DS-GA 1008: Deep Learning, Spring 2019
Homework Assignment 1

Due: 6pm on Friday, Feb 15, 2019

```
He who learns but does not think is lost.
He who thinks but does not learn is in great danger.
Confucius (551 - 479 BC)
```

1. Backprop

Backpropagation or "backward propagation through errors" is a method which calculates the gradient of the loss function of a neural network with respect to its weights.

1.1 Warm-up

The chain rule is at the heart of backpropagation. Assume you are given input $\boldsymbol{x}$ and output $\boldsymbol{y}$, both in $\mathbb{R}^2$, and the error backpropagated to the output is $\frac{\partial L}{\partial \boldsymbol{y}}$. In particular, let

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b},$$

where $\boldsymbol{W} \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^2$. Give an expression for $\frac{\partial L}{\partial \boldsymbol{W}}$ and $\frac{\partial L}{\partial \boldsymbol{b}}$ in terms of $\frac{\partial L}{\partial \boldsymbol{y}}$ and $\boldsymbol{x}$ using the chain rule.

Solution: Using the Denominator-Layout notation, we have

$$\frac{\partial L}{\partial \boldsymbol{W}} = \begin{bmatrix} \frac{\partial L}{\partial W_{11}} & \frac{\partial L}{\partial W_{12}} \\ \frac{\partial L}{\partial W_{21}} & \frac{\partial L}{\partial W_{22}} \end{bmatrix}, \frac{\partial L}{\partial \boldsymbol{y}} = \begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \end{bmatrix}, \frac{\partial L}{\partial \boldsymbol{b}} = \begin{bmatrix} \frac{\partial L}{\partial b_1} \\ \frac{\partial L}{\partial b_2} \end{bmatrix}.$$

Using the chain rule, we obtain for $i, j \in \{1, 2\}$:

$$\frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial W_{ij}} = \frac{\partial L}{\partial y_i}\frac{\partial \big(W_{i1}x_1 + W_{i2}x_2 + b_i\big)}{\partial W_{ij}} = \frac{\partial L}{\partial y_i}x_j.$$

Hence,

$$\frac{\partial L}{\partial \boldsymbol{W}} = \frac{\partial L}{\partial \boldsymbol{y}} \otimes \boldsymbol{x}^T,$$

where $\otimes$ is the outer product. Similarly, for $i \in \{1, 2\}$:

$$\frac{\partial L}{\partial b_i} = \frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial b_i} = \frac{\partial L}{\partial y_i}\frac{\partial \big(W_{i1}x_1 + W_{i2}x_2 + b_i\big)}{\partial b_i} = \frac{\partial L}{\partial y_i}$$

and we conclude that

$$\frac{\partial L}{\partial \boldsymbol{b}} = \frac{\partial L}{\partial \boldsymbol{y}}.$$

DS-GA 1008: Deep Learning, Spring 2019
Homework Assignment 1

Due: 6pm on Friday, Feb 15, 2019

1.2. Softmax

Multinomial logistic regression is a generalization of logistic regression into multiple classes. The softmax expression is at the crux of this technique. After receiving n unconstrained values, the softmax function normalizes these values to n values that all sum to 1. This can then be perceived as probabilities attributed to the various classes by a classifier. Your task here is to back-propagate error through this module. The softmax expression which indicates the probability of the j-th class is as follows:

$$\mathbb{P}(z = j \mid \boldsymbol{x}) = y_j = \frac{\exp(\beta x_j)}{\sum_i \exp(\beta x_i)} \tag{1}$$

What is the expression for $\frac{\partial y_j}{\partial x_i}$? (Hint: Answer differs when $i = j$ and $i \neq j$).

Note that the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ aren't scalars but vectors. While $\boldsymbol{x}$ represents the n values input to the system, $\boldsymbol{y}$ represents the n probabilities output from the system. Therefore, the expression y_j represents the j-th element of $\boldsymbol{y}$.

Solution: For $i \neq j$:

$$\frac{\partial y_j}{\partial x_i} = -\frac{\beta \exp[\beta(x_j + x_i)]}{(\sum_k \exp(\beta x_k))^2}.$$

For $i = j$:

$$\frac{\partial y_j}{\partial x_j} = \frac{\beta \exp(\beta x_j)[(\sum_k \exp(\beta x_k)) - \exp(\beta x_j)]}{(\sum_k \exp(\beta x_k))^2}.$$