EE488B Special Topics in EE <Deep Learning and AlphaGo> September 27, 2017
EE, KAIST, Fall 2017 Due: 1:00pm, October 11

Problem Set #2

Warning: Homeworks will not be graded if submitted after the deadline. For all problems, show detailed reasoning.

0. (Reading assignment) DL Book Chapter 7

1. (Back-propagation for 100-layer network – 10 points) Implement the training algorithm for Example 1 in page 42 in lecture notes #6. Assume $m = 2$ and $\mathbf{x} = \mathbf{y} = (-0.5, 0.5)^\mathsf{T}$, i.e., we want to train a neural network to output 0.5 when the input is 0.5 and to output -0.5 when the input is -0.5. Assume $l = 100$, i.e., 100 layers. We want to see if such a deep neural network can be trained. Assume also $\alpha = 0.1$ (learning rate), MaxIter $= 1000$, and $\phi^{(k)}(x) = \tanh(x)$, $k = 1, 2, \dots, l$. Note that $\tanh'(x) = \operatorname{sech}^2(x) = \frac{4}{(e^x+e^{-x})^2}$, which can be used in your code. Include codes and plots in your homework. You may use any programming language.

(a) Run your code by initializing all weights as 1.0. Plot the cost J defined in page 40 of lecture notes #6 as a function of the iteration. Cost should converge to 0, i.e., training should be successful. Print out $h_{k,i}$ and $g_{k,i}$, $i = 1, 2$ obtained after just one iteration as functions of $k = 1, 2, \dots, l$.

(b) Run your code by initializing all weights as 5. This time, the cost will be stuck at about 0.125, i.e., training will fail. Print out $h_{k,i}$ and $g_{k,i}$, $i = 1, 2$ obtained after just one iteration as functions of $k = 1, 2, \dots, l$. How are they different from $h_{k,i}$'s and $g_{k,i}$'s you obtained in (a)?

(c) Explain why training fails in (b). Try to see how $h_{k,i}$'s behave as k increases (forward propagation) and how the gradients $g_{k,i}$'s behave as k decreases (backward propagation). You will be able to see $g_{k,i}$'s vanish as k decreases, which is known as the vanishing gradient problem. Why does the gradient vanish as k decreases? How does the actual gradient $\nabla_\mathbf{w} J$ behave?

(d) Run your code by initializing all weights as 0.9. This time, the cost will be stuck at about 0.125, i.e., training will fail. Print out $h_{k,i}$ and $g_{k,i}$, $i = 1, 2$ obtained after just one iteration as functions of $k = 1, 2, \dots, l$. How are they different from $h_{k,i}$'s and $g_{k,i}$'s you obtained in (a) and (c)?

(e) Explain why training fails in (d). Try to see how $h_{k,i}$'s behave as k increases (forward propagation) and how the gradients $g_{k,i}$'s behave as k decreases (backward propagation). You will be able to see $g_{k,i}$'s vanish as k decreases. Is this the only reason why training fails? How does the actual gradient $\nabla_\mathbf{w} J$ behave?

2. (Back-propagation with bias terms – 10 points)

(a) Let's introduce a bias term in each layer in Example 1 in pages 40–42 in lecture notes #6, i.e., $\mathbf{h}_k = \phi^{(k)}(\mathbf{h}_{k-1} w_k + \mathbf{1} b_k)$, $k = 1, 2, \dots, l$, where $\mathbf{1}$ is an all-one vector of length m and $b_k \in \mathbb{R}$ is the bias term at the k-th layer. Note that you need $\mathbf{1}$ since there are m training examples. Define $\mathbf{u}_k = \mathbf{h}_{k-1} w_k + \mathbf{1} b_k$ and $\mathbf{g}_k = \nabla_{\mathbf{u}_k} J$, $k = 1, 2, \ldots, l$. Express $\mathbf{g}_l$ in terms of other variables such as $\mathbf{u}_l$, $\mathbf{h}_l$, and $\mathbf{y}$. Express $\mathbf{g}_{k-1}$ in terms of other variables including $\mathbf{g}_k$, i.e., backward propagation. Express $\frac{\partial J}{\partial w_k}$, $\frac{\partial J}{\partial b_k}$, $k = 1, 2., \ldots, l$ using $\mathbf{h}_k$ and $\mathbf{g}_k$. Show the whole training algorithm including forward and backward propagations and gradient descent as a pseudo code.

(b) In Example 2 in pages 43–46 in lecture notes #6, show that $\mathbf{G}^{(2)} = -\frac{1}{m}\mathbf{Y}^T$, where $\mathbf{Y}_{j,i} = 1$ if $j = y_i$ and 0 otherwise, which was defined in page 44 of lecture notes #6.

(c) Let's introduce a bias term in the first layer in Example 2 in pages 43–46 in lecture notes #6, i.e., $\mathbf{H} = \phi(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{1}\mathbf{b}^T)$, where $\mathbf{1}$ is an all-one vector of length m and $\mathbf{b}$ is the bias vector of length n_1. Note that you need $\mathbf{1}$ since there are m training examples. Define $\mathbf{U}^{(1)} = \mathbf{X}\mathbf{W}^{(1)} + \mathbf{1}\mathbf{b}^T$ and $\mathbf{G}^{(1)} = \nabla_{\mathbf{U}^{(1)}} J_{\text{MLE}}$. Assume the other variables such as $\mathbf{U}^{(2)}$ and $\mathbf{G}^{(2)}$ are defined the same way as done in Example 2. Express $\mathbf{G}^{(1)}$ in terms of other variables such as $\mathbf{U}^{(1)}$, $\mathbf{G}^{(2)}$, and $\mathbf{W}^{(2)}$. Express $\nabla_{\mathbf{W}^{(1)}} J_{\text{MLE}}$ and $\nabla_{\mathbf{b}} J_{\text{MLE}}$ using other variables such as $\mathbf{X}$ and $\mathbf{G}^{(1)}$.