

Insert Assignment Title Here

02807 Computational Tools for Big Data

Anonymous authors

Insert hand in date here

1 Week 8: Map Reduce

2 Week 9: Apache Spark

3 Week 10: Deep Learning

3.1 Exercise 1

Deep Learning is a kind of Machine Learning, that focuses on using multiple layers of non-linear processing units. Deep learning is special in the way that algorithms designed for Deep Learning is based on distributed representations. This means that we make the assumption, that all observed data have been generated by different factors on the multiple layers of different levels of abstractions. Deep Learning was originally introduced with the object of moving Machine Learning closer to Artificial Intelligence, and has been used in the research community to solve big data problems such as speech recognition, computer vision, and natural language processing.

3.2 Exercise 2

Convolutional neural networks (CNN) is a kind of neural network, that takes advantage of being able to reuse copies of neurons. This way, it is possible to have a very large neural network, while keeping the number of parameters low. This is smart because this means, that the CNN has much fewer neurons, that needs to learn during learning. Copies of the neuron is then used multiple times in the convolution layer. This works much like how a function in programming is used multiple times instead of writing the code of the function multiple times.

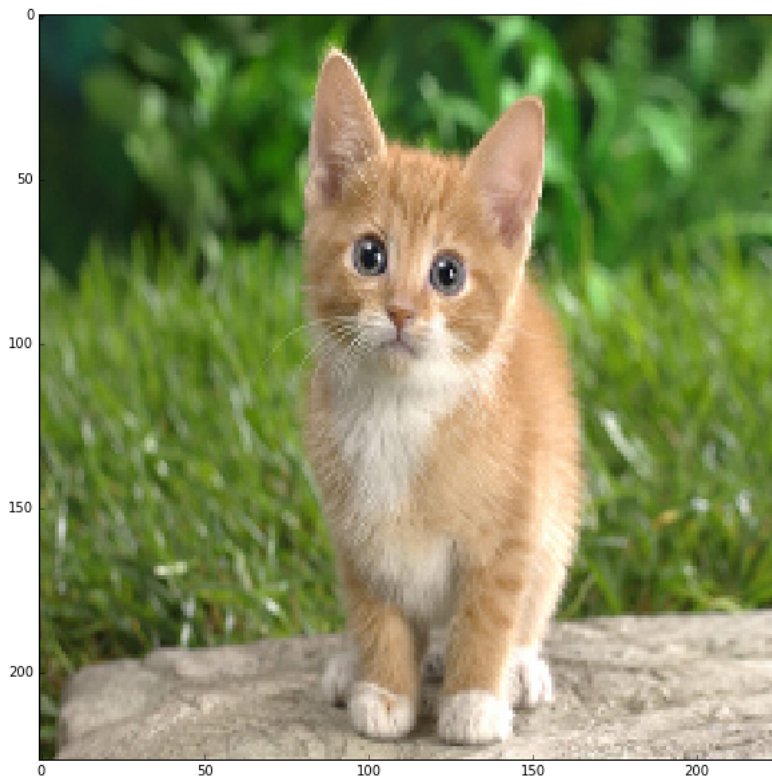
We can have a neuron a , that we use multiple times, so that it collectively covers all input data, where each neuron covers multiple data points. This group of (identical) neurons, we call A , and it makes up the convolution layer. It is possible to have multiple layers, and with each new layer, the network can find more abstract features.

Convolution layers are often interweaved, so that $a_1 \leftarrow \{1, 2\}$ and $a_2 \leftarrow \{2, 3\}$, which makes for higher precision. However, from a high level perspective, it might not be needed to have a high level of precision. For this, max-pooling layers offers better performance (with the loss of precision). The way max-pooling layers works is that the neurons don't interweave, so we have $m_1 \leftarrow \{1, 2\}$ and $m_2 \leftarrow \{3, 4\}$.

3.3 Exercise 3

```
In [8]: plt.imshow(transformer.deprocess('data', net.blobs['data'].data[0]))
```

```
Out[8]: <matplotlib.image.AxesImage at 0x7feacf29fcd0>
```



Adorable, but was our classification correct?

```
In [10]: # load labels
imagenet_labels_filename = caffe_root + 'data/ilsvrc12/synset_words.txt'
try:
    labels = np.loadtxt(imagenet_labels_filename, str, delimiter='\t')
except:
    !../data/ilsvrc12/get_ilsvrc_aux.sh
    labels = np.loadtxt(imagenet_labels_filename, str, delimiter='\t')

# sort top k predictions from softmax output
top_k = net.blobs['prob'].data[0].flatten().argsort()[-1:-6:-1]
print labels[top_k]

['n02123159 tiger cat' 'n02123045 tabby, tabby cat'
'n02124075 Egyptian cat' 'n02119022 red fox, Vulpes vulpes'
'n02127052 lynx, catamount']
```

Indeed! But how long did it take?

In the above classification of the cat from the assignment, you can see, that the classifier correctly guesses it is a cat (of some kind). All five guesses are cats (except for the fox, but I'll let it slide).

```
In [16]: plt.imshow(transformer.deprocess('data', net.blobs['data'].data[0]))
```

```
Out[16]: <matplotlib.image.AxesImage at 0x7fd9b7507790>
```



Adorable, but was our classification correct?

```
In [17]: # load labels
imagenet_labels_filename = caffe_root + 'data/ilsvrc12/synset_words.txt'
try:
    labels = np.loadtxt(imagenet_labels_filename, str, delimiter='\t')
except:
    !./data/ilsvrc12/get_ilsvrc_aux.sh
    labels = np.loadtxt(imagenet_labels_filename, str, delimiter='\t')

# sort top k predictions from softmax output
top_k = net.blobs['prob'].data[0].flatten().argsort()[-1:-6:-1]
print labels[top_k]

['n02129165 lion, king of beasts, Panthera leo'
 'n03325584 feather boa, boa'
 'n02480495 orangutan, orang, orangutang, Pongo pygmaeus'
 'n02123159 tiger cat' 'n02112137 chow, chow chow']
```

Indeed! But how long did it take?

In the above classification, where we uploaded our own image, you can see, that the classifier correctly guesses a Lion. The next 3 guesses are off, and the fifth guess is somewhat okay. There will be a larger error margin, since the uploaded image is highly photoshopped and abstract.

4 Week 11: Feature Hashing and LSH