Deep Learning Theory and Applications
(AMTH/CBB 663, CPSC 452/552)
Final Exam Spring 2021
Due Date: May 19, 2021

Total Points: 10+50+30=90
Policy:

This exam is "open book," which means you are permitted to use any materials presented in class, the two class textbooks, and any notes you have taken. The exam must be taken completely alone. Showing it or discussing it with anyone is forbidden. You may not consult any external resources. This includes any online forums, question-answering services, or course help sites. You may not use Google or any other search engines for any reason. You may not use any shared Google documents. You may not consult with any other person regarding the exam. You may not check your exam answers with any person.

Multiple choice (2 points each): You may choose multiple answers.

1. Which of the following is true of stochastic gradient descent?
    a. Large step sizes may cause it to miss a local minima
    b. Large step sizes may cause it to miss a global minima
    c. Small step sizes may cause it to miss a local minima
    d. Has been shown to yield better results when trained with large batches.
    e. Has been shown to yield better results when trained with small batches.

2. What are some known techniques that **prevent overfitting** in neural networks?
    (a) Early stopping
    (b) Continuing to train after convergence
    (c) Randomly setting edges to 0 during training (drop-off)
    (d) Decreasing the number of nodes in the network architecture
    (e) None of the above

3. Which of the following is likely to move a neural network into the ***lazy regime***?
    (a) Very large width
    (b) Very large depth
    (c) Very large dataset size
    (d) Highly scaled activations
    (e) Linearity

Lazy regime: weight and bias don't change much from initialization.

4. Based on the **information bottleneck theory**, which of the following is a sign of a goodneural network embedding in a hidden layer h ?

a. Mutual information between h and outputs is relatively high.
b. Mutual information between h and outputs is relatively low.
c. Mutual information between input and h should be relatively low. Get rid of useless info
d. Mutual information between input and h should be relatively high.
e. Mutual information between h, h+1 is relatively high. Not matter

5. What kind of **invariance** do deep convolutional neural networks induce in image features?

(a) Rotational invariance
(b) Translational invariance
(c) Scale invariance
(d) Permutation Invariance
(e) All of the above.

Conv layer and Pooling layer: can use skip connections

Translational Invariance:

- recognize a pattern regardless of its position in the input image.
- Achieved through the convolution operation, which uses the same set of filters across the entire input.
- Ensures that the learned features are robust to small shifts in the input.
- Reduces the number of parameters, as the same filters are used everywhere, making the model more efficient and less prone to overfitting.

Scale Invariance:

- recognize a pattern regardless of its size in the input image.
- Not inherently present in a standard CNN architecture.
- Can be achieved through various methods, including:
  - Data augmentation: Scaling and resizing input images during training.
  - Multi-scale feature extraction: Using different sizes of filters or incorporating image pyramids.
  - Training on multi-scale representations of input images.

Short answer (5 points each):

1. What is an advantage that a transformer has over an LSTM for machine translation?
Attention mechanism alleviate the memory issue and vanishing gradient issue

What is an advantage that an LSTM has over a transformer?
Faster training and reduced dimensional embedding layer

2. Why do large neural networks have many minima close to the global minima?
Many symmetric variant solutions (weight permutation)
More likely to have saddles points than minima, a mix of pos and neg eigenvalues of Hessian matrix

3. What **inductive biases** are thought to allow neural networks to generalize despite being massively overparameterized?

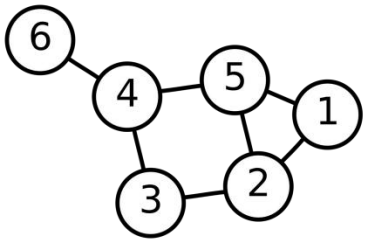
SGD and norm-based regularizations find smooth funcs that fit the data

highly parameterized function classes have more members (large function space) where you can choose smooth solutions that fit the data, allows us to construct progressively better approximations to that smallest norm function

4. Does a linear autoencoder with a K-dimensional bottleneck layer find the same embedding of data as the first K principle components of the data? Why or why not?

Embeddings Not exactly same but span the same subspace
Yes, with linear activation function, the behavior of encoder of Autoencoder is same as PCA like a dimensionality reduction method

5. Compute the unnormalized Laplacian L of the graph below.



L=D-A

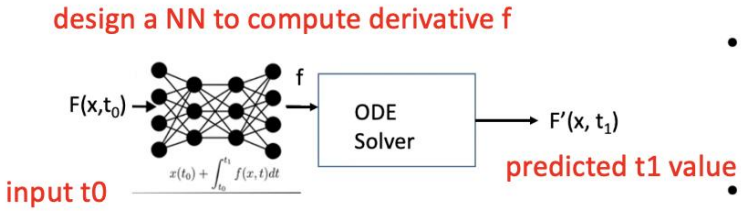
6. Suppose there is a feature f on the nodes of this graph that has the property that L**f** ≅ **0**. Is f a low frequency or high frequency signal?

   f is a column vector, 0 is a column zero vector

   low frequency signal, smooth and invariant feature, bc subtrating feature from its neighbors

7. How are weights of a neural ODE network trained?

   Objective is loss between ground truth F(x, t1) and predicted F'(x, t1)
   Backpropagate through operations of ODE-Solver and network via adjoint sensitivity which measures stability – how much a perturbation to the input of the solver changes the output
   Bc we don't care about training the weights in the ODE-Solver, we only care about train the weights in the network for computing derivative of target function F



8. Suppose you have a 16x16 image with one row/column of zero-padding all alone to be a 18*18 image, and you send it through 8 convolutional filters that are each 3x3 with a stride of 2, what is thesize of the result?

   H=w= (16-3+2*1)//2+1=8
   H*W*n_filter = 8*8*8

   Suppose the output shape is $n_H \times n_W$, we have

$$n_H = \left\lfloor \frac{n_h - k_h + 2p_h}{s_h} \right\rfloor + 1, \; n_W = \left\lfloor \frac{n_w - k_w + 2p_w}{s_w} \right\rfloor + 1$$

   where n_h and n_w is height and weight of input,

   k_h and k_w is height and weight of kernel,

   p_h and p_w is padding for the height and weight

   s_h and s_w is stride for the height and weight

9. The original GAN paper [Goodfellow et al. NeurIPS 2014] proposed a modification to the minimax loss below:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

What is the modification and what problem was it designed to mitigate?

Mitigate problem of Vanishing gradient

$$\textbf{Minimax} \quad J^{(G)}(G) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} \log[1 - D(G(\boldsymbol{z}))].$$

$$\textbf{Non-saturating} \quad J^{(G)}(G) = -\mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} \log D(G(\boldsymbol{z})).$$

Modify the objective for generator

Minimax loss: extremely small gradient when discriminator is confident the data is fake, which is bad bc generator need gradient to learn

Non-saturating loss: large gradient in this case

10. What is the "W" in a WGAN? What are some differences between a regular GAN and a W GAN?

W: Wasserstein distance
Use W distance as objective instead of Jensen-Shannon Divergence in regular GAN

$$L(p_r, p_g) = W(p_r, p_g) = \max_{w\in W} \mathbb{E}_{x\sim p_r}[f_w(x)] - \mathbb{E}_{z\sim p_r(z)}[f_w(g_\theta(z))]$$

No sigmoid on the out layer, use Lipschitz continuous, has weight clipping
Discriminator gives a value of a witness function f, low value for fake data and high value for real data, which is useful to deal with degree of difference between generated data and training data
instead of giving a decision "1" for real data and "0" for fake data no matter how the fake data is different from real data in regular GAN

Ad: improve vanishing gradient, more gradient signal for generator to improve

Long answers (10 points each):

1. What is the main advantage of VAEs over regular AEs for data generation?

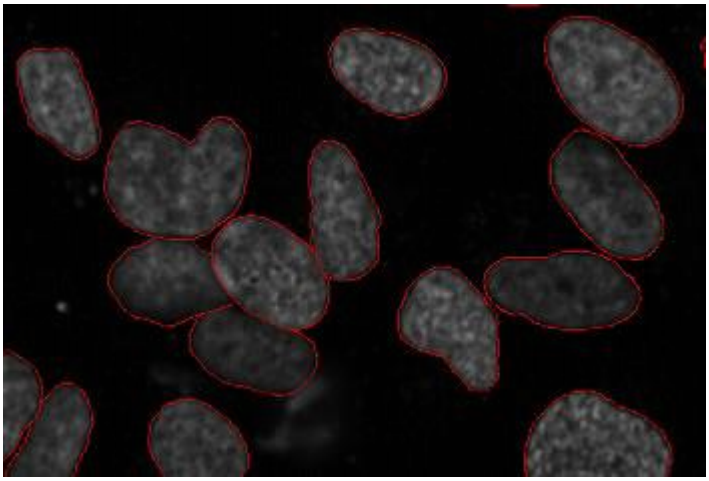
- AE can't generate unseen data
- VAE: embedding layer learns a series of mean and variance for each point,
    - KL-divergence from a prior N(0, 1) makes the space contiguous for sampling and interpolation from latent layer

Can you think of other regularizations or penalties (other than the VAE regularizations) to achievea similar effect in an AE?
Differentiable W distance
Interpolation enforced using penalty that goes between data points

2. Suppose you have pictures of cells as below but without the segmentations (given in red). Propose a way to train a neural network to segment without labels. You may augment the data or modify it in some way if you would like.

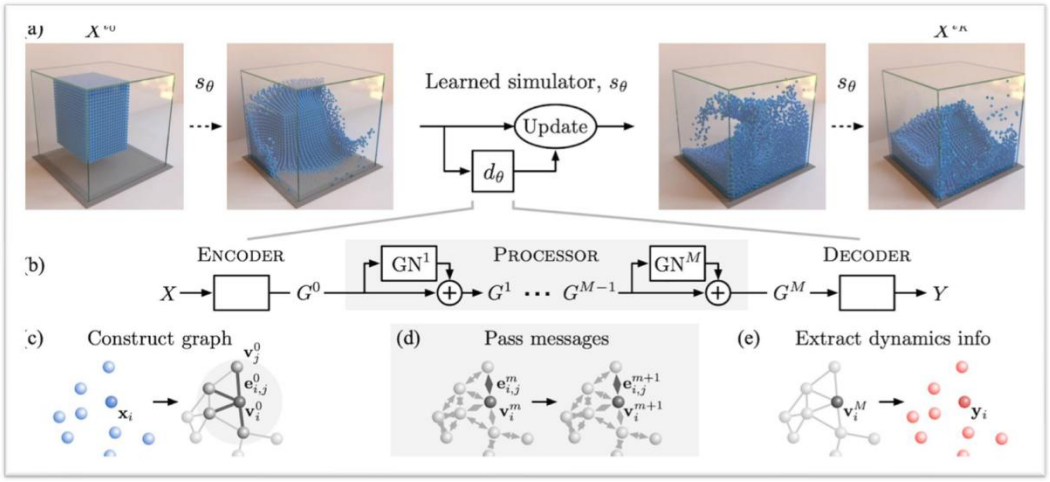
U-Net pixel-wise segmentation
Given Background is Black, cell is grey

Path: give patch for background and patch for cell, then calculate Patch-wise loss
Edge detection layer



3. Design a neural network to perform an n-body simulation (a dynamic system of particles under the influence of physical forces such as gravity). Assume you have a multitude of time traces of such a simulation given as training data. What kind of neural network would you use? What would its input and output be? How would you train it?

   Neural ODE (good if no regular time samples), sequential network



- Graph network simulator (GNS)
- Architecture: Encoder-processor-decoder
- Input: one trace node features (position, velocities, particle type), edge features (displacement)
- Output: next trace predicted node features
- Loss: L2 loss over all the particles time pair

- Training: one-step minibatch training (t_i - t_{i+1} pair), add Gaussian noise to make the GNN robust to prevent error accumulation at evaluation
- Test: 1000 steps (t_0-> t_{1000})

- Encoder: create a neighborhood graph G^0
- Processor: several message-passing layers
  - concatenate node and edge features
  - compute message function through MLP
  - output embeddings for each particle
  - predict next step dynamics G^1, G^2, …, G^M
- decoder: feed node embeddings into Euler integrator to obtain node features

Bonus

Deep Graph Neural Networks often criticized for “oversmoothing”, i.e., after K layers of mean or sum aggregation the features are averages-of-averages-of-averages…

Propose a way to combat this. In other words, describe an aggregation or design decision that allows for long range propagation of signals without oversmoothing.

Skip connection,
mix-hop: difference between scales of aggregation
complex aggregation function (LSTM, small NN)

Describe a situation in which this could be useful.
Graph classification, graph generation
Global property e.g., molecule simulation