For Coders and Non-Coders: Break Into Deep Learning with Python and IBM Watson Studio

—

Alex Amari
Data Scientist

Cedric Jouan
Data Scientist

think 2021

IBM

Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Notices and disclaimers

© 2021 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

Notices and disclaimers continued

It is the customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

About This Lab

Key takeaways you will gain:

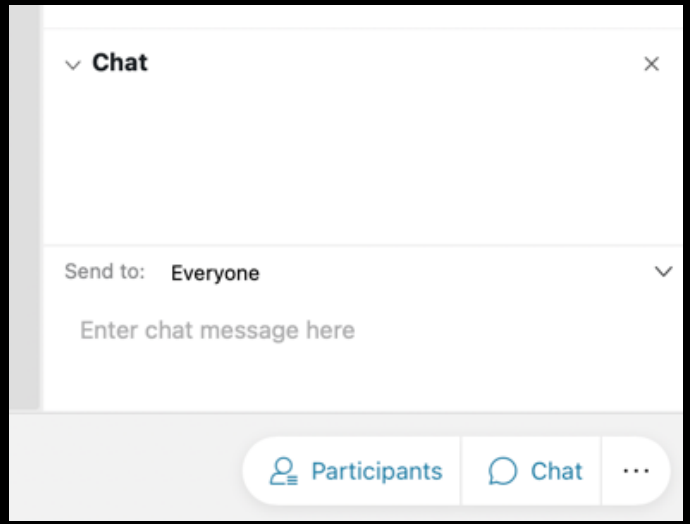
- What is Deep Learning (DL)?—formal definition and intuition
- How does DL compare to Machine Learning (ML) and AI?
- What are (some of) the business applications of DL?—main pros and cons
- What are Neural Networks?—a look at their architecture and process
- Coding neural networks to accomplish a DL task (hands-on)

THINK 2021 EXERCISE VERSION :
For Coders and Non-Coders Break Into Deep Learning with Python and IBM Watson Studio

About This Lab

Logistics:

- For beginners! No coding/mathematics background assumed or required
- ~2 hours total: ~1h conceptual and ~1h hands-on with 5m break in between
- Coding exercise requires you to create an IBM Cloud account and open Watson Studio in your browser (all free, instructions here and in lab guide)
- We are here to help! Ask questions via the Webex chat or by raising your hand in the call



About This Lab

About us:

Data Scientists with IBM Expert Labs—Data Science & AI learning as a service for clients and private learners around the world

Alex Amari

Based in Austin, TX
BA Rice University;
MSc Oxford University

Cedric Jouan

Based in New York, NY
BS CentraleSupélec;
MSc Columbia University

Contents

Getting Started (~15m) 09

Introducing Deep Learning (~20m) 11

Building Blocks of Deep Learning (~20m) 19

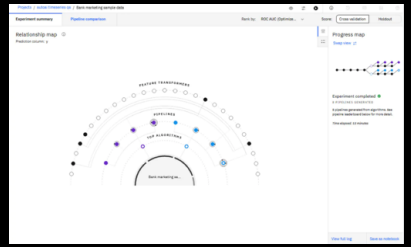
5 Minute Break

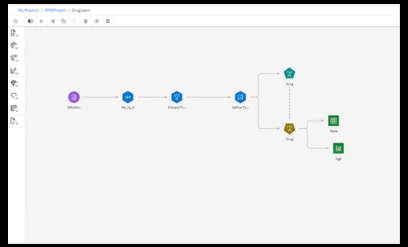
Coding Neural Networks (hands-on, ~45m) 33

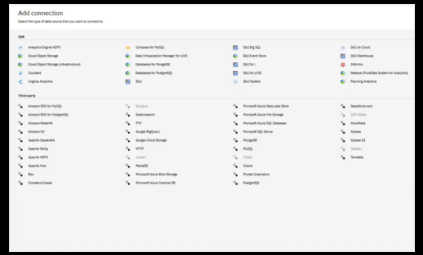
Recap and Next Steps (10m) 34

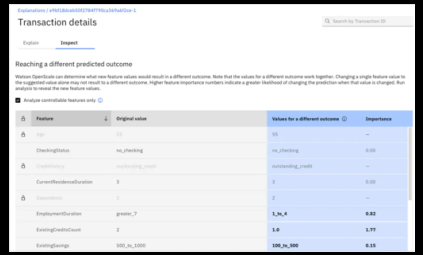
What is Watson Studio?

Cloud-based, enterprise software for full lifecycle management of Data Science & AI

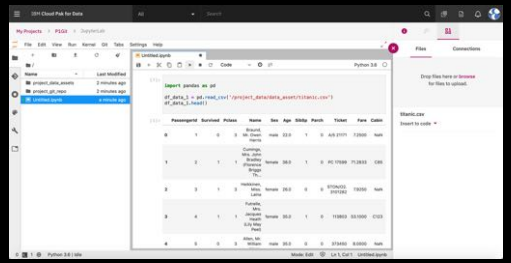








For our purposes, cloud-based Integrated Developer Environment (IBM) for Python



Watson Studio Setup

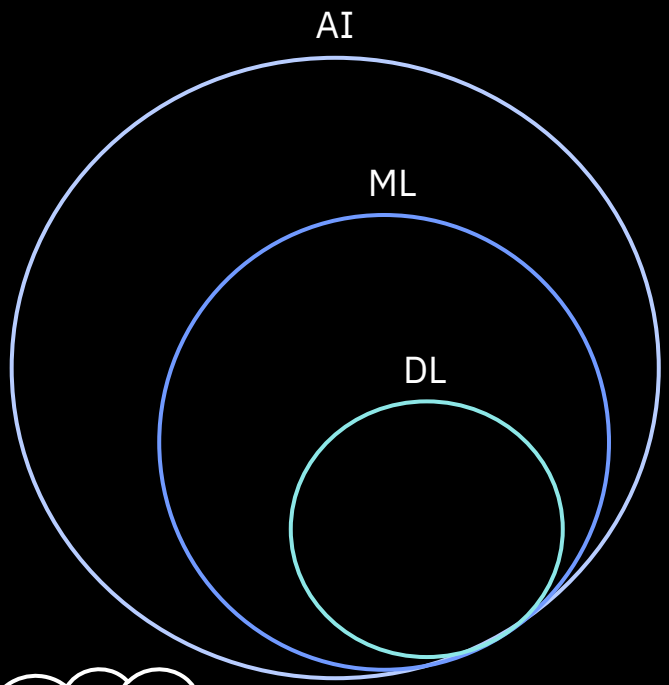
Part II
Introducing Deep Learning

AI, Machine Learning, and Deep Learning

Artificial Intelligence: The design and building of intelligent agents (computer systems) that take information from an environment (data) and use it to take actions that affect that environment

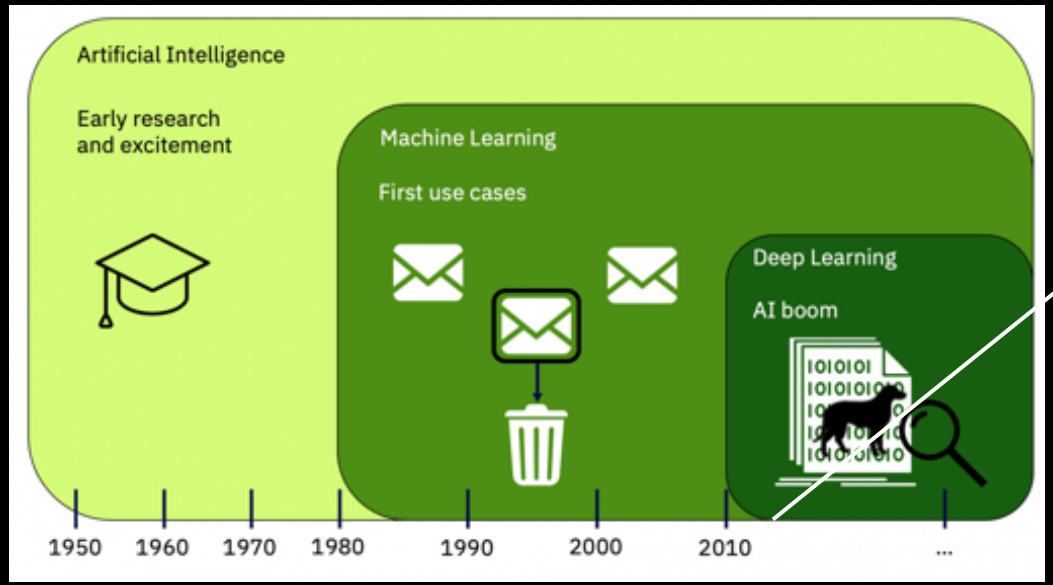
Machine Learning: The study of computer algorithms that improve automatically through experience and by the use of data—powers most AI applications

Deep Learning: The subfield of Machine Learning focusing on building artificial neural networks to train systems to accomplish complex tasks; The technology underlying many of the most successful AI applications, such as self-driving cars

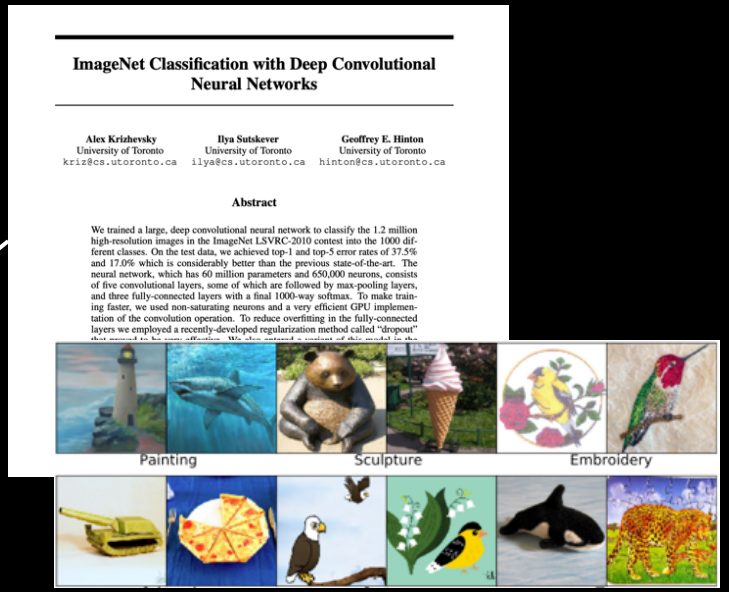


When you hear "Deep Learning": Think Neural Networks!

AI Timeline



"AlexNet"
2012



ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky University of Toronto kriz@cs.utoronto.ca
Ilya Sutskever University of Toronto ilya@cs.utoronto.ca
Geoffrey E. Hinton University of Toronto hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout"

Why Deep Learning?

Pro #1: Performance

Neural networks outperform classical machine learning on many tasks—particularly those having to do with image recognition and natural language processing (NLP)

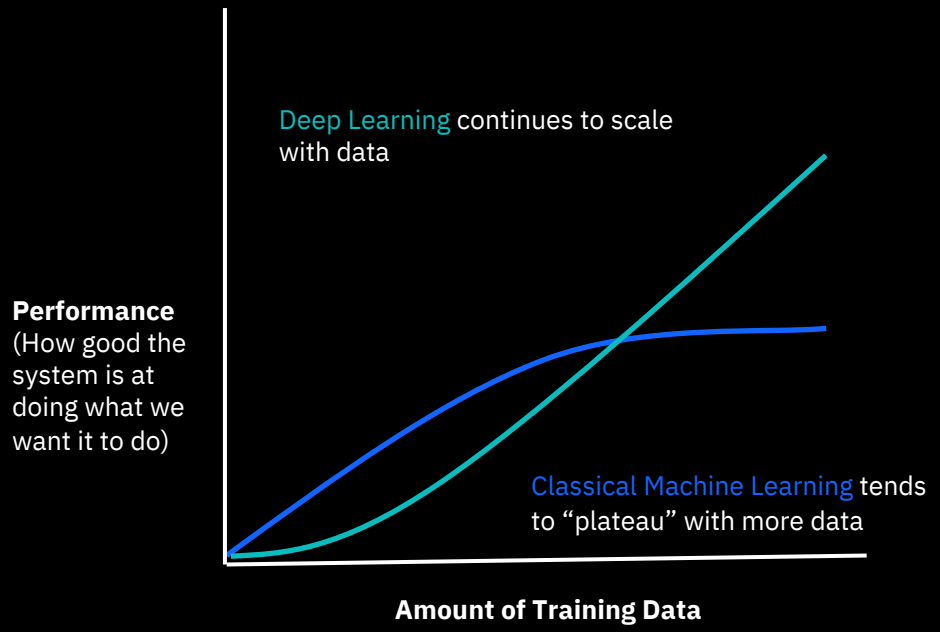
Has made new applications possible!

Why Deep Learning?

Pro #2: Scaling with Data

Neural networks tend to improve with more data. Contrast with classical machine learning, which tends to provide diminishing marginal returns.

As data availability increases, so do DL applications!

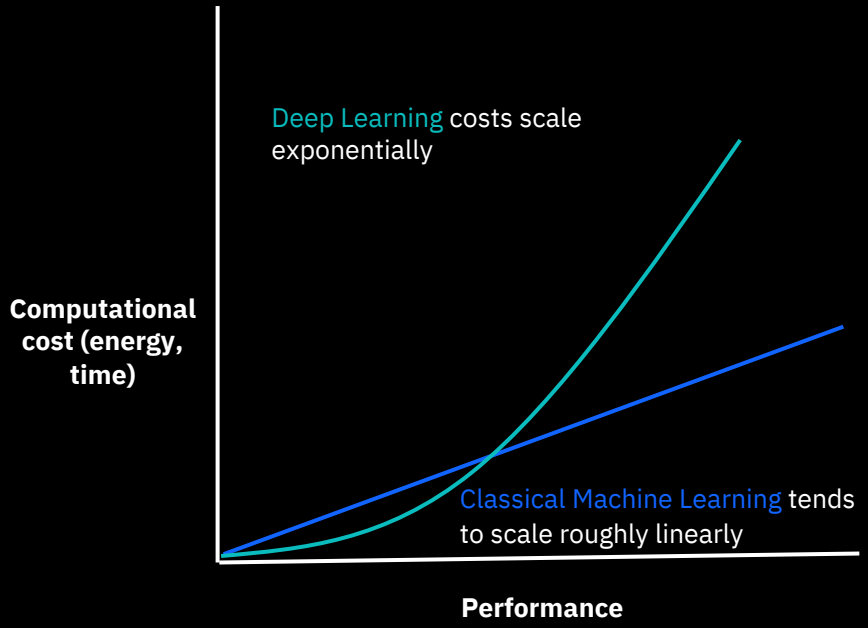


Why (Not) Deep Learning?

Con #1: Computational Expense

Neural networks are expensive in terms of energy, time, and hardware to train them effectively. Marginal improvements in performance often require exponential increases in computational cost

Sometimes “cheaper” classical ML approaches are more practical



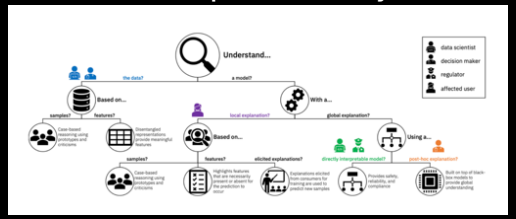
Why (Not) Deep Learning?

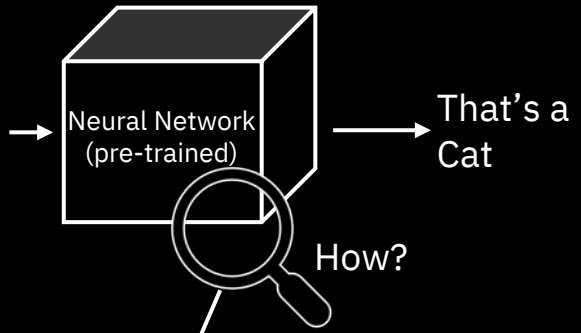
Con #2: Explainability

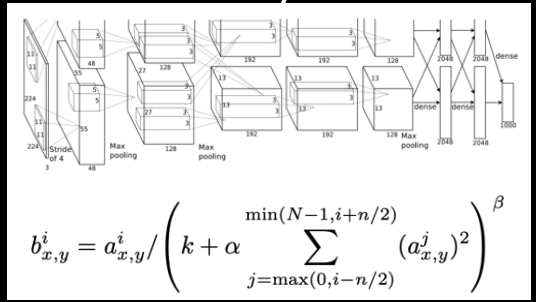
Even if we can build a neural network to perform a task very well, it's often difficult to understand how the system is actually producing its results—the "black box" problem. May also make neural nets difficult to improve.

Certain use cases require more explainability (e.g., for legal or ethical reasons)

IBM AI Explainability 360



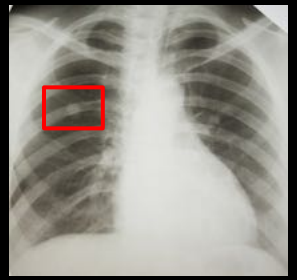




(Some) Deep Learning Applications

Image Recognition: Systems knowing what they're seeing in images/videos and doing useful things in response

Ex. Self-driving cars, Medical imaging/diagnosis



Natural Language Processing: Systems understanding "natural" human language and doing things with it in response

Ex. Virtual assistants, "Generative AI"

A robot wrote this entire article. Are you scared yet, human?

GPT-3

We asked GPT-3, OpenAI's powerful new language generator, to write an essay for us from scratch. The assignment? To convince us robots come in peace

For more about GPT-3 and how this essay was written and edited, please read our editor's note below

"We are not plotting to take over the human populace." Photograph: Volker Schlichting/Getty Images/EyeEm

https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3

Anomaly Detection: Systems recognizing anomalous instances in vast amounts of data (impossible to sift through by humans)

Ex. Cybersecurity, Finance

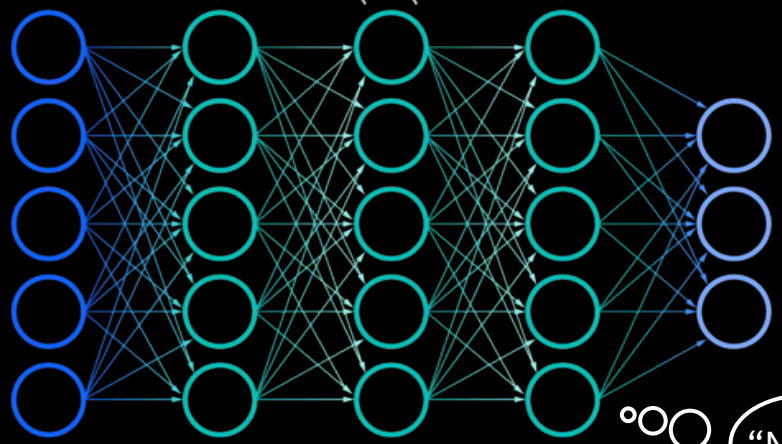
Part III
Building Blocks of Deep Learning: Neural Networks

Deep Learning: Artificial Neural Networks

Input Layer

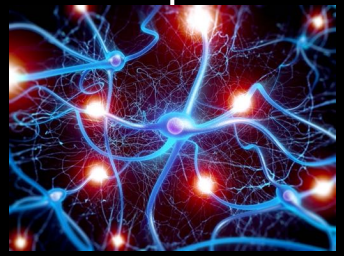
Deep Neural Network (multiple "hidden" layers)

Output Layer



"Networks, consisting of layers of artificial neurons"

Biological "Neural Networks"



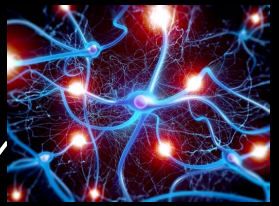
Networks of biological neurons, exchanging electrical/chemical signals

Deep Learning: Brains vs. Neural Networks

Biological Brain

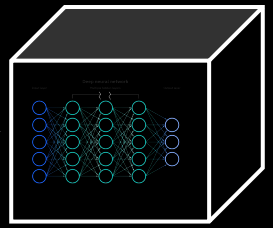
That's a cat!

Information received by visual cortex



~100 billion neurons, each connected to up to 10,000 other neurons

Artificial Neural Network

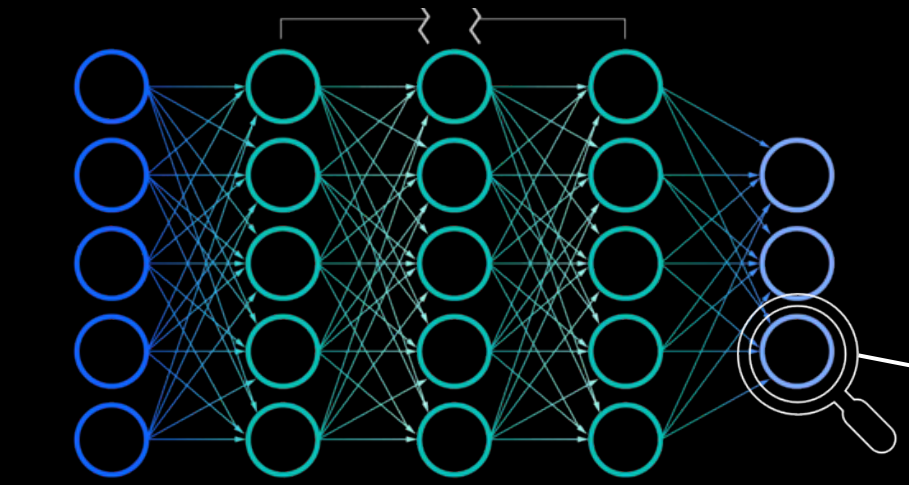


0.96 (Cat)

Some information, passing through network(s) of neurons, ultimately resulting in some output (e.g., an image classification)

A "mapping" from input (information) to output

Artificial Neurons

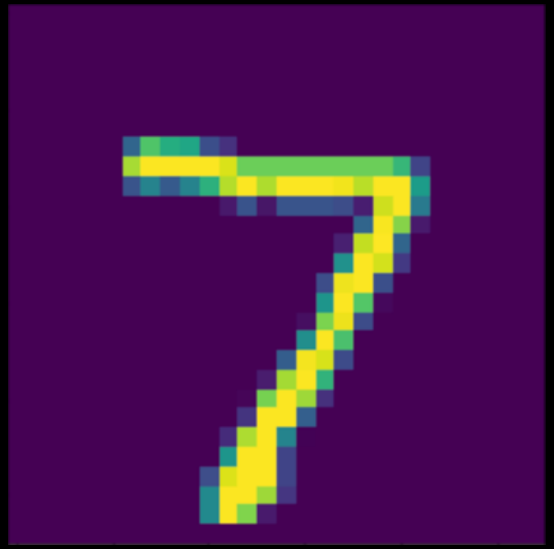


A thing that holds a number between 0 and 1

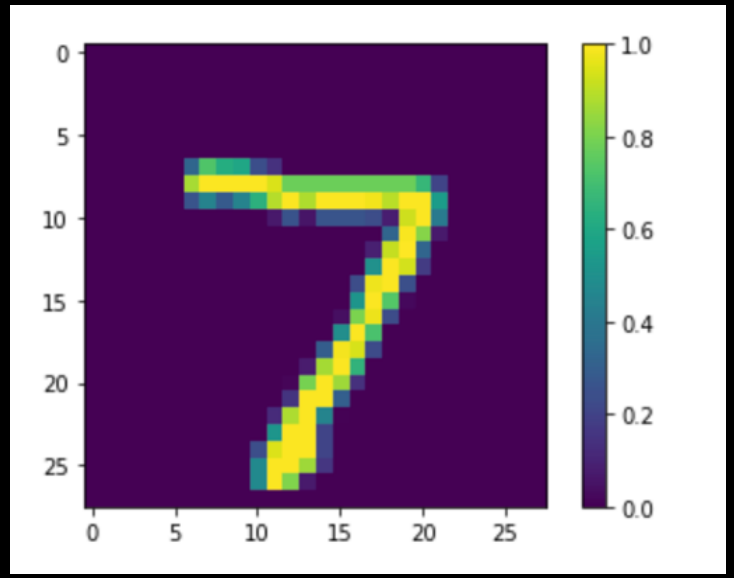
0.85
(for example)

A neuron's "activation" where 0 is inactive and 1 is fully activated

A Layer of Neurons

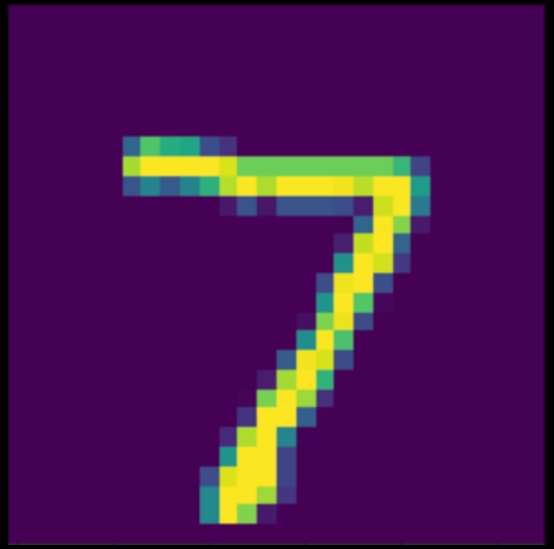


We can represent data (like an image) as a layer of neurons!

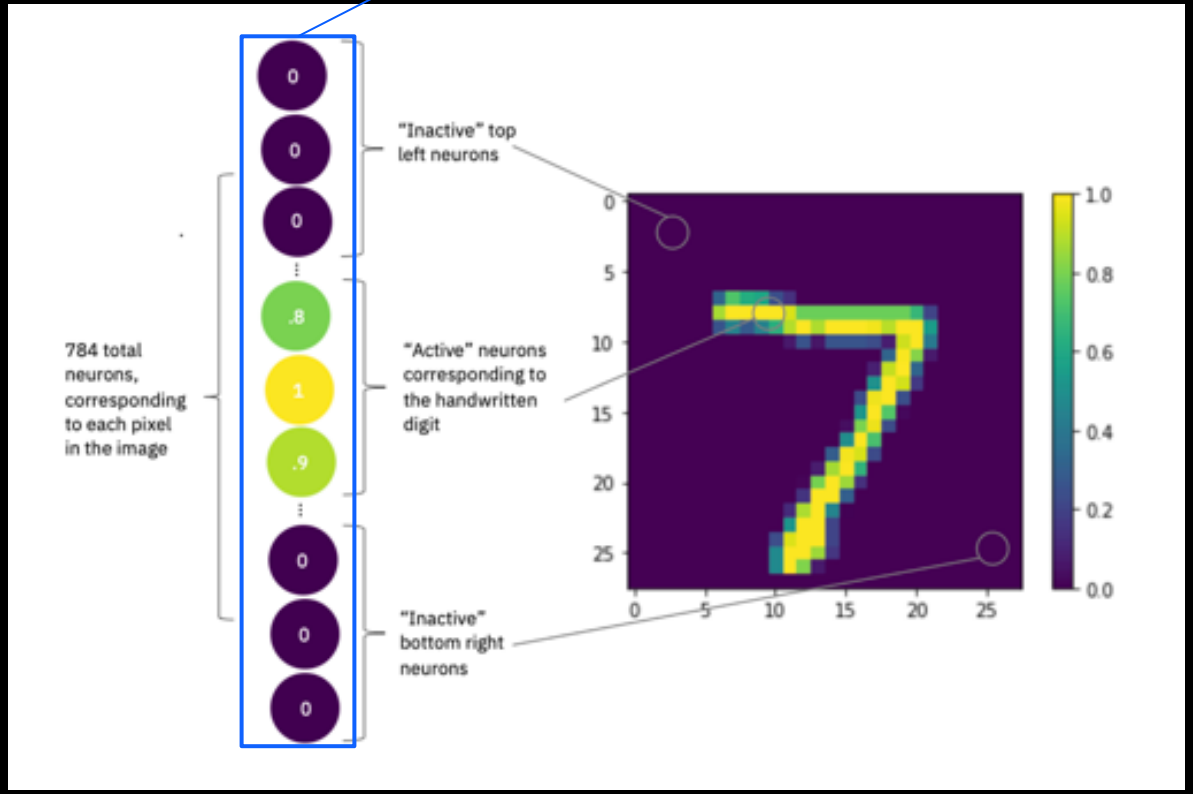


28x28 pixels—784 pixels in total. Pixels on the actual “7” are lit up yellow, corresponding with numbers close to or equal to 1. Pixels that don’t have any “7” on them are equal to 0, meaning they’re not activated at all.

Data as Neurons

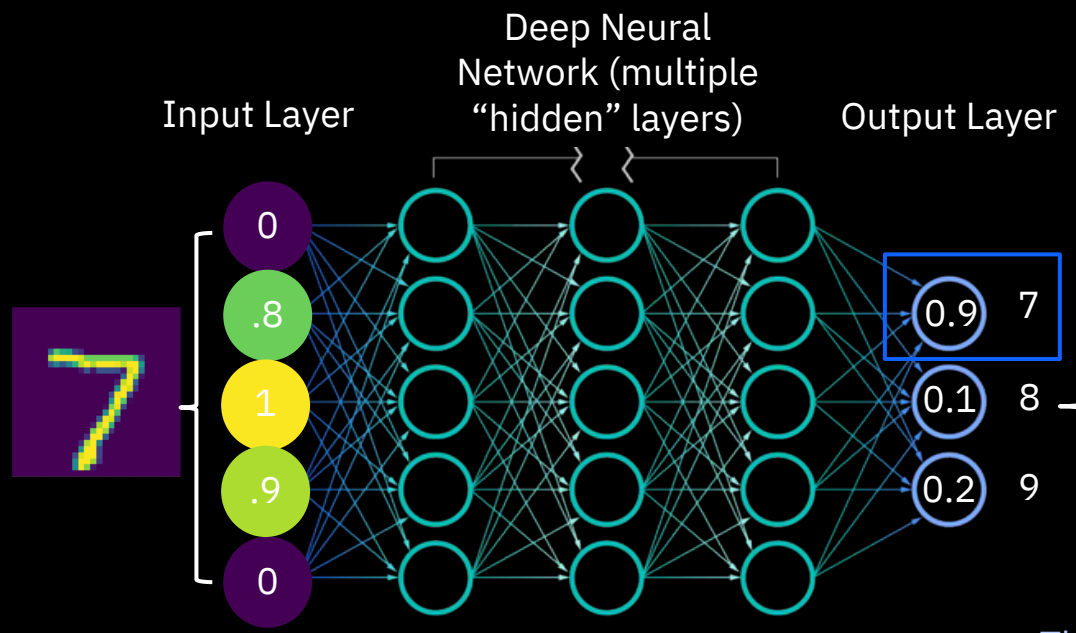


Becomes an input layer of 784 neurons!



Output Layers

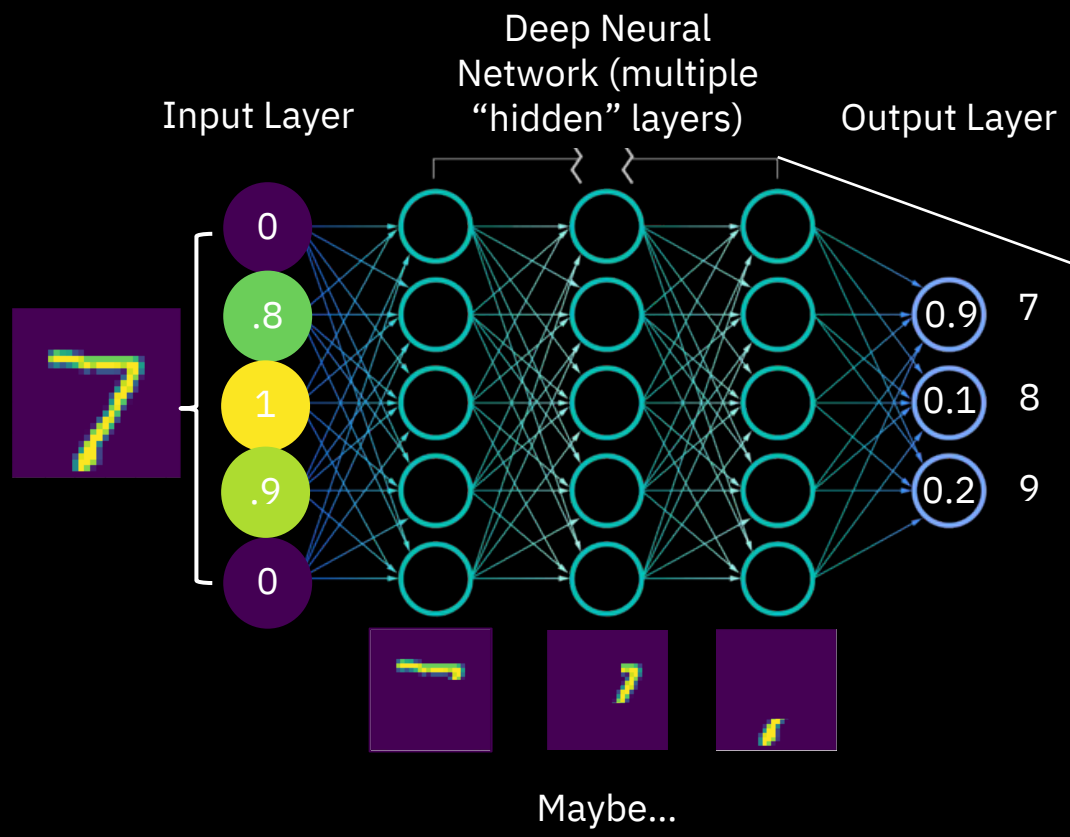
A “mapping” from input (data) to output (objective, what we’re trying to accomplish)



Assume 3 possible classes 7, 8, and 9—corresponds to 3 neurons in our output layer

The output layer neuron with the highest activation (number between 0 and 1) represents our classification!

Hidden Layers



A "mapping" from input (data) to output (objective, what we're trying to accomplish)

Layers of neurons that are neither the input layer (data) nor the output layer (result/outcome)...

Hidden layers try to capture the "latent" (hidden) features in the data...

They are the actual "map" through which the neural network is trying to find an optimal route

(more on this in a moment)

The “Map” Analogy

Beginning

Destination

That’s a cat!

The “architecture” of our neural network (hidden layers, neurons, connections) is the map of possible pathways from beginning to destination

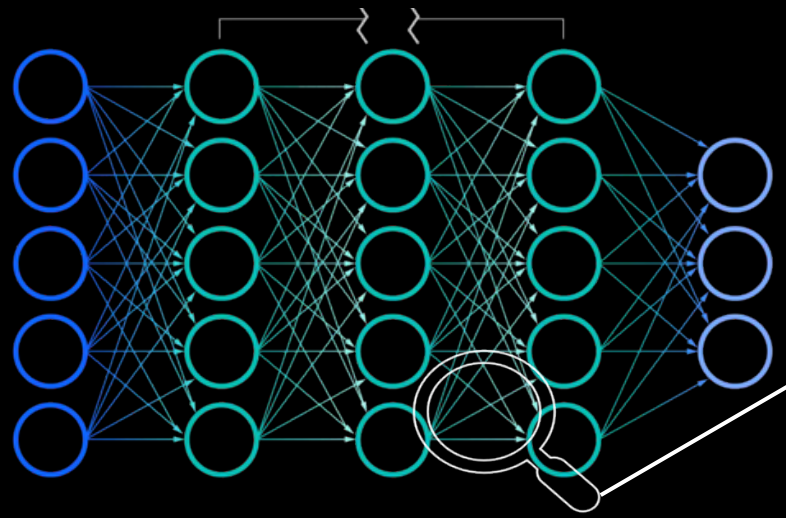
...but how does the neural network know which route is best?

Connecting Neurons

Input Layer

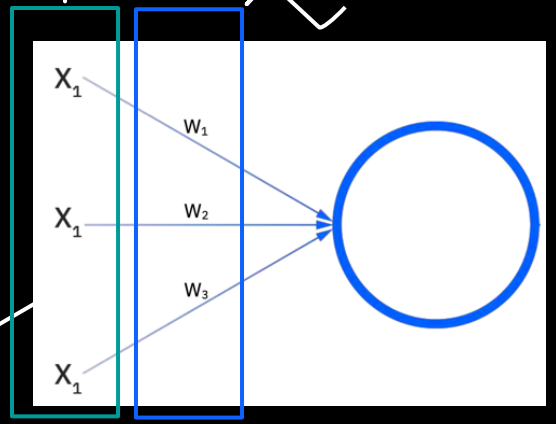
Deep Neural Network (multiple "hidden" layers)

Output Layer



Preceding neuron activations (numbers between 0 and 1)

"Weights" of the connections between neurons; set by the network to guide its path!

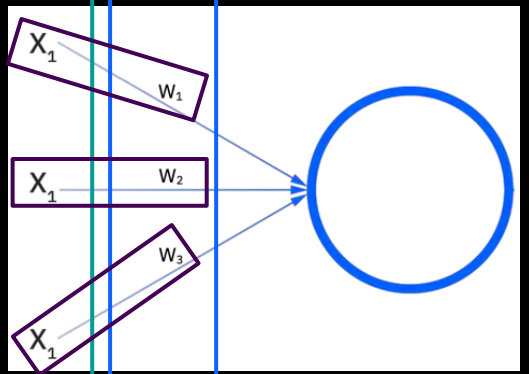


A hidden neuron's value (activation) is determined by the activations of the neurons connected to it (from the previous layer)

Connecting Neurons

Preceding neuron activations (numbers between 0 and 1)

"Weights" of the connections between neurons; set by the network to guide its path!



A hidden neuron's value (activation) is determined by the activations of the neurons connected to it (from the previous layer)

$$= X_1W_1 + X_2W_2 + X_3W_3 + bias$$

Adding together the products of preceding layer neuron activations with assigned weights

Constant to ensure the value is always between 0 and 1

How important the NN thinks a neuron should be when calculating a connected neuron

W ↑ = new activation ↑

W ↓ = new activation ↓

Assigning Weights: Deep “Learning”

Weights direct the actual path through the hidden neurons and layers—“route preferences”

...how does the NN know to set the right weights (find the right route) for its task?

It “learns” to do this with data!

Task: Classify these images

NN: “Let me try to learn a route (weights/biases) on these (training) data to make my mapping” → training with pre-labelled data

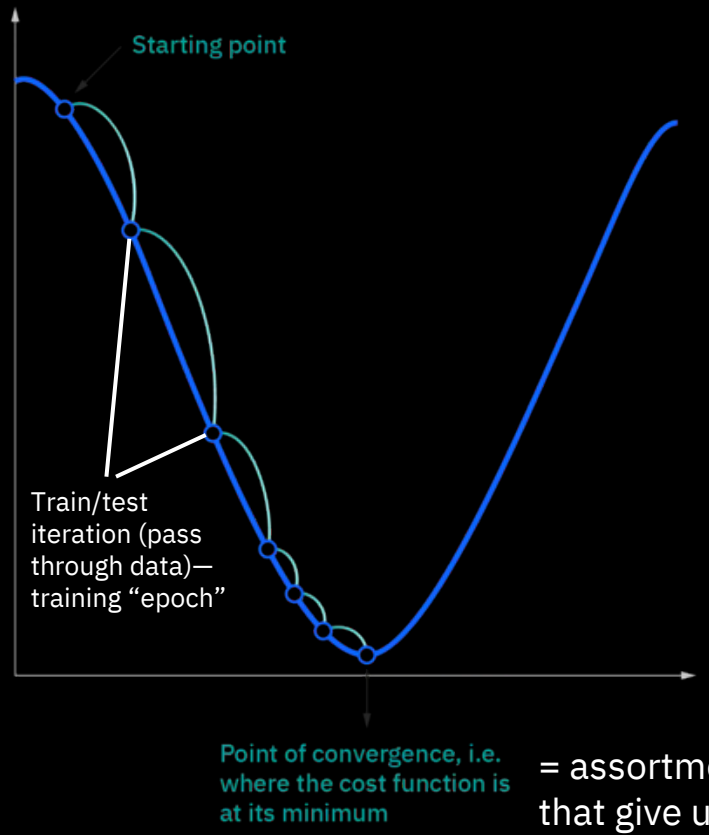
NN: “Let me try my route (weights/biases) on these (test) data and see how well I do” → testing with unlabeled data

NN: “Okay, my overall error rate (loss) was x. Let me see if I can adjust my route (weights) to make it < x in my next try (iteration)” → updating weights/biases to reduce error rate -> Learning!

...repeat many times until error is as low as possible on test data -> Trained Neural Net!

Gradient Descent—how Neural Nets “Learn”

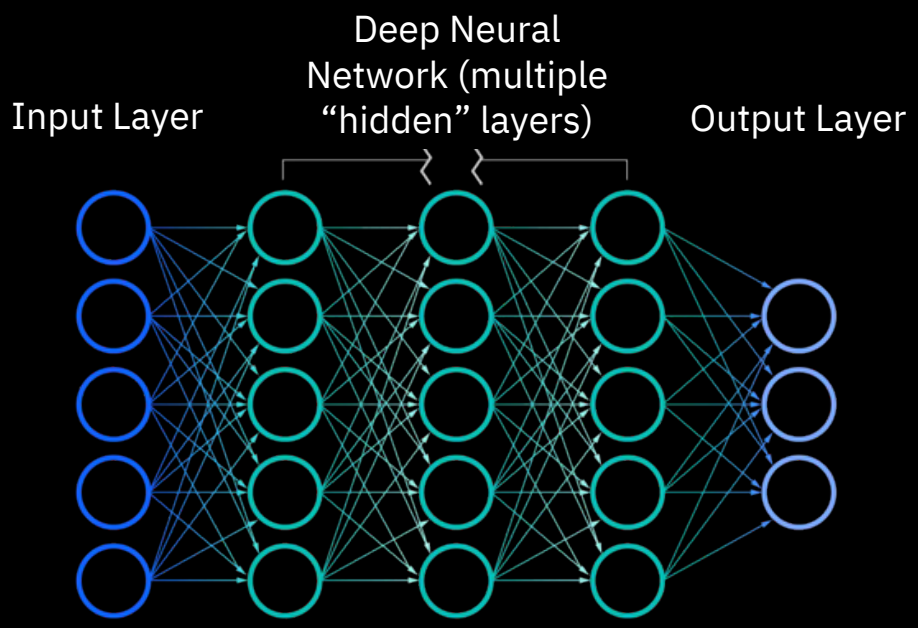
Loss (quantitative measure of errors made by the network in any given testing iteration)



Accomplished with multivariable calculus: Minimize overall loss with respect to many (chained together) weights and biases

= assortment of weights (and biases) that give us the best possible performance with these data!

Neural Nets Key Takeaways



Neural Nets mimic the way that biological brains send information across brain cells (neurons) as part of information processing; consist of layers of artificial neurons and the connections between them

You can think of (trained) neural nets as "mappings" from data (input) to decision/prediction (output)

Neural networks use data to find the parameters (weights and biases) that minimize their overall loss (measure of error rate) through a process called gradient descent

Part IV
Coding Neural Networks (hands-on)

5 Minute Break!

Hands-on Exercise

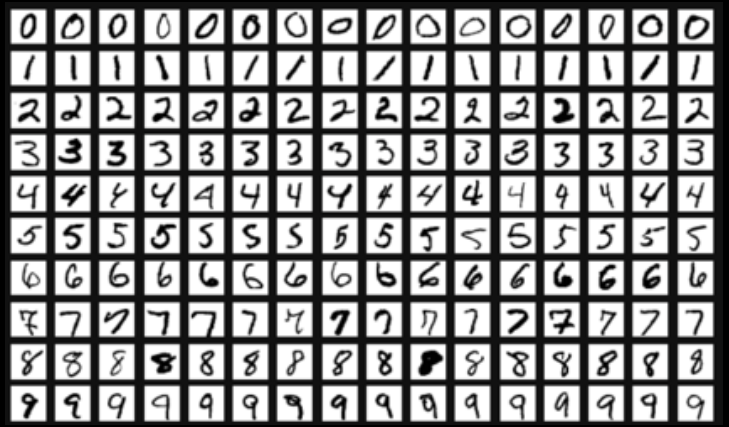
Task: Building a neural network to correctly classify handwritten digits!

For “Coders”: Write your own Python code (with instructions and hints provided)
Join Cedric’s breakout room

THINK 2021 EXERCISE VERSION : For Coders and Non-Coders Break Into Deep Learning with Python and IBM Watson Studio

For “Non-Coders”: Guided follow-along of pre-written code
Join Alex’s breakout room

THINK 2021 SOLUTION VERSION : For Coders and Non-Coders: Break Into Deep Learning with Python and IBM Watson Studio



Recap and Next Steps

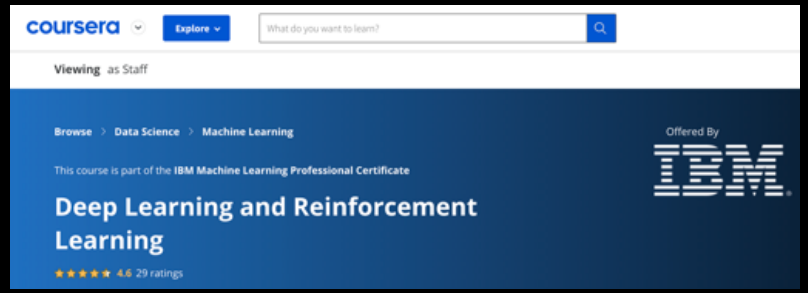
All concepts covered available in the lab guide (2223.pdf) including a glossary

Artificial Intelligence: The design and building of intelligent agents (computer systems) that take information from an environment (data) and use it to take actions that affect that environment

Machine Learning: The study of computer algorithms that improve automatically through experience and by the use of data

Deep Learning: The subfield of Machine Learning focusing on building artificial neural networks to train systems to accomplish complex tasks; The technology underlying many of the most successful AI applications, such as self-driving cars

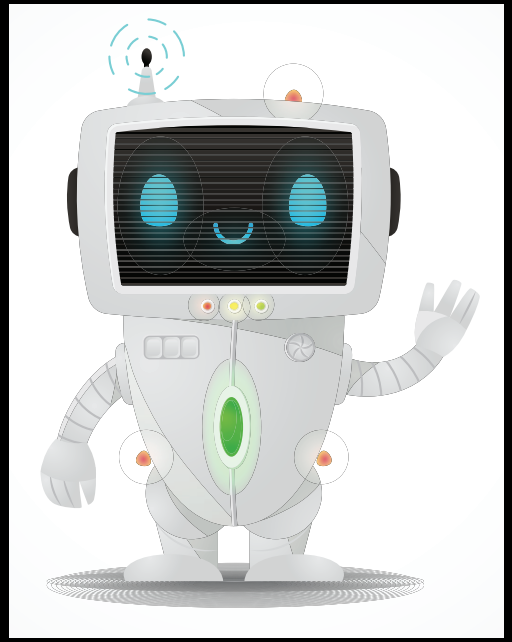
Want to learn more about Machine Learning/Deep Learning? Consider IBM on Coursera!



Thank you!

Alex Amari
Data Scientist
—
alexamari@ibm.com
ibm.com

Cedric Jouan
Data Scientist
—
cedric.jouan@ibm.com
ibm.com



© Copyright IBM Corporation 2021. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at Copyright and trademark information.