Letter Recognition using SVM and Artificial Neural Network

Jishma Siraj
Machine Learning Intern
AI Technology and Systems
www.ai-techsystems.com
jishmasiraj@gmail.com

Abstract—**Classification is one of the most primary use of data science and most frequently encountered decision making task of human activity.It's a field of active research and numerous applications. Classification is an essential feature to separate large datasets into classes for the purpose of Rule Generation, Decision Making, Pattern Recognition, Dimensionality Reduction, Data Mining etc. Support Vector Machine (SVM), which is one of the most popular classical machine learning algorithms, and Artificial Neural Networks(ANN) are powerful and robust classifying tools.They are widely used in different fields like medicine, agriculture, climate, military etc.Though ANNs are said to perform a lot better than machine learning algorithms on large datasets,they are prone to converging on local minimas and also has a tendency to overfit with increase in number of layers or when training time increases.SVM doesn't suffer from either of these problems and are seen to outperform ANNs in several cases. .Here we do a comparative study between SVM and ANN using a numerical dataset for the purpose of letter recognition.**

Keywords—Machine Learning, Support Vector Machine, Artificial Neural Networks,Deep Learning.

INTRODUCTION

Letter Recognition is an interesting and popular field of research in Pattern Recognition and Artificial Intelligence.The aim of this paper is to have a comparative study between Support Vector Machine and Artificial Neural Networks on the basis of their performance as a multiclass classifier..Both are very powerful supervised classification tools.Out of all the classical ML algorithms one can say SVM is the most robust classifier.Though there are many different approaches to solving character recognition problems ,most common and popular approaches are based on neural networks.There are several published studies that compare the paradigm of neural networks against the support vector machines. The main difference between the two paradigms lies in how the decision boundaries between classes are defined. While the neural network algorithms seek to minimize the error between the desired output and the one generated by the network, the training of an SVM seeks to maximize the margins between the borders of both classes[2].

I. DATASET

A. Dataset Description

Letter Recognition Data Set was taken from the UCI Machine Learning Repository . The samples character images were retrieved from 20 different fonts and each letter was randomly distorted to produce 20,000 unique records. Each record was converted to 16 numerical attributes which ranges from 0 to 15 . We split the datasets of 20,000 records into two parts: the training dataset (16000 records, 80% of the total dataset) and the testing (4000 records, 20% of the total set).

B. Attribute Information

1. letter capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total # on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)
10. y2bar mean y variance (integer)
11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)

17. yegvx correlation of y-ege with x (integer)

II. DATA ANALYSIS

The dataset did not require any preprocessing as its almost corrected already and it does not contain any noisy data or inconsistent values.

A . *LETTER COUNTS*

The dataset contains 20,000 data points.The letter U has the most instances with a count of 813 and letters Z and H has the least number of instances with a count of 734.This dataset with its 26 classes can be said to be almost balanced.

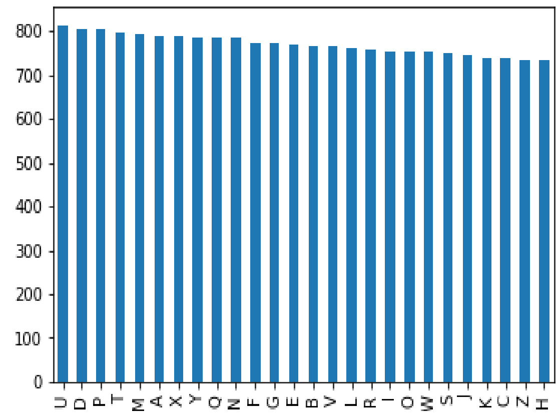

Fig.1 Bar Plot for Letter Count

B. *T-SNE VISUALIZATION*

TSNE or t-Distributed Stochastic Neighbor Embedding is an unsupervised, non-linear machine learning dimensionality reduction algorithm used primarily for data exploration and visualizing high-dimensional data. It projects multi dimensional data into a lower dimensional space.Instead of preserving global structure it preserves neighborhood structures.The performance of TSNE is fairly robust under different settings and perplexity.The appropriate values depend on the density of the data and TSNE only accepts dense matrices.

TSNE is used on the letter recognition dataset with perplexity 30 and number of iteration 1000.It resulted in a plot with well separated classes which implies a classification model can be built to separate the classes with good accuracy.

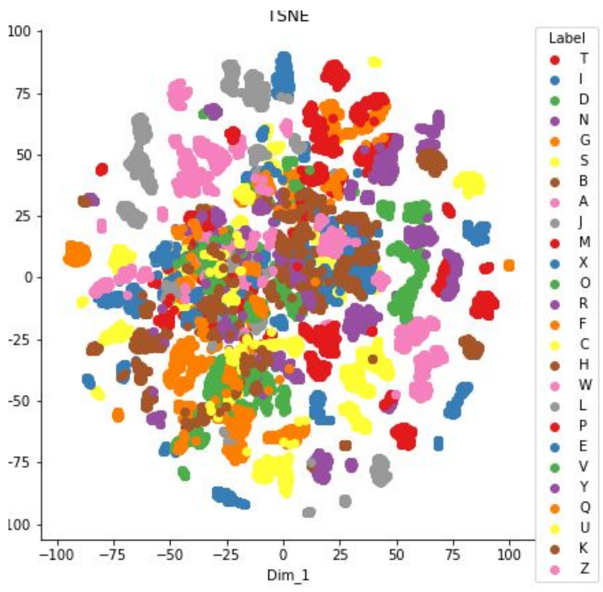

Fig.2. TSNE Visualisation of the Letter Recognition dataset

III SVM

"Support Vector Machine" (SVM) is a supervised machine learning algorithm mostly used in classification problems. SVM is based on statistical learning theory and used to find the optimal separating hyperplane in an n- dimensional space that separates data points..Each datapoint is plotted as a point in an n-dimensional space where n is the number of features in the dataset.The optimal hyperplane is the one which has the maximum margin,i.e where distance between data points of various classes are maximum.Support vectors are the data points lying close to the hyperplane which influences its position and orientation.SVM works well with clear margin of separation and is effective in high dimensional spaces.It's also memory efficient as It uses a subset of training points in the decision function called support vectors.But with large training data SVM doesn't seem to perform well as it require higher training time.Thus the major drawback of SVM is the computational cost.It also does not perform well when dataset has more noise.In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = (1 - y * f(x))_+$$

The cost is zero if both predicted and actual value are the same else we calculate the loss.We add a regularization parameter to balance margin maximization and loss.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Partial derivatives of Loss function with respect to the weights are taken to find the gradients.Using these gradients weights are updated.

$$\frac{\delta}{\delta w_k}\lambda \parallel w \parallel^2 \; = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k}\big(1 - y_i\langle x_i, w\rangle\big)_+ \; = \begin{cases} 0, & \text{if } y_i\langle x_i, w\rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

In the case our model predicts correctly gradient is only updated from regularization parameter.

$$w = w - \alpha \cdot (2\lambda w)$$

In case the model predicts incorrectly, the loss along with the regularization parameter to perform gradient update is included

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

A. LINEAR SVM

Linear SVM is a parametric model.If data is linearly separable it is better to use a linear SVM.It is less time consuming than an RBF kernel SVM but it also provides lesser accuracy than the latter.Here we use SGD (Stochastic Gradient Descent) Classifier with hinge loss to implement Linear SVM.As expected Linear Svm took less time to train.Hyperparameter tuning is done for the parameters alpha and penalty.The penalty is the regularization term used and alpha is the constant to be multiplied with the regularization term.Hyperparameter tuning is done using GridSearchCV which results in model with alpha’ 0.01’ and penalty ‘l2’ with a low accuracy of 60.61% .

B. RADIAL BASIS FUNCTION KERNEL SVM

RBF is a non parametric model and its complexity grows with the training dataset. It is one of the most popular kernels. For distance metric, squared euclidean distance is used here. It is used to draw completely non-linear hyperplanes..It is more time consuming and expensive to train an RBF kernel SVM.It is also easily prone to overfitting.Here hyperparameter tuning using GridSearchCV is done to tune the parameters Gamma and C.the gamma parameter is the inverse of the standard deviation of the RBF kernel (Gaussian function), which is used as a similarity measure between two points.C is a regularization parameter that controls the trade off between achieving a low training error and a low testing error that is the ability to generalize your classifier to unseen data.Hyperparameter tuning resulted in a model with gamma as 0.03125 and C as 32 .with test accuracy 98.17%.

.IV. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks(ANN) are computational systems modelled after human brain.It is meant to simulate the structure and function of biological neural networks.ANN are promising alternative to various conventional machine learning classification methods.. Neural networks can model real world complex relationships as they can solve non-linear problems.They also adjust to the data without any explicit programming and they can handle large datasets with ease.ANN has the ability to extract patterns and irregularities and detect multi-dimensional non-linear connections in the data.A single artificial neuron can be mathematically represented as follows

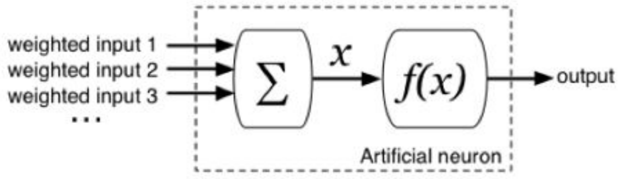

Fig.3. Structural representation of a neuron

Neural Networks is a network of interconnected neurons.Aa single layer in a network can consists few to large number of neurons.Neurons are connected to each other by links which has weight parameter associated with it.Layer in which neurons get stimuli from outside the network are called input layer and layer in which neurons’ output are used externally are called output layer.The layers lying between input and output layer are called hidden layers.The neurons of each layer consists of an activation function which decides to fire the neuron or not.They also model the input they receive into different magnitudes.Training a neural network basically involves updating the weights in the links.This updation is done through a method known as backpropagation.The figure below shows a simple three layer neural network.

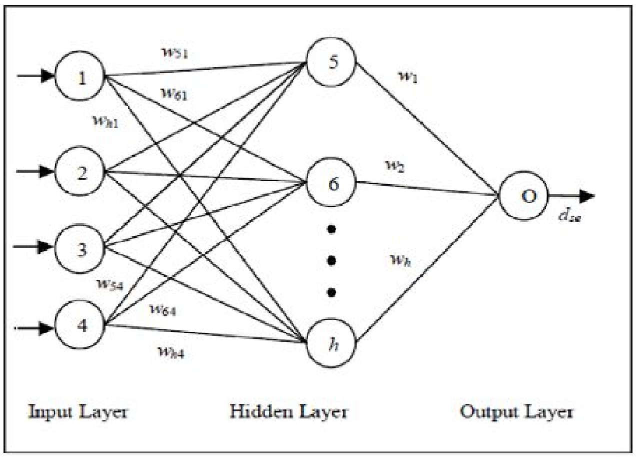

Fig. 4 3-layer neural network

A. 3 LAYER NEURAL NETWORK

A 3 layer FeedForward neural network is used to train the dataset.Training a neural network involves updating the weights through backpropagation**.**A three layer network has an input layer,a hidden layer and an output layer.The letter recognition dataset has 16 features as input and 26 capital alphabet (A-Z) as output labels.

a) Layer1(input layer): It has 512 neurons with activation function ReLu.This layer receives a 16 dimensional input.
b) Layer2(hidden layer):It has 128 neurons with activation unit ReLu.
c) Layer3(output layer):It has 26 neurons for each of the 26 class labels.The activation function in this layer is Softmax.

To prevent the internal covariance shift Batch Normalization is done after Layer 1 and Layer 2.The optimizer used here is Adam.Hyperparameter tuning using GridsearchCV resulted in batch size 100 and number of epochs 100 .

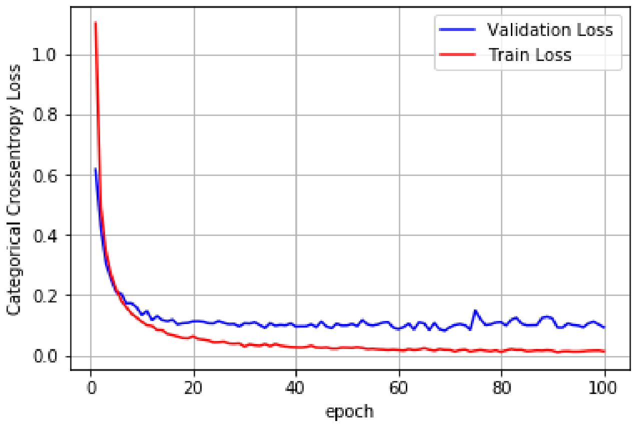

Fig.5 Validation loss v/s Train Loss for 3 layer neural network

B. 4 LAYER NEURAL NETWORK

a) Layer1(input layer):It has 512 neurons with activation function ReLu.This layer receives a 16 dimensional vector as input.
b) Layer2(hidden layer):It has 128 neurons with activation unit ReLu.
c) Layer3(hidden layer):It has 128 neurons with activation unit ReLu.
d) Layer4(output layer):It has 26 neurons for each of the 26 class labels.The activation function in this layer is Softmax.

To prevent the internal covariance shift Batch Normalization is done after Layer 1 and Layer 2.The optimizer used here is Adam.Hyperparameter tuning using GridsearchCV resulted in batch size 128 and number of epochs 100.

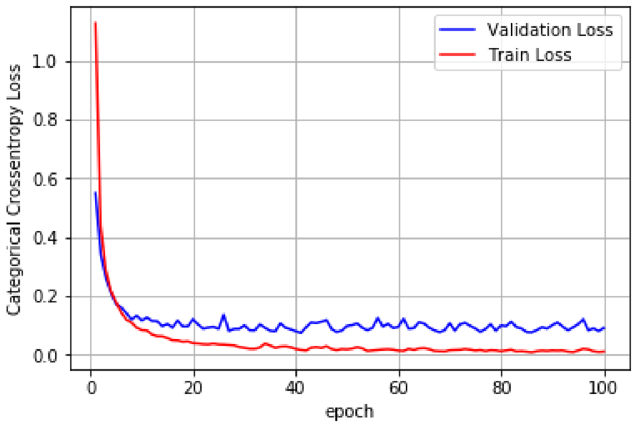

Fig.6 Validation loss v/s Train Loss for 4 layer neural network

C. ACTIVATION FUNCTIONS USED

a) ReLU: Rectified Linear Unit(ReLu) is a linear function that outputs the input if its positive else it outputs zero.

$$f(x) = max(0, x)$$

.It is the most commonly used activation function as it is easier to train,speeds up convergence,easier to compute derivatives and often achieves better performance.Unlike in sigmoid function and tanh function ReLu is not susceptible to neither vanishing gradient nor exploding gradient.But since it also outputs zero there can be a problem of dead activations.This can be avoided using a variation of ReLu known as Leaky ReLu.It allows a small, positive gradient when the unit is not active

$$f(x) = \{x \quad , \; if\ x > 0$$
$$\{0.01, \; otherwise$$

b) Softmax: A softmax activation function is used in neural networks when we have a multiclass classifier.Softmax squashes inputs to be between 0 and 1 and also divides output in such a way that the total sum of output is 1.Therefore the output vector represents the probability distribution of the class labels.It represents the probability of each of the classes being true.Using softmax function the training converges quickly than otherwise.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

D. BATCH NORMALIZATION

Batch Normalization is a technique used while training deep learning networks to improve speed,performance and stability of artificial neural networks.It standardises input to a layer for each mini batch.It helps in reducing the number of epochs needed to train the neural network.Batch normalization is done to prevent internal covariance shift which occurs as distribution of mini-batches start to differ after passing through many layers.It can also have a regularizing effect hence reducing generalization error.

E. OPTIMIZER -ADAM

Adam is a popular algorithm in the field of deep learning. It is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. Adam realizes the benefits of both AdaGrad and RMSProp.It computes individual learning rates for different parameters from estimates of first and second moments of the gradients

V PERFORMANCE RESULT

The table below shows the performance based on train and test data accuracy for each of the classifiers .

CLASSIFIER	TRAIN ACCURACY	TEST ACCURACY
Linear SVM	60.61%	61.375%
RBF kernel SVM	99.99%	98.17%
3 layer Neural Network	99.78%	97.15%
4 Layer Neural Network	99.58%	97.42%

Table 1. Performance result

VI CONCLUSION

In this report performance of SVM and simple Artificial Neural Networks as classifiers were compared using a Letter Recognition Dataset..Though the Linear SVM performed poorly even with hyperparameter tuning ,the RBF kernel SVM showed a high improvement in accuracy compared to the former.The classical RBF kernel SVM outperformed the traditional 3 layer and 4 layer Artificial Neural Networks in approximation and generalization ability.But a drawback was that with large training data RBF SVM was highly time consuming while ANN took little to no time based on the number of epochs.

References

[1] Supriya Pahwa,and Deepak Sinwar, "Comparative Study of Support Vector Machine with Artificial Neural Network Using Integer Datasets",International Journal of Advanced Research in Computer Science and Software Engineering , Volume 6, Issue 11, November 2016.

[2] Antonio Carlos Gay Thome,"SVM Classifiers – Concepts and Applications to Character Recognition",November 7th 2012.

[3] Tirtharaj Dash and Tanistha Nayak ,"English Character Recognition using Artificial Neura lNetwork",Proceedings of National Conference on AIRES-2012, Andhra University.

[4] M. R. Phangtriastu, Jeklin Harefa and Dian Felita Tanoto, "Comparison Between Neural Network and Support Vector Machine in Optical Character Recognition," Procedia Computer Science, vol. 116, pp. 351–357, 2017.

[5] V. D. Do and Dong-Min Woo, "Handwritten Character Recognition Using Feedforward Artificial Neural Network," in 7th International Conference on Latest Trends in Engineering & Technology, Pretoria, 2015.

[6] Yashima Ahuja, Sumit Kumar Yadav, "Multiclass Classification and Support Vector Machine", Global Journal of Computer Science and Technology, Vol.10 Issue 11, 2012.