Computer Vision Exam

Robert Joseph

December 2, 2020

Abstract

CMPUT 328

Questions

Q1-Q5 are attached as images at the end

Question 6 - Suppose for MNIST logits values are $[-0.1, 12.0, 3.0, -3.2, -8.0, 17.0, 1.9, 0.6, 4.9, -12.0]$. **Compute softmax function on logits and write them.**

- $P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}}$
- b1 - 3.7208742661313E-8
- b2 - 0.006692805834611
- b3 - 8.2595785683953E-7
- b4 - 1.6762241789593E-9
- b5 - 1.3794900990286E-11
- b6 - 0.99330045717478
- b7 - 2.7493748689512E-7
- b8 - 7.4929206275792E-8
- b9 - 5.5222670446003E-6
- b10 - 2.5266242504393E-13

Question 7 - : Compute cross entropy loss value between softmax values in Q6 against a ground truth vector $[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]$

- $H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$
- $E_p[l] = -E_p\left[\frac{\ln q(x)}{\ln(2)}\right] = -E_p\left[\log_2 q(x)\right] = -\sum_{x_i} p(x_i) \log_2 q(x_i) = -\sum_{x} p(x) \log_2 q(x) = H(p, q)$
- Cross Entropy Loss : = 2.17439 (Reference : Python Program that I created)

Question 8 - What is the minimum value of cross-entropy loss? What is the maximum value?

- Min value : 0
- Max value : ∞ (Not really defined as the range of the Cross Entropy loss is $\mathbb{R}^+$ but we can neglect that for the true probability)

Question 9 - For image classification task, explain in two to three sentences why machine learning is a more suitable tool than rule-based AI?

- Rule Based AI in simple terms are like a block of if-else statements and Machine Learning on the other hand is more of learning from the data presented and makes inferences and so this is much more preferable as this can work in a variable environment. A simple example would be for image classification ie : One cannot hardcode a program which can recognize an image(maybe we can but it wouldn't work on a generalized set of varied images) where as using deep learning we could not only classify most images even surpassing humans if using the right architecture but also generate realistic data, do data augmentation and various other tricks.

Question 10 - Explain in two to three sentences how k-nearest neighbor method works.

- k - nearest Neighbour algorithm as the name itself suggests classifies an object to the class of "k" nearest neighbours. KNN algorithm is a supervised ML algorithm that basically during training phase computes a L^P norm on the training data and where during the testing phase we just compute the k nearest training examples to each test data point. An average function is used then to predict the class the testpoint belongs to.

Question 11 - In two to three sentences explain underfitting and overfitting in machine learning.

- Underfitting : When a model yields both low accuracy on the training and test data.ie in terms of statistical bias and variance it means low variance and high bias.
- Overfitting : When the model yields a high accuracy on the training set but cannot generalize the result to the test set and hence a drop in accuracy. ie in terms of statistical bias and variance it means high variance and low bias.

Question 12 - You are using a convolutional neural net architecture for image classification. You notice that your convnet is underfitting the dataset. How would you overcome underfitting here?

- Underfitting results in the fact that the model might be too simple. Hence increasing the complexity of the model ie increasing the number of learnable parameters. Another way is to remove noise from the data or get more training data or even increase the training time.

Question 13 - How would you tackle overfitting in an image classification task with convnet

- Handling Overfitting is pretty much doable due to the number of recent techniques that have researchers have come up with. Some of them include
  - K - Cross Validation
  - Regularization (Included in most Optimizers as a weight decay Hyperparameters)
  - Dropout
  - Batch Normalization (Although according to the original paper they talk about the concept of internal covariant shift ; one hasnt yet tested this to be statistically true although in application it has been shown to yield better results)
  - Pruning especially during descision trees
  - Annealing the learning rate
  - Early stopping is another concept where we stop the training process during an iterative method

Question 14 - Starting with $x = -1$ and $y = 1$, apply gradient descent to minimize $f(x, y) = x^2 + y^4$ .First write gradient of f, then write gradient descent algorithm. Assume a suitable step length value. Compute values of $f(x, y)$ for four successive iterations.

- Number of iterations: 1; alpha = 0.01
  - Theta Value for x(x0) = -0.980000000000000
  - Theta1 Value for y(y0) = 0.960000000000000
- Number of iterations: 2
  - Theta Value for x(x0) = -0.960400000000000
  - Theta1 Value for y(y0) = 0.924610560000000
- Number of iterations: 3
  - Theta Value for x(x0) = -0.941192000000000
  - Theta1 Value for y(y0) = 0.892992403919713
- Final Result (Reference : Python Program that I coded)
- The function is $x^2 + y^4$
- Number of iterations: 4
  - Theta Value for x(x0) = -0.922368160000000
  - Theta1 Value for y(y0) = 0.864508252531925

Question 15 - When would you prefer to solve linear regression by gradient descent over its closed form solution?

- One important fact to note that even though closed form solutions are much more reliable however most of the times we are dealing with huge datasets and hence the data would not fit into the memory and hence an approximate solution should be used.

Question 16 - Explain in two to three sentences why convolution neural networks in more suitable than a fully connected network for visual recognition tasks?

- First of all a FCC has much more parameters(imagine an image with a billion pixels) than a Convolution NN. Unless we have a lot of GPU's and a lot of distributed optimization this is very hard to deal with.
- Secondly CNN uses the concept of spatial invariance for images and preserves translation invariance and even the locality principle.

Question 17 - Explain in two to three sentences why the number of parameters in a convolution layer is usually much smaller than that in an equivalent fully connected layer.

- As stated in the previous answer FCC have more learnable parameters as each parameter in a FCC is connected with the previous layer to all neurons. Unlike CNN where this is not the same case.

Question 18 - How do you reduce the dimension of an activation map? Consider two cases in your answer: spatial dimensions and depth (i.e., number of activation maps).

- One can use Max/Average pooling for reducing the spatial dimensions.
- For reducing the depth one can either increase the stride along with larger kernel sizes. To preserve the spatial dimensions padding relevant to the increase in strides/kernel sizes should be appropriately done.

Question 19 - Consider d activation maps of height and width h and w, respectively. You want to construct d activation maps, each of height and width $h/2$ and $w/2$, respectively. Write the specifications of a convolution (not max pooling) operation you would apply to achieve this..

- We need to solve the following equation

$$\frac{(h - f + 2p)}{s} + 1 = \frac{h}{2}$$

. Rearranging this for f gives us

$$f = (h + s + 2p) + \frac{sh}{2}$$

Now forcing this to be an integer by letting s= 2 and p = 0 implies

$$f = h + 2 - h = 2$$

Therefore this yields a filter map of height h = 2.

- We need to solve the following equation

$$\frac{(w - f + 2p)}{s} + 1 = \frac{w}{2}$$

. Rearranging this for f gives us

$$f = (w + s + 2p) + \frac{sw}{2}$$

Now forcing this to be an integer by letting s= 2 and p = 0 implies

$$f = w + 2 - w = 2$$

Therefore this yields a filter map of width w = 2.

- You can yield other valid set of h, w by playing around with p however one set generates an countably infinite set ie

$$p = n \in \mathbb{N} \rightarrow \{p + 4, p + 4\} = \{h, w\} \in \mathbb{N} * \mathbb{N}$$

Question 20 - You are applying four 3-by-3 convolution filters to ten activation maps of spatial dimensions $28x28$. Write the dimensions (height, width and depth) of the output activation map. Assume stride 1 and valid option for the convolution operation.

- Applying the direct formula 4 times

$$\frac{(h - f + 2p)}{s} + 1 = h'$$

$$\frac{28 - 3}{1} + 1 = 26 = 24 = 22 = 20 = h'$$

-

$$\frac{(w - f + 2p)}{s} + 1 = w'$$

$$\frac{28 - 3}{1} + 1 = 26 = 24 = 22 = 20 = w'$$

Therfore the final dimension is $10 * 20 * 20$.

Question 21 -How many learnable parameters are there in a layer that implements a max pooling operation with stride 2 and filter size $2x2$?.

- 0 as there are no learnable parameters in the inputlayer as well as max/average pooling layers.

Question 22 -How many learnable parameters are there in layer that implements an average pooling operation with stride 2 and filter size 2-by-2?.

- 0 as there are no learnable parameters in the inputlayer as well as max/average pooling layers.

Question 23 - Consider a convolution layer with filter size 5 x 5 x 10 x 5. How many learnable parameters are there in this layer?.

- Total Number of learnable parameters with bias is : (5*5*10+1)*5 = 1255
- Total Number of learnable parameters without bias is : (5*5*10)*5 = 1250

Question 24- Exercise on convnet forward and backward passes. This was posted on eclass.

- No comment

Question 25 - What is the main difference between ResNet and AlexNet architectures

- ResNet has blocks called residual models which try to learn the features by adding learning the output of each layer which is basically the convolution of its input plus the identity mapping.It also uses a very deep architecture (152 layers or such) and skip connections.
- AlexNet on the other hand is a much simpler architecture after LeNet and consist of only 11 layers in total.

Question 26 -Which architecture would require more memory: a DenseNet or a ResNet? Explain your answer in two to three sentences.

- DenseNet definitely requires more memory like the name suggests. It was modelled after NIN and is much more denser in its architecture. Although in size its smaller than ResNet. This is simply understood during back-propagation DenseNet requires storing every single layers output. The number of layers in a DenseNet ¿¿ ResNet hence more memory usage.

Question 27 - When you convert a fully connected network to an equivalent fully convolutional network, what do you gain?

- No change in parameters except we may now can use the concept of spatial invariance and such which could increase the accuracy. Another thing we gain is the fact that there is a variable input size.

Question 28 - When you convert a fully connected network to an equivalent fully convolutional network, how does the number of learnable parameters change?

- No change in parameters.

Question 29 - Explain in two to three sentences why YOLO object detector is faster than faster RCNN object detector

- YOLO = You only look once model has only a single stage which detects objects compared to faster RCNN which has a two stage process. First it must train the region of interest and then classification is done and hence why it is slower.

Question 30 - Explain in two to three sentences the role of a region proposal net in an object detector.

- (Refernece : TowardsDataScience) Region of Interest (ROI) pooling is used for utilising single feature map for all the proposals generated by RPN in a single pass. ROI pooling solves the problem of fixed image size requirement for object detection network.ROI pooling produces the fixed-size feature maps from non-uniform inputs by doing max-pooling on the inputs.

Question 31 - Why do you need a fully convolutional neural network for semantic segmentation task? Limit your answer to two to three sentences.

- We do need FCNN as semantic segmentation deals with images. FCNN not helps with classification but also on local tasks and advanced in bounding box object detection. Another fact is that for each pixel to pixel a class must be assigned and using upsampling techniques this can be achieved.

Question 32 - As opposed to a convolutional neural net designed for image classification, you apply a fully convolutional net for semantic segmentation. Why? Limit your answer to two to three sentences.

- Reference (Original Paper - Fully Convolutional Networks for Semantic Segmentation) An FCN naturally operates on an input of any size, and produces an output of corresponding (possibly resampled) spatial dimensions. This requires it to have more features which we can tune even more. While a general deep net computes a general nonlinear function, a net with only layers of this form computes a nonlinear filter, which we call a deep filter or FCCN.

Question 33 - What role does a transposed convolution play in semantic segmentation?

- The key role is to do upsampling in semantic segmentation.

Question 34 - What is a skip connection in semantic segmentation?

- Reference(AI Summer) Skip connections are used to pass features from the encoder path to the decoder path in order to recover spatial information lost during downsampling. Reference(Eclass Web Link) Indeed, we can recover more fine-grain detail with the addition of these skip connections.

Question 35 - Why autoencoder implements unsupervised learning and not supervised learning? Limit your answer to two to three sentences

- Supervised learning is learning to predict a given target from a given input which are all labelled.
- However unsupervised learning does not give you the target. A typical autoencoder, creates a system that reduces the dimensionality of the input and extracts the most important features, but since you're not extracting these features to classify the input based on given labels, it's unsupervised learning.

Question 36 - Explain within two to three sentences why an autoencoder tries to implement an identity function.

- A key concept of the autoencoder is to try to learn an approximation to the identity function. By doing so we can discover interesting structure about the data. However we dont want it to become an overcomplete autoencoder and hence sparsity or adding randomness achieves this.

Question 37 - Explain in two to three sentences why a variational autoencoder is called a generative model.

- This is due to the fact that for every latent vector z which is generated we ask ourselves how good is this latent vector in reconstructing the original data x. This is in a similar fashion of an autoencoder where the image is passed through a network and then a laten vector z is present from where the decoder tries to reconstruct the original image from the two bottleneck features ie : the mean and standard deviation features (In the original paper). The aim of VAE is to learn the marginal likelihood of the data in such a generative process.

Question 38 - Explain in a few sentences the working principles behind GANs.

- Reference(Deep Generative Models) A generator model G learns to capture the data distribution and a discriminator model D estimates the probability that a sample came from the data distribution rather than model distribution. Basically the task of the Generator is to generate natural looking images and the task of the Discriminator is to decide whether the image is fake or real. This can be thought of as a mini-max two player game where the performance of both the networks improves over time. In this game, the generator tries to fool the discriminator by generating real images as far as possible and the generator tries to not get fooled by the discriminator by improving its discriminative capability.

Question 39 - Which model is more powerful in your opinion: GANs or variational autoencoders? Why? Limit your answer to a few sentences.

- Generative Adversarial Networks are more powerful than Variational Autoencoders.
- One reason is that at the moment they are better at generating visual features. However there are recent advances in VAE like DeepMind VAE which are comparable to GAN however as far as most models have been tested GAN outperforms VAE.

Question 40 - What is truncated backpropagation through time? Explain in a few sentences with a simple diagram, if possible.

- Reference(Quora) - In truncated backprop through time (TBPTT), the input is treated as fixed length subsequences. For forward pass, the hidden state of previous subsequence is passed on as input to the next subsequence. However, in gradient computation, the computed gradient values are dropped at the end of every subsequence as we walk back.That is in standard backprop the gradient values at time t' used in every time step t if $t < t'$. In truncated backprop, the gradients do not flow from t' to t if $t' - t$ exceeds the subsequence length.

Question 41 - The recurrent neural network architecture is not acyclic. How does then backpropagation work for recurrent neural networks.

- Uses the concept of Unfolding which maps from left to right or use parameter sharing, this way you can easily backprop. The Recurrent net needs to be unfolded through time for a certain amount of timesteps.

Question 42 - : What are the vanishing and exploding gradient problems? How does a recurrent neural network overcome these issues?

- Reference (MachinelearningMastery)Exploding Gradients : In deep networks or recurrent neural networks, error gradients can accumulate during an update and result in very large gradients. These in turn result in large updates to the network weights, and in turn, an unstable network. At an extreme, the values of weights can become so large as to overflow and result in NaN values. The explosion occurs through exponential growth by repeatedly multiplying gradients through the network layers that have values larger than 1.0.

- Solving Exploding Gradient : Use Gradient Clipping. There is also the possibility of avoiding this by re-designing the model network and using Regularization to penalize the model.

- Vanishing Gradient Problem - A problem with training networks with many layers (e.g. deep neural networks) is that the gradient diminishes dramatically as it is propagated backward through the network. The error may be so small by the time it reaches layers close to the input of the model that it may have very little effect.

- Vanishing gradients is a particular problem with recurrent neural networks as the update of the network involves unrolling the network for each input time step, in effect creating a very deep network that requires weight updates

- Solve this issue - Use LSTM or even Echo State networks(Reference(Superdatascience)) that are designed to solve this or even use weight initialization so that this issue is minimized.

Question 43 - Explain in a few sentences how a recurrent neural network is used in image captioning task?

- Firstly an encoder is used which is basically a CNN that encodes the image into a feature vector and then a decoder is used which is a RNN that decodes this back and generates the words.

- For training our RNN/LSTM model, we predefine our label and target text.

Question 44 - Explain how a teacher-student training of model compression work in a few sentences.

- The Teacher- Student Training is another name for Model Knowledge Distillation where we reduce the size of he model without loosing too much of its predictive powers.
- In simple terms for the student model we train it on the data labels.
- For the teacher model we pre-train the the model using the a larger network which achieves a better accuracy and this is the ground truth label.
- This is then compared to the student model and checked if the knowledge has been transferred to the student. More precisely we use the teacher's loss in addition to the student's loss to calculate and backpropagate the gradients in the Student model's networ

Question 45 -: Explain why bi-linear transform is differentiable in a few sentences. Use mathematical symbols if needed.

- Not covered

1)

I

1	3	2	4	6	4
4	8	3	1	0	2
2	1	4	3	9	1
4	7	2	3	9	2

4×6

W

-1	0	1
-1	0	1
-1	0	1

3×3

Now J = (4-3+2×0)/1 +1 = 2 ∴ J = 2×4
(6-3+2×0)/1+1 = 4

∴ J =

2	-4	6	-1
-1	-9	9	-2

2×4

K = 2×2 stride = 2 [max pooling]

∴ K = (2-2+2×0)/2 +1 = 1
K = (4-2+2×0)/2 +1 = 2
∴ K = 1×2

∴ K =

2	9

1×2

∴ O = Relu(K)
= [2 | 9]

∴ QED

3) $y_i^p = \dfrac{e^{x^T w_j}}{\sum_k e^{x^T w_k}}$

y

$$L = -\sum_{k=1}^{n} y_k \log(y_k^p)$$

$$L = -\sum_{k=1}^{n} y_k \log\left(\frac{e^{x^T w_k}}{\sum_i e^{x^T w_i}}\right)$$

$$L = -\sum_{k=1}^{n} y_k \left(\log(e^{x^T w_k}) - \log\left(\sum_i e^{x^T w_i}\right)\right)$$

$$L = -\sum_k y_k \left(x^T w_k - \log\left(\sum_i e^{x^T w_i}\right)\right)$$

$$\frac{dL}{dw_{ik}} = -y_k \left(x^T - \delta_{ik} \left(\frac{1}{\log\left(\sum_i e^{x^T w_i}\right)} e^{x^T w_k}\right) \cdot x^T\right)$$

$$\frac{dL}{dw_{ik}} = -y_k \left(x^T - \delta_{ik} \frac{x^T \cdot e^{x^T w_k}}{\log\left(\sum_i e^{x^T w_i}\right)}\right)$$

δ_{ik} = Kronecker delta function

$$\begin{cases} 0 & i \neq k \\ 1 & i = k \end{cases}$$

5)

$$y_i^f = (x_i W_1^T + b_1) W_2^T + b_2$$

y

$$L^2 = \frac{1}{2} \sum_{i=1}^{n} \| y_i^p - y_i \|^2 + \frac{\gamma}{2} \| W_1 \|^2 + \frac{\beta}{2} \| W_2 \|^2$$

Let $t_i = x_i W_1^T + b_1^T$

and $z_i = t_i W_2^T + b_2$

now to find $\frac{dL^2}{dW_1^T} = \frac{dL^2}{dz_i} \frac{dz_i}{dt_i} \frac{dt_i}{dW_1^T}$

$= (z_i - y_i) \cdot W_2^T \cdot x_i = (y_i^p - y_i) \cdot W_2^T \cdot x_i$

$= \boxed{(y_i^p - y) W_2^T \cdot x}$

$\frac{dL^2}{dW_2^T} = \frac{dL^2}{dz_i} \frac{dz_i}{dW_2^T}$

$= (z_i - y_i) \cdot t_i = (y^p - y) \cdot t = \boxed{(y^p - y)(x W^T + b)}$

Now $\frac{dL^2}{db_1} = \frac{dL^2}{dz_i} \frac{dz_i}{dt_i} \frac{dt_i}{db_1}$

$= (z_i - y_i) \cdot W_2^T \cdot 1 = (y^p - y) W_2^T$

$\frac{dL^2}{db_2} = \frac{dL^2}{dz_i} \frac{dz_i}{db_2} = (z_i - y_i) \cdot 1 = (y^p - y)$

4) $y^p = \dfrac{e^{\text{Relu}(x^T W_1)\cdot W_2}}{\sum_k e^{\text{Relu}(x^T W_1)\cdot W_2}}$

$\log(y^p) = \log\left(\dfrac{e^{\text{Relu}(x^T W_1)\cdot W_2}}{\sum_k e^{\text{Relu}(x^T W_1)\cdot W_2}}\right)$

$\log(y^p) = \text{Relu}(x^T W_1) W_2 - \log\left(\underbrace{\sum_k e^{\text{Relu}(x^T W_1)\cdot W_2}}_{\Omega}\right)$

$$\frac{d\log(y^p)}{dW_1} = \begin{cases} (x^T W_1) x^T & x^T W_1 > 0 \\ 0 & x^T W_1 < 0 \\ \text{DNE} & x^T W_1 = 0 \end{cases} W_2 - \frac{d\log(\Omega)}{dW_1}$$

$$\frac{dy^p}{dW_1} = y^p \left(\begin{cases} (x^T W_1) x^T & x^T W_1 > 0 \\ 0 & x^T W_1 < 0 \\ \text{DNE} & x^T W_1 = 0 \end{cases} W_2 \right) - y^p \frac{d\log(\Omega)}{dW_1} \quad \hookrightarrow (1)$$

now to find $\dfrac{d}{dW_1}\log(\Omega) = \dfrac{e^{\left(\begin{cases} (x^T W_1) & x^T W_1 > 0 \\ 0 & x^T W_1 < 0 \end{cases}\right) x^T W_2}}{\log(\Omega)}$

$$\frac{dy^p}{dW_1} = y^p \left(\begin{cases} (x^T W_1) x^T, & (x^T W_1) > 0 \\ 0, & (x^T W_1) < 0 \\ \text{DNE}, & (x^T W_1) = 0 \end{cases} \right) W_2 - \frac{e^{x^T W_2 \begin{cases} (x^T W_1), (x_1^T W_1) > 0 \\ 0 \quad x_1^T W_1 < 0 \\ \text{DNE} \quad x_1^T W_1 = 0 \end{cases}}}{\log\left(\sum_k e^{\text{Relu}(x^T W_1) W_2}\right)}$$

$\frac{dy^p}{dW_2}$

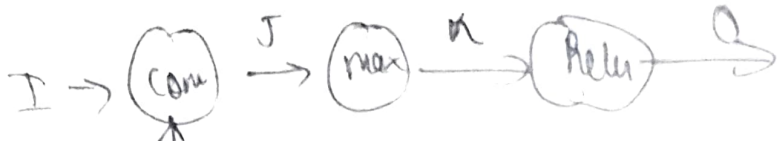
Now $\log(y^p) = \text{Relu}(x^T W_1) \cdot W_2 - \log(\Omega)$

$$\frac{d(\log(y^p))}{dW_2} = \begin{cases} x^T W_1 & x^T W_1 > 0 \\ 0 & x^T W_1 < 0 \\ \text{DNE} & x^T W_1 = 0 \end{cases} - \frac{1}{\Omega} e^{\begin{cases} x^T W_1 & x^T W_1 > 0 \\ 0 & x^T W_1 < 0 \\ \text{DNE} & x^T W_1 = 0 \end{cases}}$$

$\Rightarrow$

$$\frac{dy^p}{dW_2} = y^p \left(\begin{cases} x^T W_1 & x^T W_1 > 0 \\ 0 & x^T W_1 < 0 \\ \text{DNE} & x^T W_1 = 0 \end{cases} - \frac{e^{\begin{cases} x^T W_1 & x^T W_1 > 0 \\ 0 & x^T W_1 < 0 \\ \text{DNE} & x^T W_1 = 0 \end{cases}}}{\sum_R e^{\text{Relu}(x^T W_1) W_2}} \right)$$

where $z = \text{Relu}(x^T W_1)$



2) $I_0 = [7, 10]$

$$L = \frac{1}{2} L_2 = \frac{1}{2} \| I_0 - 0 \|^2$$

$$L = \frac{1}{2}(0 - I_0)^2$$

$$\frac{dL}{d0} = (0 - I_0) = [2, 9] - [7, 10] = [-5, -1]$$

$$\frac{dL}{dK} = [-5, -1]$$

$$\frac{d}{dJ} = \begin{bmatrix} -5 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad 2 \times 4$$

$$\frac{d}{dW} = I \times \frac{d}{dJ} = \begin{bmatrix} -6 & -15 & -10 \\ -24 & -43 & -24 \\ -12 & -8 & -29 \end{bmatrix} \quad (4-2+1) \times (6-4+1)$$

3×3