Future Challenges

Yann LeCun

The Courant Institute of Mathematical Sciences

New York University

http://yann.lecun.com

The Challenges of Machine Learning

- **How can we use learning to progress towards AI?**
  - Can we find learning methods that scale?
  - Can we find learning methods that solve **really complex problems** end-to-end, such as vision, natural language, speech....?

- **How can we learn the structure of the world?**
  - How can we build/learn internal representations of the world that allow us to discover its hidden structure?
  - How can we learn internal representations that capture the relevant information and eliminates irrelevant variabilities?

- **How can a human or a machine learn internal representations by just looking at the world?**

The Next Frontier in Machine Learning: Learning Representations

- **The big success of ML has been to learn classifiers from labeled data**
  - The representation of the input, and the metric to compare them are assumed to be "intelligently designed."
  - Example: Support Vector Machines require a good input representation, and a good kernel function.

- **The next frontier is to "learn the features"**
  - The question: how can a machine learn good internal representations
  - In language, good representations are paramount.
    - What makes the words "cat" and "dog" semantically similar?
    - How can different sentences with the same meaning be mapped to the same internal representation?

- **How can we leverage unlabeled data (which is plentiful)?**

What is Intelligence?

- **Most wetware cycles in higher animals are devoted to perception, and most of the rest to motor control.**
  - 20% of our brain does vision
- **Intelligence includes the ability to derive complex behavior from massive amounts of sensory information.**
  - Intelligence requires making sense complex sensor input (including proprioception).
- **Intelligence is modeling, prediction, and evaluation**
  - The more intelligent the organism, the better it can predict the world, predict the consequences of its actions (including rewards), and pick the “best” action.
- **How can an intelligent agent learn to predict the world?**
  - Not the whole world, only the part of the world which is relevant to the purpose of its existence, including its own actions.

What is Intelligence?

- **In the past we thought that intelligence was what smart humans could do**
  - Speech, logical reasoning, playing chess, computing integrals.....
- **But those things turned out to be pretty simple computationally**
  - It turns out that the complicated things are perception, intuition and common sense
  - Things that even a mouse can do much better than any existing robot.
- **What about rat-level intelligence?**
  - We will achieve rat-level intelligence before we achieve human-level, so let's work on rat-leve intelligence
- **It is likely that intelligent agents cannot be “intelligently designed” ;-)**
  - They have to build themselves through learning (or evolution)
  - How much prior structure is required?
  - The more intelligent, the less prior structure.

Where are we now?

- **We are still quite far from rat-level intelligence**
  - We don't really have robots that can run around without bumping into things or falling into ditches using vision only.
  - We know that the methods currently being developed are hacks.
- **Most NIPS/ICML papers do not take us any closer to rat-level intelligence**
  - In fact, some of them take us backward.
- **What problems should we work on, what type of new methods should we develop?**
- **Which methods currently being worked on the community, while being valuable, will not take us to rat-level intelligence?**
  - How do we avoid the trap of building ever so taller ladders when our goal is to reach the Moon?

Questions?

- **Is there a magic bullet?**
  - Is there a general principle for learning/AI, or is it just a bunch of tricks? (see Gary Marcus's book "Kluge").
  - Is there a universal learning algorithm/architecture which, given a small amount of appropriate prior structure, can produce intelligent agents?
  - Or do we need to accumulate a large repertoire of "modules" to solve each specific problem an intelligent agent must solve. How would we assemble those modules?
- **Let's face it, our only working example uses neurons.**
  - Does that mean rat-level intelligence will be achieved with simulated neurons?
  - Airplanes don't flap their wings
  - Yes, but they exploit the same aerodynamical properties as birds (and they are not as efficient at it)
  - What is the analog of aerodynamics for intelligence?

Questions?

- **Every reasonable learning algorithm we know minimizes some sort of loss function?**
  - Does the brain minimize a loss function?
  - If yes, what is it, and how does it do it?
  - Is there any other way to build learning systems?
- **If we accept the loss function hypothesis:**
  - Can the loss function possibly be convex?
  - How is it parameterized?
    - What is the architecture?
  - How is it minimized
    - direct solution (like quadratic problems),
    - gradient-based iterative procedures
    - perturbation/trial and error
    - all of the above?

What Do We Need?

- **Learning**
  - Supervised, unsupervised, reinforcement
- **Efficient reasoning with large numbers of variables**
  - e.g. For image segmentation/labeling...
- **Very fast inference for high-dimensional complex tasks**
  - People and animal can recognize common visual categories in less than 100ms. There is no time for complex reasoning/relaxation/inference.
- **Emotions**
  - Emotion, in the restricted sense of reward prediction.

What Do We Need?

- **Learning algorithms that scale**
  - Sub-linearly with the number of training samples
  - Not much more than linearly with the size of the learning system
  - Non-exponentially with the size of the action space and state space
- **“Deep” Learning**
  - Intelligent inference requires lots of elementary decisions (non-linear steps):
    - pixels->low-level feature->high-level features->categories
    - Global goal->macro-action sequence->action sequence->motor commands
- **A framework with which to build large-scale “deep” learning machines with millions of parameters**
  - A framework that allows us to specify prior knowledge
- **How little prior knowledge can we get away with?**

What classes of methods will NOT take us there?

- **Template matching**
  - Fast nearest-neighbor methods, kernel methods, and other “glorified template matching methods” are very useful, but they won't take us there.
- **“Shallow” learning**
  - Linear combinations of **fixed** basis functions won't take us there
  - Examples: SVM, generalized linear models, boosting with simple weak learners, Gaussian processes.....Those methods are useful, just not for our purpose
  - Shallow architectures are inefficient: most complex functions are more efficiently implemented with many layers of non-linear decisions
  - We need “hierarchical” models that learn high-level representations from low-level perceptions, features, macro features.....

What classes of methods will NOT take us there?

- **Convex Optimization with a practical number of variables**
  - If intelligence could be reduced to convex optimization, the order in which we learn things would not matter: we would go to the same minimum no matter what.
- **Purely generative methods, purely supervised methods, purely unsupervised methods.**
- **Fully probabilistic methods**
  - Because we can't normalize complicated distributions in high dimension
- **Purely discriminative methods**
  - Because not everything comes down to classification
  - We need to model the world

What problems should we solve?

Integrating reasoning with learning

- Graphical models (factor graphs in particular) are a good avenue, but we need to free ourselves from the "partition function problem"
- How do we build non-probabilistic factor graphs that can be trained in supervised, unsupervised, and reinforcement mode.

Invariant Representations

- How do we learn complex invariances in vision?
- We can pre-engineer some of it, but ultimately, we need a scheme to learn features and invariant representation automatically.
- Optimization algorithms (learning) will become better at this than human engineer. It is already the case for handwriting recognition systems and speech recognition systems.

What problems should we solve?

Deep learning

- Ultimately, we need to think again about learning in deep structures with many layers on non-linear decisions.
- We have partial solutions that are not entirely satisfactory
- Pure back-prop can't handle more than a few layers and is very inefficient for unsupervised learning
- Probabilistic belief nets have some of the right ingredients (e.g. Boltzmann machines-like algorithms), but they are plagued by the "partition function problem": learning involves estimating intractable integrals.

What Project should we work on?

Vision

- Generic object recognition, object detection, and such are some of the most challenging perceptual tasks for learning
- We can make progress with clever as-hoc preprocessing combined with simple (linear) learning methods or generative models, but ultimately, good performance will be achieved when we come up with efficient "deep" learning algorithms that can learn the whole task **end-to-end** (with the minimum amount of prior knowledge)

Robotics

- The best way to integrate perception, action, and reinforcement.

The visual system is “deep” and learned

The primate's visual system is “deep”

- It has 10-20 layers of neurons from the retina to the infero-temporal cortex (where object categories are encoded).
- How does it train itself by just looking at the world?.

Is there a magic bullet for visual learning?

- The neo-cortex is pretty much the same all over
- The “learning algorithm” it implements is not specific to a modality (what works for vision works for audition)
- There is evidence that everything is learned, down to low-level feature detectors in V1
- Is there a universal learning algorithm/architecture which, given a small amount of appropriate prior structure, can produce an intelligent vision system?
- Or do we have to keep accumulating a large repertoire of pre-engineered “modules” to solve every specific problem an intelligent vision system must solve?

Can we learn everything end-to-end?

- **There are simple situations where we have been able to demonstrate end-to-end learning in vision.**
  - Learning to recognize handwritten words from pixels to labels with no preprocessing and minimal prior knowledge (NIPS 1990-1998)
  - Learning to detect faces (NIPS 2004)
  - Learning to detect and recognize generic object categories from raw pixels to labels (CVPR 2003)
  - Training a robot to avoid obstacles from raw left/right image pairs to steering angles (NIPS 2005)

Two Big Problems in Machine Learning and Computer Vision

2. The "Deep Learning Problem"

- Training "Deep Belief Networks" is a necessary step towards solving the invariance problem in vision (and perception in general).
- How do we train deep architectures with lots of non-linear stages?

1. The "Intractable Partition Function Problem"

- Give high probability (or low energy) to good answers
- Give low probability (or high energy) to bad answers
- There are too many bad answers!
- The normalization constant of probabilistic models is a sum over too many terms.