

Classification using ML methods and Neural Networks

by Xinzhe Lyu

Submission date: 30-Mar-2023 12:12PM (UTC+0100)

Submission ID: 201005810

File name:

40633_Xinzhe_Lyu_Classification_using_ML_methods_and_Neural_Networks_495066_579699304.docx
(384.87K)

Word count: 2844

Character count: 14896



Classification using ML methods and Neural Networks

Student name: Xinzhe Lyu

Student code: 2214776

Programme: MSc Software Engineering and AI

Moudle: Application of Machine Learning

Moudle code: ACCA7025

Lecturer: Gordon Dickers

Abstract

After systematic learning last semester and using CNNs and decision trees to identify and classify the Fashion-MNIST dataset, I have a good understanding of machine learning and multi-class classification. In this paper, random forest method and Multilayer Perceptron are used to classify the Fashion-MNIST dataset and Cifar-10 dataset, and the optimal effect is achieved by adjusting the hyperparameters, so as to have a deeper understanding of the use of the algorithm itself and the characteristics of the algorithm, as well as the applicability to the dataset. By comparing the accuracy and other metrics, let us look at the pros and cons of the two algorithms to better understand the machine learning application.

Introduction

In this paper, random forest and multilayer perceptron algorithms are used to find relatively better algorithmic models by tuning hyperparameters for Fashion-MNIST dataset and Cifar-10 dataset. In this process, the optimal solution of each algorithm is obtained by controlling the variables in such a way that the meaning of each hyperparameter is understood, and what results would result from modifying its parameters, and then the advantages, disadvantages, and common features of the two algorithms are summarized.

Related works

In this work, random forest classification and Multilayer Perceptron methods are used to classify the Fashion-MNIST dataset and the Cifar-10 dataset. The sklearn class in Python was used to call the interfaces of the above two methods, adjust the hyperparameters, and train the above two datasets. To obtain the optimal solution for each algorithm, a horizontal comparison of the four optimal solutions of the two algorithms for the two datasets is performed to conclude.

Methodology

The manually downloaded Fashion-MNIST dataset and the Cifar-10 dataset from keras.datasets are applied in this study. We also use the sklearn library in Python, which

has a large collection of machine learning related basic training models and validation methods. In this study, random forest classification algorithms from the sklearn library, Multilayer Perceptron algorithms and confusion matrix are used to visualize the performance of the model.

Fasion-MNIST dataset

The Fashion-MNIST dataset is a dataset of clothing images provided by the German company Zalando, which contains a training set of 60, 000 samples and a test set of 10, 000 samples. Each sample is a 28x28 grayscale image associated with one of the ten class labels. The Fashion-MNIST dataset is intended as a direct alternative to the original MNIST dataset for benchmarking machine learning algorithms.

Label	Description
0	T-shirt
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Table 1. Fashion-MNIST dataset Label and Description

Cifar-10 dataset

The CIFAR10 dataset has a total of 60,000 samples, each sample is a 32*32 pixel RGB image (color image), and each RGB image must be divided into three channels (R channel, G channel, B channel). These 60, 000 samples are divided into 50, 000 training samples and 10, 000 test samples.

Label	Description
0	Airplane
1	Automobile
2	Bird
3	Cat
4	Deer
5	Dog
6	Frog
7	Horse
8	Ship
9	Truck

Table 2. Cifar-10 dataset Label and Description

Sklearn

Scikit-learn is an open source python-based machine learning kit. It implements efficient algorithms using python's numerical libraries such as NumPy, SciPy, and Matplotlib, covering most of the major machine learning algorithms.

For engineering applications, handwritten python code is highly unlikely to be able to implement an algorithm from scratch, which is not only time-consuming, but also not necessarily a framework or a stable model. In many cases, by analyzing the collected data, choosing the appropriate algorithm based on the characteristics of the data, calling the algorithm in the toolkit, adjusting the algorithm parameters, and obtaining the necessary information, we can improve the efficiency of the algorithm and improve the efficiency of the algorithm. Balance the effect. Sklearn is a toolkit that helps you implement algorithms efficiently.

Random forest

Random forest is an algorithm that integrates multiple trees based on the concept of ensemble learning, which in turn is based on decision trees, but is essentially a method of ensemble learning, which is an area of machine learning. The name random

forest has two keywords: "random" and "forest". The word "forest" is well understood, but if one tree is a tree, if there are hundreds or thousands of trees, it is a forest, and this is also a manifestation of the idea of "accumulation" which is the concept of random forest.

From an intuitive point of view, each decision tree is a classifier, then for an input sample, N trees will have N classification results.[1] Random forest, on the other hand, is a simple idea that combines the results of all categories and specifies the category with the most votes as the final output.

Multilayer Perceptron

Multilayer perceptron (MLP) is an artificial neural network with a positive structure. MLP can handle nonlinear separable problems.

MLP is an artificial neural network with a feed-forward structure that maps a set of input vectors to a set of output vectors. An MLP can be thought of as a directed graph consisting of node layers, each of which is completely connected to the next layer.

The MLP is trained by supervised learning using the BP propagation algorithm. MLP is a generalization of sensing machines that overcomes the bug of not being able to identify linear non-fractional data.

Compared to a single layer of detectors, MLP's multilayer detectors convert the output from one layer to multiple layers. Instead of a single layer between input and output, there are two layers: output layer and hidden layer.

In a typical feed-forward neural network, the information is transmitted forward from input layer to hidden layer and then from hidden layer to output layer.[2] The hidden layer implements a nonlinear mapping of the input space and the output layer implements a linear classification. It is possible to learn both nonlinear mapping patterns and linear discriminant functions.

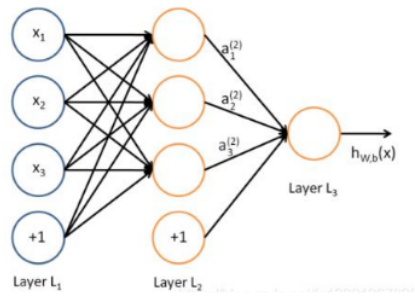


Figure 1. Diagram of an MLP

MSE

This statistical parameter is the mean of the sum of squared errors of the corresponding points of the predicted and original data.

$$MSE = \frac{SSE}{n} = \frac{1}{n} \sum_{i=1}^m w_i (y_i - \hat{y}_i)^2 \quad (1)$$

Confusion matrix

The confusion matrix, also called the error matrix, represented by n rows and n columns, is the standard form of accuracy evaluation. Specific metrics include total accuracy, mapping accuracy, and user accuracy. These precision metrics reflect the accuracy of image classification in various aspects.

In artificial intelligence, the confusion matrix is a visualization tool, especially for supervised learning.[3] In unsupervised learning, this is known as the matching matrix. The confusion matrix is computed by comparing the position and classification of each measured pixel to the corresponding position and classification in the classification image.

Results

Scenario 1 The random forest classification algorithm was used to train the Fashion-MNIST dataset to tune the hyperparameters.

Step 1: Increase n_estimators(10-290) by 10 each iteration , finding the maximum accuracy

n_estimators	10	20	30	40	50	60	70	80	90
Accuracy(%)	85.48	86.42	86.64	87.07	87.25	87.41	87.52	87.60	87.51

100	110	120	130	140	150	160	170	180	190
87.63	87.66	87.61	87.61	87.64	87.79	87.72	87.74	87.67	87.64

200	210	220	230	240	250	260	270	280	290
87.76	87.79	87.77	87.76	87.73	87.71	87.73	87.82	87.78	87.78

Table 1. Accuracy for increase n_estimators(10-290) by 10 each iteration

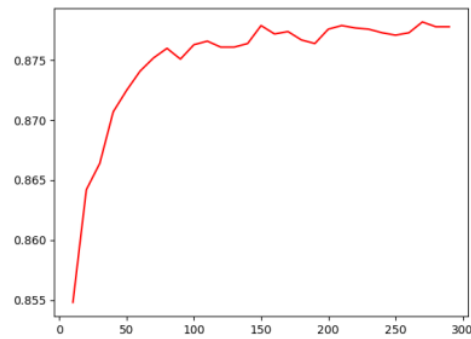


Figure 2. Accuracy for increase n_estimators(10-290) by 10 each iteration

Step 2: Increase n_estimators(260-280) by 1 each iteration , finding the maximum accuracy

n_estimators	260	261	262	263	264	265	266	267	268	269
Accuracy(%)	87.73	87.74	87.67	87.76	87.74	87.79	87.81	87.77	87.81	87.82

270	271	272	273	274	275	276	277	278	279
87.82	87.79	87.82	87.78	87.75	87.77	87.79	87.72	87.73	87.77

Table 2. Accuracy for increase n_estimators(260-279) by 1 each iteration

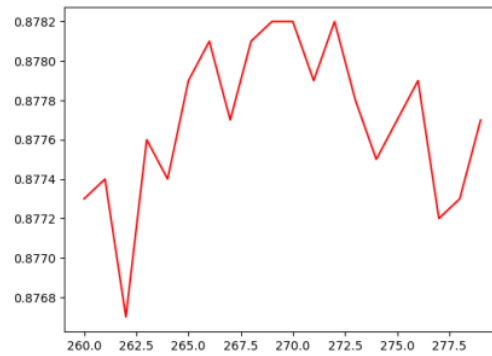


Figure 3. Accuracy for increase n_estimators(260-279) by 1 each iteration

Step 3: Increase max_depth (0-30) by 1 each iteration, finding the maximum accuracy

max_depth	1	2	3	4	5	6	7	8	9	10
Accuracy(%)	37.16	58.19	67.93	74.19	76.07	78.98	80.63	82.23	83.70	84.58

11	12	13	14	15	16	17	18	19	20
85.32	85.81	86.34	87.66	87.17	87.34	87.55	87.55	87.45	87.56

21	22	23	24	25	26	27	28	29
87.83	87.74	87.89	87.81	87.82	87.74	87.87	87.79	87.85

Table 3. Accuracy for increase max_depth (0-30) by 1 each iteration

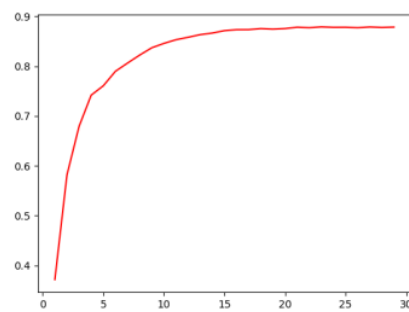


Figure 4. Accuracy for increase max_depth (0-30) by 1 each iteration

Step 4: Find best model

Best RF Accuracy: 0.8789

	precision	recall	f1-score	support
0	0.82	0.86	0.84	1000
1	0.99	0.96	0.98	1000
2	0.77	0.81	0.79	1000
3	0.87	0.91	0.89	1000
4	0.78	0.82	0.80	1000
5	0.98	0.96	0.97	1000
6	0.73	0.59	0.65	1000
7	0.93	0.95	0.94	1000
8	0.96	0.97	0.96	1000
9	0.95	0.95	0.95	1000
accuracy			0.88	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.88	0.88	10000

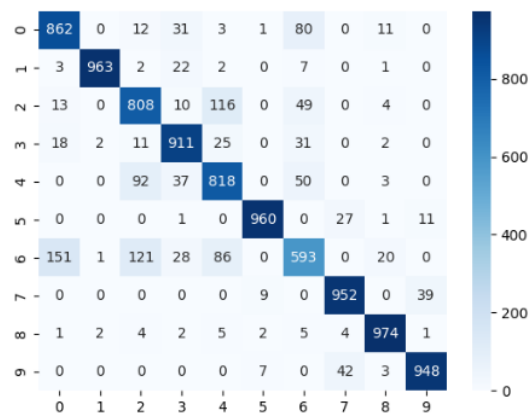


Figure 5. Confusion matrix of best model

Scenario 2 The random forest classification algorithm was used to train the cifar-10 dataset to tune the hyperparameters.

Step 1: Increase n_estimators(10-290) by 10 each iteration, finding the maximum

accuracy

n_estimators	10	20	30	40	50	60	70	80	90	100
Accuracy(%)	35.70	40.67	42.62	43.88	44.42	45.22	46.07	46.61	46.94	47.52

110	120	130	140	150	160	170	180	190
47.6	47.63	47.74	48.04	48.02	48.10	47.99	47.94	48.12

200	210	220	230	240	250	260	270	280	290
48.30	48.26	48.15	48.15	48.14	48.22	48.22	48.60	48.42	48.43

Table 4. Accuracy for increase n_estimators(10-290) by 10 each iteration

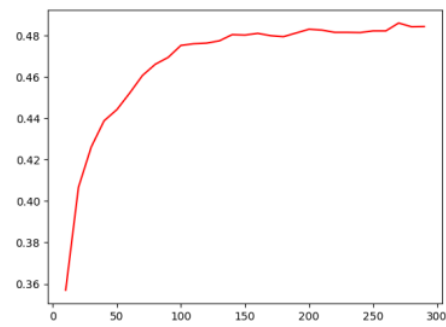


Figure 6. Accuracy for increase n_estimators(10-290) by 10 each iteration

Step 2: Increase n_estimators(260-280) by 1 each iteration , finding the maximum accuracy

n_estimators	260	261	262	263	264	265	266	267	268	269
Accuracy(%)	48.22	48.29	48.35	48.46	48.46	48.43	48.54	48.53	48.42	48.5

270	271	272	273	274	275	276	277	278	279
48.6	48.54	48.47	48.45	48.45	48.45	48.46	48.42	48.47	48.49

Table 5. Accuracy for increase n_estimators(260-280) by 10 each iteration

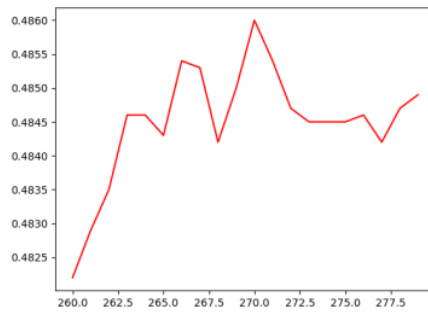


Figure 7. Accuracy for increase n_estimators(260-280) by 10 each iteration

Step 3: Increase max_depth (0-30) by 1 each iteration, finding the maximum accuracy

max_depth	1	2	3	4	5	6	7	8	9	10
Accuracy(%)	16.07	26.44	29.33	31.68	33.85	36.01	37.69	39.82	41.72	43.54

11	12	13	14	15	16	17	18	19	20
44.61	45.17	46.26	46.52	47.21	47.31	47.76	47.95	48.06	47.82

21	22	23	24	25	26	27	28	29
47.95	48.28	48.22	48.83	48.46	48.53	48.87	48.59	48.56

Table 6. Accuracy for increase max_depth (0-30) by 1 each iteration

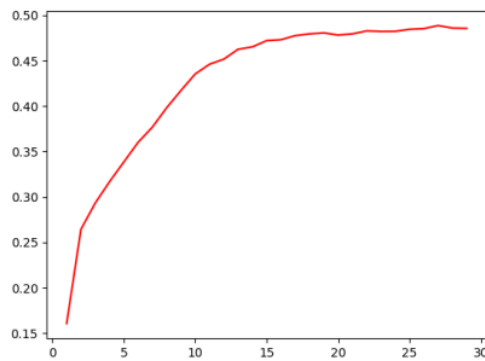


Figure 8. Accuracy for increase max_depth (0-30) by 1 each iteration

Step 4: Find best model

Best RF Accuracy: 0.4887

	precision	recall	f1-score	support
0	0.57	0.57	0.57	1000
1	0.54	0.57	0.56	1000
2	0.42	0.33	0.37	1000
3	0.37	0.28	0.32	1000
4	0.41	0.41	0.41	1000
5	0.44	0.41	0.42	1000
6	0.48	0.61	0.53	1000
7	0.53	0.48	0.51	1000
8	0.59	0.64	0.61	1000
9	0.48	0.58	0.53	1000
accuracy			0.49	10000
macro avg	0.48	0.49	0.48	10000
weighted avg	0.48	0.49	0.48	10000

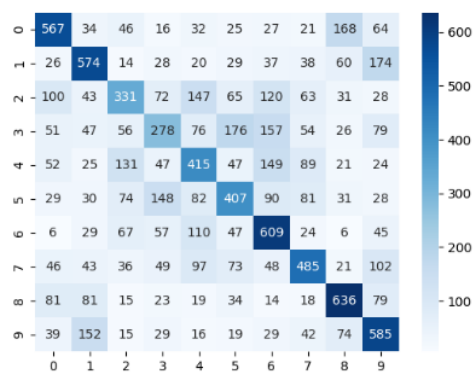


Figure 9. Confusion matrix of best model

Multilayer Perceptron

Scenario 1 The MLP(Multilayer Perceptron) algorithm was used to train the Fasion-MNIST dataset to tune the hyperparameters.

Results obtained from multiple trials and tuning hyperparameters

MLPClassifier(

```

solver='sgd',
activation='relu',
alpha=1e-4,
hidden_layer_sizes=(200,100,50),
random_state=1,
max_iter=50,
verbose=10,
learning_rate_init=.1)

```

Best model

iterations	accracy	loss	MSE
45	0.9838	0.0004341	0.2837

Table 7. Fashion-MNIST best model

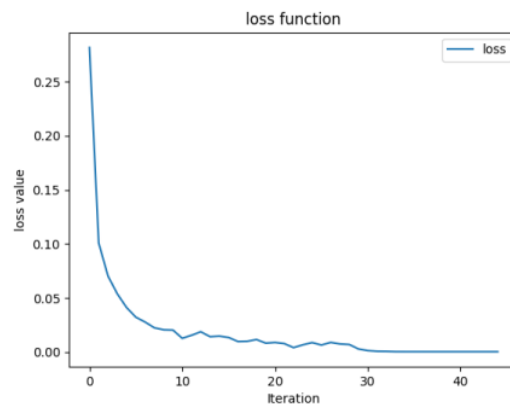


Figure 10. Fashion-MNIST loss function

Scenario 2 The MLP(Multilayer Perceptron) algorithm was used to train the cifar-10 dataset to tune the hyperparameters.

Results obtained from multiple trials and tuning hyperparameters

```

activation='relu',
alpha=0.0001,
hidden_layer_sizes=(200,100,50),
random_state=1,
max_iter=500,

```

verbose=10,

learning_rate_init=0.001)

Best model

iterations	accracy	loss	MSE
493	0.9838	0.0806818	9.4332

Table 8. Cifar-10 best model

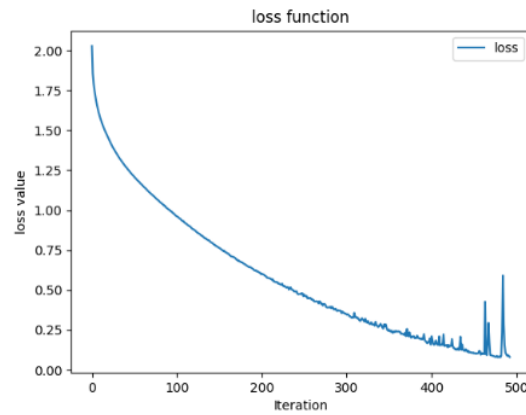


Figure 11. Cifar-10 loss function

Discussion

When using the random forest classification algorithm, only the estimator is modified when random_rate=30 is fixed and the other hyperparameters are default values. We find that although the accuracy fluctuates between 0.5% and does not show a particularly pronounced increase in the case of gradual growth, the maximum accuracy value appears to be between 260 and 270. Once we have found the best value for the estimator, we can only change the depth, which is between 0 and 17. The accuracy steadily increases with depth, oscillating to convergence within 1% between phases 18-30. For the MNIST-Fashion dataset, the random forest classifier found relatively good models as follows: 269 estimators with depth 17 and accuracy 87.89% ; The relatively good model found for the random forest classifier on the cifar10 dataset is: with 269 estimators, depth 27, and accuracy 48.87%. For the Fasion-MNIST dataset, random

forest has a relatively excellent effect, but the effect of the Cifar-10 dataset is more than 10% (the probability of ten categories being randomly selected), but the improvement effect is limited.

When using the Multilayer Perceptron algorithm for model training on the Fashion-MNIST dataset, it can be seen that after ten iterations of training, the loss can be reduced below 0.04 and the accuracy rate is roughly above 95 percent. It can be seen that the Multilayer Perceptron algorithm has a strong ability to extract features. In the subsequent training process, it still oscillates and converges during 10-30 iterations, and basically converges after 30 iterations. The final training was stopped at 45 iterations with 98.38% accuracy. For training on the Cifar-10 dataset, the effect is found to be very close to the accuracy of random forest, but it does not achieve a very desirable effect after many training epochs. After 500 iterations, it finally stopped at 469 iterations due to a loss of 0.0001, with an accuracy of 49.86%, slightly better than the random forest classification algorithm. A 10% or so improvement like this is achieved for the majority of the datasets.

Problem: I believe the problem is that the Cifar-10 dataset consists of RGB-dimensional arrays, but for the input it was changed to a 1D array, resulting in some data feature values being hard to find; In terms of diversity of data, the Cifar-10 dataset contains both animals and vehicles, which will lead to similar feature values for certain types of data and hence lower accuracy. In the process of using the Multilayer Perceptron algorithm, the loss value will suddenly rise and then fall back after the shock has converged. At present, some information has been consulted and no relevant circumstances have arisen.

Conclusion

Pros and Cons of Random Forests

Pros:

1. Because it is an ensemble algorithm, the model accuracy is often higher than that of a single decision tree.
2. Randomly selecting samples and features at each time improves the adversarial

interference capability of the model and leads to stronger generalization.

3. Strong adaptability to datasets, ability to handle discrete and missing data, and low data normalization requirements;
4. In each random sample selection, 1/3 of the samples are not selected. This part of samples is usually called OOB(out of bag), which can be directly used as the validation set without occupying the training data.
5. It can produce highly accurate classifiers for many types of data.
6. It can handle a large number of input variables;
7. It assesses the importance of a variable in identifying a class;
8. When a forest is constructed, it can internally produce an unbiased estimate of the generalization error.

Cons:

1. When the number of decision trees is large, training takes a long time.
2. This model is not very interpretable and belongs to the black-box class of models.

Pros and Cons of Multilayer Perceptron

Pros:

1. highly parallel processing;
2. non-linear advanced global action;
3. good fault tolerance;
4. associative memory;
5. very strong adaptive and self-learning function.

Cons:

1. the choice of the number of implicit nodes in the network is very difficult;
2. stop thresholds, learning rates, and exercise metrics need to be "trial-and-error", which is very time consuming;
3. slow learning speed;
4. easy to fall into local extreme value;
5. may not study enough.

Comparison between Random Forests and Multilayer Perceptron

First, Random Forests and Multilayer Perceptron are different classes of algorithms.

Random Forests is a collection of decision trees, each decision tree processes samples and predicts output labels (for classification). A neural network is a network of neurons that do not function without other neurons. Grouped together, the data at each level is processed and passed to the next level, with the last neuron making the decision.

Reference

- [1] C. Li and D. Ai, "Automatic Crack Detection Method for loaded coal in vibration failure process," PLOS ONE, vol. 12, no. 10, 2017.
- [2] D.-W. Chen, R. Miao, W.-Q. Yang, Y. Liang, H.-H. Chen, L. Huang, C.-J. Deng, and N. Han, "A feature extraction method based on differential entropy and linear discriminant analysis for emotion recognition," Sensors, vol. 19, no. 7, p. 1631, 2019.
- [3] Z. Tian and S. Fong, "Survey of meta-heuristic algorithms for deep learning training," Optimization Algorithms - Methods and Applications, 2016.

Classification using ML methods and Neural Networks

GRADEMARK REPORT

FINAL GRADE

65 / 100

GENERAL COMMENTS

Instructor

Your abstract should provide a concise summary of your aims, methods, results and conclusions.

Your report is structured but you need to do one final check to improve formatting and remove typos/grammar issues to (minor).

Your related work section should be more detailed and don't be afraid to include additional references here. You used only 3 throughout.

Your experiment/method shows you have be quite thorough here. Consider dimensionality reduction, etc in pre-processing.

When presenting results use bar graphs, etc. and tables.

Also, for MLP provide equivalent metrics to those given for the random forest - ie confusion matrices and related PER-CLASS metrics.

Overall a good effort but see comments above especially regarding related work.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

GRADING FORM: SMU GENERIC_GJ

XINZHE LYU

0

RESEARCH-INFORMED LITERATURE

KNOWLEDGE AND UNDERSTANDING

ANALYSIS

PRACTICAL APPLICATION AND DEP.

SKILLS FOR PROFESSIONAL PRAC.