Practice Midterm

This is a closed-book exam with 8 questions. You can use one sheet 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. **(34 points)** Give short answers for each of the following questions – 2 points each
    a) Describe two difficulties that computer vision algorithms face in dealing with images. i.e. two characteristics of image formation that make it difficult to recover the image content. (2 points)

    b) Deep networks typically require a lot of data to train from scratch, but can be "fine tuned" for another task quickly. Explain this in terms of the features computed in the layers. (2 points)

    c) What are expected risk and empirical risk? What do machine learning algorithms minimize? (2 points)

    d) Why isn't L2 loss used to optimize a logistic regression model? What loss is used instead and why? (2 points)

    e) Newton's second-order method converges to what kinds of points on the loss function surface? List all that apply. (2 points)

    f) SVMs use a hinge loss that maximizes a *margin* around the decision boundary. What are the benefits of a max-margin classifier such as SVM? Feel free to use a picture (2 points)

    g) The multi-class SVM classifier we derived in class used a margin loss. How was this margin defined? (2 points)

    h) What is the relationship between multiclass logistic regression with binary data and multiclass naïve Bayes with binary data? (2 points)

i) How does the effective learning rate of ADAGRAD vary with the number of steps, t ? (2 points)

j) The loss functions of deep networks are non-convex and there was concern in the early days of the field that gradient methods may get trapped in poor (i.e. high loss) local optima. Describe with a sketch how local optima were observed to behave on a simple model, than explain why SGD performs well in practice on such models. (2 points)

k) If a data block in a convolutional network has dimension $H \times W \times D = 200 \times 200 \times 128$, and we apply a convolutional filter to it of dimensions $H_F \times W_F \times \mathrm{D} = 7 \times 7 \times 128$, what is the dimension of the output data block? (2 points)

l) Why are convolutional layers more commonly used than fully-connected layers for image processing (2 points)

m) Dropout layers implement different forward functions at train and test time. Explain what they do. Let p be the probability that node value is *retained*. (2 points)

n) Why does prediction averaging work for most ensembles of models, but parameter averaging only works for models which are snapshots from a single model during training? (2 points)

o) Sketch a simple recurrent network, with input x, output y, and recurrent state h. (2 points)

p) Give the update equations for a simple RNN unit in terms of x, y, and h. Assume it uses tanh non-linearity. (2 points)

q) What is the difference between the c and h recurrent states in an LSTM (Long Short-Term Memory) recurrent unit? (2 points)

2. **(8 points)** In class, for the multivariate linear regression model $y = Ax$ we wrote the squared error loss L for n data items as

$$L(A) = \sum_{i=1}^{n} (x_i^T A^T - y_i^T)(Ax_i - y_i)$$

And showed that it is minimized when:

$$A = M_{yx} M_{xx}^{-1} \quad \text{where} \quad M_{xx} = \sum_{i=1}^{n} x_i x_i^T \quad \text{and} \quad M_{yx} = \sum_{i=1}^{n} y_i x_i^T$$

Suppose now we add an L2 regularizer term $\lambda \|A\|^2$ to the loss. What is the new solution for A? How does this regularizer make it easier to minimize the error for ill-conditioned systems?

3. **(8 points)** Suppose a deep network layer performs a simple bias normalization:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} X_i$$

$$Y_i = X_i - \mu \quad \text{for} \quad i = 1, \ldots, N$$

Where N is the dimension of input X and output Y. Note that the sum is taken over the coordinates of the input sample, not the elements of a minibatch as in batch normalization.

Compute the jacobian in element form, i.e.

$$\frac{dY_j}{dX_i} = \cdots$$

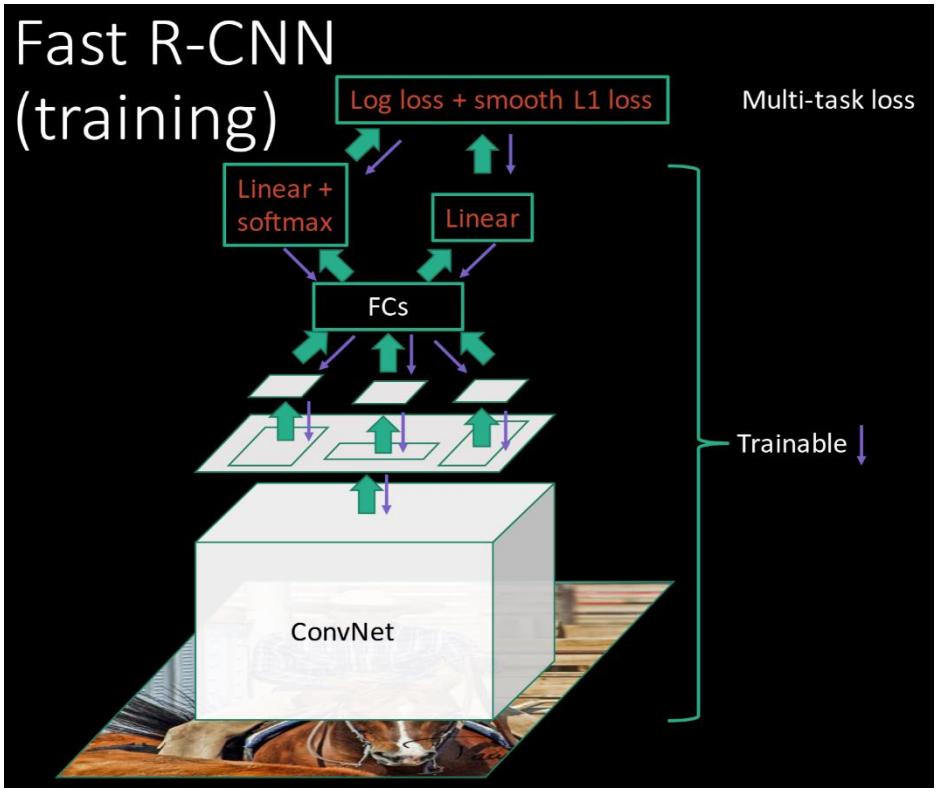
Given the output jacobian vector $J_L(Y)$, how quickly can you compute the input jacobian $J_L(X)$ as a function of N?

4. **(6 points)** Suppose you have a dataset with N samples. You would like to train a deep network and tune learning rate, minibatch size and regularization to give the best model. Describe the cross-validation design you would use to tune the hyper-parameters, and produce the most accurate possible (unbiased) estimate of the model's test loss. i.e. describe how to partition the data, and what to do with each partition.

5. **(6 points)** Using diagrams, explain how standard momentum and Nesterov accelerated gradient differ. How does their performance (convergence as a function of time T) differ on convex optimization problems?

6. **(6 points)** Compare GoogLeNet and Residual networks (ResNets). What are the main architectural features of each, and how did they lead to improvements over previous design? Use diagrams of the networks as appropriate

(6 points) Modify the diagram below which shows Fast R-CNN, to represent Faster R-CNN. Explain in words what changed between the models.



7. **(6 points)** Briefly contrast the backward updates for a ReLU layer using (i) normal backpropagation, (ii) guided backpropagation and (iii) deconvolution