INFO 7390
Advances in Data Sciences and Architecture
Exam 2

Student Name: __Zihan Zhang_____
Professor: Nik Bear Brown

Due: August 16 Thursday, 2018

All the questions REQUIRE an explanation of the question. Yes or no responses get no credit. Any text or answers from the Internet MUST be cited. You MUST work alone on this; answers too similar to other students' answers will receive no credit.

Q1 (5 Points) What is "Deep Learning?" How does it different from traditional machine learning techniques such as SVMs, Naive Bayes, Decision Trees, etc? Why has it become so popular in the last decade?

What is "Deep Learning?"
Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.

How does it different from traditional machine learning techniques such as SVMs, Naive Bayes, Decision Trees, etc?
The most important difference between deep learning and traditional machine learning is its performance as the scale of data increases. When the data is small, deep learning algorithms don't perform that well. This is because deep learning algorithms need a large amount of data to understand it perfectly. On the other hand, traditional machine learning algorithms with their handcrafted rules prevail in this scenario. Below image summarizes this fact.

Deep learning algorithms heavily depend on high-end machines, contrary to traditional machine learning algorithms, which can work on low-end machines.

Deep learning algorithms try to learn high-level features from data. This is a very distinctive part of Deep Learning and a major step ahead of traditional Machine Learning.

In a typical machine learning approach, you would divide the problem into two steps, object detection and object recognition. First, you would use a bounding box detection algorithm like grabcut, to skim through the image and find all the possible objects. Then of all the recognized objects, you would then use object recognition algorithm like SVM with HOG to recognize relevant objects.

On the contrary, in deep learning approach, you would do the process end-to-end. For example, in a YOLO net (which is a type of deep learning algorithm), you would pass in an image, and it would give out the location along with the name of object.

Why has it become so popular in the last decade?
Now with deep learning and GPUs we can achieve higher accuracy at a practical speed! Deep learning is also much more accessible in terms of the learning curve.

Q2 (5 Points) What are activation functions? What is their purpose? Must they be non-linear? Must they be continuously differentiable? Must they be monotonic? How does the derivative of the activation function effect the gradient?

What are activation functions?
It's just a thing (node) that you add to the output end of any neural network. It is also known as Transfer Function. It can also be attached in between two Neural Networks.

What is their purpose?
It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function).

Must they be non-linear?
They don't have to be non-linear, because the Activation Functions can be basically divided into 2 types:
1. Linear Activation Function
2. Non-linear Activation Functions

Must they be continuously differentiable?
No, it is not necessary that an activation function is differentiable. In fact, one of the most popular activation functions, the rectifier, is non-differentiable at zero!

Must they be monotonic?
the monotonicity criterion is not mandatory for an activation function - It is also possible to train neural nets with non-monotonic activation functions. It just gets harder to optimize the neural network.

How does the derivative of the activation function effect the gradient?
The gradient is a multi-variable generalization of the derivative

Q3 (5 Points) What is backpropagation? Name three backpropagation algorithms and explain how they work. List the pros and cons of your three backpropagation algorithms.

What is backpropagation?
Backpropagation is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network.

Name three backpropagation algorithms and explain how they work. List the pros and cons of your three backpropagation algorithms.

Gradient descent

Pro: simple idea, and each iteration is cheap

Pro: Very fast for well-conditioned, strongly convex problems

Con: Often slow, because interesting problems aren't strongly convex or well-conditioned

Con: can't handle nondifferentiable functions

Conjugate gradient method

Pro: The advantage is that the required amount of storage is small, step convergence, high stability, and no external parameters are required.

Con: They usually converge significantly slower than Newton or quasi-Newton methods. In addition, the scale of the step size is usually poorly mastered, so the line search algorithm may need more iterations at a time to find an acceptable step.

Newton's method

Pro: Second-order convergence, fast convergence

Con: Newton's method is an iterative algorithm. Each step needs to solve the inverse matrix of the Hessian matrix of the objective function. The calculation is more complicated.

Q4 (5 Points) Name two forms of regularization used in neural networks. Explain how they work. Why does one use regularization with neural networks?

Name two forms of regularization used in neural networks. Explain how they work.

early stopping regularization

It works in four steps:

1. Split the training data into a training set and a validation set, e.g. in a 2-to-1 proportion.
2. Train only on the training set and evaluate the per-example error on the validation set once in a while, e.g. after every fifth epoch.
3. Stop training as soon as the error on the validation set is higher than it was the last time it was checked.
4. Use the weights the network had in that previous step as the result of the training run.

dropout regularization

At each training stage, individual nodes are either "dropped out" of the net with probability 1 – p or kept with probability p, so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed. Only the reduced network is trained on the data in that stage. The removed nodes are then reinserted into the network with their original weights.

Why does one use regularization with neural networks?

To Prevent over-fitting

Q5 (5 Points) What is a convolution? Give an example of a convolution on image data with and without padding.

What is a convolution?
A convolution is an integral that expresses the amount of overlap of one function g as it is shifted over another function f. It therefore "blends" one function with another.
Convolution is a general purpose filter effect for images. Is a matrix applied to an image and a mathematical operation comprised of integers. It works by determining the value of a central pixel by adding the weighted values of all its neighbors together. The output is a new modified filtered image

Give an example of a convolution on image data with and without padding.

Q6 (5 Points) What is max-pooling? Why is it used? How does it compare to mean-pooling and min-pooling? Which would you use between max-pooling, mean-pooling and min-pooling?

What is max-pooling? Why is it used?
Max pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.

How does it compare to mean-pooling and min-pooling? Which would you use between max-pooling, mean-pooling and min-pooling?

A max-pool layer compressed by taking the maximum activation in a block. If you have a block with mostly small activation, but a small bit of large activation, you will loose the information on the low activations.
A mean-pool layer compresses by taking the mean activation in a block. If large activations are balanced by negative activations, the overall compressed activations will look like no activation at all.

Max pooling is generally predominantly used

Q7 (5 Points) What is the feature learning pipeline in a CNN? How does it relate to the classification in a CNN? Can the feature learning pipeline be used with techniques like SVMs, Naive Bayes, Decision Trees, or GBMs?

What is the feature learning pipeline in a CNN? How does it relate to the classification in a CNN?
The pipeline we propose for image set clustering is fairly straightforward. It consists in extracting deep features from all the images in the set, by using a deep convolutional neural network pretrained on a large dataset for image classification and then apply a "standard" clustering algorithm to these features.

Can the feature learning pipeline be used with techniques like SVMs, Naive Bayes, Decision Trees, or GBMs?
Yes, Pipeline can be used to chain multiple estimators into one. This is useful as there is often a fixed sequence of steps in processing the data, for example feature selection, normalization and classification.

Q8 (5 Points) What are loss functions? When would one choose cross-entropy vs mean-square error?

What are loss functions?
it's a method of evaluating how well your algorithm models your dataset.

When would one choose cross-entropy vs mean-square error?
Cross-entropy is prefered for classification, while mean squared error is one of the best choices for regression. This comes directly from the statement of the problems itself - in classification you work with very particular set of possible output values thus MSE is badly defined (as it does not have this kind of knowledge thus penalizes errors in incompatible way). To better understand the phenomena it is good to follow and understand the relations between

1. cross entropy
2. logistic regression (binary cross entropy)
3. linear regression (MSE)

You will notice that both can be seen as a maximum likelihood estimators, simply with different assumptions about the dependent variable.

Q9 (5 Points) What is an RNN? What kind of data is an RNN used for? How does a Vanilla RNN differ from an LSTM?

What is an RNN?

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence.

What kind of data is an RNN used for?
RNN mainly solves the processing of sequence data, such as text, voice, video and so on. There is a sequential relationship between the samples of this type of data, and each sample is associated with its previous sample.

How does a Vanilla RNN differ from an LSTM?
Empirically. The criteria is the performance on the validation set. Typically LSTM outperforms RNN, as it does a better job at avoiding the vanishing gradient problem, and can model longer dependences. Some other RNN variants sometimes outperform LSTM for some tasks, e.g. GRU.

Q10 (5 Points) What is a Markov model? What kind of data is a Markov model used for?

What is a Markov model?
a Markov model is a stochastic model used to model randomly changing systems. It is assumed that future states depend only on the current state, not on the events that occurred before it (that is, it assumes the Markov property). Generally, this assumption enables reasoning and computation with the model that would otherwise be intractable.

What kind of data is a Markov model used for?
Hidden Markov model is good at processing sequence data
Timing is just a special case of sequential data, and the Markov model applies to all forms of sequential data.
It is used to process time series data, that is, data with time series relationship between samples.

Q11 (5 Points) What is network initialization? How can it effect a neural network? Name three common approaches for network initialization and why one would choose one over another.

What is network initialization? How can it effect a neural network?
Get the initial weight value for training. The value can be random. But must need something to calculate the cost function, so that further training step have something to optimize.

Name three common approaches for network initialization and why one would choose one over another.

Uniform Distribution Initialization
It is a commonly used heuristic, which is defined as

$$W \sim U\left[-\frac{1}{\sqrt{n_{in}}}, \frac{1}{\sqrt{n_{in}}}\right]$$

where U[−a,a]U[−a,a] is the uniform distribution in the interval (−a,a)(−a,a) and nn is the size of the previous layer

Xavier Initialization
And there are some problems when initializing the weights:

1. If the weights in a network start too small, then the signal shrinks as it passes through each layer until it's too tiny to be useful.
2. If the weights in a network start too large, then the signal grows as it passes through each layer until it's too massive to be useful.

In this case, assigning the weights follow a Gaussian distribution with zero mean and a finite variance is a good choice.

the Xavier initialization, proposed by *Xavier Glorot*and *Yoshua Bengio*, is a Gaussian distribution with zero mean and a suitable variance, which makes sure the weights are "just right", keeping the signal in a reasonable range of values through many layers.

ReLU Initialization
To handle the issue that for some very deep CNNs initialized by random weights drawn from Gaussian distributions with fixed standard deviations, have difficulties to converge, (Typically, used for the layers with ReLU activation function).

Q12 (5 Points) What is transfer learning? When would one use it? Give an example of transfer learning using neural networks.

What is transfer learning?
Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

When would one use it?
Transfer learning is an optimization, a shortcut to saving time or getting better performance.
In general, it is not obvious that there will be a benefit to using transfer learning in the domain until after the model has been developed and evaluated.

There are three possible benefits to look for when using transfer learning:
1. Higher start. The initial skill (before refining the model) on the source model is higher than it otherwise would be.
2. Higher slope. The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
3. Higher asymptote. The converged skill of the trained model is better than it otherwise would be.

Give an example of transfer learning using neural networks.
Transfer Learning with Image Data. This is the result of using transfer learning to do image classification

In [74]: df

Out[74]:

	Validation accuracy finetuning	Validation accuracy freezing
Dataset Hymenoptera	0.954248	0.960784
Dataset Hymenoptera gray	0.921569	0.901961
Dataset Simpsons	0.924552	0.641944
Dataset Simpsons gray	0.901535	0.543223
Dataset Dogs vs Cats	0.9894	0.981
Dataset Dogs vs Cats gray	0.9888	0.9756
Dataset Caltech256	0.673643	0.542511
Dataset Caltech256 gray	0.612819	0.434925

Q13 (5 Points) What is an autoencoder? What are they used for? Explain how an autoencoder works.

What is an autoencoder?
An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervisedmanner.

What are they used for?
The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction.

Explain how an autoencoder works.
They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation.

Q14 (5 Points) What is a variational autoencoder? What are they used for? How do they differ from autoencoders?

What is a variational autoencoder? What are they used for?
In neural net language, a VAE consists of an encoder, a decoder, and a loss function. In probability model terms, the variational autoencoder refers to approximate inference in a latent Gaussian model where the approximate posterior and model likelihood are parametrized by neural nets (the inference and generative networks).

How do they differ from autoencoders?
The key difference beetween and autoencoder and variational autoencoder is

1. autoencoders learn a “compressed representation” of input (could be image,text sequence etc.) automatically by first compressing the input (encoder) and decompressing it back (decoder) to match the original input. The learning is aided by using distance function that quantifies the information loss that occurs from the lossy compression. So learning in an autoencoder is a form of unsupervised learning (or self-supervised as some refer to it) - there is no labeled data.
2. Instead of just learning a function representing the data (a compressed representation) like autoencoders, variational autoencoders learn the parameters of a probability distribution representing the data. Since it learns to model the data, we can sample from the distribution and generate new input data samples. So it is a generative model like, for instance, GANs.

Q15 (5 Points) What are deep generative models? What are they used for?

Deep generative models are neural network models that can replicate the data distribution that you give it. This allows you to generate “fake-but-realistic” data points from real data points.

Generative models need lesser labelled data to train, but at the same time impose stronger assumptions. This means that there is lesser hassle to collect labelled data. Autoencoder is just one architecture of generative model. There are others more powerful architectures out there that use RNNs and CNNs for the modelling. Not to forget, GANs, yet another architecture of generative models. So, depending on the kind of task you are interested in, there are different models.

Q16 (5 Points) What are multilayer perceptron? What are they used for?

What are multilayer perceptron?
A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function.

What are they used for?
MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation.

Q17 (5 Points) In deep generative models what is meant by density estimation? What is meant by latent variable models?

In deep generative models what is meant by density estimation?
Density estimation: estimate the probability density function p(x) of a random variable x, given a bunch of observations {X1, X2, ...}

What is meant by latent variable models?
A latent variable model, as the name suggests, is a statistical model that contains latent, that is, unobserved, variables.

A latent variable model formulates the conditional distribution of the response vector yi = (yi1, . . . , yiT) 0 , given the covariates (if there are) in Xi = (xi1, . . . , xiT) and a vector ui = (ui1, . . . , uil) 0 of latent variables

Q18 (5 Points) What are soft-max functions? What are they used for?

What are soft-max functions?
The softmax function, or normalized exponential function, is a generalization of the logistic function that "squashes" a K-dimensional vector z of arbitrary real values to a K-dimensional vector sigma(z) of real values, where each entry is in the range (0, 1), and all the entries add up to 1.

What are they used for?
The softmax function is often used in the final layer of a neural network-based classifier. Such networks are commonly trained under a log loss (or cross-entropy) regime, giving a non-linear variant of multinomial logistic regression.

The softmax function is used in various multiclass classification methods, such as multinomial logistic regression (also known as softmax regression), multiclass linear discriminant analysis, naive Bayes classifiers, and artificial neural networks.

Q19 (5 Points) What is a session in TensorFlow? Why aren't sessions required for scikit-learn?

What is a session in TensorFlow?
A Session object encapsulates the environment in which Operation objects are executed, and Tensor objects are evaluated.

TensorFlow uses the tf.Session class to represent a connection between the client program---typically a Python program, although a similar interface is available in other languages---and the C++ runtime.

Which means that none the operators and variables defined in the graph-definition part are being executed. For example nothing is being executed/calculated here

Why aren't sessions required for scikit-learn?
Unlike TensorFlow who mainly runs on GPU and deal with large amount of data, Scikit-learn mainly runs on CPU and deal with small amount of data, so it doesn't need sessions to allocate resources(Memory or Machine) to different tasks and get the result asynchronously.

Q20 (5 Points) TensorFlow uses a dataflow graph. What is it? Why is it used?

TensorFlow uses a dataflow graph. What is it?

Dataflow is a common programming model for parallel computing. In a dataflow graph, the nodes represent units of computation, and the edges represent the data consumed or produced by a computation.

Why is it used?

Dataflow has several advantages that TensorFlow leverages when executing your programs:

1. Parallelism. By using explicit edges to represent dependencies between operations, it is easy for the system to identify operations that can execute in parallel.
2. Distributed execution. By using explicit edges to represent the values that flow between operations, it is possible for TensorFlow to partition your program across multiple devices (CPUs, GPUs, and TPUs) attached to different machines. TensorFlow inserts the necessary communication and coordination between devices.
3. Compilation. TensorFlow's XLA compiler can use the information in your dataflow graph to generate faster code, for example, by fusing together adjacent operations.
4. Portability. The dataflow graph is a language-independent representation of the code in your model. You can build a dataflow graph in Python, store it in a SavedModel, and restore it in a C++ program for low-latency inference.

Cites

https://en.wikipedia.org/wiki/Deep_learning
https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/
https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780
https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/
https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
https://ai.stackexchange.com/questions/2526/differentiable-activation-function
https://datascience.stackexchange.com/questions/9233/why-do-activation-functions-have-to-be-monotonic
https://cedar.buffalo.edu/~srihari/CSE574/Chap5/Chap5.5-Regularization.pdf
http://staff.itee.uq.edu.au/janetw/cmc/chapters/BackProp/index2.html
https://pdfs.semanticscholar.org/0465/dca766776bdb51be0472a6654453673ae114.pdf
http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf
http://www.songho.ca/dsp/convolution/convolution2d_example.html
https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks
https://arxiv.org/pdf/1707.01700.pdf
https://blog.algorithmia.com/introduction-to-loss-functions/
https://stackoverflow.com/questions/36515202/why-is-the-cross-entropy-method-preferred-over-mean-squared-error-in-what-cases
https://en.wikipedia.org/wiki/Recurrent_neural_network
https://stats.stackexchange.com/questions/226179/how-to-choose-between-plain-vanilla-rnn-and-lstm-rnn-when-modelling-a-time-serie
https://en.wikipedia.org/wiki/Markov_model
https://www.quora.com/What-is-neural-network-initialization
https://isaacchanghau.github.io/post/weight_initialization/
https://en.wikipedia.org/wiki/Transfer_learning
https://machinelearningmastery.com/transfer-learning-for-deep-learning/
https://en.wikipedia.org/wiki/Autoencoder
https://lazyprogrammer.me/a-tutorial-on-autoencoders/
https://jaan.io/what-is-variational-autoencoder-vae-tutorial/
https://www.quora.com/Whats-the-difference-between-a-Variational-Autoencoder-VAE-and-an-Autoencoder
https://www.quora.com/What-is-a-deep-generative-model
https://www.quora.com/Why-are-deep-generative-models-useful
https://en.wikipedia.org/wiki/Multilayer_perceptron
http://www.cs.toronto.edu/~slwang/generative_model.pdf
https://en.wikipedia.org/wiki/Softmax_function#Neural_networks
https://www.tensorflow.org/guide/graphs
https://www.tensorflow.org/guide/graphs
https://en.wikipedia.org/wiki/Gradient