Deep Learning with Tensorflow

The majority of data in the world is unlabeled and unstructured, for instance images, sound, and text data. Shallow neural networks cannot easily capture relevant structures within this type of data, but deep networks are capable of discovering the hidden structures.

In this course, you will use the TensorFlow library to apply deep learning on different types of data to solve real world problems.

Learning Objectives

- Using TensorFlow for Deep Learning
- Breaking down images into their principal components and automatically generating a caption for it;
- Recommending movies, products, anything, based on what a certain person likes
- Processing incomplete sentences and predicting what was going to be written afterwards;
- Deploying your Deep Learning model on the Cloud

总结：

Syllabus

Module 1 - Intro to TensorFlow

- Intro to TensorFlow
- Intro to Deep Learning
- Deep Neural Networks

Module 2 - Convolutional Neural Networks (CNNs)

- Intro to CNNs
- CNNs for Classification
- CNN Architecture

Module 3 - Recurrent Neural Networks (RNNs)

- The Sequence Problem
- The RNN Model
- The LSTM Model
- Applying RNs to Language Modelling

Module 4 - Restricted Boltzmann Machines (RBMs)

- Intro to RBMs
- RBMs

Module 5 - Autoencoders

- Intro to Autoencoders
- Autoencoders

Module 6 - Deployment

- Train and Save Models in the Cloud
- Deploy Models in the Cloud

总结：

Module 1 - Intro to TensorFlow

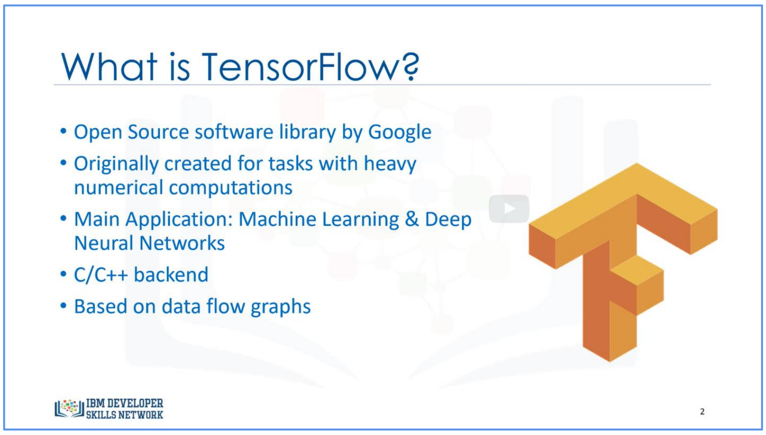
Learning Objectives

- Introduction to TensorFlow
- Linear and Logistic Regression with Tensorflow
- Fundamentals of Deeplearning

Intro to TensorFlow

Introduction to TensorFlow™

IBM DEVELOPER SKILLS NETWORK



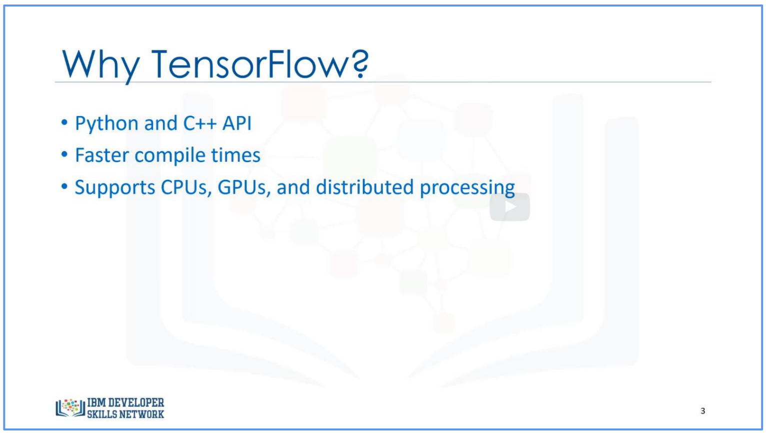
What is TensorFlow?

- Open Source software library by Google
- Originally created for tasks with heavy numerical computations
- Main Application: Machine Learning & Deep Neural Networks
- C/C++ backend
- Based on data flow graphs

IBM DEVELOPER SKILLS NETWORK

What is TensorFlow?

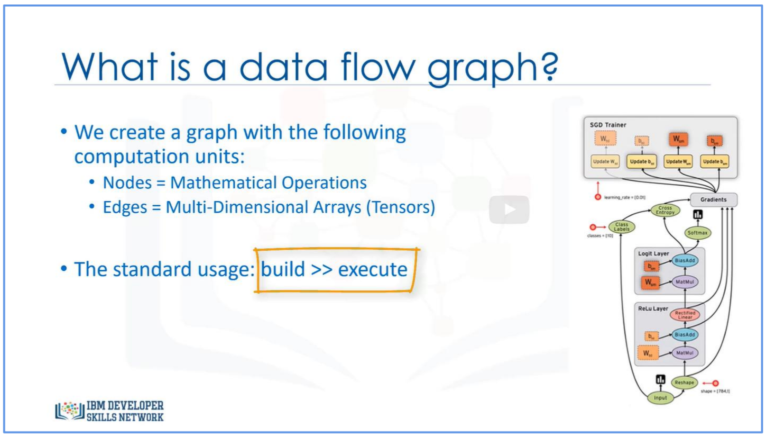
- TensorFlow is an open source library developed by the Google Brain Team.
- It's an extremely versatile library, originally created for tasks that require heavy numerical computations.
- For this reason, TensorFlow was geared towards the problem of machine learning, and deep neural networks.
- Due to a C/C++ backend, TensorFlow is able to run faster than pure Python code.
- The last thing I'd like to mention is that a TensorFlow application uses a structure known as a data flow graph, which is very useful to first build and then execute it in a session. It is also a very common programming model for parallel computing. We'll cover this in more detail shortly.



Why TensorFlow?

- Python and C++ API
- Faster compile times
- Supports CPUs, GPUs, and distributed processing

IBM DEVELOPER SKILLS NETWORK



What is a data flow graph?

- We create a graph with the following computation units:
  - Nodes = Mathematical Operations
  - Edges = Multi-Dimensional Arrays (Tensors)
- The standard usage: build >> execute

IBM DEVELOPER SKILLS NETWORK

Why TensorFlow?

TensorFlow offers several advantages for an application.

- For instance, it provides both a Python and a C++ API. It should be noted, though, that Python API is more complete and it's generally easier to use.
- TensorFlow also has great compilation times in comparison to the alternative deep learning libraries.
- And it supports CPUs, GPUs, and even distributed processing in a cluster. It is a very important feature as you can train a neural network using CPU and multiple GPUs, which makes the models very efficient on large-scale systems.

TensorFlow's structure is based on the execution of a **Data Flow Graph**. So, let's take a closer look at this.

A data flow graph, has two basic units.

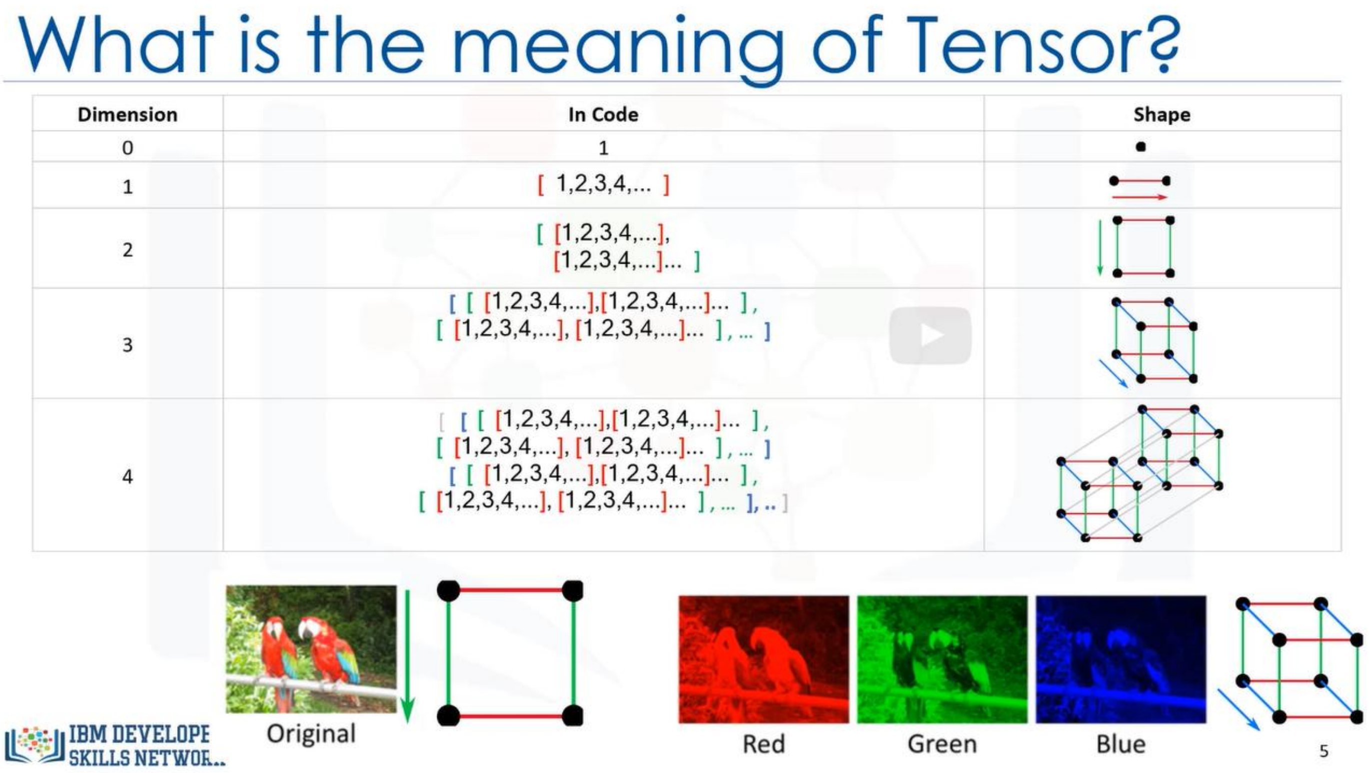
- The **nodes** that represent a **mathematical operation**,
- The **edges** which represent the **multi-dimensional arrays**, known as a **tensors**.

So this high-level abstraction reveals how the data flows between operations. So Data Flow Graph is a representation of data dependencies between operations.

Please notice that using the data flow graph, we can easily visualize different parts of the graph, which is not an option while using other Python libraries such as Numpy or SciKit.

The standard usage is to build a graph first and then execute it in a session.

总结：



What is the meaning of Tensor?

Dimension	In Code	Shape
0	1	
1	[1,2,3,4,...]	
2	[[1,2,3,4,...], [1,2,3,4,...]...]	
3	[[[1,2,3,4,...],[1,2,3,4,...]...], [[1,2,3,4,...], [1,2,3,4,...]...], ...]	
4	[[[[1,2,3,4,...],[1,2,3,4,...]...], [[1,2,3,4,...], [1,2,3,4,...]...], ...] [[[1,2,3,4,...],[1,2,3,4,...]...], [[1,2,3,4,...], [1,2,3,4,...]...], ...], ..]	

So, What is a Tensor?

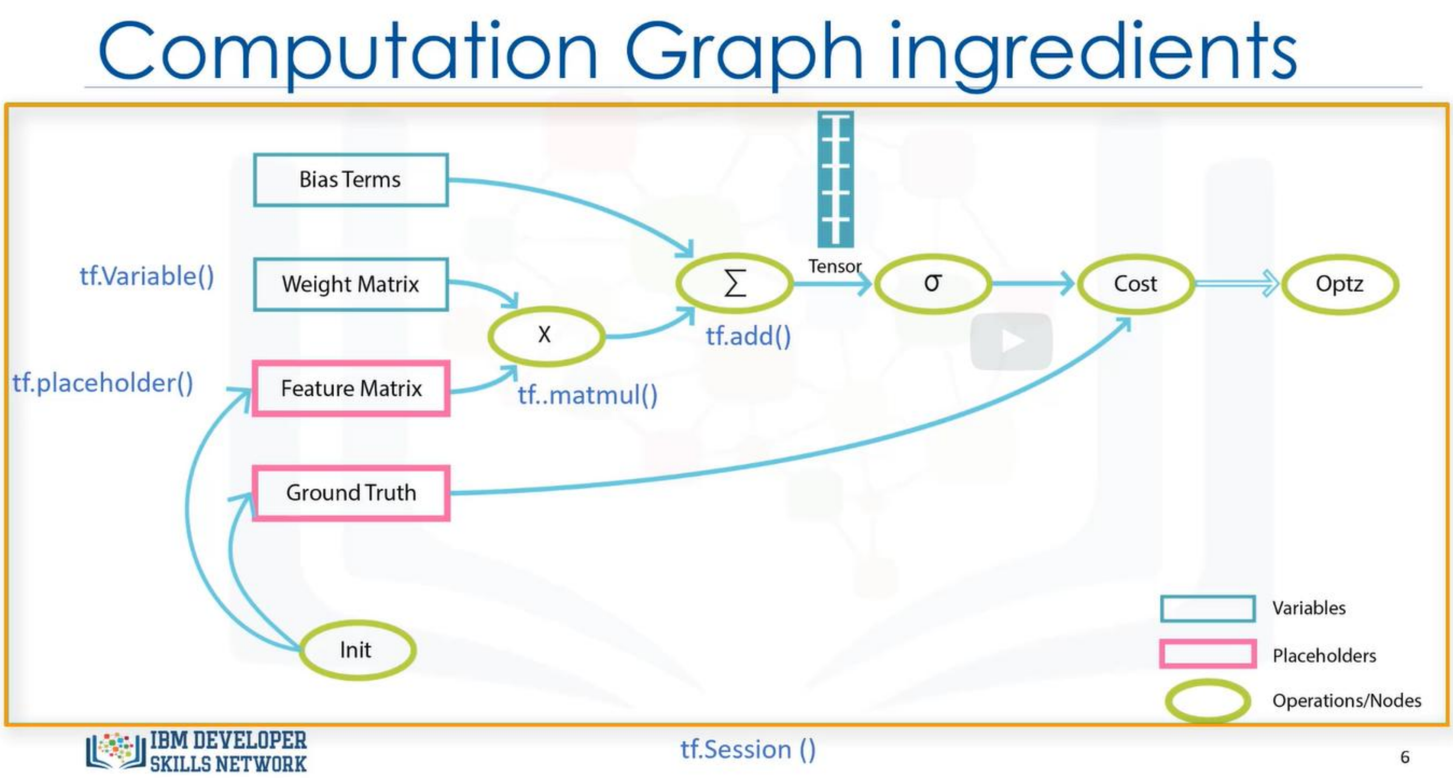
Well as we've already noted, the data that's passed between the operations are Tensors. In effect, a Tensor is a multidimensional array. It can be

- zero dimensional, such as scalar values.
- one dimensional as a line orvector.
- 2-dimensional, such as a Matrix, and so on.

The Tensor structure helps us by giving us the freedom to shape the dataset the way we want. It's also particularly helpful when dealing with images, due to the nature of how the information within images is encoded. Indeed, when you think about an image, it's easy to understand that it has height and width, so it would make sense to represent the information contained in it with a two dimensional structure, such as a matrix, for example.

But as you know, images have colors, and to add information about the colors, we need another dimension, and that's when a 3-dimensional tensor becomes particularly helpful.

总结：



Computation Graph ingredients

tf.Session ()

6

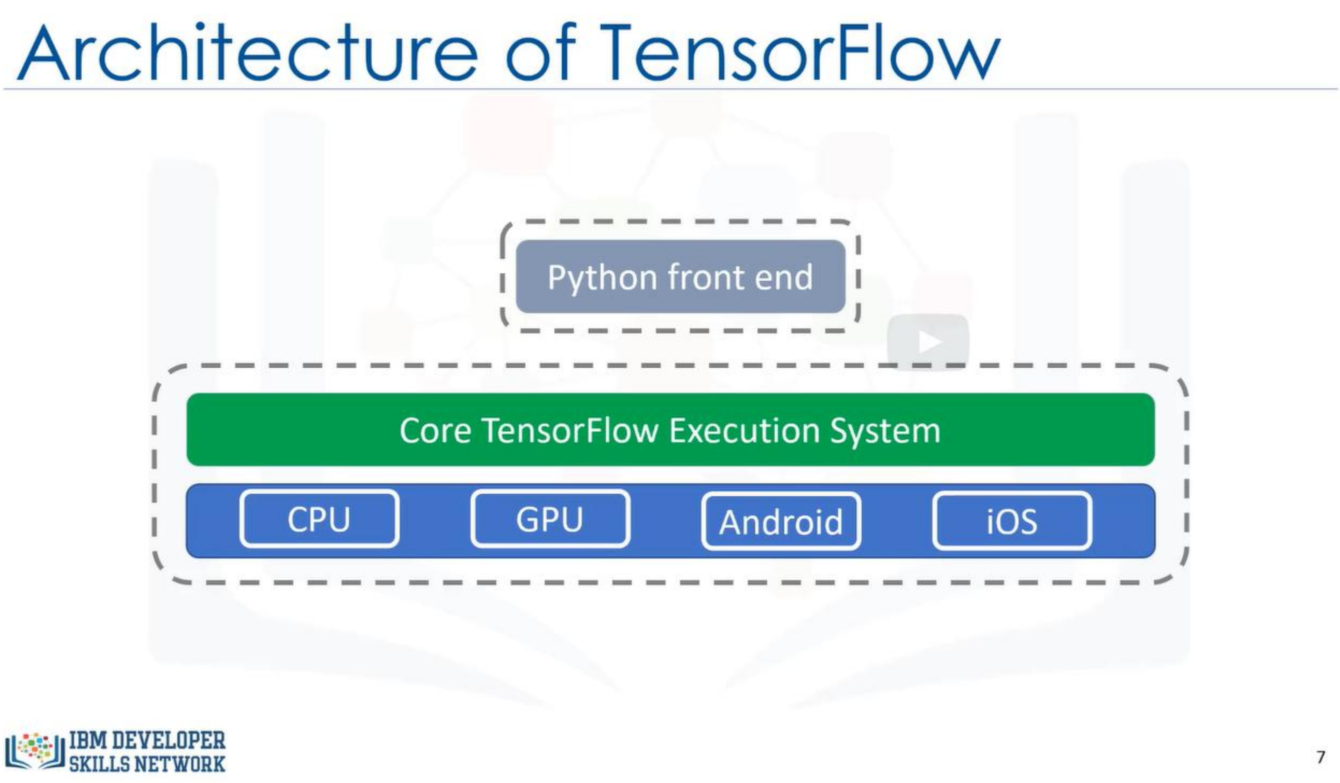
Now, let’s take a look at a dataflow graph and see how tensors and operations build the graph. As mentioned before, in a dataflow graph, the nodes are called operations, which represent units of computation. The edges are tensors which represent the data consumed or produced by an operation. In this graph,

- Feature matrix is a placeholder. Placeholders can be seen as "holes" in your model -- "holes" through which you can pass the data from outside of the graph. Placeholders allow us to create our operations in the graph, without needing the data. When we want to execute the graph, we have to feed placeholders with our input data. This is why we need to initialize placeholders before using them. 即：**tf.placeholder()**
- Let’s look at another operation which builds the variables for our program. In this graph, Weight Matrix is a variable. TensorFlow variables, are used to share and persist some values, that are manipulated by the program.
- 即：**tf.Variable()**

Please notice that when you define a placeholder or variable, TensorFlow adds an operation to your graph. In our graph, “Weight matrix” and “Feature matrix” should be multiplied using a multiplication operation. After that, Add operation is called, which adds the result of the previous operation with bias term.
The output of each operation is a tensor. The resulting tensors of each operation crosses the next one until the end where it's possible to get the desired result. After adding all these operations in a graph.

We can create a session to run the graph, and perform the computations.
For a better understanding of TensorFlow graphs, I recommend that you run the labs from this module, which walk you through different elements of the data flow, in the TensorFlow library.

总结：



Architecture of TensorFlow

Python front end

Core TensorFlow Execution System

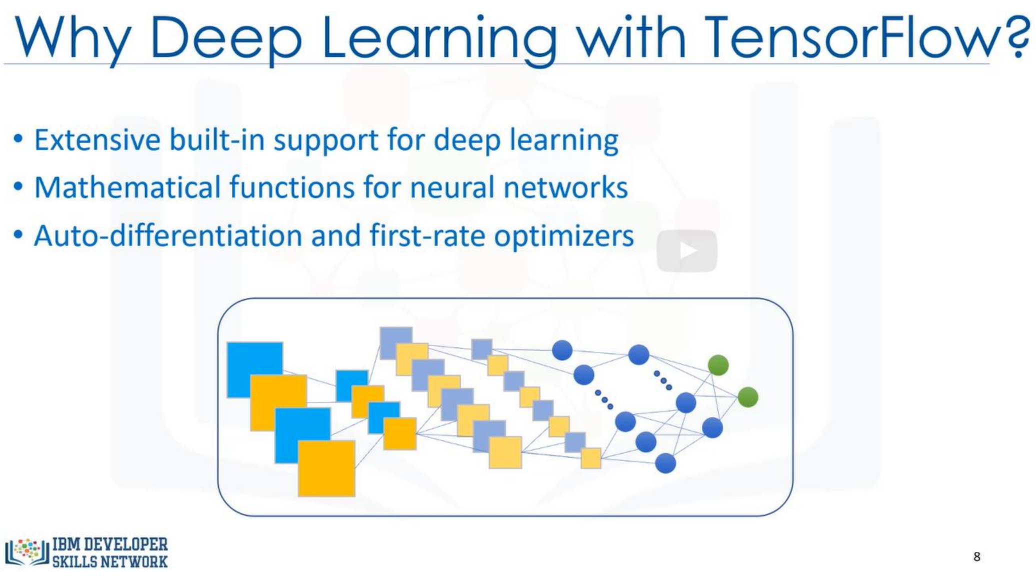
CPU GPU Android iOS

IBM DEVELOPER SKILLS NETWORK

7

As previously mentioned, TensorFlow comes with an easy to use Python interface to build and execute your computational graphs. But what makes TensorFlow so popular today, is its architecture.

TensorFlow's flexible architecture allows you to deploy computation on one or more CPUs, or GPUs, or on a desktop, server, or even a mobile device. This means you build your program once, and then you can run it easily on different devices.



Why Deep Learning with TensorFlow?

- Extensive built-in support for deep learning
- Mathematical functions for neural networks
- Auto-differentiation and first-rate optimizers

IBM DEVELOPER SKILLS NETWORK

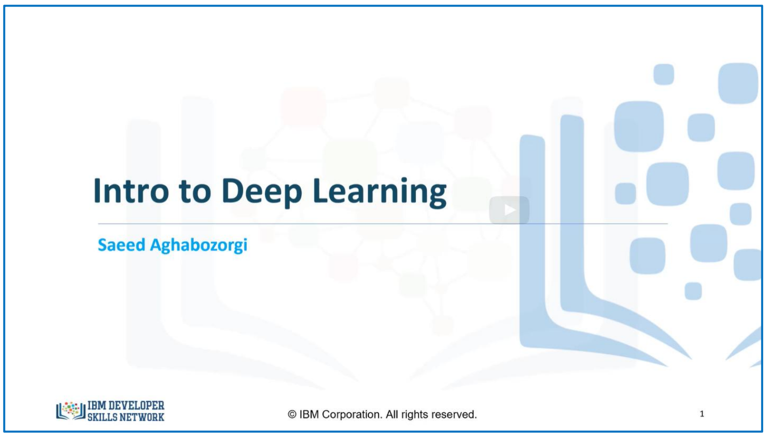
8

So let's briefly review the reasons why TensorFlow is well-suited for deep learning applications.

- First, TensorFlow has built-in support for deep learning and neural networks, so it's easy to assemble a net, assign parameters, and run the training process.
- Second, it also has a collection of simple, trainable mathematical functions that are useful for neural networks.
- Finally, deep learning as a gradient-based machine learning algorithm will benefit from TensorFlow's auto-differentiation and optimizers.

总结：

Intro to Deep Learning



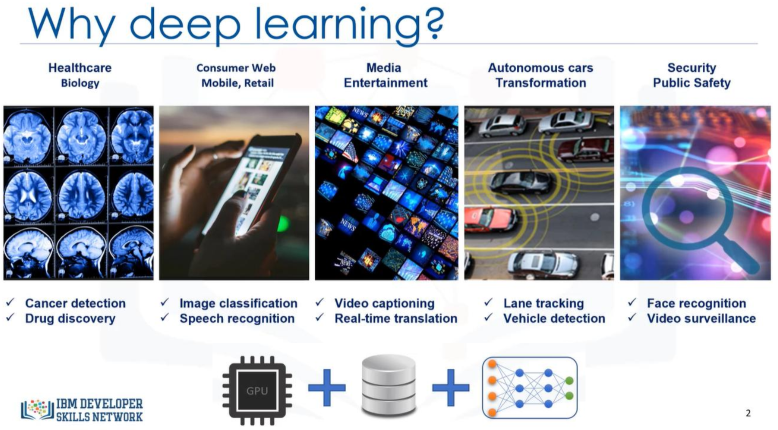
Intro to Deep Learning

Saeed Aghabozorgi

IBM DEVELOPER SKILLS NETWORK

© IBM Corporation. All rights reserved.

1



Why deep learning?

Healthcare Biology
- ✓ Cancer detection
- ✓ Drug discovery

Consumer Web Mobile, Retail
- ✓ Image classification
- ✓ Speech recognition

Media Entertainment
- ✓ Video captioning
- ✓ Real-time translation

Autonomous cars Transformation
- ✓ Lane tracking
- ✓ Vehicle detection

Security Public Safety
- ✓ Face recognition
- ✓ Video surveillance

IBM DEVELOPER SKILLS NETWORK

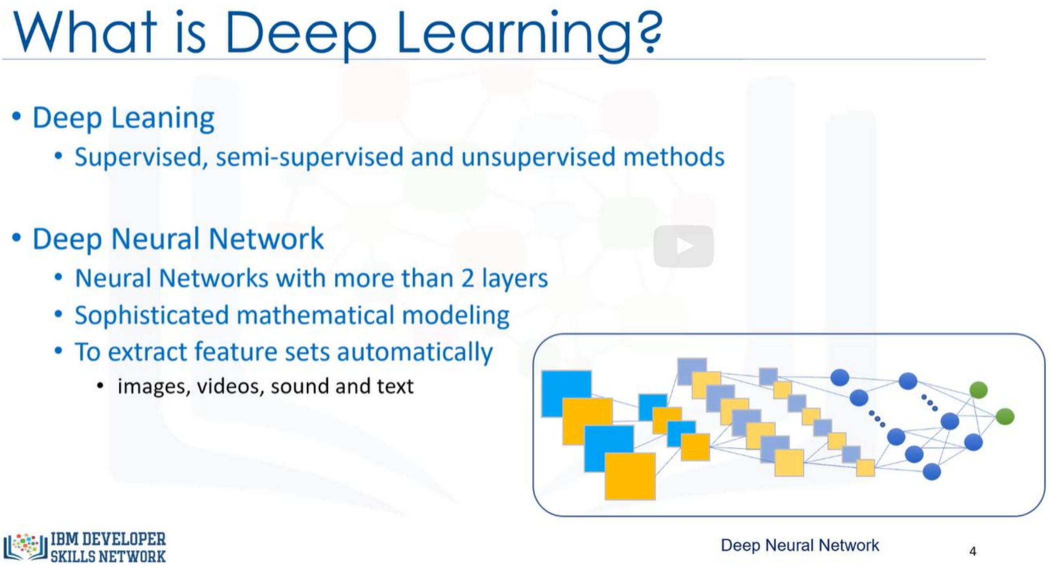
2

In this video, we will provide an introduction to Deep Learning, and see why it is such a hot topic today.
It is not an overstatement to say that Deep learning can be found all around us, as it is used in countless ways across all industries. For example,

- Deep Learning is trying to help the **health care industry** on tasks such as cancer detection and drug discovery.
- In the **internet service and mobile phone industries**, we can see various apps which are using Deep Learning for image/video classification and speech recognition, such as, Google Voice, Apple Siri, Microsoft Skype, and so on.
- In **media, entertainment, and news**, we can see applications such as video captioning, real-time translation and personalization, or recommendation systems such as Netflix.
- In the development of **self-driving cars**, Deep Learning is helping researchers to overcome significant research problems, such as sign and passenger detection or lane tracking.
- In the **Security field**, Deep Learning is used for face recognition and video surveillance.

These are just a few examples of the industries in which Deep Learning is being applied; and it is being used in many other fields and domains as well.
The increasing popularity of Deep Learning today comes from three recent advances in our field:

- First, in the dramatic increases in computer processing capabilities;
- second, in the availability of massive amounts of data for training computer systems;
- and third, in the advances in machine learning algorithms and research.



What is Deep Learning?

- Deep Leaning
  - Supervised, semi-supervised and unsupervised methods
- Deep Neural Network
  - Neural Networks with more than 2 layers
  - Sophisticated mathematical modeling
  - To extract feature sets automatically
    - images, videos, sound and text

Deep Neural Network

IBM DEVELOPER SKILLS NETWORK

4

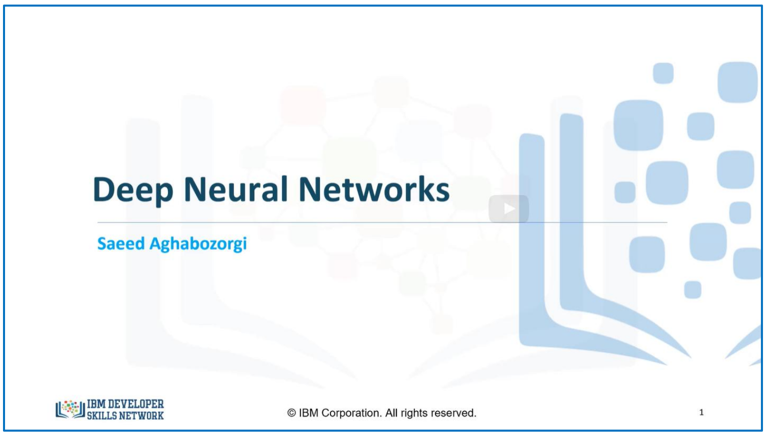
So, what is deep learning?
Deep Learning is a series of supervised, semi-supervised and unsupervised methods that try to solve some machine learning problems using deep neural networks.

A deep neural network is a neural network which often has more than two layers, and uses specific mathematical modeling in each layer to process data.

Generally speaking, these networks are trying to automatically extract feature sets from the data, and this is why, they are mostly used in data types where the feature selection process is difficult, such as when analyzing unstructured datasets, such as image data, videos, sound and text.

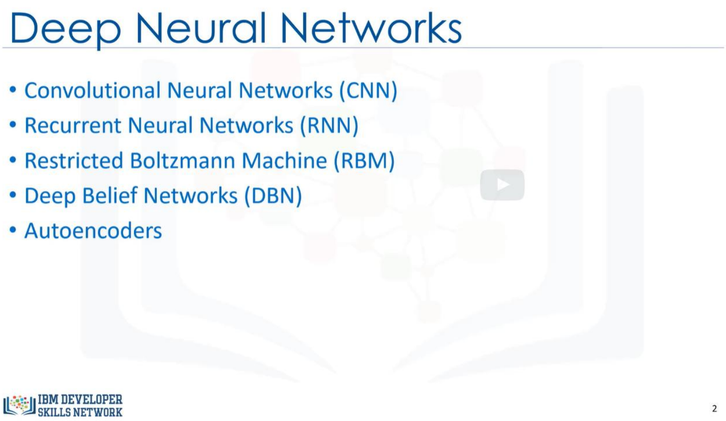
总结：

Deep Neural Networks



Deep Neural Networks

Saeed Aghabozorgi



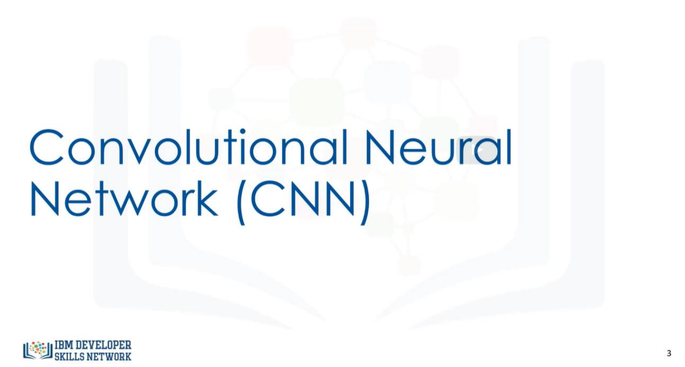
Deep Neural Networks

- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Restricted Boltzmann Machine (RBM)
- Deep Belief Networks (DBN)
- Autoencoders

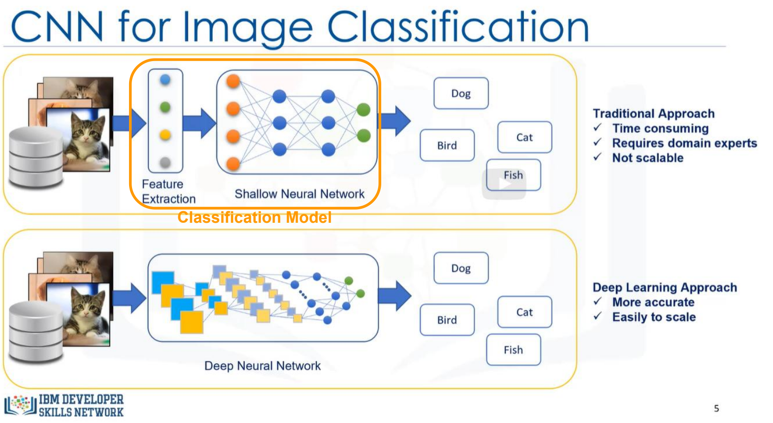
In this video, we will provide an overview of several deep neural network models, and their applications. To better understand Deep Learning, let’s first take a look at different deep neural networks and their applications, namely:

- **Convolutional Neural Networks (or CNNs)**
- **Recurrent Neural Networks (or RNNs)**
- **Restricted Boltzmann Machines (or RBMs)**
- **Deep Belief Networks (or DBNs)**
- **Autoencoders.**

总结：



Convolutional Neural Network (CNN)



CNN for Image Classification

Traditional Approach
- ✓ Time consuming
- ✓ Requires domain experts
- ✓ Not scalable

Deep Learning Approach
- ✓ More accurate
- ✓ Easily to scale

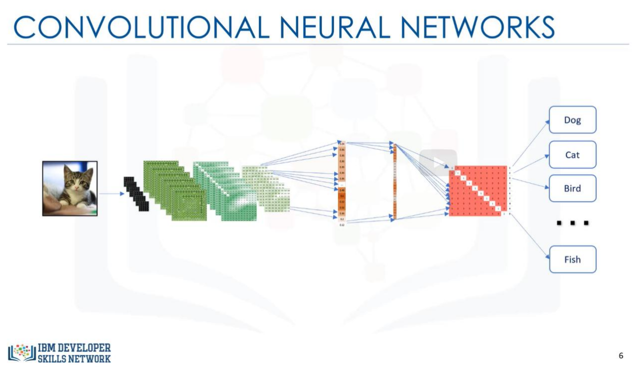
Let's start with the Convolutional Neural Network, and see how it helps us to do a task, such as image classification.
Assume that you have a dataset made up of a great many photos of cats and dogs, and you want to build a model that can recognize and differentiate them. This model is supposed to look at this particular sample set of images and learn from them, toward becoming trained. Later, given an unseen image of either a cat or a dog, we should be able to use this trained model to recognize the image as being one or the other.

Traditionally:

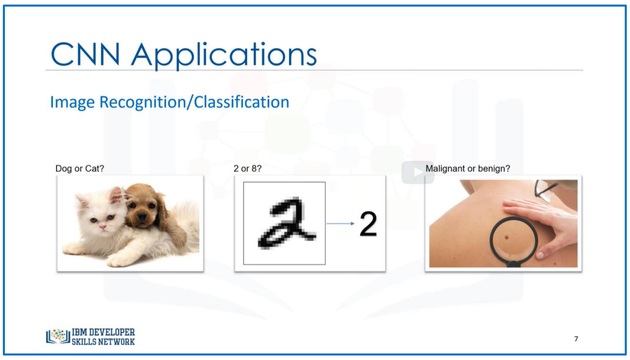
- Your first step in building such a model would be "feature extraction." That is, to choose the best features from your images, and then use those features in a classification algorithm, such as a shallow Neural Network. Ideally, the result would be a model that, upon analyzing a new image, could accurately distinguish the animal in that photo as being either a "cat" or a "dog."
- There are countless features that could be extracted from the image, such as color, object edges, pixel location, and so on. Of course, the better that you can define the feature sets, the more accurate and efficient your image-classification will be. In fact, in the last two decades, there has been a lot of scientific research in the field of image processing, that is devoted to helping data scientists find the best feature sets from images, for the purposes of classification.
- However, as you can imagine, the process of selecting and using the best features is tremendously time-consuming and is often ineffective.
- Furthermore, extending the features to other types of images becomes an even greater challenge. Indeed, it's easy to see how difficult it would be to apply the features we've used to discriminate cats and dogs, to other discrimination tasks, such as recognizing hand-written digits, for example. Therefore, the importance of feature selection can't be overstated.

Enter "convolutional neural networks".

- Suddenly, without having to find or select features, this network automatically and effectively finds the best features for you.
- So instead of you choosing what image features to use in classifying dogs vs. cats, Convolutional Neural Networks can automatically find those features and classify the images for you.
- So, we can say that a Convolutional Neural Network - or CNN for short - is a deep learning approach that learns directly from samples in a way that is much more effective than traditional Neural networks.



CONVOLUTIONAL NEURAL NETWORKS



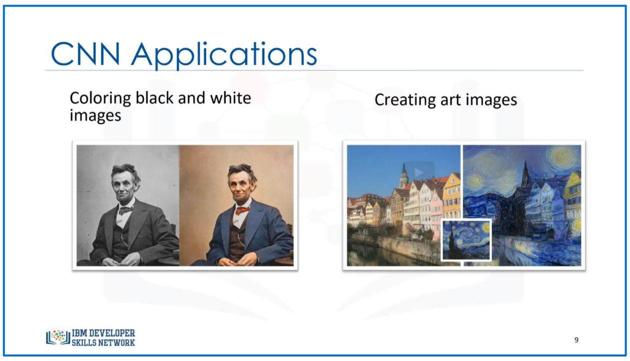
CNN Applications

Image Recognition/Classification

CNN Applications

Object Detection in images

self-driving vehicles



CNN Applications

Coloring black and white images

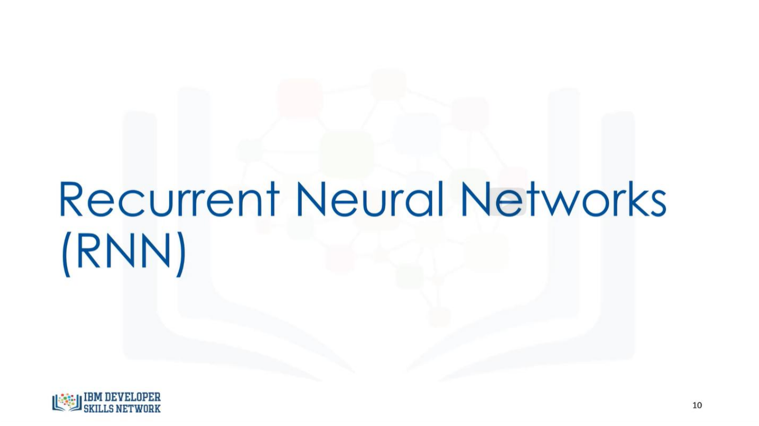
Creating art images

CNNs achieve this type of automatic feature selection and classification through multiple specific layers of sophisticated mathematical operations. Through multiple layers, a CNN learns multiple levels of feature sets at different levels of abstraction. And this leads to very effective classification.

CNNs have gained a lot of attention in the machine learning community over the last few years. This is due to the wide range of applications where CNNs excel, especially machine vision projects, including

- **image recognition or classification**, such as distinguishing animal photos or digit recognition, to skin cancer classification.
- CNNs are also used in **object detection**, for example real-time recognition of passengers in images captured by self-driving cars. Or coloring black and white images, and creating art images.

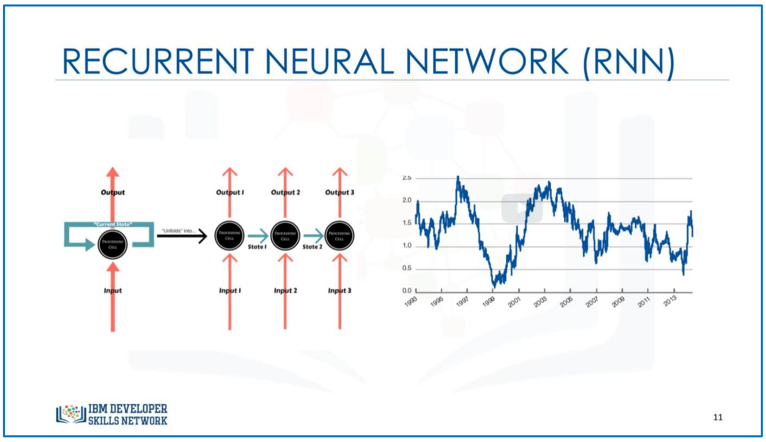
总结：



Recurrent Neural Networks (RNN)

IBM DEVELOPER SKILLS NETWORK

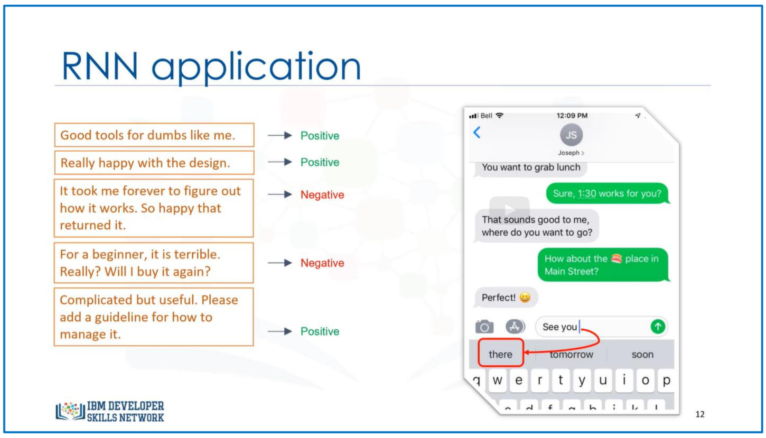
10



RECURRENT NEURAL NETWORK (RNN)

IBM DEVELOPER SKILLS NETWORK

11



RNN application

Good tools for dumbs like me.	Positive
Really happy with the design.	Positive
It took me forever to figure out how it works. So happy that returned it.	Negative
For a beginner, it is terrible. Really? Will I buy it again?	Negative
Complicated but useful. Please add a guideline for how to manage it.	Positive

Bell 12:09 PM

JS

Joseph ›

You want to grab lunch

Sure, 1:30 works for you?

That sounds good to me, where do you want to go?

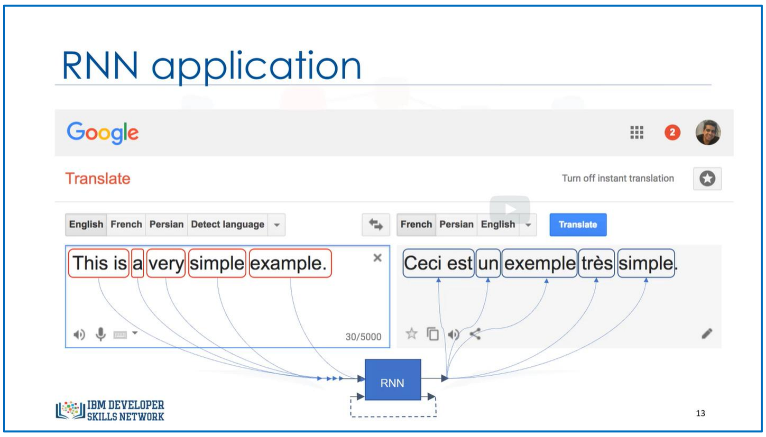
How about the 🍣 place in Main Street?

Perfect! 😀

See you

IBM DEVELOPER SKILLS NETWORK

12



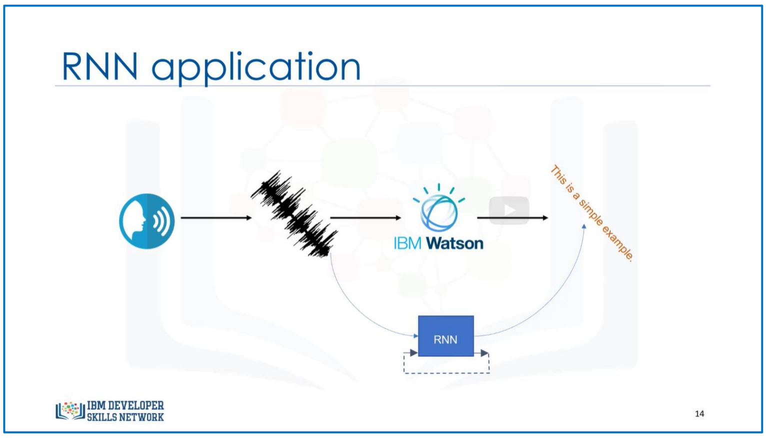
RNN application

This is a very simple example.

Ceci est un exemple très simple.

IBM DEVELOPER SKILLS NETWORK

13



RNN application

IBM DEVELOPER SKILLS NETWORK

14

Now, let's look at Recurrent Neural networks and the types of situations in which they can be used, to solve a problem.
A Recurrent Neural Network, or RNN for short, is a type of deep learning approach, that tries to solve the problem of modeling sequential data. Whenever the points in a dataset are dependent on the previous points, the data is said to be sequential.
For example：

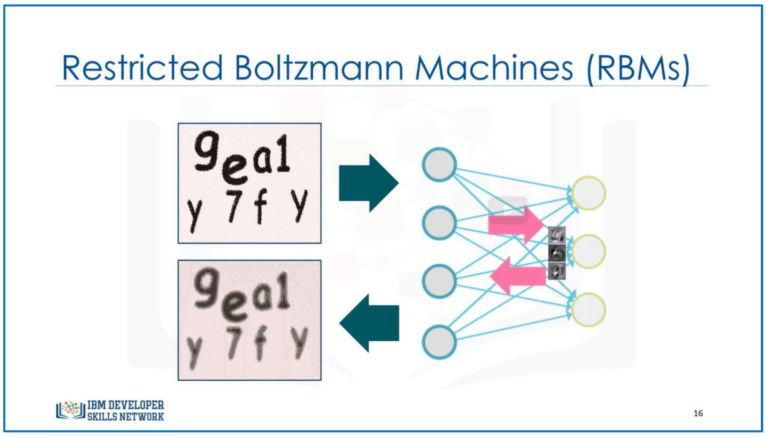
- A stock market price is a sequential type of data because the price of any given stock in tomorrow's market, to a great extent, depends on its price today. As such, predicting the stock price tomorrow, is something that RNNs can be used for. We simply need to feed the network with the sequential data, it then maintains the context of the data and thus, learns the patterns within the data.
- We can also use RNNs for **sentiment analysis**. Let's say for example, you're scrolling through your product catalogue on a social network site and you see many comments related to a particular product of yours. Rather than reading through dozens and dozens of comments yourself and having to manually calculate if they were mostly positive, you can let an RNN do that for you. Indeed, an RNN can examine the sentiment of keywords in those reviews. Please remember, though, that the sequence of the words or sentences, as well as the context in which they are used, is very important as well. By feeding a sentence into an RNN, it takes all of this into account and determines if it the sentiment within it those product reviews are positive or negative.
- RNNs can also be used to **predict the next word in a sentence**. I'm sure we've all seen how our mobile phone suggests words when we're typing an email or a text. This is a type of language modeling within RNN, where the model has learned from a big textual corpus, and now can predict the next word in the sentence. As you can see, thinking in a sequential way, the word being suggested is very dependent on the previously typed words and the context of that message.
- When needing **quick translation of certain words into another language**, a great many people today, use the translation service of Google translator. We enter a sequence of words in English and it outputs a sequence of the words in French, as seen here. This type of text translation is another example of how RNNs can be used. This task is not based on a word-by-word translation and applying grammar rules. Instead, it is a probability model that has been trained on lots of data where the exact same text is translated into another language.
- **Speech-to-text** is yet another useful and increasingly common application of RNNs. In this case, the recognized voice is not only based on the word sound; RNNs also use the context around that sound to accurately recognize of the words being spoken into the device's microphone.

总结:

Restricted Boltzmann Machine(RBM)

IBM DEVELOPER SKILLS NETWORK

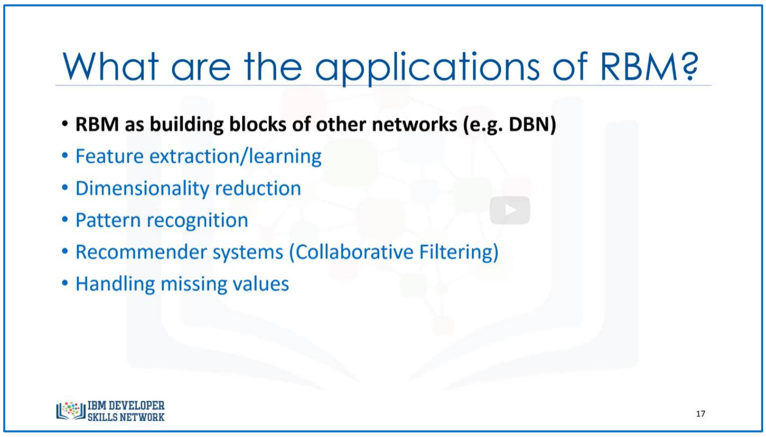
15



Restricted Boltzmann Machines (RBMs)

IBM DEVELOPER SKILLS NETWORK

16



What are the applications of RBM?

- **RBM as building blocks of other networks (e.g. DBN)**
- Feature extraction/learning
- Dimensionality reduction
- Pattern recognition
- Recommender systems (Collaborative Filtering)
- Handling missing values

IBM DEVELOPER SKILLS NETWORK

17

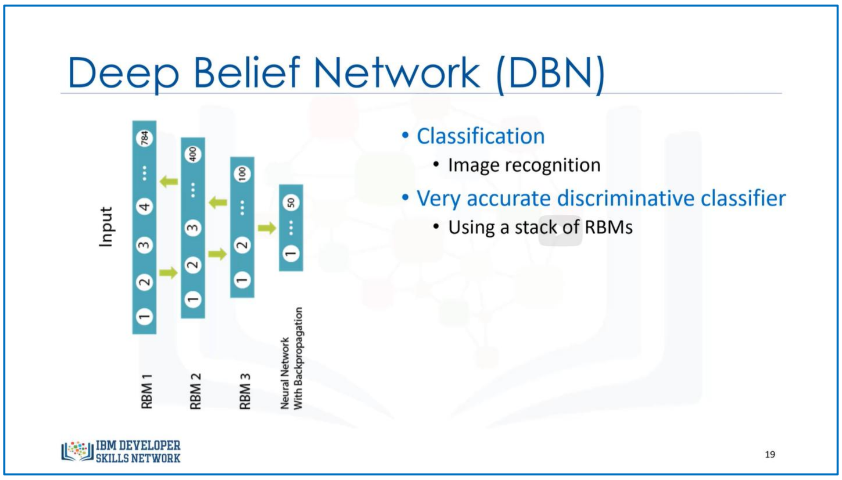
Now, let’s look at another type of neural network called a Restricted Boltzman Machine.
Restricted Boltzman Machines, or RBMs, are used to find the patterns in data in an unsupervised manner. They are shallow neural nets that learn to reconstruct data by themselves. They are very important models, because they can automatically extract meaningful features from a given input, without the need to label them.

RBMs might not be outstanding if you look at them as independent networks, but they are significant as building blocks of other networks, such as Deep Belief Networks. Essentially, RBMs are useful for unsupervised tasks such as:

- feature extraction,
- dimensionality reduction,
- pattern recognition,
- recommender systems,
- handling missing values,
- topic modeling.

总结：

Deep Belief Network (DBN)



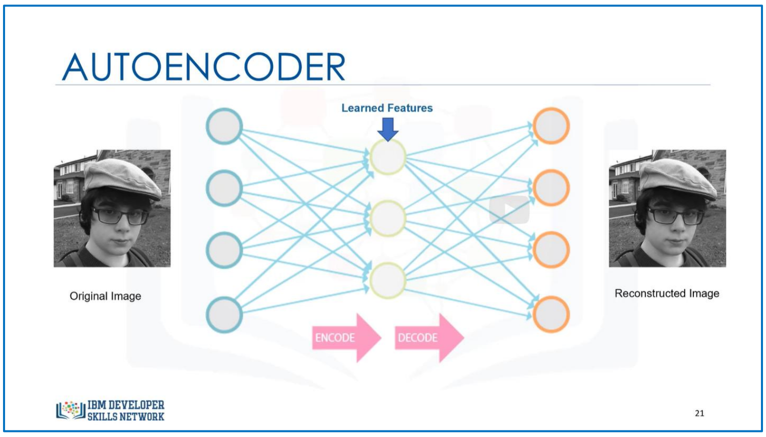
Deep Belief Network (DBN)

- Classification
  - Image recognition
- Very accurate discriminative classifier
  - Using a stack of RBMs

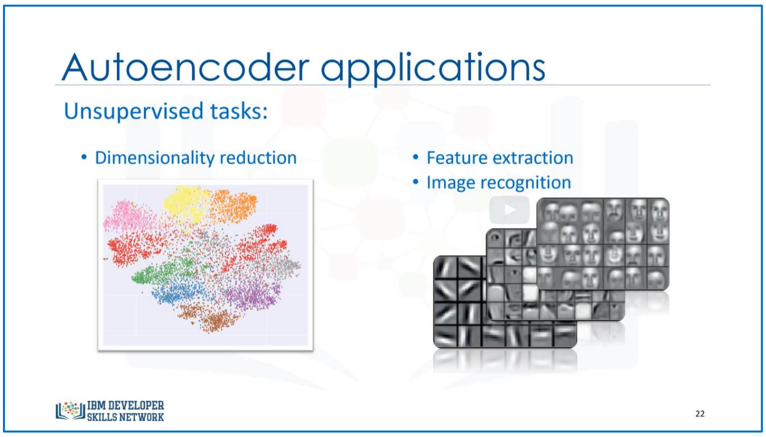
A **Deep Belief Network** is a network that was invented to solve an old problem in traditional artificial neural networks. Which problem?
The back-propagation problem, that can often cause "local minima" or "vanishing gradients" issues in the learning process. A DBN is built to solve this by the stacking of multiple RBMs.
So, what are the applications of **DBNs**? DBNs are generally used for classification -- same as traditional **MLPs**. So, one of the most important applications of DBNs is image recognition.

The important part to remember, here, is that a DBN is a very accurate discriminative classifier. As such, we don't need a big set of labeled data to train a Deep Belief Network; in fact, a small set works fine because feature extraction is unsupervised by a stack of RBMs.

Autoencoder



AUTOENCODER



Autoencoder applications

Unsupervised tasks:

- Dimensionality reduction
- Feature extraction
- Image recognition

Now, let's look at Autoencoders.

- Much like RBMs, Autoencoders were invented to address the issue of extracting desirable features. And again, much like RBMs, Autoencoders try to recreate a given input, but do so with a slightly different network architecture and learning method.
- Autoencoders take a set of unlabeled inputs, encodes them into short codes, and then uses those to reconstruct the original image, while extracting the most valuable information from the data.

What are the applications of Autoencoders? Well, Autoencoders are employed in some of the largest deep learning applications, especially for unsupervised tasks.

- As the encoder part of the network, Autoencoders compress data from the input layer into a short code -- a method that can be used for "dimensionality reduction" tasks.
- Also, in stacking multiple Autoencoder layers, the network learns multiple levels of representation at different levels of abstraction. For example, to detect a face in an image, the network encodes the primitive features, like the edges of a face. Then, the first layer's output goes to the second Autoencoder, to encode the less local features, like the nose, and so on. Therefore, it can be used for Feature Extraction and image recognition.

总结：By now, you should have a good sense of deep neural networks.

总结：