Output of check_gradients

The loss derivative looks OK.
The gradient for word_embedding_weights looks OK.
The gradient for embed_to_hid_weights looks OK.
The gradient for hid_to_output_weights looks OK.
The gradient for hid_bias looks OK.
wThe gradient for output_bias looks OK.
loss_derivative[2, 5] 0.001112231773782498
loss_derivative[2, 121] -0.9991004720395987
loss_derivative[5, 33] 0.0001903237803173703
loss_derivative[5, 31] -0.7999757709589483

param_gradient.word_embedding_weights[27, 2] -0.27199539981936866
param_gradient.word_embedding_weights[43, 3] 0.8641722267354154
param_gradient.word_embedding_weights[22, 4] -0.2546730202374648
param_gradient.word_embedding_weights[2, 5] 0.0

param_gradient.embed_to_hid_weights[10, 2] -0.6526990313918257
param_gradient.embed_to_hid_weights[15, 3] -0.13106433000472612
param_gradient.embed_to_hid_weights[30, 9] 0.11846774618169399
param_gradient.embed_to_hid_weights[35, 21] -0.10004526104604386

param_gradient.hid_bias[10] 0.2537663873815642
param_gradient.hid_bias[20] -0.03326739163635357

param_gradient.output_bias[0] -2.0627596032173052
param_gradient.output_bias[1] 0.0390200857392169
param_gradient.output_bias[2] -0.7561537928318482
param_gradient.output_bias[3] 0.21235172051123635

2019年05月12日

CSC421 PA1

1. Params in embedding: $D \times V$ matrix

   D = word embedding size = 16

   V ← vocab size = 250

   Embed to hidden parameters: $\underbrace{3D}_{48} \times \underbrace{16}_{\text{\# units in hidden layer}} + \underbrace{16}_{\text{biases}}$

   Hidden to ouput params: $16 \times 250 + 250$

   Total: $(250 \times 16) + (48 \times 16 + 16) + (16 \times 25 + 250) = 5434$

2. 4-gram e.g. $p(x_4 \mid x_3, x_2, x_1)$

   250^4 entries $\therefore 4 \times 250^4 = 1.5625 \times 10^{10}$

For batch input $X = \begin{bmatrix} — x^{(1)} — \\ \vdots \end{bmatrix}$

$$\begin{bmatrix} — x^{(1)} — \\ \vdots \\ — x^{(N)} — \end{bmatrix} \underbrace{\begin{bmatrix} | & & | \\ w^1 & \cdots & w^0 \\ | & & | \end{bmatrix}}_{W^T} = \begin{bmatrix} — z^{(1)} — \\ \vdots \\ — z^{(0)} — \end{bmatrix}$$

$\therefore XW^T$

If $y_i = 0$???

$$t = [\quad] \leftarrow 250$$

$$y = [\quad] \leftarrow 250$$

$$C = -\sum_{i=1}^{V} t_i \log y_i = t \cdot \log y$$

$$Z \to Y \to C$$

$$y_i = \frac{e^{z_i}}{\sum_j e^{z_j}} = \frac{e^{z_i}}{\Sigma}$$

$$\frac{dy}{dz} = \left[\frac{dy_i}{dz}\right] \in \mathbb{R}^{V \times V}$$

$$\frac{\partial y_i}{\partial z_i} = \frac{\Sigma e^{z_i} - e^{z_i} e^{z_i}}{(\Sigma)^2} = \frac{e^{z_i}}{\Sigma} \cdot \frac{\Sigma - e^{z_i}}{\Sigma} = (y_i)(1 - y_i)$$

$$\frac{\partial y_i}{\partial z_k} = \frac{-e^{z_i} e^{z_k}}{(\Sigma)^2} = -y_i y_k$$

$$\frac{\partial C}{\partial z_k} = -\sum_{i=1}^{V} t_i \frac{\partial \log y_i}{\partial z_k} = -\sum_{i=1}^{V} t_i \frac{1}{y_i} \frac{\partial y_i}{\partial z_k}$$

only one of these terms is non-zero

if $i = k$: $t_i \cdot \frac{1}{y_i} (y_i)(1 - y_i) = t_i (1 - y_i)$

if $i \neq k$: $t_i \frac{1}{y_i} (-y_i y_k) = -t_i y_k$

$$\frac{\partial C}{\partial z_k} = -\left(t_k (1 - y_k)\right) + \sum_{i \neq k}^{V} t_i y_k$$

$$= -t_k + t_k y_k + \sum_{i \neq k}^{V} t_i y_k$$

$$= -t_k + y_k \sum_{i=1}^{V} t_k$$

$$= \underline{\underline{-t_k + y_k}}$$

, want embed_to_hid_weights, hid_bias, hid_to_output_weights, output_bias

Say embedding vector is $X \in \mathbb{R}^{3 \times D}$

$$\begin{tikzcd} X \arrow[dr] & & p \arrow[d, equal] & W^{(2)} \arrow[dr] & & & \\ W^{(1)} \arrow[r] & h \arrow[r] & \text{logistic}(h) \arrow[rr] & & z \arrow[r] & \text{softmax}(z) = y \arrow[r] & C \\ b^{(1)} \arrow[ur] & & & b^{(2)} \arrow[ur] & & & \end{tikzcd}$$

Showed $\frac{dC}{dz_n} = y_n - t_n$

$z = W^{(2)}(1+e^{-h})^{-1} + b^{(2)} = W^{(2)}p + b^{(2)}$. Clearly $\frac{dz}{dp} = W^{(2)}$

What is $\frac{dz}{dW^{(2)}}$? Recall we showed if $y = Wh + b$, $\frac{dL}{dW} = h\frac{dL}{dy}$

$\therefore \frac{dz}{dW^{(2)}} = p$

$\frac{dL}{dW^{(2)}} = p\frac{dL}{dz}$

$p = (1+e^{-h})^{-1}$

$\frac{dp}{dh} = -(1+e^{-h})^{-2}(-e^{-h}) = \frac{e^{-h}}{(1+e^{-h})^2} = p(1-p)$

$\therefore \frac{dL}{dh} = \frac{dL}{dz} W^{(2)} p(1-p)$ ← as seen in the code ✓

$$\begin{bmatrix} | \\ p \\ | \end{bmatrix} \begin{bmatrix} — \frac{dL}{dz} — \end{bmatrix}$$

What if $P = \begin{bmatrix} — p^{(1)} — \\ \vdots \\ — p^{(N)} — \end{bmatrix}$, $\frac{dL}{dz} = \begin{bmatrix} — \frac{dL_1}{dz} — \\ \vdots \\ — \frac{dL_N}{dz} — \end{bmatrix}$

$\therefore$ transpose this

If $y = Wh + b = \begin{bmatrix} W_{11}h_1 + \dots + b_1 \\ \vdots \\ W_{D1}h_1 + \dots + b_D \end{bmatrix}$ $\therefore \frac{dy}{db} = \begin{bmatrix} 1 & \dots & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$

$b \leftarrow b - \alpha\left(\frac{dy}{db}\right)^T$ ← but the dims do not match?

Grad desc. only works for scalar cost fct at the end.

$\frac{dz}{db^{(2)}} = I$ $\therefore \frac{dL}{db^{(2)}} = \bar{z} I$ Note, for batch, this yields $\begin{bmatrix} \frac{dL_1}{db_1} & \dots & \frac{dL_1}{db_N} \\ \frac{dL_B}{db_1} & \dots & \frac{dL_B}{db_N} \end{bmatrix}$

$\therefore$ For batch, at the end, sum the columns.

Part 3 Analysis

1. city of new → york $p = 0.99007$

life in the → world $p = 0.13141$

game $p = 0.07949$

united $p = 0.05964$

he is the → same $p = 0.22464$

best $p = 0.17594$

first $p = 0.05054$

Yes, made sensible predictions. But notice e.g. ~~city~~ p(york | city of new) is perhaps too high.

you and the → same $p = 0.08333$

man $p = 0.05816$

other $p = 0.05031$

→ unseen example, but good output.

2. Custer examples:

- 'what', 'when', 'who', 'until' - 'should', 'could', 'would', 'may'
- 'does', 'do', 'did' -
- big cluster of nouns in the center. e.g. 'public', 'police', 'government', 'department', 'company', 'court', ...

3. 'new' and 'york' are not close together. The embeddings learn a words "roles", and new is an adjective, york is a noun

4. "government" "university" → 0.986966 ← closer.

"government", "political" → ~~4~~ 1.342831

Again, "government" and "university" play a more similar role in sentences b/c they are nouns, while "political" is an adjective.