Deep Learning

Slides Courtesy of Chuck Dyer

• Early NNs used only 1 or 2 hidden layers because training was slow, thought to converge to poor local minima, and overfit

• Recent results suggest that local minima are *not* a serious problem – regardless of initial conditions, bigger NNs contain many more local minima but nearly all are solutions of very similar quality. And overfitting can be avoided using regularization techniques rather than using fewer units

Deep Learning

• Use *many hidden layers* (often 10-20)
• Called **Deep Neural Nets** or **Deep Nets**

Why Deep Learning?

• Biological Plausibility – your brain works that way
• Problems that can be represented with a polynomial number of nodes with \( k \) layers, may require an exponential number of nodes with \( k-1 \) layers
• Highly varying functions can be efficiently represented with deep architectures
  • Less weights/parameters to update than a less efficient shallow representation
• Sub-features created in deep architecture can potentially be shared between multiple tasks
  • Type of Transfer/Multi-task learning
Hierarchical, Representation Learning

- Natural progression from low-level to high-level features as seen in natural complexity
- Automatically learns features from the raw data
- Easier to monitor what is being learned and to guide the learner to better subspaces

Difficulties Training Deep Networks

- Bottom layers do not get trained easily
- Error attenuates as it propagates to earlier layers
- Leads to very slow training
- For small data sets, not enough labeled data available for training

Pre-Train Multi-Layer NNs

Unsupervised pre-training:
Especially useful with small data sets to initialize weights

Pre-Train Multi-Layer NNs

Train this layer first
Pre-Train Multi-Layer NNs

Train this layer first
then this layer, etc.

EACH of the (non-output) layers is trained to be an Auto-Encoder
That is, it is forced to learn good features that describe what comes from the previous layer

Training a Deep Network

• Use when amount of labeled training data is relatively small
• Or use as a preprocessing step to initialize weights
• Then use traditional back-prop to train the full network