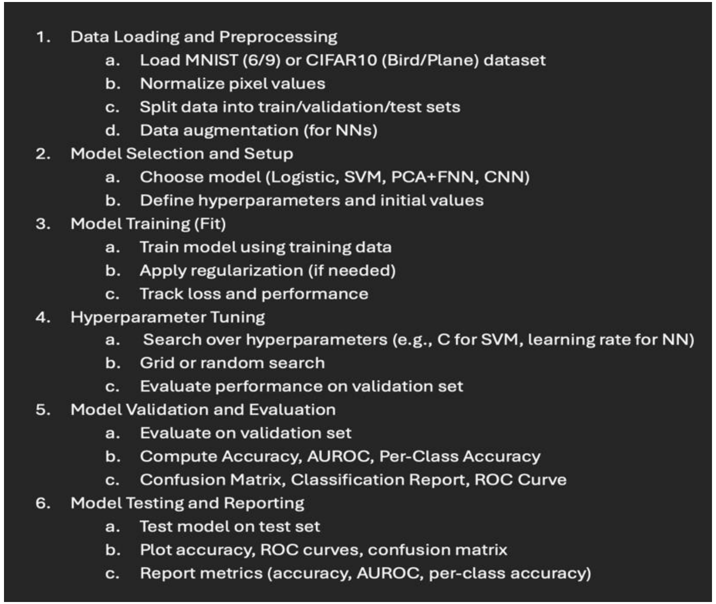
EEE 549: Statistical Machine Learning

Student: Amit Ashok Manjarly

Training, Validation, and Testing Pipeline

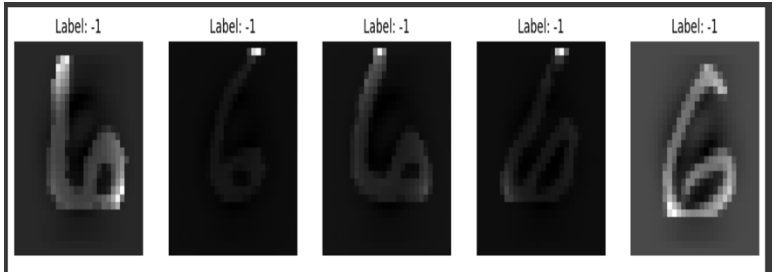


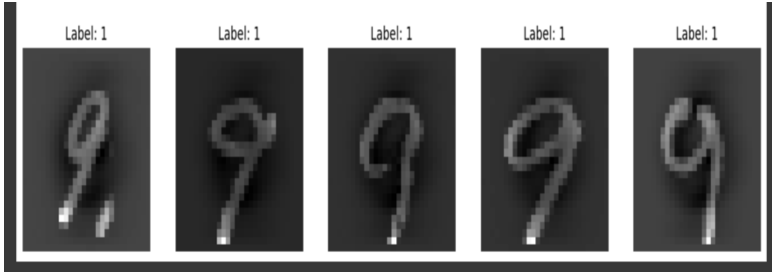
1. Data Loading and Preprocessing
    a. Load MNIST (6/9) or CIFAR10 (Bird/Plane) dataset
    b. Normalize pixel values
    c. Split data into train/validation/test sets
    d. Data augmentation (for NNs)
2. Model Selection and Setup
    a. Choose model (Logistic, SVM, PCA+FNN, CNN)
    b. Define hyperparameters and initial values
3. Model Training (Fit)
    a. Train model using training data
    b. Apply regularization (if needed)
    c. Track loss and performance
4. Hyperparameter Tuning
    a. Search over hyperparameters (e.g., C for SVM, learning rate for NN)
    b. Grid or random search
    c. Evaluate performance on validation set
5. Model Validation and Evaluation
    a. Evaluate on validation set
    b. Compute Accuracy, AUROC, Per-Class Accuracy
    c. Confusion Matrix, Classification Report, ROC Curve
6. Model Testing and Reporting
    a. Test model on test set
    b. Plot accuracy, ROC curves, confusion matrix
    c. Report metrics (accuracy, AUROC, per-class accuracy)

Insert flowchart above and clarify any steps necessary. If you used the Sol supercomputer (Jennewein, et al.), reference it

MNIST Dataset

Binary Classes





Class Description (-1)
(Insert example image above)

Class Description (+1)

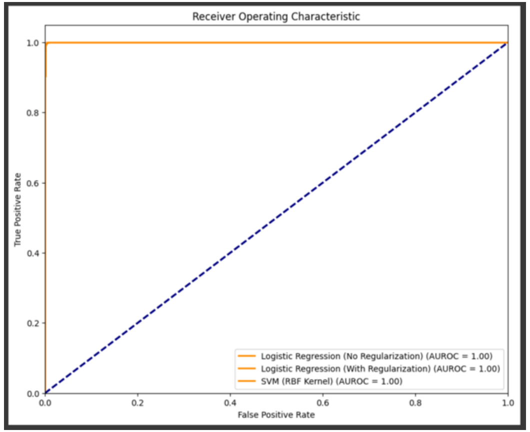
Logistic Regression

Logistic Regression Hyperparameters

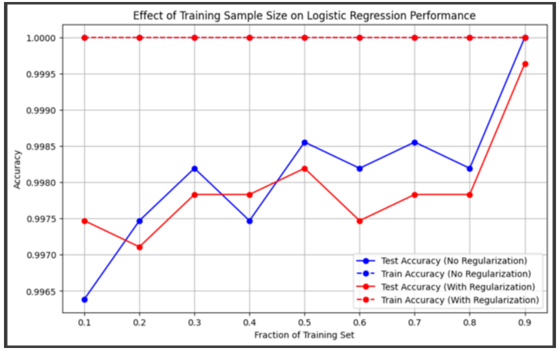
Parameter	Range	Description	Why Tuned?
C	10−4 to 10^4 (log scale)	Regularization Strength	C controls the tradeoff between bias and variance.
penalty	‘l2’, None	Specifies the regularization type.	Regularization helps prevent overfitting.
solver	'liblinear', 'lbfgs', 'newton-cg', 'saga'	Algorithm to optimize the logistic regression objective function.	Tuning the solver affects training speed, convergence, and performance.
max_iter	Integer values (e.g., 100, 1000, 2000)	Maximum number of iterations for the optimization algorithm to converge.	Tuning max_iter ensures convergence of the model.
tol	Float values (e.g., 1e-4, 1e-3)	Tolerance for stopping criteria.	Lower tolerance values result in more precise optimization, but higher values can speed up training.

Results and Plots

Method	Accuracy	Per-class Accuracy (+1/-1)	AUROC
LR Without Regularization	0.9981929887965305	Accuracy for digit 6: 0.9992952783650458 Accuracy for digit 9: 0.9992952783650458	1.00
LR With Regularization	0.9978315865558366	Accuracy for digit 6: 0.9985905567300916 Accuracy for digit 9: 0.9985905567300916	1.00



ROC Plot

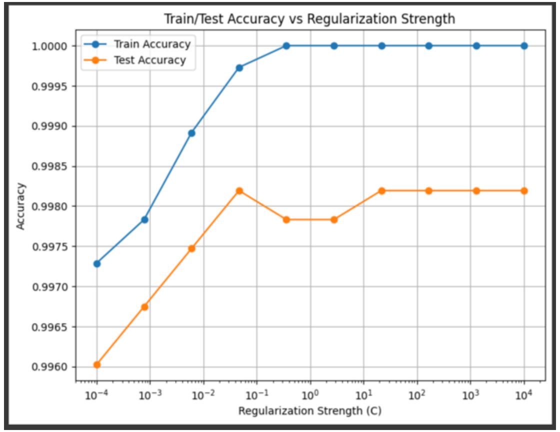


Sample Size Plot

Final Hyperparameters

LR Hyperparameters

Parameter	Value
C	1.0
penalty	l2
solver	liblinear
max_iter	1000
tol	1e-4



Plot train/test accuracy vs ONE hyperparameter of your choosing

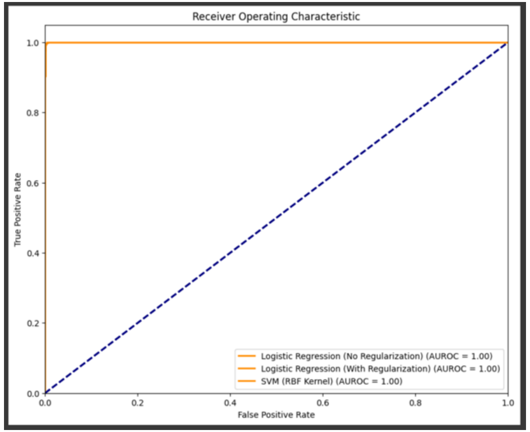
Support Vector Machines

SVM Hyperparameters

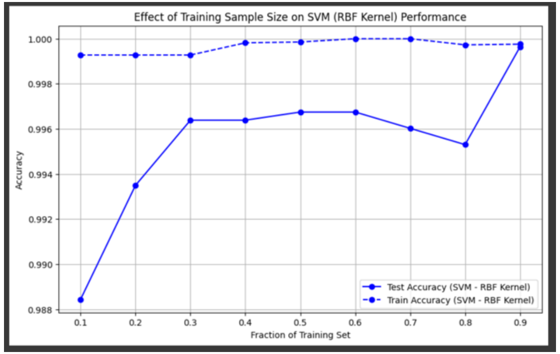
Parameter	Range	Description	Why Tuned?
C	10−4 to 10^4 (log scale)	Regularization Strength	C controls the tradeoff between bias and variance.
kernel	linear, poly, rbf, sigmoid	Specifies the kernel type to be used in the algorithm.	The kernel defines the type of decision boundary.
gamma	scale, auto, numeric values	Kernel coefficient for 'rbf', 'poly', and 'sigmoid'. 'scale' and 'auto' are predefined methods for selecting gamma, while numeric values can be directly set.	gamma controls how far the influence of a single training example reaches.

Results and Plots

Method	Accuracy	Per-class Accuracy (+1/-1)	AUROC
SVM	0.9953017708709794	Accuracy for digit 6: 0.9922480620155039 Accuracy for digit 9: 0.9922480620155039	1.0



ROC Plot

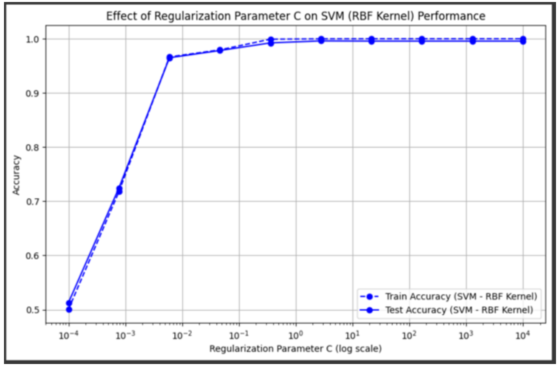


Sample Size Plot

Final Hyperparameters

SVM Hyperparameters

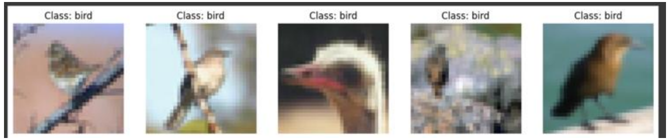
Parameter	Value
kernel	rbf
c	1.0
gamma	scale

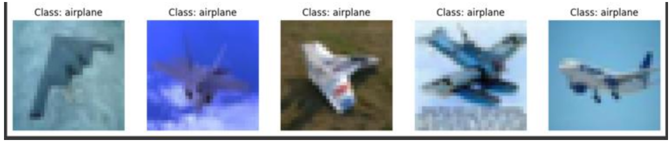


Plot train/test accuracy vs ONE hyperparameter of your choosing

CIFAR10 Dataset

Binary Classes





Class Description (-1)

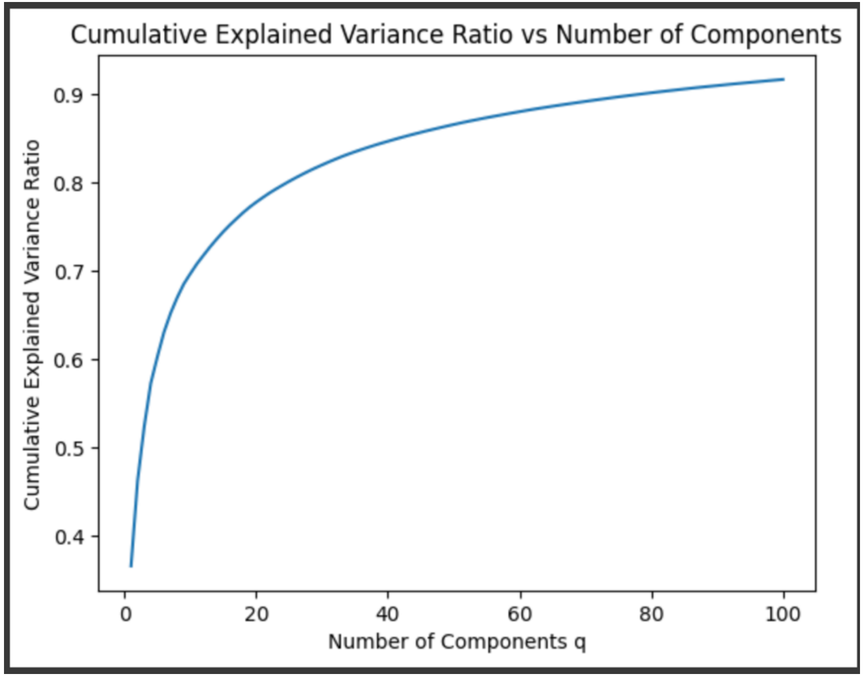
Class Description (+1)

PCA + Feed-forward NN

PCA + FNN Hyperparameters

Parameter	Range	Description	Why Tuned?
Number of Epochs	[1,100]	The number of times the model will iterate through the entire training dataset.	To control the number of training iterations and ensure model convergence
Batch size	[16,32,64,128]	The number of samples per batch processed during one iteration of training.	Affects training speed and convergence.
Learning rate	[0.0001, 0.001, 0.01, 0.1]	The rate at which the model updates its weights during training.	Important for model convergence
Hidden Layer Size	[10,50,100,200]	The number of neurons in the hidden layer of the neural network.	Controls the capacity of the model to learn complex patterns.
Input Dimension (PCA)	Fixed (100 components)	The number of primary components after applying PCA to the data.	Reduces dimensionality at the same time maintaining the most important information.
PCA Explained Variance	-	Cumulative variance ratio from PCA	Indicates how much information is maintained in the reduced space.
Weight initialization	Xavier, He, Uniform, Normal, Default	Used to initialize weights of a neural network	Affects model's convergence speed and stability
Activation Function	ReLU, Sigmoid, Tanh, Leaky ReLU, Softmax	Used to introduce non-linearity to the model by applying a non-linear function to the output of each neuron	Helps the network learn non-linear relationships.
Loss Function	MSELoss, CrossEntropyLoss, BCEWithLogitsLoss, etc.	Quantifies how well the model's prediction matches with the true labels	Defines how well the model is performing.
Optimizer	SGD, Adam, RMSprop, Adagrad, Adadelta	Adjusts model's weights during training	For better learning rate

PCA Variance Explanation

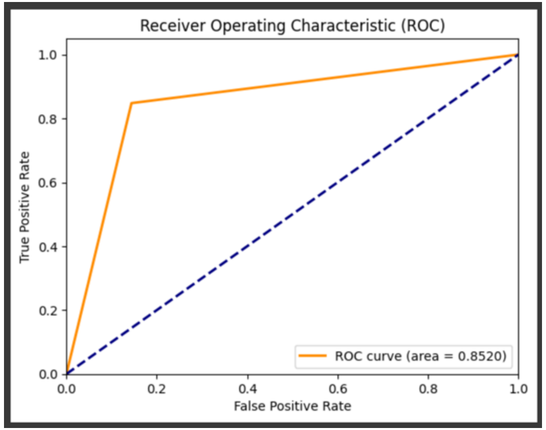


Plot the cumulative explained variance ratio calculated below as a function of *q*, where *s_i* is the *i*th singular value found via PCA. In this textbox, explain what you are plotting

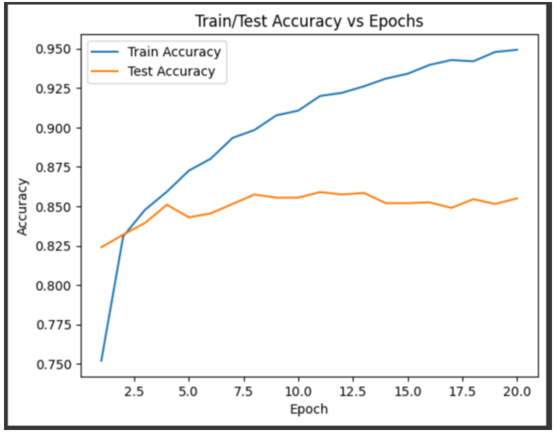
$$\sum_{i=1}^{q} \frac{s_i}{\sum_{j=1}^{d} s_j}$$

Results and Plots

Method	Accuracy	Per-class Accuracy (+1/-1)	AUROC
PCA + FNN	0.8505	Accuracy for class bird: 0.8470 Accuracy for class airplane: 0.8540	1.0



ROC Plot

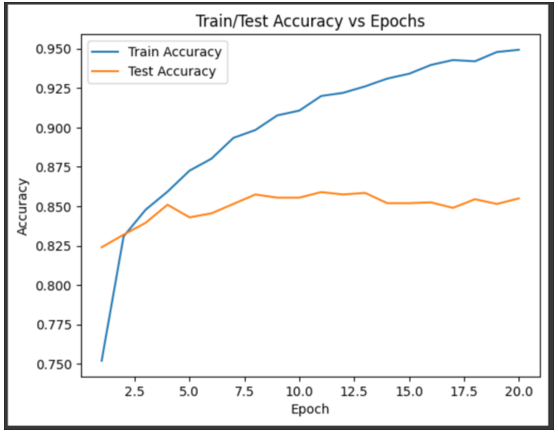


Training Epochs Plot

Final Hyperparameters

PCA + FNN Hyperparameters

Parameter	**Value**
Number of Epochs	20
Batch Size	64
Learning Rate	0.001
Hidden Layer Size	50
Input Dimensions (PCA)	100
PCA Explained Variance	Not directly chosen but dynamic
Weight Initialization	Default
Activation Function	ReLU
Loss Function	BCEWithLogitsLoss()
Optimizer	Adam



Plot train/test accuracy vs ONE hyperparameter of your choosing

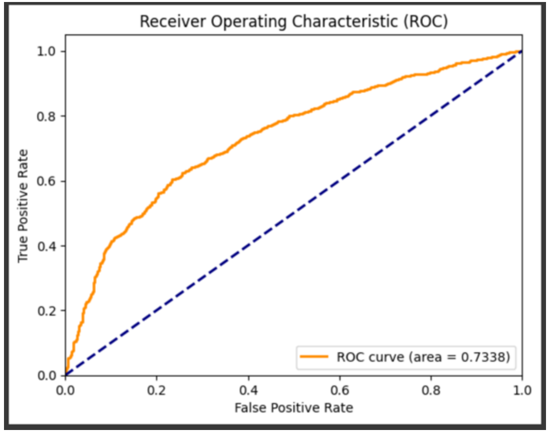
Convolutional Neural Network

CNN Hyperparameters

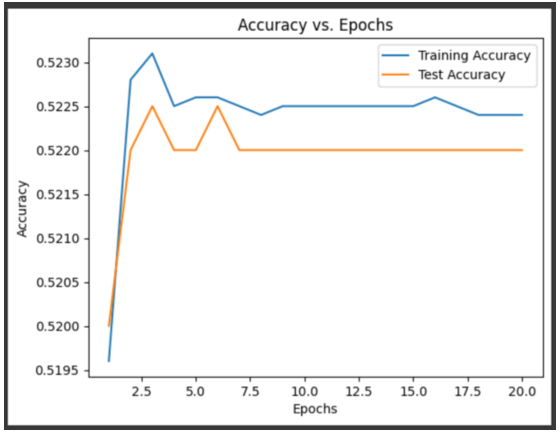
Parameter	Range	Description	Why tuned?
Learning rate	0.0001, 0.001, 0.01, 0.1	The step size at each iteration while moving toward a minimum of the loss function.	Tuning the learning rate is critical to balance convergence speed and stability.
Batch Size	[16,32,64,128]	The number of samples per batch processed during one iteration of training	Affects training speed and convergence
Epochs	[1,100]	The number of times the model will iterate through the entire training dataset.	To control the number of training iterations and ensure model convergence
Filter size (conv layers)	3x3, 5x5, 7x7	The size of the convolutional filters applied to the input image.	The filter size affects the receptive field and the number of parameters in the model.
Number of filters (conv layers)	32, 64, 128	The number of filters used in the convolutional layers.	To capture features
Pooling type	MaxPool, AveragePool	The type of pooling operation applied after convolutional layers.	To capture features
Pooling size	2x2, 3x3	Size of the pooling window	Affects the special resolution
Activation Function	ReLU, Sigmoid, Tanh, Leaky ReLU, Softmax	Used for non-linearity in hidden layers	To introduce non-linearity to networks
Weight initialization	Xavier, He, Uniform, Normal, Default	Used to initialize weights of a neural network	Affects model's convergence speed and stability
Optimizer	SGD, Adam, RMSprop, Adagrad, Adadelta	Adjusts model's weights during training	For better learning rate
Weight Decay	0.0001, 0.001, 0.01	Helps control model complexity	Prevents overfitting by penalizing large weights

Results and Plots

Method	Accuracy	Per-class Accuracy (+1/-1)	AUROC
CNN	Final Test Accuracy: 0.5220	Class -1 (airplane): 0.9920 Class 1 (bird): 0.0520	1.0



ROC Plot

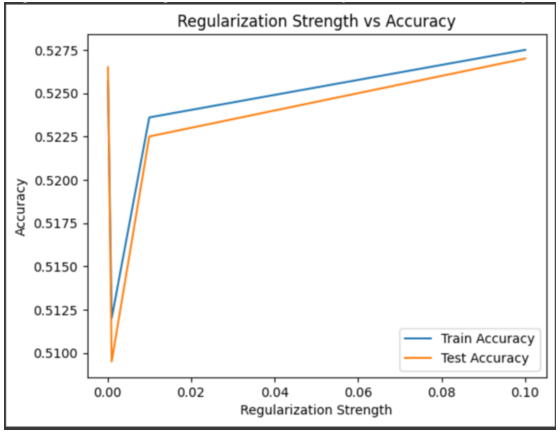


Training Epochs Plot

Final Hyperparameters

CNN Hyperparameters

Parameter	Value
Learning Rate	0.001
Batch Size	64
epochs	2
Filter Size	3x3
Number of filters	32, 64
Pooling type	MaxPool
Pooling Size	2x2
Activation function	ReLU
Weight Initialization	Default
L2 regularization	0.001
Optimizer	Adam
PCA Number of components	100
PCA Cumulative variance for Plotting	np.cumsum(pca_explained_variance_ratio_)



Plot train/test accuracy vs ONE hyperparameter of your choosing

Optional Explanations

Encountered that the CNN gives very less accuracy in predicting the images of bird whereas gives almost perfect accuracy for predicting the planes.

Because of this the confusion matrix is affected and this results in overall decrease in accuracy as compared to PCA+FNN

References

Google, Inc. *Google Colaboratory* [Online]. Available: https://colab.research.google.com.