CS194-129 Designing, Visualizing & Understanding Deep Neural Networks SP18

Practice Midterm

The exam will be closed-book, but you can bring one two-side page of notes. Here are some practice questions from a previous midterm. Since the previous exam also covered natural language and RL, the first SP18 midterm will be about 2x longer than this.

1. **(16 points)** Give short answers for each of the following questions – 2 points each
    a) 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.

    b) 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.

    c) 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. (d) Newton's method finds all zeros of gradients, which include saddle points as well as loss minima. It can therefore get stuck as saddles which are more common than minima.

    d) 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 60+.

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

The original batch normalization paper says:
"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"

f) 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.

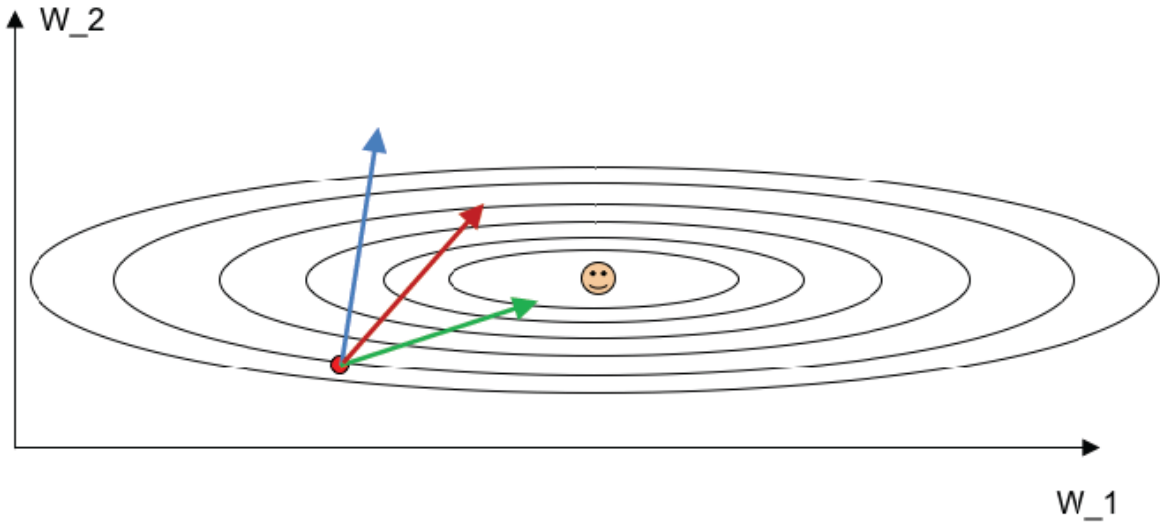
g) What is an ensemble algorithm? Give one example from the class of an ensemble model that gave improved performance over a base model. (2 points)

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). In class we described an ensemble of VGG classifiers bagged together for CIFAR prediction. We described both a true ensemble and an ensemble of snapshots of a single model during training.

h) 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.

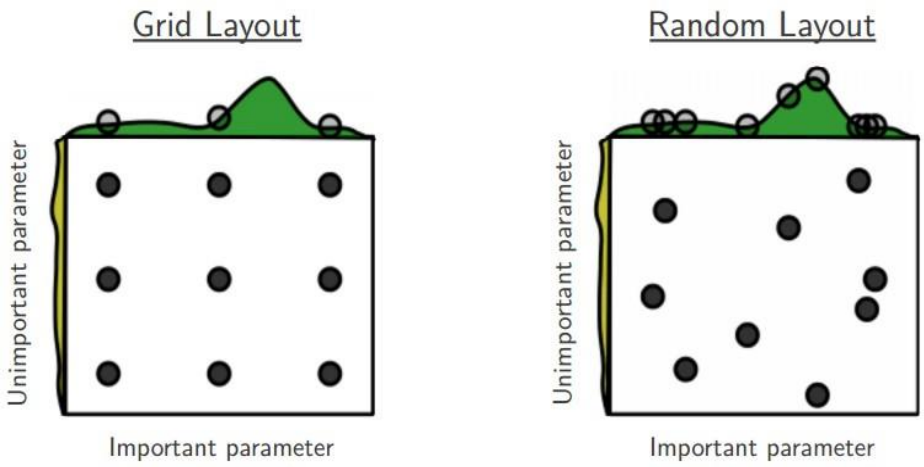
2. **(6 points)** On the plot below, show ***one gradient step*** (with an arrow) for each of these methods. Make sure you label your three arrows:
    a. Standard Gradient
    b. Natural gradient (or Newton's method)
    c. ADAGRAD or RMSprop (assume they have run for a while to accumulate gradient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



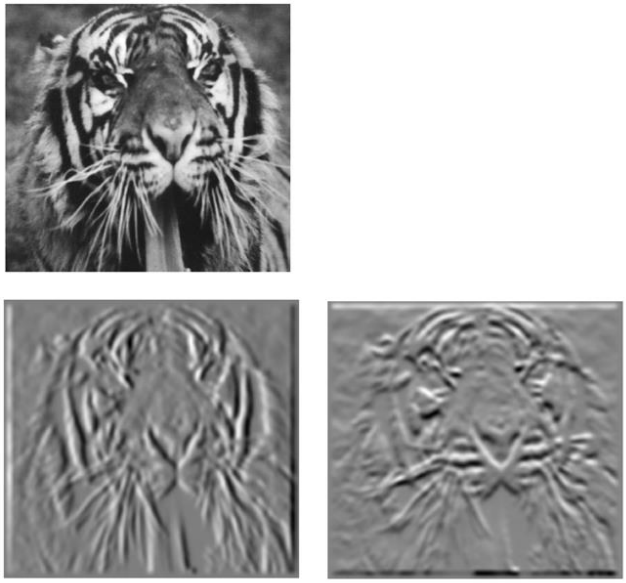
The standard gradient should be perpendicular to the tangent line at the point (blue arrow). For a loss, the standard gradient points at the **negative** of the direction of greatest increase.
The natural gradient should be diagonal and point directly to the happy face (green arrow). This is a quadratic, 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 may depend 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.

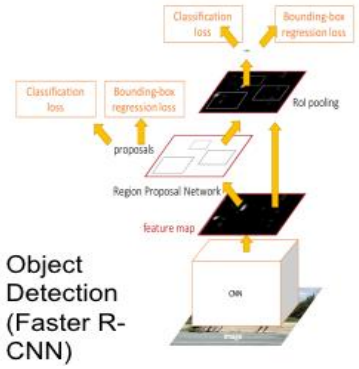


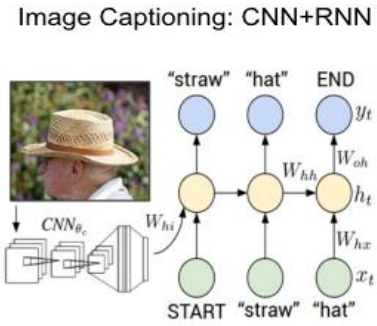
Left: a filter detecting vertical edges. This could be:
[1 -1]
[1 -1]
Right: a filter detecting horizontal edges. This could be:
[-1 -1]
[1 1]
In neither case are we dealing with **line** detectors.

5. **(6 points)** Backpropagation can be used to find an image that activates a particular neuron. "Guided backpropagation" (Visualization 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. **(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.