Part 1:

Recurrent Neural Networks (RNN) are a type of neural network used for sequence data. These types of neural networks are highly dependent on the order, time, and context of their inputs. All in all, Recurrent Neural Networks are for sequential inputs and sequential outputs.

A diagram of a network

Description automatically generated

Inside the RNN, the first input will go through the hidden layer and output layer. Once that value is outputted, the hidden layer will also save that output for the next input as context. For the next input, once it reaches the same hidden layer, the hidden layer will use the information of the previous input as a reference on what that value should be. This process can be repeated depending on the model’s number of inputs, number of neurons in the hidden layer, and number of outputs.

A diagram of a network

Description automatically generated

Some real examples of RNN usage are speech recognition, machine translation and image captioning. For example, for speech recognition, the whole speech would be considered the vocabulary, and the inputs can be either character or words. The characters or words will go through the model each output that was generated would use information from the previous input(s) as a context. This could be used to determine the next word in a sentence or the tone of the sentence (like negative or positive).

Unfortunately, there are some drawbacks to a basic RNN model. Basic RNN models do not do well with data with long sequences. The longer the sequence, the more inputs the model will have to remember, this will overwhelm the model due to memory which will lead to decrease in correctly predicting the output.

Thankfully, LTSM and GRU are a subset of Recurrent Neural Network models that are designed to handle that issue. LTSMs remember only the important aspects of the previous inputs for the model. GRUs are a computationally less expensive variation of the LTSM, and it’s the most RNN model that was developed.

A diagram of a tank

Description automatically generatedA diagram of a device

Description automatically generated

When comparing Recurrent Neural Networks to Convolutional Neural Networks (CNN) there are many similarities and differences. For similarities, there are both discriminative models, meaning that they use labels to discriminate and classify based on the inputs. In other words, they are supervised learning algorithms. Both models also use a dimensionality reduction technique to find important features of the inputs. For differences, CNNs are great for image recognition, unlike RNNs. RNNs specialize in predicting outcomes that are based on the context of previous inputs, CNNs do not do that. Lastly, RNNs are great sequential data like times-series, speech recognition and machine translation, unlike CNNs.

A diagram of a network

Description automatically generated

Part 2:

Generative Adversarial Networks, also known as GANs, are a type of generative model that is used to replicate images based on the images that it was trained. GANs are great for creating images that look realistic but are not.

A diagram of a diagram

Description automatically generated

GANs use a game theoretical approach to train the model. Inside the GAN model, are two models, the generator and discriminator. The first step in this model, real images are fed into discriminator model, the discriminator will classify if either the batch of images given are real of fake. If the discriminator model makes a wrong classification, the model will backpropagate by adjusting the weights to correct the discriminator. Next, using the updated weights, another batch of images are fed to the discriminator, these are lowered dimensioned versions of the real images that were fed previously. These images are generated from the generator model. The discriminator will either classify these images as real or fake. The goal of this process is to fool the discriminator into thinking that these images are real. If the discriminator classifies these images as fake, the model will backpropagate the generator model by adjusting its weights. This will generate the lowered dimensioned images into looking more and more realistic after each backpropagation process.

A diagram of a person's face

Description automatically generated

Recently, usage of GANs have been prominent today. GANs are used a lot in deepfakes and generating realistic photos that are not actually real. As we progress, these GANs are getting better and better, making very hard to tell the difference with the human eye. Nowadays, the only way tell that these images are fake are the slight distortion in the images that you would have really focus on for a long time to detect.

A collage of a group of people

Description automatically generated

I was very curious on some other real-world applications of GANs usage. I ended up finding there is usage of GANs for video games. For example, there is a video game called MarioGAN that generates new levels of Mario using GANs (Trinh, 2020). That’s pretty interesting in my opinion!

A screenshot of a video game

Description automatically generated

There are many similarities and differences between Generative Adversarial Networks and Convolutional Neural Networks. For similarities, both neural networks use convolutional architecture in their frameworks, this includes lowering the dimensionality of the images to only include distinct images for recognition purposes. For differences, GANs do not labeled outputs, instead the output is a variation of an image that includes noise, this can fall under unsupervised learning. CNNs’ outputs are labeled data, and they fall under supervised learning. GANs are considered generative models and CNNs are discriminator models. Lastly, GANs are harder to train that CNNs because there are is more variance in training the images because the model is training based on trying to replicate image.

A diagram of different types of cubes

Description automatically generatedA diagram of a fake image

Description automatically generated

Trinh, D. (2020). ART-ificial Intelligence in Gaming Part 2: GANs. Medium. <https://medium.com/analytics-vidhya/art-ificial-intelligence-in-gaming-part-2-gans-e2e5381c8bd7>

Moradi, H. (2020). ADTA 5550.101. Recording. <https://unt.zoom.us/rec/play/Ir6gvdiFZYMfIO4w2Ejlj3jS2f6puXPQ6jZjVey_mQ_4yZR-fESmDvbbC2Aptkcz2fUXwoNdYp_XnpBa.FGGQnG_719Mzr9fd?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Funt.zoom.us%2Frec%2Fshare%2FDfcmn0dXdkqLyZPnvtZqnfaWbmlOWBuVzLd9mn82kC39JgpehvkUE2N_F2n_yKv2.PO5S34TDQ3UQR9rV>