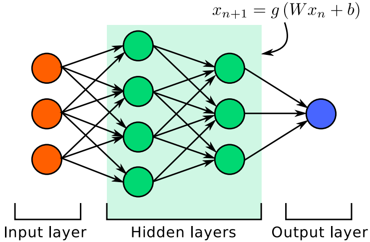
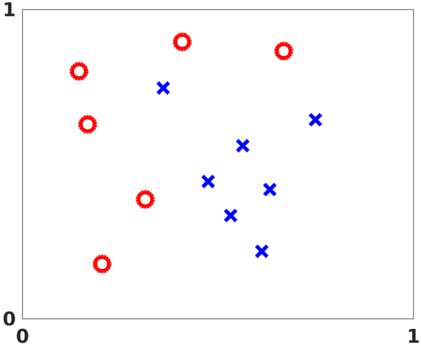
Introduction to Deep Learning



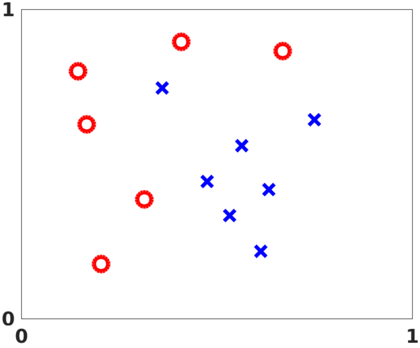
Is it a question?

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



Is it a question?

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



Question? How to classify the rest of points, say *where should we propose a new drilling site for the desired outcome?*

AI via Machine Learning

1. AI via Machine Learning has advanced radically over the past 10 year.

AI via Machine Learning

1. AI via Machine Learning has advanced radically over the past 10 year.
2. ML algorithms now achieve human-level performance or better on the tasks such as

AI via Machine Learning

1. AI via Machine Learning has advanced radically over the past 10 year.
2. ML algorithms now achieve human-level performance or better on the tasks such as
   - face recognition
   - optical character recognition
   - speech recognition
   - object recognition
   - playing the game Go – *in fact, defeated human champions*

AI via Machine Learning

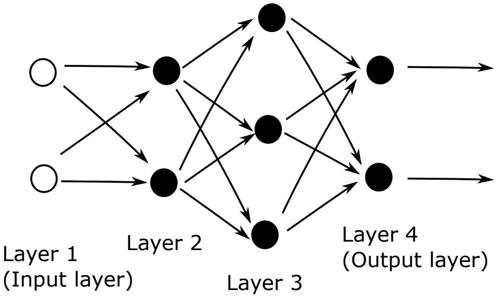
1. AI via Machine Learning has advanced radically over the past 10 year.
2. ML algorithms now achieve human-level performance or better on the tasks such as
   - face recognition
   - optical character recognition
   - speech recognition
   - object recognition
   - playing the game Go – *in fact, defeated human champions*
3. Deep Learning becomes the centerpiece of ML toolbox.

Deep Learning

- Deep Learning = multilayered Artificial Neural Network (ANN).

Deep Learning

- Deep Learning = multilayered Artificial Neural Network (ANN).
- A simple ANN with four layers



Deep Learning

- An ANN *in a mathematically term*

Deep Learning

- An ANN *in a mathematically term*

$$F(x) = \underline{\sigma\left(W^{[4]}\,\underline{\sigma\left(W^{[3]}\,\underline{\sigma(W^{[2]}x + b^{[2]})} + b^{[3]}\right)} + b^{[4]}\right)}$$

Deep Learning

- An ANN *in a mathematically term*

$$F(x) = \underline{\sigma\left(W^{[4]}\,\underline{\sigma\left(W^{[3]}\,\underline{\sigma(W^{[2]}x + b^{[2]})} + b^{[3]}\right)} + b^{[4]}\right)}$$

where

- $p := \{(W^{[2]}, b^{[2]}), (W^{[3]}, b^{[3]}), (W^{[4]}, b^{[4]})\}$ are parameters to be "trained/computed" from *training data*.

- $\sigma(\cdot)$ is an activiation function, say sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Deep Learning

- The objective of training is to "minimize" a properly defined cost function, say

$$\min_p \mathsf{Cost}(p) \equiv \frac{1}{m}\sum_{i=1}^{m} \|F(x^{(i)}) - y^{(i)}\|_2^2,$$

where $\{(x^{(i)}, y^{(i)})\}$ are *training data*

Deep Learning

- The objective of training is to "minimize" a properly defined cost function, say

$$\min_p \mathsf{Cost}(p) \equiv \frac{1}{m} \sum_{i=1}^{m} \|F(x^{(i)}) - y^{(i)}\|_2^2,$$

  where $\{(x^{(i)}, y^{(i)})\}$ are *training data*

- Steepest/gradient descent

$$p \longleftarrow p - \tau \, \nabla \mathsf{Cost}(p)$$

  where τ is known as the *learning rate*.

Deep Learning

- The objective of training is to "minimize" a properly defined cost function, say

$$\min_p \text{Cost}(p) \equiv \frac{1}{m} \sum_{i=1}^{m} \| F(x^{(i)}) - y^{(i)} \|_2^2,$$

  where $\{(x^{(i)}, y^{(i)})\}$ are *training data*

- Steepest/gradient descent

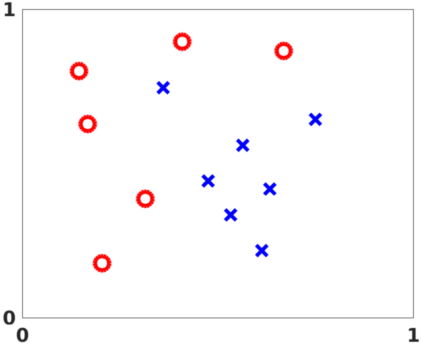
$$p \longleftarrow p - \tau \nabla \text{Cost}(p)$$

  where τ is known as the *learning rate*.

The underlying operations of DL are stunningly simple, *mostly matrix-vector products,* but extremely intense.

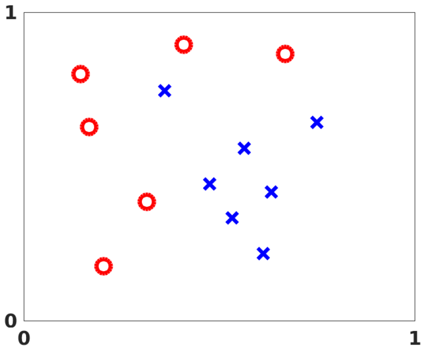
Experiment 1

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



Experiment 1

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



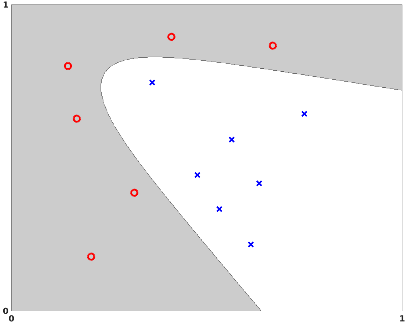
Question for DL: How to classify the rest of points, say *where should we propose a new drilling site for the desired outcome?*

Experiment 1

Classification *after 90 seconds training on my desktop*

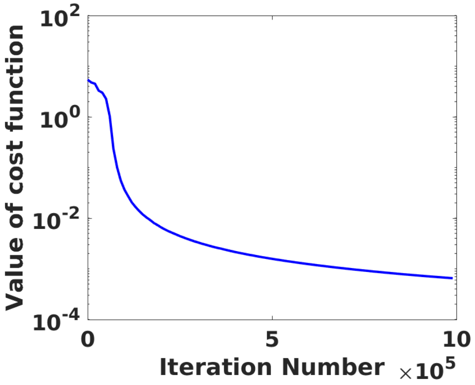
Experiment 1

Classification *after 90 seconds training on my desktop*



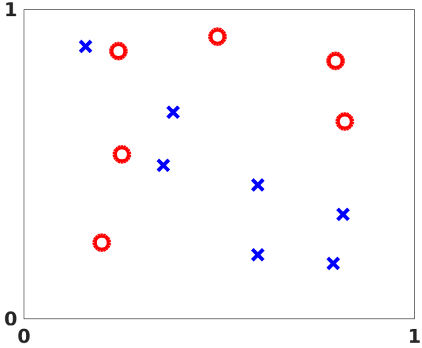
Experiment 1

The value of Cost$(W^{[\cdot]}, b^{[\cdot]})$:



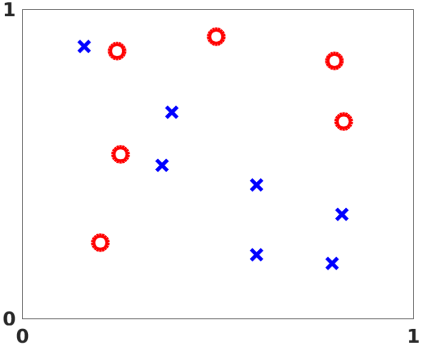
Experiment 2

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



Experiment 2

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



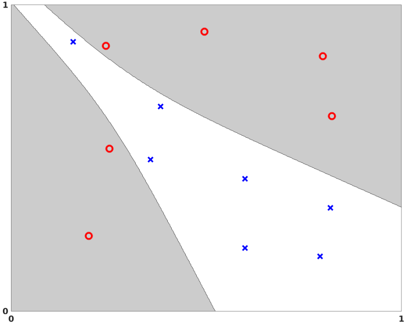
Question for DL: How to classify the rest of points, say *where should we propose a new drilling site for the desired outcome?*

Experiment 2

Classification *after 90 seconds training on my desktop*

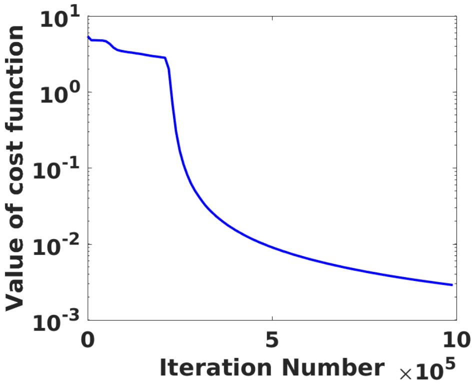
Experiment 2

Classification *after 90 seconds training on my desktop*



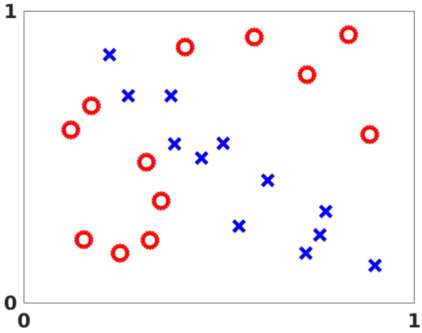
Experiment 2

The value of Cost$(W^{[\cdot]}, b^{[\cdot]})$:



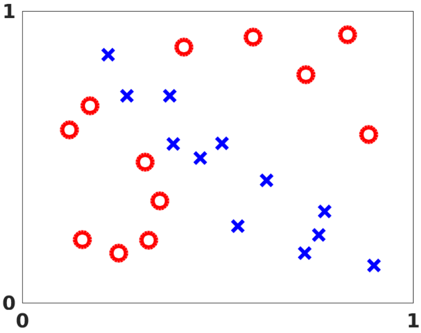
Experiment 3

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



Experiment 3

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



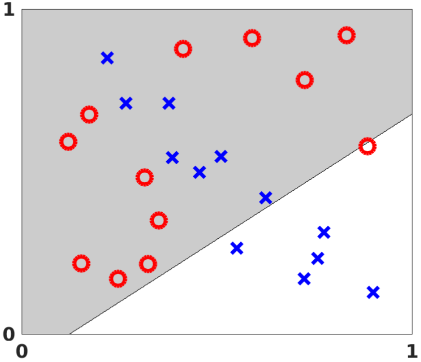
Question for DL: How to classify the rest of points, say *where should we propose a new drilling site for the desired outcome?*

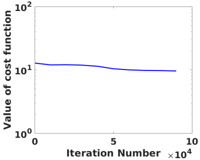
Experiment 3

Classification *after 16 seconds training on my desktop*

Experiment 3

Classification *after 16 seconds training on my desktop*



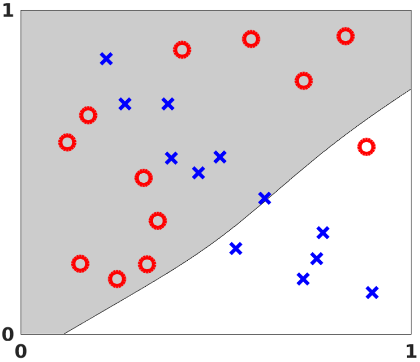


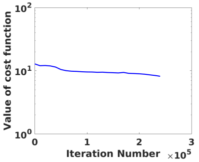
Experiment 3

Classification *after 38 seconds training on my desktop*

Experiment 3

Classification *after 38 seconds training on my desktop*



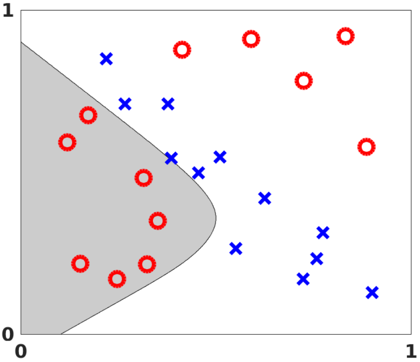


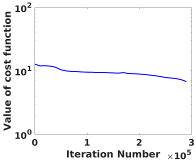
Experiment 3

Classification *after 46 seconds training on my desktop*

Experiment 3

Classification *after 46 seconds training on my desktop*



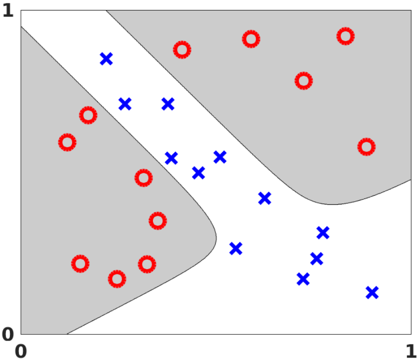


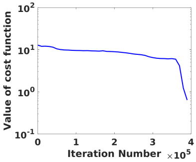
Experiment 3

Classification *after 62 seconds training on my desktop*

Experiment 3

Classification *after 62 seconds training on my desktop*



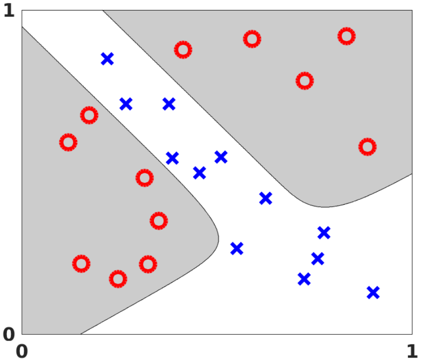


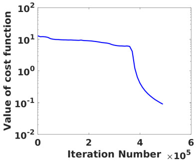
Experiment 3

Classification *after 83 seconds training on my desktop*

Experiment 3

Classification *after 83 seconds training on my desktop*



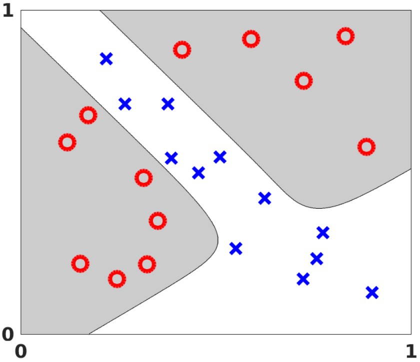


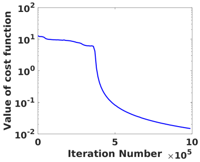
Experiment 3

Classification *after 156 seconds training on my desktop*

Experiment 3

Classification *after 156 seconds training on my desktop*

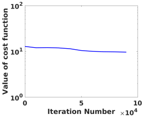




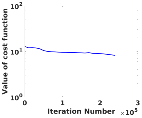
Experiment 3

The value of Cost$(W^{[\cdot]}, b^{[\cdot]})$:

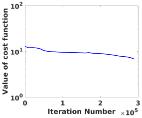
16



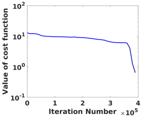
38



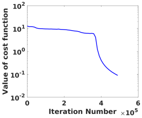
46



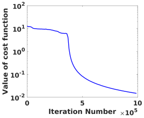
62



83

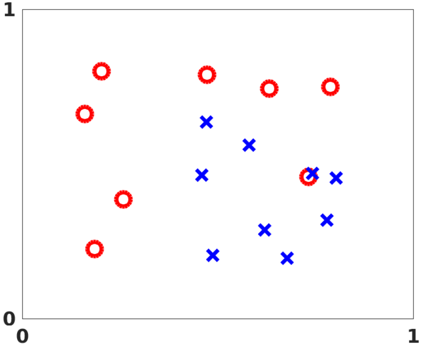


156



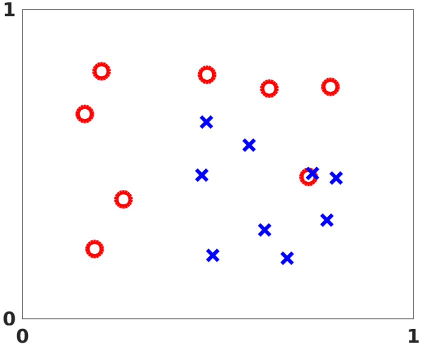
Experiment 4

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



Experiment 4

Given training data with categories A ($\circ$) and B ($\times$), say *well drilling sites with different outcomes*



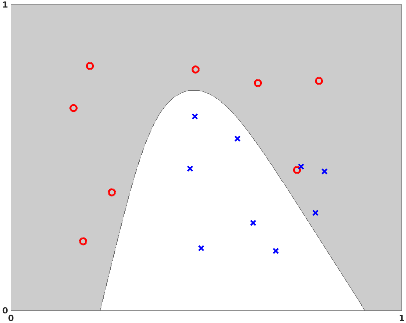
Question for DL: How to classify the rest of points, say *where should we propose a new drilling site for the desired outcome?*

Experiment 4

Classification *after 90 seconds training on my desktop*

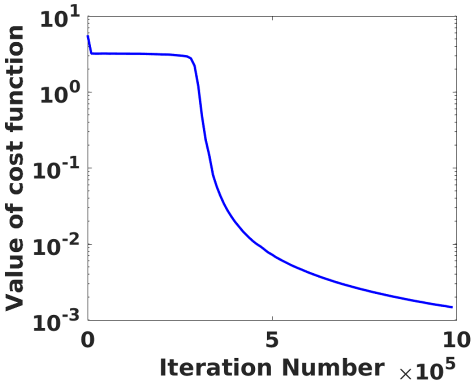
Experiment 4

Classification *after 90 seconds training on my desktop*



Experiment 4

The value of Cost$(W^{[\cdot]}, b^{[\cdot]})$:



"Perfect Storm"

1. The recent success of ANNs in ML, despite their long history, can be contributed to a "perfect storm" of

“Perfect Storm”

1. The recent success of ANNs in ML, despite their long history, can be contributed to a “perfect storm” of
   - large labeled datasets;
   - improved hardware;
   - clever parameter constraints;
   - advancements in optimization algorithms;
   - more open sharing of stable, reliable code leveraging the latest in methods.

"Perfect Storm"

1. The recent success of ANNs in ML, despite their long history, can be contributed to a "perfect storm" of
   - large labeled datasets;
   - improved hardware;
   - clever parameter constraints;
   - advancements in optimization algorithms;
   - more open sharing of stable, reliable code leveraging the latest in methods.
2. ANN is simultaneously one of the simplest and most complex methods:

“Perfect Storm”

1. The recent success of ANNs in ML, despite their long history, can be contributed to a “perfect storm” of
   - large labeled datasets;
   - improved hardware;
   - clever parameter constraints;
   - advancements in optimization algorithms;
   - more open sharing of stable, reliable code leveraging the latest in methods.
2. ANN is simultaneously one of the simplest and most complex methods:
   - learning to model and parameterization
   - capable of self-enhancement
   - generic computation architecture
   - executable on local HPC and on cloud
   - broadly applicable but requires good understanding of the underlying problems and algorthms