Deep Learning based Classification of FDG-PET Data for Alzheimer's Disease Categories

Shibani Singh[a], Anant Srivastava[a], Liang Mi[a], Richard J Caselli[b], Eric Reiman[c], Kewei Chen[c], Dhruman Goradia[c], Yalin Wang[a]

[a] School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ
[b] Department of Neurology, Mayo Clinic, Scottsdale, AZ
[c] Banner Alzheimer's Institute, Phoenix, AZ

Table of Contents

Banner Alzheimer's Institute

Introduction

Banner Alzheimer's Institute

Alzheimer's Disease (AD)

- A progressive disease that worsens gradually with time
- No cure
- Drug & non-drug treatments may help
- Early detection is very difficult, but it can help:
    - Alter the course
    - Improve quality of life

Clinical Disease Stages

CU → EMCI → LMCI → AD

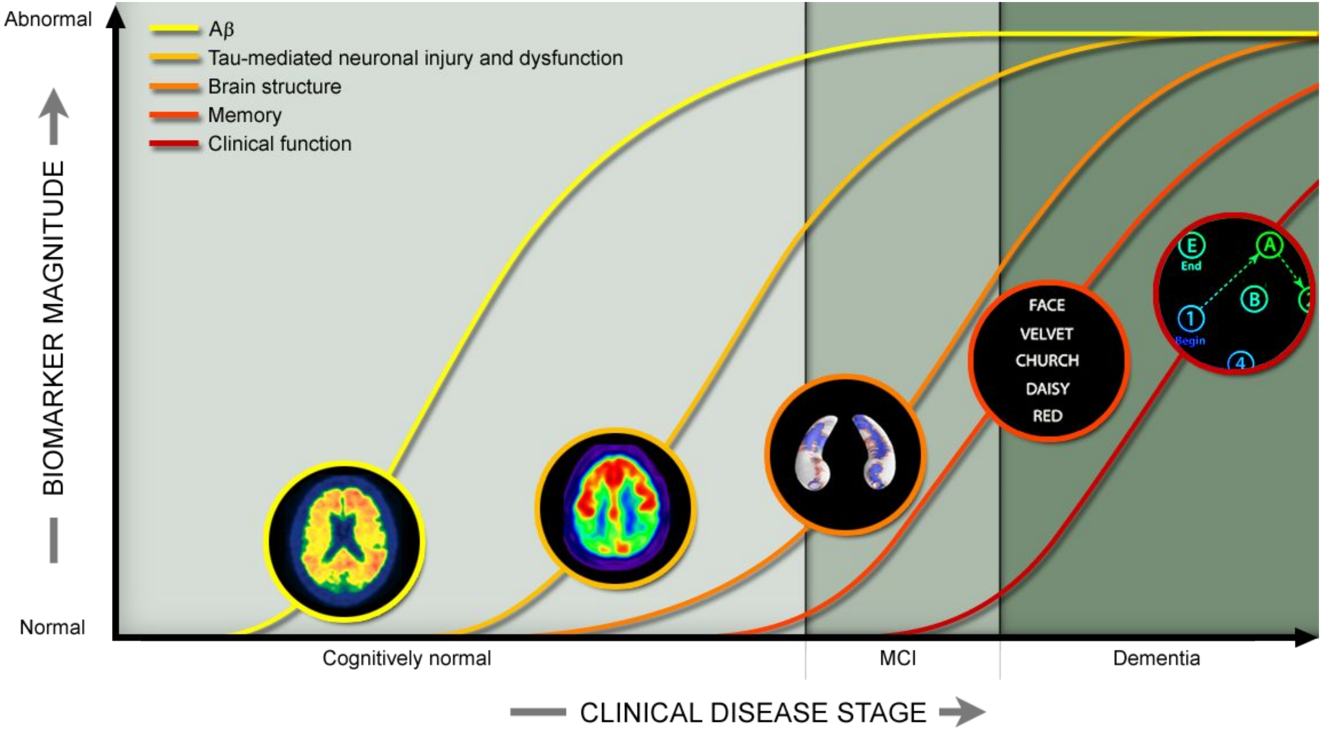
Timeline

- Cognitively Unimpaired/Normal (CU/CN):
  - No signs of impairment, depression or dementia
- Mild Cognitive Impairment (MCI):
  - Memory is impaired, but general cognitive function is preserved. Can be further categorised as:
    - Early Mild Cognitive Impairment (EMCI)
    - Late Mild Cognitive Impairment (LMCI)
- Alzheimer's Disease (AD) : Subjects meet a standard criteria for probable AD

Biomarkers

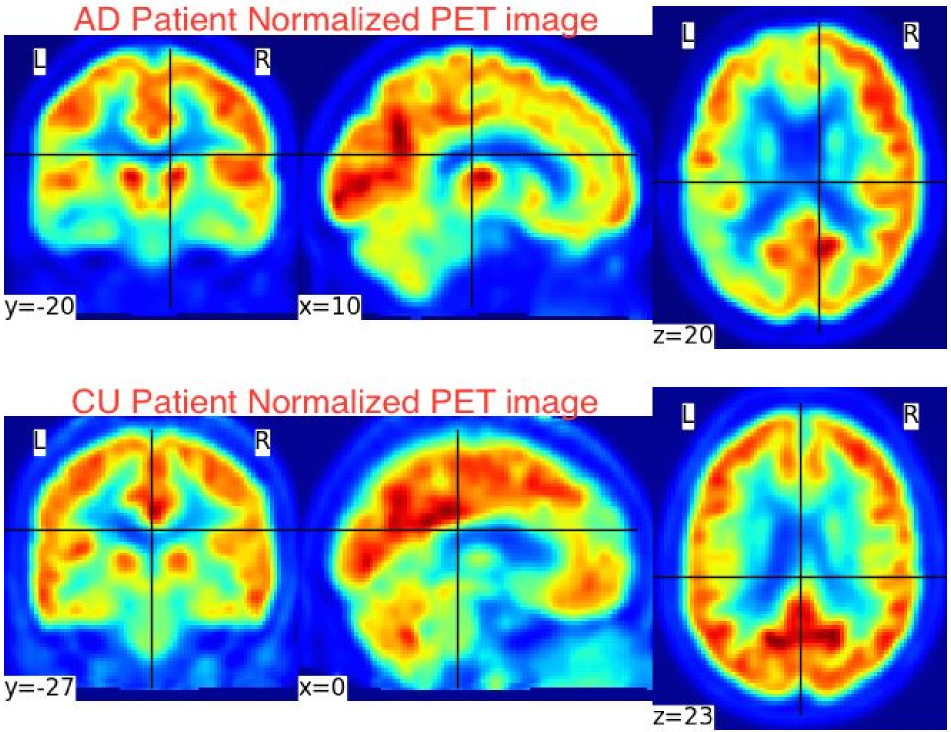
Banner Alzheimer's Institute

Biomarker measures for various clinical disease stages



[1] http://adni.loni.usc.edu/study-design/background-rationale/

Problem Background



[1]http://adni.loni.usc.edu/study-design/background-rationale/

Previous work

Previous work on AD:

- **Voxel wise analysis** to study group-wise differences and general trend in data.
- Statistically **significant pixels obtained in group difference study** may not convey strong predictions[1].

Deep Learning:

- Helped achieve state-of-the-art classification in Signal, Speech, Text & Image processing, and medical imaging.
- Feature representation using stacked auto-encoders (MRI and PET)
- Feature learning, representation and classification using MRI in combination with PET.

[1] Sun, D. et al., "Elucidating a magnetic resonance imaging-based neuroanatomic biomarker for psychosis: classification analysis using probabilistic brain atlas and machine learning algorithms", Biological psychiatry (2009)

Table of Contents

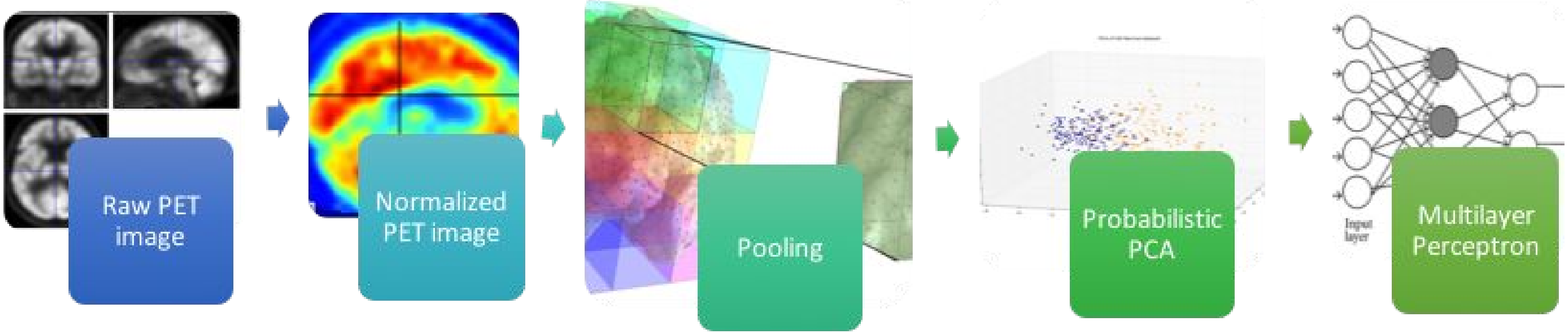
Banner Alzheimer's Institute

Data and Methods

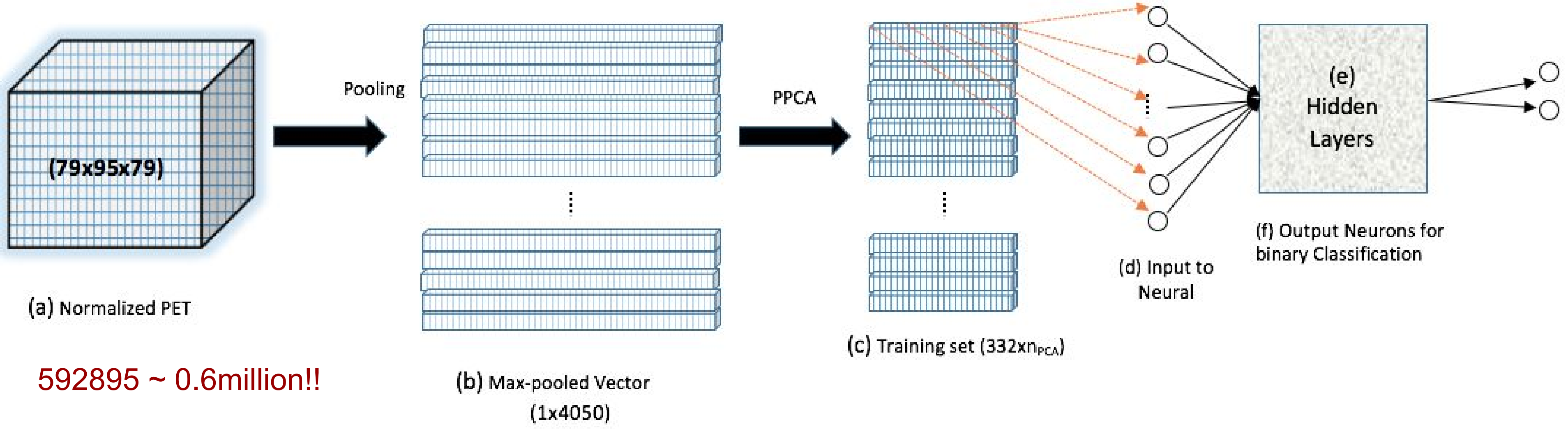
Banner Alzheimer's Institute

- Alzheimer's Disease Neuroimaging Initiative consists of these types of data:
  - MRI,
  - PET,
  - other biological markers,
  - clinical and neuropsychological assessment
- From ADNI-2, baseline visit subject data, we use:
  - FDG-PET data,
  - APOE gene information,
  - Age,
  - Gender,
  - Functional Assessment Questionnaire (FAQ) scores

System Architecture



Classification Pipeline

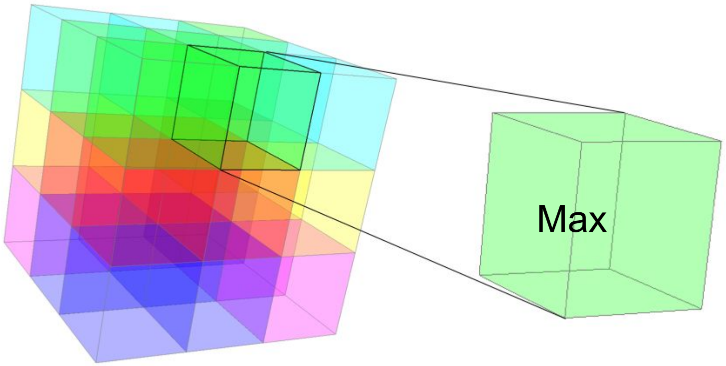


592895 ~ 0.6million!!

Pooling for 3-D cubic patches

Number of samples:

Group	Count
AD	146
CU	186
EMCI	178
LMCI	158

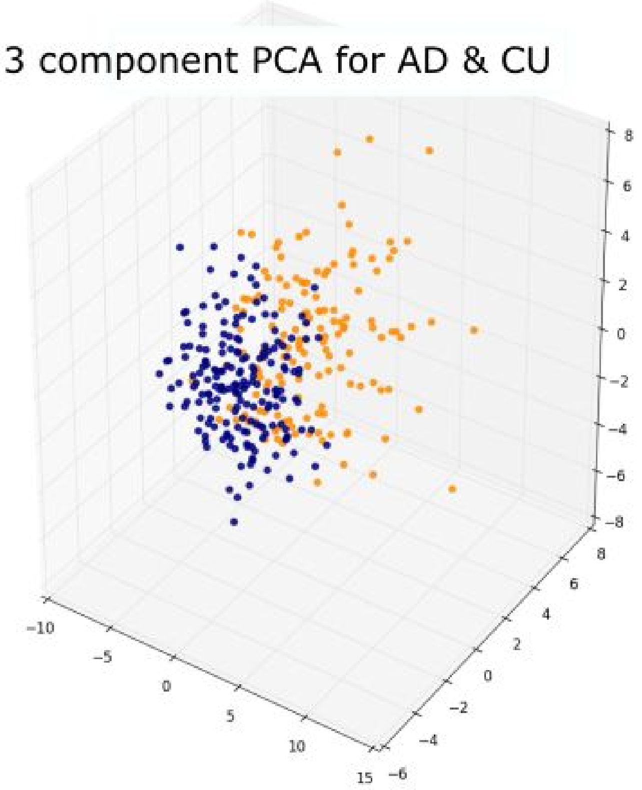


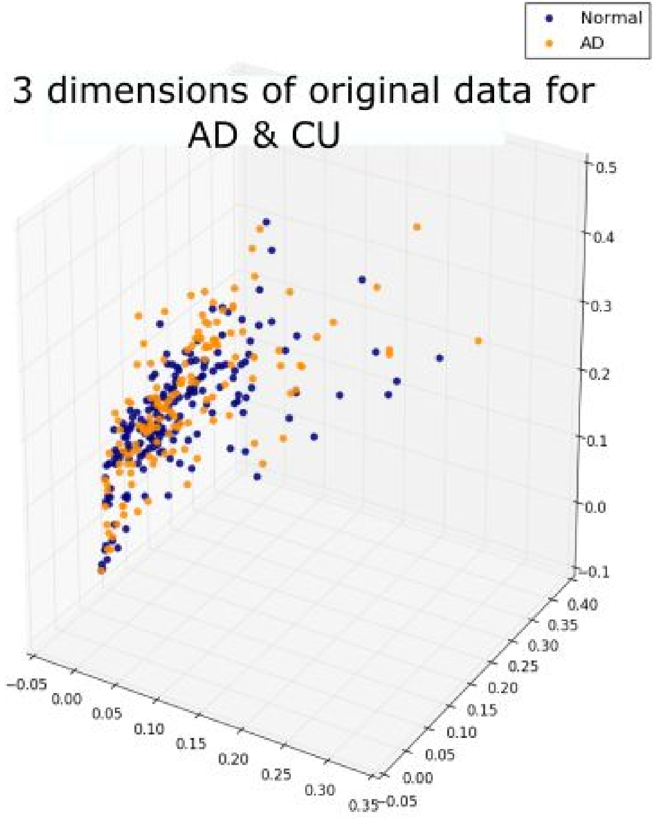
Max-pooling on a 3x3x3 patch

Number of features: 0.6 million!

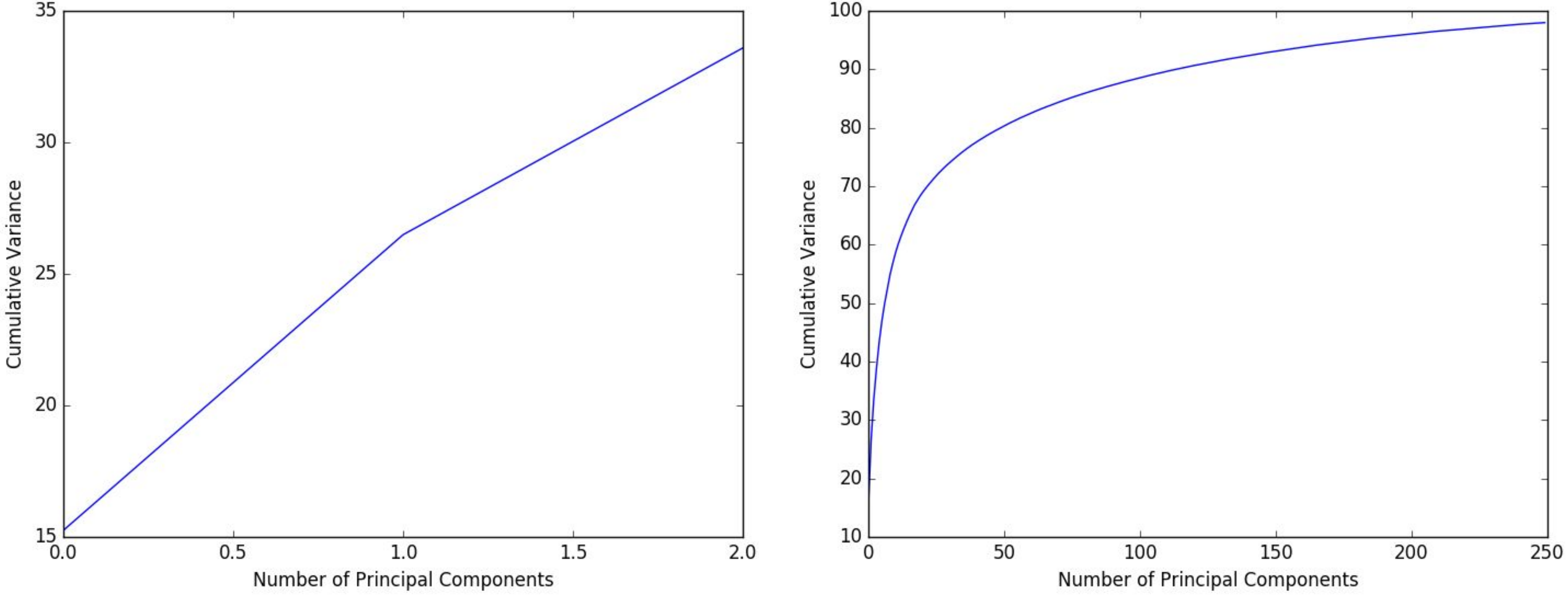
Patch Size	5	6	7	8	9	10	11	12	13	14	15
#Patches	66424	18750	18750	7942	7128	4050	2160	2016	1200	1200	891

Probabilistic PCA



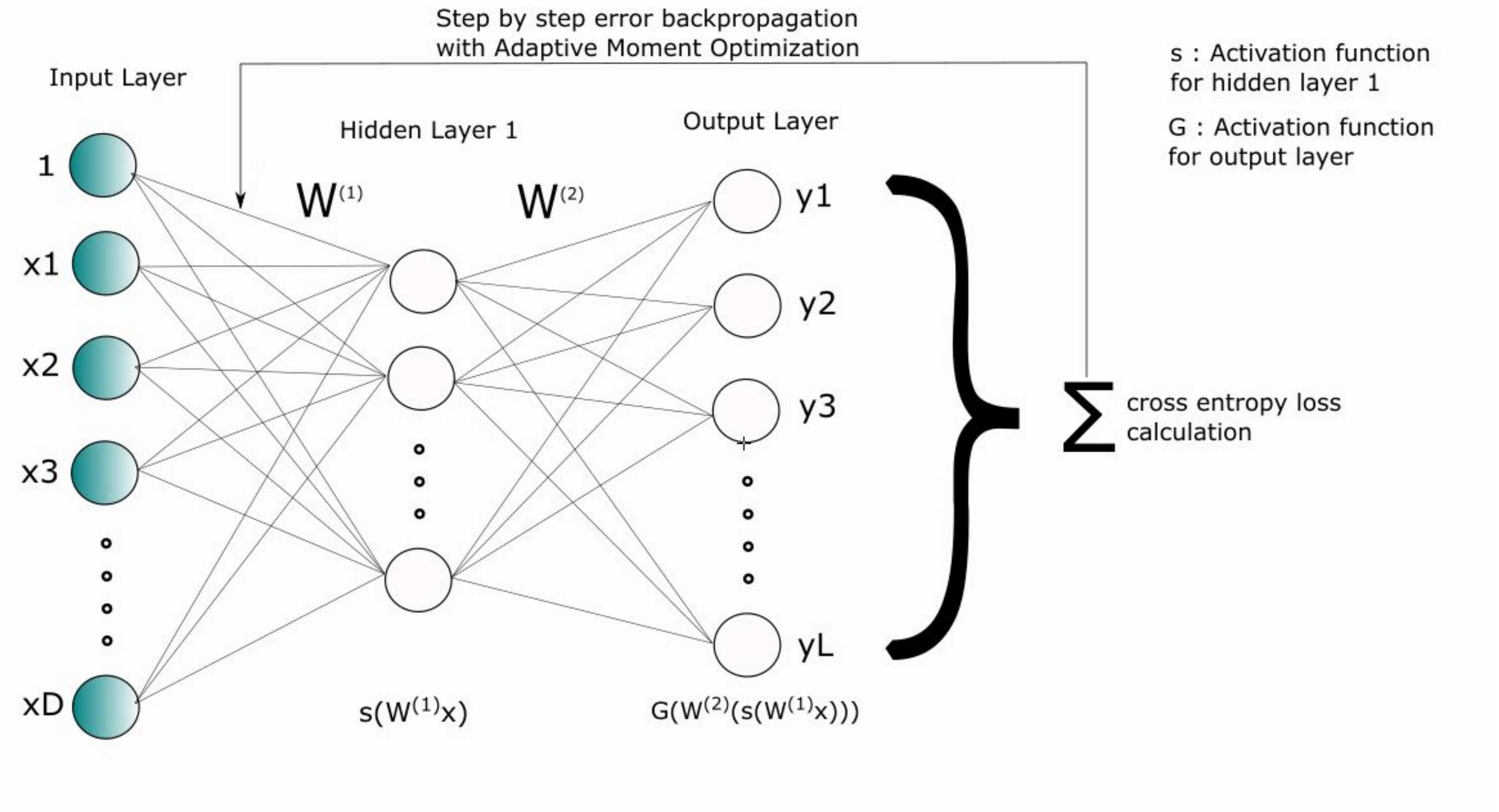


Probabilistic PCA

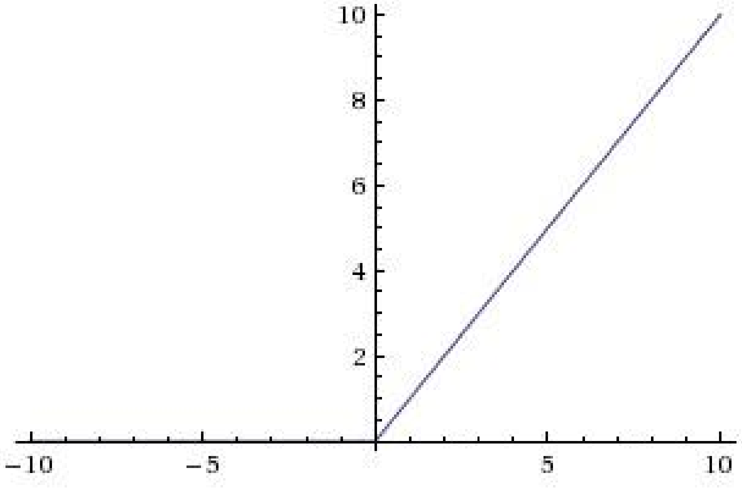


Increasing the number of Principal Components to 250-300, maximizes the variance to ~97-99%

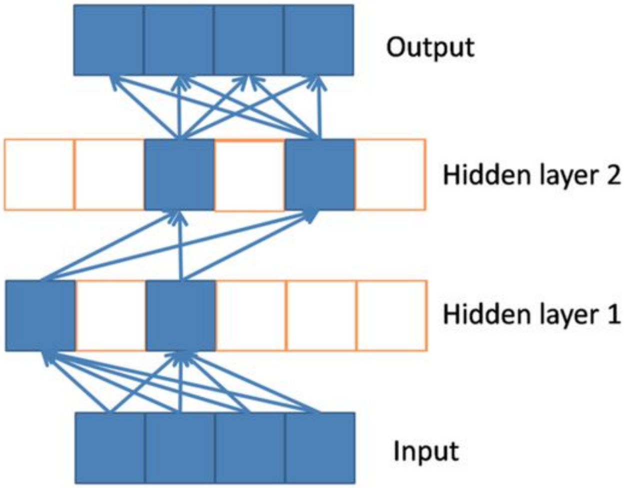
Multilayer Perceptron



Rectified Linear Unit (ReLu) Activation



$$f_{ReLu} = a_i(x) = \max(0, W_i^T x)$$



Rectified Linear Units induce sparsity in the neural network[1]

[1]Bengio, Y. et al., "Learning deep architectures for AI", Foundations and trends in Machine Learning (2009).

Backpropagation

- Forward Pass
  - To compute the output from each layer, which helps generate the error values
- Backward Pass
  - To backpropagate the error values in order to adjust the weights to be used for the forward pass again
  - Minimizes the error using Gradient Descent (Adaptive Moment estimates, which computes adaptive learning rates for each parameter)

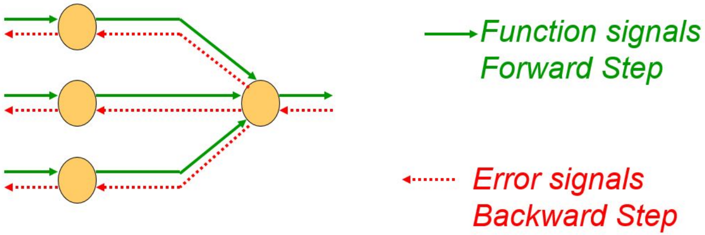


Table of Contents

Banner Alzheimer's Institute

Experimentation

- System Setup:
  - The experiments are performed on a node assigned on Saguaro.
  - Pooling operation performed using MATLAB
  - Scikit-Learn[1] used for Dimensionality Reduction (PPCA), Classification (MLP) and 10-fold cross validation.
- Assumptions:
  - No standalone technique based solely on FDG-PET exists to classify all clinical groups efficiently.
  - Class imbalance is low, and rarely leads to biased results.

Estimating Linear Separability of Data

Running Linear SVM on each of the experiments (Max-pooled data):

Linear SVM	AD/CU	AD/MCI	CU/MCI	EMCI/AD	LMCI/AD	EMCI/CU	LMCI/EMCI	LMCI/CU
F_1 Score	0.9291	0.8433	0.7082	0.8138	0.6882	0.6377	0.6260	0.6507

So how to separate non-linear data?

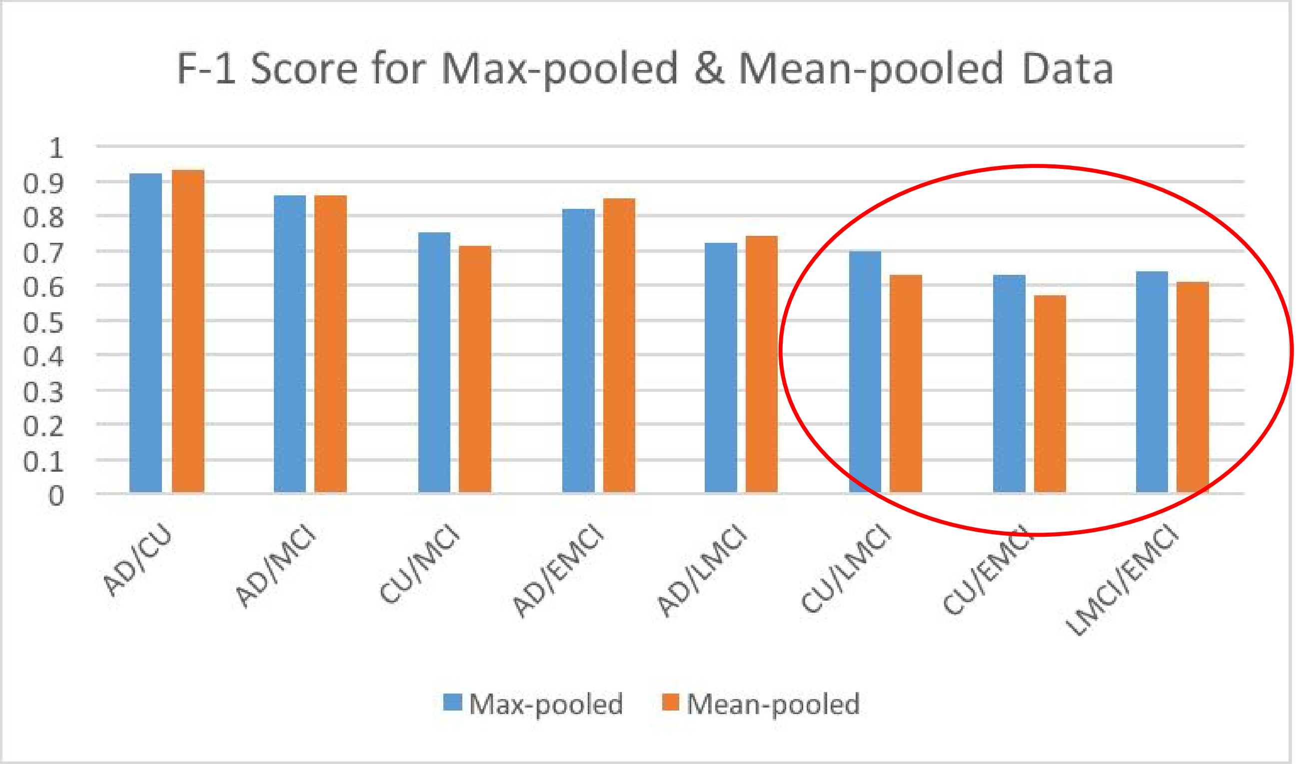
Estimating the number of Principal Components

Maximizing the cumulative variance of the Principal Components to 99% helps us choose the following number of Principal Components for each experiment:

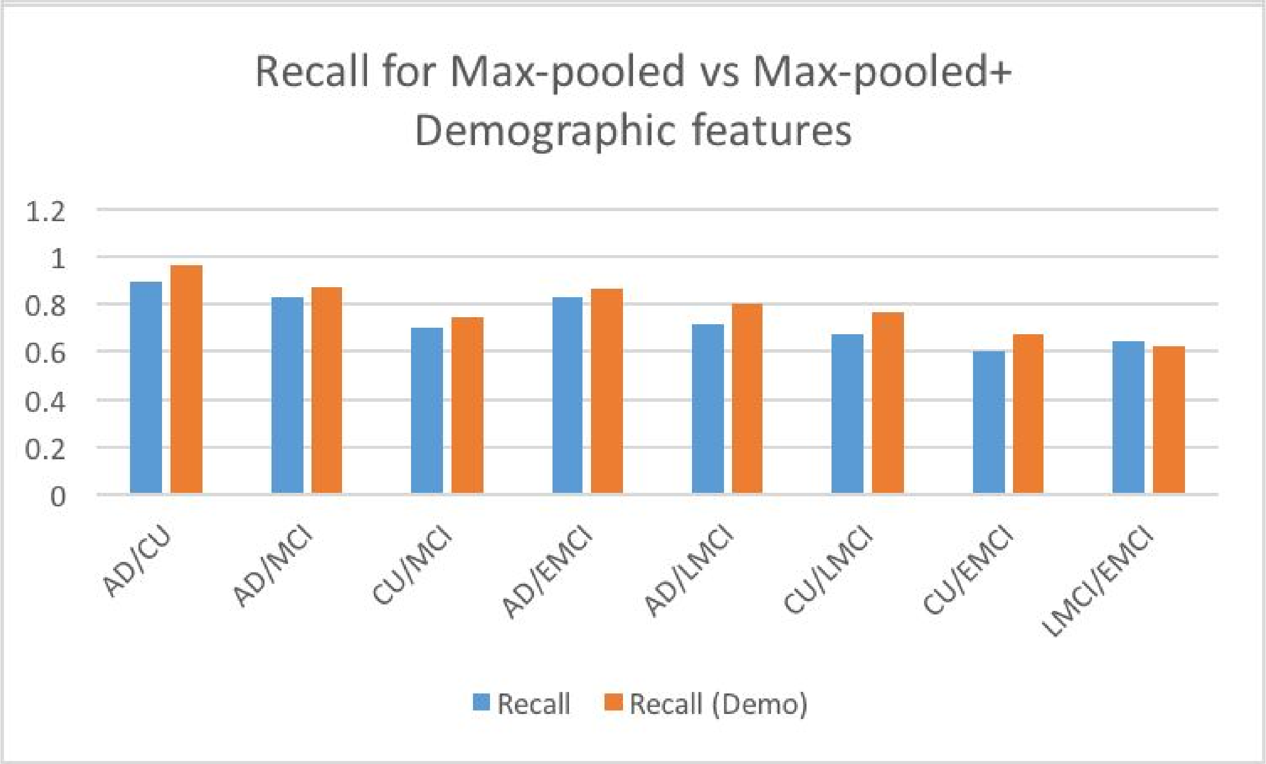
PPCA	AD/CU	AD/MCI	CU/MCI	EMCI/AD	LMCI/AD	EMCI/CU	LMCI/EMCI	LMCI/CU
#PCs	300	400	500	300	300	320	340	310

Max-pooled vs Mean-pooled data

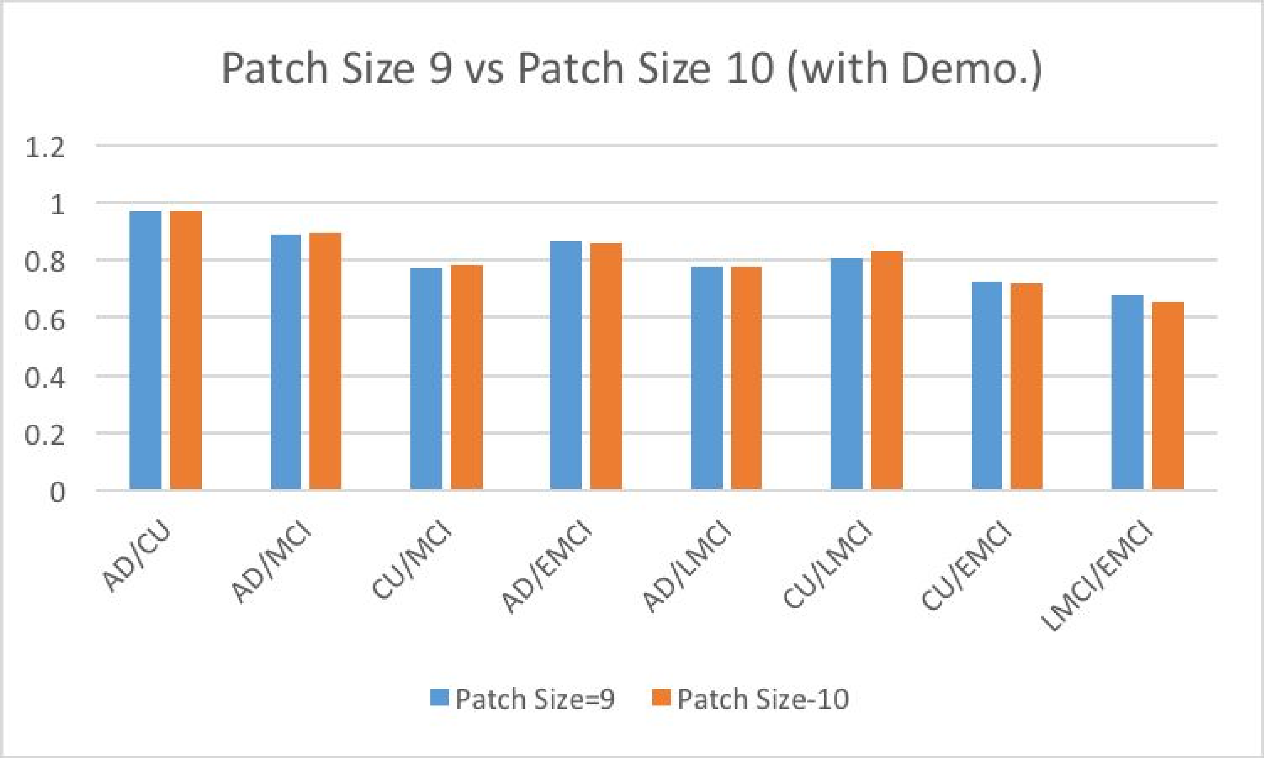
Banner Alzheimer's Institute



Effect of addition of Demographic features

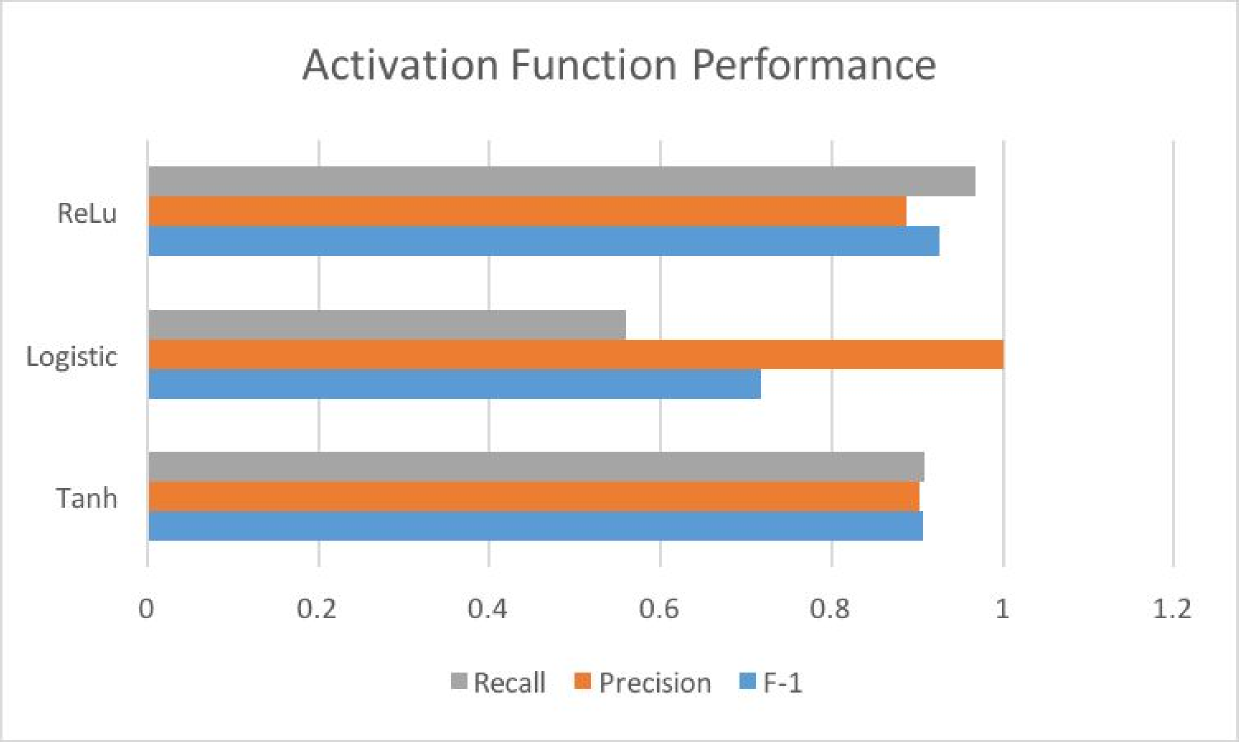


Finding a patch size

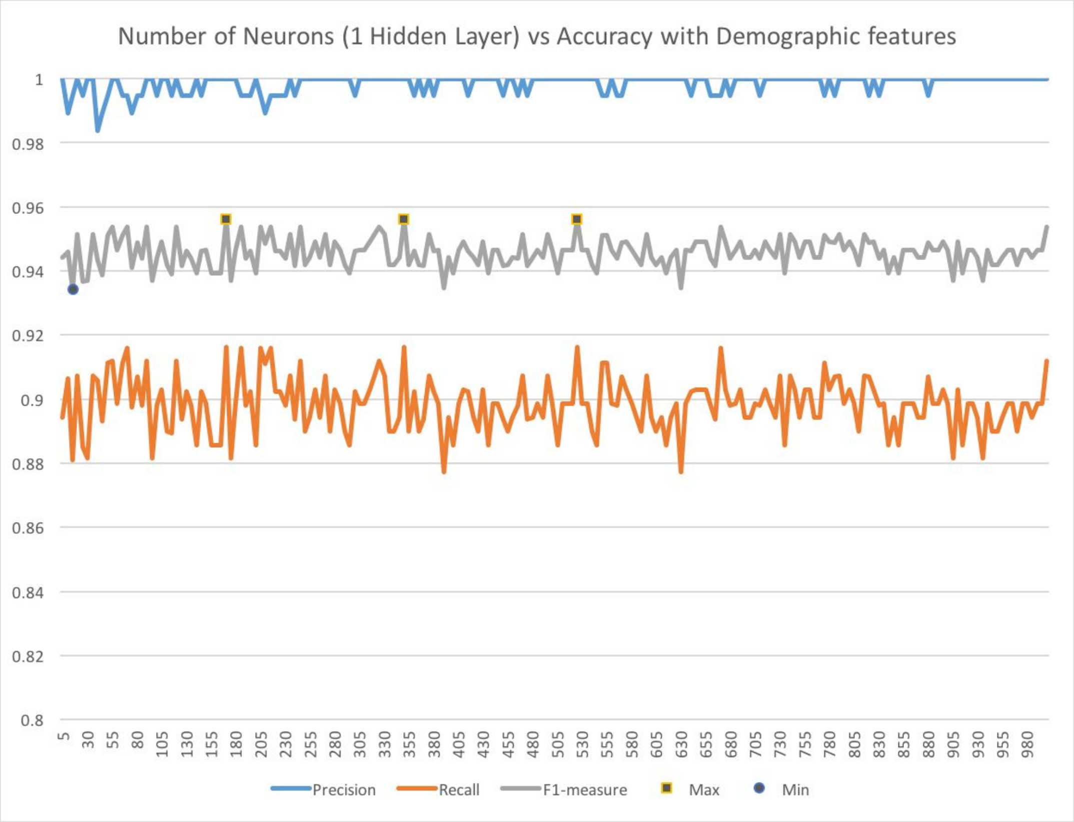


Choosing an Activation Function

Banner Alzheimer's Institute



Choosing the number of neurons for a 1 Hidden Layer MLP



Max at 170,350,525 neurons
Min at 15 neurons

Estimating the configuration for an n-hidden layer MLP

Banner Alzheimer's Institute

To automate the process for finding a better configuration and achieve better results.

```
function hiddenLayerConf = findHiddenLayerSizes(int n){

  hiddenLayerSizes = zeros(n)
  for(int i = 1; i <= n ;i++){
    maxF1=0
    maxNN=0
    for(nn = 5; nn <= 1000, nn += 5){
      hiddenLayerSizes[i] = nn
      model = MLPClassifier(hiddenLayerSizes, 'adam', 'relu')
      model.fit(trainKfold,labelKfold)
      f1Score = model.predict(testData)
      if (maxF1 < f1Score){
        maxF1 = f1Score
        maxNN = nn
      }
      hiddenLayerSizes[i] = maxNN
    }
  }
  return hiddenLayerSizes
}
```

Estimating the configuration for an n-hidden layer MLP

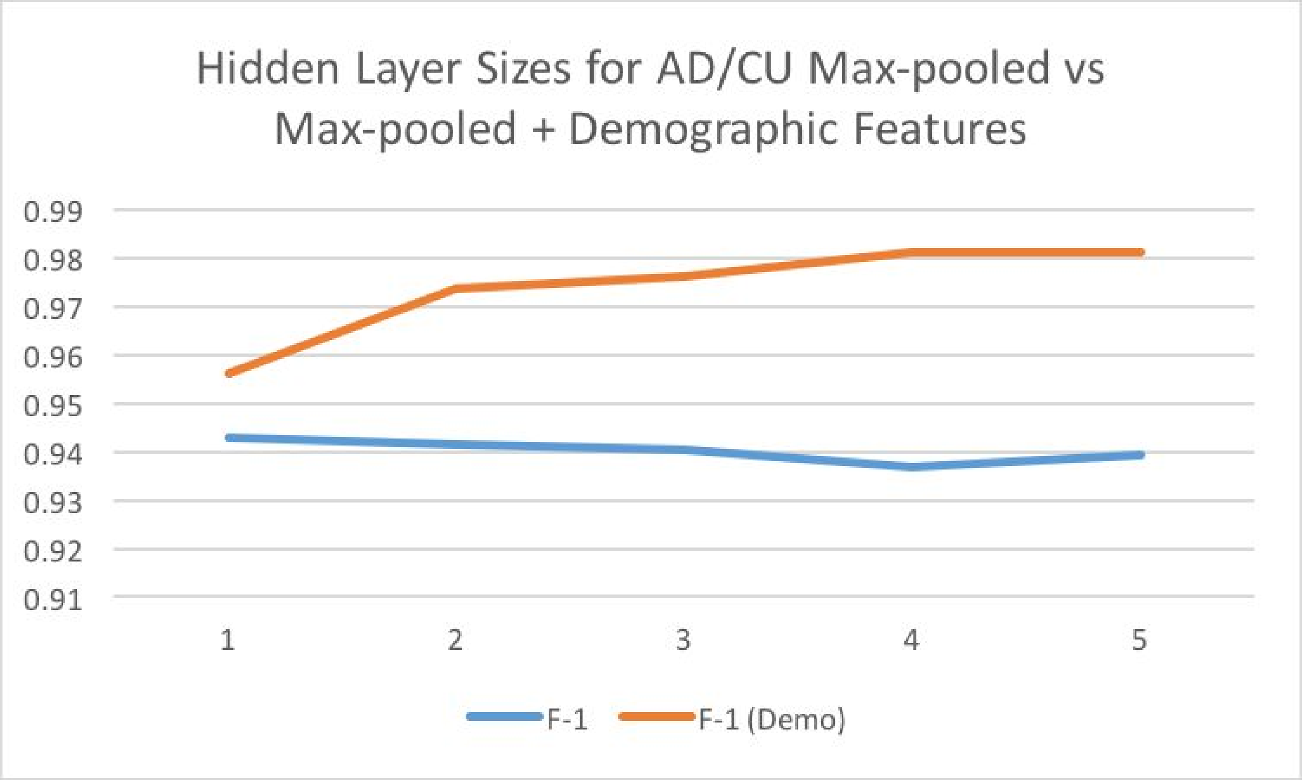
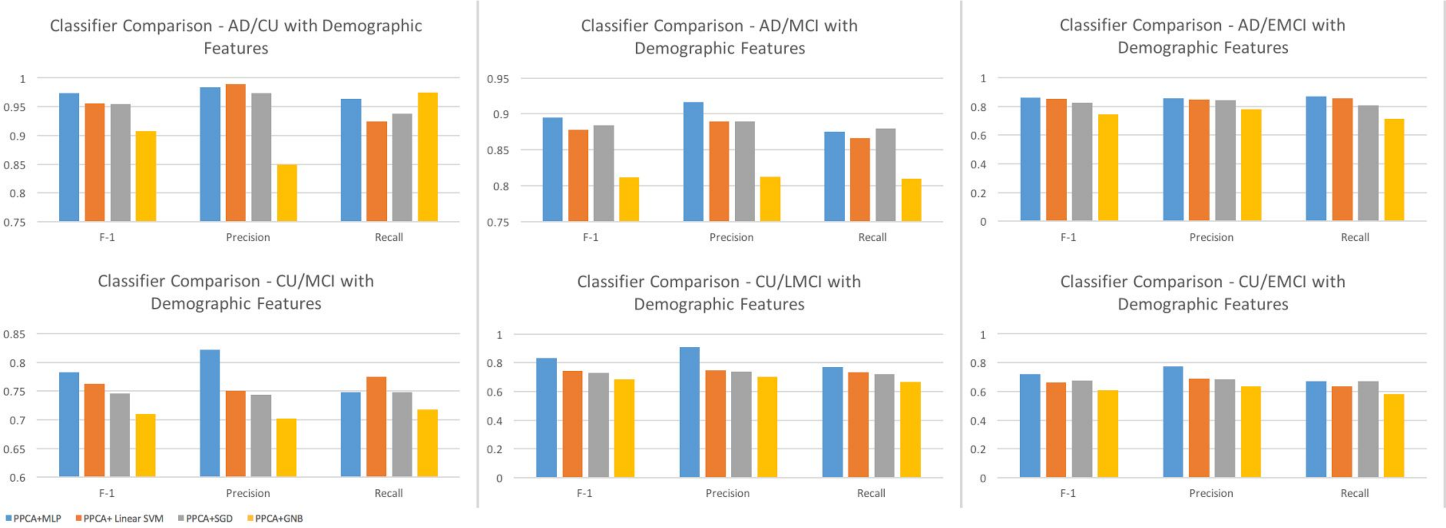


Table of Contents

Results - Comparing Classifier

Banner Alzheimer's Institute

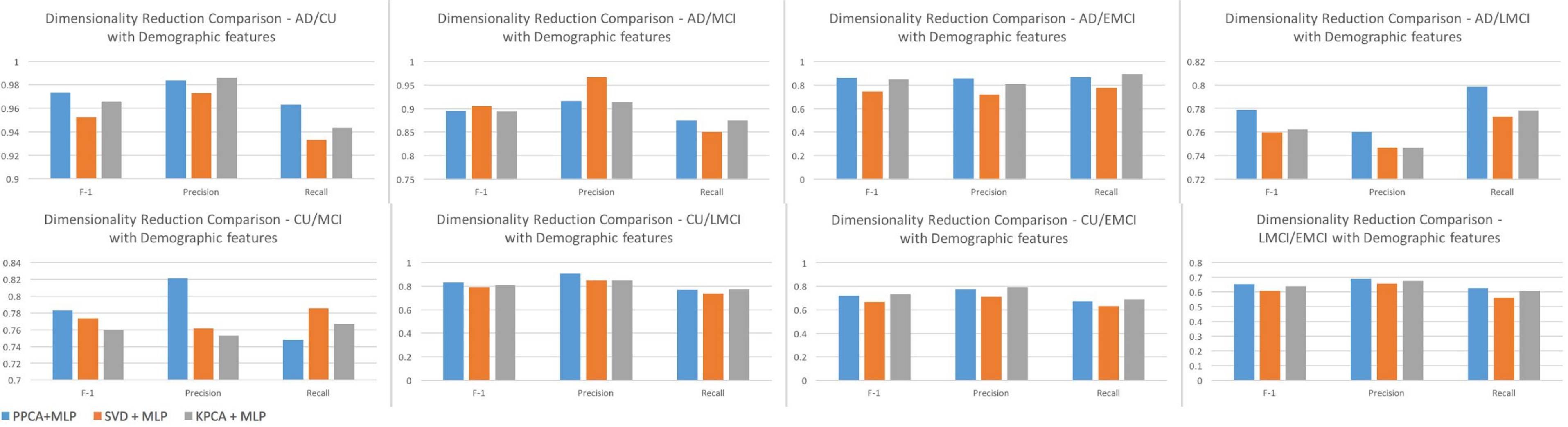
Comparison with 3 standard classifiers



- Linear SVM: Linear Support Vector Machine;
- GNB: Gaussian Naive Bayes
- SGD: Stochastic Gradient Descent Classifier;
- MLP: Multilayer Perceptron

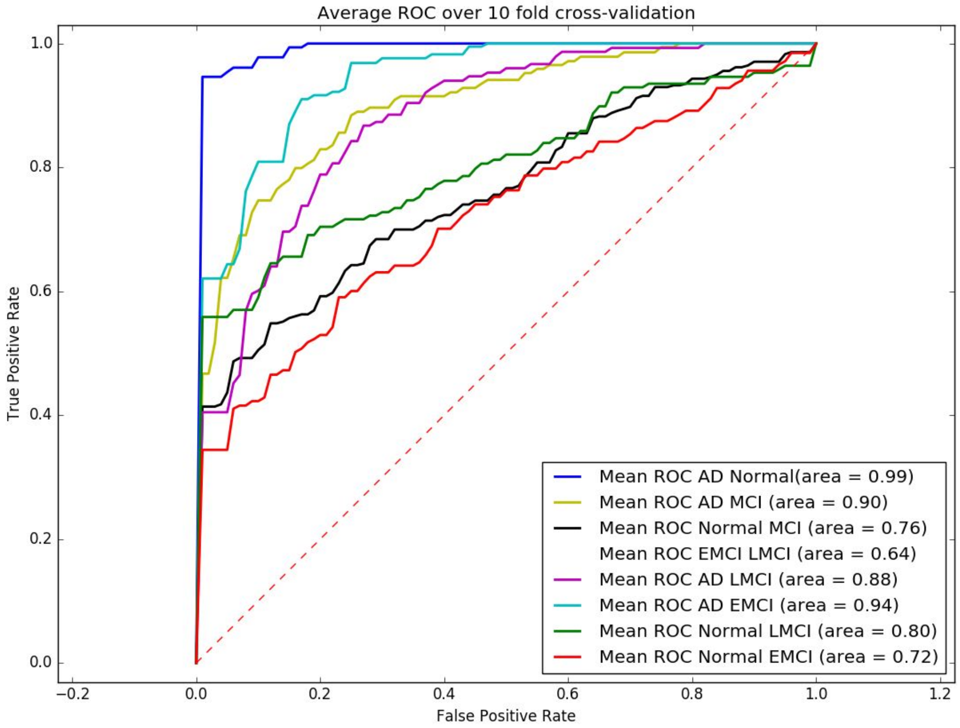
Results - Comparing PPCA

Comparison with other dimensionality reduction algorithms



- SVD: Singular Value Decomposition
- KPCA: Kernel Principal Component Analysis
- PPCA: Probabilistic Principal Component Analysis

Receiving Operator Characteristics



Summary of Results

Data	Measure	AD/CU	AD/MCI	CU/MCI	AD/EMCI	AD/LMCI	CU/LMCI	CU/EMCI	LMCI/EMCI
Max-pooled	F-1 Score	0.9430	0.8743	0.7527	0.8747	0.7706	0.6976	0.6388	**0.6844**
Max-pooled + Demo	F-1 Score	**0.9814**	**0.9125**	**0.7858**	**0.9036**	**0.8288**	**0.8325**	**0.72**	0.656

- The addition of demographic features significantly increases classification performance (except for LMCI/EMCI)
- MLP Classifiers show a general trend in outperforming other Machine Learning based classifiers.
- Probabilistic Principal Component Analysis outperforms other Dimensionality Reduction algorithms (except for AD/MCI where SVD outperformed).

Table of Contents

Banner Alzheimer's Institute

Conclusion

- The studied MLP classifier outperforms other machine learning classifiers.
- The algorithm (to automate parameter estimation) implemented improves our results.
- The addition of age, gender, gene information and FAQ scores significantly improves performance.

Contribution of Thesis work

Banner Alzheimer's Institute

- A **coherent and efficient deep learning framework** that well explores the possibility of FDG-PET for AD diagnosis.
- Work evaluated on a **relatively large dataset** and achieved competitive results.
- Exhibit the effective increase in classification performance with the **addition of demographic features** (age, gender, apoe(gene information) and FAQ scores)
- The developed **algorithm to automate parameter settings** proves useful in achieving better classification results.

Future Works

- According to the Universal Approximation Theorem ("One hidden Layer Multilayer Perceptrons are **universal approximators**"), so we should be able to find the best 1 hidden layer MLP, given the appropriate parameters.
- **Regional Analysis** using more complex architectures (such as Grid Long Short Term Memory (Grid LSTM), Parallel Multi-dimensional LSTM) could further enhance classification.
- **Region of Interest based analysis** may further improve performance.
- Use of **deep learning architectures for feature representation** (instead of PCA) could lead to more high level features. A whole system based on Neural Networks.
- The proposed framework could further be explored for Parkinson's and other diagnostic analysis purposes.

Publications

1. A published abstract for Arizona Alzheimer's Consortium 2017 Annual Scientific Conference: "Deep Learning based Classification of PET Imaging Data for Alzheimer's Diagnostic Categories"

Acknowledgment

I would like to thank my Professor, Dr. Yalin Wang for all the help throughout this project.

A big thank you to G.S.L Members especially Liang and Anant for always helping me out.

Questions?

Data Statistics

	Male	Female	Age	Min / Max Age	APOE1	APOE2	FAQ
AD	85	61	74.73 ± 8.15	56 / 90	3.11	3.63	13.39
LMCI	84	74	72.5 ± 7.5	55 / 91	3.03	3.54	03.62
EMCI	102	76	71.3 ± 7.2	55 / 89	2.94	3.42	02.08
CN	89	97	73.5 ± 6.25	57 / 89	2.86	3.24	00.16

Activation Function Comparison (AD/CU)

Table 4.7: Performance Comparison: Activation Function vs F1-Accuracy for AD/CU

Activation Function	F1	Precision	Recalll
Tanh	0.9057	0.9032	0.9081
Logistic	0.7181	1.0	0.5602
ReLu	**0.9255**	0.8867	0.9677

MLP Configuration (without use of the algorithm)

Table 4.6: MLP Configurations

Experiment	PCA components	Number of Hidden Layers	Hidden Layer Sizes	Optimization Algo	Activation
AD/CU	300	7	(1000,800,600,400,200,100,10)	Adam	ReLu
AD/MCI	400	4	(1000,500,100,10)	Adam	ReLu
CU/MCI	500	4	(1000,500,100,10)	Adam	ReLu
AD/EMCI	300	7	(1000,800,600,400,200,100,10)	Adam	ReLu
AD/LMCI	300	4	(1000,500,100,10)	Adam	ReLu
CU/LMCI	320	4	(1000,500,100,10)	Adam	ReLu
CU/EMCI	364	4	(1000,500,100,10)	Adam	ReLu
LMCI/EMCI	310	4	(1000,500,100,10)	Adam	ReLu

Algorithm Results for 5 hidden layer MLP - I

Table 4.8: Estimating an Optimal Configuration for AD CU Classification

	AD vs CU		AD vs CU with Demo	
#HL	Config	F_1 Score	Config	F_1 Score
1	(700)	**0.9430**	(525)	0.9563
2	(700,555)	0.9415	(525,880)	0.9737
3	(700,555,305)	0.9403	(525,880,880)	0.9761
4	(700,555,305,25)	0.9368	(525,880,880,255)	0.9812
5	(700,555,305,25,10)	0.9393	(525,880,880,255,775)	**0.9814**

Table 4.9: Estimating an Optimal Configuration for AD MCI Classification

	AD vs MCI		AD vs MCI with Demo	
#HL	Config	F_1 Score	Config	F_1 Score
1	(85)	0.8684	(160)	0.9086
2	(85,120)	0.8727	(160,270)	0.9125
3	(85,120,110)	**0.8743**	(160,270,405)	0.9083
4	(85,120,110,625)	0.8734	(160,270,405,350)	0.9086
5	(85,120,110,625,120)	0.8677	(160,270,405,350,215)	**0.9140**

Table 4.13: Estimating an Optimal Configuration for EMCI CU Classification

	EMCI vs CU		EMCI vs CU with Demo	
#HL	Config	F_1 Score	Config	F_1 Score
1	(5)	0.6044	(930)	0.6564
2	(5,25)	0.6192	(930,860)	0.6866
3	(5,25,5)	**0.6388**	(930,860,130)	0.6961
4	(5,25,5,185)	0.6287	(930,860,130,385)	**0.7015**
5	(5,25,5,185,5)	0.6293	(930,860,130,385,750)	0.6859

Table 4.14: Estimating an Optimal Configuration for EMCI LMCI Classification

	EMCI vs LMCI		EMCI vs LMCI with Demo	
#HL	Config	F_1 Score	Config	F_1 Score
1	(525)	0.6258	(125)	0.6214
2	(525,360)	**0.6275**	(125,585)	0.6205
3	(525,360,285)	0.6032	(125,585,5)	0.6467
4	(525,260,285,250)	0.5828	(125,585,5,430)	**0.6519**
5	(525,360,285,250,750)	0.6024	(125,585,5,125)	0.6103

Algorithm Results for 5 hidden layer MLP - II

Banner Alzheimer's Institute

Table 4.10: Estimating an Optimal Configuration for EMCI AD Classification

	EMCI vs AD		EMCI vs AD with Demo	
#HL	Config	F_1 Score	Config	F_1 Score
1	(755)	0.8696	(80)	0.9003
2	(755,55)	0.8736	(80,190)	0.9011
3	(755,55,625)	**0.8747**	(80,190,380)	**0.9036**
4	(755,55,625,15)	0.8641	(80,190,380,425)	0.9000
5	(755,55,625,15,585)	0.8644	(80,190,380,425,550)	0.8950

Table 4.11: Estimating an Optimal Configuration for LMCI AD Classification

	LMCI vs AD		LMCI vs AD with Demo	
#HL	Config	F_1 Score	Config	F_1 Score
1	(380)	0.7561	(215)	**0.8288**
2	(380,660)	**0.7706**	(215,105)	0.8193
3	(380,660,70)	0.7688	(215,105,150)	0.8098
4	(380,660,70,55)	0.7679	(215,105,150,600)	0.8086
5	(380,660,70,55,535)	0.7580	(215,105,150,600,260)	0.8086

Table 4.12: Estimating an Optimal Configuration for LMCI CU Classification

	LMCI vs CU		LMCI vs CU with Demo	
#HL	Config	F_1 Score	Config	F_1 Score
1	(915)	0.6512	(560)	0.7324
2	(915,20)	0.6536	(560,490)	**0.7774**
3	(915,20,690)	**0.6539**	(560,490,35)	0.7747
4	(915,20,690,170)	0.6471	(560,490,35,40)	0.7735
5	(915,20,690,170,15)	0.6507	(560,490,35,40,75)	0.7671

Max-pooled data vs Mean-pooled Data

Banner Alzheimer's Institute

Table 4.3: Classification comparison for Max-Pooled and Mean-Pooled Data

Performance Comparison	Pooling	AD / CU	AD / MCI	CU / MCI	AD / EMCI	AD / LMCI	CU / LMCI	CU / EMCI	LMCI /EMCI
F-1 score	Max	0.92	0.86	**0.75**	0.82	0.72	**0.70**	**0.63**	**0.64**
	Mean	**0.93**	0.86	0.71	**0.85**	**0.74**	0.63	0.57	0.61
Precision	Max	0.97	0.90	0.82	0.80	0.73	0.73	0.65	0.64
	Mean	0.96	0.89	0.73	0.87	0.77	0.62	0.59	0.59
Recall	Max	0.87	0.83	0.70	0.84	0.71	0.67	0.60	0.64
	Mean	0.90	0.83	0.70	0.82	0.71	0.65	0.55	0.63
NPV	Max	0.82	0.57	0.37	0.88	0.73	0.58	0.55	0.60
	Mean	0.87	0.58	0.42	0.77	0.66	0.72	0.54	0.70
PPV	Max	0.97	0.90	0.82	0.80	0.73	0.73	0.65	0.64
	Mean	0.96	0.89	0.73	0.87	0.77	0.62	0.60	0.59

MLP Configuration (without Algorithm use)

Table 4.6: MLP Configurations

Experiment	PCA components	Number of Hidden Layers	Hidden Layer Sizes	Optimization Algo	Activation
AD/CU	300	7	(1000,800,600,400,200,100,10)	Adam	ReLu
AD/MCI	400	4	(1000,500,100,10)	Adam	ReLu
CU/MCI	500	4	(1000,500,100,10)	Adam	ReLu
AD/EMCI	300	7	(1000,800,600,400,200,100,10)	Adam	ReLu
AD/LMCI	300	4	(1000,500,100,10)	Adam	ReLu
CU/LMCI	320	4	(1000,500,100,10)	Adam	ReLu
CU/EMCI	364	4	(1000,500,100,10)	Adam	ReLu
LMCI/EMCI	310	4	(1000,500,100,10)	Adam	ReLu

Patch Size 9 vs Patch size 10

Table 4.5: Performance Comparison: Patch Size 9 vs Patch Size 10

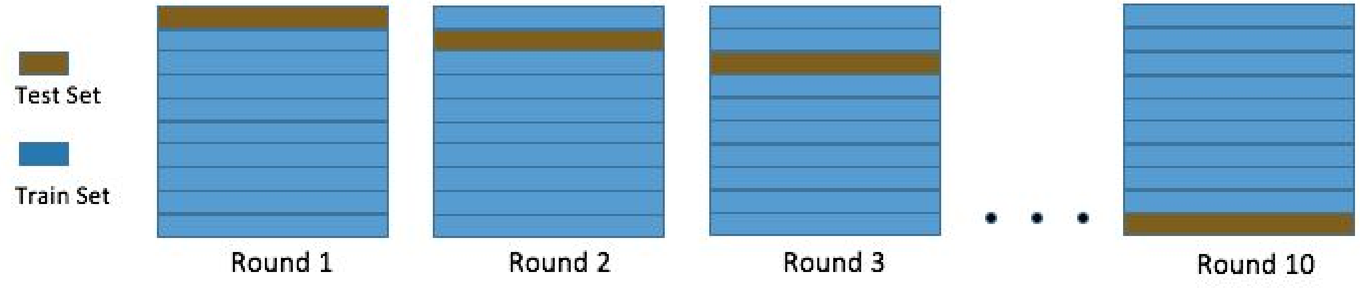
Patch Size	Measure	AD / CU	AD / MCI	CU / MCI	AD / EMCI	AD / LMCI	CU / LMCI	CU / EMCI	LMCI /EMCI
	F_1 score	**0.9737**	0.8899	0.7713	**0.8632**	0.7778	0.8060	**0.7246**	**0.6809**
$p_{size} = 9$	Precision	0.9928	0.9137	0.7679	0.8425	0.7671	0.8710	0.7850	0.7191
	Recall	0.9536	0.8672	0.7748	0.8849	0.7887	0.75	0.6728	0.6465
	F_1 score	0.9734	**0.8953**	**0.7830**	0.8621	**0.7789**	**0.8325**	0.72	0.656
$p_{size} = 10$	Precision	0.9839	0.9167	0.8214	0.8562	0.7603	0.9086	0.7742	0.6910
	Recall	0.9632	0.8750	0.7480	0.8681	0.7986	0.7682	0.6729	0.6244

Comparison of Patch Sizes (AD/CU)

Table 4.4: Performance Comparison: Patch Size vs F1-Accuracy

	Without	Demographics		With	Demographics	
Patch Size (n)	F1	Precision	Recall	F1	Precision	Recall
5	**0.9312**	0.9462	0.9167	0.9474	0.9677	0.9278
6	0.9175	0.9570	0.8812	0.9708	0.9839	0.9581
7	0.9044	0.9409	0.8706	0.9710	0.9892	0.9534
8	0.9271	0.9570	0.8990	0.9735	0.9892	0.9583
9	0.9231	0.9677	0.8824	**0.9737**	0.9946	9536
10	0.9255	0.8867	0.9677	0.9735	0.9839	0.9632
11	0.9251	0.9624	0.8905	0.9661	0.9946	0.9391
12	0.9299	0.9624	0.8995	0.9661	0.9946	0.9391
13	0.8144	0.7822	0.8495	0.9561	0.9946	0.9204
14	0.9199	0.9570	0.8856	0.9634	0.9892	0.9388
15	0.9133	0.9624	0.8689	0.9609	0.9892	0.9340

10-fold Cross Validation



Confusion Matrices M_1 M_2 M_3 M_{10}

Final Confusion Matrix is given by:

$$M = \sum_{i=1}^{N} M_i$$

Performance Metrics

- Confusion Matrix:

		Predicted class	
		Class 1	Class 0
Actual class	Class 1	true positives (TP)	false negatives (FN)
	Class 0	false positives (FP)	true negatives (TN)

- Precision $\frac{TP}{TP+FP}$
- Recall $\frac{TP}{TP+FN}$
- F_1 Score $\frac{2TP}{2TP+FP+FN}$