

United World Institute of Technology (UIT)

**Summative Assessment (SA)**

**Submitted BY**

**Milind Jain**

**(Enrl. No.: 20220701011)**

**Course Code and Title: 21BSAI99E43 – Artificial Neural Network**

**B.Sc. (Hons.) Computer Science AIML**

**IV Semester**

**CO1) Explain how neural networks can be employed in perception modules of autonomous vehicles for tasks such as object detection, lane tracking, and traffic sign recognition.**

These networks cover objects in different sizes and angles because they are experts at extracting hierarchical features from these images.  
CNNs (R-CNNs) with a regional focus: For object detection usually R-CNNs and their variants such as Fast R-CNN and Faster R-CNN are used. They determine specific regions of the pictures as candidate areas and then they detect objects from the images.  
Single Shot Multibox Detector (SSD) and You Only Look Once (YOLO): Such imitate constructions of detecting objects almost instantaneously. They achieve fast response by the method of direct prediction of the bounding box and class probability without neural networks.

Lane Directions:  
  
Recurrent Neural Nets (RNNs): RNNs can deliver sequential data processing, so they can be used in lane detection applications. RNNs can analyze video feed frames to calculate predictions about the lane trajectory in sequence.

Convolutional Neural Networks (CNNs): Moreover, neural networks can also be implemented by converting lane-tracking problems into image segmentation issues. Likewise, the neural network splits the image into different types of classes which includes the lane markings in my case.  
  
undefined  
  
Convolutional Neural Networks (CNNs): Traffic sign recognition applications use CNNs more as they have the skill of merging the noticeable features of images. Such networks can be taught by sample labeled datasets, each of which contains images of traffic signs and their classes.  
Transfer Learning: Transfer learning, as a study method, can be applied to any previously trained model (like ResNet, VGG, or MobileNet) that is already working on the specific traffic signs dataset. It is easy to apply because it utilizes the data from large image datasets like ImageNet and it is more effective than other methods.

Attention Mechanisms: The designed CNN models can be modified to improve the model's ability to identify traffic signs in complex backgrounds by adding attention mechanics that are oriented only to significant parts of an image.

In each of these tasks, neural networks learn from labeled datasets using a method called supervised learning. These models can track lanes, recognize objects in real-world driving scenarios, and identify traffic signs once they have been trained. Furthermore, advancements in neural network architectures, optimization techniques, and hardware acceleration have made significant gains in the efficacy and performance of perception modules in autonomous vehicles possible.

**CO2) Discuss the conditions under which a single-layer perceptron can approximate the decision boundary of a Bayes classifier for linearly separable and non-linearly separable classes.**

The neurons in a single-layer perceptron can fit any decision boundary for a Bayes classifier in some cases. More specifically, the Bayes classifier is geared toward minimization of misclassification risk by allocating class labels to the category with the highest a posteriori probability. The following are the requirements for both linearly and non-linearly separable classes: The following are the requirements for both linearly and non-linearly separable classes:  
  
Class Dividends in Sequence:  
  
Conditions: You can also use a special classifying hyperplane in the feature space (this method is based on the principle of linear separability) to divide the groups. If the class can linearly separate— meaning at least one hyperplane can perfectly classify data— then a single-layer perceptron can determine a boundary that approximates the Bayes classifier.

Perceptron Learning: By the way, the Variation method can be used to make the system work by which perceptron's weights and biases are changed till a solution is reached that would classify each training sample correctly. If the classes are linearly separable then the decision line will go along with the Bayes boundary and the simple updating algorithm concerning learning and convergence of the perceptron will be successful.  
Classes Not Separatable Linearly:  
  
Conditions: That notion means a linear separability cannot lead to the class partition which was in the original space. If the perceptron consists of only one layer, the one may not be capable of approximating the Bayes classifier's decision boundary with enough precision in these cases.  
  
Limitations: Due to the perceptron algorithm being tailored for finding a linear function of a separation line, it can't resolve any non-linear decision boundaries efficiently. In the same way, data classes that are not linearly separable might not be able to converge or, if so, might converge to an imperfect solution.  
  
Further Non-linearity Extensions: Non-linearly separable classes, mainly, are usually linearized by more sophisticated models, such as MLPs having hidden layers or kernel-based methods like SVMs. They are capable of making non-linear decision boundaries themselves, by performing non-linear mapping of the input space to a higher feature space where classes become linearly separable.  
  
Hence, mapping the decision boundary for the two classes, which are linearly separable, and implementing it for the classifier based on the Bayes approach, which is used to perform the functions of the single-layer perceptron architecture. The "dummy" neuron is now confronted by its own limitations whenever a distinction between disparate cases needs to be made that demands for more refined methods of approximation of the decision boundary of the Bayes classifier.

**CO3) Describe the concept of cross-validation and its importance in assessing the generalization performance of neural network models. Discuss specific cross-validation techniques suitable for neural network training.**

Experience the results of a machine learning algorithm on fake data that is doctored by the bootstrap process. It is hence vital to examine how well the models developed from neural networks generalize (how suitable they are to data that wasn't part of training). Therein, the dataset is broken into different groups, and the model trained on one of the groups, assesses its capacity using another or another group.  
  
Here are some reasons neural network models should use cross-validation: Here are some reasons neural network models should use cross-validation:  
  
The gen­eral­iz­a­tion per­for­mance of a neu­ral net work model is based on cross-val­i­da­tion using new, un­chal­lenged date sets. To make a model with the generalizing capacity to other datasets and prevent overfitting onto the training dataset; it is very necessary.  
Selection of the Model: It turns out that cross-validation is helpful in case we compare different neural network architectures, training algorithms, and hyperparameters. It is possible for experts to identify which model is the best for deployment by subjecting particular models to different segments of the data and then assessing them on their performance.  
Cross-validation might be your friend if your main goal is finding out to what extent your models may be affected by changes in the training set. In addition, the tool provides insights into the model’s regularity in facial characteristic representation, which could point out the appearance of possible biases or model weaknesses toward particular data subsets.  
Certain cross-validation methods that are appropriate for training neural networks consist of: Certain cross-validation methods that are appropriate for training neural networks consist of:  
  
K-Fold Cross-Validation: Additionally, through the use of a partitioning function, or in other words a process that segments the data into subsets of almost the same sizes, choices can be created from the shared data set. Folds K-1 are held as a training set in which the model is trained for K times while the remaining fold is used for validation. The latter one (average of performance metrics of all K folds) is a final performance metric used to give the overall estimate of the model's performance.

**Stratified K-Fold Cross-Validation:** This method guarantees that each fold maintains the percentage of classes in the dataset, making it comparable to K-fold cross-validation. Because each fold accurately reflects the class distribution of the entire dataset, it is especially helpful for classification tasks with unbalanced class distributions.

Leave-One-Out Cross-Validation (LOOCV): In LOOCV, the training data set is left with the data points after the validation set has been used on each data point once. This processing step goes for every data point in a dataset, so no data point is left behind. The fact that LOOCV does not conduct subjective evaluation of the model's performance but gives objective results on the model's performance is the problem as the latter may sometimes be computationally costly and slow, especially with large datasets.  
  
Leave-P-Out Cross-Validation: In this case, the generalized LOOCV is a method which takes the remaining data to learn the model while P data points are set out for validation. A balance is found between electronic interference with single person electronic K-fold cross-validation (KFC) and delaying time with multi-person electronic leave-P-out cross-validation (LPC).  
  
  
Models can get their best generalization performance most likely by cross-validating their neural networks which leads to model selection and robust application of them in the real world.

**CO4 a) Analyze the performance of the SOM algorithm in terms of feature representation, clustering, and classification accuracy using real- world datasets.**

An artificial neural network type called the Self-Organizing Map (SOM) algorithm is learned to generate an organization of low-dimension representation of the input space through a training process conducted without supervision. It is well known as a tool too challenging and prestigious to handle tasks such as feature representation, clustering, and even classification. Using real-world datasets, let's examine how well it performed in each of these areas: Using real-world datasets, let's examine how well it performed in each of these areas:  
  
Display of Features:  
  
  
SOMs are capable, in terms of effectively projecting the high-dimensional data onto the two-dimensional grid, of keeping the underlying structure and mathematical relations of the data.  
When input vectors are Gaussian written in the SOM grid and when they are similar to each other clusters are formed.  
We can do the exploratory data analysis to find groups or clusters within the data before knowing the actual number of clusters by using SOM which can be a very convenient tool to do such work.  
SOMs help make sense of the multidimensional data needed for customer segmentation where they are applied in real-world clustering applications such as marketing, text mining, and anomaly detection.  
  
  
The approach permits generating a topological map without altering the input body's intrinsic characteristics as the similar data is situated closely on the grid.  
SOMs can effectively map volumes of data with highly elaborated structural features, such as texts, pictures, or data from sensors into a low-dimensional plot. Deployments of SOMs have been relied upon to the classification from words and the visualization of multi-dimensional gene expression data, for example.  
  
Clustering: Unlike SOMs capable of clustering the data points of different nature into close-by nodes on a map, they are ideally meant for tasks, where grouping is the main issue.

Accuracy of Classification  
  
  
SOMs have the possibility to be the exclusive essay just for indicating goals, even if their basic application is in unsupervised teaching.  
The SOM as a pre-processing step to enact a data lowering of a representation of the input data is a widespread technique. A feature map is first computed as input for the model, this is done by applying data transformations like support vector machine or deep neural networks.  
Support Vector Machine function can also support the reduction of dimensional and feature extraction which may lead to subsequently increased accuracy of the classification process performed by the classifiers.  
In contrast with, a direct classification using SOMs might not always produce the best results in a complicated task for classification whose decision boundaries are nonlinear. Supervised learning implementations similar to the one described above could be effective in these cases.  
To sum up everything the SOM algorithm demonstrates competence in feature representation as well as clustering tasks, which appears to be just the case when we deal with complex, multidimensional data sets, which sometimes are difficult even to perceive. The straightforward application of the method for classification does not always give good results compared to that of other supervised learning types, however, it can be used as a preprocessing stage for classification work and hence, the geometry may indirectly influence the accuracy of classification. However, this should not be overlooked, and an understanding of specific tasks and the features of datasets that are used is necessary for the successful practical application of SOMs.

**CO4 b) Introduce the Hopfield model as a recurrent neural network paradigm for associative memory.**

It was developed in 1982 by John Hopfield, the physicist, who labeled Hopfield's model as an Associative Memory paradigm. It is an artificial neural net that has connections among the nodes and points to the other with the feedback shapes. It is a single layer in the design. Hopfield networks are based on memories and discharge currents used to store patterns that can be pulled out even when input data is incomplete or corrupted. The Hopfield model is explained as follows: The Hopfield model is explained as follows:  
  
Structure of the Network:  
  
  
Symmetric weights connected the neurons which constitute the honest network. Each of the neurons constituting the network is fully connected with each other. Hence, each neuron in the network is connected to every other neuron.  
  
Every neuron can be in the decimal system carrying the constant values +1 or -1. "The patterns of firing or quietness of neurons will be reflected in these values".  
The networks' pattern of expression is dictated by links between neurons, encoded by weight matrix. Of course, the majority of the time, the zeros appear on a diagonally located symmetric weight matrix.  
  
The function of Energy:  
  
The conservation of the Hopfield network's configurations relies on an energy function that is decreased as the network converges upon stable states that reflect patterns which are stored.  
The two most important parts of the quadratic form, that is the weight matrix and the states of activation of the neurons are the ones that define the energy function in its simplest form. In this respect network not only represents the total energy of the system, but also it is a balance between consumption and production.  
States that have a minimal value of energy function are those closer to where it could converge that is a state where the network is most likely to converge. These states are therefore a stable attractor pattern or a set of preserved patterns within the net.  
  
  
**The Learning Rule**  
  
Hebbian learning rules which may also be known as unsupervised learning algorithm is known to be an efficient method of setting the weights of the Hopfield network  
The connectivity between co-active neurons, that is, those with nearly conterminous activation during training, increases with the strength of their interaction according to the Hebbian rule.  
In a binary classification problem, there is only one weight matrix computed using training patterns and normalized to make a symmetric weight matrix.  
  
Associative Retention:  
  
Hopfield's network is an associative memory model with the particular property of successfully retrieving stored patterns given incomplete or noisy inputs.  
The network can effectively fill gaps or shortages of the input by amplifying that portion that is carried toward already stored patterns provided at least a partial or damaged input.  
Updated asynchronously or synchronously, by a means of which the status of neurons evolves in other neuron neighborhoods by the current state of neurons is the update scheme that is used to model the network's dynamics.  
  
Recognition, optimization, and combinatorial optimization are just some of the ways the model that was demonstrated by John Hopfield is used in the real world. Despite the pitfalls such as capacity limitations and transient virus stress, the Hopfield network still remains relevant. It provides a significant concept on associative memory networks and which has inspired new research in the field.