HW Assignment 6, Theory

Problems marked with a $(*)$ are mandatory for ITCS 8156 students. Bonus problems are optional, solving them will result in extra points.

1 Activations Functions (10 points)

Compute the derivative of the logistic sigmoid $\sigma(x)$, hyperbolic tangent $tanh(x)$, and ReLU's $ramp(x)$ activation functions. For logistic sigmoid and hyperbolic tangent, express the derivative in terms of the original function.

2 Universal Approximation $(*)$ (30 + 15 points)

Let NN be a neural network with 2 input units, 1 hidden layer with h units using sigmoid as activation function, and 1 binary logistic regression unit using logistic sigmoid as output function. Furthermore, consider a training set that contains the following 4 examples i.e., the truth table of the logical XOR function:

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	0

1. Is there a neural network NN with 1 hidden layer that perfectly classifies this training set? Prove your answer. If the answer is yes, what is the minimum h for which there is a network with h hidden neurons that fits the training data? Prove it.
2. Consider the same neural network NN, but without activation functions in the hidden layer. Answer the same questions as at item (a) above.
3. **Bonus**: Consider the same neural network NN, but this time using ReLU neurons in the hidden layer. Answer the same questions as at item (a) above.

3 Gradients & Computation Graphs (30 points)

Consider a 3D vector $\mathbf{x} = [x_1, x_2, x_3]^T$ and let $\mathbf{x} \circ \mathbf{x} = [x_1^2, x_2^2, x_3^2]^T$ be the element-wise square of $\mathbf{x}$. Let $h(\mathbf{x})$ be a function computed as follows:

$$
\begin{aligned}
h(x) &= \sigma(v_1 a_1(\mathbf{x}) + v_2 a_2(\mathbf{x})) \\
a_1(\mathbf{x}) &= z_1^2(\mathbf{x}) \\
z_1(\mathbf{x}) &= \mathbf{w}^T \mathbf{x} \\
a_2(\mathbf{x}) &= tanh(z_2(\mathbf{x})) \\
z_2(\mathbf{x}) &= \mathbf{u}^T (\mathbf{x} \circ \mathbf{x})
\end{aligned}
$$

where $\mathbf{w} = [w_1, w_2, w_3]^T$, $\mathbf{u} = [u_1, u_2, u_3]^T$.

1. Show the computation graph of $h(\mathbf{x})$, similar to how was done in class.
2. Use the chain rule to compute the gradient of h with respect to x_2. Show all your derivation steps and the final formula for the gradient as a product of various factors resulting from the application of the chain rule.

4 Backpropagation (20 + 10 points)

Consider the vectorized backpropagation algorithm for regression, shown on slides 41 and 42.

1. Specify the shape for each matrix and vector appearing in the algorithm, using the notation introduced in class and assuming the gradient is computed for the loss on just one training example, i.e. $J(W, b, \mathbf{x}, y)$.
2. Specify the shape for each matrix and vector appearing in the algorithm, this time assuming the gradient is computed for the total loss over all training examples, i.e. $J(W, b, X, \mathbf{y})$.
3. **Bonus**: Compute the time complexity of running vectorized backpropagation for the total loss $J(W, b, X, \mathbf{y})$. Compare this with the time complexity of computing the same gradient numerically. For simplicity, you can assume that there is only one unit in the output layer, and all the other layers have the same size, i.e. $s_1 = s_2 = ... = s_{n_l-1} = S$.

5 Vizualization of Hidden Units (*) (15 points)

Prove that the input vector $\mathbf{x}$ ($||\mathbf{x}||_2 \leq 1$) that maximally activates the hidden layer unit $a_i^{(2)}$ has the form shown below:

$$x_j = \frac{W_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{s_1}(W_{ij}^{(1)})^2}} \tag{1}$$

6 Submission

Submit your responses on Canvas as one file named `theory.pdf`. It is recommended to use an editor such as Latex or Word or Jupyter-Notebook that allows editing and proper formatting of equations. Alternatively, if you choose to write your solutions on paper, submit an electronic scan / photo of it on Canvas. Make sure that your writing is legible and the scan has good quality.