**Assignment 2**

Machine Learning, SS23

|  |  |  |
| --- | --- | --- |
| **Team members** | | |
| Last name | First name | Matriculation Number |
| Heng | Chloe Yi Ning | 12239126 |
| Lim | Yan Qi | 12239271 |

# Neural Networks

## PCA Classification

### PCA for dimensionality reduction

N components used:

Variance explained:

### Varying the number of neurons

For each n hid, report the accuracy on the train and test set, and the loss.

Answer the questions: How do we know (in general) if the model does not have enough capacity (the case of underfitting)? How do we know (in general) if the model starts to overfit? Does that happen with some architectures/models? (If so, say with what number of neurons that happens). Which model would you choose here and why?

Then (using a setup as in (a), (b), or (c)), report the train and test accuracy, and the loss for n hidden ∈ {2, 10, 100, 200}. Does this improve the results from the previous step? Which model would you choose now?

### Variability of the performance

How could you calculate yourself recall from support and confusion matrix entries? Explain in words what is recall. What is the most misclassified image? State which class (digit) it was, and how you concluded that.

## Model selection using GridSearch

Number of architectures that will be checked:

Best parameter:

{'activation': 'relu', 'alpha': 10, 'hidden\_layer\_sizes': 200, 'solver': 'adam'}

Best score: 0.6573657548125633

Train accuracy: 0.9406. Test accuracy: 0.7070

What is the difference between hyperparameters and parameters of a model (in general)? Explain then the difference using the example of neural networks (i.e., name a few hyperparameters and parameters of neural networks)

Parameters are values that the model learns on its own. Hyperparameters are parameters set explicitly by the user to control the model. The number of layers and units, learning rate are considered hyperparameters of neural networks while weights, biases are considered as parameters.

# Regression with Neural Networks

Function *calculate\_mse*:

def calculate\_mse(targets, predictions):

    """

    :param targets:

    :param predictions: Predictions obtained by using the model

    :return:

    """

    mse = mean\_squared\_error(y\_pred=predictions, y\_true=targets)

    return mse

Using GridSearch

Parameters:

{'alpha': [0.002, 0.003, 0.004], 'hidden\_layer\_sizes': n\_hidden\_neurons\_list,

'learning\_rate\_init: [0.0003, 0.0005, 0.0008], 'activation': ['relu', 'logistic'], 'solver': ['adam', 'lbfgs', 'sgd']}

n\_hidden\_neurons\_list = [40, 48, 56]

Best parameter:

{'activation': 'logistic', 'alpha': 0.002, 'hidden\_layer\_sizes': 56, 'learning\_rate\_init': 0.0003, 'solver': 'lbfgs'}

Best score: 0.9989232429730069

Train MSE: 0.0004. Test MSE: 0.0004