Name:

NetID:

1.) What are some of the benefits of self-attention when compared to RNNs?

RNNs have to run their loop for every token in the input sequence leading to: forgetting, slow speeds, vanishing gradients unlike self-attn. which is pairwise

2.) What is a filter used for in a Convolutional Neural Network?

Filters are applied to the image as a small grid of "pixels" to help detect patterns in an image

3.) What are over and underfitting?

Underfitting - Model is too simple to capture distribution/patterns
Overfitting - Model is too complex and fits to "noise"/outliers

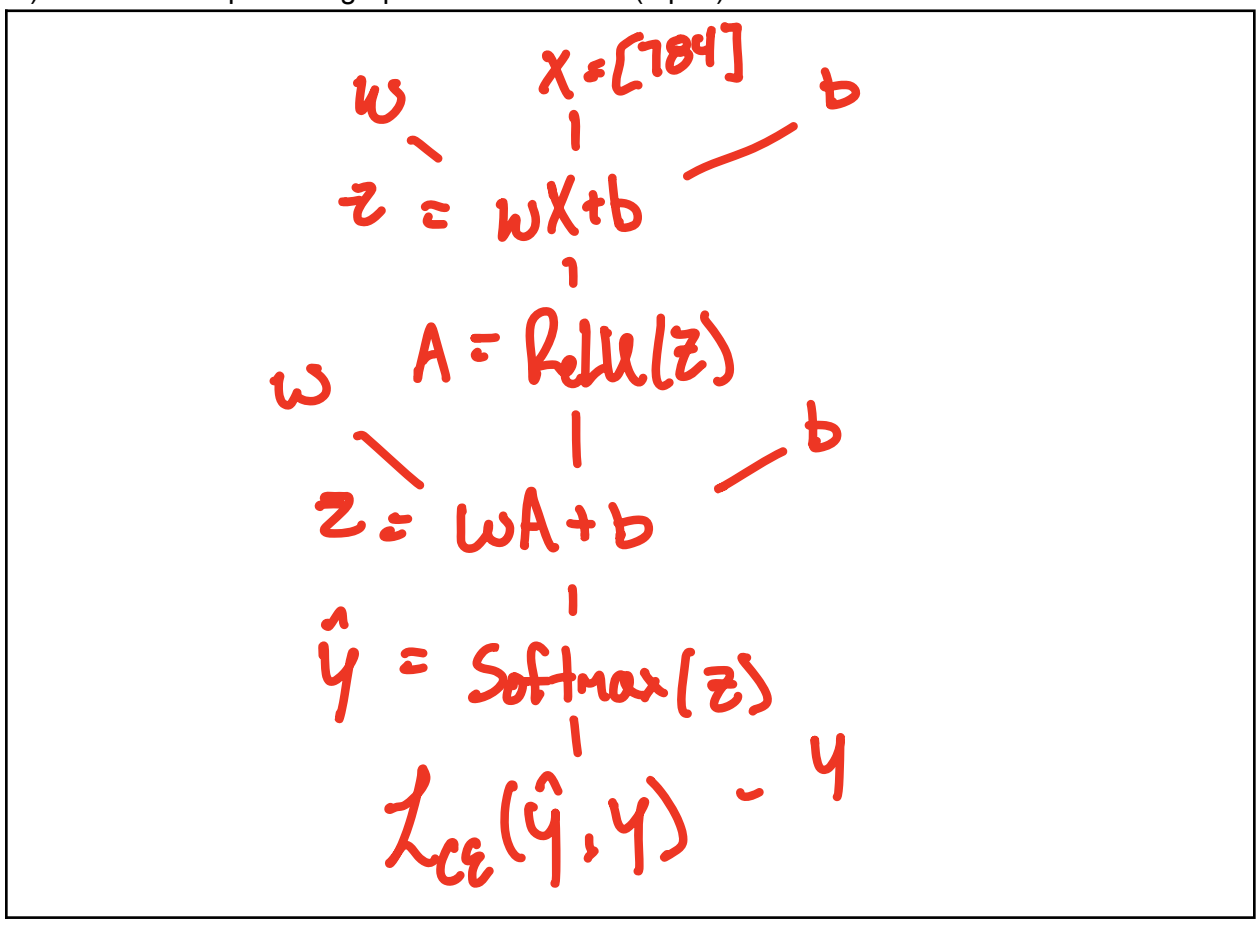
4.) Given the pytorch code below, answer the following questions.

```python
import torch
import torch.nn as nn
class Model4_1(nn.Module):
    def __init__(self):
        super(Model4_1, self).__init__()
        self.lin1 = nn.Linear(784, 100)
        self.relu = nn.ReLU()
        self.lin2 = nn.Linear(100, 10)

    def forward(self, x):
        out = self.lin1(x)
        out = self.relu(out)
        out = self.lin2(out)
        return out
```

Note: You can assume softmax and cross entropy loss are used after the final layer, the two yielding a combined derivative of: ŷ - y

a.) Draw the computation graph for the network. (5 pts.)



b.) Give the derivative chain for calculating the gradient of the bias in the first layer. (2 pts.)

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial A} \frac{\partial A}{\partial z} \frac{\partial z}{\partial b}$$

c.) Give the equation you'd use to actually calculate the gradients. (5 pts.)

$$\frac{\partial \mathcal{L}}{\partial b} = (\hat{y} - y)(w) \begin{pmatrix} x > 0 : 1 \\ x \leq 0 : 0 \end{pmatrix} (1)$$

d.) Given the gradients, what's the next step we should take?

Update the bias term (every parameter really)

5.) Why do we want activation functions in neural networks?

These are what allow us to capture non-linear relationships in the data

6.) How could you use a kNN method for a regression task?

Find the k nearest points and average their values

7.) What's an example of a feature which would cause us to want to use a non-Gaussian distribution in a Naive Bayes classifier?

Net worth isn't normally distributed

Dice rolls

Many possible answers

8.) What is a feature vector, why are these useful?

Feature vectors are a high dim representation of our data and allow us to compare samples to eachother

9.) I have a bunch of anonymized student exam grades, which model should I use to build a classifier that would allow me to predict which student got which grade?

Maybe a Naive Bayes Classifier?

Many possible answers as long as explanation was reasonable

10.) A new tech company in Norway is training their faceID tech on participants in yoga classes. They're about to roll it out globally, do you foresee any issues with this?

Yoga people in norway likely aren't representative of a global population and the model is probably overfit

11.) When we say transformers are autoregressive what are we saying?

When we generate new data with a transformer we append the new token to the input and re-run it to predict the next token

12.) When we get a sentence for an NLP task there are two steps before we can pass it into our model, what are these and what do they “look” like?

tokenize: map each token to a number

embed: map our tokens to a vector

13.) Given the training code for a pytorch model below, answer the following questions:

```
# Instantiate the model, define loss and optimizer
model = SimpleNet()
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=1.00)

# Training loop
num_epochs = 1
train_losses = []

for epoch in range(num_epochs):
    model.train()
    optimizer.zero_grad()

    # Forward pass
    outputs = model(X_train)
    loss = criterion(outputs, y_train)

    # Backward pass and optimization
    loss.backward()
    optimizer.step()
```

a.) My model is really bad, what are some things I could change that may make it better?

More than 1 epoch

Learning Rate of 1 is really high

b.) What are the names for what the lines loss.backward() and optimizer.step do?

loss.backward → backprop

optimizer.step → gradient descent

14.) On Homework 0910, what purpose did <PAD> serve?

N/A

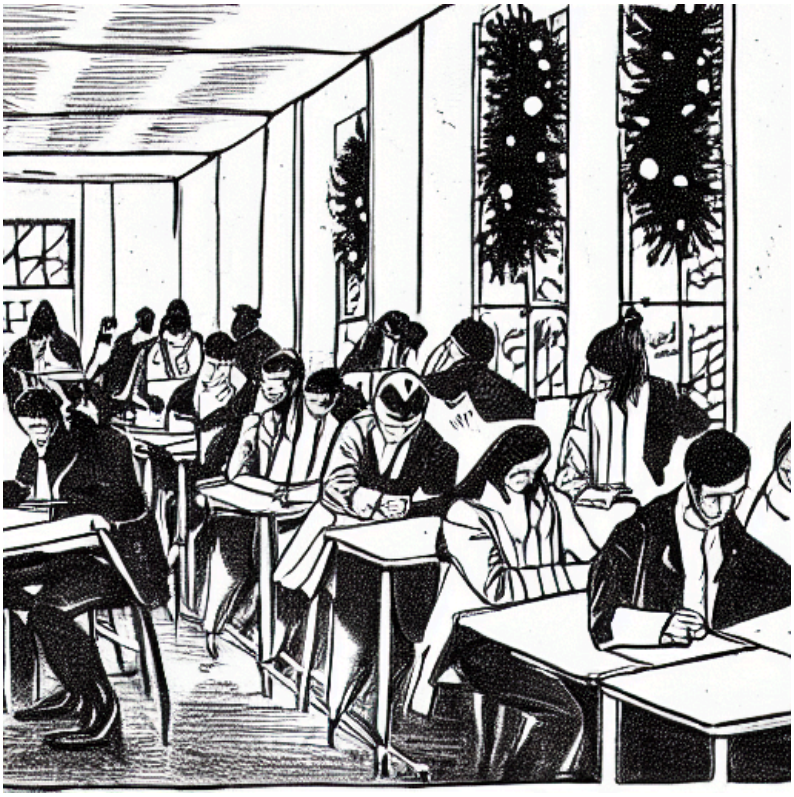
(Because inputs were standard len.)

15.) On Homework 0910, Why did we pass the output of an LSTM into a fully connected layer?

N/A

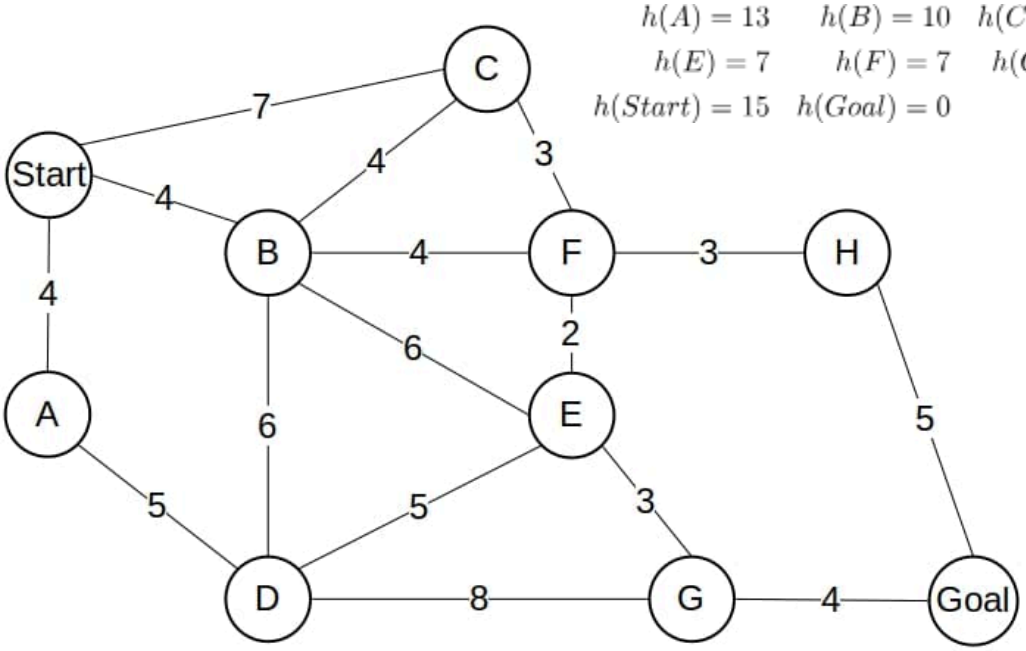
(predict our output)

Break.) Take a breath and rest a little. Enjoy this image of “black and white line drawing of students acing their final exam beneath a christmas tree” I generated using the simple stable diffusion model we looked at in class. Almost there, I believe in you, you got this!



16.) Given the following graph, starting in Node S and trying to get to Node G, trace the steps the A* Algorithm would take. To make the graph less busy, the heuristic distances are given as a table. Much like on Exam 01, my solution uses all 11 spots.

$h(A) = 13$ $h(B) = 10$ $h(C) = 10$ $h(D) = 9$
$h(E) = 7$ $h(F) = 7$ $h(G) = 2$ $h(H) = 3$
$h(Start) = 15$ $h(Goal) = 0$



*.) N -> N' = d(N, N') + h(N', G)

1.) S -> A = 4 + 13 = 17

~~2.) S -> B = 4 + 10 = 14~~

3.) S -> C = 7 + 10 = 17

4.) S -> B→F: 8+7=15

5.) S -> B→D: 10+9=19

6.) S -> B→E: 10+7=17

7.) S -> B→C: 8+10=18

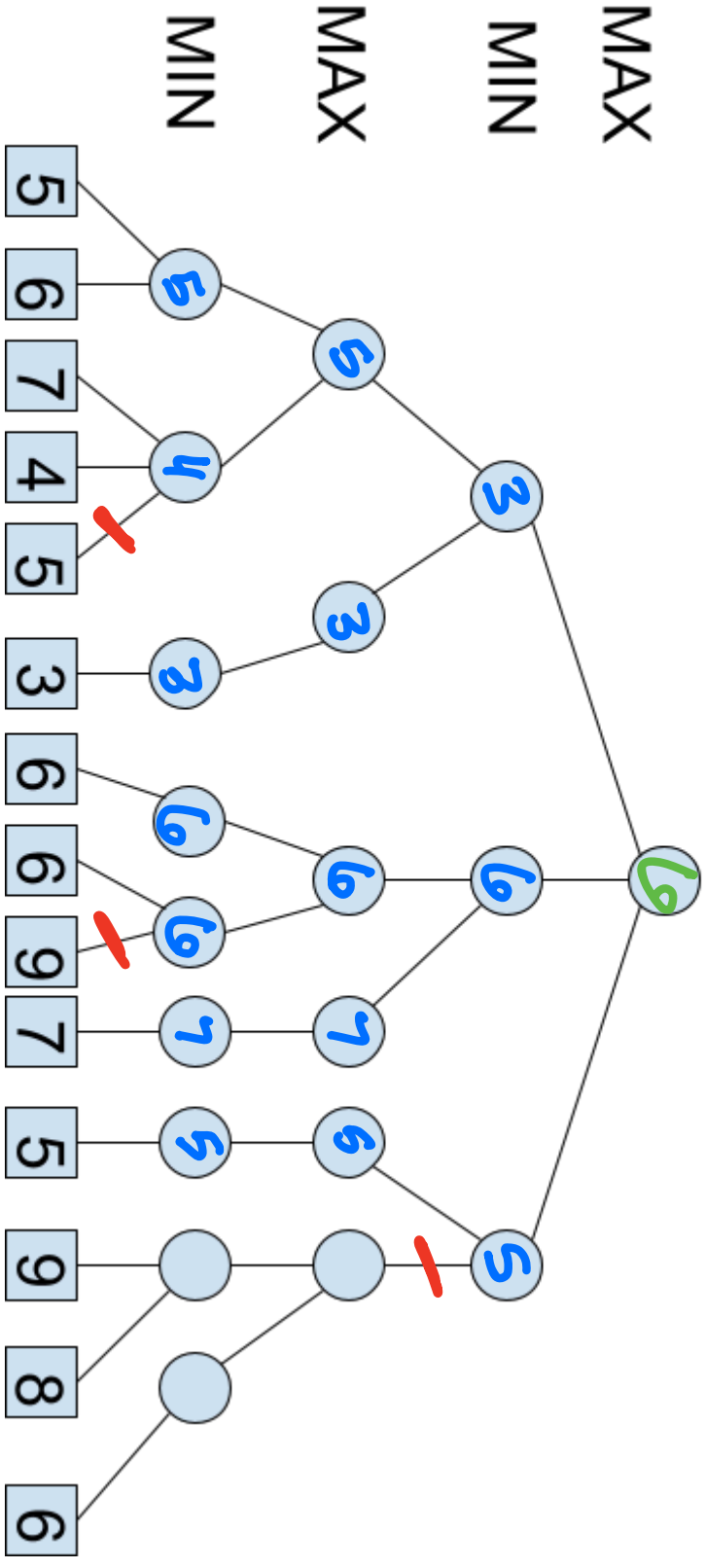
~~8.) S -> B→F→H: 11+3=14~~

9.) S -> B→F→E: 10+7=17

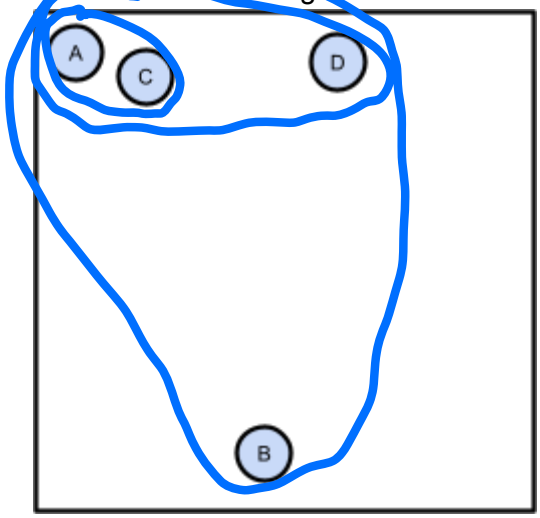
10.) S -> B→F→C: 11+10=21

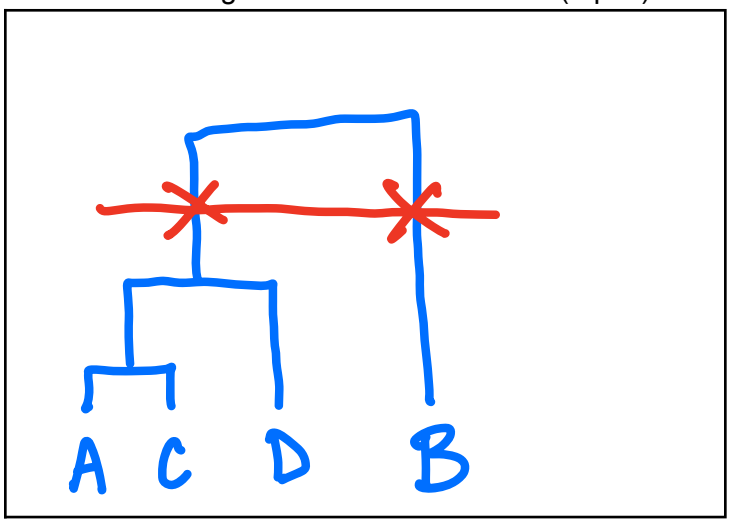
11.) S -> B→F→H→G: 16+0=16

17.) Use Alpha/Beta Pruning on the tree below, showing the Alpha and Beta at each node and indicate which branches don't need to be explored. (8 pts.)



18.) Given the data points, draw the dendrogram that would be created using agglomerative hierarchical clustering and then draw a line on the dendrogram to create 2 clusters. (3 pts.)





19.) Given the trivariate Jacobian matrix below you can compute the gradients for cost function J. Given the values 1, 3, 2 for the 3 variables nah I'm just kidding have a Merry Christmas and if you write a message for me to pass on to next semester's class I'll give 2 bonus points. Thanks for the semester! Ignore the next sentence. Solve the Jacobian for the gradients.

$$\frac{\partial(x,y,z)}{\partial(u,v,w)} = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} & \frac{\partial y}{\partial w} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} & \frac{\partial z}{\partial w} \end{vmatrix}$$

$$= \frac{\partial x}{\partial u}\begin{vmatrix} \frac{\partial y}{\partial v} & \frac{\partial y}{\partial w} \\ \frac{\partial z}{\partial v} & \frac{\partial z}{\partial w} \end{vmatrix} - \frac{\partial x}{\partial v}\begin{vmatrix} \frac{\partial y}{\partial u} & \frac{\partial y}{\partial w} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial w} \end{vmatrix} + \frac{\partial x}{\partial w}\begin{vmatrix} \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{vmatrix}$$

$$= \frac{\partial x}{\partial u}\left(\frac{\partial y}{\partial v}\cdot\frac{\partial z}{\partial w} - \frac{\partial y}{\partial w}\cdot\frac{\partial z}{\partial v}\right) - \frac{\partial x}{\partial v}\left(\frac{\partial y}{\partial u}\cdot\frac{\partial z}{\partial w} - \frac{\partial y}{\partial w}\cdot\frac{\partial z}{\partial u}\right) + \frac{\partial x}{\partial w}\left(\frac{\partial y}{\partial u}\cdot\frac{\partial z}{\partial v} - \frac{\partial y}{\partial v}\cdot\frac{\partial z}{\partial u}\right)$$