Can Learning Algorithm But Complex

Generated on: 2025-11-02 22:52

Can Learning Algorithm But Complex

Planner Decisions

Seg	Decision	Trace | Best
1	NORMAL	LLM: normal | 0.58
2	EXTERNAL	Stop: budget reached | 0.00
3	NORMAL	LLM: normal | 0.69
4	NORMAL	LLM: normal | 0.68
5	NORMAL	LLM: normal | 0.80
6	NORMAL	LLM: normal | 0.69
7	EXTERNAL	Stop: budget reached | 0.00
8	NORMAL	LLM: normal | 0.77
9	NORMAL	Normal search | 0.85
10	NORMAL	LLM: normal | 0.77
11	EXTERNAL	Stop: budget reached | 0.00
12	NORMAL	LLM: normal | 0.81
13	NORMAL	Stop: budget reached | 0.63
14	NORMAL	LLM: normal | 0.77
15	NORMAL	LLM: normal | 0.75
16	NORMAL	LLM: normal | 0.75
17	NORMAL	LLM: normal | 0.72

Can Learning Algorithm But Complex

Segment 1: Features Algorithm Artificial Can Child

Lecture segment: Many artificial intelligence tasks can be solved by designing the right set of features to extract for that task, then providing these features to a simple machine learning algorithm. For example, ... a strong clue as to whether the speaker is a man, woman, or child.

Below are closely-matching textbook excerpts supporting this segment.

ughout computer science and even daily life. In computer science, operations such as searching a collection of data can proceed exponentially faster if the collection is structured and indexed intelligently. People can easily perform arithmetic on Arabic numerals, but find arithmetic on Roman numerals much more time-consuming. It is not surprising that the choice of representation has an enormous eect on the performance of machine learning algorithms. For a simple visual example, see figure . 1.1 Many artificial intelligence tasks can be solved by designing the right set of features to extract for that task, then providing these features to a simple machine learning algorithm. For example, a useful feature for speaker identification from sound is an estimate of the size of speaker's vocal tract. It therefore gives a strong clue as to whether the speaker is a man, woman, or child. However, for many tasks, it is dicult to know what features should be extracted. For example, suppose that we would like to write a program to detect cars in photographs. We know that cars have wheels, so we might like to use the presence of a wheel as a feature. Unfortunately, it is dicult to describe exactly what a wheel looks like in terms of pixel values. A wheel has a simple geometric shape but its image may be complicated by shadows falling on the wheel, the sun glaring o the metal parts of the wheel, the fender of the car or an object in the foreground obscuring part of the wheel, and so on. 3

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 18)

(score=0.58)

Segment 2: Difficult What Wheel Between Categories

Lecture segment: r the speaker is a man, woman, or child. However, for many tasks, it is difficult to know what features should be extracted. For example, ... Unfortunately, it is difficult to describe exactly what a wheel looks like in terms of pixel values. ... or an object in the foreground obscuring part of the wheel, and so on. Suppose we want to separate two categories of data by drawing a line between them in a scatterplot.

[No relevant excerpt found in provided references]

Segment 3: Representation Also Learning Representations Adapt

Lecture segment: ng a line between them in a scatterplot. One solution to this problem is to use machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as representation learning. Learned representations often result in much better performance than can be obtained with hand-designed representations. They also allow AI systems to rapidly adapt to new tasks, with minimal human intervention.

Can Learning Algorithm But Complex

Below are closely-matching textbook excerpts supporting this segment.

CHAPTER 1. INTRODUCTION

Figure 1.1: Example of dierent representations: suppose we want to separate two categories of data by drawing a line between them in a scatterplot. In the plot on the left, we represent some data using Cartesian coordinates, and the task is impossible. In the plot on the right, we represent the data with polar coordinates and the task becomes simple to solve with a vertical line. Figure produced in collaboration with David Warde-Farley. One solution to this problem is to use machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as representation learning. Learned representations often result in much better performance than can be obtained with hand-designed representations. They also allow AI systems to rapidly adapt to new tasks, with minimal human intervention. A representation learning algorithm can discover a good set of features for a simple task in minutes, or a complex task in hours to months. Manually designing features for a complex task requires a great deal of human time and eort; it can take decades for an entire community of researchers. The quintessential example of a representation learning algorithm is the autoencoder.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 19)

(score=0.69)

the data. When it is nearly as dicult to obtain a representation as to solve the original problem, representation learning does not, at first glance, seem to help us. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts. Figure shows how a deep learning system can represent the concept of 1.2 an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges. The quintessential example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a dierent mathematical function as providing a new representation of the input. The idea of learning the right representation for the data provides one perspective on deep learning. Another perspective on deep learning is that depth allows the computer to learn a multi-step computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Sequential instructions oer great power because later 5

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 20)

(score=0.65)

reat deal of human time and eort; it can take decades for an entire community of researchers. The quintessential example of a representation learning algorithm is the autoencoder. An autoencoder is the combination of an encoder function that converts the input data into a dierent representation, and a decoder function that converts the new representation back into the original format. Autoencoders are trained to preserve as much information as possible when an input is run through the encoder and then the decoder, but are also trained to make the new representation have various nice properties. Dierent kinds of autoencoders aim to achieve dierent kinds of properties. When designing features or algorithms for learning features, our goal is usually to separate the factors of variation that explain the observed data. In this context, we use the word "factors" simply to refer to separate sources of influence; the factors are usually not combined by multiplication. Such factors are often not 4

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 19)

(score=0.64)

CHAPTER 1. INTRODUCTION AI Machine learning Representation learning Deep learning Example: Knowledge bases Example: Logistic regression Example: Shallow autoencoders Example: MLPs Figure 1.4: A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. Each section of the Venn diagram includes an example of an AI technology. 9

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 24)

(score=0.61)

ughout computer science and even daily life. In computer science, operations such as searching a collection of data can proceed exponentially faster if the collection is structured and indexed intelligently. People can easily perform arithmetic on Arabic numerals, but find arithmetic on Roman numerals much more time-consuming. It is not surprising that the choice of representation has an enormous eect on the performance of machine learning algorithms. For a simple visual example, see figure . 1.1 Many artificial intelligence tasks can be solved by designing the right set of features to extract for that task, then providing these features to a simple machine learning algorithm. For example, a useful feature for speaker identification from sound is an estimate of the size of speaker's vocal tract. It therefore gives a strong clue as to whether the speaker is a man, woman, or child. However, for many tasks, it is dicult to know what features should be extracted. For example, suppose that we would like to write a program to detect cars in photographs. We know that cars have wheels, so we might like to use the presence of a wheel as a feature. Unfortunately, it is dicult to describe exactly what a wheel looks like in terms of pixel values. A wheel has a simple geometric shape but its image may be complicated by shadows falling on the wheel, the sun glaring o the metal parts of the wheel, the fender of the car or an object in the foreground obscuring part of the wheel, and so on. 3 CHAPTER 1. INTRODUCTION introduction of machine learning allowed computers to tackle problems involving knowledge of the real world and make decisions that appear subjective. A simple machine learning algorithm called logistic regression can determine whether to recommend cesarean delivery (Mor-Yosef 1990 et al.,). A simple machine learning algorithm called naive Bayes can separate legitimate e-mail from spam e-mail. The performance of these simple machine learning algorithms depends heavily on the representation of the data they are given. For example, when logistic regression is used to recommend cesarean delivery, the AI system does not examine the patient directly. Instead, the doctor tells the system several pieces of relevant information, such as the presence or absence of a uterine scar. Each piece of information included in the representation of the patient is known as a feature. Logistic regression learns how each of these features of the patient correlates with various outcomes. However, it cannot influence the way that the features are defined in any way. If logistic regression was given an MRI scan of the patient, rather than the doctor's formalized report, it would not be able to make useful predictions. Individual pixels in an MRI scan have negligible correlation with any complications that might occur during delivery. This dependence on representations is a general phenomenon that appears throughout computer science and even daily life.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 18)

(score=0.58)

Segment 4: Task Can Complex Features Human

Lecture segment: tasks, with minimal human intervention. A representation learning algorithm can discover a good set of features for a simple task in minutes, or a complex task in hours to months. Manually designing features for a complex task requires a great deal of human time and effort; it can take decades for an entire community of researchers.

Below are closely-matching textbook excerpts supporting this segment.

reat deal of human time and eort; it can take decades for an entire community of researchers. The quintessential example of a representation learning algorithm is the autoencoder. An autoencoder is the combination of an encoder function that converts the input data into a dierent representation, and a decoder function that converts the new representation back into the original format. Autoencoders are trained to preserve as much information as possible when an input is run through the encoder and then the decoder, but are also trained to make the new representation have various nice properties. Dierent kinds of autoencoders aim to achieve dierent kinds of properties. When designing features or algorithms for learning features, our goal is usually to separate the factors of variation that explain the observed data. In this context, we use the word "factors" simply to refer to separate sources of influence; the factors are usually not combined by multiplication. Such factors are often not 4

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 19)

(score=0.68)

ughout computer science and even daily life. In computer science, operations such as searching a collection of data can proceed exponentially faster if the collection is structured and indexed intelligently. People can easily perform arithmetic on Arabic numerals, but find arithmetic on Roman numerals much more time-consuming. It is not surprising that the choice of representation has an enormous eect on the performance of machine learning algorithms. For a simple visual example, see figure . 1.1 Many artificial intelligence tasks can be solved by designing the right set of features to extract for that task, then providing these features to a simple machine learning algorithm. For example, a useful feature for speaker identification from sound is an estimate of the size of speaker's vocal tract. It therefore gives a strong clue as to whether the speaker is a man, woman, or child. However, for many tasks, it is dicult to know what features should be extracted. For example, suppose that we would like to write a program to detect cars in photographs. We know that cars have wheels, so we might like to use the presence of a wheel as a feature. Unfortunately, it is dicult to describe exactly what a wheel looks like in terms of pixel values. A wheel has a simple geometric shape but its image may be complicated by shadows falling on the wheel, the sun glaring o the metal parts of the wheel, the fender of the car or an object in the foreground obscuring part of the wheel, and so on. 3

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 18)

(score=0.62)

CHAPTER 1. INTRODUCTION

Figure 1.1: Example of dierent representations: suppose we want to
separate two categories of data by drawing a line between them in a scatterplot. In the plot on the left, we represent some data using Cartesian coordinates, and the task is impossible. In the plot on the right, we represent the data with polar coordinates and the task becomes simple to solve with a vertical line. Figure produced in collaboration with David Warde-Farley. One solution to this problem is to use machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as representation learning. Learned representations often result in much better performance than can be obtained with hand-designed representations. They also allow AI systems to rapidly adapt to new tasks, with minimal human intervention. A representation learning algorithm can discover a good set of features for a simple task in minutes, or a complex task in hours to months. Manually designing features for a complex task requires a great deal of human time and eort; it can take decades for an entire community of researchers. The quintessential example of a representation learning algorithm is the autoencoder.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 19)

(score=0.61)

CHAPTER 1. INTRODUCTION introduction of machine learning allowed computers to tackle problems involving knowledge of the real world and make decisions that appear subjective. A simple machine learning algorithm called logistic regression can determine whether to recommend cesarean delivery (Mor-Yosef 1990 et al.,). A simple machine learning algorithm called naive Bayes can separate legitimate e-mail from spam e-mail. The performance of these simple machine learning algorithms depends heavily on the representation of the data they are given. For example, when logistic regression is used to recommend cesarean delivery, the AI system does not examine the patient directly. Instead, the doctor tells the system several pieces of relevant information, such as the presence or absence of a uterine scar. Each piece of information included in the representation of the patient is known as a feature. Logistic regression learns how each of these features of the patient correlates with various outcomes. However, it cannot influence the way that the features are defined in any way. If logistic regression was given an MRI scan of the patient, rather than the doctor's formalized report, it would not be able to make useful predictions. Individual pixels in an MRI scan have negligible correlation with any complications that might occur during delivery. This dependence on representations is a general phenomenon that appears throughout computer science and even daily life.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 18)

(score=0.55)

the data. When it is nearly as dicult to obtain a representation as to solve the original problem, representation learning does not, at first glance, seem to help us. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts. Figure shows how a deep learning system can represent the concept of 1.2 an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges. The quintessential example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a dierent mathematical function as providing a new representation of the input. The idea of learning the right representation for the data provides one perspective on deep learning. Another perspective on deep learning is that depth allows the computer to learn a multi-step computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Sequential instructions oer great power because later 5

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 20)

(score=0.55)

Segment 5: Learning Deep Different Kinds Representation

Lecture segment: for an entire community of researchers. The quintessential example of a representation learning algorithm is ... Different kinds of autoencoders aim to achieve different kinds of properties. Deep learning addresses this central problem in representation learning by introducing representations that are expressed in terms of simpler, more fundamental representations. Deep learning enables the computer to construct complex concepts from simpler ones.

Below are closely-matching textbook excerpts supporting this segment.

reat deal of human time and eort; it can take decades for an entire community of researchers. The quintessential example of a representation learning algorithm is the autoencoder. An autoencoder is the combination of an encoder function that converts the input data into a dierent representation, and a decoder function that converts the new representation back into the original format. Autoencoders are trained to preserve as much information as possible when an input is run through the encoder and then the decoder, but are also trained to make the new representation have various nice properties. Dierent kinds of autoencoders aim to achieve dierent kinds of properties.

When designing features or algorithms for learning features, our goal is usually to separate the factors of variation that explain the observed data. In this context, we use the word "factors" simply to refer to separate sources of influence; the factors are usually not combined by multiplication. Such factors are often not 4 CHAPTER 1. INTRODUCTION

Figure 1.1: Example of

dierent representations: suppose we want to separate two categories of data by drawing a line between them in a scatterplot. In the plot on the left, we represent some data using Cartesian coordinates, and the task is impossible. In the plot on the right, we represent the data with polar coordinates and the task becomes simple to solve with a vertical line. Figure produced in collaboration with David Warde-Farley. One solution to this problem is to use machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as representation learning. Learned representations often result in much better performance than can be obtained with hand-designed representations. They also allow AI systems to rapidly adapt to new tasks, with minimal human intervention. A representation learning algorithm can discover a good set of features for a simple task in minutes, or a complex task in hours to months. Manually designing features for a complex task requires a great deal of human time and eort; it can take decades for an entire community of researchers. The quintessential example of a representation learning algorithm is the autoencoder.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 19)

(score=0.80)

the data. When it is nearly as dicult to obtain a representation as to solve the original problem, representation learning does not, at first glance, seem to help us. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts. Figure shows how a deep learning system can represent the concept of 1.2 an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges. The quintessential example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a dierent mathematical function as providing a new representation of the input. The idea of learning the right representation for the data provides one perspective on deep learning. Another perspective on deep learning is that depth allows the computer to learn a multi-step computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Sequential instructions oer great power because later 5

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 20)

(score=0.71)

CHAPTER 1. INTRODUCTION AI Machine learning Representation learning Deep learning Example: Knowledge bases Example: Logistic regression Example: Shallow autoencoders Example: MLPs Figure 1.4: A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. Each section of the Venn diagram includes an example of an AI technology. 9

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 24)

(score=0.70)

Segment 6: Concepts Simpler Can Combining Complex

Lecture segment: ruct complex concepts from simpler ones. Figure 1.2 illustrates how a deep learning system can represent the concept of an image of a person by combining simpler concepts, such as corners and contours, which are defined in terms of edges.

Below are closely-matching textbook excerpts supporting this segment.

the data. When it is nearly as dicult to obtain a representation as to solve the original problem, representation learning does not, at first glance, seem to help us. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts. Figure shows how a deep learning system can represent the concept of 1.2 an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges. The quintessential example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a dierent mathematical function as providing a new representation of the input. The idea of learning the right representation for the data provides one perspective on deep learning. Another perspective on deep learning is that depth allows the computer to learn a multi-step computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Sequential instructions oer great power because later 5

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 20)

(score=0.69)

CHAPTER 1. INTRODUCTION Visible layer (input pixels) 1st hidden layer (edges) 2nd hidden layer (corners and contours) 3rd hidden layer (object parts) CAR PERSON ANIMAL Output (object identity) Figure 1.2: Illustration of a deep learning model. It is dicult for a computer to understand the meaning of raw sensory input data, such as this image represented as a collection of pixel values. The function mapping from a set of pixels to an object identity is very complicated. Learning or evaluating this mapping seems insurmountable if tackled directly. Deep learning resolves this diculty by breaking the desired complicated mapping into a series of nested simple mappings, each described by a dierent layer of the model. The input is presented at the visible layer, so named because it contains the variables that we are able to observe. Then a series of hidden layers extracts increasingly abstract features from the image. These layers are called "hidden" because their values are not given in the data; instead the model must determine which concepts are useful for explaining the relationships in the observed data. The images here are visualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 21)

(score=0.66)

CHAPTER 1. INTRODUCTION of the flowchart of the computations needed to compute the representation of each concept may be much deeper than the graph of the concepts themselves. This is because the system's understanding of the simpler concepts can be refined given information about the more complex concepts. For example, an AI system observing an image of a face with one eye in shadow may initially only see one eye. After detecting that a face is present, it can then infer that a second eye is probably present as well. In this case, the graph of concepts only includes two layers-a layer for eyes and a layer for faces-but the graph of computations includes 2n layers if we refine our estimate of each concept given the other times. n Because it is not always clear which of these two views-the depth of the computational graph, or the depth of the probabilistic modeling graph-is most relevant, and because dierent people choose dierent sets of smallest elements from which to construct their graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program. Nor is there a consensus about how much depth a model requires to qualify as "deep." However, deep learning can safely be regarded as the study of models that either involve a greater amount of composition of learned functions or learned concepts than traditional machine learning does.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 23)

(score=0.65)

isualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels. Given the first hidden layer's description of the edges, the second hidden layer can easily search for corners and extended contours, which are recognizable as collections of edges. Given the second hidden layer's description of the image in terms of corners and contours, the third hidden layer can detect entire parts of specific objects, by finding specific collections of contours and corners. Finally, this description of the image in terms of the object parts it contains can be used to recognize the objects present in the image. Images reproduced with permission from Zeiler and Fergus 2014 ( ). 6

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 21)

(score=0.60)

CHAPTER 1. INTRODUCTION AI Machine learning Representation learning Deep learning Example: Knowledge bases Example: Logistic regression Example: Shallow autoencoders Example: MLPs Figure 1.4: A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. Each section of the Venn diagram includes an example of an AI technology. 9

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 24)

(score=0.57)

Segment 7: Appears Assess Attempting Challenging Collection

Lecture segment: rs, which are defined in terms of edges. It is challenging for a computer to comprehend the significance of unprocessed sensory input data, such as an image depicted as a set of pixel values. The function that maps a collection of pixels to an object identity is highly intricate. Attempting to learn or assess this mapping directly appears to be an overwhelming task.

[No relevant excerpt found in provided references]

Segment 8: Layer Abstract Addresses Appears Aptly

Lecture segment: ctly appears to be an overwhelming task. Deep learning addresses this challenge by deconstructing the complex desired mapping into a sequence of simpler nested mappings, each represented by a distinct layer of the model. The input is introduced at the visible layer, aptly named because it encompasses the variables that we can observe. Subsequently, a succession of hidden layers extracts progressively abstract features from the image.

Below are closely-matching textbook excerpts supporting this segment.

CHAPTER 1. INTRODUCTION Visible layer (input pixels) 1st hidden layer (edges) 2nd hidden layer (corners and contours) 3rd hidden layer (object parts) CAR PERSON ANIMAL Output (object identity) Figure 1.2: Illustration of a deep learning model. It is dicult for a computer to understand the meaning of raw sensory input data, such as this image represented as a collection of pixel values. The function mapping from a set of pixels to an object identity is very complicated. Learning or evaluating this mapping seems insurmountable if tackled directly. Deep learning resolves this diculty by breaking the desired complicated mapping into a series of nested simple mappings, each described by a dierent layer of the model. The input is presented at the visible layer, so named because it contains the variables that we are able to observe. Then a series of hidden layers extracts increasingly abstract features from the image. These layers are called "hidden" because their values are not given in the data; instead the model must determine which concepts are useful for explaining the relationships in the observed data. The images here are visualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 21)

(score=0.77)

the data. When it is nearly as dicult to obtain a representation as to solve the original problem, representation learning does not, at first glance, seem to help us. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts. Figure shows how a deep learning system can represent the concept of 1.2 an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges. The quintessential example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a dierent mathematical function as providing a new representation of the input. The idea of learning the right representation for the data provides one perspective on deep learning. Another perspective on deep learning is that depth allows the computer to learn a multi-step computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Sequential instructions oer great power because later 5

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 20)

(score=0.71)

Dive into Deep Learning, Release 0.7 directly from the pixels in images, and mapping between sentences of arbitrary lengths and across languages are problems where deep learning excels and traditional ML tools faltered. Deep models are deep in precisely the sense that they learn many layers of computation. It turns out that these many-layered (or hierarchical) models are capable of addressing low-level perceptual data in a way that previous tools could not. In bygone days, the crucial part of applying ML to these problems consisted of coming up with manually engineered ways of transforming the data into some form amenable to shallow models. One key advantage of deep learning is that it replaces not only the shallow models at the end of traditional learning pipelines, but also the labor-intensive feature engineering. Secondly, by replacing much of the domain-specific preprocessing, deep learning has eliminated many of the boundaries that previously separated computer vision, speech recognition, natural language processing, medical informatics, and other application areas, offering a unified set of tools for tackling diverse problems. 3.2 The Key Components: Data, Models, and Algorithms In our wake-word example, we described a dataset consisting of audio snippets and binary labels gave a hand-wavy sense of how we might train a model to approximate a mapping from snippets to classifications.

(Source: Dive into Deep Learning_Zhang_Lipton_Li_Smola_Part1.pdf, p. 20)

(score=0.66)

isualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels. Given the first hidden layer's description of the edges, the second hidden layer can easily search for corners and extended contours, which are recognizable as collections of edges. Given the second hidden layer's description of the image in terms of corners and contours, the third hidden layer can detect entire parts of specific objects, by finding specific collections of contours and corners. Finally, this description of the image in terms of the object parts it contains can be used to recognize the objects present in the image. Images reproduced with permission from Zeiler and Fergus 2014 ( ). 6

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 21)

(score=0.64)

CHAPTER 1. INTRODUCTION of the flowchart of the computations needed to compute the representation of each concept may be much deeper than the graph of the concepts themselves. This is because the system's understanding of the simpler concepts can be refined given information about the more complex concepts. For example, an AI system observing an image of a face with one eye in shadow may initially only see one eye. After detecting that a face is present, it can then infer that a second eye is probably present as well. In this case, the graph of concepts only includes two layers-a layer for eyes and a layer for faces-but the graph of computations includes 2n layers if we refine our estimate of each concept given the other times. n Because it is not always clear which of these two views-the depth of the computational graph, or the depth of the probabilistic modeling graph-is most relevant, and because dierent people choose dierent sets of smallest elements from which to construct their graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program. Nor is there a consensus about how much depth a model requires to qualify as "deep." However, deep learning can safely be regarded as the study of models that either involve a greater amount of composition of learned functions or learned concepts than traditional machine learning does.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 23)

(score=0.63)

Segment 9: Data Features Hidden Abstract Adjacent

Lecture segment: sively abstract features from the image. These layers are termed ‘hidden' as their values are not provided in the data; rather, the model must ascertain which concepts are pertinent for elucidating the relationships within the observed data. The images presented here are visual representations of the types of features associated with each hidden unit. Given the pixel data, the first layer can readily identify edges by comparing the brightness of adjacent pixels.

Below are closely-matching textbook excerpts supporting this segment.

isualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels. Given the first hidden layer's description of the edges, the second hidden layer can easily search for corners and extended contours, which are recognizable as collections of edges. Given the second hidden layer's description of the image in terms of corners and contours, the third hidden layer can detect entire parts of specific objects, by finding specific collections of contours and corners. Finally, this description of the image in terms of the object parts it contains can be used to recognize the objects present in the image. Images reproduced with permission from Zeiler and Fergus 2014 ( ). 6 CHAPTER 1. INTRODUCTION Visible layer (input pixels) 1st hidden layer (edges) 2nd hidden layer (corners and contours) 3rd hidden layer (object parts) CAR PERSON ANIMAL Output (object identity) Figure 1.2: Illustration of a deep learning model. It is dicult for a computer to understand the meaning of raw sensory input data, such as this image represented as a collection of pixel values. The function mapping from a set of pixels to an object identity is very complicated. Learning or evaluating this mapping seems insurmountable if tackled directly. Deep learning resolves this diculty by breaking the desired complicated mapping into a series of nested simple mappings, each described by a dierent layer of the model. The input is presented at the visible layer, so named because it contains the variables that we are able to observe. Then a series of hidden layers extracts increasingly abstract features from the image. These layers are called "hidden" because their values are not given in the data; instead the model must determine which concepts are useful for explaining the relationships in the observed data. The images here are visualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 21)

(score=0.85)

Segment 10: Hidden Layer Can Contours Corners

Lecture segment: aring the brightness of adjacent pixels. Utilizing the first hidden layer's depiction of edges, the second hidden layer can efficiently search for corners and extended contours, which are identifiable as groups of edges. Based on the second hidden layer's interpretation of the image in terms of corners and contours, the third hidden layer can recognize entire segments of specific objects by locating particular collections of

Below are closely-matching textbook excerpts supporting this segment.

isualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels. Given the first hidden layer's description of the edges, the second hidden layer can easily search for corners and extended contours, which are recognizable as collections of edges. Given the second hidden layer's description of the image in terms of corners and contours, the third hidden layer can detect entire parts of specific objects, by finding specific collections of contours and corners. Finally, this description of the image in terms of the object parts it contains can be used to recognize the objects present in the image. Images reproduced with permission from Zeiler and Fergus 2014 ( ). 6 CHAPTER 1. INTRODUCTION Visible layer (input pixels) 1st hidden layer (edges) 2nd hidden layer (corners and contours) 3rd hidden layer (object parts) CAR PERSON ANIMAL Output (object identity) Figure 1.2: Illustration of a deep learning model. It is dicult for a computer to understand the meaning of raw sensory input data, such as this image represented as a collection of pixel values. The function mapping from a set of pixels to an object identity is very complicated. Learning or evaluating this mapping seems insurmountable if tackled directly. Deep learning resolves this diculty by breaking the desired complicated mapping into a series of nested simple mappings, each described by a dierent layer of the model. The input is presented at the visible layer, so named because it contains the variables that we are able to observe. Then a series of hidden layers extracts increasingly abstract features from the image. These layers are called "hidden" because their values are not given in the data; instead the model must determine which concepts are useful for explaining the relationships in the observed data. The images here are visualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 21)

(score=0.77)

Segment 11: Image Can Characterization Collections Contours

Lecture segment: ts by locating particular collections of contours and corners. Ultimately, this characterization of the image in relation to the object parts it encompasses can facilitate the recognition of the objects present within the image.

[No relevant excerpt found in provided references]

Segment 12: Depth Logistic Model Has Output

Lecture segment: mage. Depth is defined as the longest path's length from input to output. The computation depicted in these graphs is the output of a logistic regression model, o(wT x), where o is the logistic sigmoid function. If we use addition, multiplication and logistic sigmoids as the elements of our computer language, then this model has depth three. If we view logistic regression as an element itself, then this model has depth one. Two main ways of measuring the depth of a model are

Below are closely-matching textbook excerpts supporting this segment.

sing. There are two main ways of measuring the depth of a model. The first view is based on the number of sequential instructions that must be executed to evaluate the architecture. We can think of this as the length of the longest path through a flow chart that describes how to compute each of the model's outputs given its inputs. Just as two equivalent computer programs will have dierent lengths depending on which language the program is written in, the same function may be drawn as a flowchart with dierent depths depending on which functions we allow to be used as individual steps in the flowchart. Figure illustrates how this 1.3 choice of language can give two dierent measurements for the same architecture. Another approach, used by deep probabilistic models, regards the depth of a model as being not the depth of the computational graph but the depth of the graph describing how concepts are related to each other. In this case, the depth 7 CHAPTER 1.

Can Learning Algorithm But Complex

INTRODUCTION x1 x1 σ w1 w1 × x2 x2 w2 w2 × + Element Set + × σ x w Element Set Logistic Regression Logistic Regression Figure 1.3: Illustration of computational graphs mapping an input to an output where each node performs an operation. Depth is the length of the longest path from input to output but depends on the definition of what constitutes a possible computational step. The computation depicted in these graphs is the output of a logistic regression model, σ(wT x), where σ is the logistic sigmoid function. If we use addition, multiplication and logistic sigmoids as the elements of our computer language, then this model has depth three. If we view logistic regression as an element itself, then this model has depth one. instructions can refer back to the results of earlier instructions. According to this view of deep learning, not all of the information in a layer's activations necessarily encodes factors of variation that explain the input. The representation also stores state information that helps to execute a program that can make sense of the input. This state information could be analogous to a counter or pointer in a traditional computer program. It has nothing to do with the content of the input specifically, but it helps the model to organize its processing. There are two main ways of measuring the depth of a model. The first view is based on the number of sequential instructions that must be executed to evaluate the architecture.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 22)

(score=0.81)

CHAPTER 1. INTRODUCTION of the flowchart of the computations needed to compute the representation of each concept may be much deeper than the graph of the concepts themselves. This is because the system's understanding of the simpler concepts can be refined given information about the more complex concepts. For example, an AI system observing an image of a face with one eye in shadow may initially only see one eye. After detecting that a face is present, it can then infer that a second eye is probably present as well. In this case, the graph of concepts only includes two layers-a layer for eyes and a layer for faces-but the graph of computations includes 2n layers if we refine our estimate of each concept given the other times. n Because it is not always clear which of these two views-the depth of the computational graph, or the depth of the probabilistic modeling graph-is most relevant, and because dierent people choose dierent sets of smallest elements from which to construct their graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program. Nor is there a consensus about how much depth a model requires to qualify as "deep." However, deep learning can safely be regarded as the study of models that either involve a greater amount of composition of learned functions or learned concepts than traditional machine learning does.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 23)

(score=0.65)

Segment 13: Depth Concepts Each Graph How

Lecture segment: ys of measuring the depth of a model are 1) using the number of sequential instructions. 2) depth of the graph saying how concepts are linked to each other

Can Learning Algorithm But Complex

Below are closely-matching textbook excerpts supporting this segment.

sing. There are two main ways of measuring the depth of a model. The first view is based on the number of sequential instructions that must be executed to evaluate the architecture. We can think of this as the length of the longest path through a flow chart that describes how to compute each of the model's outputs given its inputs. Just as two equivalent computer programs will have dierent lengths depending on which language the program is written in, the same function may be drawn as a flowchart with dierent depths depending on which functions we allow to be used as individual steps in the flowchart. Figure illustrates how this 1.3 choice of language can give two dierent measurements for the same architecture. Another approach, used by deep probabilistic models, regards the depth of a model as being not the depth of the computational graph but the depth of the graph describing how concepts are related to each other. In this case, the depth 7 CHAPTER 1. INTRODUCTION x1 x1 σ w1 w1 × x2 x2 w2 w2 × + Element Set + × σ x w Element Set Logistic Regression Logistic Regression Figure 1.3: Illustration of computational graphs mapping an input to an output where each node performs an operation. Depth is the length of the longest path from input to output but depends on the definition of what constitutes a possible computational step. The computation depicted in these graphs is the output of a logistic regression model, σ(wT x), where σ is the logistic sigmoid function. If we use addition, multiplication and logistic sigmoids as the elements of our computer language, then this model has depth three. If we view logistic regression as an element itself, then this model has depth one. instructions can refer back to the results of earlier instructions. According to this view of deep learning, not all of the information in a layer's activations necessarily encodes factors of variation that explain the input. The representation also stores state information that helps to execute a program that can make sense of the input. This state information could be analogous to a counter or pointer in a traditional computer program. It has nothing to do with the content of the input specifically, but it helps the model to organize its processing. There are two main ways of measuring the depth of a model. The first view is based on the number of sequential instructions that must be executed to evaluate the architecture.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 22)

(score=0.63)

Segment 14: Depth Because Correct Different Single

Lecture segment: ng how concepts are linked to each other Because it is not always clear which of these two views-the depth of the computational graph, or the depth of the probabilistic modeling graph-is most relevant, and because different people choose different sets of smallest elements from which to construct their graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program.

Below are closely-matching textbook excerpts supporting this segment.

CHAPTER 1. INTRODUCTION of the flowchart of the computations needed to compute the representation of each concept may be much deeper than the graph of the concepts themselves.

Can Learning Algorithm But Complex

This is because the system's understanding of the simpler concepts can be refined given information about the more complex concepts. For example, an AI system observing an image of a face with one eye in shadow may initially only see one eye. After detecting that a face is present, it can then infer that a second eye is probably present as well. In this case, the graph of concepts only includes two layers-a layer for eyes and a layer for faces-but the graph of computations includes 2n layers if we refine our estimate of each concept given the other times. n Because it is not always clear which of these two views-the depth of the computational graph, or the depth of the probabilistic modeling graph-is most relevant, and because dierent people choose dierent sets of smallest elements from which to construct their graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program. Nor is there a consensus about how much depth a model requires to qualify as "deep." However, deep learning can safely be regarded as the study of models that either involve a greater amount of composition of learned functions or learned concepts than traditional machine learning does.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 23)

(score=0.77)

sing. There are two main ways of measuring the depth of a model. The first view is based on the number of sequential instructions that must be executed to evaluate the architecture. We can think of this as the length of the longest path through a flow chart that describes how to compute each of the model's outputs given its inputs. Just as two equivalent computer programs will have dierent lengths depending on which language the program is written in, the same function may be drawn as a flowchart with dierent depths depending on which functions we allow to be used as individual steps in the flowchart. Figure illustrates how this 1.3 choice of language can give two dierent measurements for the same architecture. Another approach, used by deep probabilistic models, regards the depth of a model as being not the depth of the computational graph but the depth of the graph describing how concepts are related to each other. In this case, the depth 7 CHAPTER 1. INTRODUCTION x1 x1 σ w1 w1 × x2 x2 w2 w2 × + Element Set + × σ x w Element Set Logistic Regression Logistic Regression Figure 1.3: Illustration of computational graphs mapping an input to an output where each node performs an operation. Depth is the length of the longest path from input to output but depends on the definition of what constitutes a possible computational step. The computation depicted in these graphs is the output of a logistic regression model, σ(wT x), where σ is the logistic sigmoid function. If we use addition, multiplication and logistic sigmoids as the elements of our computer language, then this model has depth three. If we view logistic regression as an element itself, then this model has depth one. instructions can refer back to the results of earlier instructions. According to this view of deep learning, not all of the information in a layer's activations necessarily encodes factors of variation that explain the input. The representation also stores state information that helps to execute a program that can make sense of the input. This state information could be analogous to a counter or pointer in a traditional computer program. It has nothing to do with the content of the input specifically, but it helps the model to organize its processing. There are two main ways of measuring the depth of a model. The first view is based on the number of sequential instructions that must be executed to evaluate the architecture.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 22)

(score=0.75)

the data. When it is nearly as dicult to obtain a representation as to solve the original problem, representation learning does not, at first glance, seem to help us. Deep learning solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts. Figure shows how a deep learning system can represent the concept of 1.2 an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges. The quintessential example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP). A multilayer perceptron is just a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a dierent mathematical function as providing a new representation of the input. The idea of learning the right representation for the data provides one perspective on deep learning. Another perspective on deep learning is that depth allows the computer to learn a multi-step computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Sequential instructions oer great power because later 5

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 20)

(score=0.55)

Segment 15: Deep Learned Learning Amount Book

Lecture segment: ue for the length of a computer program. Nor is there a consensus about how much depth a model requires to qualify as "deep." However, deep learning can safely be regarded as the study of models that either involve a greater amount of composition of learned functions or learned concepts than traditional machine learning does. 1.1 Who Should Read This Book?

Below are closely-matching textbook excerpts supporting this segment.

amount of composition of learned functions or learned concepts than traditional machine learning does. To summarize, deep learning, the subject of this book, is an approach to AI. Specifically, it is a type of machine learning, a technique that allows computer systems to improve with experience and data. According to the authors of this book, machine learning is the only viable approach to building AI systems that can operate in complicated, real-world environments. Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. Figure illustrates the relationship between these dierent 1.4 AI disciplines. Figure gives a high-level schematic of how each works. 1.5 1.1 Who Should Read This Book? This book can be useful for a variety of readers, but we wrote it with two main target audiences in mind. One of these target audiences is university students (undergraduate or graduate) learning about machine learning, including those who are beginning a career in deep learning and artificial intelligence research. The other target audience is software engineers who do not have a machine learning or statistics background, but want to rapidly acquire one and begin using deep learning in their product or platform. Deep learning has already proven useful in 8 CHAPTER 1. INTRODUCTION of the flowchart of the computations needed to compute the representation of each concept may be much deeper than the graph of the concepts themselves. This is because the system's understanding of the simpler concepts can be refined given information about the more complex concepts. For example, an AI system observing an image of a face with one eye in shadow may initially only see one eye. After detecting that a face is present, it can then infer that a second eye is probably present as well. In this case, the graph of concepts only includes two layers-a layer for eyes and a layer for faces-but the graph of computations includes 2n layers if we refine our estimate of each concept given the other times. n Because it is not always clear which of these two views-the depth of the computational graph, or the depth of the probabilistic modeling graph-is most relevant, and because dierent people choose dierent sets of smallest elements from which to construct their graphs, there is no single correct value for the depth of an architecture, just as there is no single correct value for the length of a computer program. Nor is there a consensus about how much depth a model requires to qualify as "deep." However, deep learning can safely be regarded as the study of models that either involve a greater amount of composition of learned functions or learned concepts than traditional machine learning does.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 23)

(score=0.75)

CHAPTER 1. INTRODUCTION many software disciplines including computer vision, speech and audio processing, natural language processing, robotics, bioinformatics and chemistry, video games, search engines, online advertising and finance. This book has been organized into three parts in order to best accommodate a variety of readers. Part introduces basic mathematical tools and machine learning I concepts. Part describes the most established deep learning algorithms that are II essentially solved technologies. Part describes more speculative ideas that are III widely believed to be important for future research in deep learning. Readers should feel free to skip parts that are not relevant given their interests or background. Readers familiar with linear algebra, probability, and fundamental machine learning concepts can skip part , for example, while readers who just want I to implement a working system need not read beyond part . To help choose which II chapters to read, figure provides a flowchart showing the high-level organization 1.6 of the book. We do assume that all readers come from a computer science background. We assume familiarity with programming, a basic understanding of computational performance issues, complexity theory, introductory level calculus and some of the terminology of graph theory. 1.2 Historical Trends in Deep Learning It is easiest to understand deep learning with some historical context.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 26)

(score=0.68)

ne replies, helping people dig out from mountains of email, and software agents that dominate the world's best humans at board games like Go, a feat once deemed to be decades away. Already, these tools are exerting a widening impact, changing the way movies are made, diseases are diagnosed, and playing a growing role in basic sciences - from astrophysics to biology. This book represents our attempt to make deep learning approachable, teaching you both the concepts, the context, and the code. 1.1 About This Book 1.1.1 One Medium Combining Code, Math, and HTML For any computing technology to reach its full impact, it must be well-understood, well-documented, and supported by mature, well-maintained tools. The key ideas should be clearly distilled, minimizing the onboarding time needed to bring new practitioners up to date. Mature libraries should automate common tasks, and exemplar code should make it easy for practitioners to modify, apply, and extend common applications to suit their needs. Take dynamic web applications as an example. Despite a large number of companies, like Amazon, developing successful database-driven web applications in the 1990s, the full potential of this technology to aid creative entrepreneurs has only been realized over the past ten years, owing to the development of powerful, well-documented frameworks. Realizing deep learning presents unique challenges because any single application brings together various disciplines.

(Source: Dive into Deep Learning_Zhang_Lipton_Li_Smola_Part1.pdf, p. 7)

(score=0.67)

17.7. 1.1.5 Target Audience This book is for students (undergraduate or graduate), engineers, and researchers, who seek a solid grasp of the practical techniques of deep learning. Because we explain every concept from scratch, no previous background in deep learning or machine learning is required. Fully explaining the methods of deep learning 4 Chapter 1. Preface

(Source: Dive into Deep Learning_Zhang_Lipton_Li_Smola_Part1.pdf, p. 10)

(score=0.66)

CHAPTER 1. INTRODUCTION that neuroscientists can study (,). Deep learning also provides useful DiCarlo 2013 tools for processing massive amounts of data and making useful predictions in scientific fields. It has been successfully used to predict how molecules will interact in order to help pharmaceutical companies design new drugs (,), Dahl et al. 2014 to search for subatomic particles (,), and to automatically parse Baldi et al. 2014 microscope images used to construct a 3-D map of the human brain (KnowlesBarley 2014 et al.,). We expect deep learning to appear in more and more scientific fields in the future. In summary, deep learning is an approach to machine learning that has drawn heavily on our knowledge of the human brain, statistics and applied math as it developed over the past several decades. In recent years, it has seen tremendous growth in its popularity and usefulness, due in large part to more powerful computers, larger datasets and techniques to train deeper networks. The years ahead are full of challenges and opportunities to improve deep learning even further and bring it to new frontiers. 26

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 41)

Can Learning Algorithm But Complex

(score=0.63)

Segment 16: Learning Target Who Audiences Book

Lecture segment: ng does. 1.1 Who Should Read This Book? This book can be useful for a variety of readers, but we wrote it with two main target audiences in mind. One of these target audiences is university students (undergraduate or graduate) learning about machine learning, including those who are beginning a career in deep learning and artificial intelligence research. The other target audience is software engineers who do not have a machine learning

Below are closely-matching textbook excerpts supporting this segment.

17.7. 1.1.5 Target Audience This book is for students (undergraduate or graduate), engineers, and researchers, who seek a solid grasp of the practical techniques of deep learning. Because we explain every concept from scratch, no previous background in deep learning or machine learning is required. Fully explaining the methods of deep learning 4 Chapter 1. Preface

(Source: Dive into Deep Learning_Zhang_Lipton_Li_Smola_Part1.pdf, p. 10)

(score=0.75)

amount of composition of learned functions or learned concepts than traditional machine learning does. To summarize, deep learning, the subject of this book, is an approach to AI. Specifically, it is a type of machine learning, a technique that allows computer systems to improve with experience and data. According to the authors of this book, machine learning is the only viable approach to building AI systems that can operate in complicated, real-world environments. Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. Figure illustrates the relationship between these dierent 1.4 AI disciplines. Figure gives a high-level schematic of how each works. 1.5 1.1 Who Should Read This Book? This book can be useful for a variety of readers, but we wrote it with two main target audiences in mind. One of these target audiences is university students (undergraduate or graduate) learning about machine learning, including those who are beginning a career in deep learning and artificial intelligence research. The other target audience is software engineers who do not have a machine learning or statistics background, but want to rapidly acquire one and begin using deep learning in their product or platform. Deep learning has already proven useful in 8

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 23)

(score=0.72)

this book is based on Apache MXNet. MXNet is an open-source framework for deep learning and the preferred choice of AWS (Amazon Web Services), as well as many colleges and companies. All of the code in this book has passed tests under the newest MXNet version. However, due to the rapid development of deep learning, some code in the print edition may not work properly in future versions of MXNet. However, we plan to keep the online version up-to-date. In case of such problems, please consult Installation (page 7) to update the code and runtime environment. At times, to avoid unnecessary repetition, we encapsulate the frequently-imported and referred-to functions, classes, etc. in this book in the d2l package. For any block such as a function, a class, or multiple imports to be saved in the package, we will mark it with # Save to the d2l package. For example, these are the packages and modules used by the d2l package. # Save to the d2l package from IPython import display import collections import os import sys import numpy as np import math from matplotlib import pyplot as plt from mxnet import nd, autograd, gluon, init, context, image from mxnet.gluon import nn, rnn import random import re import time import tarfile import zipfile We give a detailed overview of these functions and classes in Section 17.7. 1.1.5 Target Audience This book is for students (undergraduate or graduate), engineers, and researchers, who seek a solid grasp of the practical techniques of deep learning.

(Source: Dive into Deep Learning_Zhang_Lipton_Li_Smola_Part1.pdf, p. 10)

(score=0.67)

ne replies, helping people dig out from mountains of email, and software agents that dominate the world's best humans at board games like Go, a feat once deemed to be decades away. Already, these tools are exerting a widening impact, changing the way movies are made, diseases are diagnosed, and playing a growing role in basic sciences - from astrophysics to biology. This book represents our attempt to make deep learning approachable, teaching you both the concepts, the context, and the code. 1.1 About This Book 1.1.1 One Medium Combining Code, Math, and HTML For any computing technology to reach its full impact, it must be well-understood, well-documented, and supported by mature, well-maintained tools. The key ideas should be clearly distilled, minimizing the onboarding time needed to bring new practitioners up to date. Mature libraries should automate common tasks, and exemplar code should make it easy for practitioners to modify, apply, and extend common applications to suit their needs. Take dynamic web applications as an example. Despite a large number of companies, like Amazon, developing successful database-driven web applications in the 1990s, the full potential of this technology to aid creative entrepreneurs has only been realized over the past ten years, owing to the development of powerful, well-documented frameworks. Realizing deep learning presents unique challenges because any single application brings together various disciplines.

(Source: Dive into Deep Learning_Zhang_Lipton_Li_Smola_Part1.pdf, p. 7)

(score=0.62)

Preface Introduction With the ever increasing amounts of data in electronic form, the need for automated methods for data analysis continues to grow. The goal of machine learning is to develop methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data or other outcomes of interest. Machine learning is thus closely related to the fields of statistics and data mining, but differs slightly in terms of its emphasis and terminology. This book provides a detailed introduction to the field, and includes worked examples drawn from application domains such as molecular biology, text processing, computer vision, and robotics.

Can Learning Algorithm But Complex

Target audience This book is suitable for upper-level undergraduate students and beginning graduate students in computer science, statistics, electrical engineering, econometrics, or any one else who has the appropriate mathematical background. Specifically, the reader is assumed to already be familiar with basic multivariate calculus, probability, linear algebra, and computer programming. Prior exposure to statistics is helpful but not necessary. A probabilistic approach This books adopts the view that the best way to make machines that can learn from data is to use the tools of probability theory, which has been the mainstay of statistics and engineering for centuries. Probability theory can be applied to any problem involving uncertainty.

(Source: Machine Learning - A Probabilistic Perspective_Murphy_Part1.pdf, p. 28)

(score=0.62)

Segment 17: Learning Acquire Adaline Adapt Algorithm

Lecture segment: neers who do not have a machine learning or statistics background, but want to rapidly acquire one and begin using deep learning in their product or platform. The training algorithm used to adapt the weights of the ADALINE was a special case of ...? In the 1980s, the second wave of neural network research emerged in great part via a movement called ...? ... DeepMind demonstrated that a reinforcement learning system ...

Below are closely-matching textbook excerpts supporting this segment.

e ADALINE was a special case of an algorithm called stochastic gradient descent. Slightly modified versions of the stochastic gradient descent algorithm remain the dominant training algorithms for deep learning models today. Models based on the f(x w ,) used by the perceptron and ADALINE are called linear models. These models remain some of the most widely used machine learning models, though in many cases they are trained in dierent ways than the original models were trained. Linear models have many limitations. Most famously, they cannot learn the XOR function, where f ([0,1], w) = 1 and f([1, 0], w) = 1 but f([1, 1], w) = 0 and f([0, 0], w) = 0. Critics who observed these flaws in linear models caused a backlash against biologically inspired learning in general (Minsky and Papert, 1969). This was the first major dip in the popularity of neural networks. Today, neuroscience is regarded as an important source of inspiration for deep learning researchers, but it is no longer the predominant guide for the field. The main reason for the diminished role of neuroscience in deep learning research today is that we simply do not have enough information about the brain to use it as a guide. To obtain a deep understanding of the actual algorithms used by the brain, we would need to be able to monitor the activity of (at the very least) thousands of interconnected neurons simultaneously. Because we are not able to do this, we are far from understanding even some of the most simple and 15 CHAPTER 1. INTRODUCTION The earliest predecessors of modern deep learning were simple linear models motivated from a neuroscientific perspective. These models were designed to take a set of n input values x1, . . . , xn and associate them with an output y. These models would learn a set of weights w1, . . . , wn and compute their output f(x w ,) = x1w1 + · · · + xnwn. This first wave of neural networks research was known as

Can Learning Algorithm But Complex

cybernetics, as illustrated in figure . 1.7 The McCulloch-Pitts Neuron (,) was an early model McCulloch and Pitts 1943 of brain function. This linear model could recognize two dierent categories of inputs by testing whether f (x w ,) is positive or negative. Of course, for the model to correspond to the desired definition of the categories, the weights needed to be set correctly. These weights could be set by the human operator. In the 1950s, the perceptron (Rosenblatt 1958 1962 , ,) became the first model that could learn the weights defining the categories given examples of inputs from each category. The adaptive linear element (ADALINE), which dates from about the same time, simply returned the value of f(x) itself to predict a real number (Widrow and Ho1960 ,), and could also learn to predict these numbers from data. These simple learning algorithms greatly aected the modern landscape of machine learning. The training algorithm used to adapt the weights of the ADALINE was a special case of an algorithm called stochastic gradient descent.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 30)

(score=0.72)

h did not fulfill these unreasonable expectations, investors were disappointed. Simultaneously, other fields of machine learning made advances. Kernel machines (, Boser et al. 1992 Cortes and Vapnik 1995 Schölkopf 1999 Jor- ; , ; et al.,) and graphical models (dan 1998 ,) both achieved good results on many important tasks. These two factors led to a decline in the popularity of neural networks that lasted until 2007. During this time, neural networks continued to obtain impressive performance on some tasks (, ; ,). The Canadian Institute LeCun et al. 1998b Bengio et al. 2001 for Advanced Research (CIFAR) helped to keep neural networks research alive via its Neural Computation and Adaptive Perception (NCAP) research initiative. This program united machine learning research groups led by Georey Hinton at University of Toronto, Yoshua Bengio at University of Montreal, and Yann LeCun at New York University. The CIFAR NCAP research initiative had a multi-disciplinary nature that also included neuroscientists and experts in human and computer vision. At this point in time, deep networks were generally believed to be very dicult to train. We now know that algorithms that have existed since the 1980s work quite well, but this was not apparent circa 2006. The issue is perhaps simply that these algorithms were too computationally costly to allow much experimentation with the hardware available at the time. The third wave of neural networks research began with a breakthrough in 18

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 33)

(score=0.61)

CHAPTER 1. INTRODUCTION 2006. Georey Hinton showed that a kind of neural network called a deep belief network could be eciently trained using a strategy called greedy layer-wise pretraining (,), which will be described in more detail in section . Hinton et al. 2006 15.1 The other CIFAR-aliated research groups quickly showed that the same strategy could be used to train many other kinds of deep networks (, ; Bengio et al. 2007 Ranzato 2007a et al.,) and systematically helped to improve generalization on test examples. This wave of neural networks research popularized the use of the term "deep learning" to emphasize that researchers were now able to train deeper neural networks than had been possible before, and to focus attention on the theoretical importance of depth (, ; , Bengio and LeCun 2007 Delalleau and Bengio 2011 Pascanu 2014a Montufar 2014 ; et al., ; et al.,). At this time, deep neural networks outperformed competing AI systems based on other machine learning technologies as well as hand-designed functionality. This third wave of popularity of neural networks continues to the time of this writing, though the focus of deep learning research has changed dramatically within the time of this wave.

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 34)

(score=0.61)

y is in its infancy, but in the future could in principle be applied to nearly any task. Another crowning achievement of deep learning is its extension to the domain of reinforcement learning. In the context of reinforcement learning, an autonomous agent must learn to perform a task by trial and error, without any guidance from the human operator. DeepMind demonstrated that a reinforcement learning system based on deep learning is capable of learning to play Atari video games, reaching human-level performance on many tasks (,). Deep learning has Mnih et al. 2015 also significantly improved the performance of reinforcement learning for robotics (,). Finn et al. 2015 Many of these applications of deep learning are highly profitable. Deep learning is now used by many top technology companies including Google, Microsoft, Facebook, IBM, Baidu, Apple, Adobe, Netflix, NVIDIA and NEC. Advances in deep learning have also depended heavily on advances in software infrastructure. Software libraries such as Theano (, ; Bergstra et al. 2010 Bastien et al. et al. ,), PyLearn2 (2012 Goodfellow ,), Torch (,), 2013c Collobert et al. 2011b DistBelief (,), Cae (,), MXNet (,), and Dean et al. 2012 Jia 2013 Chen et al. 2015 TensorFlow (,) have all supported important research projects or Abadi et al. 2015 commercial products. Deep learning has also made contributions back to other sciences. Modern convolutional networks for object recognition provide a model of visual processing 25

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 40)

(score=0.60)

CHAPTER 1. INTRODUCTION 1.2.1 The Many Names and Changing Fortunes of Neural Networks We expect that many readers of this book have heard of deep learning as an exciting new technology, and are surprised to see a mention of "history" in a book about an emerging field. In fact, deep learning dates back to the 1940s. Deep learning only appears to be new, because it was relatively unpopular for several years preceding its current popularity, and because it has gone through many dierent names, and has only recently become called "deep learning." The field has been rebranded many times, reflecting the influence of dierent researchers and dierent perspectives. A comprehensive history of deep learning is beyond the scope of this textbook. However, some basic context is useful for understanding deep learning. Broadly speaking, there have been three waves of development of deep learning: deep learning known as cybernetics in the 1940s-1960s, deep learning known as connectionism in the 1980s-1990s, and the current resurgence under the name deep learning beginning in 2006. This is quantitatively illustrated in

Can Learning Algorithm But Complex

figure . 1.7 Some of the earliest learning algorithms we recognize today were intended to be computational models of biological learning, i.e. models of how learning happens or could happen in the brain. As a result, one of the names that deep learning has gone by is artificial neural networks (ANNs).

(Source: Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 28)

(score=0.59)

Can Learning Algorithm But Complex

Figure Pointers

Figure 1.1 - see Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 19

Figure 1.2 - see Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 21

Figure 1.3 - see Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 22

Figure 1.4 - see Deep Learning_Goodfellow_Bengio_Courville_Part1.pdf, p. 24

Table of Contents (moved to end)