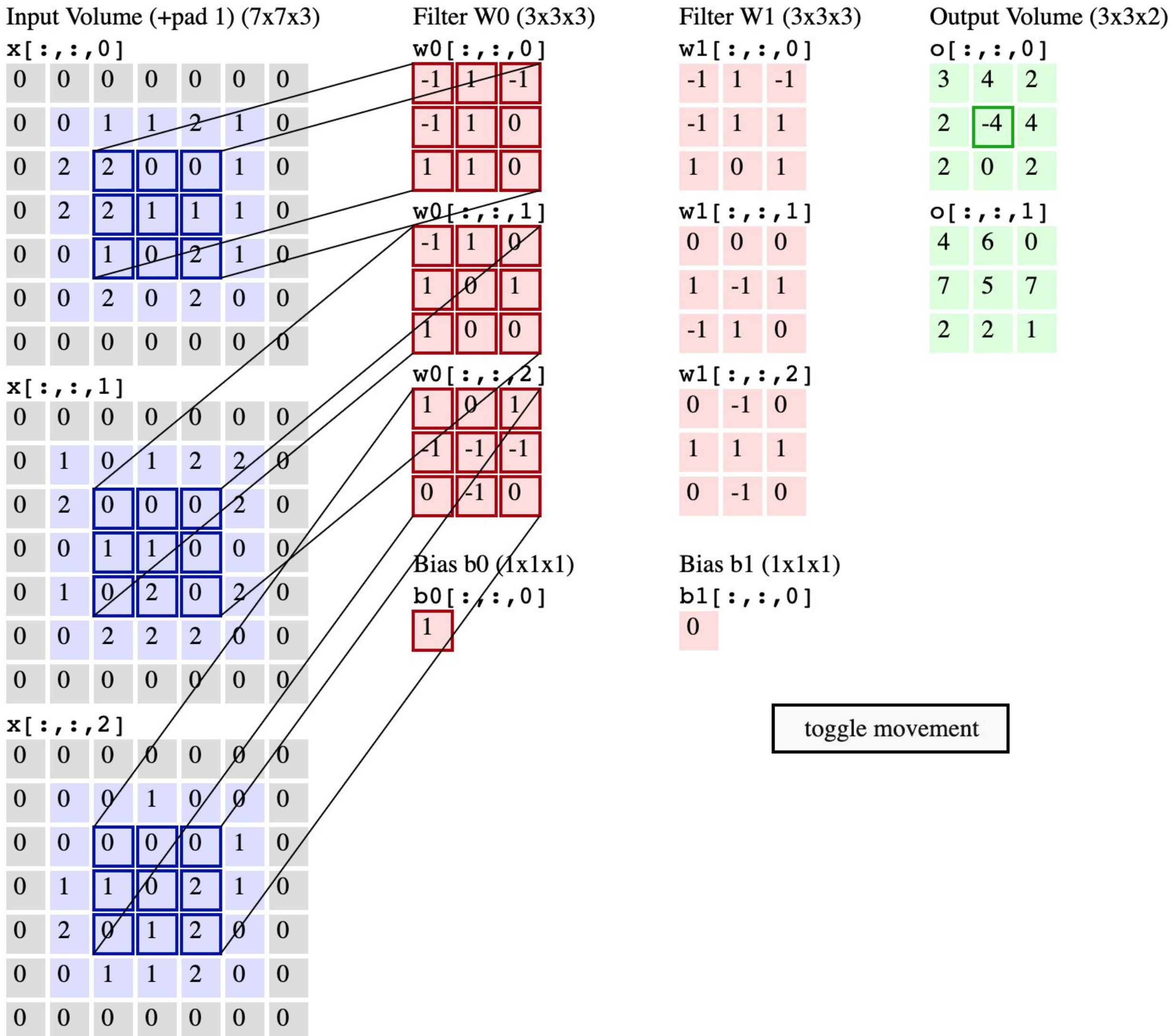
Q1 Forward pass of CNN

Note **this one is zero padded!**



Q2

Step-by-step tutorial for backpropagation in CNN.

https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199

Q3

https://stats.stackexchange.com/questions/235528/backpropagation-with-softmax-cross-entropy

Q4

Check section 5 and 6. Full derivation of formulas https://github.com/Kulbear/deep-learning-coursera/blob/master/Neural%20Networks%20and%20Deep%20Learning/Building%20your%20Deep%20Neural%20Network%20-%20Step%20by%20Step.ipynb

Q5
Same as above

Q6

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

Q7
Softmax part is as same as above, cross entropy is defined as

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

Pytorch NLLLoss, CrossEntropy

Q8
Check the definition of entropy.
Entropy is the measurement of chaosity.

The cross entropy loss function for multiclass can be computed as:

$$-\sum_{i=1}^{N} y_i log\hat{y}_i$$

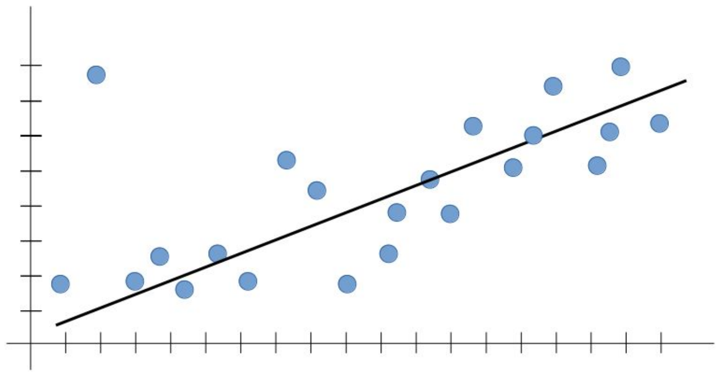
When y_i and \hat{y}_i is very close, (say 1 and 0.999999) then the loss is almost 0. But it can be infinitely large (think about it)

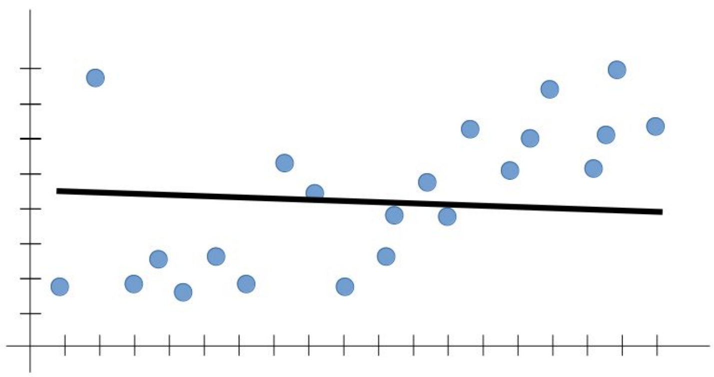
Q9

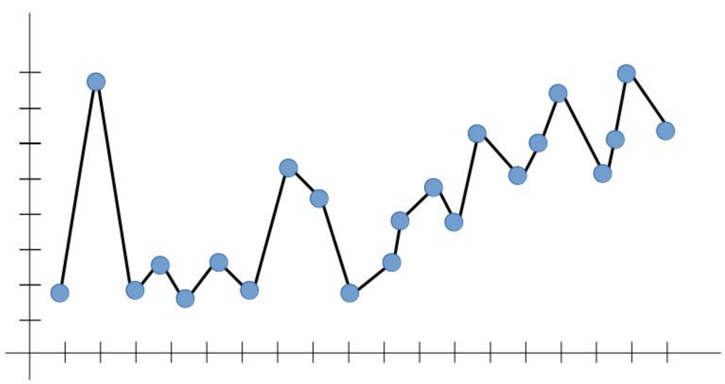
Q10
Knn - non-parameterized

Q11

https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765







Q12

Q13

Q14
https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html

Q15
考虑闭式解所需要的矩阵运算

For linear regression on a model of the form $y = X\beta$, where X is a matrix with full column rank, the least squares solution,

$$\hat{\beta} = \arg\min \|X\beta - y\|_2$$

is given by

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Now, imagine that X is a very large but sparse matrix. e.g. X might have 100,000 columns and 1,000,000 rows, but only 0.001% of the entries in X are nonzero. There are specialized data structures for storing only the nonzero entries of such sparse matrices.

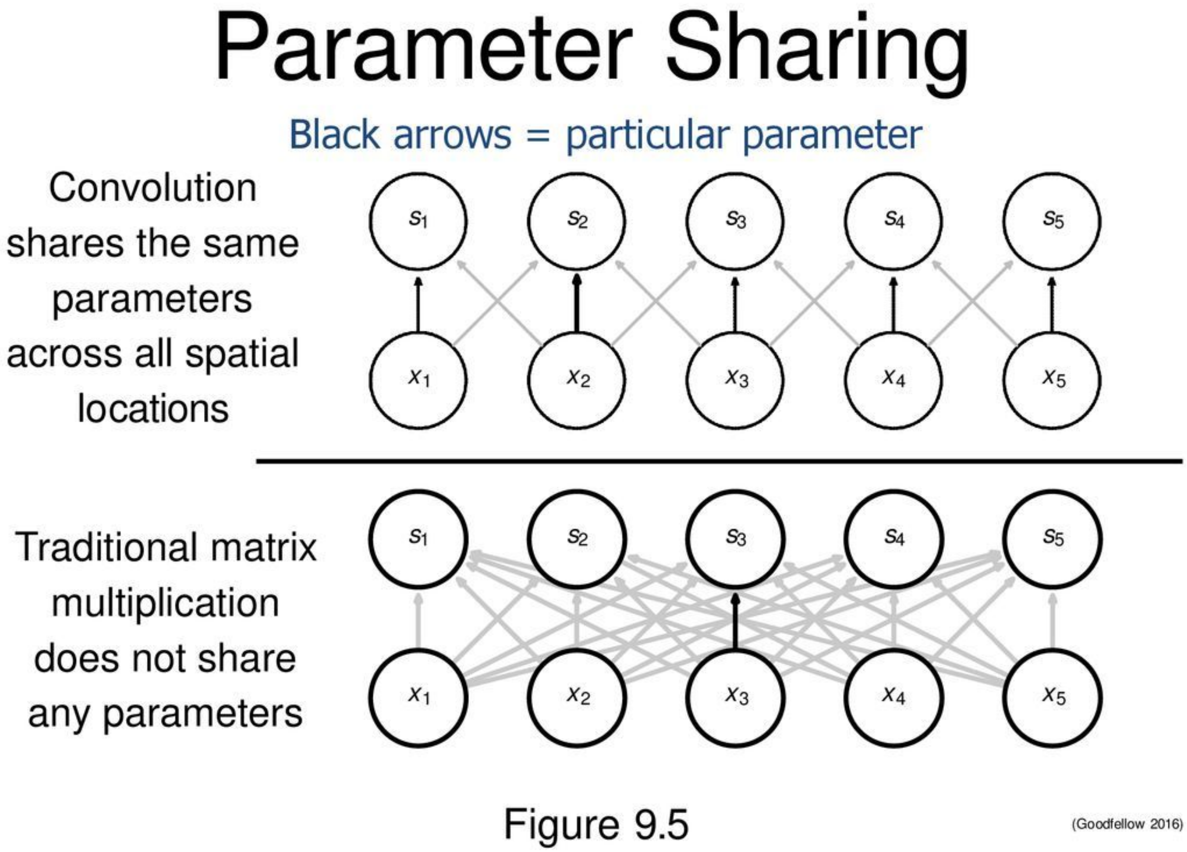
Also imagine that we're unlucky, and $X^T X$ is a fairly dense matrix with a much higher percentage of nonzero entries. Storing a dense 100,000 by 100,000 element $X^T X$ matrix would then require 1×10^{10} floating point numbers (at 8 bytes per number, this comes to 80 gigabytes.) This would be impractical to store on anything but a supercomputer. Furthermore, the inverse of this matrix (or more commonly a Cholesky factor) would also tend to have mostly nonzero entries.

However, there are iterative methods for solving the least squares problem that require no more storage than X, y, and $\hat{\beta}$ and never explicitly form the matrix product $X^T X$.

In this situation, using an iterative method is much more computationally efficient than using the closed form solution to the least squares problem.

Q16
https://www.quora.com/Why-is-CNN-used-for-image-classification-and-why-not-other-algorithms

Q17
Parameter sharing



Q18
Spatially: pooling
Number of activation maps: change conv layer size

Conv layer shape: kernel height x kernel width x in_channel x out_channel(number of filters/activation maps)
5 x 5 x 3 x **2 filters**

Q19
new_height = (input_height - filter_height + 2 * P)/S + 1
new_width = (input_width - filter_width + 2 * P)/S + 1

A conv layer with 2x2xdxd stride size 2

Q20
Use the formula above

Q21

0

Q22
0

Q23

5 x 5 x 10 x 5 + **5**

Q24
See links in Q2

Q25
residual connection
http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture09.pdf

Q26
DenseNet needs more memory

Q27
Hard to tell

Q28
Not changing

Q29

1-stage vs 2-stage
1-stage tends to miss more small objects

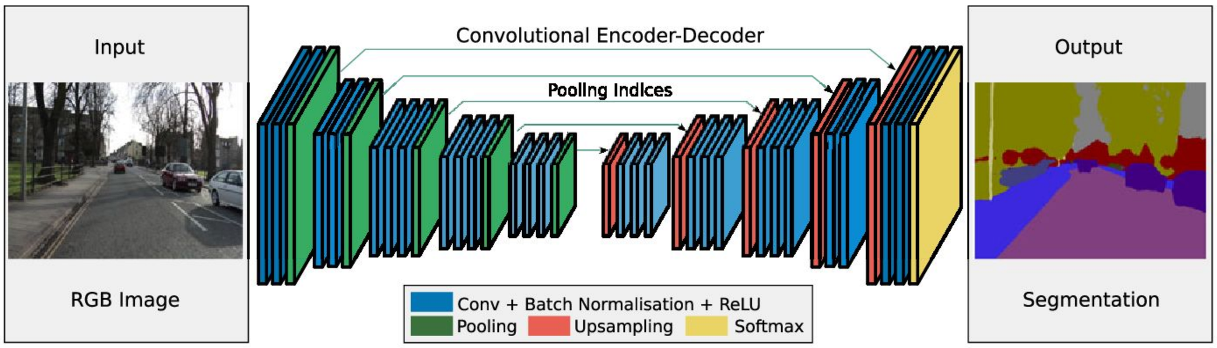
https://everitt257.github.io/post/2018/08/10/object_detection.html

Q30
2-stage methods separate the object detection task into proposal and classification

Q31
The fully connected layer requires you flatten the input to a vector representation which loses the spatial and depth information.

Q32
Same as Q31?

Q33
Upsampling
https://medium.com/activating-robotic-minds/up-sampling-with-transposed-convolution-9ae4f2df52d0

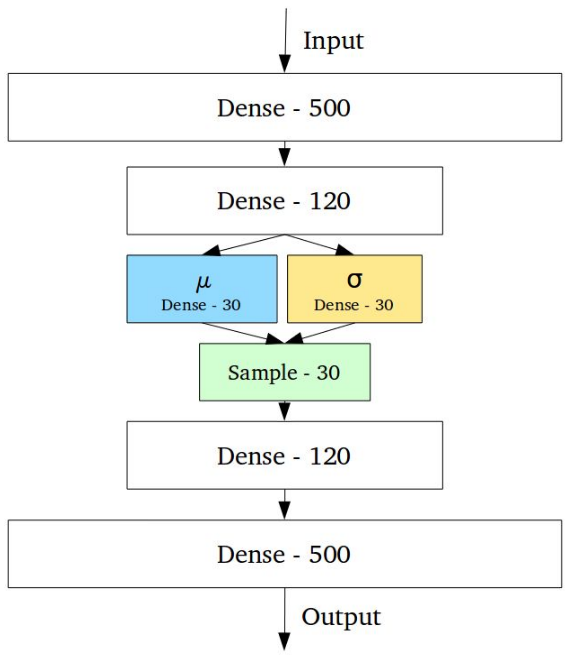


Q34

Q35
Not requirement of annotated data

Q36

Q37
Variational Autoencoders (VAEs) have one fundamentally unique property that separates them from vanilla autoencoders, and it is this property that makes them so useful for generative modeling: their latent spaces are, by design, continuous, allowing easy random sampling and interpolation.
It achieves this by doing something that seems rather surprising at first: making its encoder not output an encoding vector of size n, rather, outputting two vectors of size n: a vector of means, μ, and another vector of standard deviations, σ.



https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

Q38

A minimax game (game theory)

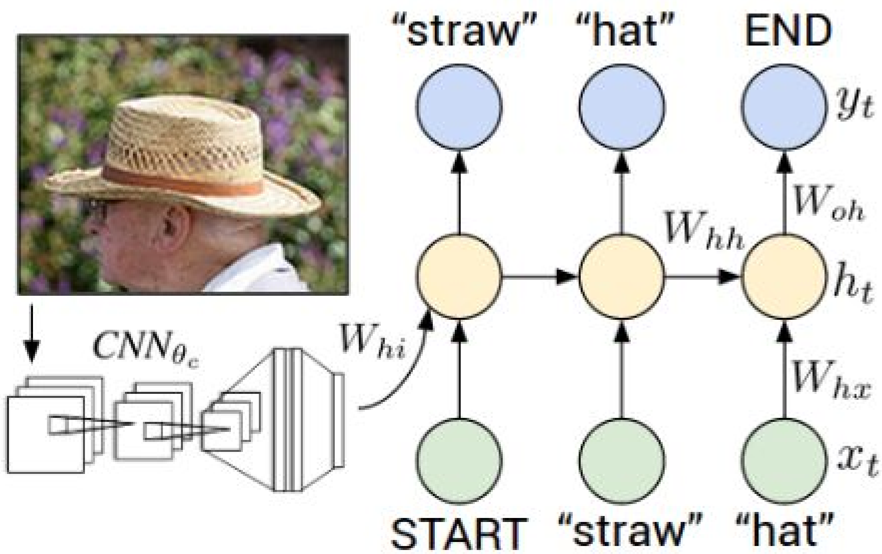
Q39
Kinda Open question

Q40
https://r2rt.com/styles-of-truncated-backpropagation.html

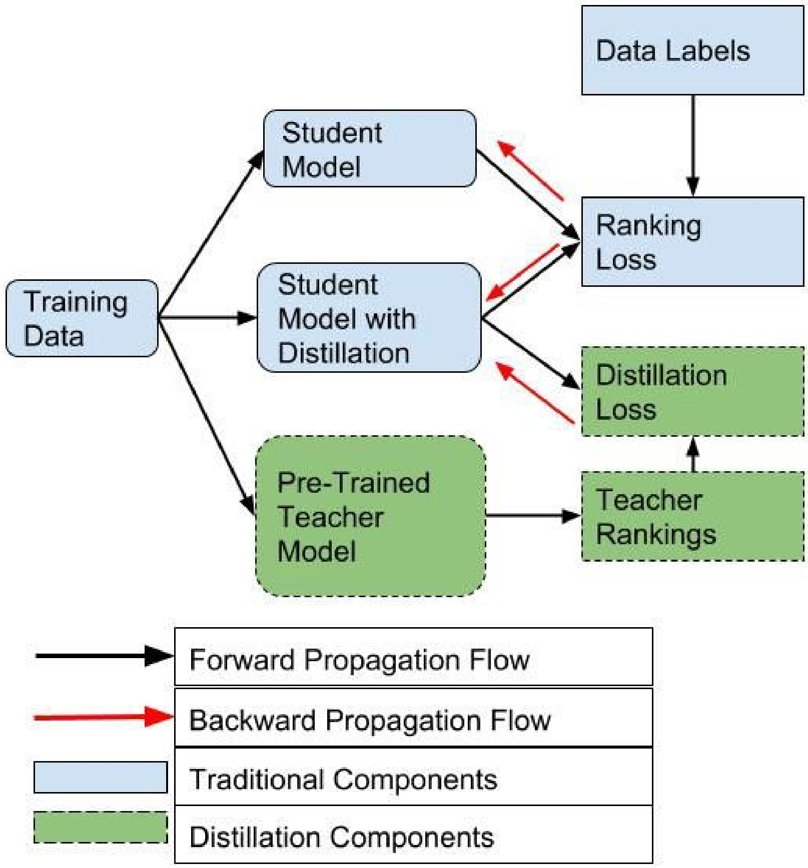
Q41
BPTT

Q42

Q43



Q44
https://towardsdatascience.com/model-distillation-and-compression-for-recommender-systems-in-pytorch-5d81c0f2c0ec
6min read



Q45
Explain why bi-linear transform is differentiable in a few sentences. Use mathematical symbols if needed.