal to the number of classes in a classification task.

Add two hidden layers as well, the first containing 400 neurons and the second containing 100 neurons. Why those numbers? They're pretty arbitrary decisions, really, typically set through trial and error, and based on the experience and intuition of the experts.

TECHNICAL STUFF

So, this neural network has four layers, containing 900, 400, 100, and 2 neurons. The neurons between each pair of adjacent layers are connected to each other via weight (these connections are also referred to as *synapses*, just like in the human brain).

REMEMBER

The simplest type of networks are *fully connected* neural networks, which means that all the neurons in each layer are connected to all the neurons in the subsequent layer. The current example would have 900 x 400 weights in the first layer, 400 x 100 weights in the next layer, and 100 x 2 weights in the output layer.

The weights of a neural network are initialized randomly, and they're usually small values around zero. As the neural network is initialized, it doesn't yet have any useful knowledge. Given an image, it won't do any better than a coin toss when it comes to detecting whether the image contains a cat or not.

The neural network obtains the knowledge about how to recognize a cat through the training process. At each point during the training process, you'll select a training sample and feed it into the neural network (in this example, the sample is an image which may or may not contain a cat).

The training is done in two stages:

- **Feed-forward:** Neurons in the input layer send their values to the neurons in the next layer. Those neurons aggregate all the input, pass it through an *activation function* (a function that receives all inputs to a neuron and decides what the neuron's output should be), and fire the results onward to the next layer of neurons. This continues until the output neurons fire their results.
- **Backpropagation:** When the output neurons have fired their result, you can measure their *output errors*, which are the difference between the produced output and the *expected output*, which we already have available. In the cat detector example, assume a sample that's an image containing a cat. You would expect the output neuron that represents the cat class to fire 1, and the other neuron to fire 0. Now imagine that they fired 0.6 and 0.4. The respective output errors would be 1 – 0.6 = 0.4 and 0 – 0.4 = –0.4. The idea is to backpropagate these errors through all the layers (from the output layer back to the input layer), and use the backpropagation algorithm to update the weights of the neural network so that it does better next time.

REMEMBER

Note that each training sample results in very small updates to the weights. The network trains through many iterations over the entire training set. A single pass over the whole training set is referred to as an *epoch*. Typically a network trains for several hundred epochs before it converges. All the actual training is performed through gradual updates made to the weights of the network, and these updates are exclusively made during the backpropagation phase.

When you've finished training a neural network, you must further test the accuracy using a set of samples that weren't used during the training. This involves the *test set*, and its purpose is to make sure the network has not simply "memorized" the training set without learning the principles behind it. This memorization effect is referred to as *overfitting*, and you can employ a number of methods to keep that from happening while encouraging the network to *generalize* instead.

REMEMBER

If the results on the test set are satisfactory, you can now use the neural network for real-world *prediction*. You're putting the neural network through the same process, feeding in new data. The network outputs the results, but only the feed-forward phase takes place during prediction.

Types of Neural Networks

The neural network spotlighted in the previous section was a simple one. In practice, there are many types of neural networks, used for different tasks. Following are some examples.

Fully connected neural network

This is the simplest form of neural network, in which all the neurons in each layer are connected to all the neurons in the subsequent layer. Take a look at Figure 4-5 for a sense of how this plays out.

Fully connected networks are popular because they are robust, and because they don't assume anything about the properties of the input. Also note that because all the neurons in each layer are connected to all the neurons in the subsequent layer, the actual position of a neuron within a layer really doesn't matter.

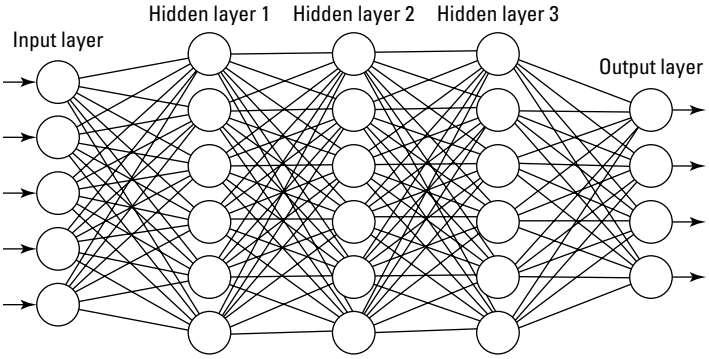


FIGURE 4-5: A fully connected neural network.

Recurrent neural network

REMEMBER

Each current input always matters as a neural network goes through the decision-making process. For sequential tasks, those that involve an element of time, making the correct decision doesn't rely just on the current input. Previous inputs are relevant, too.

Language understanding is an ideal example of a task that requires you to present previous data samples to the network. For example, imagine you want to train a network for the task of next character prediction. Each time you're presented with a single character, you want to predict what the next character is going to be. What's likely to come after the letter "o"? Who knows? But if you're given the characters "n, e, t, w, o" in a sequence, you'd obviously predict the next character to be "r," followed by "k." Being able to consider that whole sequence, not just a single preceding letter, made your prediction possible.

An even more complex example would be the text "United Kingdom's capital is." Bet you could guess the next character pretty easily, and in fact, the next six: "L," followed by "o, n, d, o, n."

When you're creating a neural network, how should you provide these types of inputs? One basic approach would be to use a "sliding window" of a pre-specified size. For example, instead of providing only the last character to the neural network, you'd provide the last ten characters. This would work just fine for the first of the previous two examples ("network"). It would not have helped for the second one ("London"), because the relevant information was more than ten characters back.

REMEMBER

A *recurrent neural network* (RNN) is a special class of neural network allowing for an indefinite memory of previous events. You give it this special power by adding recurrent connections between the neurons in the hidden layers. In other words, these recurrent connections are weights between the neurons in the same layer. They provide the values of neurons in previous time steps — for example, the connection is from time *t minus 1* to time *t*. Check out Figure 4-6.

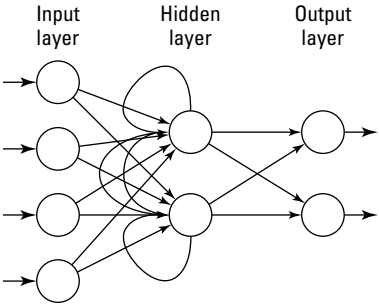


FIGURE 4-6: A recurrent neural network.

RNNs are the ticket whenever you're presenting sequential data. Language understanding is a great example, and so are cases involving financial time-series, such as algo-trading.

You won't be surprised to learn that as the years have passed, more advanced variants of RNN have emerged. These allow for considerably higher accuracy when learning long-term patterns and relationships.

REMEMBER

The most popular variant is *LSTM*, an acronym for *long short-term memory*, invented in 1997 by computer scientists Sepp Hochreiter and Jürgen Schmidhuber. LSTM contains several additional "gates," which allow for a better control over what data is remembered for longer time periods.

Take a look at Figure 4-7 to see what a single LSTM neuron contains.

Sparsely connected neural network

REMEMBER

The opposite of fully connected is known as sparsely connected. The term *sparsely connected neural nets* refers to any type of network that isn't fully connected. In other words, only a portion of the neurons between two adjacent layers are connected to each other.

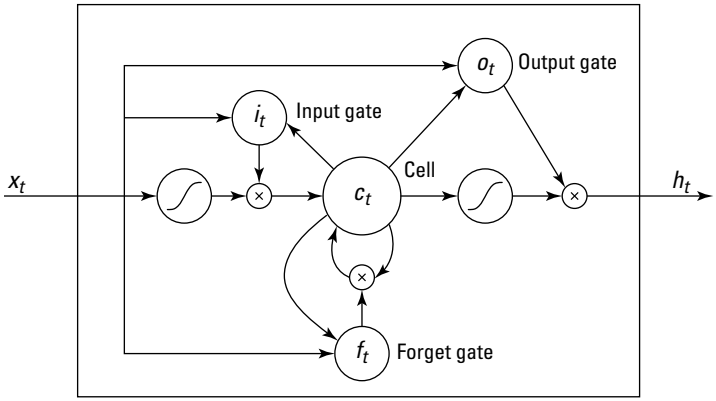


FIGURE 4-7: Long short-term memory.

How do you decide which neurons are connected? These selective connections are determined by considering some of the properties present in the data.

REMEMBER

Among the most popular and successful variants among sparsely connected networks is one known as the *convolutional neural network* (CNN). These networks are typically applied to computer vision problems. CNNs rely on the fact that in real-world images, there's a high correlation between adjacent pixels. A fully connected network would look at all the pixels in an image at once, while a CNN will use a small receptive field that slides over the image. In Figure 4-8, the receptive field of each neuron in the convolutional layer is of size 3 x 3.

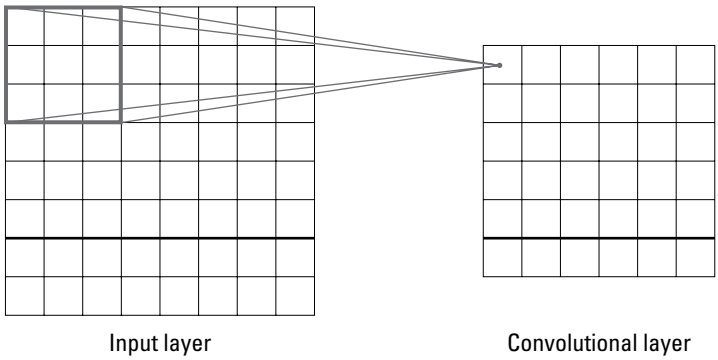


FIGURE 4-8: A convolutional neural network.

The Evolution of Neural Networks Continues

Interest in neural networks picked up in the 1980s because of the invention of the backpropagation algorithm, which allowed for training neural networks with several layers. So why has their popularity picked up steam only in more recent years?

Consider that by the 1990s and early 2000s, there were many successful use-cases of neural networks. But at that point, conventional machine learning methods obtained mostly similar results, without all the complexity of neural networks. And if you're getting more or less similar accuracy results, there are good reasons why you would not prefer neural networks:

- **They're complex to train.** Neural networks are poorly understood and require arbitrary design decisions. These decisions include the number of layers, the number of neurons in each layer, the type of neural network, and all kinds of associated parameters. And these multiple decisions are based on cumbersome trial and error, or more often intuition.
- **They're CPU- and memory-intensive.** Neural nets contain a much larger number of trainable parameters in comparison to other machine learning methods (most notably, their weights). Also, training is performed by gradual updates, which requires processing of all the training data hundreds of times. It takes a lot of processing power and memory to make that happen.

Given these potential drawbacks, very few scientists and research groups have been focusing on neural networks through the years. Until more recently, most preferred to spend their time and effort on other areas of machine learning that seemed more promising.

REMEMBER

In fact, neural networks were so frowned upon in late 1990s that simply mentioning the term "neural networks" was often enough to have your research disregarded by the mainstream machine learning community. Mainstream researchers would raise their eyebrows and look bewildered.

And who could blame them for feeling that way? At that point you could train rather shallow neural networks, and the training was excruciatingly time-consuming. Most important, for the majority of applications, the results were inferior to other traditional machine learning methods that happened to be much faster and more convenient to deploy.

Even as late as 2012, a 409-page long book titled *Machine Learning,* written by the editor-in-chief of the most respectable journal on machine learning, didn't cover neural networks at all!

I certainly experienced this on a personal level. About five years ago, I was talking to a friend who headed one of the world's leading computer vision groups. He referred to deep learning as "another transient hype." That was then. Today, his entire research group is focused only on deep learning, obtaining amazing breakthroughs that barely use any of the traditional image processing methods that were prevalent in previous decades.

The Deepest of the Neural Networks

Deep learning is by far the hottest field within artificial intelligence today. That's pretty remarkable when you consider that until a few years ago, just about everyone other than a few research groups had completely given up on neural nets.

REMEMBER

What caused the shift toward deep learning? There were two main factors:

- The advent of improved training methods that made it possible to train deep neural nets
- The use of GPUs (graphics processing units), which allowed for up to 100 times faster training

Training deeper neural networks

As explained earlier in this chapter, neural networks are trained using the backpropagation algorithm. That is, input is fed forward until an output is produced, the output error is calculated, and then working backward through the layers, these errors are backpropagated and the weights are updated.

Even in the 1990s it was understood that deeper neural networks, containing a larger number of layers, would outperform shallower neural networks that had just a few layers. The problem was that it was very difficult to train deep neural networks because of what is known as the *gradient vanishing* problem.


TECHNICAL STUFF

To illustrate this point intuitively, understand that the output error is a signal containing information for updating the weights of the neural network. During the backpropagation phase, after you work back through each layer, this signal gets weaker and weaker. Thus, after several layers the signal strength is not sufficient to contain enough useful information for updating the weights. This made it practically impossible to train neural networks that had more than two or three hidden layers.

During the past few years, several inventions have helped address the vanishing gradient problem. The most important innovation uses a certain activation function that preserves the signal as it passes through numerous layers during backpropagation. Combining this breakthrough with multiple other improvements has allowed researchers to train deeper neural networks, even those that contain many tens of layers and billions of synapses.

With that issue addressed, the advances of deeper neural networks really start to emerge. For one thing, they allow for a hierarchical pattern learning structure. Using this structure, higher layers gradually learn and recognize more complex patterns.

Also, deep neural networks don't require the use of traditional feature extraction, where you must decide in advance which are the few, most important features in the input data. Instead, deep neural networks can receive raw data such as pixels, and use their deep layers as feature extractors. This process can extract complex patterns that human experts can't manually specify (there are more details on feature extraction in Chapter 2).

Using GPUs for training

The relatively sudden interest in deep neural networks would never have happened without the use of graphics processing units (GPUs) instead of central processing units (CPUs).

As their name suggests, GPUs are hardware designed specifically for processing graphics data. Given that, it may be surprising that they are actually suitable for training neural networks as well.

To answer that puzzle, take a look at what GPUs actually do. Graphics processing requires the hardware to render values of millions of pixels in parallel. When you're playing a game with high-end graphics, the GPU must work hard to continuously process the pixel values for each frame on the screen. That's the secret behind a smooth gaming experience, without any lag.

TECHNICAL STUFF

GPUs can do this because they are designed as massively parallel processing units, which excel at performing the same instructions over many different data values in parallel. Does that requirement sound a bit familiar? Yes, this is exactly what neural networks must do, too. The training process of neural nets (both in the feed-forward and backpropagation phases) requires calculating values of large numbers of neurons in parallel.

Nvidia, the largest GPU producer, spotted the huge potential. The company seized the opportunity and developed a more robust and versatile software suite called Cuda. This software allows for developers to directly tap into the parallel processing capabilities of GPUs, even if they aren't dealing with graphics.

REMEMBER

Training a deep neural network on a GPU is typically tens of times faster than training it on a CPU. In other words, training tasks that would take several months on a CPU could be completed in a few days on GPU. Thanks to this capability, nearly all deep learning training today is conducted on GPUs, and each improvement in GPU processing capability directly enables improvements in deep learning training times.

IN THIS CHAPTER

- Spotlighting the advantages of deep learning
- Seeing improvements in computer vision
- Analyzing and understanding text
- Recognizing human speech
- Playing more satisfying computer games
- Taking the cybersecurity case

Chapter 5
Looking at Applications of Deep Learning

What can deep learning do for you? A better question is, what can't it do? Compared with the various earlier incarnations of artificial intelligence and machine learning, the principles of deep learning really knock the ball out of the ballpark.

This chapter explores why deep learning works so much better in the real world than other methods of machine learning. Then it takes a sector-by-sector journey through the many ways deep learning has had an amazing impact on the world. It details the deep learning advantages in computer vision, and explores how deep learning has advanced the ability of computers to analyze and understand text. It documents the advances deep learning has brought to speech recognition as well as synthesis. It spells out how deep learning is advancing the popular world of computer gaming. And, of vital importance, it outlines why deep learning may be the ultimate answer to the ever-growing threats to cybersecurity.

Advantages of Deep Learning

To apply traditional machine learning to any problem, you first must perform a lot of preprocessing. In particular, you have to determine in advance which are the important properties or features in the problem domain. As explained in more detail in Chapter 2, this process requires manual feature specification, and you end up disregarding most of the raw data.

That chapter's example of a dog detector, shown here in Figure 5-1, shows how this works. But any dog lover will tell you a dog is a whole lot more than a bunch of numbers. Even with the best feature specifications, it simply isn't possible to grasp the complex patterns in the data.

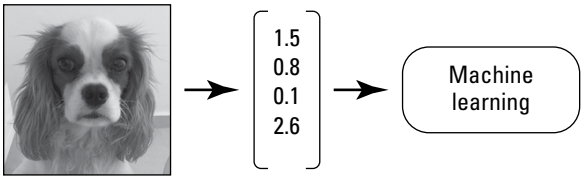


FIGURE 5-1: Traditional machine learning for detecting dogs.

Deep learning, on the other hand, doesn't rely on feature extraction. It's the first family of methods within machine learning that doesn't need it, and at the moment it's still the only one.


REMEMBER

Instead of human experts explicitly specifying the features beforehand, deep neural networks use their deep hierarchy of layers to learn the complex features by themselves. The idea is illustrated in Figure 5-2. This is very similar to how the human brain learns new concepts by being exposed to new data.

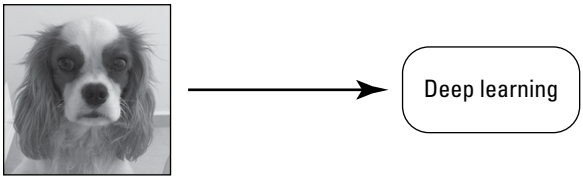


FIGURE 5-2: Using deep learning to identify dogs.

This robustness of deep learning has brought about great improvements in most benchmarks of computer vision, speech

recognition, language understanding, and other domains. In past years, improvements were gradual, spread over the course of many years. Deep learning has been creating benchmark improvements of 20 to 30 percent a year.

REMEMBER

With deep learning, many tasks previously viewed as impossible are now achievable. Add it all together and you can view deep learning's contribution as the greatest leap ever in the history of artificial intelligence.

Here's how it was summed up by Geoffrey Hinton, considered to be the father of deep learning. Honoring a career dedicated to neural network research, he was presented the IEEE/RSE James Clerk Maxwell Medal in 2016, and this is what he said in his acceptance speech:

Fifty years ago, the fathers of artificial intelligence convinced everybody that logic was the key to intelligence. Somehow we had to get computers to do logical reasoning. The alternative approach, which they thought was crazy, was to forget logic and try and understand how networks of brain cells learn things. Curiously, two people who rejected the logic-based approach to AI were Turing and Von Neumann. If either of them had lived I think things would have turned out differently . . .. Now neural networks are everywhere and the crazy approach is winning.

Just what kind of impact has deep learning had in the real world? Read on for examples of how it has revolutionized nearly every field to which it has been applied.

Computer Vision

Some of the most dramatic improvements brought about by deep learning have been in the field of computer vision. For decades, computer vision relied heavily on image processing methods, which means a whole lot of manual tuning and specialization. Deep learning, on the other hand, ignores nearly all traditional image processing, and it has resulted in dramatic improvements to every computer vision task.

ImageNet is a great example. It's the largest publicly available dataset of labeled images, with more than 10 million images

sorted into a thousand different classes. Since 2010, there's been an annual ImageNet Large Scale Visual Recognition Challenge, aiming to measure the classification accuracy of different computer vision models. Accuracy is measured on a test set of images that have not previously been used for training the models.

These are real-world images, many of which show more than a single object. Each predicting module is allowed a total of five guesses from that list of a thousand different categories, and if one of them is correct, it is declared that the image has been classified correctly. The final results are measured in terms of classification error rate, which is the percentage of images classified incorrectly.

The results are illustrated in Figure 5-3. In 2011, the best computer vision models relying on traditional machine learning and image processing obtained a 25 percent error rate. In 2012, when a deep neural network joined the competition, the error rate dropped to 16 percent, and since then deep learning has cut the error rate to 4 percent or less.

REMEMBER

Wow, that's almost as good as what a person could do, right? Actually, it's even better. As a comparison, humans typically achieve an error rate of about 5 percent in this challenge. The bottom line is that deep learning has cut the error rate by 20-plus percentage points, and has now even surpassed human accuracy!

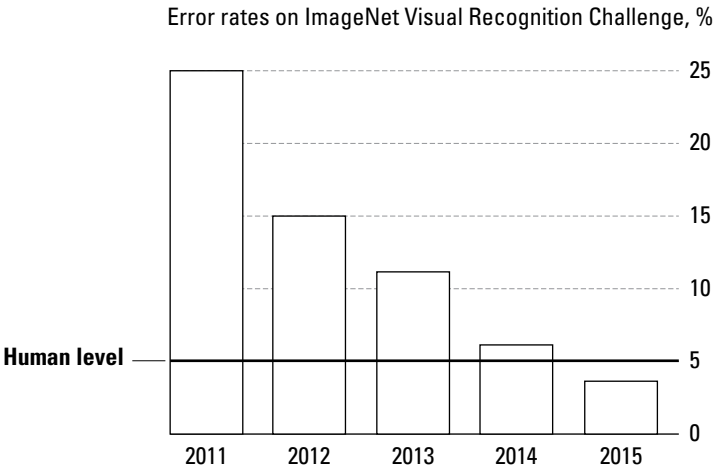


FIGURE 5-3: Deep learning outdoes humans.

So, what kinds of things can computer vision recognize with the help of deep learning? Today, all state-of-the-art object recognition modules rely solely on deep learning. Google Photos is a prime example. It automatically uses deep learning to classify images and group them together. Because of deep learning, you can search your Google Photos albums for "Cavalier King Charles Spaniel," and it provides all the relevant results, even if you have not done any manual labeling.

REMEMBER

Find that hard to believe? Just check out Figure 5-4. As you can see, in most of the images the dog is not clearly visible, but Google Photos saw it. Traditional non-deep learning modules would have great difficulty detecting that there is a dog in the image, let alone accurately classifying its breed.

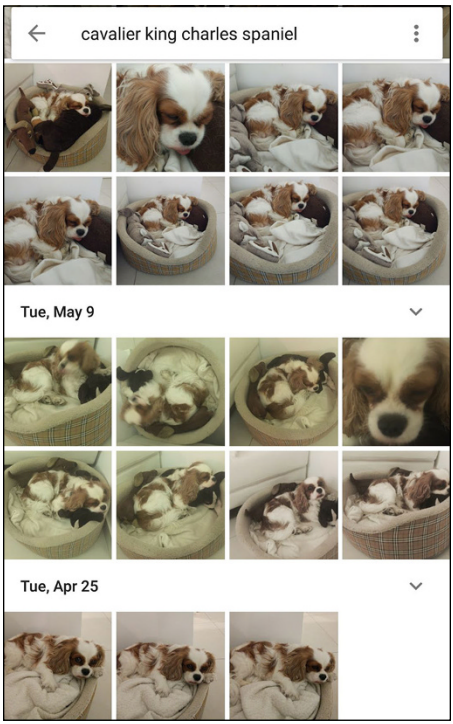


FIGURE 5-4: Google Photos knows these are photos of a King Charles Cavalier Spaniel dog.

Although different categories of objects are visually very different from one another — cars, for example, really don't look like dolphins — faces are much more similar to each other, with differences that often are very subtle. For decades, face recognition software relied on years of image processing methods that improved only gradually and incrementally. Today, deep learning has resulted in a huge improvement in the accuracy of face recognition, without relying on traditional image processing features.

End-to-end deep learning can be applied to practically any computer vision task involving classification. For example, artist classification is an interesting problem — can deep learning take a look at a painting and identify who painted it? Traditional image processing has worked its way up to 78 percent accuracy on a test set of three painters: Renoir, Rembrandt, and van Gogh. In 2016, deep learning succeeded in improving the accuracy to 96 percent, without relying on any feature due to image processing.


REMEMBER

Deep learning's huge accuracy improvement in computer vision has resulted in numerous real-world breakthroughs. These days deep learning is performing on a par with human radiologists in detecting many forms of cancer, and it's widely used in medical image analysis. A company known as Zebra Medical, for example, is one of the leading organizations using deep learning for medical image analysis.

And then there's deep learning behind the wheel. All of today's state-of-the-art autonomous driving modules rely on deep learning, and their accuracy and safety measures will soon exceed those of human drivers.


REMEMBER

In all these example areas, traditional machine learning was given a try before deep learning took its turn, and the application of deep learning resulted in a huge improvement. Beyond that, deep learning has been tackling issues that were previously considered completely intractable.

Imagine that you take a nice picture, and want to turn it into something resembling a painting. Your favorite painting is van Gogh's *The Starry Night,* or perhaps Edvard Munch's *The Scream.* It would be great to turn your photo into a painting in the specific style of those classics.

In 2015, researcher Leon Gatys and colleagues used deep learning for what they called "artistic style transfer." They described how

deep learning can be used to learn the artistic style of a painting, and then use that knowledge to transform another existing picture into a painting. Figure 5-5 shows an experiment using the same technique.

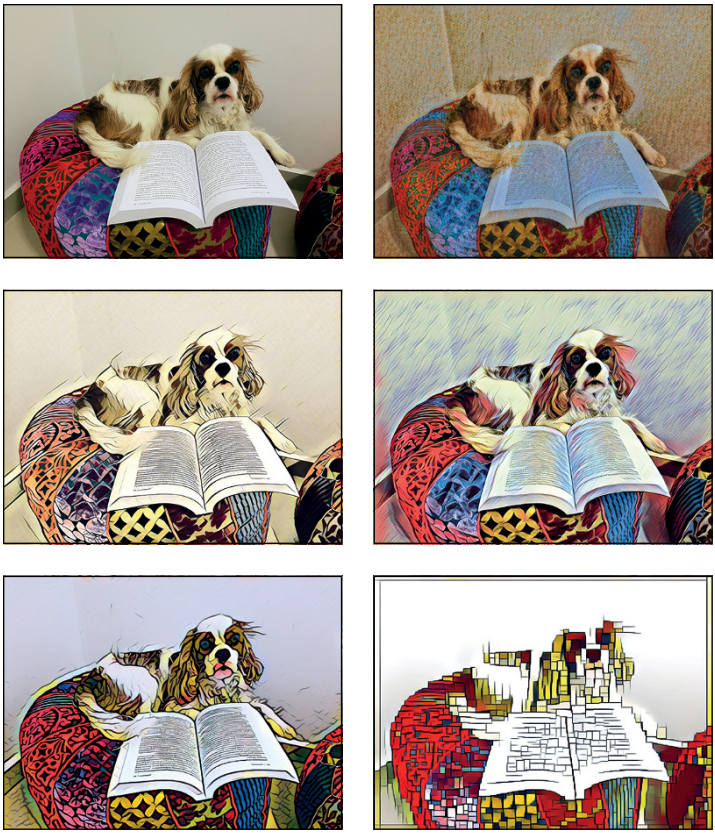


FIGURE 5-5: Turning photos into paintings.

The top-left image is the original photo. Each of the other images is a transformation of the original photo, turned into a painting based on a particular style.

For nearly all computer vision tasks, convolutional neural networks are used most often. That's due to the presence of location correlations in the input data.

Text Analysis and Understanding

During the past few years, deep learning has been successfully applied to numerous problems in text analysis and understanding. These include document classification, sentiment analysis, automatic translation, and that kind of thing, with usually dramatic improvements. Recurrent neural networks are especially useful here, because of the sequential nature of textual data.

REMEMBER

One of the most important contributions in this area has been deep learning's ability to train a language model from raw text data. Imagine that you have large amount of text in a certain language — let's say it's a dataset a billion characters long. You can train a neural net that receives a character and tries to predict what the next character is going to be. At first it simply guesses random characters, but it gradually learns the vocabulary in this language. Then, to improve its prediction accuracy, it learns grammar, context, and other important traits. The higher the accuracy at this "next character prediction" becomes, the better it understands the language. It is developing a better language model.

Deep learning language models can even be trained together with deep learning models for computer vision, providing results that until just recently were considered impossible in the near future. For example, image captions can be generated as the result of a deep learning model. They don't rely on any manual image processing or natural language processing. Just the fact that the caption is a correct English sentence is amazing in itself — after all, nobody taught English to the model. It learned the language by itself by training on large amounts of English text. The understanding of what's happening in the image, combined with the use of language to describe it, is incredibly close to what humans can do.

REMEMBER

Still more amazing are the results of training a deep learning model to answer questions about an image it sees. This problem is more complex, because the model needs to understand the question, know where to look in the image to find the answer, find it, and then use language to accurately provide the answer.

Deep learning can also be used to generate a completely new image based on a text description. These images can be created entirely by a neural network, pixel by pixel, without relying on any previous image.

Speech Recognition

Speech recognition includes several major families of problems. The most widely researched is *voice to text,* or taking the spoken word and turning it into text on the screen. The problem may not seem all that complex at first glance, because it seems like it's just a matter of converting each sound to a corresponding character. In fact, though, it's one of the most complex areas in signal processing.

REMEMBER

The auditory cortex in our brain is trained over several years in childhood to recognize voice and convert it to language, and humans become very good at this, despite the fact that completely different sentences can sound very similar vocally. An example Geoffrey Hinton frequently cites involves the phrases "recognize speech" and "wreck a nice beach." They certainly sound very similar, but their meaning is completely different, and humans can only tell the difference because they understand the language and are always looking for context clues. In the same way, in order to perform speech recognition, a model needs to have a good understanding of the underlying language and context.

While the progress in speech recognition has been incremental over many decades, in recent years deep learning has revolutionized this field in the same way it has moved others into the future. Traditional speech recognition relied on cumbersome feature extraction processes, which were limited in their nature. Deep learning, on the other hand, is capable of directly operating on raw data, and being trained on large datasets of audio recording. It can exceed the accuracy of traditional models by a huge margin, with accuracy improvement of 20 to 30 percent.

Today most smart assistants rely on deep learning, and their understanding level is rapidly increasing in question answering tasks. Google Assistant, which relies almost entirely on deep learning, has the highest accuracy in the latest benchmarks, followed by continuously improving smart assistants from Microsoft (Cortana), Amazon (Alexa), and Apple (Siri).

REMEMBER

Deep learning has also been successfully applied to speech generation or synthesis, often known as *text to voice*. Recently, Google DeepMind presented a novel method called WaveNet for directly training deep learning models on raw audio so that they can

generate their own raw audio. Their results show near human performance for voice and speech generation.

Speaker recognition — or recognizing who is talking — is another area where deep learning has improved accuracy substantially. This is especially important for national security. Fifth Dimension, one of the leading developers of investigation platforms based on deep learning, successfully employs speech recognition such that a terrorist making an anonymous phone call can be identified by matching his voice sample against a large dataset of known voices.

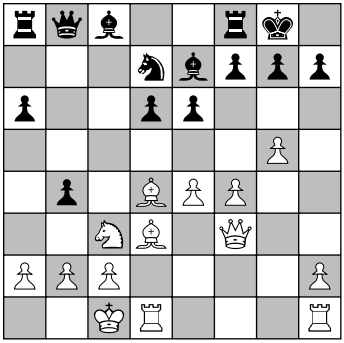
Computer Games

Since the dawn of computer science, computer chess was an especially challenging problem. Goethe called chess "the touchstone of the intellect," and Alan Turing, the forefather of modern computer science, designed the first chess-playing algorithm before he could even run it on any computer.

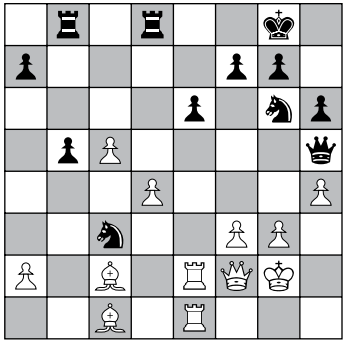
Computer chess, while being one of the most researched fields within AI, has not lent itself well to the successful application of conventional learning methods, because of its enormous complexity.

In a recent work titled "DeepChess," which won the Best Paper Award at the International Conference on Artificial Neural Networks, my co-authors and I demonstrated how end-to-end deep learning could be applied for training a chess-playing program, without any prior knowledge. By merely training on millions of chess positions taken from grandmaster games, the program reaches a super-human performance level. Figure 5-6 shows some moves selected by DeepChess, which cannot be found by most regular chess programs.

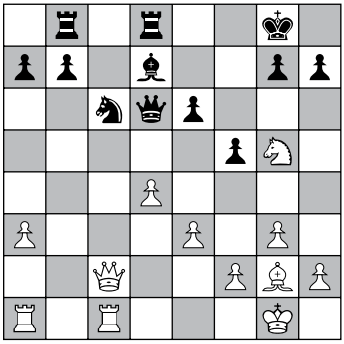
The game of Go is another complex game, which for many years could not be tackled by any traditional machine learning approach. Google DeepMind used deep learning to train its "AlphaGo" program and defeat Lee Sedol, one of the strongest human Go players.



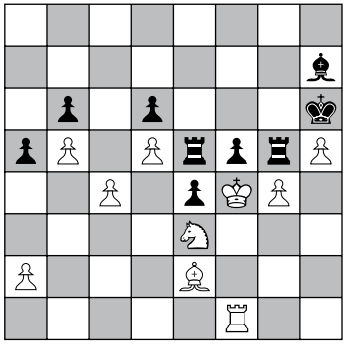
Tal - Larsen
Move: **Nd5**



Aronian - Leko
Move: **Re5**



Alekhine - Golombek
Move: **d5**



Seirawan - Kozul
Move: **c5**

FIGURE 5-6: DeepChess has amazing moves.

Cybersecurity

One of the most crucial real-world problems today, one that concerns every large and small company, is cybersecurity. More than a million new malware threats (malicious software) are created every single day, and sophisticated attacks are continuously crippling entire companies — or even nations — by targeting critical national infrastructures, as would happen in the case of nation-state cyberattacks.

WARNING

There are many, many cybersecurity solutions out there, but all are struggling to detect new malware. It's easy to mutate a malware and evade detection by even the most sophisticated cybersecurity solutions, which perform dynamic analysis on files and use traditional machine learning.

Traditional cybersecurity

For nearly two decades, antivirus solutions mainly relied on signatures to detect malicious files. In their simplest form, the signatures could be a list of file hashes. In more sophisticated cases, such as most advanced antivirus solutions today, they detect the presence of certain features in files, such as a string that is associated with a malicious file family.

Although antivirus solutions today are quite effective for protecting against previously existing malware, they are incapable of detecting the millions of new malicious files that are continuously created. Due to these severe limitations, in the past few years a new generation of more advanced solutions have emerged, focusing on detection of new malware.

REMEMBER

Most of these "next gen" cybersecurity solutions use *sandboxing*, which is the dynamic analysis of suspected files. This is a lengthy process and it can't be used for threat prevention, only detection. Detection means finding and stopping the malware after it has already started running and has potentially caused damage, while prevention means stopping the malicious file before it is able to start running in the first place.

Many of these solutions also rely on machine learning to increase their detection rates. Applying traditional machine learning in this case can require several years of effort devoted to feature extraction.

For example, given a Windows executable file, what are its most important features? The most obvious features would be function calls (API), strings, and tens or hundreds of additional handcrafted features.

This feature extraction phase has several severe limitations that become particularly evident in cybersecurity:

- **Losing most of the data:** Even the most extensive feature extraction process can address only a small fraction of the

data within the file. This is similar to the way feature extraction in photo analysis ignores most of the useful data by converting a raw image file into a list of features. In the case of cybersecurity, converting a complex computer file into a set of features means only a very small amount of information is retained. That makes it easy for malware developers to engineer a very small change in the malware and evade detection.

- **Missing the subtle interactions:** The extracted features grasp only the obvious patterns, ignoring all the complexities in the file. Not only does the feature extraction process ignore most of the data in the file, even those features it does consider are of limited nature. The process misses subtle interactions that are critical for distinguishing between legitimate and malicious code.
- **Serving one format at a time:** Feature extraction is format-specific. That means a completely different feature extraction process should be used for each file format. Thus, a machine learning-based solution for the Windows operating system has to rely on one feature extraction procedure for executable files, while there must be another for PDF files, another for Office files, and so forth. Each such procedure is manually developed over a long period of time. For the same reason, most machine learning-based solutions support only one operating system, perhaps Windows or Android. To add support for additional operating systems would require a completely new feature extraction procedure from scratch, and that is usually not feasible.

Deep learning for cybersecurity

On the face of it, deep learning addresses all the limitations of traditional machine learning in cybersecurity. Specifically, deep learning processes raw data and does not rely on feature extraction. That doesn't make it easy, though. Applying deep learning is much more challenging in the domain of cybersecurity.

For example, unlike in computer vision, where different image sizes can be adjusted to a pre-specified size and fed into a neural network, a computer file can be of any size, from a few kilobytes up to many gigabytes. Also, different file formats have different file structures, and none of these structures has any obvious local

correlations that could be used by neural network types such as convolutional neural networks.

REMEMBER

Despite these challenges, deep learning has been successfully applied to cybersecurity. Deep Instinct has demonstrated how a dedicated deep learning framework adapted specifically for cybersecurity can overcome the difficulties mentioned in the preceding section, and can train a deep learning model on raw files.

The training phase is performed in the laboratory, using hundreds of millions of malicious and legitimate files of different file formats. This training process takes only a single day or so using GPUs. After the training has converged, the resulting deep learning model is only a few tens of megabytes in size, and it can provide a prediction for any given file within a few milliseconds. And it achieves that speed on the average CPU. The GPU is used only in the training phase, not the prediction phase. Because of that, it can be deployed on any endpoint using only a negligible amount of resources, and provide full pre-execution prevention.

The deep learning-based model is capable of obtaining a much higher detection rate and a much lower false positive rate for new, previously unseen files, when compared with the best traditional machine learning solutions available. And because deep learning is agnostic to file types, it can be applied to any file format, and even to any operating system, without requiring substantial modifications or adaptations. Compare that to traditional machine learning, where each effort pretty much has to start from scratch, and you can see one more reason why deep learning is so powerful.

In addition to determining whether a file is malicious or not, deep learning can be used to identify what type of malware it is (for example, ransomware or Trojan). Recently my co-authors and I presented a paper at the International Conference on Artificial Neural Networks demonstrating how deep learning can even detect which nation-state is behind an attack (for example, China or Russia).

IN THIS CHAPTER

- Searching for experts
- Finding frameworks
- Deciding on deep learning
- Predicting the future

Chapter 6
Exploring Deep Learning in the Real World

The things that deep learning can accomplish are remarkable. If, indeed, deep learning is a magical black box, one that doesn't require any preprocessing or feature extraction, one that just needs to be fed large amounts of data for training, then why isn't everyone using deep learning?

This chapter explores some of the reasons deep learning is where it is, and why it has not yet reached its potential. The chapter explores barriers of entry and other challenges, and covers how deep learning is applied in real-world settings.

Grasping the Scarcity of Deep Learning Experts

Until a few years ago, the field of neural networks was one of the most unpopular areas within machine learning. Just a few research groups focused entirely on this field.

The recent, nearly overnight success of deep learning has resulted in a huge gap between the demand for deep learning experts and

the very limited supply of such expertise. It's such a shortfall that small deep learning startups are being basically "acqui-hired" by giant companies for up to hundreds of millions of dollars. For example, DeepMind was acquired by Google for $650 million. And recently, the *New York Times* reported the following:

Typical A.I. specialists, including both Ph.D.s fresh out of school and people with less education and just a few years of experience, can be paid from $300,000 to $500,000 a year or more in salary and company stock. . . . Well-known names in the A.I. field have received compensation in salary and shares in a company's stock that total single- or double-digit millions over a four- or five-year period. And at some point they renew or negotiate a new contract, much like a professional athlete.

These kinds of high financial incentives have lured most deep learning researchers from academia into industry. That just exacerbates the scarcity of deep learning experts, because many of the researchers who used to train the next generation of researchers are no longer on campus.

REMEMBER

The bottom line is that at the moment, nearly all top deep learning experts work for one of the five giants. Google has by far the largest number of deep learning experts, followed by Facebook, Microsoft, Amazon, and Apple. Other companies, even the largest Fortune 500 companies, are struggling to bring in much-needed deep learning experts.

What makes deep learning so difficult that these experts are both scarce and indispensable for its successful application? Truth is, the field of neural networks is in many ways more of an art than a science. It's poorly understood, and successful training of deep learning models can heavily rely on the experts' intuition.

TECHNICAL STUFF

For example, there isn't really a systematic way to answer what type of neural network should be used, how many layers, how many neurons, and what other hyper-parameters should be selected, such as learning rate, weight decay, momentum, batch normalization, activation function, and tens of other parameters. These kinds of decisions are made by experts, based on their experience and intuition.

WARNING

In fact, it's often the case that inexperienced researchers spend months experimenting with different deep learning models without success. A single experienced expert could have successfully trained the model within a single week. No wonder these experts earn the big bucks.

Recognizing the Limitations of Publicly Available Frameworks

Until a few years ago, any application of deep learning required implementing an entire deep learning software framework. This is no easy feat in any field, and is by orders of magnitude more complex for deep learning. Typically, tens of complex algorithms must be implemented entirely in Cuda for running on GPUs, and that's a task so arduous that only a few companies have ever successfully accomplished it.

REMEMBER

Today, several deep learning frameworks are publicly available. The most popular is Google's TensorFlow, and others include PyTorch by Facebook and Cognitive Toolkit by Microsoft. The availability of these frameworks has significantly boosted the research in deep learning, because today any deep learning researcher can directly implement ideas on these frameworks and run the experiments without having to write a single line of low-level code on GPUs.

These publicly available frameworks are all developed by researchers for researchers. They're very good for academic research, but they aren't efficient enough for use in many real-world commercial solutions. They have some significant shortcomings in terms of speed, memory inefficiencies, and dependencies on many external libraries. The other issue is that these publicly available frameworks implement only high-level building blocks. It's practically impossible to modify the intrinsic implementation to adapt them to a specific task.

WARNING

Today's publicly available frameworks also suffer from performance inefficiency in inference mode or prediction mode. They usually require dedicated hardware for providing real-time predictions, and several companies are now working on dedicated hardware for accelerating inference mode. That's nice, but their

application on devices without dedicated hardware — including trying to run on standard CPUs — is severely limited.

The current available deep learning frameworks are efficient mainly for computer vision tasks. In these cases, only a single algorithm, namely a convolutional neural network, is required. For this reason, the vast majority of deep learning-based companies today use them primarily for computer vision. Even then, they typically can't be applied in inference mode without the use of dedicated accelerator hardware, and so their application to edge devices is limited.

Combine the impact of these two limitations — the scarcity of deep learning expertise and the severe limitations of publicly available frameworks for commercial deployment — and you find that a few giant companies account for nearly all real-world applications of deep learning within commercially deployed products. Outside that, deep learning has been successfully employed mostly in computer vision tasks alone.

Knowing When to Apply Deep Learning


TIP

While deep learning has revolutionized many fields, not every problem is suitable for it. In order to apply deep learning, the following three conditions should hold true:

- Traditional machine learning methods have insufficient results.
- A large amount of training data is available.
- The data type is complex and poorly understood.

Read on for more detail on these conditions, because they are crucial for deciding whether deep learning should be applied.

Traditional machine learning has limited success


TIP

Deep learning is by far the most complex family of methods within machine learning. For that reason, it makes sense to fully consider traditional machine learning first before concluding that deep learning is needed for a certain task. If the results obtained

by traditional machine learning are insufficient, that's a good first sign that deep learning may be the answer.

Of course, deep learning will nearly always improve upon the results obtained by traditional machine learning. The question is, are these improvements worth the tremendous additional efforts required? Often they are, but not always.

If you think about the deep learning use cases outlined in Chapter 5 — such as computer vision, text understanding, speech recognition, cybersecurity, and computer games — those are areas in which the results obtained by traditional machine learning have been far from optimal. That fact has helped to justify the use of deep learning, which has provided dramatic improvements.

Large amount of training data is available

The more training data that's available, the greater the potential performance improvement deep learning can deliver over traditional machine learning. While traditional machine learning relies on manually selected features, deep learning processes raw data, so it has to learn all the nonlinear features and patterns as well during training. You'll likely need more than a hundred thousand samples in order for deep learning to significantly outperform alternative methods.

REMEMBER

Also, with traditional machine learning there's a ceiling above which additional training data doesn't really improve accuracy. And you hit that ceiling pretty quickly. Deep learning, on the other hand, is especially good at continuously improving the more data it has. Even as you reach scales of tens of millions of training samples, deep learning keeps improving its accuracy.

For most tasks within computer vision, speech recognition, text understanding, cybersecurity, and computer games, you can gather millions of training samples. That may be millions of labeled images, billions of characters, millions of malicious files, and that kind of thing. That's one reason behind the substantially superior results yielded by deep learning.

WARNING

If, on the other hand, your particular problem doesn't have sufficient training data, you should opt for traditional machine learning. If the dataset is too small, deep learning probably won't outperform traditional machine learning significantly enough to warrant the major additional effort.

Data type is complex and poorly understood

Traditional machine learning is limited by its feature extraction requirements. As discussed elsewhere in this book, for most complex problems it's extremely difficult to manually craft features that truly encompass the patterns present. Inevitably, most of the relevant data and patterns are disregarded. Deep learning, on the other hand, performs feature learning by itself by processing raw data, and thus obtains an accurate internal representation.

In the cases of problems for which the data is already represented as features, or if feature extraction is straightforward without the loss of important patterns, then the advantages offered by deep learning would be more limited. In such cases, traditional machine learning probably would be sufficient.


TIP

That's not so, however, with all the use cases covered in Chapter 5. There, the data is incredibly complex. The key question to ask is, "Can you convert the raw data into a list of features without losing important information?" You can't do that for computer vision, speech recognition, text understanding, cybersecurity, or computer games. That's another reason why deep learning handles such tasks so much better.

Selecting a Deep Learning-Based Solution

A few years ago, Dan Ariely, one of the prominent researchers in behavioral economy, discussed Big Data by saying "everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it." You could say pretty much the same thing about deep learning today.

On one hand, deep learning is everywhere. You continuously hear about exciting breakthroughs and amazing feats that just a few years ago would have been considered science fiction. On the other hand, the barrier of entry for real-world application of deep learning remains high, and few companies are capable of deploying such solutions.

Thus, many companies attempt to associate themselves with this success by blurring the messages. You frequently encounter PR stating "the most advanced AI" or "using advanced machine learning." These claims are absurd, because nearly every product created in the past 20 years already uses some sort of AI in the broad sense! And the only major breakthrough in AI and machine learning during the past few years has been deep learning.


TIP

What about those solutions that directly claim that they are doing deep learning? It isn't always easy to distinguish those that are truly using deep learning from those that are only claiming to do so. Here are some key questions that will help shed some light on that question:

- **Who are the deep learning researchers involved?** Developing state-of-the-art deep learning solutions requires experts, and they're in short supply. Ask about the researchers, and learn about their past experience in deep learning (which will likely be mostly academic experiences).
- **What deep learning framework is being used?** Developing a deep learning framework is an extremely complex task that only a few companies have successfully accomplished. Publicly available deep learning frameworks are good enough for most computer vision tasks, but are too inefficient to be used for most other products.
- **What hardware is being used for training the models?** Currently only GPUs are relevant for training deep learning models. If the answer to this question doesn't involve GPUs, it's a dead giveaway that deep learning is not being used. This may change in the future as other suitable hardware is developed, but that's the way it is today.
- **What input data is being fed to the models?** Deep learning is typically applied directly to raw data. Answers that involve "feature extraction" or manual preprocessing suggest that traditional machine learning is used, rather than deep learning.

Looking at the Future of Deep Learning

As neural networks have moved from being unpopular and frowned upon, to being the talk of the town, many more people are singing their praises. Expect many traditional methods to go by the wayside, and watch for every leading business in every industry to heavily rely on deep learning in the coming years.

One promising area of research is *evolutionary computation*. Just as neural networks take inspiration from how the human brain works, evolutionary algorithms (for example, genetic algorithms or genetic programming) take inspiration from the evolution in nature. After all, natural evolution is the only "algorithm" that evolved single-cell organisms billions of years ago into the complex intelligent organisms of today, and during this process evolved the brain as well. Currently, training a deep learning model requires lots of trial and error, and heavily relies on the experience and the intuition of the experts. By evolving such deep learning models, rather that manually specifying their configuration, researchers may be able to create more robust models, and do so much more efficiently.

For the past 30 years, backpropagation has remained the main method for training neural networks and is the backbone of every deep learning system. As Geoffrey Hinton, the father of deep learning and one of the inventors of backpropagation, recently observed, researchers would need to find a better, more "brain inspired" alternative to backpropagation in order to achieve further substantial improvements in deep learning.

As mentioned in Chapter 4, research in real intelligence (within humans and other animals) strongly suggests that there is a correlation between the number of neurons in the brain and overall intelligence. This seems to hold true for artificial neural networks, as well. One could argue that the current state-of-the-art deep learning models are very similar to the neural networks from the 1990s, with the main difference that they contain about a million times more synapses or connections.

REMEMBER

You can extrapolate from this trend. Within the next several decades, the size of deep learning models will grow exponentially, and I expect we will find better and better neural network architectures and parameters. My belief is that as that happens, we

will get close to or surpass human-level intelligence, as shown in Figure 6-1. My guess is that this will happen within most of our lifetimes.

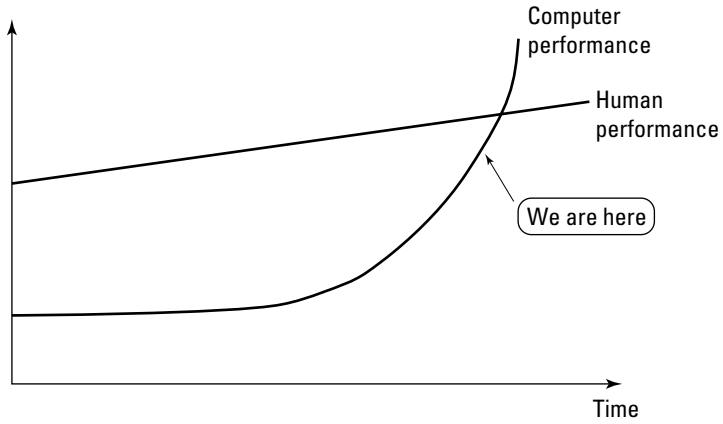


FIGURE 6-1: The future of deep learning.

REMEMBER

When that happens, many tasks currently performed by humans will be done just as well (or usually much better) by machines. Although workers may face an initial wave of job losses, many new jobs will be created. Instead of competing with AI, humans and machines will work together to achieve complex tasks not feasible today.

IN THIS CHAPTER

- Unscrambling the confusion
- Comparing technologies
- Setting AI on fire
- Obtaining the data and hardware
- Learning about frameworks
- Tracking the accelerating progress

Chapter 7
Ten Key Takeaways about Deep Learning

The other chapters in this book review many aspects of deep learning, from historical background to use cases to amazing breakthroughs. Read on through this final chapter for a summary of the most important things to remember about deep learning.

AI versus ML versus DL

Artificial intelligence (AI), machine learning (ML), and deep learning (DL) are neither synonyms nor competing technologies. Deep learning is a subfield of machine learning, which is itself a subfield of artificial intelligence. That's simple enough, but is ripe for confusion. Some players may even be blurring the lines a bit on purpose.

The next time you hear about a solution that is "using AI," be sure to ask what type of AI it's using. Is it machine learning? If it is machine learning, then ask what type of machine learning it is.

Despite the current excitement about AI in general, it's important to note that deep learning is really the only current breakthrough that is fueling this excitement. Outside of deep learning, there hasn't been significant progress in other areas of AI and machine learning in recent years. So if this "using AI" solution isn't using deep learning, it isn't riding the current wave that's causing all the excitement.

Deep Learning Is Inspired by Our Brains

Deep learning, also known as deep neural networks, takes inspiration from the human brain. The brain contains tens of billions of neurons, which are connected to each other via synapses. All the training and learning is done by optimizing the values of these synapses.

Similarly, deep neural networks contain many layers of artificial neurons, which are connected to each other via synapses. Deep neural nets are trained by being exposed to large datasets of training data.

Deep Learning Is Different from Traditional Machine Learning

Deep learning is the first family of methods within machine learning that does not require feature extraction, and for now, it's the only one. Traditional machine learning relies on manually specified properties or features that are used to convert raw data into a set of values. Those are then fed into the traditional machine learning module.

The result is that most of the data is discarded, and the features that are used grasp only the trivial linear patterns. This ignores the most interesting (and impossible to manually describe) features.

By directly operating on raw data, deep learning uses all the patterns available. That's the main reason for the large performance gap in its favor, compared with traditional machine learning.

Deep Learning Is the Greatest Leap in AI Performance Ever

For years we were used to seeing very gradual improvements in all areas of artificial intelligence and even the field of computer science as a whole. Deep learning is a unique case that has resulted in an overnight revolution in entire areas of artificial intelligence and computer science.

In many tasks that required domain-specific processing developed through decades of research, deep learning provides mind-boggling accuracy improvements of 20 percent or 30 percent or more. It completely ignores all the domain-specific knowledge, and instead processes raw data alone.

Deep Learning Requires a Large Amount of Training Data

Because deep neural nets learn the features by themselves from the raw data, rather than being provided a set of pre-specified features, it takes a much larger amount of data to obtain significant performance improvements. That's one reason for the dramatic improvements in such areas as computer vision, speech recognition, text understanding, cybersecurity, and computer games. Those are areas for which millions of training samples are readily available.

Deep Learning Requires Dedicated Hardware

Deep learning involves training large models consisting of up to billions of synapses in the neural networks, and the models are typically trained on millions of training data samples. This is extremely intensive work from a computer processing perspective, and it was infeasible before graphics processing units (GPUs) became widely used for training deep neural nets.

A single GPU is many times faster than a single central processing unit (CPU) for training neural nets. That allows experts to train within days what otherwise would take months. As a result, all deep learning tasks require using GPUs for the training phase.

Publicly Available Frameworks Have Benefits and Limitations

Publicly available deep learning frameworks such as Google's TensorFlow have revolutionized academic research in this field. They allow every researcher to quickly experiment with deep learning models.

While these frameworks work great for research, they have some severe limitations and inefficiencies for real-world deployment within mature products. In many cases, bringing deep learning products to market requires developing dedicated deep learning frameworks, and that's a complex task.

Deep Learning-Based Products in the Real World

The five tech giants — Google, Microsoft, Facebook, Amazon, and Apple — are investing hundreds of millions of dollars annually for acquiring deep learning startups and recruiting scarce deep learning researchers. And their efforts are paying off. More and more products they offer are utilizing deep learning. These companies' products are becoming increasingly adept at face recognition, object recognition, automatic translation and recommendation, speech recognition, and speech synthesis, to name a few technologies already in various stages of commercialization.

Outside the five giants, more and more startups have successfully utilized deep learning to provide significantly enhanced results in

their field. Notable examples of gains include autonomous driving, medical image analysis, and cybersecurity.

Despite the current high barriers of entry for deep learning, it is expected that in the years ahead, every leading solution in every domain will be based on deep learning. Those products remaining behind with non-deep learning technology will be at a significant disadvantage.

Deep Learning Progress Is Accelerating

Already in 1970s, neural networks stirred lots of interest and excitement. But when it became evident that the reality at the time did not match the high expectations, nearly everyone lost interest and neural networks became a pariah.

Back then, the excitement was based on what neural networks *could* achieve. Today, the excitement is based on what neural networks *have already* achieved. This book covers examples of significant problems that deep learning has already revolutionized. Looking at the pace of deep learning research and the results obtained, there's a clear exponential growth trend that shows no sign of slowing down. Check it out in Figure 7-1.

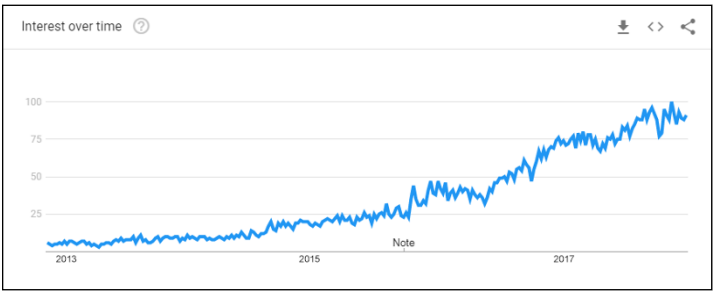


FIGURE 7-1: Deep learning's impact continues to grow. The graph is from Google Trends, showing interest in the keyword "deep learning."

Deep Learning's Impact on Our Lives

It would be appropriate to conclude this book by citing the following quote by Andrew Ng, the former Baidu chief scientist, focused on how deep learning will affect our lives:

About a century ago, we started to electrify the world through the electrical revolution. By replacing steam-powered machines with those using electricity, we transformed transportation, manufacturing, agriculture, health care, and so on. Now, AI is poised to start an equally large transformation on many industries. For example, the IT industry is totally transformed by AI. FinTech is also being totally transformed. Health care is starting to be transformed, and there are huge opportunities there. Self-driving is an industry built on AI. Others industries, like search engines and food delivery, are also supported by AI. The only industry which will not be transformed will probably be hairdressing.

Help shape the future of deep learning

Can a computer tell the difference between a dog and a cat? Can it distinguish a West Highland White Terrier from a Bichon Frise? With the help of some serious artificial intelligence (AI), yes, a computer can do that. It's possible through an exciting subset of AI known as *deep learning*. This book is your guide to this remarkable new world of deep learning, which has revolutionized AI. Find out what deep learning is, how it developed from earlier methods of AI and machine learning, and why it's so much more powerful.

Inside...

- Explore the basics of machine learning
- Learn how neural networks evolved
- Understand the deep learning revolution
- Set up datasets and measure performance
- Learn when and how to use deep learning

Eli David is a leading expert in the field of deep learning and evolutionary computation and has published more than forty papers in these fields, several of which received Best Paper awards. He is the co-founder of Deep Instinct, the first company to apply deep learning to cybersecurity.

for dummies®
A Wiley Brand

ISBN: 978-1-119-48358-8

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.