L1 - Intro

2017/11/01 09:24

- Questions
  - what is ML
  - which kinds of problems requires ML

- KEY
  - ML
    - data + learning algorithm = ML
    - tasks best solved by learning
      - pattern recognition
      - detect anomalies
      - prediction
    - fruit flies
      - MNIST hand-written digits
      - ImageNet task
      - Speech Recognition Task
  - NN
    - why NN
      - parallel computation + adaptive connections = powerful to solve tasks that human good at.
    - Model of Neurons - Idealized Neuron
      - Linear Neuron
      - Binary threshold Neuron
      - rectified linear Neuron
      - Sigmoid Neuron - Logistic function
      - Stochastic binary neuron
      - Hyperbolic tangent neuron
  - An example

            - A two layer network with a single winner is equivalent to having a rigid template for each shape. The winner is the template that has the biggest overlap with the ink
            - weights work as:
                - A pixel gets to vote if it has ink on it. Each inked pixel can vote for several different shapes
                - The shape that gets the most votes wins.
    - Three types of learning
        - Supervised learning
            - To Learn to predict an output when given an input vector.
            - subtypes
                - Regression
                - Classification
            - How supervised learning typically works
                - first, choose a model-class: y = f(X; W)
                    - A model-class, f, is a way of using some numerical parameters, W, to map each input vector, x, into a predicted output y

- **second,** Learning usually means adjusting the parameters to reduce the discrepancy between the target output t, and the true label y.

- Reinforcement learning -- not quite understand
  - Learn to select an action to maximize payoff
  - In reinforcement learning, the output is an action or sequence of actions and the only supervisory signal is an occasional scalar reward.
    - The goal in selecting each action is to maximize the expected sum of the future rewards
    - We usually use a **discount factor** for delayed rewards so that we don’t have to look too far into the future
  - Reinforcement learning is difficult
    - The rewards are typically delayed so hard to localize the wrong
    - A scalar reward does not supply

much information.

- So typically, only <1000 parameters possible

- Unsupervised learning
    - To Discover a good internal representation of the input.
    - Applications
        - **create an internal representation of the input** for **subsequent supervised or reinforcement learning**.
        - It provides a compact, low-dimensional representation of the input.
            - **Reduce dim**, e.g. PCA
        - It provides **an economical high-dimensional representation** of the input in terms of learned features.
            - Binary features are economical.
            - So are real-valued features that are nearly all zero.

- clustering

  -

- Why do we need machine learning?
  - What is Machine Learning?
    - cases can't be solved by normal algorithms
      - some problems need complicated programs with unreliable rules
      - for some, It is very hard to write programs that solve problems like recognizing a three-dimensional object
        - We don't know what program to write because we don't know how its done in our brain
        - Even if we had a good idea about how to do it, the program might be horrendously complicated
    - The Machine Learning Approach
      - Instead of writing a program by hand for each specific task, we collect lots of examples that specify the correct output for a given input.
      - A machine learning algorithm then takes these examples and produces a program that does the job.
        - The program is different from a typical hand-written program. It may contain millions of numbers.
        - the program works for new cases as

well as the ones we trained it on.
        - If the data changes the program can change too by training on the new data.
    - Massive amounts of computation are now cheaper than paying someone to write a task-specific program.
- Some examples of tasks best solved by learning
    - Recognizing patterns:
        - Objects in real scenes
        - Facial identities or facial expressions
        - Spoken words
    - Recognizing anomalies:
        - Unusual sequences of credit card transactions
        - Unusual patterns of sensor readings in a nuclear power plant
    - Prediction:
        - Future stock prices or currency exchange rates
        - Which movies will a person like?
- A standard example of machine learning used for explanations
    - A lot of genetics is done on fruit flies. because they breed fast and We already know a lot about them.
    - The MNIST database of hand-written digits is

the machine learning equivalent of fruit flies

  - publicly available, quite fast in a moderate-sized neural net
  - We know a huge amount about how well various machine learning methods do on MNIST

- The ImageNet task
  - 1000 different object classes in 1.3 million high-resolution training images from the web.
    - Best system in 2010 competition got 47% error for its first choice and 25% error for its top 5 choices
    - A very deep neural net (Krizhevsky et. al. 2012) gets less than 40% error for its first choice and less than 20% for its top 5 choices
  - Jitendra Malik (an eminent neural net sceptic) said that this competition is a good test of whether deep neural networks work well for object recognition.
- The Speech Recognition Task

- A speech recognition system has several stages
    - Pre-processing: Convert the sound wave into a vector of acoustic coefficients. Extract a new vector about every 10 mille seconds.
    - The acoustic model: Use a few adjacent vectors of acoustic coefficients to place bets on which part of which phoneme [音素] is being spoken.
    - Decoding: Find the sequence of bets that does the best job of fitting the acoustic data and also fitting a model of the kinds of things people say.
- Deep neural networks pioneered by George Dahl and Abdel-rahman Mohamed are now replacing the previous machine learning method for the acoustic model.

- bi-phone + deep net with 8 layers, 20.7% error
- the best previous result is 24.4%

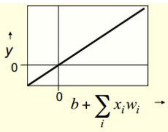
- Google does best in the field of speech recognition

- What are neural networks?
  - Reasons to study neural computation
    - To understand how the brain actually works. use computer simulations.
    - To understand a style of parallel computation inspired by neurons and their adaptive connections
      - should be good for things that brains are good at (e.g. vision)
      - Should be bad for things that brains are bad at (e.g. 23 x 71)
    - To solve practical problems by using novel learning algorithms inspired by the brain (this course)
  - Neural Network
    - positive input + positive weight => excite the other neurons
      - others....
    - the synaptic weights can adapt so that brains learn things
    - The structure of the brain gives rapid parallel computation plus flexibility.

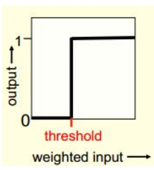
- Some simple models of neurons
  - Idealized neurons
    - To model things we have to idealize them
      - Idealization removes inessential complicated details

            - It allows us to apply mathematics and to make analogies to other, familiar systems
            - add complexity to make the model more faithful gradually
        - It is often worth understanding models that are known to be wrong
            - E.g. neurons that communicate real values rather than discrete spikes of activity.
    - Current types
        - Linear neurons
            - These are simple but computationally limited
            - If we can make them learn we may get insight into more complicated neurons

$$y = b + \sum_i x_i w_i$$



        - Binary threshold neurons
            - First compute a weighted sum of the inputs.
            - Then send out a fixed size spike of activity if the weighted sum exceeds a threshold.



$$z = \sum_i x_i w_i \qquad y = \begin{cases} 1 \text{ if } z \geq \theta \\ 0 \text{ otherwise} \end{cases}$$

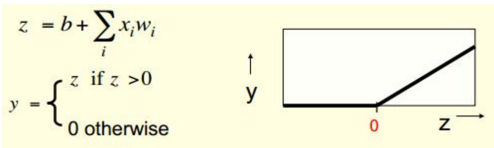
$$\boxed{\theta = -b}$$

$$z = b + \sum_i x_i w_i \qquad y = \begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

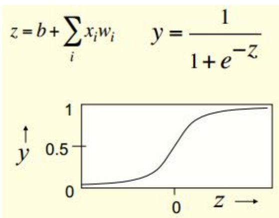
        - Rectified Linear Neurons or called linear threshold neurons
            - They compute a linear weighted sum of their inputs
            - The output is a non-linear function of the total input.
            - property

- give linear character when > 0
- allow to make decision at 0

- It is more effective than logistic sigmoid and hyperbolic tangent when applied in convolutional networks.
- a unit employing the rectifier is also called a rectified linear unit (ReLU)
- a smooth approximation to the rectifier is the analytic function
  - f(z) = ln(1 + e^z) - also called softplus function
  - it's derivative is the logistic function



- Sigmoid neurons - most used
  - These give a real-valued output that is a smooth and bounded function of their total input.
  - Typically they use the logistic function
  - They have nice derivatives which make learning easy (see lecture 3).



- Stochastic binary neurons
  - These use the same equations as logistic units. But they treat the output of the logistic as the probability of producing a spike.
  - We can do a similar trick for rectified linear

units:

- The output is treated as the Poisson rate for spikes.
  - It's a Poisson process. the rectified linear unit determines the rate of producing spikes, but intrinsic randomness in the unit determines when the spikes are actually produced.
  - possion rate 泊松比，横向变性系数

- tanh function -Hyperbolic tangent function
  - y = tanh(z), range (-1, 1)

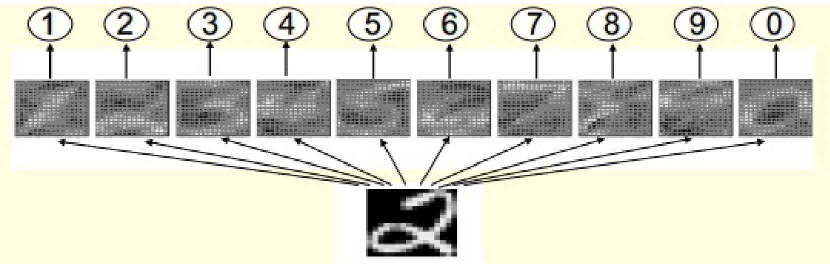
Hyperbolic tangent:

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = \frac{1 - e^{-2x}}{1 + e^{-2x}}.$$

- A simple example of learning
  - a two-layer NN to handwritten recognition
  - Mechanism
    - A pixel gets to vote if it has ink on it. Each inked pixel can vote for several different shapes
    - The shape that gets the most votes wins.
  - How to display the weights
    - Give each output unit its own "map" of the input image and display the weight coming from each pixel in the location of that pixel in the map.
    - Use a black or white block with the area representing the **magnitude** of the weight and the

color representing the sign.

- How to learn the weights
    - Show the network an image and increment the weights from active pixels to the correct class.
    - Then decrement the weights from all active pixels to avoid that the weight get too bigger and **every class will get huge input whenever we show it to the image.**
- The learned weights
    - look very similar to the digits. White shows positive votes. The weights forms a templet for each correct digit.



- Why the simple learning algorithm is insufficient
    - A two layer network with a single winner is equivalent to having a rigid template for each shape. The winner is the template that has the biggest overlap with the ink
    - The variation of hand-written digits is too complicated to be captured by simple template.

- Three types of learning
    - Supervised learning
        - goal
            - To Learn to predict an output when given an input vector.
        - subtypes

- Regression
  - real-number output
- Classification
  - discrete output - class label.

▪ How supervised learning typically works

- first, choose a model-class: y = f(X; W)
  - A model-class, f, is a way of using some numerical parameters, W, to map each input vector, x, into a predicted output y
- **second,** Learning usually means adjusting the parameters to reduce the discrepancy between the target output t, and the true label y.

◦ Reinforcement learning

▪ goal

- Learn to select an action to maximize payoff

▪ In reinforcement learning, the output is an action or sequence of actions and the only supervisory signal is an occasional scalar reward.

- The goal in selecting each action is to maximize the expected sum of the future rewards
- We usually use a **discount factor** for delayed rewards so that we don’ t have to look too far into the future

▪ Reinforcement learning is difficult

- The rewards are typically delayed so its hard to know where we went wrong (or right).
- A scalar reward does not supply much information.
- So typically, you can't learn millions of parameters using reinforcement learning. dozens of parameters or maybe 1,000 parameters, but not millions.

◦ Unsupervised learning

- Goal
    - Discover a good internal representation of the input.
- It is hard to say what the aim of unsupervised learning is. Some are listed below
    - One major aim is to **create an internal representation of the input** that is useful for **subsequent supervised or reinforcement learning.**
    - It provides a compact, low-dimensional representation of the input.
        - High-dimensional inputs typically live on or near a lowdimensional manifold (or several such manifolds).
            - a image with 1 million pixel may only have a few hundred degree of freedom in what can happen.
        - Principal Component Analysis is a widely used linear method for finding a low-dimensional representation.
            - one manifold, which is a plane in a high dimensional space
    - It provides an economical high-dimensional representation of the input in terms of learned features.
        - Binary features are economical.
        - So are real-valued features that are

nearly all zero.

- It finds sensible clusters in the input.
    - clustering is really just an extreme case of finding sparse features.