NLP hw3

Huizhen Jin [hj1314]

October 2025

1 Recurrent Neural Network

RNN Derivatives

1 The per-example cross-entropy loss for the classification task is defined as:

$$\ell(y, \mathbf{p}_T) = -\sum_{i=1}^{k} y[i] \cdot \log(p_T[i])$$

Since $y \in \{0,1\}^k$ is a one-hot vector, only the true label contributes to the sum. Let c be the index such that $y[c] = 1$. Then the expression simplifies to:

$$\ell(y, \mathbf{p}_T) = -\log(p_T[c])$$

2(a) Let $w = W_{hh}[i,j]$ be the (i,j)-th entry of the recurrent weight matrix. We treat h_{i-1} as constant and compute the immediate gradient $\frac{\partial h_i^+}{\partial w}$ using the chain rule:

$$\frac{\partial h_i^+}{\partial W_{hh}[i,j]} = \sigma'(z_i[i]) \cdot h_{i-1}[j]$$

where:

- $z_i = W_{hh}h_{i-1} + W_{ih}x_i + b_h$ is the pre-activation input to the nonlinearity at time step i,
- $\sigma(\cdot)$ is the tanh activation function, and
- $\sigma'(z_i[i]) = 1 - \tanh^2(z_i[i])$

2(b) We are asked to expand the gradient vector $\frac{\partial h_t}{\partial h_i}$ using the chain rule, expressing it as a product of partial derivatives between successive hidden states.

Using the chain rule, we have:

$$\frac{\partial h_t}{\partial h_i} = \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial h_{t-2}} \cdot \dots \cdot \frac{\partial h_{i+1}}{\partial h_i} = \prod_{j=i+1}^{t} \frac{\partial h_j}{\partial h_{j-1}}$$

This expression captures how the hidden state at time i influences the final hidden state h_t through all intermediate steps from $i+1$ to t.

3 We are asked to write the Jacobian matrix $\frac{\partial h_{i+1}}{\partial h_i}$ by applying the rules of differentiation.

Recall that the hidden state is updated as:

$$h_{i+1} = \sigma(W_{hh}h_i + W_{ih}x_{i+1} + b_h)$$

Let $z_{i+1} = W_{hh}h_i + W_{ih}x_{i+1} + b_h$, so that $h_{i+1} = \sigma(z_{i+1})$. Applying the chain rule:

$$\frac{\partial h_{i+1}}{\partial h_i} = \frac{\partial \sigma(z_{i+1})}{\partial z_{i+1}} \cdot \frac{\partial z_{i+1}}{\partial h_i}$$

We now compute each term:

- $\frac{\partial z_{i+1}}{\partial h_i} = W_{hh}$
- $\frac{\partial \sigma(z_{i+1})}{\partial z_{i+1}} = \text{diag}(\sigma'(z_{i+1}))$, since the activation function is applied element-wise

Therefore, the full Jacobian is:

$$\frac{\partial h_{i+1}}{\partial h_i} = \text{diag}(\sigma'(z_{i+1})) \cdot W_{hh}$$

Bounding Gradient Norm

1 Given the Jacobian matrix derived earlier,

$$\frac{\partial h_i}{\partial h_{i-1}} = \text{diag}(\sigma'(z_i)) \cdot W_{hh},$$

we can apply the submultiplicative property of the spectral norm:

$$\|AB\|_2 \leq \|A\|_2 \cdot \|B\|_2$$

to obtain the following bound:

$$\boxed{\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 \leq \|\text{diag}(\sigma'(z_i))\|_2 \cdot \|W_{hh}\|_2}$$

Since the activation function $\sigma(\cdot) = \tanh(\cdot)$ satisfies $\sigma'(z_i) \leq 1$ for all z_i, we also have the looser upper bound:

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 \leq \|W_{hh}\|_2$$

2 We can express the derivative of the hidden state at time t with respect to that at time i as a product of Jacobians:

$$\frac{\partial h_t}{\partial h_i} = \prod_{k=i+1}^{t} \frac{\partial h_k}{\partial h_{k-1}}$$

Taking the spectral norm on both sides and applying the submultiplicative property of matrix norms gives:

$$\left\| \frac{\partial h_t}{\partial h_i} \right\|_2 \leq \prod_{k=i+1}^{t} \left\| \frac{\partial h_k}{\partial h_{k-1}} \right\|_2 \leq \prod_{k=i+1}^{t} \left(\|\text{diag}(\sigma'(z_k))\|_2 \cdot \|W_{hh}\|_2 \right)$$

Since the derivative of the tanh activation satisfies $\sigma'(z) \leq 1$, each $\|\text{diag}(\sigma'(z_k))\|_2 \leq 1$. Hence, the gradient norm is bounded by:

$$\boxed{\left\| \frac{\partial h_t}{\partial h_i} \right\|_2 \leq \|W_{hh}\|_2^{(t-i)}}$$

If $\|W_{hh}\|_2 < 1$, this bound decays exponentially, leading to **vanishing gradients**. If $\|W_{hh}\|_2 > 1$, it grows exponentially, causing **exploding gradients**. Only when $\|W_{hh}\|_2 \approx 1$ do gradients remain stable during backpropagation through time.