2

20005010 3

GNR638
Quiz 1

1) A deep CNN trained on a large dataset is prone to overfitting when its weights are used for transfer learning on a small dataset.

2) tanh (i.e., the fxn tanh(x) saturates as $x \to \infty$ / $x \to -\infty$ in that case, grad / differentiation of tanh(x) $\to 0$ which causes vanishing grad problem)

3) The assumptions made are only valid at the start of the training with Xavier initialisation.

4) Yes. (Depends on some other factors such as the distribution of the rest 10k new images but overall, more training data for a regularized deep network is a good approach to improve test accuracy and make the network more generalised to the test data)
or any

5) a class-specific sigmoid is the best loss fxn in a multi-label classification problem. It helps give a binary output 'yes' or 'no' if a particular class exists / is found in a picture given as input.

6) Dropout is a regularization technique and helps prevent overfitting of the training data and improve the overall accuracy of the network.

7) multi-task learning : when 2 tasks have the same dataset.
(eg. classification and object location detection)

8) $z = w^T \cdot a\_prev + b$ [definition of a logit]

$\therefore$ z = np.matmul(np.transpose(w), a_prev) + b

9) no. of parameters in conv layer depends on
- no. of filters (determines output no. of channels)
- size of filters (3x3 or 5x5 or 7x7 ...)
- depth of filters (depends on depth of input image)
- no. of biases (= no. of filters cause $z = \underset{1}{W} * \underset{1}{a\_prev} + \underset{1}{b}$)
kernel inp. image bias

10) learned features are invariant to noise in both cases.

11) $\text{Relu}(z) = \max(0,z)$

$\sigma(\text{Relu}(z)) = \frac{1}{1+e^{-\max(0,z)}} \longrightarrow \begin{cases} \frac{1}{1+e^{-z}} > 0.5 & z > 0 \\ \frac{1}{1+e^{-0}} = 0.5 & z \leq 0 \end{cases}$

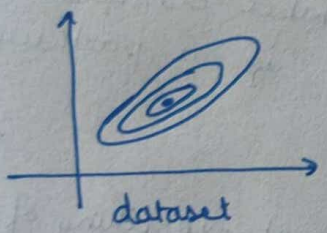
12) $\therefore \sigma(\text{Relu}(z)) \geq 0.5$ always.

$\Rightarrow y' \geq 0.5$ always!

$\therefore$ prediction is that the image is a cat always which makes the classifier obsolete and useless.

(All predictions will belong to one class)

12) batchnorm makes training faster.



using BN

here bigger α (learning rate) can be used to converge faster and make training faster...

(axis just for ref. ~~axis~~ actual axes may differ)

13) Batch GD gives a smooth ~~loss~~ curve: true

Mini BatchGD " " " " " with small batch size : false

MBGD's ~~loss~~ loss curve is overall smooth but has perturbations. and smaller the batch size, greater the perturbations and ruggedness.

14) end-to-end learning doesn't need the costly operation of feature extraction and feature engineering. Generally leads to lower bias. ~~and~~

15) Both a & b are true. A deep net. may overfit and this is generally understood. A very wide network may also overfit. In this case, it may be the case ~~that~~ that particular nodes in the hidden layer are ~~dead~~ activated much more than other nodes/neurons and so overfit the training data in that manner...