CS540 Introduction to Artificial Intelligence
Lecture 24

Young Wu
Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

July 5, 2021

Dynamic System

Motivation

- The hidden units are used as the hidden states.
- They are related by the same function over time.

$$
\begin{aligned}
h_{t+1} &= f\left(h_t, w\right) \\
h_{t+2} &= f\left(h_{t+1}, w\right) \\
h_{t+3} &= f\left(h_{t+2}, w\right) \\
&\dots
\end{aligned}
$$

Dynamic System with Input

Motivation

- The input units can also drive the dynamics of the system.
- They are still related by the same function over time.

$$
\begin{aligned}
h_{t+1} &= f\left(h_t, x_{t+1}, w\right) \\
h_{t+2} &= f\left(h_{t+1}, x_{t+2}, w\right) \\
h_{t+3} &= f\left(h_{t+2}, x_{t+3}, w\right) \\
&\dots
\end{aligned}
$$

Dynamic System with Output

Motivation

- The output units only depend on the hidden states.

$$
\begin{aligned}
y_{t+1} &= f\left(h_{t+1}\right) \\
y_{t+2} &= f\left(h_{t+2}\right) \\
y_{t+3} &= f\left(h_{t+3}\right) \\
&\ldots
\end{aligned}
$$

Dynamic System Diagram

Motivation

Recurrent Neural Network Structure Diagram

Motivation

Activation Functions

Definition

- The hidden layer activation function can be the tanh activation, and the output layer activation function can be the softmax function.

$$
\begin{aligned}
z_t^{(x)} &= W^{(x)} x_t + W^{(h)} a_{t-1}^{(x)} + b^{(x)} \\
a_t^{(x)} &= g\left(z_t^{(x)}\right), g\left(\boxed{\cdot}\right) = \text{ tanh } \left(\boxed{\cdot}\right) \\
z_t^{(y)} &= W^{(y)} a^{(1,t)} + b^{(y)} \\
a_t^{(y)} &= g\left(z_t^{(y)}\right), g\left(\boxed{\cdot}\right) = \text{ softmax } \left(\boxed{\cdot}\right)
\end{aligned}
$$

Cost Functions

Definition

- Cross entropy loss is used with softmax activation as usual.

$$C_t = H\left(y_t, a_t^{(y)}\right)$$
$$C = \sum_t C_t$$

Multiple Sequential Data Notations

Definition

- There could multiple sequences in the training set index by $i = 1, 2, ..., n$. For one training instance, at time t, there are m features.
- x_{ijt} is the feature j of instance i at time t (position t of the sequence).
- y_{ijt} is the output j of instance i at time t (position t of the sequence).

Multiple Sequential Activations Notations

Definition

- $z_{ijt}^{(x)}$ denotes the linear part of instance i unit j at time t in the hidden layer.
- $a_{ijt}^{(x)}$ denotes the activation of instance i unit j at time t in the hidden layer.
- $z_{ijt}^{(y)}$ denotes the linear part of instance i output j at time t in the output layer.
- $a_{ijt}^{(y)}$ denotes the activation of instance i output j at time t in the output layer

Multiple Sequential Weights Notations, Part 1

Definition

- There are weights and biases between the input layer and the hidden layer, between the hidden layer and the output layer, as in usual neural networks.
- $w_{j'j}^{(x)}, j' = 1, ..., m, j = 1, ..., m^{(h)}$ denotes the weight from input feature j' to hidden unit j.
- $b_j^{(x)}, j = 1, ..., m^{(h)}$ denotes the bias of hidden unit j.
- $w_{jj'}^{(y)}, j = 1, ..., m^{(h)}, j' = 1, ..., K$ denotes the weight from hidden unit j to output unit j'.
- $b_{j'}^{(y)}, j' = 1, ..., K$ denotes the bias of output unit j'.

Multiple Sequential Weights Notations, Part 2

Definition

- There are also weights between units within the hidden layer through time.
- $w^{(h)}_{j'j}, j, j' = 1, ..., m^{(h)}$ denotes the weight from hidden unit j' at time t to hidden unit j at time $t + 1$.

BackPropogation Through Time

Definition

- The gradient descent algorithm for recurrent neural networks is called BackPropogation Through Time (BPTT). The update procedure is the same as standard neural networks using the chain rule.

$$w = w - \alpha \frac{\partial C}{\partial w}$$
$$b = b - \alpha \frac{\partial C}{\partial b}$$

Unfolded Network Diagram

Definition

Backpropagation Diagram 1

Definition

Backpropagation Diagram 2

Definition

Backpropagation, Part 1

Definition

- The cost derivative is the same as softmax neural networks.

$$\frac{\partial C}{\partial C_t} = 1$$
$$\frac{\partial C_t}{\partial z_{ijt}^{(y)}} = z_{ijt}^{(y)} - \mathbb{1}_{\{y_{it}=j\}}$$

Backpropagation, Part 2

Definition

- The other derivatives are similar to fully connected neural networks.

$$\frac{\partial z^{(y)}_{ij't}}{\partial a^{(x)}_{ijt}} = w^{(y)}_{jj'}$$

$$\frac{\partial z^{(y)}_{ij't}}{\partial w^{(y)}_{jj'}} = a^{(x)}_{ijt}$$

$$\frac{\partial z^{(y)}_{ij't}}{\partial b^{(y)}_{j'}} = 1$$

Backpropagation, Part 3

Definition

- The other derivatives are similar to fully connected neural networks.

$$\frac{\partial a_{ijt}^{(x)}}{\partial z_{ijt}^{(x)}} = g'\left(z_{ijt}^{(x)}\right) = 1 - \left(a_{ijt}^{(x)}\right)^2$$

$$\frac{\partial z_{ijt}^{(x)}}{\partial w_{j'j}^{(x)}} = x_{ij't}$$

$$\frac{\partial z_{ijt}^{(x)}}{\partial b_{j}^{(x)}} = 1$$

Backpropagation, Part 4

Definition

- The chain rule goes through time, so each gradient involves a long chain of the partial derivatives between $a_t^{(x)}$ and $a_{t-1}^{(x)}$ for $t = 1, 2, ..., T$.

$$\frac{\partial a_{ijt}^{(x)}}{\partial z_{ijt}^{(x)}} = 1 - \left(a_{ijt}^{(x)}\right)^2$$

$$\frac{\partial z_{ijt}^{(x)}}{\partial a_{ij't-1}^{(x)}} = w_{j'j}^{(h)}$$

Vanishing and Exploding Gradient

Discussion

- If the weights are small, the gradient through many layers will shrink exponentially. This is called the vanishing gradient problem.
- If the weights are large, the gradient through many layers will grow exponentially. This is called the exploding gradient problem.
- Fully connected and convolutional neural networks only have a few hidden layers, so vanishing and exploding gradient is not a problem in training those networks.
- In a recurrent neural network, if the sequences are long, the gradients can easily vanish or explode.

Long Term Memory

Discussion

- It is also very hard to detect that the current output depends on an input from many time steps ago.
- Recurrent neural networks have difficulty dealing with long-range dependencies.

Long Short Term Memory

Discussion

- Long Short Term Memory (LSTM) network adds more connected hidden units for memories controlled by gates. The activation functions used for these gates are usually logistic functions.
- An LSTM unit usually contains an input gate, an output gate, and a forget gate, to keep track of the dependencies in the input sequence.

Long Short Term Memory Diagram

Discussion

Gated Recurrent Unit

Discussion

- Gated Recurrent Unit (GRU) does something similar to an LSTM unit.
- A GRU contains input and forget gates, and does not contain an output gate.

Gated Recurrent Unit Diagram

Discussion