

MMI 727 Deep Learning: Methods and Applications

Assignment 1

1. Backpropagation

Sigmoid, softmax, tanh and rectified linear unit (ReLU) are commonly used activation functions in deep learning. They are defined as:

$$\begin{array}{c} \text{Sigmoid} \\ x_{\text{out}} = \sigma(x_{\text{in}}) = \frac{1}{1 + e^{-x_{\text{in}}}} \end{array}$$

$$\begin{array}{c} \text{Softmax} \\ (X_{\text{out}})_i = \frac{e^{-\beta(X_{\text{in}})_i}}{\sum_k e^{-\beta(X_{\text{in}})_k}} \end{array}$$

$$\begin{array}{c} \text{Tanh} \\ x_{\text{out}} = \tanh(x_{\text{in}}) = \frac{\exp(2x_{\text{in}}) - 1}{\exp(2x_{\text{in}}) + 1} \end{array}$$

$$\begin{array}{c} \text{ReLU} \\ x_{\text{out}} = \text{rect}(x_{\text{in}}) = \max(0, x_{\text{in}}) \end{array}$$

For each activation function:

- What is the expression for $\frac{\partial x_{\text{out}}}{\partial x_{\text{in}}}$ (pen&paper or digital, write all intermediate steps) (5 pts)
- Plot the function graph in a new notebook. Use Tensorflow for activation functions. (10 pts)
- What are the advantages and disadvantages of each activation function. In which cases it is useful or not? Discuss. (5 pts)

2. Parameter tuning

Using the Lab notebook, make all changes below.

- Change learning rate. You must try at least one lower and one higher value (e.g. 0.0005 and 50) than default. (5 pts)
- Change the optimizer. Try at least 2 different optimizers e.g. AdagradOptimizer or AdamOptimizer. (5 pts)
- Change batch-size at least twice. (5 pts)
- Visualize weights in 28x28 resolution for each class. (5 pts)

Plot accuracy graph for each experiment for 1000 iterations. How do these changes affect accuracy and duration? Discuss.

3. Network model

In the lab notebook the network has single layer. We can observe whether adding more layers affects performance. Using the lab notebook:

- a) Original single-layer network. (10 report pts)
- b) Add a fully connected layer (identical to single layer) to network and make it 2-layer network. What is the outcome? What is the reason? Try to make it work (>%90) (10 report + 10 code pts)
- c) Add two fully connected layers (identical to single layer) to network and make it 3-layer network. Try to get better result with this network by changing hyperparameters. (10 report + 10 code pts)
- d) Implement the 3-layer network above with high level `tf.layers` module. Don't forget to delete unnecessary cells. (*Hint : `tf.layers.dense`*) (10 code pts)

Plot accuracy graph for each for 1000 iterations. Discuss the accuracy and durations.

Submit problem solutions for backpropagation (pen&paper or digital), code files of 1.b, 3.b, 3.c and 3.d (only “.ipynb” notebook files) and the report to ODTUclass. We must be able to reproduce your results.