Wednesday 6th July, 2016
jeh/self/outside activities/ABCDEF/Jiao Tong University/2016 July program at Cornell /SJTU July program 2016

Area of deep learning

Networks

`http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial`

`http://neuralnetworksanddeeplearning.com/chap6.html#convolutional_neural_networks_in_prac`

A reading list `http://deeplearning`net/reading-list/ .

Deep learning is concerned with layered networks of threshold logic units where there are many layers as shown in Figure 1.1. Experimental evidence suggests that the more layers used the better the accuracy. The last level is softmax.

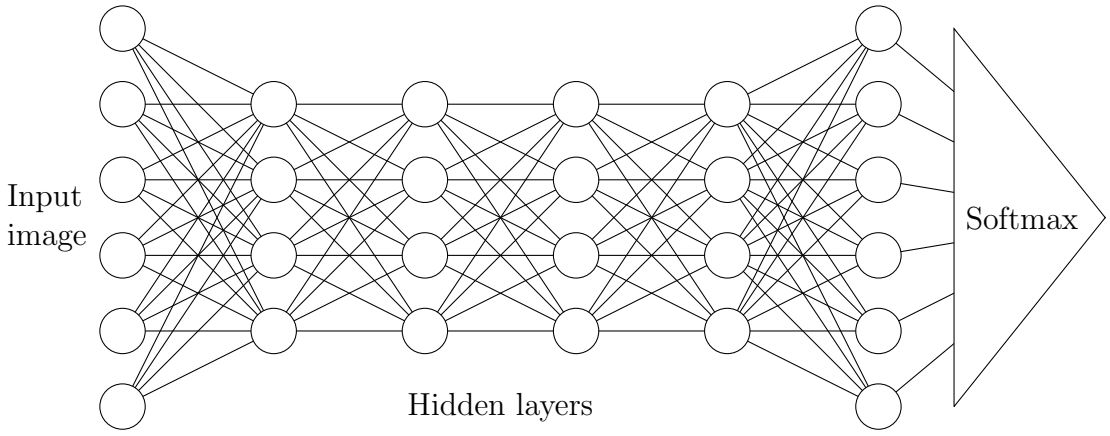


Figure 1.1: A deep learning network

Some researchers use 100 to 150 hidden levels where each level may have a thousand or more gates. In such a network there 10^6 weights per level and with one hundred levels, 10^8 weights. One way to train such a network is to train one level at a time by an auto encoder.

Supervised learning versus unsupervised learning

In supervised learning one labels many images and trains a network to correctly classify the labelled images. The network is then used to classify other images. One would like to be able to train a network without labelling a training set of images. Researchers are now learning how to do this with a technique called auto encoding.

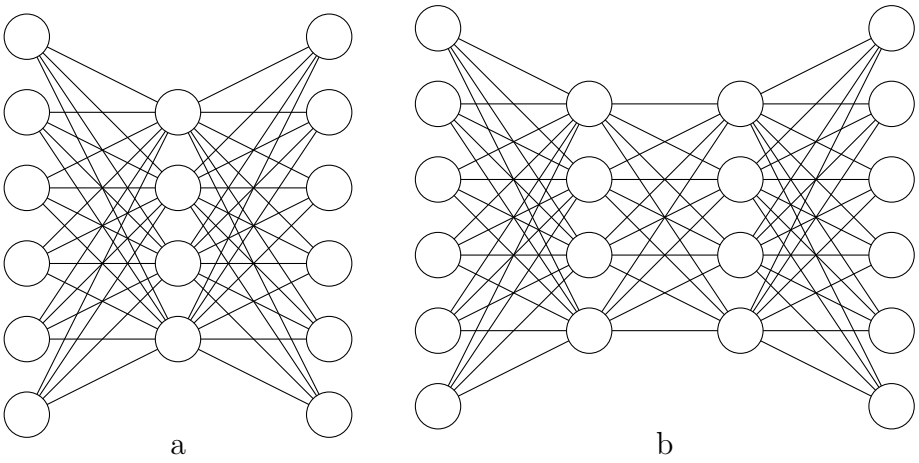


Figure 1.2: A deep learning network

auto encoders

An auto encoder has just one level of hidden gates as shown in Figure 2a. The weights are adjusted so that the output agrees with the input. Once the first level weights are adjusted, they are frozen and a second level of hidden gates is added as shown in Figure 1.2b.

convolution

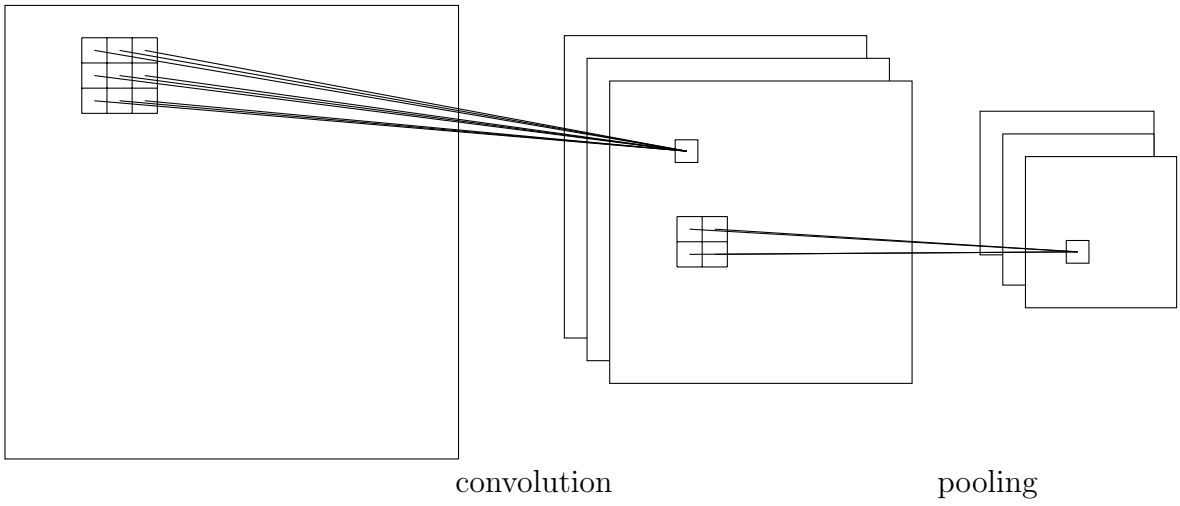
Early work with deep networks frequently was concerned with image recognition. The values of adjacent pixels in images are highly correlated. Thus, researchers use a different form for the first few layers then the complete connectivity as shown in Figure 1.1. In the first layer typically a 5×5 grid was used to scan the image. If the image was 1000×1000, the first layer would consist of 996×996 gates each with 25 inputs.

The weights for each gate are tied together so there are only 25 weights instead of $966 \times 966 \times 25$ weights. The gates all learn the same feature. For this reason maybe 1000 sets of 996×996 gates are used detecting 1000 different features.
`http://cs231n`github.io/convolutional-networks/.

pooling

If a convolution layer detects an edge it is not critical exactly where the edge is so a pooling layer is used to reduce the number of weights. The layer consists of a 2×2 grid that scans the output and averages the four values together. The grid is shifted two unit for each gate so the 2×2 grids do not overlap. This layer is called a pooling layer.

Logistic regression and softmax



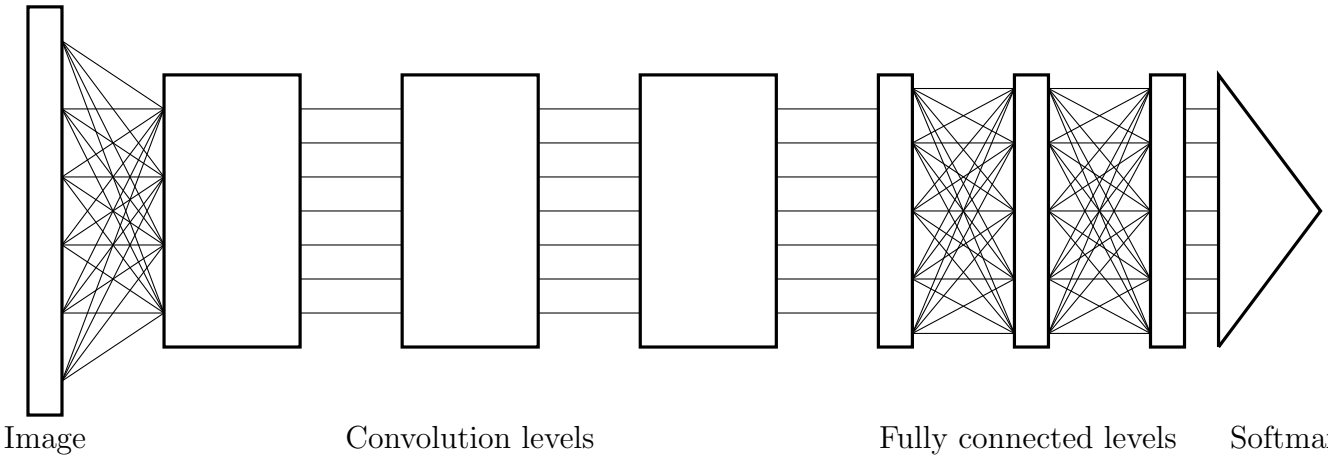


Figure 1.3: Convolution network

In logistic regression, one has a set of vectors $\{x_i | 1 \leq i \leq n\}$ with labels $l_i \in \{0, 1\}$, $1 \leq i \leq n$ and a weight vector w. The probability that a vector x has label l is

$$\text{Prob}(l = 1|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w^T x}}} = \sigma(\mathbf{w^T x})$$

and

$$\text{Prob}(l = 0|x) = 1 - \text{Prob}(l = 1/x)$$

One selects a w that minimizes the cost function

$$J(\mathbf{w}) = \sum_i \Big(l_i \log(\text{Prob}(l = 1|\mathbf{x})) + (1 - l_i) \log(1 - \text{Prob}(l = 1|\mathbf{x})) \Big)$$

Softmax is a generalization of logistic regression to multiple classes. Thus, the labels l_i take on values $\{1, 2, \ldots, k\}$. For an input $\mathbf{x}$ softmax estimates the probability of each label. The hypothesis is of the form

$$h_w(x) = \begin{bmatrix} \text{Prob}(l = 1|\mathbf{x}, \mathbf{w_1}) \\ \text{Prob}(l = 2|\mathbf{x}, \mathbf{w_2}) \\ \vdots \\ \text{Prob}(l = k|\mathbf{x}, \mathbf{w_k}) \end{bmatrix} = \frac{1}{\sum_{i=1}^{k} e^{\mathbf{w_i^T x}}} \begin{bmatrix} e^{\mathbf{w_1^T x}} \\ e^{\mathbf{w_2^T x}} \\ \vdots \\ e^{\mathbf{w_k^T x}} \end{bmatrix}$$

where the matrix formed by the weight vectors is

$$W = \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w_2} \\ \vdots \\ \mathbf{w_k} \end{bmatrix}$$

W is a matrix since for each label l_i, there is a vector $\mathbf{w_i}$ of weights.

sparsity

In a deep network there often are many more weights than input patterns and over fitting would occur. One way to solve this issue is by building into the cost function a term to force many of the gates to be inactive. Let $a_j(x_i)$ be the activation of the j^{th} unit on the i^{th} input. Let ρ_j be the average over all inputs of the activation of the j^{th} gate. $\rho_j = \frac{1}{n} \sum_{i=1}^{n} a_j(x_i)$ Let ρ be a sparsity constant which we specify. To force sparsity add to the cost function the term

$$\sum_j \left(\rho \log \frac{\rho}{\rho_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \rho_j} \right)$$

regularization

Regularization is a method of preventing over fitting by adding a term to the cost function as in sparsity. Another option is to penalize large weights.

codes

To upload some of the codes you may first need Virtual Box unless you are running a Linux operating system. https://www.virtualbox.org/

1. CAFFE http://caffe.berkeleyvision.org
2. Torch http://torch.ch/

   https://github.com/torch/nn/blob/master/doc/simple.md

   https://github.com/torch/nn/tree/master/doc

   A state of the art code https://github.com/szagoruyko/cifar.torch.
3. Tensorflow (Google OpenSource): https://www.tensorflow.org
4. Theano
5. Hinton code for MNIST data

   http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html
6. MICROSOFT https://github.com/Microsoft/CNTK .
7. Matlab MatConvNet:CNNs http://www.vlfeat.org/matconvnet/.

Reference COMPARATIVE STUDY OF CAFFE, NEON, THEANO, AND TORCH FOR DEEP LEARNING http://arxiv.org/pdf/1511.06435v1.pdf
Comparative Study of Deep Learning Software Frameworks http://arxiv.org/pdf/1511.06435v3.pdf
The above two papers are the same papers
DL4J vs. Torch vs. Theano vs. Caffe vs. TensorFlow http://deeplearning4j.org/compare-dl4j-torch7

access to computers

GPU

data sets

1. MNIST http://yann.lecun.com/exdb/mnist/. consists of training set of 60,000 images and 10,000 text images. The images are 28×28.
2. CIFAR http://www.cs.utoronto.ca/ kriz/cifar.html. 60,000 32×32 color images in 10 classes.

3. TIMIT speech requires license and fee.

4. 13,000 face images `http://vis-www`cs.umass.edu/lfw/index.html.

5. Web site of image databases
   `http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm#scene`

6. `image-net http://image-net`org/download-attributes.
   1.5 million imageset dataset ILSVRC 2012

Network structure

1. Alexnet `http://www`cs.toronto.edu/ fritz/absps/imagenet.pdf. good reading
   5 convolution layers followed by 3 fully connected layers and then softmax

2. VGG `http://www`robots.ox.ac.uk/ vgg/practicals/cnn/.
   16 convolution and 5 pooling layers followed by 3 fully connected layers

3. GoogleNet `http://arxiv`org/abs/1409.4842.

Research projects

Exercise 1.1 ***Training deep learning networks***
Explore different methods for training a deep network. How fast are various methods and how good are their accuracies? As additional levels are added to the network does accuracy increase? What happens with overfitting?

Exercise 1.2 ***Size of problem***
Deep learning research at companies uses problem sizes that are beyond that which we can use in university research. However, industrial researchers claim that the size is important to study the results. I believe that one could find university size problems that exhibit the results. Explore the possibility of finding university size problems that exhibit important behavior or explore why the size needs to be so large.

Over fitting

Exercise 1.3 ***Determine good parameters for CIFAR data***
Deep networks have an enormous number of parameters and over fitting is often a problem. Explore methods for determining good parameter values for training networks.

1. *Explore methods for determining good values for parameters of a network.*

2. *Determine parameters of good network configurations for the CIFAR data.*

3. *Determine general principles for determining parameters.*

Exercise 1.4 ***The art of resolving over fitting***
Explore methods to solve over fitting such as dropout and increasing training set by random adjustment to data.

1. *Find a way to train on CIFAR data so over fitting does not occur.*

2. *Reducing over fitting seems to be an art. Develop methodology for determining network methodology and parameters to avoid over fitting.*

3. *Develop understanding of how to deal with over fitting. What techniques work best and in what situations?*

Exercise 1.5 ***augmentation***

1. *Does creating more data by jittering the data help? These are small random translations or rotations.*

2. *Does creating more data by constructing images that map to activation of existing image help?*

3. *Explore other methods of increasing the amount training data.*

Exercise 1.6 ***Explore dropout***

1. *Dropout is a technique for preventing over fitting.*

2. *Develop a good research project involving dropout.*
   `http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf`

3. *Explore methods that reduce training time such as randomly reducing the depth of the network during training.*
   Deep Networks with Stochastic Depth `http://arxiv.org/abs/1603.09382v1`

4. *Is there some simple experiment where one could establish a theory?*

Exercise 1.7 ***Explore regularization and validation***

1. *Regularization is a technique to prevent over fitting.*

2. *Develop a good research project involving regularization.*

Network structure

Exercise 1.8 ***Explore pooling***
The exact location of features is not that important so pooling is used in convolutional networks to reduce the number of gates.

1. *Develop when to use pooling and what size.*

2. *Explore fractional pooling*
   Fractional Max-Pooling, Benjamin Graham, Dept of Statistics, University of Warwick,

Exercise 1.9 ***Structure of networks***
Explore the structure of different networks in the literature and develop general principles for determining the best structure to use in various applications.

Exercise 1.10 ***Convolution structure***
Explore size of window and effect of multiple level of windows.

Exercise 1.11 ***Accuracy*** *In some small networks say one with one convolution level and two fully connected levels one might get 67% accuracy. However if one adds two or three additional fully connected layers sometimes they can only get 60% accuracy. What is going on here?*

Learning

Exercise 1.12 ***What features can convolution levels learn. Levels 1,2,3***

1. *What features do convolution levels learn on CIFAR data?*
2. *How does size of window effect what is learned. For example, is a small window best for edges and a larger window for detecting texture?*
3. *Is there a concept of scale somewhat like frequency? Using different convolution sizes detects different scales?*
4. *Does the features learned on CIFAR data depend on the actual data set or are the features learned by the early levels features of images independent of the actual set of images?*
5. *In convolution networks with 100 channels what features does each channel learn?*

`http://arxiv.org/pdf/1411.1792v1.pdf`

Exercise 1.13 ***What do units learn? – visualization tools***

1. *Survey the literature for visualization tools.*
2. *Develop method for determining what individual or small sets of gates learn.*
3.

`Visualization library  Visualizing and Understanding Convolutional Networks: http://arxiv.org/abs/1311.2901`

`(Deconv Nets)`
`Deep realization tool box  Understanding Neural Networks Through Deep Visualization: http://yosinski.com/deepvis`

Domain-Adversarial Training of Neural Networks: http://arxiv.org/abs/1505.07818

Exercise 1.14 ***Neuron activation: Correlation and mutual information***
Convergent Learning: Do different neural networks learn the same representations?

1. *Train a network several times on the same data with different random starting weights. Can one develop a relationship between the activation vectors in the different runs? Alternative to training is to find a trained network on the web that was trained twice.*

2. *Find the average activation vector for each class of images. Sort the coordinates of the average activation vector for each class according to magnitude. Do this for each network. Use the average activation vector of a category such as cat to match coordinates for each run and see what happens to the other categories.*

3. *For each category of images calculate the variance of each coordinate value of activation vectors. Does high variance mean coordinate does not play a role in that category.*

http://arxiv.org/abs/1511.07543

Exercise 1.15 ***Concepts***

1. *How does one extract a concept such as age?*

2. *How does one define a concept?*

3. *Is there a concept such as natural image?*

Exercise 1.16 ***For CIFAR data does each category need every coordinate of activation vector***
See Figure 1.4

Exercise 1.17 ***Can a deep learning network learn arithmetic***

1. *One might create a number of 1000 dimensional vectors for addition of the format* $1^n01^m0^{1000-n-m}1^{n+m}$ *and see if training on some can learn the remainder. If addition can be learned can multiplication be learned?*

2. *Can a deep learning network learn binary arithmetic? If inputs are a 16 dimensional vector, each input of the form* 1^i0^{16-i} *with one hidden layer autoencoder one might think that four gates are necessary.However, with sigmoid nonlinearity only one gate is needed since the integer input is encoded as a real number. However, if the output was trained to be the binary representation of i how many gates are needed in the hidden layer? What if the inputs were in binary? What if two hidden layers were used with binary inputs?*

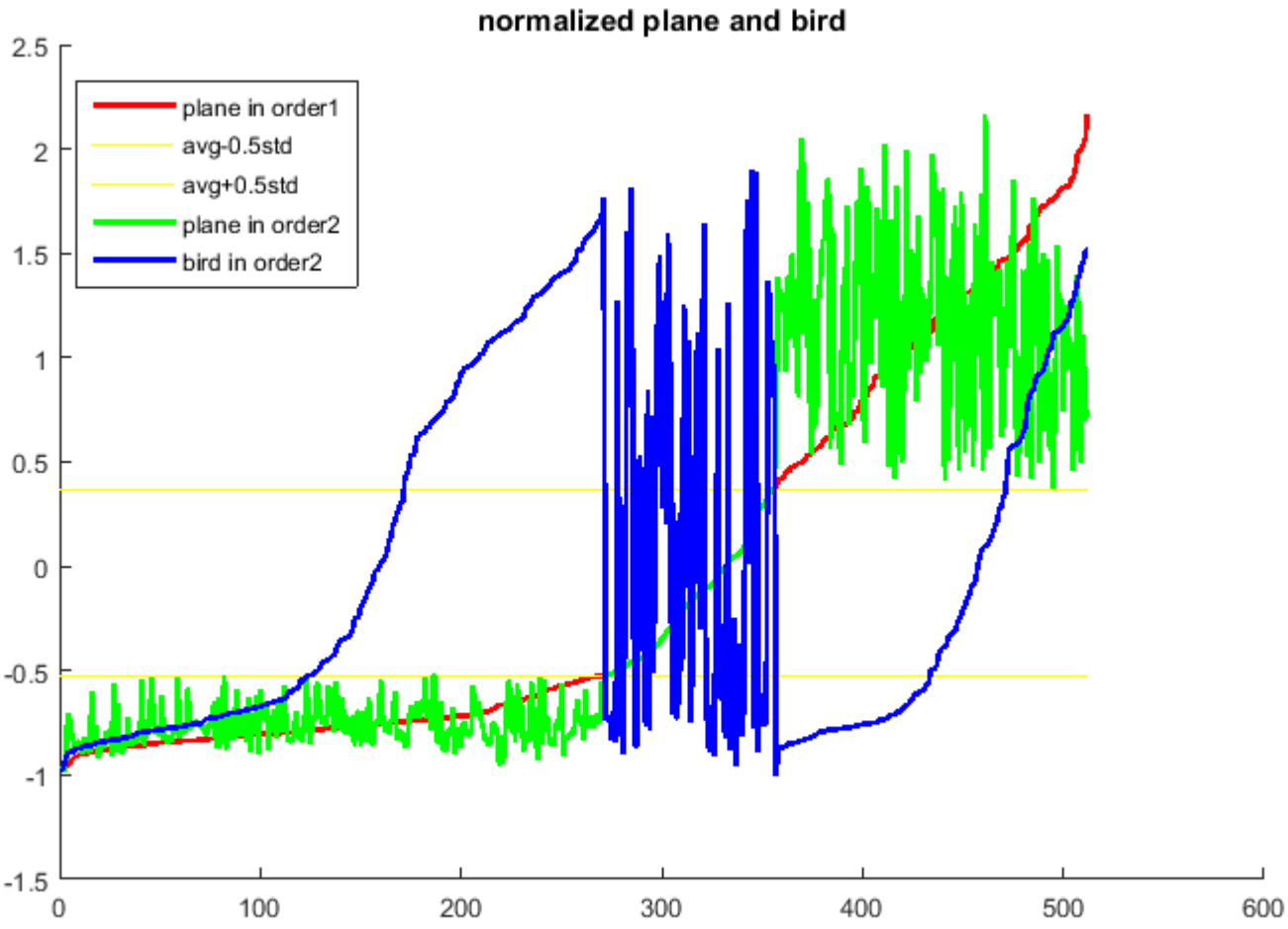


Figure 1.4: The red line is the activation vector for airplane when sorted by magnitude. The green curve is the activation vector for airplane when portions of the curve have been resorted by bird. Bird is in blue.

Exercise 1.18 *How does what is learned change with level?*

1. *How does content change to style*
2. *How does information learned change from common to specific?*

Image from activation vector

Exercise 1.19 ***Create image corresponding to an activation vector***
Understanding Deep Image Representations by Inverting Them `http://arxiv.org/pdf/1412.0035v1.pdf`

1. *One can reconstruct the image i_c that produced an activation vector a_c by applying a random image i_r, finding its activation vector a_r, and using gradient descent to modify i_r so that a_r becomes a_i. If one uses the metric $(a_c - a_r)^2$ and reduces the metric to zero one should get back the unique image i_c. If one reduces some other metric one may not get back the unique image.*
2. *Explore several metrics and see what happens. If one tries only to get the covariance $a_c^T a_c$ to agree with the covariance $a_r^T a_r$ one will get back versions of the image.*

http://arxiv.org/abs/1511.06421

Exercise 1.20 ***Inverse network***
Construct network that learns the inverse mapping. Let x be an image and $\Phi(x)$ its activation vector in a deep network. Train a second network to learn Φ^{-1}. This would allow one to quickly explore activation space.

Exercise 1.21 ***Metrics and levels***

1. *Compare Gram matrix and Euclidean distance for error function in reconstruction of image*
2. *Examine other possible metrics.*
3. *Explore mappings from different levels.*

Exercise 1.22 ***Creating figures from activation vector***
In networks used for image recognition the activation space at each level is reach enough so that every images gets mapped to a unique activation vector. What if we used a simpler network where this was not true.

Consider an auto encoder with one hidden layer with say 500 gates. The input and output layers are pictures with one million pixels. The hidden layer is a more compact representation of the pictures. Say you train the network with a 1,00 pictures of cats. A new cat picture can be generated as follows. Select the hidden layer representation of one of the cat figures. Then starting with a random input pattern where each pixel is a Gaussian, zero mean, unit variance, random variable. Keep the weight values frozen and adjust by gradient descent the input pixels until you get the hidden layer representation of the selected cat picture. The input should be a picture of a cat but not the one you selected since many cat figures should map to the same representation.

1. *Generate several cat pictures from the activation vector of a cat image and assess how will this works.*

2. *The activation vectors for the set of cat pictures should lie on a manifold. What dimension is this manifold?*

3. *Draw a straight line between two activation vectors corresponding to cat pictures. As you move along this line do you stay on the cat manifold?*

4. *The technique of generating data works for music, text, and other domains. Try generating some music or items from some other domain.*

Exercise 1.23 ***Structured activation space***

1. *Since activation space of deep levels are often of lower dimension than the input space it maybe that many possible images are mapped to the same activation vector. However, from some activation vectors one can construct a unique image. Call the set of such vectors structured activation space and the corresponding set of images structured image space. Are there vectors in unstructured activation space from which many images can be reconstructed?*

2. *Is there some way to measure distance between images?*

3. *If there is no unstructured set in activation space, would this be true even if one trained on random vectors in image space?*

Exercise 1.24 ***Changing label of image with minor changes to image***
One can change the label of a picture with minimal change of the picture that is hardly visible to the human eye. This is done as follows. Using a single layer auto correlation network, train the network on a set of labelled images. Take an image x and find the activation $\phi(x)$ and possibly even reduce the dimension further by random projection to a lower dimensional subspace. Then find the minimum r by some metric such that $\phi(x) + r$ has the new desired label. Find an image x_r whose activation vector is $\phi(x) + r$.

1. *Select a category of images from some large image corpus and train an auto encoder, single level network, to recognize the images. Try converting several images with one label to images of another and see what happens.*

2. *Convert one integer in MNIST data set to another with minor changes. Reconstruct the modified image.*

Exercise 1.25 ***Intriguing properties Fooling***

1. *How can you find the minimal change to an image that will change the category of an image?*

2. *Make the minimum change to an image for a cat to change its category to a dog. Can you see any change in the image or is the change too small?*

3. *What are regions of the activation space that have various labels?*

4. *Develop technique to add maximum noise without changing label.*

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images: http://arxiv.org/abs/1412.1897 Intriguing properties of neural networks: http://arxiv.org/abs/1312.6199

Exercise 1.26 ***Convert young person to old***
One can convert a picture of a young person to a picture of an old version of the person. This is done as follows.

1. *Start with an activation vector v of the young person.*

2. *Find activation vectors for a large number of old persons.*

3. *Find the minimum δv such that $v + \delta v$ has an activation vector in the region of the activation vectors of old people.*

4. *Alternatively define the first level activation vector to correspond to the person and latter levels to correspond to age. Find new picture whose first level activation vector corresponds to the person and latter levels to oldness.*

Exercise 1.27 ***artistic style***

1. *Separate content of image into style and content.*

2. *Change style and create new image.*

3. *Explore what other properties of an image can be defined*

`http://arxiv.org/pdf/1508.06576v2.pdf` *A Neural Algorithm of Artistic Style*

Exercise 1.28 ***What properties such as style can one learn?***

Manifolds

Exercise 1.29 ***Structure of activation space***
Apparently a tiny change in the image of a dog which is not visible will change the activation vector to that of a cat. This suggests exploring the structure of activation space.

Exercise 1.30 ***Explore manifolds of objects such as cats.***

1. *Are manifolds such as cats disjoint from dogs?*

2. *What are images between cats and dogs?*

3. *Find a hierarchy of manifolds such as cats, animals, etc.*

4. *Is there a category that has some cats and some dogs but not all? What is the category?*

Exercise 1.31 ***Manifold Traversal:***

1. *Given two activation vectors for cats does the activation vector on the line between the two activation vectors stay in the cat manifold?*
2. *How can you traverse between two activation vectors for cats and stay in the cat manifold?*
3. *What dimension is the space of cat activations.*
4. *Is it connected?*

Deep Manifold Traversal: Changing Labels with Convolutional Features http://arxiv.org/abs/1511.06421

Exercise 1.32 ***Explore clustering in activation space.***

1. *Try clustering in activation space? How does it compare to clustering of the data?*
2. *Is there hidden structure in the activation space?*

Exercise 1.33 ***Automatically create new categories of photos such as standing animals with four legs.***

Exercise 1.34 ***Clustering using k-means vectors in activation space for different values of*** k.

1. *Do the categories of images correspond to clusters?*
2. *Are there other clusters?*
3. *Explore layers of hidden structure.*
4. *Is there another way to create categories?*

Exercise 1.35 ***Related images not used in train - transfer learning***
Suppose you had 100 categories of flowers each category having a specific type of flower and you trained a network to recognize ten of the types of flowers. If you then clustered in activation space would you find all 100 categories?

Exercise 1.36 ***Untrained networks***
An example of an untrained network would be AlexNet with random weights.

1. *Compute the activation vectors for a number of images such as mountains, dogs, cars, etc. Cluster the activation vectors. Do the clusters correspond to category of images?*

2. *Do the activation vectors at different levels correspond to different features such as texture, figure type, etc.?*

Compression

Exercise 1.37 ***Compressing the number of levels in a deep learning network***
Increasing the number of levels in deep learning appears to improve the accuracy. Some researchers use as many as 150 hidden layers. It appears that one can train a network with 150 layers and use the activation of the 150 layer network to train a network with a small number of hidden layers and achieve the same accuracy.

To explore this phenomenon select a data set such as the handwritten digits (MNIST). Train deep layer networks where the hidden layers have say five gates and the number of layers is $1, 2, \ldots, 5$ *and plot the accuracy verses the number of layers. Each network will have a final softmax layer. Next train a network with only one hidden layer with five gates. But instead of using softmax in the training, train the hidden layer using the activation function of the fifth layer in the five hidden layer network trained earlier. Then use softmax to train the output and compare the accuracy to that of the five layer network.*

If we train a deep network with many hidden levels and use the activation vector of the last level to train a network with few levels why do we get a better network than if we trained the network with few levels directly? Is it because the activation vector space has many local minima and the really good local minima have only a very small neighborhood in which if you start you will converge to the really good local minima?

THIS EXPERIMENT DID NOT WORK SINCE WITH MNIST DATA AS WE ADDED LAYERS THE ACCURACY DID NOT IMPROVE. CREATE A VERSION OF THIS EXERCISE THAT WORKS. SEE NEXT EXERCISE.

`http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.pdf`

Exercise 1.38 ***Compression number of levels***
Find a trained convolution network with CIFAR data and consider the first eight levels. Use softmax with first three levels and first eight levels. If eight levels has more accuracy then try compressing levels four to eight with two fully connected levels.

Exercise 1.39 ***Convert convolution neural network to smaller fully connected network***

1. *Train a single level convolution network on CIFAR data. Then try to train a single level fully connected network to learn the activation vectors of the convolution network.*

2. *Train a three level convolution network and then train a single level fully connected network to learn the activation vectors of the the three level convolution network.*

Exercise 1.40 ***Trading breadth for depth***
Hints for thin deep nets `http://arxiv.org/abs/1412.6550` *Train a network of width say 100 and depth 5 and then use the activation vectors of the network to train a network of width only 50 but depth 10.*

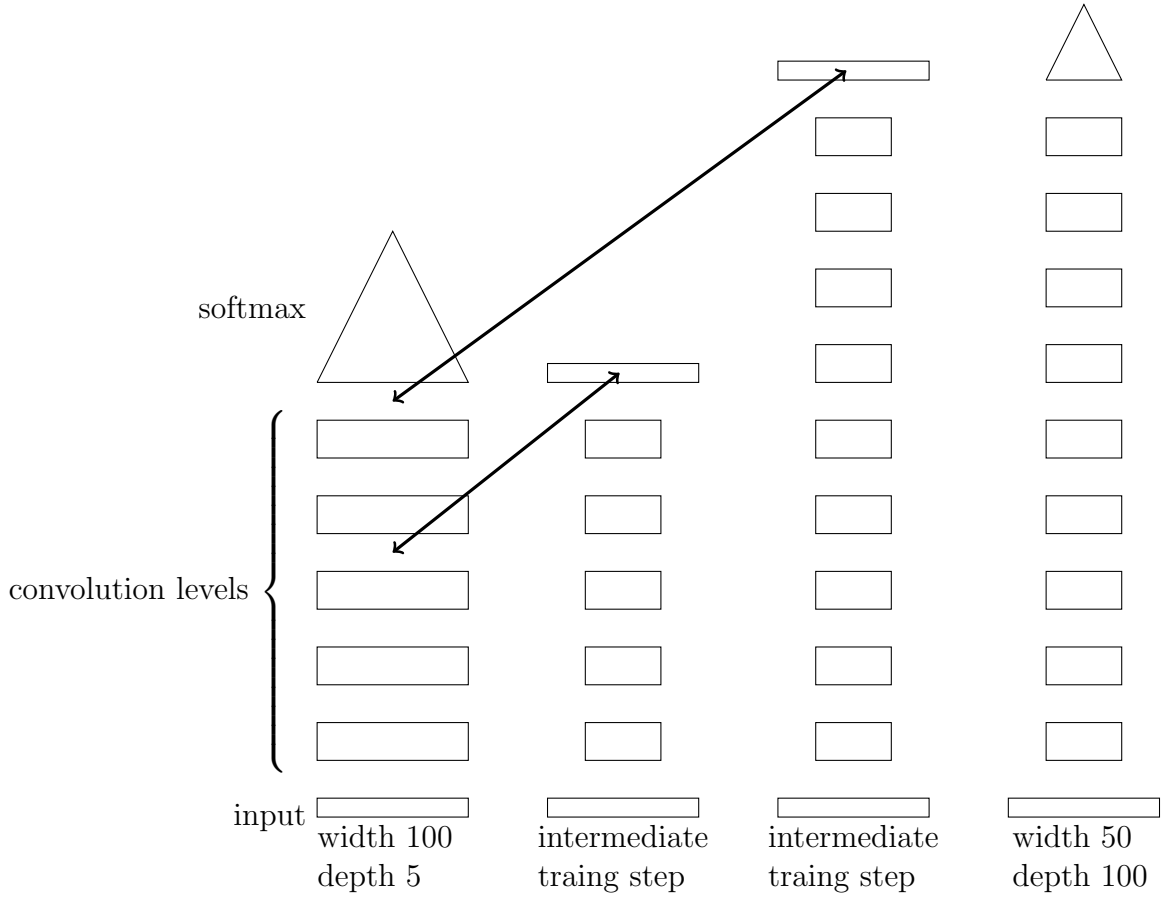


Figure 1.5: In training the network, one trains to get the activation vector of the broad network

Misc

Exercise 1.41 ***Transfer learning***
How transferable are features in deep neural networks?

1. *Train a network with three convolution levels and five auto correlation levels on some photos.*

2. *Freeze the convolution level weights and train on a different set of photos: preferably of a different type. How well does the network learn?*

3. *Finally train a network with all weights free on the second set of data. Can you do better?*

4. *Do early levels transfer if not convolution levels?*

It may be that the convolution levels learn features of photos independent of the actual photos and that is why one can use weights of convolution gates trained on one set of photos for a different set of photos.

`http://arxiv.org/abs/1411.1792`

`ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf`

Exercise 1.42 ***Random weights***
Alexnet has 5 convolutional levels followed by 3 fully connected levels and then softmax. Could one achieve the same result with only the three fully connected layers or maybe four fully connected layers? It does not appear so if we simply train with the images. However, if we use the activation vector of the first fully connected level of Alexnet and trained the first level in the fully connected network to learn the activation we might succeed. To experiment one could use random weights in AlexNet and see if one could with trained weights in the fully connected network achieve the activation vector of AlexNet with random weights.

1. *With a one level fully connected network can one learn the activation of the convolution levels of Alexnet.*

2. *With a two level fully connected network can one learn the activation vector. here the first level of gates could be much wider than the output level.*

Exercise 1.43 ***Evaluating network architecture without training***

1. *Train several simple networks each with a final softmax level and compare how good they are.*

2. *Compare the networks with random weights training only the last softmax level.*

Exercise 1.44 ***Teacher student***
FitNets: Hints for Thin Deep Nets `http://arxiv.org/abs/1412.6550`

Exercise 1.45 ***multi task learning***

1. *It may be faster to simultaneously learn a related task then a single task. Explore this idea and see if you can come up with a convincing example.*

2. *See if you can explain what is happening.*

http://www.cs.cornell.edu/~caruana/mlj97.pdf

ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf

Exercise 1.46 ***Cropping***
Explore cropping of an image to extract a sub object.

Exercise 1.47 ***Learning from a single picture*** *A child learns from a single image, not thousands of images.*

1. *How well does a deep network do if trained on one image from each category when tested on all images?*
2. *Try training nine categories on all images in the categories and then train for the tenth category with just one image.*
3. *Explore ideas that would allow learning with one image per category.*

Exercise 1.48 ***Tree of life***
Can one construct the tree of life from activation vectors of animals learned in a deep network?

Exercise 1.49 ***Movie***
Consider a short movie of a bird in flight. Can one substitute the image of a different bird and get a realistic movie? Think of other possibilities like this.

Exercise 1.50 ***MMD Maximum mean discrepancy***

1. *Consider the set of points (-1,2),(-1,1),(-1,0),(0,0),(1,0),(1,1),(1,2) and create contour of equal MMD.*
2. *Move point (0,4) to the above region by the shortest path. Plot the path.*

Exercise 1.51 ***Recurrent networks***

1. *Explore the research on networks with memory.*
2. *Develop an interesting problem to explore.*

http://spectrum.ieee.org/computing/software/the-neural-network-that-remembers

Exercise 1.52 ***Create a good exposition of deep learning***

1. *Provide a good survey of deep learning that would bring new researcher up to speed quickly*

2. Produce a good set of references

Exercise 1.53 ***Create your own exercise***

References

1. Deep Learning Tutorials http://deeplearning.net/tutorial/
2. Book http://neuralnetworksanddeeplearning.com/

   Chapter 6 http://neuralnetworksanddeeplearning.com/chap6.html