

# IE 8990: Midterm Exam

## Q1: True or False

Please indicate True or False for the following statements. If it is False, please explain the reason. [20 pts]

- (a) Activation function ReLU can lead to vanishing gradient problem.
  - a. FALSE, the ReLU activation function can lead to an exploding gradient.
- (b) After training a neural network for a binary classification problem, you observe a very low test accuracy (20%). This is due to overfitting.
  - a. TRUE, for binary classification the accuracy of an untrained neural network should be around 50%. Since the accuracy is below 50% the model must have an error in training.
- (c) Batch normalization is another way of performing dropout.
  - a. FALSE, these are separate processes that optimize training of a neural network.
- (d) Consider using a Generative Adversarial Network (GAN) to produce images of human face, the discriminator aims to classify images as face vs. non-face.
  - a. FALSE, the objective of the discriminator is predicting if the image is a real image.

## Q2

[10 pts] In Lecture note 10, we discussed the loss functions for classification problems. Categorical cross-entropy loss measures the performance of a multi-class classification model whose output is a probability value between 0 and 1, provided by formula below:

$$H(y, \hat{y}) = -\frac{1}{N} \sum_i^N \sum_j^M y_{ij} \log \hat{y}_{ij}$$

N is the number of samples, and M is the number of classes.  $\hat{y}_i$  is the output of softmax. Assume we have one sample with true label

$$y = (1, 0, 0),$$

$$\hat{y} = (0.1, 0.4, 0.5).$$

- Compute the cross-entropy  $H(y, \hat{y})$

TRUE LABEL	PREDICTED LABEL	COMBINED
1	$\log(.1)$	-1
0	$\log(.4)$	0
0	$\log(.5)$	0
Negative SUM:		1

- Is cross-entropy symmetric? i.e. if we reverse  $\hat{y}$  and  $y$ , will we get the same cross-entropy?
  - No, because of the binary nature of true labels, the log function changes those values to either 0 or undefined based on the binary class. This shows cross-entropy is not symmetric.

TRUE LABEL	PREDICTED LABEL	COMBINED
.1	$\text{Log}(1)$	0
.4	$\text{Log}(0)$	Undefined
.5	$\text{Log}(0)$	Undefined
	Negative SUM:	1

- Suppose we obtain a better estimate  $\hat{y}' = (0.6, 0.3, 0.1)$ , calculate  $H(y, \hat{y}')$ 
  - Your loss function decreases.

TRUE LABEL	PREDICTED LABEL	COMBINED
1	$\text{Log}(.6)$	-.2218
0	$\text{Log}(.3)$	0
0	$\text{Log}(.1)$	0
	Negative SUM:	.2218

- What would happen to the categorical cross-entropy loss if any of the value in the estimated label were zero? Is this possible?
  - It would be considered undefined, rendering the problem unsolvable. Is it possible? No or just very (**very**) unlikely. Most of the time a neural network uses the sigmoid function as the output activation function, so it always going towards zero or one but never reaches either value.

### Q3

[10 pts] To solve a classification task, an engineer first trains a network on 50 samples. Training converges, but the training loss is very high. He then decides to train this network on 50,000 examples. Is his approach to fixing the problem, correct? If yes, explain the most likely results of training with 50,000 examples. If not, give a solution to this problem.

Yes, this is moving in the correct direction. As long as each new sample brings value to the training of the neural networks and the samples are not 49950 bad samples. The more unique training samples, the better for the neural network training. However, it can get computationally expensive. The general rule is that the more samples the better, if the samples are good, unique, and diverse. However, at a certain point the increase to accuracy from adding another sample will not be worth how much more computationally expensive the model becomes.

### Q4

[20 pts] Discuss the following questions:

- Does the classification accuracy on the training data generally improve with increasing training data size? At what point do training and testing accuracy become similar?
  - Yes, the increase of training data and number of epochs will increase the training data. The point where training and testing accuracy become similar is based on the dataset. It is important to not underfit or overfit the model.
- What is the effect of increasing the regularization parameter on the training and testing accuracy? At what point do training and testing accuracy become similar?
  - Increasing the regularization parameter of the model will most likely decrease the training accuracy but improve the testing accuracy. This is because the objective of the regularization parameter keeps the model from overfitting. The testing and training accuracy become similar at a point that depends on the dataset; however, it will be theoretically sooner with a higher regularization parameter.

## Q5

[20 pts] Calculate the number of parameters in each spatial layer for column D for the VGG model.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
conv2d_1 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_3 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_4 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_5 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 256)	0
conv2d_7 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_8 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 512)	0
conv2d_10 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_11 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 4096)	102764544
dense_1 (Dense)	(None, 4096)	16781312
dense_2 (Dense)	(None, 2)	8194

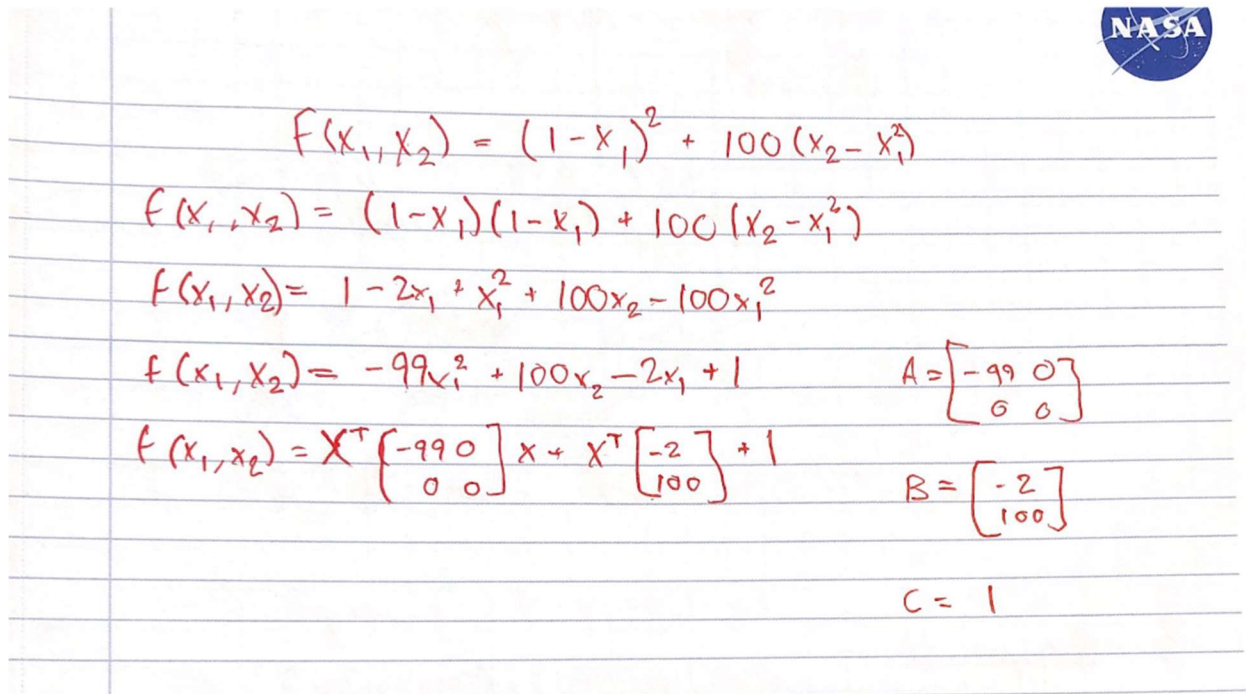
Total params: 134,268,738

Trainable params: 134,268,738

Non-trainable params: 0

## Q6

[10 pts] Consider the problem  $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)$ , let's define  $x = (x_1, x_2)^T$ . Please find a symmetric matrix  $A$ , vector  $B$ , and a constant  $c$  so that the original  $f(x)$  can be converted into quadratic form as  $f(x) = x^T A x + x^T B + c$



Handwritten solution for the quadratic form conversion of  $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)$ .

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)$$

$$f(x_1, x_2) = (1 - x_1)(1 - x_1) + 100(x_2 - x_1^2)$$

$$f(x_1, x_2) = 1 - 2x_1 + x_1^2 + 100x_2 - 100x_1^2$$

$$f(x_1, x_2) = -99x_1^2 + 100x_2 - 2x_1 + 1$$

$$f(x_1, x_2) = x^T \begin{bmatrix} -99 & 0 \\ 0 & 0 \end{bmatrix} x + x^T \begin{bmatrix} -2 \\ 100 \end{bmatrix} + 1$$

$$A = \begin{bmatrix} -99 & 0 \\ 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -2 \\ 100 \end{bmatrix}$$

$$c = 1$$

## Q7

[10 pts] You have a single hidden-layer neural network for a binary classification task. The input is  $x \in \mathbb{R}^n \times m$ , output  $\hat{y} \in \mathbb{R}^1 \times m$  and true label  $y \in \mathbb{R}^1 \times m$ . The forward propagation equations are:

$$z = Wx + b$$

$$a = \sigma(z)$$

$$\hat{y} = a$$

$$L = - \sum_{i=1}^m y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

Please calculate  $\partial L / \partial W$ .

$$-L = \sum y \log(\hat{y}) + (1-y) \log(1-\hat{y})$$

$$a = \hat{y}$$

$$a = \sigma(z)$$

$$-L = \sum y \log(\sigma(z)) + (1-y) \log(1-\sigma(z))$$

$$\frac{d}{dx} = \log(x) \frac{d}{dx} = \frac{1}{x}$$

$$\frac{\partial L}{\partial w} = \left( \frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)} \right) \frac{\sigma(z)}{\partial w}$$

$$\sigma(x)$$

$$\frac{y(1-\sigma(z))}{\sigma(z)(1-\sigma(z))} - \frac{(1-y)(\sigma(z))}{1-\sigma(z)(\sigma(z))}$$

$$-L = \frac{y - y\sigma(z) - \sigma(z) + y\sigma(z)}{\sigma(z)(1-\sigma(z))} \left( \frac{\partial \sigma(z)}{\partial w} \right)$$

$$L = \frac{\sigma(z) - y}{\sigma(z)(1-\sigma(z))} \left( \frac{\partial \sigma}{\partial w} \right)$$

$$\frac{\partial \sigma}{\partial w} = \sigma'(z) = \sigma(z)(1-\sigma(z))$$

$$\frac{\partial L}{\partial w} = \left( \frac{\sigma(z) - y}{\sigma(z)(1-\sigma(z))} \right) \sigma(z)(1-\sigma(z))$$

$$\frac{\partial L}{\partial w} = \sigma(z) - y$$

Code Below:

# Midterm Question 6

April 5, 2022

## 1 Midterm Question 6

### 1.1 Loading Libraries

```
[1]: # Importing Libraries

## Importing Packages
→#####
import tensorflow as tf
from tensorflow import keras
import numpy as np
import os

## Importing Model
→#####
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D , Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam

# Personal Preference
→#####
import warnings
warnings.filterwarnings('ignore')
```

### 1.2 Making the VGGnet 16 Layer

```
[2]: # Creating the CNN

model = Sequential()
model.
→add(Conv2D(input_shape=(224,224,3),filters=64,kernel_size=(3,3),padding="same",
→activation="relu"))
model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
```

```

model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    ↪activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

```

[3]: *# Final Dense Layers*

```

model.add(Flatten())
model.add(Dense(units=4096,activation="relu"))
model.add(Dense(units=4096,activation="relu"))
model.add(Dense(units=2, activation="softmax"))

```

[4]: *# Compiling the model*

```

opt = Adam(lr=0.001)
model.compile(optimizer=opt, loss= keras.losses.categorical_crossentropy,
    ↪metrics=['accuracy'])

```



```

ModuleNotFoundError                                Traceback (most recent call last)
<ipython-input-4-b2a73ac123a0> in <module>
      2
      3 opt = Adam(lr=0.001)
----> 4 model.compile(optimizer=opt, loss= keras.losses.categorical_crossentropy,
      ↪ metrics=['accuracy'])

~\AppData\Roaming\Python\Python38\site-packages\tensorflow\python\util\lazy_loader.py in __getattr__(self, item)
     60
     61 def __getattr__(self, item):
--> 62     module = self._load()
     63     return getattr(module, item)
     64

~\AppData\Roaming\Python\Python38\site-packages\tensorflow\python\util\lazy_loader.py in _load(self)
     43     """Load the module and insert it into the parent's globals."""
     44     # Import the target module and insert it into the parent's namespace
--> 45     module = importlib.import_module(self.__name__)
     46     self._parent_module_globals[self._local_name] = module
     47

~\Anaconda3\lib\importlib\__init__.py in import_module(name, package)
    125         break
    126         level += 1
--> 127     return _bootstrap._gcd_import(name[level:], package, level)
    128
    129

~\Anaconda3\lib\importlib\_bootstrap.py in _gcd_import(name, package, level)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load(name, import_)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load_unlocked(name, import_)

~\Anaconda3\lib\importlib\_bootstrap.py in _call_with_frames_removed(f, *args, **kwargs)

~\Anaconda3\lib\importlib\_bootstrap.py in _gcd_import(name, package, level)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load(name, import_)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load_unlocked(name, import_)

~\Anaconda3\lib\importlib\_bootstrap.py in _call_with_frames_removed(f, *args, **kwargs)

```

```

~\Anaconda3\lib\importlib\_bootstrap.py in _gcd_import(name, package, level)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load(name, import_)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load_unlocked(name, import_)

~\Anaconda3\lib\importlib\_bootstrap.py in _call_with_frames_removed(f, *args,
↳ **kws)

~\Anaconda3\lib\importlib\_bootstrap.py in _gcd_import(name, package, level)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load(name, import_)

~\Anaconda3\lib\importlib\_bootstrap.py in _find_and_load_unlocked(name, import_)

ModuleNotFoundError: No module named 'keras'

```

```
[5]: model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
conv2d_1 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_3 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_4 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_5 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 256)	0
conv2d_7 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_8 (Conv2D)	(None, 28, 28, 512)	2359808

conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
-----		
max_pooling2d_3 (MaxPooling2)	(None, 14, 14, 512)	0
-----		
conv2d_10 (Conv2D)	(None, 14, 14, 512)	2359808
-----		
conv2d_11 (Conv2D)	(None, 14, 14, 512)	2359808
-----		
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2359808
-----		
max_pooling2d_4 (MaxPooling2)	(None, 7, 7, 512)	0
-----		
flatten (Flatten)	(None, 25088)	0
-----		
dense (Dense)	(None, 4096)	102764544
-----		
dense_1 (Dense)	(None, 4096)	16781312
-----		
dense_2 (Dense)	(None, 2)	8194
=====		
Total params: 134,268,738		
Trainable params: 134,268,738		
Non-trainable params: 0		
-----		

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: