CS194-129 Designing, Visualizing & Understanding Deep Neural Networks F2016

Midterm

NAME_____ **SID**_____

This is a closed-book exam with 11 questions. You can use one page of notes. Please write all your answers in this book. There are a total of 80 points for all questions, and you have 80 minutes to complete the exam. Please budget your time accordingly. Good luck!

1. **(28 points)** Give short answers for each of the following questions – 2 points each
   a) For a fully-connected deep network with one hidden layer, increasing the number of hidden units should have what effect on bias and variance? Explain briefly.(2 points)

Adding more hidden units should decrease bias and increase variance. In general, more complicated models will result in lower bias but larger variance, and adding more hidden units certainly makes the model more complex. See the Lecture 2 slides on the "rule of thumb" for bias and variance.

   b) What's the risk with tuning hyperparameters using a test dataset? (2 points)

Tuning model hyperparameters to a test set means that the hyperparameters may overfit to that test set. If the same test set is used to estimate performance, it will produce an overestimate. Using a separate validation set for tuning and test set for measuring performance provides unbiased, realistic measurement of performance.

   c) What kinds of dataset are difficult for a linear classifier to correctly classify? (2 points)

Linear classifiers delimit linearly separable classes, which roughly corresponds to a single exemplifier. In input space, each class must be a convex set. The simplest counter-example is a non-convex set which is a bimodal class, e.g. which contains images of two very different breeds of cat.

d) Compare bias and variance of a linear classifier to kNN (2 points)

kNN can fit a dataset to arbitrarily high accuracy given enough points, so its bias can be arbitrarilty small. A linear classifier can never do better than fit a single hyperplane so its bias will be larger. The predictions from kNN depend only on a small number of data points (k) and so are subject to high variance – they will change with changes in any of these points. The predictions from a linear classifier depend on all the data points. Its variance is typically much smaller for that reason.

e) Compare SVM and Softmax loss for linear classifiers. Which (in general) has a unique solution for its parameters? What is typically done when there is no unique solution? (2 points)

The softmax loss has continuous and smooth derivatives everywhere. It typically has a finite set of local optima, one of which is also a global optimum. SVM loss is not smooth because of the max against 0. Its derivative will also often vanish over large volumes of the input, leading to many output points with the same loss. In particular, the optimum loss may correspond to a large volume of the input. If there is no unique solution, we add a regularization term to the loss which creates an additional gradient on in the input space. Generally L2 loss is used since it is curved everywhere and provides unique optima.

f) Newton's second-order method is rarely used for deep network optimization because? (2 points)

There are several reasons (a) It requires computation of the Hessian over the entire dataset, which is incompatible with minibatch updates (b) Neural networks often have very high parameter dimension. If the dimension is w the Hessian would require O(w^2) elements to represent and O(w^3) steps to invert. (c) such an inverse is sensitive to individual data values which are often "noisy" for practical datasets and the inversion step will be inaccurate.

g) What is the typical goal of (good) weight initialization? (2 points)

The goal of good weight initialization is to break symmetry and generate activations that lead to non-zero initial gradients. Weights should be large enough that gradients don't decay to zero through a deep network, and not so large that non-linear layers saturate. A good approach is "Xavier initialization." See lecture 6, slides 53+.

h) Why is scaling and shifting often applied after batch normalization? (2 points)

The original batch normalization paper has the following nice description:

"Simply normalizing each input of a layer may change what the layer can represent. For instance, normalizing the inputs of a sigmoid would constrain them to the linear regime of the nonlinearity. To address this, we make sure that the transformation inserted in the network can represent the identity transform. […] These parameters are learned along with the original model parameters, and restore the representation power of the network"

i) What is the role of pooling layers in a CNN? (2 points)

Pooling layers have several roles: (a) they allow flexibility in the location of certain features in the image, therefore allowing non-rigid or rotated or scaled objects to be recognized, (b) they allow unions of features to be computed, e.g. blue eyes or green eyes (c) they reduce the output image size.

j) What is an ensemble algorithm? Give one example from the class of an ensemble model that gave improved performance over a base model. (2 points)

Ensemble algorithms are models that are built from several (typically simpler) models. Methods include bagging (sum of independent models) or boosting (sequential training with "difficult" examples for earlier classifiers emphasized in later ones). The models being combined may be the same (with random variation between them) or different. The Manning et al. translation model used both – different models with hard and soft attention, different similarity measures for words, and several random examples of each.

k) The word "adversarial" in the acronym for GANs suggest a two-player game. What are the two players, and what are their respective goals? (2 points)

The two players are the "generator" and the "discriminator." The generator creates a stream of synthetic images. The discriminator processes a stream of randomly interleaved real and synthetic images produced by the generator and attempts to classify them. The discriminator is trained to maximize the log probability of the correct class label. The generator by contrast attempts to *minimize* this log probability, i.e. it tries to defeat (fool) the discriminator. Thus the design is adversarial and game-like.

l) We described neural parsers which used sequence-to-sequence RNN arrays and were trained on sentence/parse-tree pairs. These models are very sensitive to consistency in the training data. Give a method to improve the consistency of sentence/parse trees pairings. (2 points)

The parser trained better on synthetic labels generated by another state-of-the-art parser. It did even better when trained on a dataset of sentences on which two different machine parsers agreed.

m) A convolutional neural network has 4 consecutive 3x3 convolutional layers with stride 1 and no pooling. How large is the support of (the set of image pixels which activate) a neuron in the 4^{th} non-image layer of this network? (2 points)

With a stride of 1, and a 3x3 filter, and no pooling, this means the "outer ring" of the image gets chopped off each time this is applied. Hence, this reduces the dimension from (nxn) to ((n-2)x(n-2)). We get, working backwards:

1x1 <- 3x3 <- 5x5 <- 7x7 <- 9x9

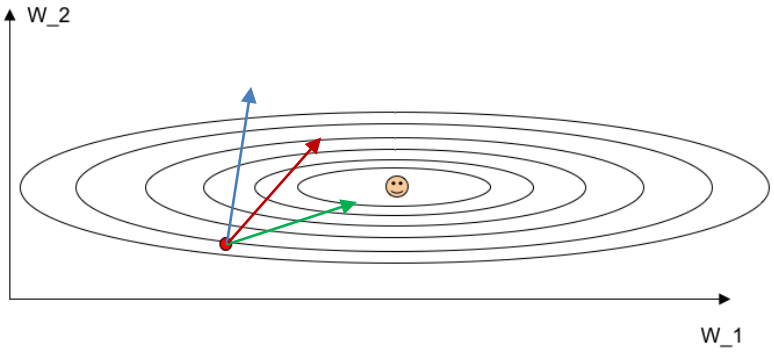
Thus, the support is 9x9=81 pixels.

n) Define the following terms:
   i. Hard attention
   ii. Soft attention

   Which attention models can be trained with backpropagation only? What other training method is required? (2 points)

Hard attention: use a subset of the input to derive the output, where the influence of other parts of the input is zero. That means that gradients in parts of the network not involved in processing the attention area are zero. It also means that hard attention models cannot be trained directly with gradient methods, since their weights are 0 or 1. Thus non-gradient methods, such as REINFORCE must be used.
Soft attention: use non-zero, continuous weights across the entire input. Gradients are non-zero, and standard back-propagation methods can be used.

2. **(6 points)** On the plot below, show ***one gradient step*** (with an arrow) for each of these methods. Make sure you label your three arrows:
    a. Standard Gradient
    b. Natural gradient (or Newton's method)
    c. ADAGRAD or RMSprop (assume they have run for a while to accumulate gradient information)



The standard gradient should be perpendicular to the tangent line at the point (blue arrow). As a general rule, the standard gradient points at the **negative** of the direction of greatest increase.

The natural gradient should be diagonal and point directly to the happy face (green arrow). This is a quadratic, so Newton's method approximates the curvature very well.

ADAGrad/RMSprop normalize gradients based on the average root-mean-squared gradient in a region. Here the Y:X gradients are about 3:1. Dividing the true gradient by these values gives a vector which is approximately at 45 degrees (red arrow).

3. **(6 points)** Why is random hyperparameter search usually preferred over grid search? Draw a picture to illustrate your answer.

The loss function typically depends on only a few hyperparameters (i.e. a subset of them). Performing grid search means that hyperparameter values project onto each other, which reduces the effective number of points in the search. By doing random search, the distribution of points on a subspace or parameter space is still random and uniform. Thus all points are used and the point density is much higher than for grid search.

4. **(4 points)** Look at the original image (top) and then the left and right images below. Infer (qualitatively) what kind of (2x2) convolutional filter was applied to each of the lower images. Write a quick description.

Left: a filter detecting vertical edges. This could be:

[1 -1]

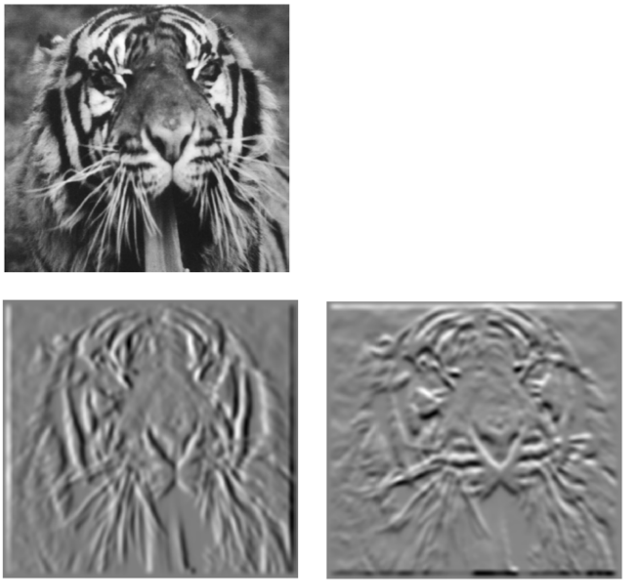
[1 -1]

Right: a filter detecting horizontal edges. This could be:

[-1 -1]

[1 1]

In neither case are we dealing with **line** detectors.



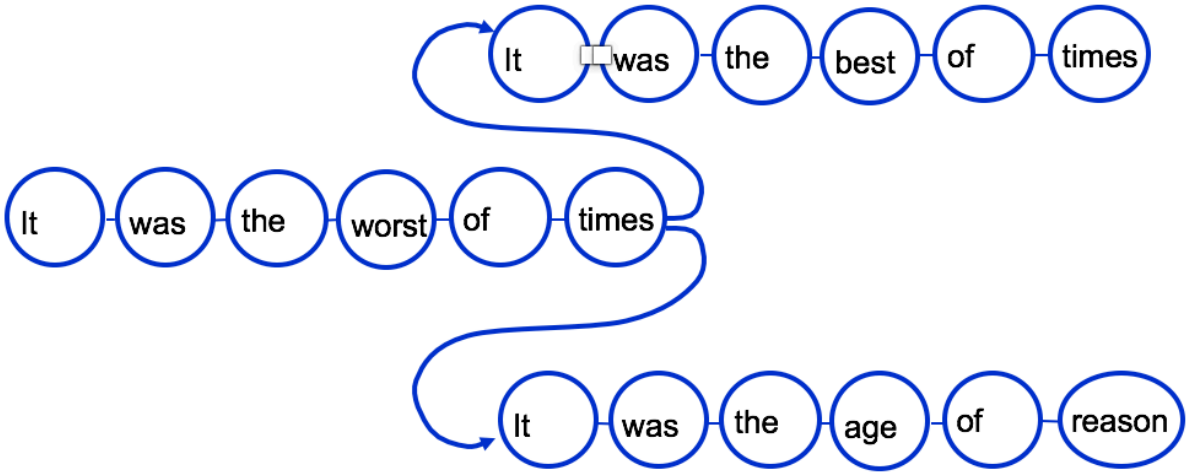
5. **(6 points)** Backpropagation can be used to find an image that activates a particular neuron. “Guided backpropagation” (Karpathy lecture) is a refined approach which generates more informative images. How does it work?

Guided backpropagation means that, as the gradient propagates backward through ReLU regions, the network will not only apply the standard backpropagation rule through ReLU regions (i.e. performing a “max” based on the forward direction), but it will also apply an extra max rule over the backpropagated gradients to discard any of the negative gradients. This allows only positive values to be emphasized. It’s a bit of a hack, but seems to have intriguing results.

Note: for more details, see the paper “Striving for Simplicity: the All Convolutional Net”.

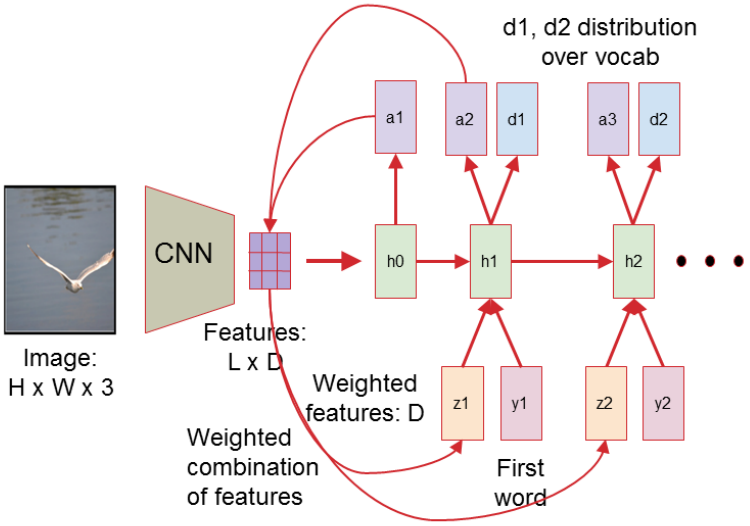
6. **(6 points)** For the three sentences: "**It was the best of times. It was the worst of times. It was the age of wisdom**" map the content onto a skip-thought model. Sketch the model with circles to represent RNN nodes such as LSTMs.

The result is shown below. The center sentence forms the encoder. It feeds two decoder networks, one for the preceding sentence and one for the following sentence.

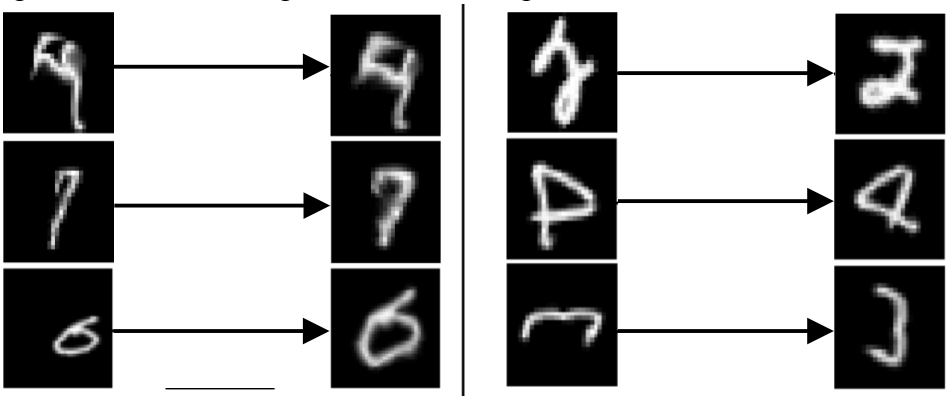


7. **(6 points)** For the attention-based captioning system below, explain what "actions" are taken by the action output modules (labeled a1, a2 etc).

This is an attention model, which means it selects a global weighted combination (assuming soft attention) or a subset (hard attention) from the CNN's output (not the original image!). This kind of CNN should output a last spatial layer (not an FC layer). The actions choose the weighted combination (or subset) of features to emphasizes the regions in image space corresponding to the most relevant part of the image for the caption under the current context (which *varies* according to the spot we are in the caption). For instance, the last word may be "water" if the caption is attempting to say something like "bird flying over water". In that case, the actions would decrease the strength of weights on the actual bird and increase weights on the water region.



8. **(4 points)** Infer the operation performed by the following network from these input-output image pairs. It's the same operation across all pairs. Describe:



Notice that in each case the network "normalizes" the input. i.e. it corrects for rotation, skew, flip and aspect ratio. This is the purpose of *spatial transformer networks*. It saves a later network form having to learn those transformations, and generally improves accuracy and reduces the amount of training data needed. Spatial transform module are typically inserted in front of standard deep networks and can be trained using backpropagation.
Note: there were a lot of students who attempted to describe one operation for the three on the left, and one for the three on the right. But the question stated that the same network was used across all pairs.

9. **(4 points)** For the policy gradient update below, what is the role of the baseline?:

Rewards (and returns) can be large, vary dramatically over time, and may also always be positive. That means the terms that sum to form a policy gradient $\hat{g}$ have large magnitude. Although they may cancel to produce an unbiased result, the sum has large variance. An *advantage* estimator instead has approximately zero mean, and is formed by subtracting from the return a baseline estimator computed over the entire state space. The terms in the sum for $\hat{g}$ are now smaller and the sum has lower variance. In the simplest case, the baseline is a linear function of the current state.

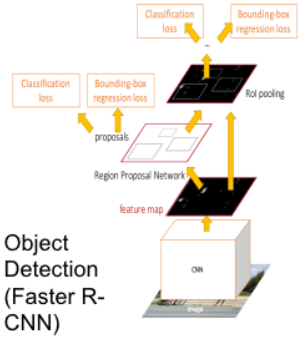
See **"**High-Dimensional Continuous Control Using Generalized Advantage Estimation**"** John Schulman et al.

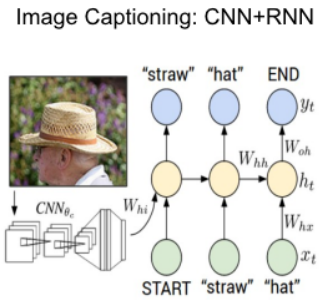
.

Initialize policy parameter θ, baseline b
for iteration=$1, 2, \ldots$ **do**
Collect a set of trajectories by executing the current policy
At each timestep in each trajectory, compute
the *return* $R_t = \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$, and
the *advantage estimate* $\hat{A}_t = R_t - b(s_t)$.
Re-fit the baseline, by minimizing $\|b(s_t) - R_t\|^2$,
summed over all trajectories and timesteps.
Update the policy, using a policy gradient estimate $\hat{g}$,
which is a sum of terms $\nabla_\theta \log \pi(a_t \mid s_t, \theta)\hat{A}_t$
end for

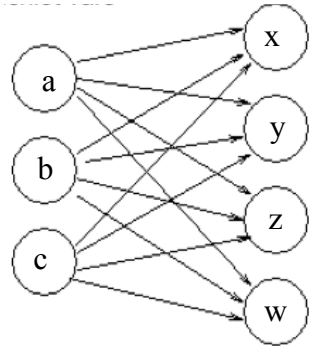
10. **(5 points)** For the image below, highlight two modules that could leverage pretrained models. What pretrained models could you use?:

Two modules that could leverage pretrained models could be the CNN modules in both images. They could use models such as AlexNet, WaveNet, ResNet, etc, or anything trained on ImageNet. Another option would be the word encoding and decoding nodes in the RNN in the image caption region, which are green and purple respectively. These could be trained using a word2vec model, e.g. the publicly available word2vec models from Google.





11. **(5 points)** Contrast collapsed Gibbs sampling and block Gibbs sampling on the following directed graphical model: (assume distributions on left and right nodes are conjugate so collapsing is possible).



which method of sampling (collapsed or block) is typically used in Restricted Boltzmann machines and why?

Collapsed: this will integrate out variables so we can perform sampling on the remainder. For example, to sample x above, normal Gibbs sampling would sample from p(x | a, b, c). A collapsed sampler would integrate out a, b, c and allow sampling from p(x | y, z, w). While this saves some explicit sampling for a,b,c variables, it does so at the expense of additional dependencies between x,y,z,w which were previously independent.

Block: this will perform Gibbs sampling by sampling groups of independent variables in parallel. In this particular graph, we can sample a,b,c in parallel and then z,y,z,w in parallel, and then continue alternating. A bipartite graph has two blocks of independent variables, and so the entire graph can be sampled in two rounds.

In RBMs, we typically use block sampling which is more efficient when the graph structure has large independent blocks. RBMs have a bipartite structure by design and so are a good match for block sampling.

Note: for full credit on this question, students needed to describe briefly how collapsed and block sampling work on the actual graph. To restate the question: "**contrast** […] **on the following directed graphical model**."