16-720B Homework 4 Write-up

Gu, Qiao

November 12, 2019

Q1.1

The given translation is applied to all x_j for $j \in \mathbb{R}$. Then for each $x_i i$

$$softmax(x_i + c) = \frac{e^{x_i+c}}{\sum_j e^{x_j+c}} = \frac{e^{x_i} e^c}{\sum_j e^{x_j} e^c} = \frac{e^{x_i} e^c}{(\sum_j e^{x_j}) e^c} = \frac{e^{x_i}}{\sum_j e^{x_j}} = softmax(x_i). \tag{1}$$

This shows that softmax is invariant to translation.

When $c = -\max x_i$, all $x_i + c$ is between zero and one, and the difference between e^{x_I} is relatively small. And when $c = 0$, e^{x_i} can be exponentially large and may cause numerical instability.

Q1.2

- Each element of softmax $softmax(x_i)$ is in range $(0, 1)$, and the sum over all elements is $\sum_j softmax(x_j) = 1$.
- Probability.
- $s_i = e^{x_i}$ is to map each x_i to its probability weight. $S = \sum s_i$ is find the sum of the weights. $softmax(x_I) = \frac{1}{S} x_i$ is to normalize each weight by all weights to get probability.

Q1.3

Each layer of a neural network can be written mathmatically as $f_i(\mathbf{x}) = \mathbf{W}_i\mathbf{x}+\mathbf{b}$, and thus is we concatenate n layers together without non-linear layers, we get the output as

$$
\begin{aligned}
\mathbf{y} &= f_n(f_{n-1}(\cdots f_1(\mathbf{x})\cdots)) && (2)\\
&= \mathbf{W}_n(\mathbf{W}_{n-1}(\cdots(\mathbf{W}_1\mathbf{x}+\mathbf{b}_1)\cdots)+\mathbf{b}_{n-1})+\mathbf{b}_n && (3)\\
&= \mathbf{W}_n\mathbf{W}_{n-1}\cdots\mathbf{W}_1\mathbf{x}+\mathbf{W}_n\mathbf{W}_{n-1}\cdots\mathbf{W}_2\mathbf{b}_1+\mathbf{W}_n\mathbf{W}_{n-1}\cdots\mathbf{W}_3\mathbf{b}_2+\cdots+\mathbf{W}_n\mathbf{b}_{n-1}+\mathbf{b}_n && (4)\\
&= \mathbf{W}\mathbf{x}+b, && (5)
\end{aligned}
$$

where $\mathbf{W} = \mathbf{W}_n\mathbf{W}_{n-1}\cdots\mathbf{W}_1$ and $\mathbf{b} = \mathbf{W}_n\mathbf{W}_{n-1}\cdots\mathbf{W}_2\mathbf{b}_1+\mathbf{W}_n\mathbf{W}_{n-1}\cdots\mathbf{W}_3\mathbf{b}_2+\cdots+\mathbf{W}_n\mathbf{b}_{n-1}+\mathbf{b}_n$. This can be regarded as a single linear layer, and the whole network is equivalent to linear regression.

Q1.4

$$
\begin{aligned}
\frac{d\sigma(x)}{dx} &= \frac{d}{dx}\frac{1}{1+e^{-x}} && (6)\\
&= (-1)\frac{1}{(1+e^{-x})^2}\frac{d}{dx}(1+e^{-x}) && (7)\\
&= (-1)\frac{1}{(1+e^{-x})^2}(-e^{-x}) && (8)\\
&= \frac{1+e^{-x}-1}{(1+e^{-x})^2} && (9)\\
&- \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2} && (10)\\
&= \frac{1}{1+e^{-x}}(1-\frac{1}{1+e^{-x}}) && (11)\\
&= \sigma(x)(1-\sigma(x)) && (12)
\end{aligned}
$$

Q1.5

The loss function is unknown and therefore we assume $\frac{\partial J}{\partial y_i} = \delta_i$. Therefore

$$\frac{\partial J}{\partial W_{ij}} = \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial W_{ij}} = \delta_i x_j \tag{13}$$

$$\frac{\partial J}{\partial x_j} = \sum_{i=0}^{k} \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial x_j} = \delta_i \sum_{i=0}^{k} W_{ij} \tag{14}$$

$$\frac{\partial J}{\partial b_i} = \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial b_i} = \delta_i \tag{15}$$

And then we can further rewrite it to matrix form

$$\frac{\partial J}{\partial \mathbf{W}} = \delta \mathbf{x}^T \tag{16}$$

$$\frac{\partial J}{\partial \mathbf{x}} = \mathbf{W}^T \delta \tag{17}$$

$$\frac{\partial J}{\partial \mathbf{b}} = \delta \tag{18}$$

Q1.6

1. As shown in Figure. 1, when the input to the sigmoid x is far away from zero, the magnitude of the gradient becomes very close to zero and thus the gradient from higher layers are scaled by a very small number, even "vanishing". Therefore, when we update the network weights, the changes will be very small.

2. The output range of sigmoid function is $(0, 1)$ and the output range of $tanh(x)$ is $(-1, 1)$. If our input data are centered at 0, the output given by tanh are also centered at 0, which will make the input to different layers consistent.

3. From Figure. 1, we can see that $\tanh(x)$ has a stronger gradient (larger magnitude) than $\sigma(x)$ does.

4. $\tanh(x) = 2\sigma(2x) - 1$ as

$$\begin{aligned} \sigma(x) = \frac{1}{1+e^{-x}} \Rightarrow \sigma(2x) = \frac{1}{1+e^{-2x}} \Rightarrow 2\sigma(2x) = \frac{2}{1+e^{-2x}} \\ \Rightarrow 2\sigma(2x) - 1 = \frac{2-1-e^{-2x}}{1+e^{-2x}} = \frac{1-e^{-2x}}{1+e^{-2x}} = \tanh(x) \end{aligned} \tag{19}$$

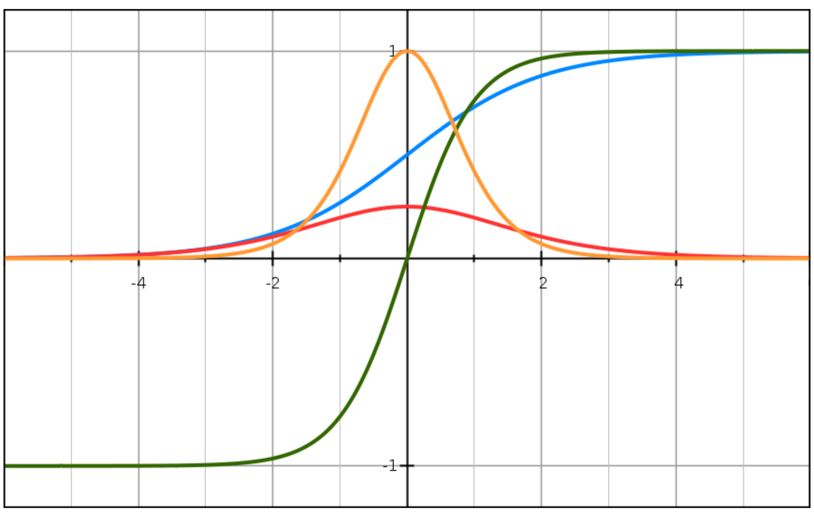

Figure 1: The plot of sigmoid function $\sigma(x)$ (in blue), its gradient $(1 - \sigma(x))\sigma(x)$ (in red), the $\tanh(x)$ function (in green) and its gradient $1 - \tanh^2(x)$ (in orange).

0.1 Q2.1.1

If the weights and biases of every layer are initialized with zeros and the activition function produces zero output given zero input, the output of the whole network will be zeros regardless the input data. The gradients of the network therefore will be zeros during training and the network weights will remain zeros after training.