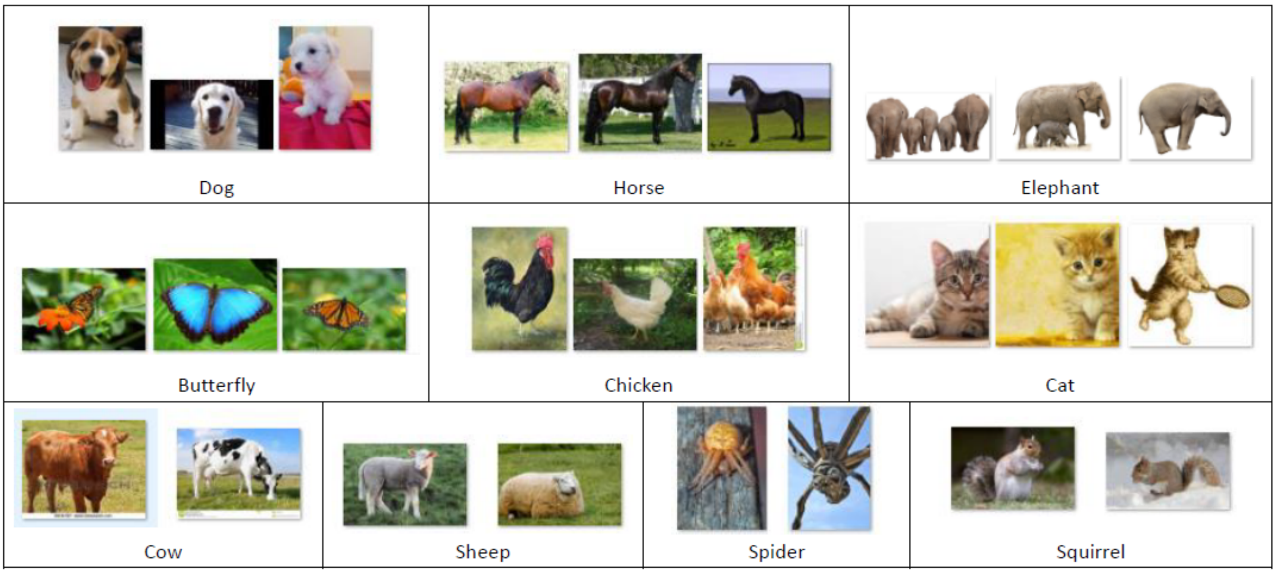
Deep Learning (Homework 2)

Due date : 5/17/2019

- High-level API are forbidden in this homework, such as Keras, slim, TFLearn, etc. You should implement the forward computation by yourself.
- Homework submission – Please zip each of your source code and report into a single compress file and name the file using this format : HW2_StudentID_StudentName.zip (rar, 7z, tar.gz, ... etc are *not* acceptable)

1 Convolutional Neural Network for Image Recognition

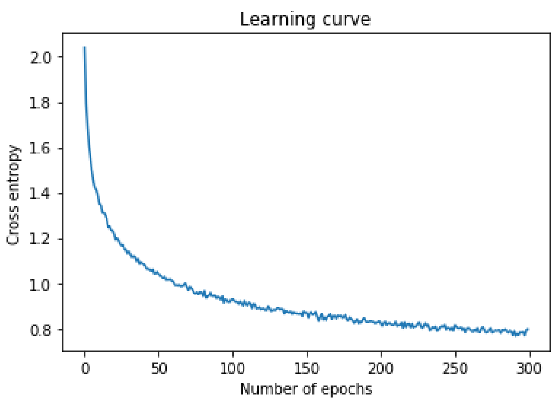
In this exercise, you will construct a Convolutional Neural Network (CNN) for image recognition by using the Animals-10 dataset. The original dataset contains about 28K animal images with medium quality where there are 10 categories/animals. The 10 categories are *Dog*, *Horse*, *Elephant*, *Butterfly*, *Chicken*, *Cat*, *Cow*, *Sheep*, *Spider*, and *Squirrel*. Some example images of 10 categories are shown below.

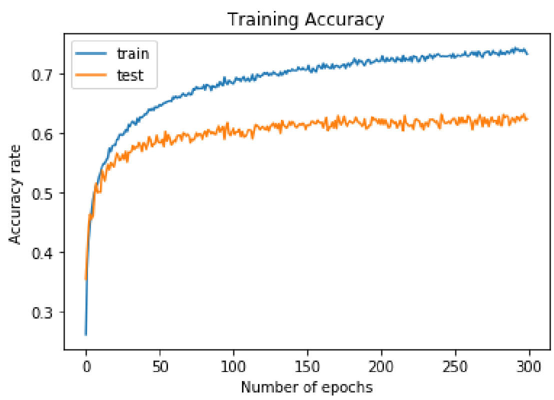


The collected dataset is a subset of the original dataset where the number of images for training and testing in different categories is listed in the following table. This dataset is accessible in the link. You should perform preprocessing over the images such as resizing or cropping by yourself in the implementation.

i. Please describe in details how to preprocess the images in Animal-10 dataset with different resolutions and explain why. You have to submit your preprocessing code.

Category	Training	Testing
Dog	1000	400
Horse	1000	400
Elephant	1000	400
Butterfly	1000	400
Chicken	1000	400
Cat	1000	400
Cow	1000	400
Sheep	1000	400
Spider	1000	400
Squirrel	1000	400
Total	10000	4000

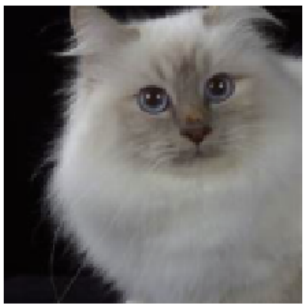




ii. Please implement a CNN for image recognition. You need to design the network architecture, describe your network architecture and analyze the effect of different settings including stride size and filter/kernel size. Plot learning curve, classification accuracy of training and test sets as displayed in above figure.

iii. Show some examples of correctly and incorrectly classified images, list your accuracy for each test classes (similar to the following figure), and discuss about your results.

labels: cat , predict: cat



labels: dog , predict: cow

```
Accuracy of classes
      dog : 64 %
    horse : 58 %
 elephant : 54 %
butterfly : 69 %
  chicken : 68 %
      cat : 48 %
      cow : 50 %
    sheep : 50 %
   spider : 77 %
 squirrel : 46 %
```

2 Recurrent Neural Network for Prediction of Paper Acceptance

In this exercise, you will implement a Recurrent Neural Network (RNN) model for Prediction of Paper Acceptance by using the machine learning conference papers from ICLR (International Conference on Learning Representations).

This dataset contains all the titles of the papers from ICLR 2017 and ICLR 2018. This dataset is separated into two files. ICLR_accepted.xlsx contains all the accepted papers, and ICLR_rejected.xlsx contains all the rejected papers. Please build the test data from these two files by using the first 50 papers from each of these two files.

Build your own dictionary and use the word embedding technique for data preprocessing. For example, given the sequence data {"NCTU is good"} and we can build a dictionary { 0 : "NCTU", 1 : "is", 2 : "good" }. After the dictionary is built, we can convert the sequence to [0, 1, 2]. Then, we use the word embedding technique to deal with the sequence.

The dataset came from
https://openreview.net/group?id=ICLR.cc/2017/conference
https://openreview.net/group?id=ICLR.cc/2018/Conference

i. Please construct a standard RNN for acceptance prediction. For classification purpose, we aim to minimize the cross entropy error function, which is defined as

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x}_n, \mathbf{w})$$

where $K = 2$ in this task. Minimize the objective function $E(\mathbf{w})$ by running the error backpropagation algorithm using the stochastic gradient descent

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}).$$

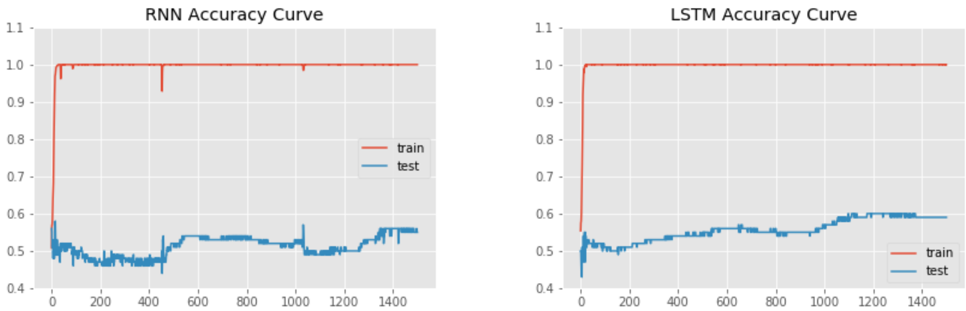
Design the network architecture by yourself, including number of hidden layers, number of hidden units, learning rate, number of iterations and mini-batch size. You have to show your (a) learning curve, (b) training error rate and (c) test error rate in the report.

- crop or append with 0 to make the sequence fixed in length, and the size is set to be 10.
- feel free to use any optimizer (SGD, Adadelta , ADAM and so on).
- you can implement the model by built-in functions, such as *tf.contrib.rnn, tf.nn.static_rnn, torch.nn.RNN* and so on.
- you will gain **BONUS point** if you implement the model by yourself

ii. Redo i. by using the Long Short-Term Memory network (LSTM), which is formulated by

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \\
\mathbf{g}_t &= \tanh(\mathbf{W}_g[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_g) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}
$$

iii. Please discuss the difference between i. and ii. The following figure show an example of result.



Accuracy curve of RNN (Left) and LSTM (Right). Red curve stands for Training and blue curve stands for Testing