Deep neural networks
Outline

• What’s new in ANNs in the last 5-10 years?
  – Deeper networks, more data, and faster training
    • Scalability and use of GPUs ✔
    • Symbolic differentiation ✔
      – reverse-mode automatic differentiation
      – “Generalized backprop”
    • Some subtle changes to cost function, architectures, optimization methods ✔
• What types of ANNs are most successful and why?
  – Convolutional networks (CNNs) ✔
  – Long term/short term memory networks (LSTM) ✔
  – Word2vec and embeddings
• What are the hot research topics for deep learning?
Review
Recap: parallelism in ANNs

Let $X$ be a matrix with $k$ examples
Let $w_i$ be the input weights for the $i$-th hidden unit
Then $Z = X W$ is output for all $m$ units
for all $k$ examples

There's a lot of chances to do this in parallel… with parallel matrix multiplication
Recap: parallel ANN training

- Modern libraries (Matlab, numpy, ...) do matrix operations fast, in parallel, on multicore machines
- Many ANN implementations exploit this parallelism automatically
- Key implementation issue is working with matrices comfortably
- GPUs do matrix operations very fast, in parallel
  - For dense matrixes, not sparse ones!
- Training ANNs on GPUs is common
  - SGD and minibatch sizes of 128
Recap: autodiff for a 2-layer neural network

**Step 1: forward**

**inputs:** $x_1, x_2, \ldots, x_n$

for $i = n + 1, n + 2, \ldots, N$

$x_i \leftarrow f_i(x_{\pi(i)})$

return $x_N$

**Inputs:** $X, W1, B1, W2, B2$

$Z1a = \text{mul}(X, W1)$ // matrix mult

$Z1b = \text{add}^*(Z11, B1)$ // add bias vec

$A1 = \tanh(Z1b)$ // element-wise

$Z2a = \text{mul}(A1, W2)$

$Z2b = \text{add}^*(Z2a, B2)$

$A2 = \tanh(Z2b)$ // element-wise

$P = \text{softmax}(A2)$ // vec to vec

$C = \text{crossEnt}_Y(P)$ // cost function

**Step 1: backprop**

for $i = N - 1, N - 2, \ldots, 1$

$$\frac{dx_N}{dx_i} \leftarrow \sum_{j : i \in \pi(j)} \frac{dx_N}{dx_j} \frac{\partial f_j}{\partial x_i}$$

$dC/dC = 1$

$dC/dP = dC/dC \times d\text{CrossEnt}_Y/dP$

$dC/dA2 = dC/dP \times d\text{softmax}/dA2$

$dC/Z2b = dC/dA2 \times d\tanh/dZ2b$

$dC/dZ1a = dC/dZ2b \times (d\text{add}^*/dZ2a + d\text{add}/dB2)$

$dC/dB2 = dC/Z2b \times 1$

$dC/dZ2a = dC/dZ2b \times (d\text{mul}/dA1 + d\text{mul}/dW2)$

$dC/dW2 = dC/dZ2a \times 1$

$dC/dA1 = ...$

Target $Y; N$ rows; $K$ outs; $D$ feats, $H$ hidden
Recap: 2-layer neural network

Step 1: forward

Inputs: $x_1, x_2, \ldots, x_n$

for $i = n + 1, n + 2, \ldots, N$

$x_i \leftarrow f_i(x_{\pi(i)})$

return $x_N$

An autodiff package usually includes

- A collection of matrix-oriented operations (mul, add*, …)
- For each operation
  - A **forward** implementation
  - A **backward implementation** for each argument

- A way of composing operations into expressions (often using operator overloading) which evaluate to **expression trees**
- Expression simplification/compilation

- Lots of tools: Theano, Torch, TensorFlow, ….
Recap: incremental improvements

- Use of softmax and cross-entropy loss
- Use of alternate non-linearities
  - reLU, hyperbolic tangent, ...
- Better understanding of weight initialization
- Tricks like data augmentation
Outline

• What’s new in ANNs in the last 5-10 years?
  – Deeper networks, more data, and faster training
    • Scalability and use of GPUs ✔
    • Symbolic differentiation ✔
      – reverse-mode automatic differentiation
      – “Generalized backprop”
    • Some subtle changes to cost function, architectures, optimization methods ✔

• What types of ANNs are most successful and why?
  – Convolutional networks (CNNs) ✔
  – Long term/short term memory networks (LSTM) ✔
  – Word2vec and embeddings

• What are the hot research topics for deep learning?
Recap: convolving an image with an ANN

Note that the parameters in the matrix defining the convolution are **tied** across all places that it is used.
Alternating convolution and downsampling

5 layers up

The subfield in a large dataset that gives the strongest output for a neuron
Similar technique applies to audio

Convolutional DBN for audio

(Lee et al., 2009)
Implementing an LSTM

For \( t = 1, \ldots, T \):

1. \[
f_t = \sigma \left( W_f \cdot [h_{t-1}, x_t] + b_f \right)
\]
   \[
i_t = \sigma \left( W_i \cdot [h_{t-1}, x_t] + b_i \right)
\]
   \[
\tilde{C}_t = \tanh \left( W_C \cdot [h_{t-1}, x_t] + b_C \right)
\]

2. \[
C_t = f_t \ast C_{t-1} + i_t \ast \tilde{C}_t
\]

3. \[
o_t = \sigma \left( W_o \cdot [h_{t-1}, x_t] + b_o \right)
\]
   \[
h_t = o_t \ast \tanh \left( C_t \right)
\]

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Character-level language model

```c
/**
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */

static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
         */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
        goto bail;
    }
    segaddr = in_SB(in.addr);
    selector = seg / 16;
    setup_works = true;
    for (i = 0; i < blocks; i++) {
        seq = buf[i++];
        bpf = bd->bd.next + i * search;
        if (fd) {
            current = blocked;
        }
    }
```
Outline

• What’s new in ANNs in the last 5-10 years?
  – **Deeper networks**, more data, and faster training
    • Scalability and use of GPUs ✔
    • Symbolic differentiation ✔
    • Some subtle changes to cost function, architectures, optimization methods ✔

• What types of ANNs are most successful and why?
  – Convolutional networks (CNNs) ✔
  – Long term/short term memory networks (LSTM) ✔
  – **Word2vec and embeddings**

• What are the hot research topics for deep learning?
Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com
Basic idea behind skip-gram embeddings

- Construct hidden layer that "encodes" that word from an input word \( w(t) \) in a document
- So that the hidden layer will predict likely nearby words \( w(t-K), \ldots, w(t+K) \)
- Final step of this prediction is a softmax over lots of outputs
Basic idea behind skip-gram embeddings

Training data: **positive** examples are pairs of words \( w(t), w(t+j) \) that co-occur.

Training data: **negative** examples are samples of pairs of words \( w(t), w(t+j) \) that don’t co-occur.

You want to train over a very large corpus (100M words+) and hundreds+ dimensions.
Results from word2vec

https://www.tensorflow.org/versions/r0.7/tutorials/word2vec/index.html
Results from word2vec

https://www.tensorflow.org/versions/r0.7/tutorials/word2vec/index.html
Results from word2vec

https://www.tensorflow.org/versions/r0.7/tutorials/word2vec/index.html
Outline

• What’s new in ANNs in the last 5-10 years?
  – **Deeper networks**, more data, and faster training
    • Scalability and use of GPUs ✔
    • Symbolic differentiation ✔
    • Some subtle changes to cost function, architectures, optimization methods ✔

• What types of ANNs are most successful and why?
  – Convolutional networks (CNNs) ✔
  – Long term/short term memory networks (LSTM) ✔
  – Word2vec and embeddings ✔

• What are the **hot research topics** for deep learning?
Some current hot topics

- Multi-task learning
  - Does it help to learn to predict many things at once? e.g., POS tags and NER tags in a word sequence?
  - Similar to word2vec learning to produce all context words
- Extensions of LSTMs that model memory more generally
  - e.g. for question answering about a story
Some current hot topics

• Optimization methods (>> SGD)
• Neural models that include “attention”
  – Ability to “explain” a decision
Basic idea: similarly to the way an LSTM chooses what to "forget" and "insert" into memory, allow a network to choose what inputs to "attend to" in generation phase.
Examples of attention

ACL 15, Li, Luong, Jurafsky
<table>
<thead>
<tr>
<th></th>
<th>Russia</th>
<th>Calls</th>
<th>For</th>
<th>Joint</th>
<th>Front</th>
<th>Against</th>
<th>Terrorism</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;s&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Russia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defense</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minister</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ivanov</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Called</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sunday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Of a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Front</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terrorism</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Examples of attention

(a) Translated MNIST inputs.

(b) Cluttered Translated MNIST inputs.

Examples of attention

Basic idea: similarly to the way an LSTM chooses what to “forget” and “insert” into memory, allow a network to choose a path to focus on in the visual field

Some current hot topics

• Knowledge-base embedding: extending word2vec to embed large databases of facts about the world into a low-dimensional space.
  — TransE, TransR, ...

• “NLP from scratch”: sequence-labeling and other NLP tasks with minimal amount of feature engineering, only networks and character- or word-level embeddings
Some current hot topics

• Computer vision: complex tasks like generating a natural language caption from an image or understanding a video clip

• Machine translation
  – English to Spanish, ...

• Using neural networks to perform tasks
  – Driving a car
  – Playing games (like Go or ...)
    • Reinforcement learning