Assignment 2
Pattern Recognition and Neural Network

Alokendu Mazumder (20134)* , Manan Tayal (20133)**, Piyush Tiwary (19897)†, and Saurabh Shrivastava (19881)‡

Email: {*alokendum, **manantayal, †piyushtiwary, ‡saurabhshriv}@iisc.ac.in

***Abstract*—We perform several experiments on binary, multi-class and time series dataset using the algorithms taught in lecture. Specifiaclly, in this assignment, we have implemented SVM, FLDA, MLP, CNNs from scratch and tested on relevant datasets. We have also used pytorch to implement some SOTA models like VGGNet, AlexNet and LSTMs.**

I. Introduction

In this assignment we cover experiments/investigations on SVM, back-propagation on various neural network models like vanilla MLP, RNN and CNNs. The reference codes (where-ever needed) is tagged.

II. SVM

The SVM objective function is defined as:

$$\frac{1}{2}\|x\|_2^2 + c\sum_{j=1}^{\infty} max[0, 1 - y_j(K(w, x_j) + b_j)] \tag{1}$$

Where $K(.)$ is a kernel function. In this part we choose a specific kernel and iterate multiple values of c and kernel parameters to find the optimal set. In general, we define a valid kernel $K(.)$ as follows:

$$K(x, y) = \phi(x)^T\phi(y) \tag{2}$$

Where, x and y are d dimensional vectors and ϕ: $\mathbb{R}^d \rightarrow \mathbb{R}^k$ and $k >> n$. We use the following kernels in our experiment:

$$\begin{gathered} K(x, y) = e^{\frac{\|x-y\|_2^2}{2\sigma^2}} \\ K(x, y) = (x^T y + 1)^d \\ K(x, y) = tanh(x^T y + 1) \end{gathered} \tag{3}$$

The first one is RBF kernel with parameter $\frac{1}{\sigma^2}$, second one is called polynomial kernel with parameters being it's degree (d) and last one is called sigmoid kernel. We grid search over their respective parameters and c, the regularization constant for binary and multi-class classification task.

We use pneumonia MNIST for binary task and blood MNIST for multi-class task. Blood MNIST has each image with dimension $(28, 28, 3)$ with total of 15,380 samples (includes both train and test). Pneumonia MNIST has each image with dimension $(28, 28, 1)$ with total of 5232 samples (includes both train and test). We do $0.9 - 0.1$ split for train and test data for both task. We feed-ed the images in SVM constraints by flattening them. The results are available at Table I and Table II.

III. Fischer Linear Discriminant

We have Used Fischer Linear Discriminant to Classify the Pneumonia MNIST data set, which is a Binary Classification Problem. Pneumonia MNIST has each image with dimension $(28, 28, 1)$ with total of 5232 samples (includes both train and test). We do $0.9 - 0.1$ split for train and test data for both task. We feed-ed the images in FLD constraints by flattening them. The experimental results are summarised in the Table VIII

IV. Convolutional Neural Networks

We have used Convolutional Neural Networks (CNNs) for classification task in the Blood-MNIST dataset. The dataset is divided in training and test data with a $70\% - 30\%$ split. All the architectures given below have the convolutional layer with filter size of 8×8, the total number of filters is fixed to 3. We experimented with 2 CNN structures, one with a shallow depth of one and the other with a deep depth of three. The two cases are shown in Table III and Table IV respectively. We have also tried to implement different regularization methods used in CNN like dropouts, Batch-Normalization and L_2 regularization. The results are shown in the respective tables, it can be seen that Batch-Normalization actually improves the result significantly.

V. State of the Art CNNs

We have deployed two State of the art CNNs - AlexNet and VGG for this part of the Assignment. The experiment was conducted on 2 different data sets Pneumonia MNIST (Binary Classification) and Blood MNIST (Multi Class Classification). Blood MNIST has each image with dimension $(28, 28, 3)$ with total of 15,380 samples (includes both train and test). Pneumonia MNIST has each image with dimension $(28, 28, 1)$ with total of 5232 samples (includes both train and test). We do $0.9 - 0.1$ split for train and test data for both task. The experimental results are summarised in the Table V, Table VI and Table VII.

VI. Multi-Layered Perceptrons

We implemented the MLPs from scratch without having to resort to the existing libraries available except numpy. We coded the back propagation algorithm in its entirety with the forward pass and the backward pass. Implemented various regularization techniques. Used multiple architectures and multiple regularization constants. We performed the above

Kernel	Optimal Regularization Parameter (c)	Optimal Parameter of respective Kernel	Training Accuracy	Test Accuracy
RBF	10	Std dev (γ) = 0.1	0.970	0.963
Polynomial	0.01	Degree = 4	0.968	0.961
Sigmoid	1	γ = 0.01	0.960	0.961

TABLE I: Results of grid search on hyper-parameters on different kernels on Binary Classification

Kernel	Optimal Regularization Parameter (c)	Optimal Parameter of respective Kernel	Test Accuracy
Sigmoid	0.001	γ = 0.001	0.436
RBF	0.001	γ = 0.01	0.839
Polynomial	0.001	5	0.753

TABLE II: Results of grid search on hyper-parameters on different kernels on Multi-Class Classification

Regularization:	Dropout	Batch Normalization	L_2 Regularization	None
Training Accuracy	0.710	0.843	0.710	0.740
Training Loss	0.895	0.904	0.789	0.774
Testing Accuracy	0.741	0.790	0.670	0.700
Testing Loss	1.067	0.906	0.911	0.871

TABLE III: Training and Testing accuracy and loss for different regularization methods when using a Shallow CNN

Regularization:	Dropout	Batch Normalization	L_2 Regularization	None
Training Accuracy	0.820	0.930	0.830	0.790
Training Loss	0.915	0.768	0.0.925	0.866
Testing Accuracy	0.780	0.893	0.780	0.761
Testing Loss	0.670	0.713	0.982	0.836

TABLE IV: Training and Testing accuracy and loss for different regularization methods when using a Deep CNN

	TP	TN	FP	FN	Precision	Recall	F1-Score	AUC	Accuracy
AlexNet	135	383	99	7	0.951	0.577	0.718	0.928	0.830
VGGNet	164	378	70	12	0.932	0.701	0.80	0.932	0.869

TABLE V: Binary Classification metrics of AlexNet and VGGNet on Pneumonia-MNIST dataset

Metrics	Class						
	0	**1**	**2**	**3**	**4**	**5**	**6**
Accuracy	0.799						
Precision	0.757	0.969	0.897	0.510	0.769	0.725	0.943
Recall	0.230	0.952	0.810	0.905	0.782	0.176	0.919
F1-Score	0.352	0.960	0.851	0.652	0.776	0.283	0.931
AUC	0.967						
TP	56	594	252	524	190	50	612
TN	3159	2778	3081	2338	3121	3118	2718
FP	188	30	59	55	53	234	54
FN	18	19	29	504	57	19	37

TABLE VI: Classification metrics of AlexNet on Blood-MNIST dataset

Metric	Class							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Accuracy	0.872							
Precision	0.717	0.954	0.845	0.745	0.770	0.771	0.965	0.996
Recall	0.623	0.997	0.929	0.760	0.909	0.592	0.940	0.996
F1-Score	0.667	0.975	0.885	0.752	0.834	0.669	0.952	0.996
AUC	0.983							
TP	152	622	289	440	221	168	626	468
TN	3117	2767	3057	2691	3112	3087	2732	2949
FP	92	2	22	139	22	116	40	2
FN	60	30	53	151	66	50	23	2

TABLE VII: Classification metrics of VGG-Net on Blood-MNIST dataset

	TP	TN	FP	FN	Precison	Recall	F1-Score	Accuracy
FLDA	123	381	111	9	0.932	0.526	0.672	0.807

TABLE VIII: Classification metrics for Binary Classification using FLDA

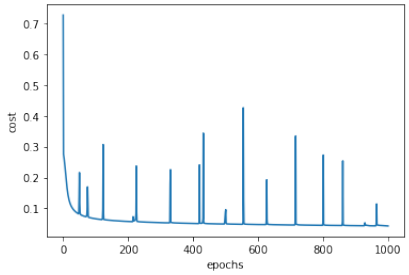
	Training Accuracy	Test Accuracy	Precision	Recall
1 layer LSTM	0.861	0.848	0.719	0.482
2 layer LSTM	0.890	0.863	0.782	0.441
4 layer LSTM	0.932	0.896	0.756	0.496

TABLE IX: Results of LSTM on TIMIT dataset for different depth

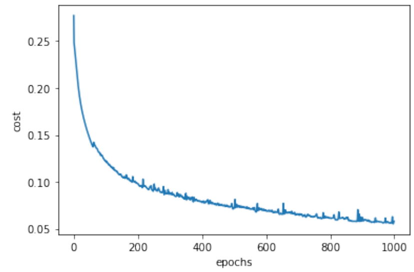
for the classification tasks, both for binary as well as for multi-class classification. The figures plotted are for the cost vs the epochs. Plot a is for binary classification with no regularisation with 50-nodes-intermediate layer. Plot b is for multiclass classification with dropout rate of 0.5 with 50-nodes-intermediate layer. Plot c is for multi-class classification with no regularisation with 50-nodes-intermediate layer. Plot d is for multi-class classification with dropout rate of 0.66 with 50-nodes-intermediate layer. Plot e is for binary classification with dropout rate of 0.4 with 50, 25 nodes in intermediate layers. Plot f is for binary classification with no regularisation with 50, 25 nodes in intermediate layers. Plot g is for multi-class classification with dropout rate of 0.5 with 50, 25 nodes in intermediate layers. Plot h is for multi-class classification with dropout rate of 0.33 with 50, 25 nodes in intermediate layers. We can clearly discern the role of regularisers and variegated architectures in the performances from the results.

VII. Long Short Term Memory

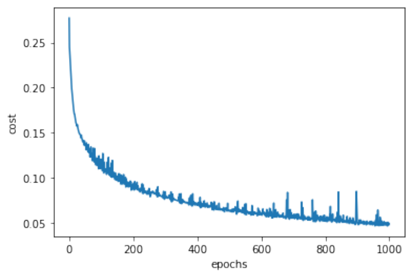
We employ a long short term memory network for "vowel prediction" task in the TIMIT dataseet. The train and test sets are obtained from the given dataset. We have directly used the pytorch code provided "here". We tried different number of layers in the LSTM for this task. The results by varying the number of layers is shown in Table IX. It can be seen that when the number of layers increases, the accuracy metrics also improves.



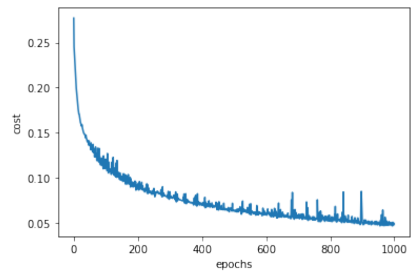
(a) Binary classification with no regularization



(b) Binary classification with regularization constant = 0.66

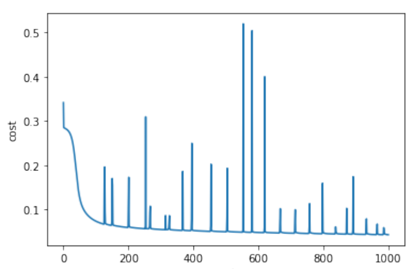


(c) Multi-class classification with no regularization

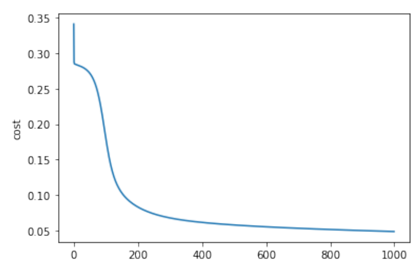


(d) Multi-class classification with regularization constant = 0.66

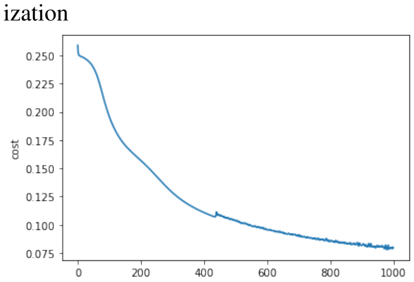
Fig. 1: Training with neural network with just one hidden layer with 50 nodes



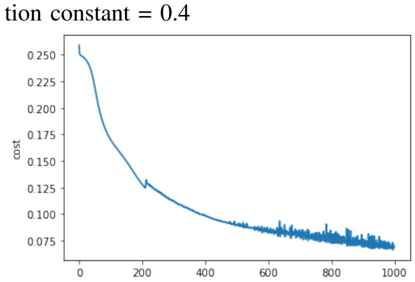
(a) Binary classification with no regularization



(b) Binary classification with regularization constant = 0.4



(c) Multi-class classification with no regularization



(d) Multi-class classification with regularization constant = 0.66

Fig. 2: Training with neural network with just two hidden layer with 50 nodes and 25 nodes respectively