Sequence Models and HAL Deep Learning Cluster

Presented By: Aaron D. Saxton, PhD
Get Setup!

> ssh hal

hal> git clone https://git.ncsa.illinois.edu/saxton/hal-training-sequence-modeling.git

On Firefox or Chrome log onto,

https://hal.ncsa.illinois.edu:8888

https://xkcd.com/2265/

YOU MAY CLAIM UP TO 10% OF DEFENDANTS ON YOUR SEITAN LOCAL INCOME TAX FOR FISCAL YEAR 20202 BY TAKING THE STANDARD DEDUCTION AND ATOMIZING YOUR CLAIMS.

I USED A NEURAL NET TO PREPARE MY TAX RETURNS, BUT I THINK I CUT OFF ITS TRAINING TOO EARLY.
Sequence Modeling

- **Autoregression**
  \[ X_t = c + \sum_{i=1}^{p} \phi_i B^i X_t + \epsilon_t \]
  Back Shift Operation: \( B^i \)
- **Autocorrelation**
  \[ R_{XX}(t_1, t_2) = E[X_{t_1} X_{t_2}] \]
- **Other tasks**
  - **Semantic Labeling**

The quick red fox jumps over the lazy brown dog
Sequence Modeling

• Sequence 2 Sequence models
  • Continuing a Sequence of Number
    Sequence 1: 1, 2, 4, 8
    Sequence 2: 16, 32, 64, 128
  • Generating a Text Response
    Sequence 1: Did you hear that Gary won the contract? That’s going to be a lot of work.
    Sequence 2: You’re right, but it will be an exciting project.
Recurrent Neural Networks: Sequence Model

Input Is: \( \{x_0, x_1, x_2, \ldots, x_N\} \)
Activation Function: \( \sigma(x) \)
State At Step \( t \): \( h_t \)
Hidden Input Weights: \( U \)
Hidden output Weights: \( V \)
Hidden State Step Weights: \( W \)

\[
h_t = \sigma(U \cdot x_t + W \cdot h_{t-1})
\]

\[
y_t = \text{Softmax}(V \cdot h_t)
\]
Recurrent Neural Networks: Sequence Model

- Continuing the sequence 1, 2, 4, 8

- A sample from the training set would look like
  
  **Input:** {1, 2, 4, 8, 16, 32, 64, 128}
  
  **Output:** {16, 32, 64, 128, 256}

- Loss function could be
  
  $$R = \sum_{i=4}^{6} (x_i - \tilde{y}_{i-1})^2$$
Gradient Decent

- Searching for minimum

\[ \nabla R = \left\langle R_{\theta_0}, R_{\theta_2}, \ldots, R_{\theta_n} \right\rangle \]

\[ R(\vec{\theta}_{t+1}) = R(\vec{\theta}_t + \gamma \nabla R) \]

- \( \gamma \): Learning Rate
- Recall, Loss depends on data

Expand notation,

\[ R(\vec{\theta}_t; \{ (x_i, y_i) \}_{i=1}^{n}) \]

- Recall \( R \) and \( \nabla R \) is a sum over \( i \)
- Want \( R \) with ALL DATA ..... ?

\[ R = \sum_{i=1}^{n} [(y_i - f_{\theta_0}(x_i))^2] \]
Stochastic Gradient Decent

- Recall $R$ is a sum over $i$ ($R = \sum_{i=1}^{n} [(y_i - f_{\theta_t}(x_i))^2]$)
- Single training example, $(x_i, y_i)$, Sum over only one training example
  \[ \nabla R(x_i, y_i) = \left\langle R_{\theta_0}, R_{\theta_2}, \ldots, R_{\theta_n} \right\rangle_{(x_i, y_i)} \]
  \[ R(x_i, y_i)(\vec{\theta}_{t+1}) = R(x_i, y_i)(\vec{\theta}_t + \gamma \nabla R(x_i, y_i)) \]
- $\gamma$: Learning Rate
- Computing $R$ involved composing $W, x_t, h_t$ into $h_{t+1}, x_{t+1}$ to get $\tilde{y}_{t+1}$
- Therefore $\nabla R$ requires the entire RNN to be “Unrolled”
- RNN can be though of as super deep NN, as deep as your sequence

\[ h_t = \sigma(U \cdot x + W \cdot h_{t-1}) \]
\[ y_t = \text{Softmax} (V \cdot h_t) \]
Demo
LSTM: Sequence Model

- Input Is: \( \{x_0, x_1, x_2, \ldots, x_N\} \)
- Activation Function: \( \sigma(x) \)
- "Dense Layer": \( nn(x) \)
- \( f_t = nn(W_f \cdot [y_{t-1}, x_t] + b_f) \), Forget Gate
- \( i_t = nn(W_i \cdot [y_{t-1}, x_t] + b_i) \), Input Gate
- \( \tilde{C}_t = \sigma(W_C \cdot [y_{t-1}, x_t] + b_C) \)
- \( o_t = nn(W_o \cdot [y_{t-1}, x_t] + b_o) \), Output Gate
- \( C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \)
- \( y_t = o_t \cdot \sigma(C_t) \)
- Note: \( \cdot \) is point wise multiplication
- Note: \( [y_{t-1}, x_t] \) is concatenation
LSTM: Sequence Model

- ChapBot Data, Movie Dialog
  - lines
    
    L1045 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ They do not!
    L1044 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ They do to!
    L985 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ I hope so.
    m0 +++$+++ CAMERON +++$+++ She okay?
    L925 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Let's go.
    L924 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ Wow

- conversations
  
  u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L194', 'L195', 'L196', 'L197']
  u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L198', 'L199']
  u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L200', 'L201', 'L202', 'L203']
  u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L204', 'L205', 'L206']
  u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L207', 'L208']
  u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L271', 'L272', 'L273', 'L274', 'L275']

- movie title meta
  
  m0 +++$+++ 10 things i hate about you +++$+++ 1999 +++$+++ 6.90 +++$+++ 62847 +++$+++ ['comedy', 'romance']
  m1 +++$+++ 1492: conquest of paradise +++$+++ 1992 +++$+++ 6.20 +++$+++ 10421 +++$+++ ['adventure', 'biography', 'drama', 'history']
  m2 +++$+++ 15 minutes +++$+++ 2001 +++$+++ 6.10 +++$+++ 25854 +++$+++ ['action', 'crime', 'drama', 'thriller']
  m3 +++$+++ 2001: a space odyssey +++$+++ 1968 +++$+++ 8.40 +++$+++ 163227 +++$+++ ['adventure', 'mystery', 'sci-fi']
  m4 +++$+++ 48 hrs. +++$+++ 1982 +++$+++ 6.90 +++$+++ 22209 +++$+++ ['action', 'comedy', 'crime', 'drama', 'thriller']
  m5 +++$+++ the fifth element +++$+++ 1997 +++$+++ 7.50 +++$+++ 133756 +++$+++ ['action', 'adventure', 'romance', 'sci-fi', 'thriller']

- character meta
  
  u0 +++$+++ BIANCA +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ f +++$+++ 4
  u1 +++$+++ BRUCE +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ ? +++$+++ ?
  u2 +++$+++ CAMERON +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ m +++$+++ 3
  u3 +++$+++ CHASTITY +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ ? +++$+++ ?
  u4 +++$+++ JOEY +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ m +++$+++ 6
Demo
Extras
Gradient Decent

Fictitious Loss Surface With Gradient Field
Neural Networks

- Parameterized function
  - \( Z_M = \sigma(\alpha_{0m} + \alpha_m X) \)
  - \( T_K = \beta_{0k} + \beta_k Z \)
  - \( f_K(X) = g_k(T) \)
  - Linear Transformations with bias (Affine?) and point wise evaluation of nonlinear function, \( \sigma \)

- \( \beta_{0i}, \beta_i, \alpha_{0m}, \alpha_m \)
  - Weights to be optimized