Week 8

Lecture

- CNN
  - Architecture (**INPUT – CONV – RELU – POOL - FC**)
- RNN
  - Neural work + memory
  - LSTM (Input gate, forget gate, output gate)
  - GRU (reset gate, update gate)
  - Difference between LSTM and GRU

Tutorial

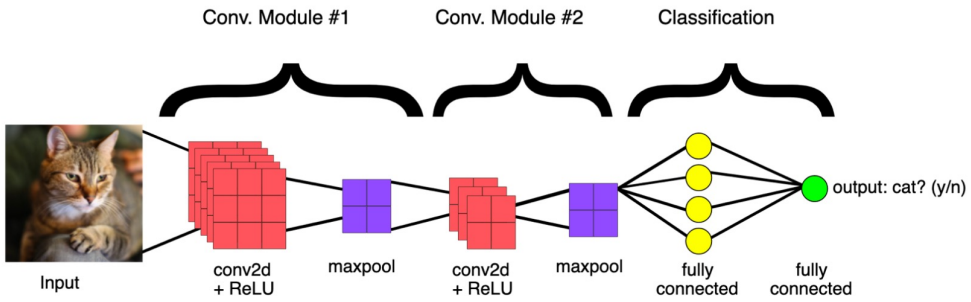
Task 1: Using CNN for MNIST data.
Step 1: Load MNIST data and create validation set.
Step 2: Define the CNN model by using Keras.
Step 3: Optimization and evaluation.

Task 2: Using LSTM for movie review.
Step 1: load the movie review data.
Step 2: Define LSTM model.
Step 3: Optimization and evaluation.

TODO:
1. Create CNN for CIFAR-10

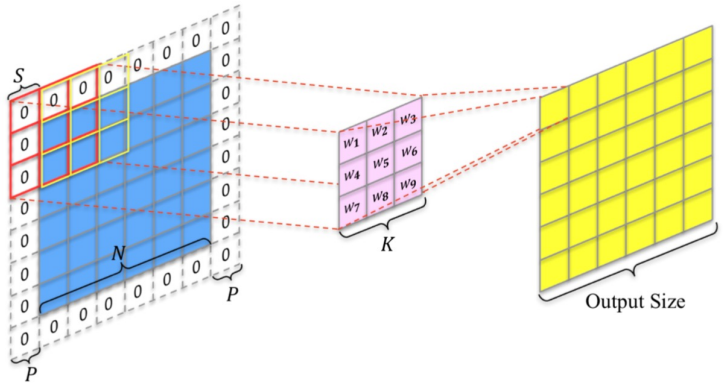
Deep Learning

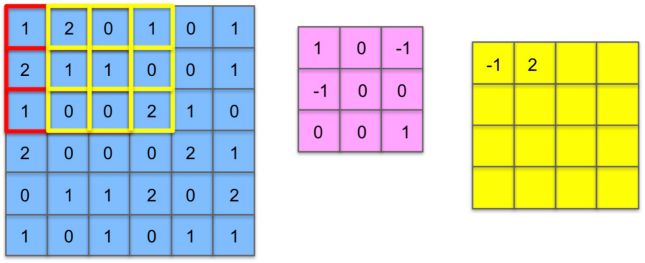
CNN



Basic CNN structure:
- Convolutional Layer
- Pooling
- Fully-connected Layer

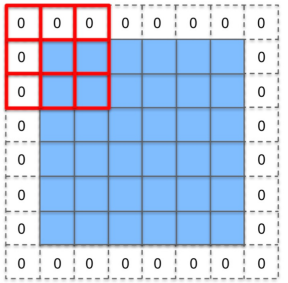
$$\text{Output Size} = \frac{N+2P-K}{S} + 1$$





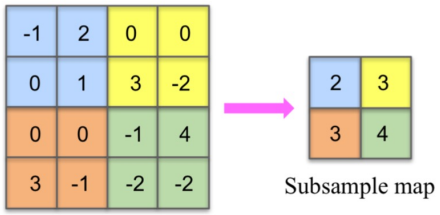
Stride = 1

Zero padding (pad = 1)



improves performance by keeping information at the borders

Max pooling
- Filter size: (2,2)
- Stride: (2,2)
- Pooling ops: max(·)

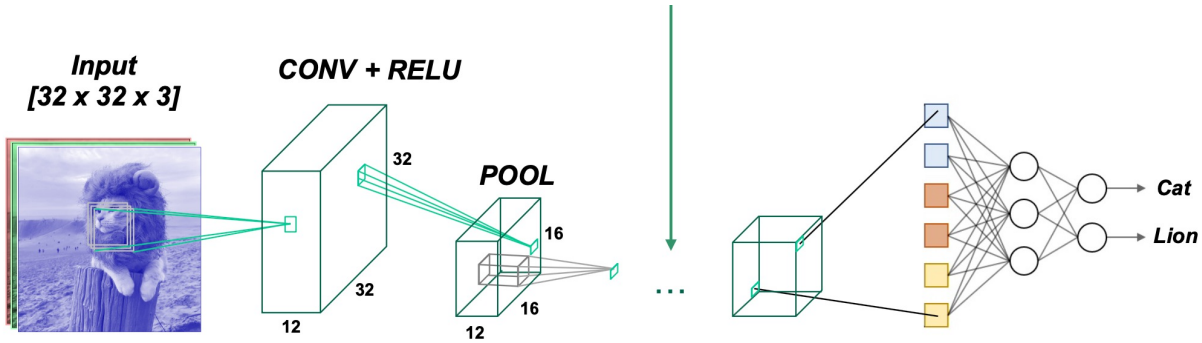


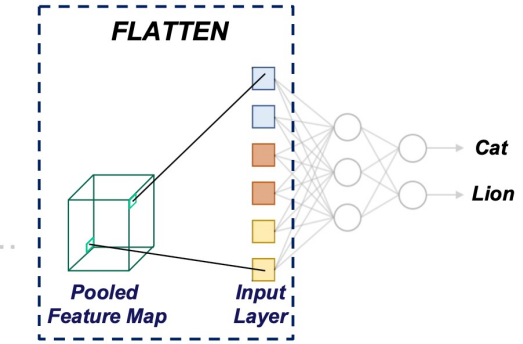
Deep Learning

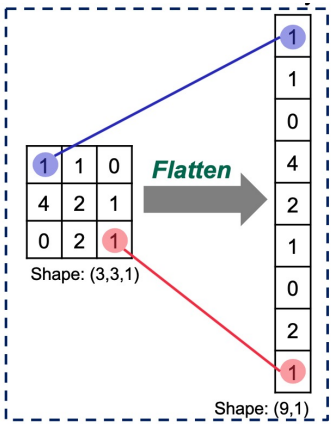
CNN

FLATTEN (Flattening) Layer

In between the convolutional layer and the fully connected layer, there is a '***Flatten***' layer. Flattening transforms a multi-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier.

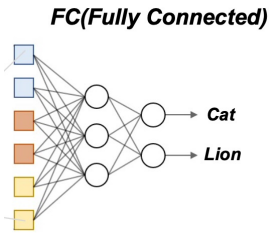






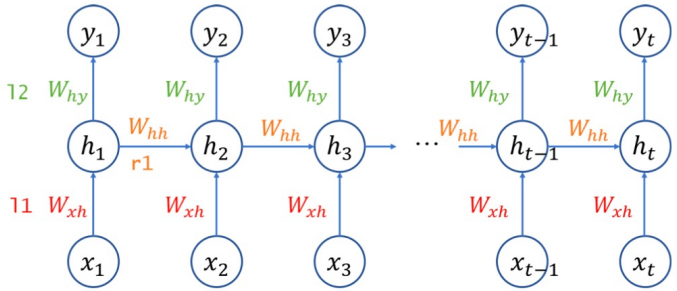
FC (Full Connected) layer

Compute the class scores, each of the 2 categories correspond to a class score, such as among the 2 categories of our dataset. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume



Deep Learning

RNN vs. LSTM

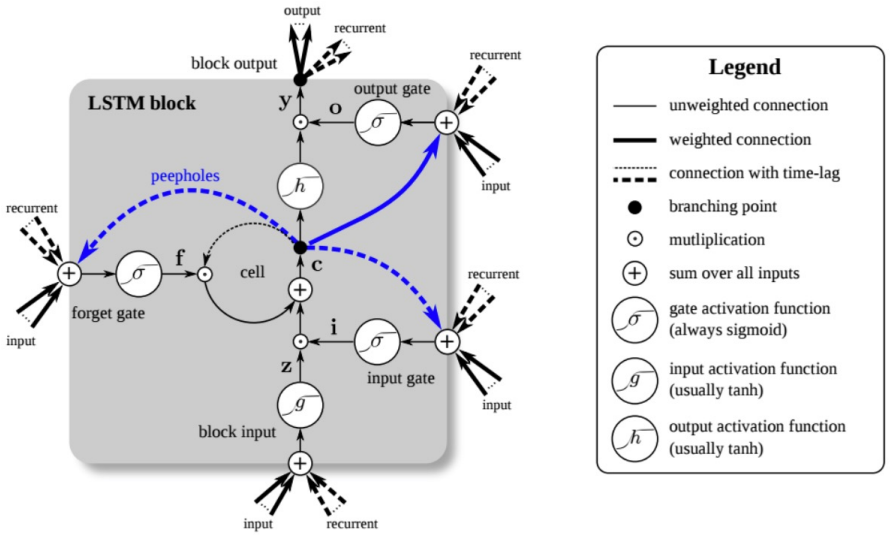


$$h_t = \tanh(\, l1(x_t) + r1(h_{t-1})\,)$$
$$y_t = l2(h_t)$$

The output of hidden layer are stored in memory. Memory can be considered as another input.

So information cycles through a loop.

1. current input
2. what it has leaned from the inputs it received previously.



LSTM is an extension for RNN.

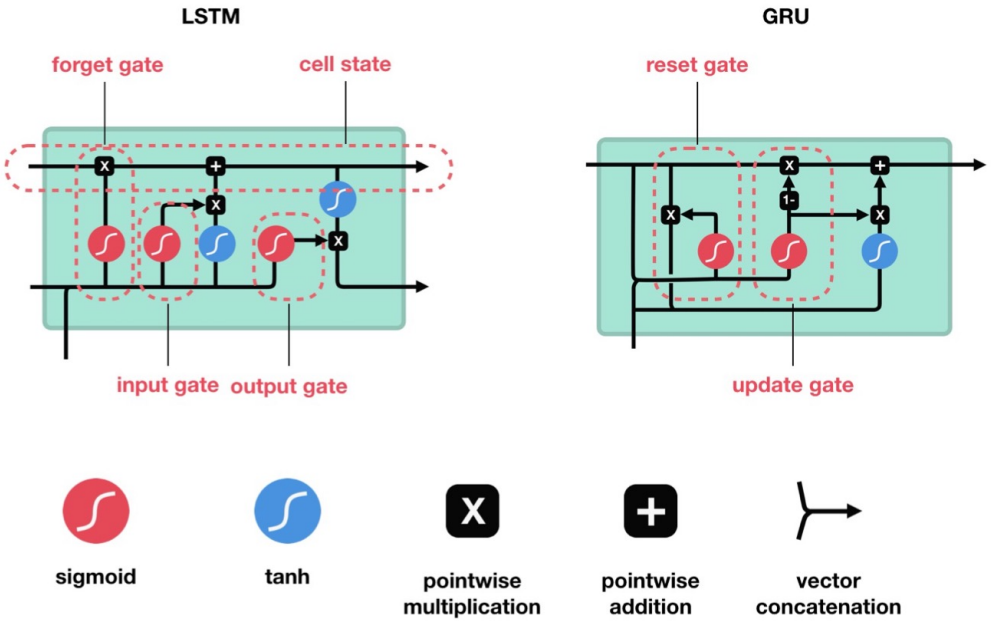
The memory in LSTM can be seen as a gated cell (store or delete based on the importance it assigns to the information).

Three gates:

1. input gate: whether or not let new input in
2. forget gate: delete information
3. output gate: output at the current time step

Deep Learning

RNN vs. LSTM



GRU first computes an **update gate** based on **current input word vector** and **hidden state**

Compute reset gate similarly but with different weights

- If reset gate unit is ~0, then this ignores previous memory and only stores the new word information

Final memory at time step combines current and previous time steps