Mathew Williams
2/2/2023
Data Mining 2
Homework 1 Challenge Question

My question is: what are some of the most common neural network architectures (besides Perceptrons) used? Give a brief overview of how they work.

- **Autoencoders[1]**:
    - An unsupervised machine learning algorithm where we use neural networks for the task of representation learning.
        - Representation learning is a machine learning approach that allows a model to discover the representations of raw data for the use of feature detection and classification[2].
    - This type of neural network is designed in such a way that we impose a "bottleneck" in the network which forces a compressed knowledge representation of the original input.
        - If some sort of structure exists in the data, this structure can be learned and leveraged when forcing the input through the network's bottleneck.
        - The bottleneck is a key attribute of an Autoencoder's design, without it, the network could simply memorize the input values by passing these values through the network.
        - A bottleneck constrains the amount of information that can traverse the full network, which forces a learned compression of the input data.
    - An Autoencoder model balances the following traits:
        - "Sensitive to the inputs enough to accurately build a reconstruction."
        - "Insensitive enough to the inputs that the model doesn't simply memorize or overfit the training data."
    - Shares some similarities with Generative Adversarial Networks.
- **Residual Neural Networks[3]**:
    - A neural network architecture that was created with the idea of trying to solve the vanishing gradient problem.
        - Vanishing gradient Problem:
            - Occurs when training a deep neural network using gradient based learning and backpropagation.
            - In backpropagation, we use gradients to update the weights in the network in hopes that the model will perform its task better on the next iteration.
            - Sometimes that gradient becomes vanishingly small, effectively preventing the weights to change values, which causes the network to effectively stop training.

    - Residual neural networks are a type of neural network that applies identity mapping, where the input to some layer is passed directly or as a shortcut (skip connection) to some other layer, in order to solve the vanishing gradient problem.
        - Traditionally, layers found earlier in the network during forward propagation pass their inputs to layers found later in the network.

- **Recurrent Neural Networks[4]**
    - A type of neural network which trains on sequential or time series data.
    - These models are distinguished by their "memory" as they take information from prior inputs to influence the current input and output (similar to that of residual neural networks).
        - In residual neural networks, input is passed from a previous layer to a layer found later in the network in order to prevent the vanquishing gradient problem.
        - In a recurrent neural network, the output of the model is fed into the hidden layers as input in addition to its normal input.
    - Recurrent neural networks share parameters across each layer of the network.
        - Regular neural networks have different weights across each node, recurrent neural networks share the same weight parameters within each layer of the network.
    - Recurrent neural networks use a variation of backpropagation called backpropagation through time (BPTT) which is made to work with sequence data.
        - BPTT differs from the traditional backpropagation algorithm in that it sums errors at each time step whereas regular neural networks do not need to sum errors since there are no shared parameters across each layer.
- **Generative adversarial network[5]**
    - An approach to generative modeling using deep learning methods, such as CNNs.
        - "Generative modeling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns of data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset."
    - GANs are a generative model that frames the problem as a supervised learning problem with two sub-models.
        - The generator model is trained to generate new examples.
        - The discriminator model that is trained to classify examples as either real (from the dataset) or fake (made from generator).
        - The two models are trained together in a zero sum game until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples.
            - A zero sum game is a game involving two sides where one side's loss is the other sides' gain.

        - If the generator can trick the discriminator, then the discriminator needs to update its weights.
        - If the discriminator isn't tricked by the generator, then the generator needs to update its weights.
    - Shares some similarities with Autoencoders.
- **Convolutional Neural Network[6]**
    - A type of neural network which was primarily designed on images, but can also work with speech as well.
    - For image classification models prior to CNNs, programmers need to do time-consuming feature extraction from images. These features were then fed into a neural network for classification.
    - CNNs provide a more scalable approach to image classification and object recognition tasks by utilizing matrix multiplication to identify patterns within an image.
    - A CNN consists of 3 main layers: the convolutional layer, the pooling layer, and the fully connected layer.
        - Convolutional Layer:
            - The core of the CNN where most of the computation takes place.
            - Contains 3 major components: the input image, a filter, and a feature map.
            - The filter (also known as a kernel) is responsible for moving across pixels on an image checking for present features via the dot product operation.
            - The combined outputs of the dot product mentioned previously produce the feature map.
        - Pooling Layer:
            - This layer is responsible for dimensionality reduction or downsampling.
            - The pooling operation also has a kernel that moves across the feature map and applies an aggregate function to a group of pixels.
            - Done to help reduce the number of parameters, improve efficiency, and limit the risk of overfitting.
        - Fully Connected Layer:
            - The fully connected layer can be thought of as a simple fully connected feedforward network that does the actual classification or object detection.
    - One can use multiple Convolution and/or pooling layers before passing the input to the fully connected layer.

References:

1. https://www.jeremyjordan.me/autoencoders/
2. https://analyticsindiamag.com/a-comprehensive-guide-to-representation-learning-for-beginners/

3. https://iq.opengenus.org/residual-neural-networks/
4. https://www.ibm.com/topics/recurrent-neural-networks
5. https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
6. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53