# CISC 3440 Fall 2020 Final

# Part II

Due **Monday, 12/14 8pm**

Share with [rebecca.levitan@share.brooklyn.edu](mailto:rebecca.levitan@share.brooklyn.edu) and denys.katerenchuk@gmail.com

## Intro to deep learning

|  |  |
| --- | --- |
| What is another name for a **fully connected layer**? What are its characteristics? | Dense layer; each layer in the dense layer is connected to every neuron in the previous layer. [Each neuron takes input from said neurons and multiplies them by their weights, then adds in a bias vector to calculate the input. The input is then passed through a (non-linear) activation function to produce output for the next layer.] |
| What is the equation for the output of a neural network node with a ReLU activation function? | output = max(0, input)  where:  input = bias + dot\_product(previous\_layer\_outputs, previous\_layer\_output\_weights) |
| What is the **backpropagation** algorithm? | 1. Forward Pass - Used to predict a result / calculate the output of the output layer by going through the layers one by one and preserving all intermediate results. 2. Check loss via loss function to get the output error 3. Backward Pass - Calculate previous layer’s part in the output error (using calculus), using the results to calculate layer by layer through the first hidden layer. 4. Gradient Descent - Adjust weights using some form of gradient descent. |
| What is the output of a single forward pass through this network with the input given? Assume the hidden units have a ReLU activation function. | Neuron names: i1, i2, i3 for input layer  h1, h2 for hidden layer  o1 for output layer  Assuming no bias for simplicity’s sake.  h1\_input = 0.1\*0.1 + 0.8\*0.3 + 0.5\*0.7 = 0.6  h2\_input = 0.1\*0.2 + 0.8\*0.6 + 0.5\*0.5 = 0.75  h1\_output = max(0, 0.6) = 0.6  h2\_output = max(0, 0.75) = 0.75  o1\_input = 0.6\*0.4 + 0.75\*0.2 = 0.39  o1\_output = max(0, 0.39) = **0.39** |
| Write keras code to implement a deep neural network for a classification task with four labels. It should include four hidden layers with 256 ReLU units each, 0.2 dropout and batch normalization. | # Skipping imports  # Initialize with a flatten layer to format input data  model = keras.models.Sequential([keras.layers.Flatten()])  # Add 4 dense, dropout, and batch normalization layers  for x in range(0, 4) :  model.add(keras.layers.BatchNormalization())  model.add(keras.layers.Dropout(rate=0.2))  model.add(keras.layers.Dense(256, activation=”relu”))  # Add output layer with 4 neurons  model.add(keras.layers.Dense(4, activation=”softmax”)) |

## Training DNNs

(For your reference: these questions can be answered in 1-2 sentences, <25 words)

|  |  |
| --- | --- |
| What is the purpose of He initialization? | To remove or reduce the problem of unstable gradients. |
| What is batch normalization? What is its purpose? | It standardizes and normalizes input data [in batches]. If you put a batch normalization layer for every hidden layer, you remove the problem of unstable gradients and overall regularize your neural net. |
| What is momentum? What is its purpose? What is generally the best learning optimizer? | Each gradient is remembered [usually with a decay rate] and the new gradient is added on top to increase training speed without preventing convergence. Adam is empirically considered the best optimizer. |
| What is early stopping? What is its purpose? | You stop training the model when the validation error stops going down. This is used to minimize overfitting that neural nets are very prone to do. |
| What is dropout? What is its purpose? | Dropout is the concept of ignoring some random neurons in a DNN each training step, which allows for a more robust model. |

## CNNs

|  |  |
| --- | --- |
| What is the typical architecture of a convolutional neural network? How do the convolutional and pooling layers work together? | Convolutional layers take some of the input space, filter it to find patterns, and pass it on to the next layer to discern for detailed patterns. These filters add up since you output multiple filters for one set of input. Pooling layers subsample the input, reducing the dimensionality and computing load. The CNN alternates between these two layers to both learn and stay consistent, usually with a few dense layers and an output layer at the end. |

## GANs

|  |  |
| --- | --- |
| What are the two parts of a GAN and what does each one do? | Generator: Generates fake data to try to trick the discriminator into labelling as real data.  Discriminator: Is sent both real data from the dataset and fake data from the generator and tries to label them real or fake. |

## RNNs

|  |  |
| --- | --- |
| What are RNNs used for? | Stock price prediction, sentiment analysis, text generation. Things that require predictions to be based on previous predictions, not only the input data. |
| What makes something a recurrent neuron? | Its output is used as part of the input to its next iteration/time along with the dataset’s input.. |
| What problem of plain RNNs are LSTMs and GRUs designed to address? | Memory loss. Too much data is continually passed along, so LSTMs and GRUs remove the less meaningful parts to let the stars shine. |

## NLP

|  |  |
| --- | --- |
| What is self-supervised learning? What is an example? | A learning method that first generates labels from the data, and then operates as a normal supervised model by training on that labeled data. One use is colourizing monochrome images (I would love to try one of these models with coloring books or black and white TV images). |
| What are two advantages of word embeddings over one-hot embeddings? | 1. It can use a dense vectors instead of sparse vectors 2. The vector’s weights are trained, not assigned; hence they are a meaningful representation of the words’ relationships. |
| Why does GPT-3 perform so much better than its predecessors? | It has a significantly higher amount of parameters and data contained in the model, giving it higher accuracy and better ability to generalize (so far as to perform well on data not included in its training). |