CS 182/282A "Deep Learning"
Spring 2022 Marvin Zhang

Midterm

- Do not open the exam before you are instructed to do so.
- **Electronic devices should be turned off for the entire exam duration**, including cell phones, tablets, headphones, and laptops. Turn your cell phone off, or risk getting a zero on the exam.
- The exam is closed book, closed notes except your one-page cheat sheet. You are allowed one double-sided 8.5x11 inch cheatsheet.
- You have 1 hour and 50 minutes (unless you are in the DSP program and have an allowance of 150% or 200% time).
- Please write your initials at the top right of each page after this one (e.g., write "JD" if you are John Doe). Finish this by the end of your 1 hour and 50 minutes.
- Mark your answers on the exam itself in the space provided. Do **not** attach any extra sheets.
- For the questions in Section 2, you should mark true or false for each statement. For each question, there may be more than one true option, but there is always at least one true option. Note that there is a penalty for marking the incorrect option, but there is no penalty for not marking either option.

First name	
Last name	
SID	
First and last name of student to your left	
First and last name of student to your right	

○ CS 182

○ CS 282A

1 Multiple Choice (Single Answer)

1. (1 point) Consider the vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\mathbf{x}) = \frac{n\mathbf{x}}{\sum_{i=1}^n x_i}$$

where $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$. This is a transformation that normalizes the input vector $\mathbf{x}$ to have mean 1. What is the Jacobian $\frac{df}{d\mathbf{x}}$ of this function? Let $\mathbf{I}_n$ denote the identity matrix in $\mathbb{R}^{n \times n}$ and $\mathbf{1}_n$ denote the vector of all ones in $\mathbb{R}^n$. Use the Jacobian convention where $\left(\frac{df}{d\mathbf{x}}\right)_{ij} = \frac{\partial f(\mathbf{x})_i}{\partial x_j}$.

○ $\frac{n}{\sum_{i=1}^n x_i}(\mathbf{I}_n * (\sum_{i=1}^n x_i) - \mathbf{x}\mathbf{x}^\top)$

○ $\frac{n}{\sum_{i=1}^n x_i}(\mathbf{I}_n * (\sum_{i=1}^n x_i)^2 - \mathbf{x}\mathbf{1}_n^\top)$

○ $\frac{n}{(\sum_{i=1}^n x_i)^2}(\mathbf{I}_n * (\sum_{i=1}^n x_i) - \mathbf{x}\mathbf{x}^\top)$

○ $\frac{n}{(\sum_{i=1}^n x_i)^2}(\mathbf{I}_n * (\sum_{i=1}^n x_i) - \mathbf{x}\mathbf{1}_n^\top)$

Solution: D

2. (1 point) Suppose we have a one hidden layer ReLU network $f(\mathbf{x}) = \mathbf{w}^\top \max(\mathbf{A}\mathbf{x}, 0)$ with input $\mathbf{x} \in \mathbb{R}^d$, first weight matrix $\mathbf{A} \in \mathbb{R}^{k \times d}$, and second weight vector $\mathbf{w} \in \mathbb{R}^k$. What is the gradient of $f(\mathbf{x})$ with respect to the input $\mathbf{x}$?

○ $\max(\mathbf{A}^\top\mathbf{w}, 0)$

○ $\max(\mathbf{A}^\top, 0)$

○ $\mathbf{A}^\top\mathbf{Z}\mathbf{w}$ where $\mathbf{Z} \in \mathbb{R}^{k \times k}$ is a diagonal matrix such that $(\mathbf{Z})_{ii}$ is equal to 1 if $(\mathbf{A}\mathbf{x})_i > 0$ and 0 if $(\mathbf{A}\mathbf{x})_i \leq 0$.

○ $\mathbf{A}^\top\mathbf{w}$

○ $\mathbf{A}^\top\mathbf{Z}\mathbf{w}$ where $\mathbf{Z} \in \mathbb{R}^{k \times k}$ is a diagonal matrix such that $(\mathbf{Z})_{ii}$ is equal to 1 if $(\mathbf{x})_i > 0$ and 0 if $(\mathbf{x})_i \leq 0$.

Solution: C

3. (1 point) Assume we have a classifier with K classes. Let the probability of the i-th class be $p_i = \text{softmax}(\mathbf{z})_i$, where $\mathbf{z} = [z_1, \dots, z_K]$ is the logits vector. We treat p as a distribution over $\{1, \dots, K\}$; let U denote the uniform distribution over $\{1, \dots, K\}$. Also, recall that cross entropy between two distributions p and q on a random variable X is defined as

$$H(p; q) = -\sum_x p(x) \log q(x).$$

Consider the loss function given by the expression

$$-\frac{1}{K}\left[\sum_{i=1}^{K} z_i\right] + \log \sum_{i=1}^{K} \exp(z_i)\,.$$

Which of the expressions below is equal to this loss?

○ $H(U; p)$

○ $H(p; U)$

○ $H(p; p)$ (the entropy of the softmax distribution)

○ None of the above

Solution: A

4. (1 point) Which of the following loss functions is convex in a deep neural network's parameters?

○ Mean squared error loss

○ Binary cross-entropy loss

○ Softmax cross-entropy loss

○ None of the above

Solution: D

5. (1 point) You notice that your network has converged to a high training loss. Which of the following options is most likely to help you address this issue?

○ Add in completely new datapoints to the training set.

○ Use data augmentation strategies to synthetically generate more varied training data.

○ Increase the number of layers in your network.

○ Train an ensemble of models and make predictions based on an average of the models.

Solution: C

6. (1 point) Consider this hypothetical optimizer:

$$
\begin{aligned}
&n \leftarrow n + 1 \text{ \# } (n \text{ initialized at } 0) \\
&\mathbf{g} \leftarrow \frac{1}{n} \nabla \theta \sum_{i=1}^{N} \ell(\theta; \mathbf{x}_i, y_i) + \frac{n-1}{n} \mathbf{g} \\
&\theta \leftarrow \theta - \alpha \mathbf{g}
\end{aligned}
$$

How does this optimizer differ from SGD with momentum?

- ○ This optimizer gives less weight to “important” gradient dimensions, which could slow time until convergence.
- ○ This optimizer’s running average gives more weight to older gradients, which could slow or prevent convergence.
- ○ The magnitude of the optimizer’s updates is guaranteed to converge to zero more rapidly.

Solution: B

7. (1 point) Given maximum steps N and initial learning α_{initial}, which of the following correctly fills in the blanks of $\alpha_i = \alpha_{\text{initial}} \cdot \square \cdot \left[\square + \square\left(\square \cdot \frac{i}{N}\right)\right]$ to make it cosine learning rate decay? Recall that $\cos(0) = \cos(2\pi) = 1$ and $\cos(\pi) = -1$.

- ○ $\frac{1}{2}$, 1, cos, π.
- ○ 1, 2, cos, π
- ○ 1, 2, cos, 2π
- ○ $\frac{1}{2}$, 1, sin, 2π

Solution: A

8. (1 point) What is the distinction between primitive and composite functions in automatic differentiation?

   ○ Composite functions evaluate both the primitive function value and its gradient.

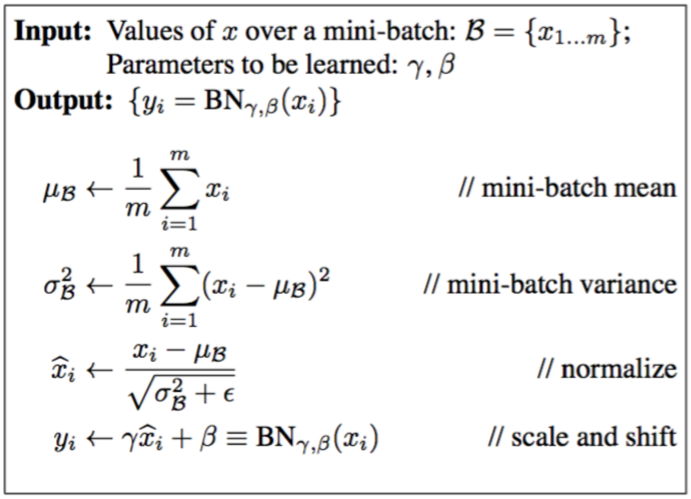
   ○ Composite functions require the user to specify the gradient computation.

   ○ Differentiating primitive functions only involves a lookup table, whereas differentiating composite functions requires invoking the chain rule.

   ○ Forward mode AD handles differentiating primitive functions whereas reverse mode AD handles differentiating composite functions.

   Solution: C

9. (1 point) Consider a batch normalization layer with scalar input x and scalar output y. Which of the following are the correct derivatives for the scale and shift parameters γ and β?



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1 \ldots m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

   ○ $\frac{\partial f}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i}$ and $\frac{\partial f}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \hat{x}_i$

   ○ $\frac{\partial f}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i}$ and $\frac{\partial f}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot x_i$

   ○ $\frac{\partial f}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial f}{\partial x_i}$ and $\frac{\partial f}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot x_i$

   ○ $\frac{\partial f}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \hat{x}_i$ and $\frac{\partial f}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \cdot \hat{x}_i$

   Solution: A

10. (1 point) Consider a neural network with L hidden layers followed by an output layer. Suppose for simplicity that the dimensions of the input, output, and hidden activations are all d. Further disregard the effects of any nonlinearities. If the input is standardized and all of the weight matrices are initialized as $\mathbf{W}_{ij}^{(l)} \sim \mathcal{N}(0, 1)$, which of the following choices best characterizes the variance of the model output?

   ○ 1

   ○ $d(L+1)$

   ○ $d^2(L+1)$

   ○ d^{L+1}

   ○ d^{2L+1}

   Solution: D

11. (1 point) Which of the following statements is true about neural network ensembling?

   ○ Compared to training a single neural network with a particular architecture, training an ensemble of randomly initialized networks all with that architecture will typically lead to lower variance at the cost of increasing bias.

   ○ Compared to training a single neural network with a particular architecture, training an ensemble of randomly initialized networks all with that architecture will typically lead to lower bias at the cost of increasing variance.

   ○ Neural network ensembles are typically trained with bagging, i.e., bootstrap resampling of the dataset to create different training sets.

   ○ Empirically, neural network ensembles are effective at estimating the uncertainty of their predictions, i.e., quantifying when they are likely to make a mistake.

   Solution: D

12. (1 point) Consider a convolution layer with input size $256 \times 256 \times 3$ and where the output has one channel. If the convolution layer contains a single 1×1 convolution filter, how many parameters are there including bias?

   ○ 65536

   ○ 256

   ○ 3

   ○ 4

   Solution: D

13. (1 point) You have a dataset consisting of assorted house attributes (market price, square footage, number of rooms, etc.) You wish to train a model to predict market price from the other attributes. Should you use a fully connected network (FCN) or a convolutional network (CNN) with 1-D convolutions, and why?

    ○ Use a CNN, since CNNs typically achieve better performance than FCNs.

    ○ Use a FCN, since CNNs should be reserved for data with spatially local patterns.

    ○ Use a FCN, since CNNs can only be used for images.

    Solution: B

14. (1 point) What are the 3R's of computer vision?

    ○ Rest, relaxation, and recovery

    ○ Recognition, reconstruction, and reorganization

    ○ Region refinement through representation

    ○ Reinforcement, regularization, and randomization

    Solution: B

15. (1 point) What is a bounding box in the context of computer vision?

    ○ A rectangle which represents the detection of an object in an image.

    ○ A rectangle which represents a crop of an image for data augmentation.

    ○ A rectangle which represents an area of similar pixels between two images.

    ○ A rectangle which represents an area of an image to be removed for data cleaning purposes.

    Solution: A

2 Multiple Choice (True False)

Fill in either true or false for all statements: there may be more than one true option, but there is always at least one true option. Each question is worth two points, and each statement is worth 0.5 points. Marking the incorrect option will result in a penalty of -0.5 points, whereas not choosing an option does not have a penalty. Your score for each question will be ReLU(points).

16. (2 points) When we optimize the parameters of a neural network, we are searching for the best function f that is consistent with a particular neural architecture. For example, suppose that our network takes an input $\mathbf{x} \in \mathbb{R}^{100}$, outputs a vector $f(\mathbf{x}) \in \mathbb{R}^{24}$, and has one hidden layer $\mathbf{z} \in \mathbb{R}^{40}$ followed by a **tanh** activation. The set of all functions consistent with that architecture is

$$\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathbf{W}_2\mathbf{tanh}(W_1\mathbf{x}), \mathbf{x} \in \mathbb{R}^{100}, \mathbf{W}_1 \in \mathbb{R}^{40\times 100}, \mathbf{W}_2 \in \mathbb{R}^{24\times 40}\}.$$

We say that two network architectures α and β have the same *expressivity* if the set of functions $\mathcal{F}_\alpha$ and $\mathcal{F}_\beta$ which are consistent with each architecture are equivalent, i.e. $\mathcal{F}_\alpha = \mathcal{F}_\beta$. Mark true for the **two** architectures below which have the same expressivity (and false for the other two architectures).

○ T ○ F : Two linear layers, **RELU**, two linear layers.

$$\mathcal{F}_a = \{f \mid f(\mathbf{x}) = \mathbf{W}_4\mathbf{W}_3\mathbf{RELU}(\mathbf{W}_2\mathbf{W}_1\mathbf{x}), \\ \mathbf{x} \in \mathbb{R}^{100}, \mathbf{W}_1 \in \mathbb{R}^{60\times 100}, \mathbf{W}_2 \in \mathbb{R}^{40\times 60}, \mathbf{W}_3 \in \mathbb{R}^{32\times 40}, \mathbf{W}_4 \in \mathbb{R}^{24\times 32}\}$$

○ T ○ F : Two linear layers, **tanh**, one linear layer, **RELU**

$$\mathcal{F}_b = \{f \mid f(\mathbf{x}) = \mathbf{RELU}(\mathbf{W}_3\mathbf{tanh}(\mathbf{W}_2\mathbf{W}_1\mathbf{x})), \\ \mathbf{x} \in \mathbb{R}^{100}, \mathbf{W}_1 \in \mathbb{R}^{60\times 100}, \mathbf{W}_2 \in \mathbb{R}^{40\times 60}, \mathbf{W}_3 \in \mathbb{R}^{24\times 40}\}$$

○ T ○ F : One linear layer, **RELU**, two linear layers

$$\mathcal{F}_c = \{f \mid f(\mathbf{x}) = \mathbf{W}_3\mathbf{W}_2\mathbf{RELU}(\mathbf{W}_1\mathbf{x}), \\ \mathbf{x} \in \mathbb{R}^{100}, \mathbf{W}_1 \in \mathbb{R}^{40\times 100}, \mathbf{W}_2 \in \mathbb{R}^{39\times 40}, \mathbf{W}_3 \in \mathbb{R}^{24\times 39}\}$$

○ T ○ F : Two linear layers, **RELU**, one linear layer, **RELU**

$$\mathcal{F}_d = \{f \mid f(\mathbf{x}) = \mathbf{RELU}(\mathbf{W}_3\mathbf{RELU}(\mathbf{W}_2\mathbf{W}_1\mathbf{x})), \\ \mathbf{x} \in \mathbb{R}^{100}, \mathbf{W}_1 \in \mathbb{R}^{60\times 100}, \mathbf{W}_2 \in \mathbb{R}^{40\times 60}, \mathbf{W}_3 \in \mathbb{R}^{24\times 40}\}$$

Solution: A and C are equivalent.

17. (2 points) Consider the function $f(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1$. Mark true or false for the following statements regarding this function.

   ○ T ○ F : This function can be used as the loss function for regression, where **x** is the prediction of our model, and **y** is the regression target.

   ○ T ○ F : This function is convex with respect to both of its inputs, so if we use it as our loss function for neural network training, where the output of our network is **x** and the target is **y**, our optimization problem will be convex in all network parameters.

   ○ T ○ F : This function encourages sparsity in the learned network parameters when used as the loss function for regression as described above.

   ○ T ○ F : This function is not differentiable at $\mathbf{x} = \mathbf{y}$.

   Solution: A and D are true.

18. (2 points) Suppose you have a loss function ℓ that computes the squared error between a classifier's prediction $\hat{y}$ and the true label y. Mark true for the alternate loss functions below which would produce the same gradient update as ℓ (and false for the loss functions which would not).

   ○ T ○ F : $\ell(\hat{y}, y) + 10$

   ○ T ○ F : $10 \times \ell(\hat{y}, y)$

   ○ T ○ F : $\ell(10\hat{y}, 10y)$

   ○ T ○ F : 0 if $\ell(\hat{y}, y) < 0.5$ else 1

   Solution: A is true.

19. (2 points) Supposing the same squared error loss function ℓ as above, mark true for the alternate loss functions below (which are the same as the choices above) which could be optimized to achieve good performance, assuming you select an appropriate learning rate and neural net architecture (and false for the loss functions which could not).

   ○ T ○ F : $\ell(\hat{y}, y) + 10$

   ○ T ○ F : $10 \times \ell(\hat{y}, y)$

   ○ T ○ F : $\ell(10\hat{y}, 10y)$

   ○ T ○ F : 0 if $\ell(\hat{y}, y) < 0.5$ else 1

   Solution: A, B, and C are true.

20. (2 points) Mark true or false for the following statements about ℓ_2 regularization.

○ T ○ F : For stochastic gradient descent, it is the same as weight decay.

○ T ○ F : For Adam, it is the same as weight decay.

○ T ○ F : For stochastic gradient descent, we add $\lambda||\theta||_2^2$ to the loss, where λ is a hyperparameter.

○ T ○ F : For Adam, we subtract $\lambda||\theta||_2^2$ from the loss.

Solution: A and C are true.

21. (2 points) You are training a neural network using SGD, and see your loss decreases very slowly per training iteration. Furthermore you see that your training accuracy has not reached the desired value. Mark true for the options below which have a reasonable chance of resolving this issue (and false for the options which do not).

○ T ○ F : Increase the learning rate

○ T ○ F : Increase the batch size of SGD

○ T ○ F : Use SGD with momentum

○ T ○ F : Increase the weight of ℓ_2 regularization on the network parameters

Solution: A, B, C are true.

22. (2 points) Mark true for the statements below that accurately characterize why we prefer to use automatic differentiation over numerical differentiation in deep learning (and false for the statements which do not).

○ T ○ F : Automatic differentiation allows us to compute derivatives accurately (to computer precision).

○ T ○ F : Numerical differentiation is more challenging to implement than automatic differentiation.

○ T ○ F : Backpropagating using numerical differentiation would incur greater compounding derivative approximation errors in the number of layers.

○ T ○ F : Automatic differentiation allows for more time efficient gradient computation than numerical differentiation.

Solution: A, C, and D are true.

23. (2 points) Mark true or false for the following statements about reverse mode automatic differentiation.

○ T ○ F : It is preferred over forward mode automatic differentiation when the input dimension is smaller than the output dimension.

○ T ○ F : It is preferred over forward mode automatic differentiation when the output dimension is smaller than the input dimension.

○ T ○ F : It computes each intermediate operation's outputs and their derivatives simultaneously.

○ T ○ F : It requires storing intermediate variables corresponding to computations of intermediate operations.

Solution: B and D are true.

24. (2 points) Consider a network $f(\mathbf{x}) = \text{ReLU}(\mathbf{Wx} + \mathbf{b})$ with one nonlinear layer and no final linear layer. Suppose you independently initialize all $\mathbf{W}_{ij} \sim \mathcal{N}(0, \sigma_{\mathbf{W}}^2)$ and all $\mathbf{b}_i \sim \mathcal{N}(0, \sigma_{\mathbf{b}}^2)$. Further suppose that $\sigma_{\mathbf{W}}$ and $\sigma_{\mathbf{b}}^2$ are chosen such that, given unit Gaussian input, the output variance is 1. Now you want to replace the ReLU with a leaky ReLU with leaky rate 0.2. Remember that the definition for leaky ReLU with leaky rate 0.2 is:

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0.2x, & \text{otherwise} \end{cases}$$

Mark true for the following options that may allow us to maintain an output variance of 1 (and false for the options which will not).

○ T ○ F : Increase $\sigma_{\mathbf{W}}$ and increase $\sigma_{\mathbf{b}}$

○ T ○ F : Keep $\sigma_{\mathbf{W}}$ the same and increase $\sigma_{\mathbf{b}}$

○ T ○ F : Keep $\sigma_{\mathbf{W}}$ the same and decrease $\sigma_{\mathbf{b}}$

○ T ○ F : Decrease $\sigma_{\mathbf{W}}$ and decrease $\sigma_{\mathbf{b}}$

Solution: C and D are true.

25. (2 points) Mark true or false for the following statements about batch normalization and layer normalization.

○ T ○ F : After performing layer normalization on activations $\mathbf{z}^{(l)}$, it is possible to have some activation dimensions $\mathbf{z}_i^{(l)}$ that are do not have zero mean and unit variance.

○ T ○ F : Batch normalization is preferable to layer normalization in training situations where available memory constrains the batch size B to be very small.

○ T ○ F : Layer normalization is preferable to batch normalization in testing situations where only one test point is observed at a time.

○ T ○ F : Both batch normalization and layer normalization typically allow for larger learning rates and reduce the need for careful regularization.

Solution: A and D are true.

26. (2 points) Consider a fully connected neural network that outputs a vector $\mathbf{y}$ for each input vector $\mathbf{x}$. Suppose you have a batch of 3 input examples, $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, and the corresponding outputs from your network
$\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$. Mark true or false for the following statements about batch normalization and layer normalization during training.

○ T ○ F : If the network contains only layer normalization, changing the input $\mathbf{x}_1$ would only change the output $\mathbf{y}_1$, but not $\mathbf{y}_2$ and $\mathbf{y}_3$.

○ T ○ F : If the network contains only batch normalization, changing the input $\mathbf{x}_1$ would only change the output $\mathbf{y}_1$, but not $\mathbf{y}_2$ and $\mathbf{y}_3$.

○ T ○ F : If the network contains only batch normalization, the Jacobian matrix $\frac{\partial \mathbf{y}_2}{\partial \mathbf{x}_1}$ is always the zero matrix.

○ T ○ F : If the network contains only layer normalization, the Jacobian matrix $\frac{\partial \mathbf{y}_2}{\partial \mathbf{x}_1}$ can be a nonzero matrix.

Solution: A is true.

27. (2 points) Mark true or false for the following statements about convolutional networks.

○ T ○ F : They are more expressive than fully connected neural networks, i.e., they can approximate functions that fully connected neural networks cannot.

○ T ○ F : Convolutional layers can help regularize neural networks through parameter sharing, i.e., reusing the same parameters to transform different parts of the preceding representation.

○ T ○ F : Convolutional layers can help regularize neural networks through parameter reduction, i.e., they can contain fewer trainable parameters compared to a fully-connected linear layer that produces an equal-sized representation.

○ T ○ F : They are well-suited for image inputs because they incorporate inductive biases about how image data is usually structured.

Solution: B, C, and D are true.

28. (2 points) Mark true or false for the following statements about max pooling.

○ T ○ F : Adding max pooling to a network directly reduces the amount of parameters in the convolutional layers in the network.

○ T ○ F : Max pooling is a linear operation.

○ T ○ F : Max pooling increases the size of the receptive field (the input dimensions which can influence a particular activation) of each activation in the following layer.

○ T ○ F : Max pooling includes learnable parameters.

Solution: C is true.

29. (2 points) Which hyperparameter choices from the options below did the ConvNeXt designers find resulted in the largest accuracy gains on ImageNet? Mark true for the **two** most important choices (and false for the other two choices).

   ○ T ○ F : Modern training techniques (longer training, AdamW, data augmentation)
   ○ T ○ F : Depthwise convolutions + width expansion
   ○ T ○ F : Switching from ReLU to GELU nonlinearities
   ○ T ○ F : Switching from batch normalization to layer normalization

   Solution: A and B were the two most important.

3 Multiple Choice (Multiple Part)

30. (2 points) Suppose you are optimizing a model using a gradient descent algorithm in PyTorch. You start with the following lines:

```
x,y = batch
optimizer.zero_grad()
```

Consider the following four lines of code, labeled (1)-(4):

(1) `l.backward()`

(2) `optimizer.step()`

(3) `l = loss(output,y)`

(4) `output = model(x)`

Arrange these lines in the correct order following the first two lines provided. Use each line of code exactly once – you will not receive credit otherwise.

(a) (0.5 points) Line 1:

○ (1)
○ (2)
○ (3)
○ (4)

(b) (0.5 points) Line 2:

○ (1)
○ (2)
○ (3)
○ (4)

(c) (0.5 points) Line 3:

○ (1)
○ (2)
○ (3)
○ (4)

(d) (0.5 points) Line 4:

○ (1)
○ (2)
○ (3)
○ (4)

Solution: D, C, A, B

31. (2 points) You are training a sentiment classification model to predict the sentiment of movie reviews (binary classification). Someone has handed you a standard neural network. Assume the following:

- This neural network will do well on this task if the optimizer hyperparameters and dropout rate are chosen well.
- Your batch size is typical (e.g., 32) and that you have hundreds of thousands of clean training examples.
- You are using the Adam optimizer to train, and you are using dropout in your network.

Here is a fraction of the documentation for the PyTorch dropout function:
"During training, randomly zeroes some of the elements of the input tensor with probability p using samples from a Bernoulli distribution. Each channel will be zeroed out independently on every forward call."

Which hyperparameter settings are most reasonable out of the following options? Select only one answer for each hyperparameter.

```
torch.optim.Adam(params, lr=α, betas=(β₁, 0.999), eps=ε)
torch.nn.Dropout(p=p, inplace=False)
```

(a) (0.5 points) α

○ 3
○ 1
○ 3e-3
○ 1e-8

(b) (0.5 points) β_1

○ 0
○ 0.001
○ 0.9
○ 0.999
○ 1

(c) (0.5 points) ε

○ 3
○ 1
○ 3e-3
○ 1e-08

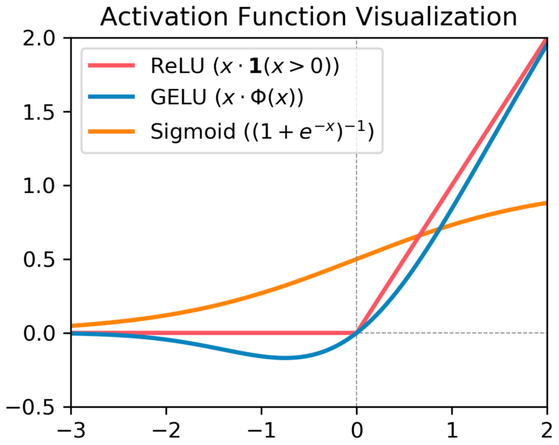
(d) (0.5 points) p

○ 0.2
○ 0.8
○ 0.9
○ 0.999
○ 1

Solution: C, C, D, A

32. (3 points) Let us analyze properties of three activation functions. (Note these properties do not decisively influence a deep network's performance.)

$\mathrm{ReLU}(x) = \max(0, x)$
$\mathrm{GELU}(x) = x\Phi(x)$, where $\Phi(x) = P(X \leq x)$ and $X \sim \mathcal{N}(0, 1)$
$\sigma(x) = (1 + e^{-x})^{-1}$



In this problem, we define convexity and monotonically nondecreasing as follows:

A function is **convex** if for all $0 < t < 1$ and for all $x_1, x_2 \in R$ such that $x_1 \neq x_2$, $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$.

A function is **monotonically nondecreasing** if for all $x \leq y$, $f(x) \leq f(y)$.

Mark true or false for all of the properties below for each activation function. The scoring rules are the same as the true false questions from Section 2, with the only difference being that each statement is worth only ±0.25 points rather than ±0.5 points.

(a) (1 point) ReLU

○ T ○ F : Convex

○ T ○ F : Monotonically nondecreasing

○ T ○ F : Invertible

○ T ○ F : At least one value for which the function has a zero gradient

(b) (1 point) GELU

○ T ○ F : Convex

○ T ○ F : Monotonically nondecreasing

○ T ○ F : Invertible

○ T ○ F : At least one value for which the function has a zero gradient

(c) (1 point) Sigmoid

○ T ○ F : Convex

○ T ○ F : Monotonically nondecreasing

○ T ○ F : Invertible

○ T ○ F : At least one value for which the function has a zero gradient

Solution: For ReLU: A, B, and D are true. For GELU: D is true. For sigmoid: B and C are true.