

Assignment 1 Notes

mn

Contents

1	Definitions	1
2	Assignment 1	1
2.1	1. Softmax	1
2.1.1	1.a) Softmax function	1
2.2	2. Neural Network Basics	1
2.2.1	2.a) Derivative of the sigmoid function	1
2.2.2	2.b) Derivative of the Cross Entropy Loss	2
2.2.3	2.c) Gradient of the one-layer network	2
2.2.4	2.d) Number of parameters	3
2.3	3. word2vec	3
2.3.1	3.a) Derivative of the skipgram 1 — with respect to \mathbf{v}_c	3
2.3.2	3.b) Derivative of the skipgram 2 — with respect to \mathbf{u}_w	3
2.3.3	3.c) Derivative of the skipgram 3	3

1 Definitions

- row vectors:
- 1 layer network:
 - Input dimension: D_x ($\mathbf{x} \in [N, D_x]$)
 - Hidden units: H ($\mathbf{W}_1 \in [D_x, H]$, $\mathbf{b}_1 \in [H]$)
 - Output dimension: D_y ($\mathbf{W}_2 \in [H, D_y]$, $\mathbf{b}_2 \in [D_y]$)

2 Assignment 1

Org latex export is not that great yet, and one still has to work on the alignment.

2.1 1. Softmax

2.1.1 1.a) Softmax function

Softmax function: $\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$

$$\text{softmax}(x + c)_i = \frac{e^{x_i + c}}{\sum_j e^{x_j + c}} = \frac{e^c e^{x_i}}{e^c \sum_j e^{x_j}} = \frac{e^{x_i}}{\sum_j e^{x_j}} = \text{softmax}(x)_i$$

2.2 2. Neural Network Basics

2.2.1 2.a) Derivative of the sigmoid function

Sigmoid function: $\sigma(x) = \frac{1}{1 + e^{-x}}$

$$\frac{\partial}{\partial x} \sigma(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{1}{1 + e^{-x}} \frac{e^{-x}}{1 + e^{-x}} = \frac{1}{1 + e^{-x}} \frac{(1 + e^{-x}) - 1}{1 + e^{-x}} = \sigma(x)(1 - \sigma(x))$$

2.2.2 2.b) Derivative of the Cross Entropy Loss

Cross Entropy Loss: $CE(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_i y_i \log(\hat{y}_i)$

$$\hat{\mathbf{y}} = \text{softmax}(\theta)$$

$$\frac{\partial}{\partial \theta_j} CE(\mathbf{y}, \hat{\mathbf{y}})_j = -\frac{\partial}{\partial \theta_j} \sum_i y_i \log \hat{y}_i = -\frac{\partial}{\partial \theta_j} \sum_i y_i \log(e^{\theta_i} / \sum_l e^{\theta_l}) = -\frac{\partial}{\partial \theta_j} \sum_i y_i (\theta_i - \log \sum_l e^{\theta_l})$$

\mathbf{y} is a *one-hot* label vector with a non-zero value (=1) at index k .

Given that

$$\frac{\partial}{\partial \theta_j} \log \sum_l e^{\theta_l} = \frac{e^{\theta_j}}{\sum_l e^{\theta_l}} = \hat{y}_j$$

if $j \neq k$:

$$\frac{\partial}{\partial \theta_j} CE(\mathbf{y}, \hat{\mathbf{y}})_j = \frac{\partial}{\partial \theta_j} \log \sum_l e^{\theta_l} = \hat{y}_j$$

if $j = k$:

$$\frac{\partial}{\partial \theta_j} CE(\mathbf{y}, \hat{\mathbf{y}})_j = \hat{y}_j - 1$$

Combining (vectorizing) these results:

$$\frac{\partial}{\partial \theta} CE(\mathbf{y}, \hat{\mathbf{y}}) = \hat{\mathbf{y}} - \mathbf{y}$$

2.2.3 2.c) Gradient of the one-layer network

$$\frac{\partial J(x)}{\partial x} = \frac{\partial}{\partial x} CE(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{\partial}{\partial x} \sum_i y_i \log(\text{softmax}(\sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2))$$

With $\hat{\mathbf{y}} = \text{softmax}(\theta)$, $\theta = \sigma(\mathbf{h})\mathbf{W}_2 + \mathbf{b}_2$ and $\mathbf{h} = \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1$

$$\frac{\partial J(x)}{\partial x} = \sum_j \frac{\partial J}{\partial \theta_j} \frac{\partial \theta_j}{\partial x} = \sum_j \sigma(\mathbf{h})(1 - \sigma(\mathbf{h}))\mathbf{W}_1\mathbf{W}_{2,j} \times \begin{cases} \hat{y}_j & \text{if } j \neq k \\ \hat{y}_j - 1 & \text{if } j = k \end{cases}$$

1. Alternative (more layer-systematic) derivation

- Forward pass:

$$J(x) = CE(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_i y_i \log(\text{softmax}(\sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2))$$

- Input: \mathbf{x}
- L1 affine transformation: $\mathbf{z}_1 = \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1$
- L1 activation: $\mathbf{h} = \sigma(\mathbf{z}_1)$
- L2 affine transformation: $\mathbf{z}_2 = \mathbf{h}\mathbf{W}_2 + \mathbf{b}_2$
- Scores computation: $\hat{\mathbf{y}} = \text{softmax}(\mathbf{z}_2)$
- Cross-Entropy Loss: $CE = \sum_i \mathbf{y} \log \hat{\mathbf{y}}$

- Backward pass (gradient derivation):

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_2} \frac{\partial \mathbf{z}_2}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}_1} \frac{\partial \mathbf{z}_1}{\partial x} = \frac{\partial J}{\partial x} = (\hat{\mathbf{y}} - \mathbf{y})\mathbf{W}_2^T \sigma(\mathbf{z}_1)(1 - \sigma(\mathbf{z}_1))\mathbf{W}_1^T$$

- $d\mathbf{z}_2 = \frac{\partial J}{\partial \mathbf{z}_2} = \hat{\mathbf{y}} - \mathbf{y}$
- $d\mathbf{h}_1 = \frac{\partial J}{\partial \mathbf{h}} = d\mathbf{z}_2 \frac{\partial \mathbf{z}_2}{\partial \mathbf{h}} = d\mathbf{z}_2 \mathbf{W}_2^T$
- $d\mathbf{z}_1 = \frac{\partial J}{\partial \mathbf{z}_1} = d\mathbf{h}_1 \frac{\partial \mathbf{h}}{\partial \mathbf{z}_1} = d\mathbf{h}_1 \circ \frac{\partial \sigma(\mathbf{z}_1)}{\partial \mathbf{z}_1}$
- $d\mathbf{x} = \frac{\partial J}{\partial \mathbf{x}} = d\mathbf{z}_1 \frac{\partial \mathbf{z}_1}{\partial \mathbf{x}} = d\mathbf{z}_1 \mathbf{W}_1^T$

- Other gradients:

- $d\mathbf{W}_2 = \frac{\partial J}{\partial \mathbf{W}_2} = \frac{\partial J}{\partial \mathbf{z}_2} \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_2} = \mathbf{h}^T d\mathbf{z}_2$
- $db_2 = \frac{\partial J}{\partial b_2} = \frac{\partial J}{\partial \mathbf{z}_2} \frac{\partial \mathbf{z}_2}{\partial b_2} = d\mathbf{z}_2$ (summed over the first dimension)
- $d\mathbf{W}_1 = \frac{\partial J}{\partial \mathbf{W}_1} = \frac{\partial J}{\partial \mathbf{z}_1} \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_1} = \mathbf{x}^T d\mathbf{z}_1$
- $db_1 = \frac{\partial J}{\partial b_1} = \frac{\partial J}{\partial \mathbf{z}_1} \frac{\partial \mathbf{z}_1}{\partial b_1} = d\mathbf{z}_1$ (summed over the first dimension)

2.2.4 2.d) Number of parameters

Number of parameters: $(D_x + 1)H + (H + 1)D_y$

2.3 3. word2vec

2.3.1 3.a) Derivative of the skipgram 1 — with respect to v_c

Skipgram: given a center word c ($v_c \in \mathbb{R}^n$, $v_c = \mathbf{V}\mathbf{c}$, $\mathbf{c} \in \mathbb{R}^{|V|}$ is one-hot-vector of word c), predict the surrounding context words ($2m$ words).

Word prediction softmax function:

$$\hat{y}_o = p(o|c) = \frac{\exp(\mathbf{u}_o^T \mathbf{v}_c)}{\sum_w \exp(\mathbf{u}_w^T \mathbf{v}_c)}$$

where \mathbf{o} is the expected output word, and $\mathbf{u}_w \in \mathbb{R}^n$ ($w \in [1, W]$) are the output word vectors.

$$J_{softmax-CE}(\mathbf{o}, \mathbf{v}_c, \mathbf{U}) = CE(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_i \mathbf{y} \log \hat{\mathbf{y}} = -\mathbf{u}_o^T \mathbf{v}_c + \log \left(\sum_w \exp(\mathbf{u}_w^T \mathbf{v}_c) \right) = -\mathbf{u}_o^T \mathbf{v}_c + \log \left(\sum [\exp(\mathbf{U}\mathbf{v}_c)] \right)$$

where $\mathbf{U} \in \mathbb{R}^{|V| \times n}$ is the output word matrix (n : dimension of embedding space; $|V|$: dimension of vocabulary).

$$\frac{\partial}{\partial \mathbf{v}_c} CE(\mathbf{y}, \hat{\mathbf{y}}) = -\mathbf{u}_o^T + \sum_i \mathbf{u}_i^T \frac{\exp(\mathbf{u}_i^T \mathbf{v}_c)}{\sum_w \exp(\mathbf{u}_w^T \mathbf{v}_c)} = -\mathbf{u}_o^T + \sum_i \mathbf{u}_i^T \hat{y}_i = (\hat{\mathbf{y}} - \mathbf{y})^T \mathbf{U}$$

since $\mathbf{u}_o = \mathbf{U}^T \mathbf{y}$, $\sum_i \mathbf{u}_i^T \hat{y}_i = \hat{\mathbf{y}}^T \mathbf{U}$, where \mathbf{y} is the one-hot-vector with 1 at o and 0 elsewhere.

2.3.2 3.b) Derivative of the skipgram 2 — with respect to \mathbf{u}_w

$$\frac{\partial}{\partial \mathbf{U}} CE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\partial}{\partial \mathbf{u}_i} CE(\mathbf{y}, \hat{\mathbf{y}}) = \hat{y}_i \mathbf{v}_c + \begin{cases} -\mathbf{v}_c & \mathbf{u}_i = \mathbf{u}_o \\ 0 & \mathbf{u}_i \neq \mathbf{u}_o \end{cases}$$

$$\frac{\partial}{\partial \mathbf{U}} CE(\mathbf{y}, \hat{\mathbf{y}}) = (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{v}_c^T$$

2.3.3 3.c) Derivative of the skipgram 3