CS 189 Spring 2019 — Introduction to Machine Learning

Exam Prep Dis 4 Solutio

Q1. Multiple Choice

(a) Suppose when you are training your convolutional neural network, you find that the training loss just doesn't go down after initialization. What changes will likely fix this problem?

- [✓ Change the network architecture.]
- [✓ Change learning rates.]
- [✓ Ensure training data is being read correctly.]
- ○ Use dropout.
- [✓ Normalize the inputs to the network.]
- ○ Add ℓ_2 regularization.

A too-simple network might be unable to learn any significant features of the data (consider a single convolution layer with a single filter, kernel width/height 1, and high stride, so that the resulting feature map is a single value). A high learning rate can (almost) always lead to divergent training loss. If training data is processed incorrectly, the network might not learn anything useful (e.g. if it were transformed into white noise). Dropout and ℓ_2 penalties are both forms of regularization that likely only improve validation loss. Normalizing the inputs can make the objective better-conditioned, which can improve the convergence rate of SGD.

(b) In this question, we will contrast convolutions with fully connected layers. Suppose the input is an $n \times n$ image with c channels. First, consider a convolution with kernel width, height, and stride all equal to 1. Let m denote the number of output filters. Second, consider a convolution with kernel width and height equal to n, with stride 1 and m output filters. Which of the following are true statements?

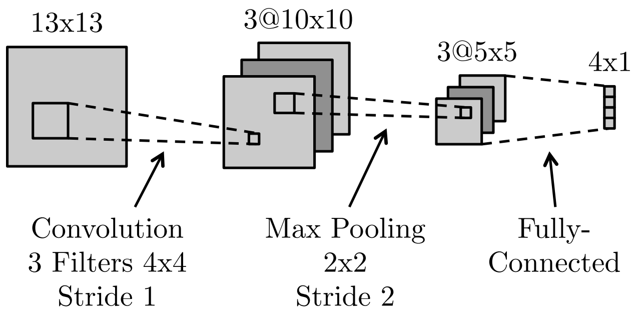
- ○ The first convolutional layer is equivalent to a fully connected layer.
- [✓ The second convolutional layer is equivalent to a fully connected layer.]
- ○ The second convolutional layer has fewer parameters than the first.
- [✓ Both convolutions are a form of dimensionality reduction if $m < c$.]

The first convolution is equivalent to learning a fully connected layer for each coordinate (i, j) of the input. For the second one, each kernel takes a linear combination of all the entries of the input image. The first convolution has $c \times m$ parameters, while the second convolution has $n \times n \times c \times m$ parameters. Each of the m outputs of the second convolution are a linear combination of all entries in the input, i.e. it is a linear map from $\mathbb{R}^{n*n*c}$ to $\mathbb{R}^m$.

The first convolution reduces the number of channels from c to m, and the second convolution reduces the image to m scalars.

Q2. Convolutional Neural Nets

Below is a diagram of a small convolutional neural network that converts a 13x13 image into 4 output values. The network has the following layers/operations from input to output: convolution with 3 filters, max pooling, ReLu, and finally a fully-connected layer. For this network we will not be using any bias/offset parameters (b). Please answer the following questions about this network.



(a) How many weights in the convolutional layer do we need to learn?

48 weights. Three filters with 4x4=16 weights each.

(b) How many ReLu operations are performed on the forward pass?

75 ReLu operations. ReLu is performed after the pooling step. ReLu is performed on each pixel of the three 5x5 feature images.

(c) How many weights do we need to learn for the entire network?

348 weights. 48 for the convolutional layer. Fully-connected has 3x5x5=75 pixels each connected to four outputs, which is 300 weights. Pooling layer does not have any weights.

(d) True or false: A fully-connected neural network with the same size layers as the above network (13x13 $\rightarrow$ 3x10x10 $\rightarrow$ 3x5x5 $\rightarrow$ 4x1) can represent any classifier that the above convolutional network can represent.

[✓ True] ○ False

(e) What is the disadvantage of a fully-connected neural network compared to a convolutional neural network with the same size layers?

There are too many weights to effectively learn.

Q3. PCA

You are given a design matrix $X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$. Let's use PCA to reduce the dimension from 2 to 1.

- Compute the covariance matrix for the sample points. (Warning: Observe that X is not centered.) Then compute the **unit** eigenvectors, and the corresponding eigenvalues, of the covariance matrix. *Hint:* If you graph the points, you can probably guess the eigenvectors (then verify that they really are eigenvectors).

  The mean value of the data is $\hat{\mu} = [2, 1]$. The centered data is $X_c = X - \hat{\mu}$. The covariance matrix is $\frac{1}{n}X_c^\top X_c = \frac{1}{4}\begin{bmatrix} 82 & -80 \\ -80 & 82 \end{bmatrix}$. Normalizing with $1/(n-1)$ is also acceptable.

  Its unit eigenvectors are $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ with eigenvalue $2/4$ and $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$ with eigenvalue $162/4$. (Note: either eigenvector can be replaced with its negation.)

- Suppose we use PCA to project the sample points onto a one-dimensional space. What one-dimensional subspace are we projecting onto? For each of the four sample points in X (not the centered version of X!), write the coordinate (in principal coordinate space, not in $\mathbb{R}^2$) that the point is projected to.

  We are projecting onto the subspace spanned by $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$. (Equivalently, onto the space spanned by $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Equivalently, onto the line $x + y = 0$.) The projections are $(6, -4) \rightarrow \frac{10}{\sqrt{2}}$, $(-3, 5) \rightarrow -\frac{8}{\sqrt{2}}$, $(-2, 6) \rightarrow -\frac{8}{\sqrt{2}}$, $(7, -3) \rightarrow \frac{10}{\sqrt{2}}$.

- Given a design matrix X that is taller than it is wide, prove that every right singular vector of X with singular value σ is an eigenvector of the covariance matrix with eigenvalue σ^2.

  If v is a right singular vector of X, then there is a singular value decomposition $X = UDV^\top$ such that v is a column of V. Here each of U and V has orthonormal columns, V is square, and D is square and diagonal. The covariance matrix is $X^\top X = VDU^\top UDV^\top = VD^2V^\top$. This is an eigendecomposition of $X^\top X$, so each singular vector in V with singular value σ is an eigenvector of $X^\top X$ with eigenvalue σ^2.