Neural Networks IV
Why Neural Networks?

Perceptron
- Proposed by Frank Rosenblatt in 1957
- Real inputs/outputs, threshold activation function
Revival in the 1980’s

Backpropagation discovered in 1970’s but popularized in 1986

MLP is a universal approximator
• Can approximate any non-linear function in theory, given enough neurons, data
• Kurt Hornik, Maxwell Stinchcombe, Halbert White. “Multilayer feedforward networks are universal approximators.” Neural Networks, 1989

Generated lots of excitement and applications

Neural Networks Applied to Vision

LeNet – vision application

- USPS digit recognition, later check reading

New “winter” and revival in early 2000’s

New “winter” in the early 2000’s due to
• problems with training NNs
• Support Vector Machines (SVMs), Random Forests (RF) – easy to train, nice theory

Revival again by 2011-2012
• Name change (“neural networks” -> “deep learning”)
• + Algorithmic developments that made training somewhat easier
• + Big data + GPU computing
• = performance gains on many tasks (esp Computer Vision)

Big Data

- **ImageNet Large Scale Visual Recognition Challenge**
  - 1000 categories w/ 1000 images per category
  - 1.2 million training images, 50,000 validation, 150,000 testing

AlexNet Architecture

60 million parameters!
Various tricks
• ReLU nonlinearity
• Overlapping pooling
• Local response normalization
• Dropout – set hidden neuron output to 0 with probability .5
• Data augmentation
• Training on GPUs

GPU Computing

• **Big data and big models** require lots of computational power

• **GPUs**
  – thousands of cores for parallel operations
  – multiple GPUs
  – still took about 5-6 days to train AlexNet on two NVIDIA GTX 580 3GB GPUs (much faster today)
Image Classification Performance

Image Classification Top-5 Errors (%)

- ILSVRC'15 ResNet: 3.57
- ILSVRC'14 GoogleNet: 6.7 (22 layers)
- ILSVRC'14 VGG: 7.3 (19 layers)
- ILSVRC'13: 11.7 (8 layers)
- ILSVRC'12 AlexNet: 16.4 (8 layers)
- ILSVRC'11: 25.8 (shallow)
- ILSVRC'10: 28.2 (152 layers)

Robustness of Classifiers

- Adversarial Examples (Szegedy et al.’14, Biggio et al.’13, Goodfellow et al.’14)
Additive L_p Attack Threat

```
\[
\text{Optimization: } \max_\delta \ell_{cls} (f_\theta(x + \delta), y)
\]
\[
\delta \in \Delta := \{ \delta \in \mathbb{R}^n : \|\delta\|_p \leq \rho \}
\]
```

Solve using Projected Gradient Descent (Madry et al.’17, Goodfellow et al.’15, Carlini & Wagner ‘16)
Adversarial Training

- **Standard ERM training:**
  \[
  \min_\theta \mathbb{E}_{(x,y)} [\ell_{cls}(f_\theta(x), y)]
  \]

- **Adversarial training** for additive attacks (Madry et al.’17):
  \[
  \min_\theta \mathbb{E}_{(x,y)} \left[ \max_\delta \ell_{cls}(f_\theta(x + \delta), y) \right]
  \]
  \[
  \delta \in \Delta := \{\delta \in \mathbb{R}^n : \|\delta\|_p \leq \rho\}
  \]

- Solve using alternative SGD+PGD
Certifiable Defense Against Adversarial Attacks
Speech Recognition

According to Microsoft’s speech group:

Word error rate on Switchboard

1990  2000  2010

Using DL
Recurrent Neural Networks for Language Modeling

• Speech recognition is difficult due to ambiguity
  – “how to recognize speech”
  – or “how to wreck a nice beach“?

• Language model gives probability of next word given history
  – $P(\text{"speech"} | \text{"how to recognize"})$?
Recurrent Neural Networks

Networks with loops

- The output of a layer is used as input for the same (or lower) layer
- Can model dynamics (e.g. in space or time)

Loops are unrolled

- Now a standard feed-forward network with many layers
- Suffers from vanishing gradient problem
- In theory, can learn long term memory, in practice not (Bengio et al, 1994)

Image credit: Christopher Olah’s blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Y. Bengio, P. Simard, P. Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult. In TNN
1994.
Long Short Term Memory (LSTM)

- A type of RNN explicitly designed not to have the vanishing or exploding gradient problem
- Models long-term dependencies
- Memory is propagated and accessed by gates
- Used for speech recognition, language modeling ...


Image credit: Christopher Colah’s blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Long Short Term Memory (LSTM)

Image credit: Christopher Colah’s blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/
What you should know about deep neural networks

• Why they are difficult to train
  – Initialization
  – Overfitting
  – Vanishing gradient
  – Require large number of training examples

• What can be done about it
  – Improvements to gradient descent
  – Stochastic gradient descent
  – Momentum
  – Weight decay
  – Alternate non-linearities and new architectures

References (& great tutorials) if you want to explore further:
http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Keeping things in perspective...

In 1958, the New York Times reported the perceptron to be "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."