# Machine Intelligence Notes

September 12th, 2018
Justin Chen, Rachel Manzelli

## 1. Who are we?

### 1.1 A Brief History of the Machine Intelligence Community

The **Machine Intelligence Community (MIC)** was founded on March 10th, 2016 at the Massachusetts Institute of Technology (MIT) after being inspired by the Google AlphaGo Challenge. MIC was later brought across the river on April 7th, 2017, and the Boston University Machine Intelligence Community was officially founded on September 8th, 2017.

### 1.2 What is MIC?

We are an organization focused on providing opportunities for students to learn about machine intelligence in a community environment. Boston University Machine Intelligence Community is officially sponsored by Boston University's Rafik B. Hariri Institute for Computing and Computational Science and Engineering Software & Application Innovation Lab (SAIL), and BU Spark.

The name Machine Intelligence Community was chosen to emphasize our interest in all aspects of studying and making intelligent machines.

### 1.3 Core Values

BUMIC was founded on three core values: openness, education, and community.

- **Openness** - We are committed to operating transparently and passionately believe that knowledge should be freely accessible and disseminated for all.
- **Education** - Our core mission is to distill and teach advanced machine intelligence concepts, and encourage and excite students to engage in this field.
- **Community** - We aim to introduce students to the greater community and facilitate finding lectures, seminars, workshops, and networks studying machine intelligence.

### 1.4 Goal of workshop series

- Introduce students to fundamental concepts of machine learning and deep learning
- Distill application and theory
- Provide tools so that students can work on projects

MACHINE INTELLIGENCE COMMUNITY

## 2. What is Artificial Intelligence?

Artificial intelligence has many definitions.

- Measured in terms of human intelligence
- Capacity to perform tasks that require what humans would consider intelligent
- **Behave** intelligently
- **Cognitive capacity** for intelligence.

The Modern Approach to Artificial Intelligence textbook offers four different views on artificial intelligence:

1. Thinking humanly - requiring cognitive capacity similar to that of humans
2. Thinking rationally - requiring computational capacity to reason
3. Acting humanly - outwardly behaving similar to humans
4. Acting rationally - outwardly behaving using reason

As software becomes more capable of performing increasingly complex tasks, the definition of artificial intelligence changes and recedes. Programs once considered intelligent become just another app. So then what is artificial intelligence? What is machine intelligence?

*"The essence of intelligence is the ability to predict."*
- Yann LeCun [1]

*"If you had to reduce really intelligence to kind of a single concept, it would be the **ability to predict**. Not just predict what's gonna happen in the world because the world is being the world but what's going to happen in the world as a consequence of your actions. Because that ability allows you to plan to plan ahead and planning - **long-term planning** - is really what requires intelligence. It requires to deal with the **uncertainty** of the prediction. It requires what sequence of actions will make the world or particular part of the world reach a particular goal. And so the ability to form **models of the world**, is an essential piece of intelligence and there is some evidence that a good chunk of our brain is dedicated to forming models of the world."*
- Yann LeCun [1]

## 2.1 What is Machine Learning?

- Machine learning is learning behavior from data
- Term "machine learning" coin by Arthur Samuel in 1959
- *"Machine learning is essentially a form of **applied statistics** with increased emphasis on the use of computers to **statistically estimate complicated functions** and a decreased emphasis on proving confidence intervals around these functions…"* [2]

## 2.2 What is Deep Learning?

- A subfield of machine learning that focuses on learning algorithms modeled after idealized biological neural networks.

**History**

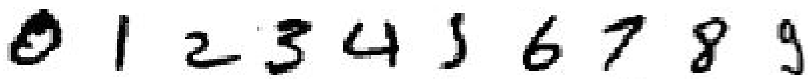
- The study of artificial neural networks started back in **1943** with the seminal paper, ***A Logical Calculus of Ideas Immanent in Nervous Activity*** by Warren S. McCulloch and Walter Pitts (neural networks can simulate any machine - logical modules can compute more complicated functions from stimuli and produce responses). [3]
  - Their perceptron: neurons with a binary threshold activation function were analogous to first order logic sentences (and, or)
  - Very general theory
  - Rosenblatt's perceptron
- Alan Turing defines the Turing test (criteria for an intelligent machine)
- Marvin Minsky - how can I make a more particular machine that can learn specific tasks? Solving problems and creating hypotheses. Perceptrons didn't do it - couldn't count the number of objects in an image, for example (could only predict linearly separable functions - no circular component) (***Perceptrons: An Introduction to Computational Geometry***) in 1969
- Research on NNs halts until the 1980's (symbolic AI)
- Backpropagation is introduced by Geoff Hinton in 1986 in ***Learning Representations by Back-propagating Errors*** **-** a way to adjust weights of the network so that it can minimize a loss function by comparing its output to the desired output
  - In 1998, Yann LeCun applies gradient-based learning to document recognition using the backpropagation technique
  - Resurgence of NNs
- ***Deep Belief Networks*** by Hinton, 2009 (first fully-connected deep learning architecture, a graphical model with many hidden layers)
- This has skyrocketed into deep learning models learning tasks from data for many purposes…
- ImageNet 2012, etc..

- Deep Learning is a combination of calculus, linear algebra, probability, statistics, optimization theory, and learning theory
- Refer to workshop 3 for more details

## 2.3 Things Machines Can Learn: Classification

- **Discrete valued outputs** e.g. class 1, class 2, class 3… where each class is a different category of objects like dog, cat, computer, etc.
- Map an input, which is typically represented as a vector to a class (single integer)
- Classification is important for making decisions, learning about objects, and identifying entities
- Learning algorithms learn to "draw" **decision boundaries/hyperplanes** to discern between different classes
- **MNIST** dataset is a canonical example
  - Consists of handwritten digits 0 through 9

    - Given an image of a symbol, what number/letter is it?
        - Also known as **optical character recognition problem**



- This also works with natural images, text, audio, sensor data, essentially anything that can be **vectorized** - represented as a vector (a list - **order matters** - of numbers)
    - Images are converted to vectors by flattening the images
        - e.g. MNIST images are 28x28 pixels. Vectorization converts them into 784x1 dimensional vectors, meaning it's a point in 784-dimensional space
            - By point, I mean point in the common usage of Euclidean space. Now, imagine 784 dimensions where each dimension can take on values from 0 to 255. :p
            - When you project these points back down into 3 dimensions, you'll see that similar images, here digits, will form clusters together. Neat, right?

## 2.4 Things Machines Can Learn: Regression

- **Real-valued outputs** e.g. the price of a house, cost of grocery, etc.
- Again given a vectorized input, predict its value
- Concretely, given an **input vector/feature vector** - these terms are used interchangeably - representing features of a house, what is the price?

## 2.5 Things Machines Can Learn: Generation

- Learning algorithms can even learn to generate original content - generate data points
- Instead of predicting the output (cost of a house, or object in an image), it could predict the function that generated the output.
- Can use the learned function to generate plausible points
- As an example, after learning from a large corpus of poems, the learning algorithm can learn the distribution of words and grammar to generate poems
- These are called **generative models**
    - A bit more about them in workshop 9

## 2.6 How Does One Learn?

Several types of learning:

1. **Supervised** - dataset consists of both input and correct output (**labeled dataset**)
    a. Expensive to collect labeled datasets
    b. Labels can be noisy if not correctly processed or collected
    c. Majority of techniques in practice today are supervised
    d. Labeled data has been one of the key components that enabled and is currently fueling the deep learning revolution

2. **Semi-supervised** - dataset consists of both **labeled pairs** and **unlabeled pairs**
    a. If you don't have as much labeled data, you could use unlabeled data
    b. Try to learn unsupervised features and then label them with what you know is true from you small amount of labeled data (infer labels on set of unlabeled data)
    c. Could cluster unlabeled points together that looks similar, then compare how far those clusters are from a labeled point, and then label all points in the cluster with the label of the closest labeled point
    d. Can be noisy
3. **Unsupervised** - dataset consists only of **unlabeled data**
    a. The world is abundant with unlabeled data
    b. We haven't yet mastered how to effectively have machines learn from unlabeled data
    c. Humans learn mostly unsupervised thus suggesting it's possible
    d. Most popular unsupervised deep learning techniques:
        - **Generative Adversarial Networks (GANs)**
        - **Autoencoders**
        - **Restricted Boltzmann Machines**
        - Stay tuned for workshop 9
4. **Reinforcement (RL)** - dataset is a simulated or real **environment** (any setting that can dictate the correct **behavior**)
    a. Agent learns by interacting with its environment
    b. Think of Pavlov's dog or Skinner Box from psychology
    c. Agent and environment can be an avatar in a videogame, a robot in the real world, or agent trying to learn how to design another algorithm
    d. Neither supervised nor unsupervised (does not rely on labeled data, but we know the reward up front)
    e. Refer to workshop 7 for more details
5. **Meta Learning** - learning to learn
    a. Using a learning algorithm to design or train another learning algorithm
    b. Learning algorithms turn out to be better at designing learning algorithms than humans are
    c. We will not cover this topic in the workshop series, but feel free to ask if you're interested. This is a super interesting subfield.
    d. We have a research paper summary on Meta Networks, which is a specific meta learning algorithm, on our Github: [https://goo.gl/QEBAwr](https://goo.gl/QEBAwr)
6. **Transfer Learning** - using learned features from a similar dataset to learn something new
    a. Can ameliorate the cost of collecting data and training from scratch
    b. I.e. training on ImageNet, and using the parameters from that model tested on another image dataset that is similar
    c. This will be further covered in workshop 6

## 3. Dataset

- Deep learning algorithms are referred to a **data-driven** also cutely described as **data-hungry**
  - Deep networks typically require very large datasets (on the order of $10^5$ to even $10^6$ samples)
  - After around one million points, deep networks perform increasingly well in practice on, you can make the networks larger and they just perform better - increase in accuracy of prediction.
    - Current theoretical understanding of why this occurs is not yet well defined
- Datasets should be **balanced** so that the algorithm cannot just guess the correct answer
  - Imagine you're doing a **binary classification** task (two classes only) and your dataset is 90% dogs and 10% cats
    - Then 9/10 times, your algorithm can guess correctly during training without having learned anything interesting about what features makes a dog a dog and a cat a cat.
    - When you test your algorithm on new data, it will fail, achieve much less than 90% accuracy

## 3.1 Training, Validation, and Testing

- Dataset should be divided into three parts for training, validation, and testing
  - About 80% of the data should be in the **training set**
  - The remaining 20% should be left for the **validation set** and **test set/ holdout set**
    - Partitions depend on amount of data in total
    - Should never reuse validation or testing data for training
    - Should never have duplicate data points - will cause the learning alg to bias towards those points and similar points
- Training is divided into **epochs** - one epoch iterates over the entire dataset
- Typically use validation set to measure progress of training at the end of each epoch
- Use test set to test on unseen data to measure generalization
- More about training in workshops 3 and 4

## References

1. https://youtu.be/_CJ9aTFcUMs?t=7m6s
2. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
3. McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5.4 (1943): 115-133.