“Introduction to Deep Learning” Q & A

Q: With deep learning, are we skipping feature engineering? No need for input from domain experts?
A: Yes, the whole advantage of deep neural networks is that they automatically learn the features relevant to the training data, and in most cases drastically outperform hand-engineered features in accuracy. This does not mean that DNNs are going to replace domain experts - domain expertise is still critical in making decisions such as choosing the right training objective for the DNN and coming up with physically plausible augmentations that can be made to the input data to control overfitting and improve network generalization.

Q: How do we choose the number of layers in the network? Are more layers better?
A: In general, more neurons in a network gives a more complex model, and with the same total number of neurons, more layers can model even more complex mappings. There are different types of layers which can affect the type of features that are learned. However, you have to make sure that you have sufficient data for your network size to avoid over-fitting - although, as Geoffrey Hinton says, if you are not overfitting your network isn't large enough - so it's better to have a big network and then control overfitting using regularization methods. It makes sense to start with a network that has performed well on a similar dataset and problem to yours and refine from there - either using domain knowledge or using a hyperparameter search (the best methods for this are a research topic).

Q: Is there a way to map back the results to original input to identify what feature in the original input is important for classification?
A: Yes, the neural network can be run in reverse, back-projecting the features that were salient to the classification that was made. Look up deconvolution networks for some interesting results that generate the input images which would maximally activate higher level neurons in the network. These techniques are a powerful answer to criticism that DNNs are “black boxes”.

Q: What pre-requisites are there for the remainder of this course? What math background is useful for learning DL?
A: No pre-requisites for the course - we will be taking a practical approach to using the available DL tools in this course. More generally, the mathematics of DL is not overly complicated - basic linear algebra and matrix algebra is sufficient for applying DL to well understood problems such as object recognition in imagery. Basic calculus is necessary if you want to understand how the training process works under the hood - but this is rarely necessary in applications. Understanding optimization techniques is also helpful, but not critical.

Q: What resources do you recommend for learning about the mathematical details of Deep Learning?
A: http://cs231n.stanford.edu/ - Stanford CNN course, Andrew Ng's Coursera course on machine learning, and the Theano deep learning tutorials. These are complementary to the

material you will be receiving in this course - we are helping a DL practitioner get started using the available software frameworks. These other courses provide the mathematical underpinnings which are necessary for DL research or advanced applications.

Q: Does DL make other ML approaches obsolete? If not, what problems is DL suitable for and not suited for?
A: No - DL appears to work best when you use raw data as input and you attempt to learn a complex function mapping inputs to outputs. In cases where data is necessarily very structured and abstracted from raw sensory input, DL has yet to prove an advantage over traditional methods such as ensembles of linear classifiers or decision trees. That is not to say that DL can't be applied, but these are really not the problems where DL shines - it is in the space of problems that traditional methods fail, raw sensory inputs and complex functions.

Q: Can a normal desktop with a video card train a reasonable deep neural net? Do we need Google-size resources to leverage this technique? How is the GTX 980 for CNN? Does one really need a Tesla GPU?
A: Yes, many deep neural nets will fit on a single GPU and can be trained in a reasonable time on one. Our best consumer card for this is the TitanX, with 12 GB of RAM. In early development, DNNs can be trained on GeForce cards like the GTX 980 and Titan X, and in fact we have the Digits Devbox with has 4x TitanX GPUs for that purpose. Once you move to a more serious deployment where you are using servers, Tesla is the correct card to use.

Q: Can deep neural networks be trained in an unsupervised fashion, i.e. when there are no labels for the data?
A: Yes - instead of setting the objective for the training to be minimizing the difference between a predicted label and the ground truth training label, you can instead formulate objective functions which attempt to find qualitatively "good" choices of network parameters, e.g. sparse representations of the input data which enable effective input data clustering. Alternatively, you can formulate what is called an auto-encoder where the network attempts to reduce the dimensionality of the input data, i.e. compress it, and then reconstruct it - you optimize the parameters to minimize the difference between the reconstructions and the original input.

Q: Once a network has been trained, how well does it adapt to a new set of data that is added with a new classification category?
A: This actually works quite well and is called fine-tuning. The last layers in the neural net can be replaced with the new categories, then the neural net is trained on a new dataset. This works because many of the low-level features in the first layers are common to different datasets. In addition, the same fine-tuning process can be used to incorporate feedback from a deployed inference system to update the model.

Q: Can DL be applied to non-classification problems?

A: Yes - there are many objective functions that can be formulated on top of a DNN. In addition to classification, we can have regression functions which attempt to estimate some continuous variable or unsupervised objective functions.

Q: How many epochs do you typically train for? When do you stop?
A: This is problem dependent - number of data samples, batch size, objective. You stop when your validation error is no longer decreasing.

Q: What pre-processing is usually applied to data before being fed into a DNN?
A: Subtract mean, standardize values esp. for images, PCA/ZCA/whitening. Randomize data within batches to ensure class stratification. Data augmentations.

Q: How can you leverage multiple GPUs to train a single DNN?
A: Data parallelism, model parallelism

Q: What examples are there of DL being applied outside of vision, speech and natural language understanding?
A: Some work in drug discovery and interactions

Q: Does DIGITS include debugging tools? for example, check my gradients and activations along many iterations
A: Not currently, but this is an excellent suggestion for a future feature

Q: Does cuDNN essentially exploit the GPUs to get convolution faster?
A: Yes, cuDNN accelerates both the forward and backward convolution in CNNs to drastically speed up training and inference.

Q: How are the GPU cores handled in deep learning? Does it use the same concepts of threads, blocks and grid like in CUDA?
A: Under the hood, any deep learning application or toolkit (Caffe/Torch/Theano) that leverages cuDNN is relying on CUDA and its programming model, and this is precisely how they get the best training performance. So the short answer is “yes”. By using one of the high level toolkits you can focus on neural network architecture and performance, and you don’t need to be a CUDA programmer at all -- but you still get the performance benefit. You won’t have to think about blocks and threads when you develop deep neural networks with Digits/Caffe/Torch etc.

Q: Can deep learning techniques / algorithms be deployed on FPGAs instead of conventional CPUs and GPUs?
A: Yes they can, but the flexibility and ease of GPU programming allows one to more quickly adapt learning algorithms as better techniques are discovered.

Q: Are any of the recent algorithm advances already "baked" into the DL Frameworks, or is it up to the user to choose the correct preprocessing methods and implement them outside the libraries?
A: Depends what you mean by “algorithm advances”. In general, the DL frameworks make every attempt to keep up with the current state of the art in deep learning algorithms and so they often implement these directly in the frameworks.

Q: Could you compare computation complexity between training and deploying DNN?
A: Answered in the audio recording, although I think questioner is asking about Big O notation, i.e., is the training N^3 or other.

Q: Hi I have a question. Does the size of the input data (number of pixles for example) need to be the same for all training data? If not, how does the size of the input layer correlate with the size of your training data?
A: Answered in the audio recording. Typically they are the same size as input into the DL training network. If they aren’t the same size you can pre-process images such as crop, stretch, etc.

Q: On slide 7 in the course #1 lecture, it shows the process of Training with errors identifications. How we find these errors, based on any model or user input?
A: The errors are calculated based on the ground truth, i.e., the images are all labeled and the training error is calculated based on how accurate the neural network predictions are compared with the ground truth of the images.

Q: Is there a book of deep learning with exercises???
A: There are likely many books, but if you want exercises I recommend trying Andrew Ng’s course on machine learning found on Coursera. The deep learning frameworks like Caffe, Torch or Theano also have lots of documentation including tutorials and exercises.

Q: Can you please talk about the advantage by using batch processing? I am using Caffe, and it seems the batch processing has no benefits with CPUs. Any insights from DIGITS about batch processing?
A: Batch processing is very popular on GPUs because you can tune the batch size to correspond with the amount of memory (RAM) on the GPU. When the batch size fits into GPU memory the training computations are quite efficient on GPUs.

Q: How stable are the weights of a training network? If you continued to add more and more training data, would the weights constantly fluctuate or are weights guaranteed to converge (perhaps within some small bound) to a stable value?
A: When training you are trying to find the minimum of the objective function. If you continue to add more training data then you will potentially change where the minimum of the objective function is so certainly the weights will continue to change slightly. Provided your network is appropriate for the data you’re using.

Q: How suitable is OpenCV for Deep learning?
A: OpenCV is used very often when doing computer vision tasks. It is quite popular.

Q: Is there a benchmark for commodity GPUs for say a common dataset, for training and testing?
A: There are published results for some of the public deep learning frameworks. There is also the ImageNet competition which is open to the public. I'm not aware of a pre-canned benchmark that is ubiquitous as e.g., High Performance Linpack (HPL) is for HPC systems.

Q: Can the models be deployed with a reasonable run time on a machine that does not have a GPU; that is, CPU(s) only?
A: One can always do both training and classification both with and without GPUs. For training especially GPUs are becoming the defacto standard in many cases. For classification people do use GPUs and they also use CPU-only systems. Classification is much less computationally-intensive so CPU-only systems there seem to be quite popular.

Q: For a time series classification, should I use CNN or Recurrent NN?
A: Answered in the audio recording.

Q: Can a typical deep learning application be run on a modern PC? what computing resources are typically needed?
A: This really depends what you mean by "typical". It is certainly the case that a modern PC with GPU has the capability to train quite large networks. For example, while not a typical PC, the DIGITS devbox (https://developer.nvidia.com/devbox) has the ability to handle a great deal of popular deep learning training tasks.

Q: Given a GPU with some fixed RAM which we know how is the correct way to choose the batchsize, e.g. is there any mathematical formula by which to know what is the Memory usage of the backprop and thus to select the optimal size batch size.
A: Answered in the audio recording. Not strictly no. There are some typical ranges of batchsizes used but no hard and fast analytic formula.

Q: Is CUDA based on OpenCV?
A: No, CUDA is NOT based on OpenCV. They are separate but there is a portion of OpenCV that is written using CUDA to provide for efficient performance on NVIDIA GPUs.

Q: I'm a CUDA programmer. Can I build a DNN through CUDA for a custom task, that can potentially be faster than the general frameworks such as Caffe?
A: Potentially you could. If you can exploit some specific properties of your DNN and input data that isn't captured specifically by Caffe you could possibly write it faster. It will be lots of work but potentially possible.

Q: Will any DL framework work on "mobile CUDA" (like Jetson TK1)?
A: Depends if the framework developers have built/tested using ARM CPU. If so, then it will work on Jetson TK1.

Q: Is there a particular network architecture suitable for NLP, unsupervised clustering of text data?
A: For NLP, people generally use a class of deep neural network called Recurrent Neural Networks (RNN). A great intro tutorial on RNN for NLP can be found here:
http://nlp.stanford.edu/courses/NAACL2013/

Q: Is a big DLNN always better performing than a smaller one?
A: That depends, actually. Remember that for DNNs you need a lot of data to avoid overfitting. If you are not able to provide more data to your larger DNN model, then it may overfit (ie. simply "memorize" all the examples)...and thus perform poorly on new real-world examples. That said, if you have lots of data, then yes, larger models generally do better.

Q: Are DNNs suitable for sequences such as language?
A: Yes, absolutely. DNNs are having a lot of success in speech and text processing. We use a particular type of DNN called an RNN - Recurrent Neural Networks (RNN). A great intro tutorial on RNN for NLP can be found here: http://nlp.stanford.edu/courses/NAACL2013/

Q: Could you please leave some comments about how deep learning relates to SVM?
A: SVM = Support Vector Machine The SVM is not a neural network technique, it is a "traditional" statistical machine learning technique. It remains one of the most popular machine techniques today. Some people would say that it is more difficult to scale an SVM classifier to a really large dataset -- that DNNs scale better.

Q: Which DL framework and architecture is the most suitable for working with speech signals?
A: The most popular toolkit for speech and text processing is probably Kaldi:
http://kaldi.sourceforge.net/about.html
However, certainly Torch and Theano are popular for speech as well. With regards to architecture, the RNN (Recurrent Neural Network) is the main one.

Q: How do you come up with a configuration of neural network for a particular task? In other words, why do you set up a convolution layer followed by a pooling layer following by a Local Response Normalization layer etc for image recognition? Are there any principles in selecting the order of layers, or this is a black magick process?
A: Great question. We will speak to this later in the course. At a high level, building a neural network architecture requires a solid understanding of machine learning techniques. The details of the core network architecture are driven by a combination of mathematics, the experience of ML researchers (trial and error) and ideas taken from related areas (such as neurophysiology - how does a "real" neural network do it). If you are new to DNNs, then you definitely want to

start with one of the well known existing architectures, and then make tweaks from that starting point as you get experience.

Q: Would it be better to use a Titan X instead of a gtx 980 TI?
A: A Titan X GPU is indeed a more powerful GPU than a GTX 980 TI. Both are excellent choices for starting out with deep neural networks. The Titan X has a bit more compute capability, because it has a few more SMs (streaming multiprocessors) than a 980. The most important difference between the two GPUs is that the Titan X has twice the memory, 12GB versus 6 GB. You can fit larger models on the TItan X. Both are excellent.

Q: Can I use CNNs for Image registration problem which uses two images from two different sources and produce a combined image?
A: Certainly the answer is "yes". I don't have a reference handy to provide you. This is a more advanced topic so I will note it and hopefully we can address this in a later session.

Q: I have heard that sometimes the output of a CNN is fed into a SVM. Why is this necessary? Isn't CNN adequate for classification purposes? Is CNN used only for feature extraction?
A: In general people use DNN/CNN for the entire pipeline, not just feature extraction. However, SVM is a very powerful technique. By using DNN lower layers you get to harness the ability of the DNN to scale to large inputs and datasets. Putting an SVM on top then gives you a strong statistical technique to feed those features into. SVMs are probably more well understood, by more people, from a mathematical perspective than DNNs. I'll note this question as "advanced" and we get a further answer for you in a later session.

Q: I'm planning to analyze in real time the directions the consumers eyes are watching. Are the size of pupils enough large to be intercepted in images visualizing half to full body? What it the lower size limit a portion of images has to be, to be meaningful?
A: This sounds like an empirical question. For very accurate eye-tracking people usually use cameras that are mounted right in front of the face. I'm sure it will come down to the resolution of your images, and what level of direction accuracy you need. Sounds like a great project though -- good luck!

Q: Can DIGITS handle arbitrary data types without a lot of programming, or is it mainly designed for pictures?
A: Right now you can use square or rectangular images with DIGITS. They can be either color or grayscale. We can also handle different image formats. We plan to expand this in the future.

Q: I am planning to get an image recognition system on an embedded system, where I want a real time object recognition. Do you think CNN's suit for realtime on embedded processors, say Tegra K1?
A: I have been able to classify images with trained networks by Caffe on the Jetson. Classification times can vary depending on how you are classifying your images. We have seen up more than 30 images per second.

Q: If I have built a model with DIGITS, how would I get a list of classifications on a large >50k new set of images?
A: You could use the Classify Many feature to show you at a glance how well your network is able to classify the new/unseen images. Personally, I would download my trained network files and then use a python script to classify and log the classification results.

Q: There seem to be very strict limits on the size of the images (never seem more than 256x256). Do all the images have to be the same size and what id the big "O" of the image size
A: Yes, the images need to be the same size before being ingested by Caffe. However you can change your dimensions to a larger size like 512x512 or even rectangular such as 200x400.

Q: how can one decide which DL framework to use? (Caffee, Torch, Theano, etc.)
A: Great question. I personally think all of the frameworks are good. They all have great documentation and examples to help one get started.

Q: I have downloaded DIGITS, but since it depends on CUDA, I can not use it. I couldn't install CUDA because I don't have a graphic card that supports CUDA. Is there any way to use DIGITS without a GPU, only using CPU?
A: You can. DIGITS works with Caffe on the backend. So you need to rebuild Caffe without CUDA. Go into your MakeFile.config file and uncomment CPU_ONLY:=1.

Q: In a multi GPU configuration, does DIGITS need the GPUs to be of an exact same model or can they be different, eg. a Titan X and a GTX 780?
A: You can have different GPUs. But you may have to reduce your batch size to account for memory on the smaller GPU. If you have 3 or 4 GPUs, say 2 Titan X and 2 GTX 780, you can select which GPUs you would like to use for each training.

Q: Does the ordering of the training data matter?
A: With some of my tests, shuffling helped me train more effectively. Without shuffling for my small datasets I found an oscillation in my training results.

Q: How do you come up with a configuration of neural network for a particular task? In other words, why do you set up a convolution layer followed by a pooling layer following by a Local Response Normalization layer etc for image recognition? Are there any principles in selecting the order of layers, or there is no such methodology?
A: Interesting question. There is a lot of activity around developing the right DNN for your data. AlexNet the winner of the ImageNet 2012 Competition has 5 convolutional layers and two fully connected one. Then the GoogleNet network which won in 2014 is have even more layers and proved to be more accurate.
One reason to pool the data is to reduce dimensionality while still showing filters response with respect to that portion of the input image.

Couple things to remember about convolutional layers, if you don't zero pad your input image is reduced by the radius of the input. This combined with pooling reduces the input data.

Q: Are there any easy-to-use frameworks that would be suitable for audio or arbitrary data types?
A: You may want to look at Kaldi - http://kaldi.sourceforge.net/index.html for audio. Torch also has a audio module that you might want to look at https://github.com/torch/torch7/wiki/Cheatsheet#audio

Q: What technique do you recommend to counter overfitting and when ?
A: Monitor your accuracy with the test data and loss. I have found that over time, my accuracy with the training data will continue to improve and at some point over fitting. The accuracy with the test data reaches a minimum and this is the point where my trained NN is most effective, WRT my test data (data it has not seen and is not used for training).

Q: Is the AWS cloud a suitable platform for training DLNNs? Or is it better to source my own GPU?
A: I use AWS for a lot of my DL tinkering. It gets the job done. However, my training times are greatly reduced when I use other GPUs, such as K40 and Titan X.

Q: What are the common reasons behind a network not training properly (for example the loss staying constant and the accuracy not improving)?
A: I have found that when my network is not training, I often need to reduce my learning rate. For Caffe, I look at my loss value and check to see if it is increasing. If so, I first try to reduce my learning rate.

Q: Can digits accept CAFFE any .prototxt file user defined? Like fully connected layer? Currently I only see CNN based model like Lenet, Alexnet…
A: If it works with Caffe it should work with DIGITS. Although I have to admit I have not tried this. My fully connected network testing has been done in Python not with DIGITS + Caffe

Q: Is it possible to image recognition real time in a stream (video)?
A: Yes, image recognition in a video can be done by applying a convolutional neural network to the decoded frames. If additional information about what is happening in the video is desired, a recurrent neural network can be added below the CNN to capture temporal features in the video

Q: Once a network has been trained, how well does it adapt to a new set of data that is added with a new classification category?
A: Yes, this is referred to as fine tuning, and it works because many of the lower-level features are common between datasets. Only the weights for the fully-connected layers need to be adjusted.

Q: Is DNN same as recurrent neural learning?

A: Recurrent neural networks are a type of deep neural network where connections can feed back into the network, allowing cyclical loops in the network. These are good for learning temporal features

Q: Would a deep learning network be able to pick up patterns over different time periods? Would a variable from the past be able to influence the present outcome?
A: This is a good use case for a recurrent neural network, which can discern temporal and contextual information from the data.

Q: Actually i wanted natural language processing in images, such as identifying names, organization, address so what is the most useful architecture i could use,simple neural nets, RNN or convolutional NN?
A: I would use an LSTM RNN with a CNN on top. The CNN processes the image to pick up the salient features, and the RNN can decode the sequences of letters/numbers

Q: Do we resize the images before presenting it to the input layer of the DNN (Construct a image pyramid) as in techniques like DPM ?
A: Yes, usually images are resized to a common resolution before being input to a DNN. They can also be normalized.

Q: What if i have limited data and i want to avoid overfitting with DNN. How to fine tune DNN in that case.
A: You can regularize the weights by penalizing large values in the cost function

Q: Is it possible to use deep learning to recognize a particular action within video, like waving a hand?
A: Yes, this is commonly done with a combination of a 3D CNN to extract spatio-temporal features and an RNN to recognize sequences of those features.An example:
http://liris.cnrs.fr/Documents/Liris-5228.pdf

Q: Can deep learning be used for online training, using very few images e.g. from a mobile phone?
A: Commonly you need hundreds of thousands, or even millions of images to train a convolutional neural network properly. With only a few images you run the risk of overfitting

Q: Are DNNs fast enough in classification tasks for commercial applications (Siri/ Google Voice Search), or are they used in conjunction with some classical ML techniques?
A: While we can't divulge specifics of what companies are using, I think it is safe to assume that DNNs can be used in real-time speech recognition and natural language processing

Q: Could you please introduce an easy-to-use CNN for image/video classification?
A: NVIDIA Digits on top of Caffe is a great place to start for image classification, and Caffe with 3D convolutions would work for video.

Q: The image recognition application that we used last week in the lesson seemed rather fast but failed to recognize unusual objects that I tried. Would a better trained network be much slower during the usage?
A: The speed of classification should be independent of the training time. The network need only be evaluated in a single forward pass