Assignment 4: Convolutional Neural Networks
MA-INF 2313: Deep Learning for Visual Recognition

Due Date Theoretical:	12.12.2019
Due Date Programming:	12.12.2019
Assistant:	Soumajit Majumder

1 Theoretical Exercises

1. *(8 pts) Backpropagation through Convolutional Layer :*

   The output (before the nonlinearity) of a convolutional layer with weights $w \in \mathbb{R}^{k \times k}$ with respect to input image (or intermediate feature maps) $x \in \mathbb{R}^{d \times d}$ is given by the convolution operation $y = x * w$. Note that for now, we consider only a single input and output channel.

   Derive the quantities $\frac{\delta y}{\delta w}$ and $\frac{\delta y}{\delta x}$.

2. *(7 pts) Adding constraints on the weights of a CNN:*

   Consider a neural network with general feed-forward structure, where each unit computes a weighted sum of the inputs,

   $$a_j = \sum_i w_{ji} z_i,$$

   where z_i is the activation of a unit, or input, that sends a connection to unit j, and w_{ji} is the weight associated with that connection. Let h be a nonlinear activation function, then $z_i = h(a_i)$. We now wish to modify this neural network such that multiple weights are constrained to have the same value. Discuss how the standard back-propagation algorithm must be modified in order to ensure that the constraints are satisfied when evaluating the derivatives of the error function with respect to the adjustable parameters in the network.

2 Programming Exercises

In this programming exercise, you will set up and train a CNN to identify and distinguish between hand-written digits from the MNIST dataset[1]. Build a CNN with at least two convolutional layers (feel free to choose any kernel size and the number of output feature

[1] `https://paperswithcode.com/sota/image-classification-on-mnist`

maps). Assume that each convolutional layer consists of a convolution operation followed by a relu and max-pooling (stride of 2, filter size 2×2, or depending on your choice of kernel size, padding and stride).

1. Train your CNN using different learning rates (other than opting for the library default). For each learning rate observe how the classification error decreases over each iteration. Pick the best learning rate and keep it fixed for the following steps.
2. Plot the training loss over the iterations and report the final test accuracy.
3. Add regularization to your CNN in the form of dropout layers and weight decay. Train the CNN on the training set and report the final test accuracy.

Now, using the same CNN, perform the classification task on the more challenging CIFAR-10 dataset. Report the test accuracy.